



TUGAS AKHIR - KM184801

**IMPLEMENTASI *SOCIAL MEDIA MINING* UNTUK
PENGAMBILAN KEPUTUSAN DALAM PERENCANAAN
PRODUK BERDASARKAN PEMODELAN TOPIK DAN
ANALISIS SENTIMEN PADA TWITTER**

RIKO WIJAYANTO
NRP. 0611154000010

Dosen Pembimbing:
Prof. Dr. Mohammad Isa Irawan, M.T.
Muhammad Luthfi Shahab, S. Si., M. Si.

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



FINAL PROJECT - KM184801

***IMPLEMENTATION OF SOCIAL MEDIA MINING FOR
DECISION MAKING IN PRODUCT PLANNING BASED ON
TOPIC MODELLING AND SENTIMENT ANALYSIS ON
TWITTER***

RIKO WIJAYANTO

Student Number. 0611154000010

Supervisors:

Prof. Dr. Mohammad Isa Irawan, M.T.

Muhammad Luthfi Shahab, S. Si., M. Si.

DEPARTMENT OF MATHEMATICS

Faculty of Mathematics, Computing, and Data Sciences

Institut Teknologi Sepuluh Nopember

Surabaya 2019

LEMBAR PENGESAHAN
IMPLEMENTASI *SOCIAL MEDIA MINING* UNTUK
PENGAMBILAN KEPUTUSAN DALAM PERENCANAAN
PRODUK BERDASARKAN PEMODELAN TOPIK DAN
ANALISIS SENTIMEN PADA TWITTER
IMPLEMENTATION OF SOCIAL MEDIA MINING FOR
DECISION MAKING IN PRODUCT PLANNING BASED
ON TOPIC MODELLING AND SENTIMENT ANALYSIS
ON TWITTER
TUGAS AKHIR

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

Oleh:

RIKO WIJAYANTO
NRP. 0611154000010
Menyetujui,

Dosen Pembimbing II



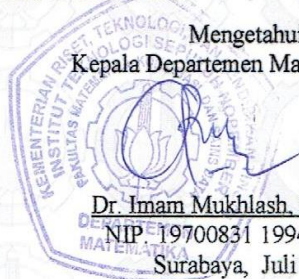
Muhammad Luthfi Shahab, S. Si., M.Si.
NIP. 19950331 201803 1 001

Dosen Pembimbing I



Prof. Dr. Mohammad Isa Irawan, M.T.
NIP. 19631225 198903 1 001

Mengetahui,
Kepala Departemen Matematika ITS



Dr. Imam Mukhlash, S.Si., M.T.
NIP. 19700831 199403 1 003
Surabaya, Juli 2019

**IMPLEMENTASI *SOCIAL MEDIA MINING* UNTUK
PENGAMBILAN KEPUTUSAN DALAM PERENCANAAN
PRODUK BERDASARKAN PEMODELAN TOPIK DAN
ANALISIS SENTIMEN PADA TWITTER**

Nama Mahasiswa : RIKO WIJAYANTO
NRP : 0611154000010
Departemen : Matematika ITS
Dosen Pembimbing : Prof. Dr. Mohammad Isa Irawan, M.T.
M. Luthfi Shahab, S. Si., M. Si.

ABSTRAK

Perkembangan teknologi informasi dan komunikasi, khususnya dalam bidang media sosial, membawa suatu perubahan besar pada bidang perencanaan produk dari suatu perusahaan yang menaungi. Saat ini, perusahaan dapat mengidentifikasi peluang untuk perencanaan produk dari opini pelanggan yang disampaikan melalui *posting* Twitter. Peluang ini dapat dimanfaatkan untuk meningkatkan daya saing terhadap kompetitor. Metode utama yang digunakan adalah *social media mining* melalui pemodelan topik dan analisis sentimen. Topik yang sedang ramai dibicarakan pelanggan diidentifikasi dengan derajat *importance*, sedangkan hasil analisis sentimen diidentifikasi dengan derajat *satisfaction*. Pemodelan topik dengan *Latent Semantic Analysis* dan K-Means serta analisis sentimen berdasarkan daftar *lexicon* dari C. J. Hutto dan Eric Gilbert berhasil diterapkan dengan baik serta dikemas dalam sebuah perangkat lunak berbasis bahasa pemrograman Python. Studi kasus pada McDonald's dengan memanfaatkan 2000 data *tweet*, didapatkan suatu hasil bahwa “pelayanan di pagi hari” memiliki *opportunity* tertinggi dan berada dalam wilayah *under-served*. Ini memberikan arti bahwa “pelayanan di pagi hari” merupakan salah

satu layanan yang memiliki peluang bagus di pasar. Dengan demikian, McDonald's dapat mempertimbangkan perencanaan pelayanan di pagi hari.

Kata Kunci : Social Media Mining, Pemodelan Topik, Latent Semantic Analysis (LSA), K-Means, Analisis Sentimen, Opportunity Mining

**IMPLEMENTATION OF SOCIAL MEDIA MINING FOR
DECISION MAKING IN PRODUCT PLANNING BASED
ON TOPIC MODELLING AND SENTIMENT ANALYSIS
ON TWITTER**

Student Name : RIKO WIJAYANTO
Student Number : 0611154000010
Department : Mathematics ITS
Supervisors : Prof. Dr. Mohammad Isa Irawan, M.T.
M. Luthfi Shahab, S. Si., M. Si.

ABSTRACT

The development of information and communication technology, especially in the social media, brought an important change of product planning from an overshadowed company. Recently, using the the text mining and social media mining, the companies can identify opportunities for product planning based on customer's opinion from Twitter's postings to increase their competitiveness with competitors. The hottest topic and the customers' satisfaction can be identified by topic modelling and sentiment analysis. The hottest topic is acquired as the importance degree and the result of sentiment analysis is acquired as the satisfaction degree. Topic modelling with Latent Semantic Analysis and K-Means cluster, also sentiment analysis based on lexicon list from C. J. Hutto and Eric Gilbert, succesfully implemented and packaged in a build-based using Python programming language. Case studies at McDonald's by utilizing 2000 tweets, a result was obtained that "operational in morning day" had the highest opportunity value in under-served area. It means that "operational in morning day" has a nice opportunity at the market. As such, McDonald's can consider following up on the proposes thing that is "operational in morning day".

Keywords : ***Social Media Mining, Topic Modelling, Latent Semantic Analysis (LSA), K-Means, Sentiment Analysis, Opportunity Mining***

KATA PENGANTAR

Puji syukur dan terima kasih yang luar biasa kepada Tuhan Yang Maha Esa dan Maha Baik yang mana telah memberikan keberkahan dan juga rezeki-Nya yang tiada tara sehingga penulis dapat menyelesaikan Tugas Akhir ini yang berjudul “Implementasi *Social Media Mining* untuk Pengambilan Keputusan dalam Perencanaan Produk berdasarkan Pemodelan Topik dan Analisis Sentimen pada Twitter” dengan baik. Tugas Akhir ini adalah bagian penting dari rangkaian studi penulis, yakni sebagai salah satu syarat untuk menggenapi proses belajar penulis selama di jenjang sarjana.

Terselesaikannya Tugas Akhir ini adalah salah satu hal luar biasa yang penulis lakukan hingga saat ini. Banyak sekali pengalaman, suka duka, dan juga ilmu baru yang sudah penulis dapatkan mulai saat perencanaan Tugas Akhir, saat pengerjaan, hingga penulis menyelesaikan laporan Tugas Akhir ini. Selain itu, terselesaikannya Tugas Akhir dan juga laporan ini tidak pernah luput dari bantuan serta dukungan banyak pihak. Untuk itu, penulis ucapkan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Bapak Dr. Imam Mukhlash, S. Si., M.T. selaku Kepala Departemen Matematika ITS,
2. Bapak Prof. Dr. Mohammad Isa Irawan, M.T. dan Bapak Muhammad Luthfi Shahab, S. Si., M. Si. selaku dosen pembimbing Tugas Akhir penulis yang sudah meluangkan banyak sekali waktu untuk membimbing penulis dan terima kasih atas dedikasinya untuk kebaikan penulis,
3. Bapak Dr. Budi Setiyono, S.Si., M.T., Ibu Dr. Rinurwati, dan Ibu Alvida Mustika Rukmi, M. Si. selaku dosen penguji Tugas Akhir penulis yang telah memberikan masukan, arahan, dan juga saran-saran yang sangat berguna bagi penulis,

4. Bapak Dr. Dieky Adzkiya, S. Si., M. Si. atas kesabaran, kebaikan, dan perhatiannya selama menjadi dosen wali penulis dari mahasiswa baru hingga lulus saat ini,
5. 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 banyak membantu dan menanggapi kebutuhan penulis,
6. Seluruh Bapak/Ibu dosen dan seluruh staf Departemen Matematika ITS,
7. Pak Sonny, Ibu Maria, Mbah Toto, Adji, Djati, Dimar, Mbak Lika, Mbak Fai, Debi, dan seluruh keluarga Blok U-137 yang selalu mendukung dan selalu ada untuk penulis,
8. Teman-teman Matematika ITS angkatan 2015, yang selalu mendukung dan menjadi keluarga penulis selama kuliah, khususnya Andrew Wiles,
9. Teman-teman rapat dan organisasi penulis, HIMATIKA ITS, LMB ITS, BEM FMKSD ITS, BEM ITS, teman-teman pemandu ITS, Cakrabaskara (Paduka I Made Kusnanta khususnya),
10. Teman-teman asisten Kalkulus hingga Matematika, asisten Lab. Komputasi hingga asisten mata kuliah yang lain, *thank you all*,
11. Plat AG Squad, *my best final partner* (Mbekay, Lilik, Safir, Dasilva), *my gils partner* (Hayu, Lina, Tommy, Komandan, Mail, Inay, Nenek Dori, Sumidik, Komtiang, Ijul, Ami, Nadiya, Amira, Ryan, Vira, dll). *thanks a lot, guyss*,
12. Berikutnya, ucapan terima kasih tak terhingga kepada *my late dad* di surga, walaupun perjumpaan kita hanya sesaat, tetapi *I will always miss and gratefull having you in my life, I believe that you always look at me there*. Terima kasih tak terhingga batasnya pada *my beloved and strongest mom*. Dunia pun

mengakui bahwa kehidupan kita sangat berat, berbagai keadaan yang serba minus, bahkan titik terendah kehidupan sudah kita lalui berdua pada saat itu, *but you said to me “the storm will go one day and you have to continue your life, don’t worry about me and your future just a milimeter in front of your eyes. All the things are possible”, thank you for giving me a lot of chances specially to go university, and now I can make you believe me, I’m graduated,*

13. Semua pihak yang tidak dapat penulis sebutkan namanya satu per-satu, terima kasih sudah hadir dalam kehidupan penulis dan memberikan banyak sekali pelajaran untuk penulis.

Akhir kata, semoga Tugas Akhir ini dapat dijadikan sumber pembelajaran dan bermanfaat untuk semua, dan dapat dijadikan sebagai salah satu karya yang berguna. Penulis sangat terbuka akan diskusi dan kritik serta saran yang membangun.

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	Error! Bookmark not defined.
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR LAMPIRAN	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	5
1.4 Tujuan.....	5
1.5 Manfaat.....	6
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA.....	9
2.1 Penelitian Terdahulu.....	9
2.2 Media Sosial	11
2.3 <i>Term Frequency-Inverse Document Frequency</i>	13
2.4 Pemodelan Topik.....	15
2.5 Analisis Sentimen.....	21
2.5 <i>Opportunity Mining</i>	26
2.6 <i>Word Intrusion Task</i>	29
BAB III METODOLOGI PENELITIAN	35
3.1 Obyek dan Aspek Penelitian	35
3.2 Peralatan	35
3.3 Tahapan Penelitian	36
3.4 Diagram Blok	39

BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM	41
4.1 Analisis Sistem	41
4.1.1 Analisis Kebutuhan Fungsional.....	42
4.1.2 Analisis Kebutuhan Non-Fungsional.....	43
4.2 Perancangan Sistem.....	43
4.2.1 Perancangan Data	43
4.2.2 Perancangan Proses	46
4.2.3 Perancangan Desain Antarmuka.....	72
4.3 Implementasi Sistem	74
4.3.1 Implementasi Proses Impor Data.....	74
4.3.2 Implementasi Praproses Data	74
4.3.3 Implementasi Penyusunan <i>Sparse Matrix</i>	76
4.3.4 Implementasi Pemodelan Topik	77
4.3.5 Implementasi Analisis Sentimen	80
4.3.6 Implementasi <i>Opportunity Mining</i>	96
4.3.7 Implementasi Tombol Simpan.....	101
4.3.8 Implementasi Tombol <i>Load Model</i>	102
4.3.9 Implementasi Menu <i>Details</i>	103
4.3.10 Implementasi Desain Antarmuka	113
BAB V UJI COBA DAN PEMBAHASAN	115
5.1 Deskripsi Data Uji Coba.....	115
5.2 Proses Uji Coba	116
5.2.1 Uji Coba Impor Data	116
5.2.2 Uji Coba Praproses Data	117
5.2.3 Uji Coba Penyusunan <i>Sparse Matrix</i>	118
5.2.4 Uji Coba Pemodelan Topik	119
5.2.5 Uji Coba Analisis Sentimen	123
5.2.6 Uji Coba <i>Opportunity Mining</i>	124
5.2.7 Uji Coba Desain Antarmuka	125
5.3 Penentuan Skenario dan Uji Coba	126
5.3.1 Uji Coba Skenario I.....	128

5.3.2	Uji Coba Skenario II.....	133
5.4	Pemilihan Skenario Terbaik dan Pembahasan	139
5.4.1	<i>Word Intrusion Task</i>	139
5.4.2	Pembahasan Analisis Sentimen.....	143
5.4.3	Perincian <i>Opportunity Mining</i>	145
5.4.4	Pembahasan Keputusan.....	152
BAB VI KESIMPULAN DAN SARAN.....		155
6.1	Kesimpulan.....	155
6.2	Saran.....	156
DAFTAR PUSTAKA.....		159
LAMPIRAN		
BIODATA PENULIS		

DAFTAR GAMBAR

Gambar 2.1	Ilustrasi <i>Singular Value Decomposition</i>	16
Gambar 2.2	Ilustrasi <i>Truncated SVD</i>	17
Gambar 2.3	Diagram <i>Product Landscape Map</i>	28
Gambar 2.4	Gambar <i>Word Intrusion Task</i>	29
Gambar 4.1	Contoh Data Masukan.....	44
Gambar 4.2	Grafik Nilai <i>Inertia</i>	60
Gambar 4.3	Kutipan Daftar <i>Lexicon</i>	63
Gambar 4.4	Diagram <i>Product Landscape Map</i>	72
Gambar 4.5	Perancangan Desain Antarmuka	73
Gambar 4.6	Tampilan Desain Antarmuka Halaman Utama ..	113
Gambar 4.7	Tampilan Desain Antarmuka Halaman Details..	114
Gambar 5.1	Gambar Kutipan Data <i>Tweet</i>	115
Gambar 5.2	<i>Dataframe</i> dari Data <i>Tweet</i>	117
Gambar 5.3	List dari Data <i>Tweet</i>	117
Gambar 5.4	List Hasil Praproses Data.....	118
Gambar 5.5	(a) Daftar <i>Stopwords</i> , (b) Hasil N-gram.....	118
Gambar 5.6	Kutipan <i>Sparse Matrix</i>	119
Gambar 5.7	Kutipan Matriks <i>S</i>	119
Gambar 5.8	Kutipan Matriks V^T	120
Gambar 5.9	Kutipan Nilai-Nilai Singular	120
Gambar 5.10	(a) Nilai <i>Sum Explained Variance Ratio</i> , (b) Plot Grafik <i>Sum Explained Variance Ratio</i>	121
Gambar 5.11	Kutipan Matriks <i>Tweet-Principal Components</i>	121
Gambar 5.12	(a) Nilai <i>Inertia</i> , (b) Plot Nilai <i>Inertia</i>	122
Gambar 5.13	Kutipan Hasil Klaster.....	122
Gambar 5.14	Hasil Pemodelan Topik.....	123
Gambar 5.15	Kutipan Hasil Uji Coba Analisis Sentimen Pada <i>Tweet</i>	123

Gambar 5.16 Kutipan Hasil Analisis Sentimen pada Klaster	124
Gambar 5.17 Hasil <i>Opportunity Mining</i>	124
Gambar 5.18 Kutipan Nilai Derajat Hasil <i>Opportunity Mining</i>	125
Gambar 5.19 Gambar Uji Coba Desain Antarmuka	125
Gambar 5.20 Plot Grafik Nilai <i>Sum Explained Variance Ratio</i>	127
Gambar 5.21 Plot Grafik Nilai <i>Inertia</i> Skenario I	128
Gambar 5.22 Visualisasi <i>Product Landscape Map</i> Skenario I	132
Gambar 5.23 Plot Grafik Nilai <i>Inertia</i> Skenario II.....	133
Gambar 5.24 Visualisasi <i>Product Landscape Map</i> Skenario II	138
Gambar 5.25 Grafik Perbandingan Nilai WIT.....	140
Gambar 5.26 Persentase Persebaran Hasil Analisis Sentimen oleh Sistem.....	144
Gambar 5.27 Persentase Persebaran Hasil Analisis Sentimen secara Aktual	144

DAFTAR TABEL

Tabel 2.1	Tabel Bobot <i>Tweet</i> -Fitur.....	14
Tabel 2.2	Tabel <i>Tweet-Principal Components</i>	17
Tabel 2.3	Kutipan Daftar <i>Lexicon</i> Hutto dan Eric	22
Tabel 2.4	Tabel Pengamatan RAL.....	31
Tabel 2.5	Tabel ANOVA Satu Faktor dalam RAL	32
Tabel 3.1	Tabel Peralatan yang Digunakan	35
Tabel 4.1	Tabel Data Proses	45
Tabel 4.2	Tabel Data Fiktif <i>Tweet</i>	47
Tabel 4.3	Contoh Data Sebelum dan Sesudah Praproses	48
Tabel 4.4	Contoh Data Sebelum dan Sesudah Seleksi	49
Tabel 4.5	Contoh Data Sebelum dan Sesudah N-gram	50
Tabel 4.6	Tabel TF pada <i>Tweet</i> Pertama	52
Tabel 4.7	Tabel IDF pada <i>Tweet</i> Pertama.....	53
Tabel 4.8	Tabel TF-IDF <i>Tweet</i> Pertama	53
Tabel 4.9	Tabel Bobot Akhir pada <i>Tweet</i> Pertama	54
Tabel 4.10	Tabel Hasil Klasterisasi Data Fiktif.....	61
Tabel 4.11	Tabel Hasil Pemodelan Topik	62
Tabel 4.12	Tabel Hasil Analisis Sentimen.....	69
Tabel 4.13	Tabel Hasil Perhitungan Awal.....	70
Tabel 4.14	Tabel Kesimpulan <i>Opportunity Mining</i>	71
Tabel 5.1	Jumlah Data <i>Tweet</i> per-Tanggal	116
Tabel 5.2	Tabel Nilai <i>Sum Explained Variance Ratio</i>	126
Tabel 5.3	Hasil Pemodelan Topik Skenario I.....	129
Tabel 5.4	Hasil Pemodelan Topik Skenario II.....	134
Tabel 5.5	Tabel ANOVA Rancangan Acak Lengkap.....	142
Tabel 5.6	Hasil Perhitungan <i>Opportunity Mining</i>	146
Tabel 5.7	Pelabelan Hasil Pemodelan Topik	147
Tabel 5.8	Pemetaan Posisi Bagian Produk	151

DAFTAR LAMPIRAN

- LAMPIRAN A.** Tabel Perbandingan Penelitian Terdahulu
- LAMPIRAN B.** Perhitungan *Singular Value Decomposition*
- LAMPIRAN C.** Kuesioner *Word Intrusion Task*
- LAMPIRAN D.** Hasil Penilaian *Word Intrusion Task*
- LAMPIRAN E.** Perhitungan Uji Wilcoxon
- LAMPIRAN F.** Tabel Pengamatan Uji ANOVA

BAB I

PENDAHULUAN

Pada bab ini diuraikan latar belakang masalah Tugas Akhir ini, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan laporan Tugas Akhir ini.

1.1 Latar Belakang

Opini pelanggan pada suatu produk perlu mendapat perlakuan khusus dari perusahaan yang mempromosikannya. Opini yang dimaksud dapat berupa ulasan, keluhan, kesan, dan juga klarifikasi. Perusahaan dapat memanfaatkan opini yang disampaikan pelanggan untuk meningkatkan layanan atau memberikan inovasi pada produknya bila dianalisis lebih lanjut [1]. Pendekatan dalam sektor perencanaan dan pengembangan produk melalui analisis opini pelanggan adalah salah satu upaya yang dapat dilakukan perusahaan dalam rangka mengetahui bagaimana suatu peningkatan layanan dan inovasi dibutuhkan oleh pelanggan serta bagaimana peluangnya. Identifikasi peluang produk dengan analisis opini pelanggan dapat dijadikan suatu hal pokok yang mendasari pengembangan suatu produk atau proses peningkatan mutu produk [2]. Hal ini karena identifikasi peluang suatu produk memungkinkan perusahaan untuk mengembangkan sesuatu yang unik, dibutuhkan oleh pelanggan, serta tepat pada bagian yang menjadi fokus opini pelanggan [2].

Pada masa yang serba digital seperti sekarang ini, pelanggan dapat menyampaikan opini melalui berbagai media sosial. Media sosial seperti Blogger, Twitter, Facebook, Reddit, dan juga situs ulasan secara daring banyak digunakan karena akses yang mudah. Perusahaan dapat memanfaatkan media sosial tersebut karena data yang tersedia selalu mengalami pembaharuan (*real-time data*) dan dapat berasal dari segala lapisan [3]. Jumlah pengguna

media sosial saat ini dinilai sangat besar yakni mencapai 3,196 miliar¹. Dengan jumlah pengguna internet di seluruh dunia yang mencapai angka 4.208.571.287 dengan populasi penduduk mencapai 7.634.758.428², media sosial merupakan salah satu jembatan yang menyediakan data opini pelanggan yang besar dan beragam. Salah satu media sosial yang banyak digunakan pelanggan adalah Twitter.

Twitter adalah sebuah media sosial berbasis *microblogging* yang memungkinkan pengguna dapat mengirim *posting* (tulisan) untuk publik atau *tweet*, mengirimkan *posting* kepada orang tertentu (*mention*), dan membaca sebuah pesan teks serta gambar dalam 280. Saat ini, jumlah *posting* yang dikirim oleh pengguna adalah mencapai 500 juta *posting* per hari atau sebanding dengan sekitar 6000 *posting* setiap detik³. Fasilitas perpesanan yang dapat langsung menuju ke perusahaan terkait (biasanya menggunakan nama produknya), menjadikan Twitter disenangi pelanggan dalam menyampaikan opini. Banyaknya *tweet* berupa opini yang merujuk pada perusahaan, membutuhkan suatu analisis dan metode tertentu terkait topik apa yang mayoritas dikeluhkan pelanggan, bagaimana pandangan pelanggan pada hal tersebut, serta bagaimana peluang jika bagian tersebut dikembangkan. Kesulitan dalam memahami apa yang pelanggan sampaikan melalui media sosial karena jumlahnya yang cukup besar, menjadikan penggalan informasi (*information retrieval*) perlu

¹ Smith, Aaron, dan Monica Anderson. 2018. *Social Media Use in 2018*. Pew Research Center : Internet & Technology. [Online]. Tersedia: www.pewinternet.org/2018/03/01/social-media-use-in-2018 . (diakses 16 Januari 2019).

² Internet World Stats. 2018. *World Internet Users and 2018 Population Stats*. Miniwatts Marketing Group. [Online]. Tersedia: www.internetworldstats.com/stats.htm . (diakses 16 Januari 2019).

³ Smith, Kit. 2018. *121 Amazing Social Media Statistics and Facts*. Brandwatch. [Online]. Tersedia: www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/amp . (diakses 16 Januari 2019).

dilakukan dan dikembangkan. Ditambah lagi dengan tipe data teks opini media sosial yang tidak terstruktur (*unstructured text*) menyebabkan diperlukan metode *text mining* dan *information retrieval theory* dalam mengatasinya.

Dalam kasus ini, pendekatan dengan *social media mining* untuk mendapatkan analisis peluang suatu produk dilakukan dengan memanfaatkan topik pembicaraan pelanggan dan analisis sentimen pelanggan melalui opini. Pada pendekatan ini, topik produk diidentifikasi melalui pemodelan topik dan klusterisasi dari opini pelanggan untuk selanjutnya menjadi derajat kepentingan (*importance degree*). Analisis sentimen pelanggan melalui nilai kuantitatif polaritas opini dilakukan untuk mendapatkan derajat kepuasan pelanggan (*satisfaction degree*). Akhirnya, peluang suatu produk dapat dihitung dengan menggunakan metode yang diusulkan Ulwick dalam *Outcome Driven Innovation (ODI)* yakni metode konsep penggalan peluang produk (*Product Opportunity Mining concept*) [4]. Metode tersebut akan memetakan topik-topik yang sedang dibicarakan pelanggan ke dalam beberapa kategori, yakni *over-served*, *served-right*, dan *under-served* serta peluangnya [4]. Oleh karena itu, pengembangan dan inovasi produk dapat dilakukan pada bagian-bagian yang membutuhkan tinjauan khusus.

Kelebihan dari metode tersebut adalah dapat digunakan tidak hanya dalam identifikasi peluang suatu produk berupa barang, namun juga dapat digunakan dalam bidang jasa. Selain itu, metode tersebut dapat mengevaluasi peluang yang potensial untuk pengembangan produk melalui sosial media, serta dapat digunakan untuk memantau secara *real-time*. Pemanfaatan survei secara statistik juga dapat diterapkan untuk mendapatkan data yang mendukung, di samping menggunakan opini pelanggan.

Penelitian dengan *social media mining* untuk perencanaan suatu produk, pernah dilakukan oleh Byeongki Jeong, Janghyeok

Yook, dan Jae-Min Lee pada tahun 2017 dengan menggunakan pendekatan *product opportunity mining* berbasis pemodelan topik LDA dan analisis sentimen [5]. Jeong melakukan analisis *social media mining* pada Samsung Galaxy Note 5. Data yang digunakan pada penelitian tersebut adalah data ulasan pelanggan yang dihimpun dari situs Reddit. Kesimpulan dari penelitian Jeong antara lain adalah topik “Samsung Pay” adalah topik yang sedang ramai dibicarakan pelanggan melalui opini dan memiliki peluang yang tinggi untuk dikembangkan. Ini sebagai akibat dari fasilitas *Near Field Communication (NFC)* yang saat itu sedang marak dan perlu mendapat perhatian khusus [5].

Pada penelitian ini, dilakukan suatu analisis produk McDonald’s dengan menggunakan pendekatan *social media mining* yang memanfaatkan opini pelanggan yang disampaikan melalui media sosial Twitter atau *tweet*. Pemodelan suatu topik dengan *Latent Semantic Analysis* dan K-Means, serta perhitungan polaritas opini dengan analisis sentimen VADER, kemudian dikalkulasi dan didapatkan keputusan dari suatu peluang produk (topik produk) untuk dikembangkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut di atas, adapun rumusan masalah yang dapat diambil pada penelitian Tugas Akhir ini adalah sebagai berikut.

1. Bagaimana mengimplementasikan *Latent Semantic Analysis* dan K-Means dalam pemodelan topik pada kumpulan *tweet*?
2. Bagaimana melakukan analisis sentimen dengan VADER pada kumpulan *tweet*?
3. Bagaimana mengimplementasikan *social media mining* dalam pengambilan keputusan untuk pengembangan produk dengan pendekatan *product opportunity mining* pada Twitter?

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut.

1. Data yang digunakan adalah data *tweet* dengan kata kunci @McDonalds berjumlah 2000 *tweet* dari tanggal 4 Maret 2019 sampai dengan tanggal 7 Maret 2019.
2. Data *tweet* yang digunakan merupakan teks dalam bahasa Inggris dan lokasi *posting* diabaikan.
3. Penulisan kalimat dengan gaya bahasa satire (gaya bahasa sindiran) diabaikan.
4. Batas ambang minimum *sum explained variance ratio* yang digunakan adalah 0,30.
5. Daftar *lexicon* polaritas opini yang digunakan adalah daftar *lexicon* C. J. Hutto dan Eric Gilbert.
6. Bahasa pemrograman yang digunakan adalah Python.

1.4 Tujuan

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut.

1. Mendapatkan hasil pemodelan topik dari kumpulan *tweet* melalui *Latent Semantic Analysis* dan K-Means.
2. Mendapatkan hasil analisis sentimen dengan VADER pada kumpulan *tweet*.
3. Mendapatkan keputusan terkait bagian dari produk yang akan dikembangkan dengan pendekatan *product opportunity mining*.

1.5 Manfaat

Manfaat yang dapat diambil dari penelitian Tugas Akhir ini adalah mendapatkan gambaran tentang implementasi *social media mining* berdasarkan pemodelan topik dan analisis sentimen pada opini pelanggan yang disampaikan melalui media sosial Twitter dalam pengambilan keputusan terkait perencanaan suatu produk.

1.6 Sistematika Penulisan

Adapun sistematika penulisan pada laporan Tugas Akhir ini adalah sebagai berikut:

BAB I. PENDAHULUAN

Pada bab ini dijelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, serta sistematika penulisan yang berkaitan dengan Tugas Akhir ini.

BAB II. TINJAUAN PUSTAKA

Pada bab ini dijelaskan tentang penelitian terdahulu dan juga referensi-referensi materi yang digunakan dalam penelitian Tugas Akhir ini.

BAB III. METODOLOGI PENELITIAN

Pada bab ini dijelaskan metodologi, tahapan-tahapan, blok diagram, dan juga detail pekerjaan pada penelitian Tugas Akhir ini.

BAB IV. PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini dijelaskan perancangan dan implementasi sistem yang dikerjakan pada penelitian Tugas Akhir ini.

BAB V. UJI COBA DAN PEMBAHASAN

Pada bab ini diuraikan uji coba sistem pada data uji coba. Hasil uji coba dibahas dan interpretasikan pada bab ini sehingga akan ditemukan kesimpulan yang sesuai dengan rumusan masalah.

BAB VI. KESIMPULAN DAN SARAN

Pada bab ini dipaparkan secara singkat kesimpulan dari penelitian Tugas Akhir ini serta saran-saran dapat digunakan untuk pengembangan penelitian di masa yang akan datang.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

Pada bab ini dipaparkan penelitian terdahulu berkaitan dengan penelitian Tugas Akhir ini, referensi, serta berbagai literatur yang digunakan dalam penelitian Tugas Akhir ini.

2.1 Penelitian Terdahulu

Penelitian terkait identifikasi peluang suatu produk menggunakan *text mining* dan *association rule mining* pernah dilakukan oleh Wonchul Seo, dkk pada tahun 2016. Seo melakukan penelitian dengan mengaplikasikan *text mining* untuk mengekstraksi bobot kalimat dan teori *association rule mining* dalam membentuk suatu susunan kata yang sering disampaikan oleh pelanggan dengan *support minimum* tertentu [6]. Studi kasus yang digunakan oleh Seo adalah pada produk dari Canon Inc. Susunan kata yang didapatkan dikelompokkan dengan susunan kata lain hingga mendapatkan suatu topik tertentu. Perhitungan derajat *confidence* dilakukan pada topik yang sudah terbentuk untuk mendapatkan topik yang merepresentasikan bidang yang potensial. Kesimpulan dari penelitian Seo adalah bahwa dua derajat *confidence* tertinggi dimiliki oleh topik *information processing apparatus* dan *external apparatus*. Hal ini mengindikasikan bahwa dua hal tersebut adalah bagian dari produk yang berpotensi untuk dikembangkan dan diinovasikan oleh perusahaan Canon Inc.

Implementasi *Latent Semantic Analysis* (LSA) untuk mencari dokumen yang mempunyai kemiripan sesuai kata kunci pernah dilakukan oleh Gang Kou dari Southwestern University of Finance and Economic, China pada tahun 2015 [7]. Kou melakukan penelitian terhadap dokumen yang memuat penelitian terkait MCDM (*Multi Criteria Decision Making*) dari tahun 1985

hingga 1990 dengan melakukan generalisasi seluruh dokumen ke dalam 6,11, dan 13 topik. Masing-masing topik dipilih 30 kata dengan bobot tertinggi. Kesimpulan yang diperoleh adalah bahwa LSA mampu memberikan hasil yang memuaskan dalam melakukan pengelompokan terhadap dokumen-dokumen yang mirip atau memiliki kata kunci yang hampir sama.

Byeongki Jeong, dkk pada tahun 2017 melakukan penelitian tentang perencanaan produk dengan *social media mining* berbasis pemodelan topik dan analisis sentimen pada produk barang, yakni Samsung Galaxy Note 5 [5]. Jeong mengimplementasikan teori peluang produk dalam ODI yakni *opportunity-importance-satisfaction* dalam menganalisis bagian produk yang memiliki peluang tinggi untuk dikembangkan. Data opini bersumber dari situs Reddit dikumpulkan dan selanjutnya diproses dengan *Latent Dirichlet Allocation* (LDA) hingga mendapatkan topik yang sedang marak dibicarakan pelanggan. Dari topik yang terbentuk, pengukuran secara kuantitatif diterapkan dengan teori distribusi peluang topik-dokumen dan menghasilkan derajat *importance*. Analisis sentimen dengan AlchemyAPI diimplementasikan untuk menghasilkan derajat *satisfaction*. Dari dua hal tersebut kemudian dikalkulasi derajat *opportunity* bagian produk dengan formula ODI. Kesimpulan dari penelitian Jeong adalah bahwa *Samsung Pay*, *Fast Charge*, dan *Detect Pen* adalah tiga bagian produk dari Samsung Galaxy Note 5 yang memiliki peluang terbesar untuk dikembangkan.

Penelitian tentang analisis sentimen berbahasa Inggris untuk menganalisis data *tweet* pernah dilakukan oleh Vipul Kumar, dkk dari Maharaja Agrasen Institute of Technology, India pada tahun 2018 [8]. Pada penelitiannya, Kumar melakukan analisis Twitter dengan menggunakan algoritma analisis sentimen VADER. Beberapa kategori *tweet* yang digunakan dalam pengujian adalah *tweet* seputar ekonomi, kesehatan, dan politik. Kesimpulan dari

penelitian tersebut adalah bahwa analisis sentimen Twitter dengan VADER dapat memberikan akurasi berkisar 85-90% dan cocok digunakan analisis suatu sentimen suatu kalimat.

Adapun perbandingan pada masing-masing penelitian terdahulu secara ringkas dapat diperhatikan pada Lampiran A.

2.2 Media Sosial

Media sosial merupakan sebuah variasi dari aplikasi internet yang memungkinkan pengguna membuat sebuah konten dan berinteraksi dengan pengguna lain [9]. Jenis-jenis media sosial yang berkembang saat ini sangat beragam, seperti Facebook, Twitter, blogger, wordpress, dan sebagainya. Melihat dari perkembangan pada tahun-tahun sebelumnya, media sosial berkembang dari model surat elektronik (*E-mail*), *mailing list*, IRC, hingga bertransformasi menjadi *microblogging* seperti sekarang ini. *Microblogging* adalah fasilitas berkirim pesan pendek yang dibatasi oleh jumlah karakter melalui sebuah aplikasi. Beberapa jenis *microblogging* yang sampai saat ini masih diterima oleh masyarakat secara baik adalah Facebook dan Twitter. Facebook dan Twitter keduanya memiliki karakteristik yang berbeda. Facebook lebih tertutup daripada Twitter, sehingga pengguna yang tidak memiliki hubungan pertemanan tidak dapat melihat isi *posting* satu dengan yang lain. Berbeda dengan Facebook, Twitter lebih terbuka sehingga pengguna yang tidak saling kenal dapat melihat *posting* dan profil pengguna lain. Keterbukaan Twitter yang demikian, menjadikan salah seorang pengguna merasa lebih populer dibanding yang lain yakni dengan adanya fitur *following* (orang yang diikuti) dan *follower* (pengikut).

Orang-orang yang berpengaruh atau merupakan suatu tokoh masyarakat, *brand* ternama, ataupun suatu akun populer, cenderung memiliki *follower* lebih banyak daripada *following*

(orang yang diikuti). Istilah yang populer dalam Twitter, antara lain adalah *tweet*, *mention*, *retweet*, dan juga tagar. *Tweet* adalah istilah yang menunjukkan sebuah *posting* dalam twitter. *Mention* adalah istilah untuk menunjukkan bahwa terdapat suatu pesan secara terbuka yang ditujukan kepada orang tertentu. Bila menyetujui, suka, atau ingin mendukung sebuah *tweet* biasanya digunakan *retweet*. Berikutnya adalah penggunaan kata tertentu yang diawali dengan tanda “#” atau tagar biasanya digunakan untuk menandai suatu topik obrolan atau topik yang sedang populer pada waktu tersebut. Twitter yang didirikan di San Fransisco ada tahun 2001 ini, memiliki batasan pada setiap *tweet*, *mention*, dan pesan lain yakni sejumlah 280 karakter (dapat diperpanjang dengan aplikasi tertentu). Inilah yang menjadikan pesan-pesan Twitter tampak singkat dan padat.

Keunggulan penggunaan Twitter dibandingkan dengan media sosial berbasis *microblogging* lain adalah Twitter memiliki jumlah karakter sedikit dalam *posting*-nya sehingga pesan yang dikirimkan padat, akun asli dan palsu dari suatu tokoh masyarakat dapat diidentifikasi melalui fasilitas *verified account* serta tidak perlu memiliki hubungan pertemanan khusus untuk dapat mengirim pesan atau melihat aktivitas akun Twitter orang lain.

Seiring berkembangnya zaman, ilmu yang mempelajari media sosial serta elemen-elemen dalam media sosial semakin berkembang, termasuk proses-proses dalam mengenali informasi yang dikirimkan. Suatu proses untuk merepresentasikan, menganalisis, dan mengekstraksi pola data bersumber dari media sosial disebut dengan *social media mining* [10]. Beberapa hal yang diimplementasikan dalam *social media mining* adalah teori graf, algoritma klasifikasi, algoritma *clustering*, serta analisis komunitas atau *social network analysis*.

2.3 Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency atau TF-IDF adalah salah satu metode pembobotan fitur (kata atau frasa) yang didasarkan pada frekuensi kemunculan fitur dan jumlah dokumen (*tweet*) yang memuat fitur tersebut [11]. Secara sederhana, TF-IDF melakukan perhitungan jumlah fitur tertentu pada sebuah *tweet* dengan memerhatikan *tweet* lain yang mengandung fitur tersebut. Bobot TF-IDF memberikan nilai yang tinggi untuk fitur yang sering muncul di suatu *tweet* tetapi jarang muncul di *tweet* lainnya, serta nilai yang rendah untuk fitur yang berfrekuensi tinggi pada suatu *tweet* tetapi juga sering muncul di *tweet* lain. Jumlah *tweet* yang digunakan dalam penelitian Tugas Akhir ini adalah sebanyak 2000 *tweet*. Misalkan m adalah banyaknya *tweet* dan n adalah banyaknya fitur hasil ekstraksi. Rumus dalam mencari bobot TF adalah sebagai berikut.

$$tf_{i,j} = t_{i,j} \quad (2.1)$$

dengan:

- $tf_{i,j}$: *term frequency* fitur ke- j dalam *tweet* ke- i dengan $i = 1,2, \dots, m$ dan $j = 1,2, \dots, n$
- $t_{i,j}$: jumlah fitur ke- j dalam *tweet* ke- i dengan $i = 1,2, \dots, m$ dan $j = 1,2, \dots, n$

Sedangkan rumus IDF adalah sebagai berikut.

$$idf_j = \log\left(\frac{m}{doc_j}\right) + 1 \quad (2.2)$$

dengan:

- idf_j : *inverse document frequency* fitur ke- j
 m : jumlah *tweet* yang digunakan
 doc_j : jumlah *tweet* yang mengandung fitur ke- j

Dari rumus (2.1) dan rumus (2.2) didapatkan nilai bobot TF-IDF untuk fitur ke- j pada *tweet* ke- i adalah sebagai berikut.

$$w_{i,j(pra)} = tf_{i,j} \times idf_j \quad (2.3)$$

Normalisasi bobot TF-IDF dari hasil pada rumus (2.3) ditunjukkan oleh rumus (2.4) berikut.

$$w_{i,j} = \frac{w_{i,j(pra)}}{\sqrt{\sum_{j=1}^n w_{i,j(pra)}^2}} \quad (2.4)$$

dengan $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

Dari perhitungan dengan rumus (2.4) maka akan diperoleh tabel *tweet*-fitur yang berisi bobot-bobot fitur ternormalisasi seperti Tabel 2.1 berikut.

Tabel 2.1 Tabel Bobot *Tweet*-Fitur

	Fitur ke-1	Fitur ke-2	...	Fitur ke- j
<i>Tweet</i> ke-1	$w_{1,1}$	$w_{1,2}$...	$w_{1,j}$
<i>Tweet</i> ke-2	$w_{2,1}$	$w_{2,2}$...	$w_{2,j}$
...
<i>Tweet</i> ke- i	$w_{i,1}$	$w_{i,2}$...	$w_{i,j}$

Elemen-elemen *tweet*-fitur pada Tabel 2.1 di atas, dapat direpresentasikan dalam bentuk *sparse matrix* yang digunakan sebagai masukan pada proses selanjutnya. Asumsikan A adalah *sparse matrix* yang terbentuk sebagai berikut.

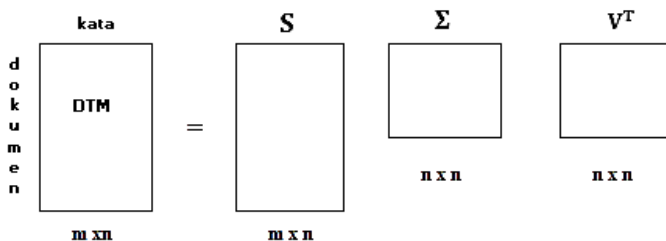
$$A = \begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} & w_{1,4} & \dots & w_{1,j} \\ w_{2,1} & w_{2,2} & w_{2,3} & w_{2,4} & \dots & w_{2,j} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ w_{i,1} & w_{i,2} & w_{i,3} & w_{i,4} & \dots & w_{i,j} \end{bmatrix} \quad (2.5)$$

2.4 Pemodelan Topik

Pemodelan topik adalah salah satu metode yang digunakan untuk mengenali topik-topik dalam sekumpulan data teks. Salah satu metode pemodelan topik adalah *Latent Semantic Analysis*. *Latent Semantic Analysis* (LSA) atau dalam *information retrieval* disebut dengan *Latent Semantic Indexing*. LSA adalah suatu metode pemodelan topik dengan mereduksi fitur kurang penting atau tidak membangun *sparse matrix* untuk mendapatkan *principal components* menggunakan teori *Singular Value Decomposition* (SVD) [12]. Implementasi LSA adalah dengan mendekomposisi matriks A_{mn} seperti pada matriks (2.5) menjadi matriks $S\Sigma V^T$ dengan mereduksi menjadi k komponen dengan $1 \leq k < \min(m, n)$ dengan menggunakan *truncated SVD*.

Misalkan suatu *sparse matrix* dideskripsikan A_{mn} dengan m adalah jumlah *tweet* dan n adalah jumlah fitur hasil ekstraksi. Dari A akan didapatkan 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 $r \times r$ dimana $r = \min(m, n)$. Nilai eigen (λ) dari matriks B berjumlah r dan bernilai positif diurutkan dari yang terbesar hingga terkecil sehingga $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_a \geq \lambda_{a+1} = \dots = \lambda_r = 0$. Σ merupakan matriks diagonal dengan elemen nilai-nilai singular (δ) yang merupakan akar dari

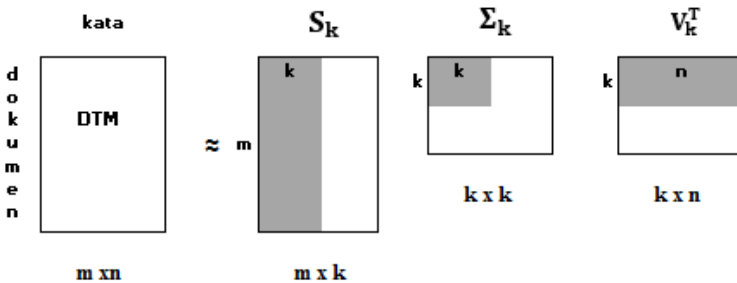
nilai eigen yang bersesuaian atau $\delta_i = \sqrt{\lambda_i}$. S merupakan matriks dengan elemen vektor-vektor singular kiri dan V merupakan matriks dengan elemen vektor-vektor singular kanan. Vektor singular kanan, v_i , didapat dari vektor eigen orthonormal bersesuaian dengan δ_i dari matriks N . Sedangkan vektor singular kiri, s_i , didapat dari $s_i = \frac{1}{\delta_i} A v_i$. Adapun ilustrasi dari dekomposisi A menjadi matriks $S \Sigma V^T$ ditunjukkan dalam Gambar 2.1 berikut.



Gambar 2.1 Ilustrasi *Singular Value Decomposition*

Dengan reduksi menjadi k komponen dengan $1 \leq k < r$, dapat disusun matriks $S_k \times \Sigma_k$ berukuran $m \times k$ yang dikenal dengan matriks *tweet-principal components*. V_k adalah matriks V^T yang sudah direduksi dengan mempertahankan k baris pertama dan keseluruhan kolom dari matriks V^T , S_k adalah matriks S yang sudah direduksi dengan mempertahankan keseluruhan baris dan k kolom pertama dari matriks S , dan Σ_k adalah matriks Σ yang sudah direduksi dengan mempertahankan k baris pertama dan k kolom pertama dari matriks Σ .

Adapun ilustrasi dari reduksi tersebut ditunjukkan oleh Gambar 2.2.



Gambar 2.2 Ilustrasi *Truncated SVD*

Tampak pada Gambar 2.2 bahwa matriks S_k berukuran $m \times k$ dan Σ_k dengan ukuran $k \times k$. Hasil reduksi dengan *truncated SVD* menjadi *principal components* ($S_k \Sigma_k$) akan menghasilkan matriks yang direpresentasikan dalam Tabel 2.2 berikut.

Tabel 2.2 Tabel *Tweet-Principal Components*

	PC1	PC2	PC K
<i>Tweet ke-1</i>	$b_{1,1}$	$b_{1,2}$...	$b_{1,k}$
<i>Tweet ke-2</i>	$b_{2,1}$	$b_{2,2}$...	$b_{2,k}$
...
<i>Tweet ke-i</i>	$b_{i,1}$	$b_{i,2}$...	$b_{i,k}$

Tabel 2.2 menunjukkan representasi bobot *tweet* dalam sebuah matriks representasi *tweet-principal components*. Penentuan jumlah k pada interval tertentu dalam suatu kumpulan *tweet* yang dimodelkan dengan LSA ditentukan dengan mempertimbangkan nilai *sum explained variance ratio* atau jumlah varians penginterpretasi (jumlah rasio varians *truncated SVD* terhadap *sparse matrix* awal) dengan nilai 0 hingga 1 (0% hingga 100%). Nilai k dapat dipilih dengan melihat nilai *sum explained variance ratio* yang berada di atas batas ambang

tertentu. Varians tiap kolom matriks *truncated SVD* dan jumlah varians tiap baris *sparse matrix* dirumuskan sebagai berikut.

$$\sigma_{trunc_i}^2 = \sigma_i^2(S_k \Sigma_k) \quad (2.6)$$

$$\sigma_{sum}^2 = \sum_{j=1}^m \sigma_j^2(A) \quad (2.7)$$

dengan:

- $\sigma_{trunc_i}^2$: varians *truncated SVD* kolom ke i dengan $i = 1, 2, \dots, k$
- σ_{sum}^2 : jumlah varians dari tiap baris *sparse matrix* dengan $j = 1, 2, \dots, m$

Sehingga dengan rumus (2.6) dan (2.7), didapat *sum explained variance ratio* untuk k *principal components* sebagai berikut.

$$\sigma_{ratio}^2 = \sum_{i=1}^k \frac{\sigma_{trunc_i}^2}{\sigma_{sum}^2} \quad (2.8)$$

Untuk melakukan pengelompokan atau klusterisasi pada *tweet*, atau untuk mendapatkan *tweet* yang memiliki kesamaan pola, maka akan digunakan algoritma *K-Means cluster*. *K-Means cluster* adalah metode klusterisasi yang didasarkan pada jarak antara elemen-elemen anggota *tweet* dengan *centroid* atau titik pusat kluster [13]. *K-Means* termasuk dalam metode kluster berbasis partisi dengan memanfaatkan jarak antara poin dengan titik pusat klasternya yang memungkinkan terjadi iterasi hingga

suatu kondisi tertentu. Masukan K-Means adalah matriks yang sudah di-*truncated* pada proses sebelumnya.

Adapun algoritma K-Means *cluster* yang digunakan adalah sebagai berikut.

- a. Misalkan terdapat k jumlah klaster. Pilih k obyek secara acak sebagai titik pusat klaster.
- b. Masukkan obyek-obyek yang sama atau mirip ke dalam kelompok sesuai dengan titik pusat klaster (*centroid*). Adapun perhitungan kemiripan suatu *tweet* adalah dengan menggunakan jarak *euclidean*.

$$d_{ij} = \sqrt{\sum_{a=1}^n (w_{a,i} - c_{a,j})^2} \quad (2.9)$$

dengan:

- d_{ij} : jarak *tweet* i dengan *centroid* j , dengan $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, k$
- $w_{a,i}$: bobot fitur ke a dari *tweet* i
- $c_{a,j}$: bobot fitur ke a dari *centroid* j

Klaster yang tepat setelah dihitung jarak *tweet* dengan *centroid* dengan rumus (2.9) adalah dengan memilih d terkecil. Jadi, untuk d_{ij} terkecil artinya *tweet* i akan dimasukkan dalam klaster dengan *centroid* j .

- c. Hitung nilai rata-rata atribut dari masing-masing klaster untuk didapatkan *centroid* yang baru. Rata-rata atribut dari suatu klaster didapat dengan menghitung rata-rata bobot untuk setiap fitur dalam klaster yang sama.
- d. Bila nilai rata-rata pada perhitungan ke $n - 1$ berbeda dengan yang ke n , perbarui rata-rata klaster dengan rata-rata terbaru dan kembali ke tahap b, bila nilai rata-rata sama maka

proses selesai. Dalam hal ini dapat dibatasi dengan maksimal iterasi.

Sebagai bahan pertimbangan dalam menentukan jumlah kluster, dihitung nilai *inertia* pada interval pengujian jumlah kluster dengan menggunakan metode *elbow*. Rumus metode *elbow* ditunjukkan pada rumus (2.10) berikut.

$$I = \sum_{i=1}^m d_i^2 \quad (2.10)$$

dengan:

- I : nilai *inertia*
 d_i : jarak *tweet* i dengan *centroid*-nya, dengan m adalah banyak *tweet*

Nilai *inertia* terbaik didapatkan apabila pada interval jumlah kluster tertentu didapatkan nilai I terkecil, atau apabila konvergen ke suatu titik tertentu maka diambil nilai yang diperkirakan mulai memiliki selisih terkecil dengan nilai setelahnya (mulai konstan) dan selisih terjauh dengan setelah nilai sebelumnya. Terlepas dari jumlah kluster ditentukan secara subyektif.

Dalam *text mining*, klasterisasi digunakan untuk mengelompokkan dokumen-dokumen yang memiliki kesamaan dalam topik, fitur, ataupun suatu bahasan tertentu. Untuk mendapatkan fitur yang menjadi kata kunci dari tiap kluster, diambil fitur yang memiliki bobot terbesar sejumlah n fitur. Kumpulan fitur-fitur tersebut akan menjadi sebuah *keyword* atau kata kunci dari kluster tersebut. Untuk memberikan label atas kluster yang terbentuk, identifikasi secara manual perlu dilakukan dengan mempertimbangkan konteks isi beberapa *tweet* yang menjadi anggota kluster tersebut.

2.5 Analisis Sentimen

Analisis sentimen adalah penelitian multidisipliner yang menganalisis tentang opini masyarakat berupa sentimen, polaritas, dan juga kecenderungan emosi dari opini masyarakat. Analisis sentimen banyak digunakan oleh peneliti, data analis, dan juga kalangan akademisi dalam menganalisis kecenderungan tanggapan masyarakat terhadap suatu hal. Baik berupa produk, kampanye, topik khusus, dan lainnya. Dalam penelitian Tugas Akhir ini, fokus utama analisis sentimen adalah pada perhitungan nilai polaritas *tweet*. Nilai polaritas *tweet* adalah sebuah nilai yang menunjukkan derajat sentimen suatu *tweet*. Misalnya, nilai polaritas dari kata “*hate*” adalah -1 dan “*great*” adalah 1. Adapun pengukuran polaritas dalam penelitian ini adalah dengan menggunakan aturan *Valence Aware Dictionary for Sentiment Reasoning* (VADER) berbasis *a parsimonius rule-based model* yang diusulkan oleh C.J. Hutto dan Eric Gilbert dari Georgia Institute of Technology, Atlanta [14].

Hutto dan Eric melakukan survei pada sepuluh ahli untuk memberikan penilaian pada kata perasa (*sense word*) dan simbol emosi (emoji) dalam bahasa Inggris. Polaritas suatu kata merupakan rata-rata dari pendapat para ahli. Polaritas kata yang bernilai positif merupakan kata yang terindikasi memiliki maksud positif, dan sebaliknya. Adapun kutipan daftar *lexicon* dan polaritas yang disusun Hutto adalah seperti Tabel 2.3.

Selain polaritas kata yang terdefinisi seperti pada Tabel 2.3 di atas, Hutto dan Eric juga memberikan pembobotan kata pada beberapa kata *booster* seperti *extremely*, *very*, *so*, kata bernilai negasi seperti *don't*, *doesn't*, *not* serta pembobotan untuk penekanan pada suatu kata. Penekanan pada suatu kata dibatasi oleh suatu kata perasa yang ditulis kapital sedangkan tidak untuk kata lain (*The chicken is very GOOD*) dan adanya tanda perintah

(!) serta tanda tanya (?) yang ditulis berulang (*The chicken is very good!!!, You know it is very nice ??*).

Tabel 2.3 Kutipan Daftar *Lexicon* Hutto dan Eric

Kata	Polaritas Kata	Standar Deviasi	Hasil Penilaian Ahli
<i>good</i>	1,9	0,9434	[2, 1, 1, 3, 2, 4, 2, 2, 1, 1]
<i>dirty</i>	-1,9	0,83066	[-2, -1, -1, -1, -2, -2, -1, -3, -3, -3]
<i>best</i>	3,2	0,6	[2, 4, 4, 3, 4, 3, 3, 3, 3, 3]

Dalam perhitungan polaritas suatu *tweet*, *tweet* berbentuk kalimat akan dilakukan tokenisasi (penguraian ke dalam bentuk kata) kemudian diberikan penilaian berdasarkan daftar *lexicon*. Misalkan, diberikan *tweet* yang akan dianalisis t , kemudian n yang merupakan banyak kata dari *tweet* t , dan w_i yang merupakan kata ke i dengan $i = 1, 2, \dots, n$. Nilai polaritas dihitung dengan rumus (2.11) sebagai berikut.

$$P_{total}(t) = \sum_{i=1}^n P_{kata}(i) \quad (2.11)$$

dengan:

- P_{total} : polaritas *tweet* awal t
 P_{kata} : polaritas *tweet* ke- i

Hutto dan Eric memberikan beberapa kondisi khusus dalam perhitungan polaritas *tweet* sebagai berikut.

- a. Bila kata i yang ada dalam *lexicon* ditulis dengan huruf kapital sedangkan tidak untuk kata lain, maka kata tersebut diberikan penekanan sehingga $P_{total}(t)$ ditambah dengan 0,733 bila kata tersebut kata positif dan dikurangi dengan 0,733 bila kata merupakan kata negatif.
- b. Bila kata i didahului oleh kata *booster* maka kata ditambah dengan polaritas kata *booster* (jika kata merupakan kata negatif maka polaritas *booster* dikalikan dengan -1).
- c. Bila terdapat kata “*but*”, maka polaritas bagian kalimat sebelum kata “*but*” akan diturunkan intensitasnya dengan dikalikan 0,5 sedangkan polaritas bagian kalimat setelah kata “*but*” dinaikkan intensitasnya dengan dikalikan 1,5.
- d. Bila sebelum kata i terdapat kata “*least*” maka polaritas kata i dikalikan -0,74.
- e. Bila kalimat mengandung tanda perintah (!):
 - i. Bila jumlah tanda perintah (!) kurang dari empat:
 - I. Bila $P_{total}(t)$ bernilai positif, maka ditambah dengan $P_{tanda1} = jumlah\ tanda \times 0,292$.
 - II. Bila $P_{total}(t)$ bernilai negatif, maka ditambah dengan $P_{tanda1} = -(jumlah\ tanda \times 0,292)$.
 - ii. Bila jumlah tanda perintah (!) lebih dari sama dengan empat:
 - I. Bila $P_{total}(t)$ bernilai positif, maka ditambah dengan $P_{tanda1} = 4 \times 0,292$.
 - II. Bila $P_{total}(t)$ bernilai negatif, maka ditambah dengan $P_{tanda1} = -(4 \times 0,292)$.
- f. Bila kalimat mengandung tanda tanya (?):
 - i. Bila jumlah tanda tanya (?) lebih dari satu dan kurang dari sama dengan tiga:
 - I. Bila $P_{total}(t)$ bernilai positif, maka ditambah dengan $P_{tanda2} = jumlah\ tanda \times 0,18$.
 - II. Bila $P_{total}(t)$ bernilai negatif, maka ditambah dengan $P_{tanda2} = -(jumlah\ tanda \times 0,18)$.

ii. Bila jumlah tanda perintah (?) adalah satu atau lebih dari tiga:

- I. Bila $P_{total}(t)$ bernilai positif, maka ditambah dengan $P_{tanda2} = 0,96$.
- II. Bila $P_{total}(t)$ bernilai negatif, maka ditambah dengan $P_{tanda2} = -0,96$.

Bila tidak mengandung tanda perintah (!) dan tidak mengandung tanda tanya (?), atau $P_{total}(t) = 0$ maka $P_{tanda1} = P_{tanda2} = 0$.

Sehingga polaritas akhir *tweet* dapat dihitung dengan rumus (2.12) berikut.

$$P_{akhir}(t) = P_{total}(t) + P_{tanda1} + P_{tanda2} \quad (2.12)$$

Agar nilai polaritas *tweet* tidak memiliki ragam yang tinggi, maka nilai polaritas dinormalisasi pada interval $[-1,1]$. Rumus yang digunakan untuk menormalisasi nilai polaritas sebelumnya ditunjukkan pada rumus (2.13).

$$P_{norm}(t) = \frac{P_{akhir}(t)}{\sqrt{(P_{akhir}(t))^2 + \alpha}} \quad (2.13)$$

dengan:

- $P_{norm}(t)$: polaritas *tweet* akhir normalisasi
 α : parameter normalisasi dengan $\alpha > 1$

Pengukuran kualitas sistem analisis sentimen mampu melakukan klasifikasi secara benar pada kelas positif (polaritas bernilai positif), negatif (polaritas bernilai negatif), ataupun netral (polaritas sama dengan nol) adalah dengan menganalisis matriks

pengukuran (*classifier evaluation matrix*) yang disebut dengan *confusion matrix*. Adapun pengukuran yang digunakan adalah *accuracy*, *precision*, dan *recall*. *Accuracy* adalah persentase opini yang diklasifikasikan benar ke dalam kelas sesuai dengan kelas target sebenarnya. *Precision* adalah persentase opini positif yang diklasifikasikan oleh sistem secara benar ke kelas target positif sebenarnya (*exactness*). Sedangkan *recall* adalah persentase opini positif yang diklasifikasikan oleh sistem secara benar ke kelas target positif sistem (*completeness*). Rumus yang digunakan untuk mengukur *accuracy*, *precision*, dan *recall* adalah sebagai berikut.

$$accuracy = \frac{TP + TN}{P + N} \times 100\% \quad (2.14)$$

$$recall = \frac{TP}{TP + FN} \times 100\% \quad (2.15)$$

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (2.16)$$

dengan:

- TP : jumlah *tweet* yang diprediksi positif dan kelas target sebenarnya adalah positif
- TN : jumlah *tweet* yang diprediksi non-positif dan kelas target sebenarnya adalah non-positif
- FP : jumlah *tweet* yang diprediksi positif sedangkan kelas target sebenarnya adalah non-positif
- FN : jumlah *tweet* yang diprediksi non-positif sedangkan kelas target sebenarnya adalah positif

- P : jumlah *tweet* yang sebenarnya bernilai positif
 N : jumlah *tweet* yang sebenarnya bernilai non-positif

2.5 Opportunity Mining

Penggalian peluang produk (*product opportunity mining*) merupakan suatu metode yang digunakan dalam menentukan bagian produk yang potensial atau memiliki peluang untuk dikembangkan oleh suatu perusahaan [4]. Peluang produk yang dimaksud juga disebutkan oleh Anthony Ulwick dalam *Outcome Driven Innovation* sebagai peluang yang didasarkan pada derajat *importance* (kepentingan) dan derajat *satisfaction* (kepuasan) [4]. Secara filosofi, Ulwick mengasumsikan bahwa peluang suatu bagian produk diperlukan ketika kebutuhan pelanggan adalah penting tetapi kepuasan tidak dicapai secara maksimal. Berdasarkan perspektif ini, kebutuhan yang lebih penting tapi kepuasan yang rendah akan mendapat prioritas tertinggi. Hal ini kemudian akan dikombinasikan dengan peluang produk yang akan dikembangkan tersebut. Adapun rumus untuk menghitung peluang bagian produk tersebut adalah sebagai berikut.

$$\begin{aligned}
 \text{opportunity} = & \text{importance} + \max(\text{importance} \\
 & - \text{satisfaction}, 0)
 \end{aligned}
 \quad (2.17)$$

Derajat *importance* dan derajat *satisfaction* dicari dengan menggunakan hasil pemodelan topik dan analisis sentimen. Derajat *importance* awal (CS) didapat dengan menghitung banyaknya *tweet* yang dikelompokkan dalam suatu kluster tertentu.

Normalisasi dalam rentang $[0,10]$ dari CS akan menghasilkan derajat *importance* yang dirumuskan sebagai berikut.

$$importance_i = 10 \times \frac{CS_i - CS_{\min}}{CS_{\max} - CS_{\min}} \quad (2.18)$$

dengan:

- $importance_i$: derajat *importance* topik ke i
- CS_i : derajat *importance* awal topik ke- i
- CS_{\min} : derajat *importance* awal minimum
- CS_{\max} : derajat *importance* awal maksimum

Untuk setiap *tweet* yang muncul pada topik (klaster) ke i akan dihitung rata-rata polaritas dari semua opini yang termasuk dalam klaster tersebut sehingga akan didapat derajat *satisfaction* awal (SS). Normalisasi dalam rentang $[0,10]$ dari hasil sebelumnya akan menghasilkan derajat *satisfaction* yang dirumuskan sebagai berikut.

$$satisfaction_i = 10 \times \frac{SS_i - SS_{\min}}{SS_{\max} - SS_{\min}} \quad (2.19)$$

dengan:

- $satisfaction_i$: derajat *satisfaction* topik ke i
- SS_i : derajat *satisfaction* awal topik ke- i
- SS_{\min} : derajat *satisfaction* awal minimum
- SS_{\max} : derajat *satisfaction* awal maksimum

Sehingga dari rumus (2.17), (2.18), dan rumus (2.19) didapatkan rumus *opportunity* yang sudah ternormalisasi untuk setiap topik ke i sebagai berikut.

$$opportunity_i = importance_i + \max(importance_i - satisfaction_i, 0) \quad (2.20)$$

Dalam aplikasinya, untuk setiap pasangan nilai *importance* dan *satisfaction* ke *i*, dapat divisualisasikan dalam *product landscape map* seperti pada Gambar 2.3. Gambar 2.3 juga menunjukkan bagian-bagian produk yang digolongkan dalam *over-served*, *served-right*, ataupun *under-served* sesuai dengan garis batas yang sudah ditentukan. Titik-titik yang muncul pada diagram, merupakan topik-topik yang merepresentasikan bagian produk yang sedang ramai dibicarakan oleh pelanggan melalui *tweet*.



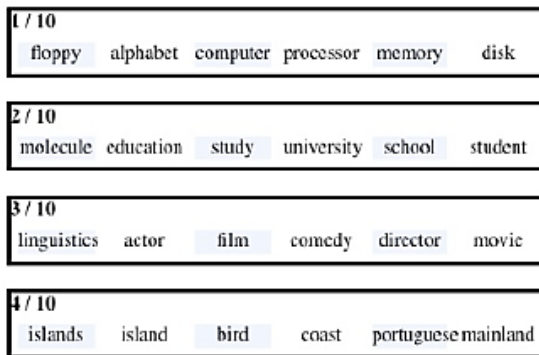
Gambar 2.3 Diagram *Product Landscape Map*

Garis yang membatasi bagian *served-right* dan *under-served* adalah garis lurus yang menghubungkan titik $(a, 0)$ dan $(10,10)$ dengan a adalah rata-rata dari derajat *importance*. Garis yang membatasi bagian *over-served* dan *served-right* adalah garis lurus

yang menghubungkan titik $(0, b)$ dan $(10, 10)$ dengan b adalah rata-rata dari derajat *satisfaction*.

2.6 Word Intrusion Task

Salah satu metode yang dapat digunakan untuk menguji suatu pemodelan topik adalah metode *word intrusion task* yang diusulkan oleh Chang, dkk [15]. Topik yang teridentifikasi dari kumpulan *tweet* dalam suatu kluster, merupakan penggabungan dari banyak *tweet* yang diambil fitur-fitur dengan bobot yang tertinggi. Ilustrasi dari keberadaan topik dalam kumpulan *tweet* diilustrasikan pada Gambar 2.4 di bawah ini. Berdasarkan topik-topik yang sudah dihasilkan pada pemodelan topik, *word intrusion task* bekerja dengan menggabungkan fitur-fitur dalam sebuah kluster sebagai satu topik dengan fitur dari kluster lain. Pengukuran dilakukan dengan cara meminta responden untuk menebak fitur mana yang bukan merupakan bagian dari kluster tersebut.



Gambar 2.4 Gambar *Word Intrusion Task*

Semakin banyak responden dengan persentase jawaban benar menandakan bahwa topik yang disusun dari kumpulan *tweet* dalam suatu kluster dapat diinterpretasikan dengan mudah oleh masyarakat.

Bila dalam sebuah rangkaian eksperimen terdapat lebih dari satu skenario, untuk mendapatkan skenario terbaik menggunakan *word intrusion task* adalah dengan melakukan perbandingan dua rata-rata tanpa data rujukan yang relevan (sampel acak dengan σ^2 tidak diketahui) [16]. Misalkan dua hasil eksperimen adalah X_i dan Y_j dengan $i = j$ dimana $i, j \in 1, 2, 3, \dots, n$, didapatkan rata-rata X_i dan Y_j secara berurutan adalah \bar{X} dan \bar{Y} dimana prediksi $\bar{X} - \bar{Y} > 0$. Pengukuran statistik dilakukan dengan pengujian hipotesis.

Hipotesis:

$$H_0 : \eta_X = \eta_Y$$

$$H_1 : \eta_X > \eta_Y$$

Statistik uji:

$$S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2 + \sum_{j=1}^n (Y_j - \bar{Y})^2}{2n - 2} \quad (2.21)$$

$$t^* = \frac{(\bar{X} - \bar{Y}) - (\eta_X - \eta_Y)}{S \sqrt{\frac{1}{n} + \frac{1}{n}}} \quad (2.22)$$

Kriteria pengujian:

Jika $p_{value} = P(t > t^*) < \alpha$, maka H_0 ditolak.

Untuk menguji pengaruh perlakuan pada skenario yang ada tanpa pengaruh pengelompokan pada satu perlakuan, dapat

menggunakan rancangan satu faktor dalam rancangan acak lengkap. Tabel pengamatan yang terbentuk setelah dilakukan eksperimen adalah sebagai berikut.

Tabel 2.4 Tabel Pengamatan RAL

Ulangan	Perlakuan				Total
	P ₁	P ₂	...	P _m	
U ₁	$e_{1,1}$	$e_{2,1}$		$e_{m,1}$	$e_{.1}$
U ₂	$e_{1,2}$	$e_{2,2}$		$e_{m,2}$	$e_{.2}$
...					
U _n	$e_{1,n}$	$e_{2,n}$		$e_{m,n}$	$e_{.n}$
Total	$e_{1.}$	$e_{2.}$		$e_{m.}$	$e_{..}$

Model linear yang digunakan dalam rancangan satu faktor dalam rancangan acak lengkap adalah sebagai berikut.

$$e_{i,j} = \mu + \tau_i + \varepsilon_{i,j} \quad (2.23)$$

dengan:

- $e_{i,j}$: respons atau hasil akibat perlakuan ke i dan ulangan ke j dimana $i = 1,2,3,4, \dots, m$ dan $j = 1,2,3,4, \dots, n$.
- μ : rata-rata umum
- τ_i : pengaruh perlakuan ke- i
- $\varepsilon_{i,j}$: pengaruh acak akibat perlakuan ke i dan ulangan ke- j

Dari model linear pada rumus (2.23), didapatkan tabel analisis varians atau tabel ANOVA pada Tabel 2.5.

Tabel 2.5 Tabel ANOVA Satu Faktor dalam RAL

Sumber variasi	Derajat kebebasan	Sum Squared	Mean Squared	F-measure
Perlakuan	$m - 1$	SS_p	$MS_p = \frac{SS_p}{m - 1}$	$\frac{MS_p}{MS_E}$
Galat	$m(n - 1)$	SS_E	$MS_E = \frac{SS_E}{m(n - 1)}$	
Total	$mn - 1$	SS_T	MS_T	

dengan:

$$FK = \frac{e_{..}^2}{mn} \quad (2.24)$$

$$SS_p = \frac{\sum_{i=1}^m e_i \cdot^2}{m} - FK \quad (2.25)$$

$$SS_T = \sum_{i=1}^m \sum_{j=1}^n e_{i,j}^2 - FK \quad (2.26)$$

$$SS_E = SS_T - SS_p \quad (2.27)$$

Asumsi yang digunakan dalam penelitian Tugas Akhir ini adalah asumsi model acak untuk menguji apakah perlakuan memberikan hasil yang berbeda pada skenario yang diusulkan atau tidak. Asumsi model acak memiliki batasan sebagai berikut.

$$E(\tau_i) = 0$$

$$Var(\tau_i) = \sigma_i^2$$

$$Var(\varepsilon_{i,j}) = \sigma^2, \forall i, j, \varepsilon_{i,j} IID \sim N(0, \sigma^2)$$

Langkah analisis statistik yang digunakan adalah memenuhi tiga hal berikut.

Hipotesis:

$$H_0 : \mu_1 = \mu_2 = \mu_3 = \dots = \mu_m = \mu$$

$$H_1 : \text{paling sedikit ada sepasang perlakuan } (i, i') \text{ dimana } \mu_i \neq \mu_{i'}$$

Statistik uji:

$$F - \text{measure} = \frac{MS_p}{MS_E} \quad (2.27)$$

$$F - \text{score} = f_{v_1, v_2, \alpha}$$

dimana:

$$v_1 = m - 1, v_2 = m(n - 1)$$

Kriteria pengujian:

Jika $F - \text{measure} > F - \text{score}$, maka H_0 ditolak sehingga paling sedikit ada sepasang perlakuan (i, i') dimana $\mu_i \neq \mu_{i'}$. Dengan kata lain perlakuan memberikan hasil yang berbeda secara signifikan.

BAB III METODOLOGI PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam pengerjaan Tugas Akhir ini. Selain itu juga akan dijelaskan detail pekerjaan serta diagram blok dalam penelitian Tugas Akhir ini.

3.1 Obyek dan Aspek Penelitian

Obyek yang digunakan dalam penelitian Tugas Akhir ini adalah data *tweet* dari Twitter dengan kata kunci @McDonalds. Di samping itu, aspek yang diteliti dalam penelitian Tugas Akhir ini adalah *social media mining* dengan pemodelan topik dan analisis sentimen untuk pengambilan keputusan dalam suatu perencanaan produk.

3.2 Peralatan

Adapun peralatan baik perangkat keras maupun perangkat lunak yang digunakan ditunjukkan oleh Tabel 3.1 berikut.

Tabel 3.1 Tabel Peralatan yang Digunakan

Perangkat keras	<ol style="list-style-type: none">1. Laptop HP 14", HDD 500GB2. Sistem Operasi Windows 10 Education 64 bit3. Prosesor AMD A-8 7410, 2,2 GHz4. RAM 8 GB
Perangkat lunak	<ol style="list-style-type: none">1. Anaconda Navigator dengan aplikasi Jupyter Notebook 5.0.0 dan Spyder dengan bahasa pemrograman Python2. Minitab3. Microsoft Office (Word, Excel)

3.3 Tahapan Penelitian

Adapun langkah-langkah yang digunakan pada penelitian Tugas Akhir ini adalah sebagai berikut:

1. Studi literatur

Pada tahap ini dilakukan studi literatur dan penggalian informasi dari referensi-referensi yang mendukung penelitian Tugas Akhir ini.

2. Pengumpulan Data Masukan

Data masukan yang digunakan adalah data yang berasal dari Twitter dengan kata kunci @McDonalds sebanyak 2000 *tweet*.

3. Perancangan sistem

Pada tahap ini dilakukan perancangan dan implementasi sistem yang akan digunakan dalam uji coba, diantaranya.

a. Perancangan data masukan.

b. Perancangan proses *text mining* dan penyusunan *sparse matrix*.

c. Perancangan proses pemodelan topik, analisis sentimen, dan perhitungan *opportunity*.

d. Perancangan desain antarmuka

4. Implementasi sistem

Pada tahap ini dilakukan implementasi sistem yang sudah dirancang pada tahap sebelumnya yang meliputi:

i. Implementasi sistem dalam mengimpor data dalam sistem serta memanipulasi bentuk dalam *list*.

ii. Praproses data *tweet* dengan teori-teori dalam *text mining* seperti transformasi, *case folding*, *filtering*, *stopwords-removal*, dan *tokenizing*. Proses yang dilakukan pada tahap ini adalah mengubah *tweet* menjadi huruf nonkapital, menghilangkan unsur *mention*, tagar, url, dan *stopwords* yang sering digunakan.

- iii. Penyusunan *sparse matrix* dengan TF-IDF.
Dengan mengimplementasikan rumus pada (2.1), (2.2), (2.3), dan (2.4) pada data *tweet* yang sudah dilakukan praproses pada langkah (ii), maka akan dibangun sebuah *sparse matrix* yang akan digunakan sebagai masukan proses pada langkah berikutnya.
- iv. Implementasi proses pemodelan topik.
Pemodelan topik dengan LSA dan K-Means, memanfaatkan matriks yang dibentuk pada langkah (iii). Dalam prosesnya, *sparse matrix* akan direduksi menjadi matriks *tweet-principal components* sehingga akan didapatkan kata kunci untuk masing-masing kluster yang. Pertimbangan untuk menentukan jumlah *principal components* adalah dengan plot nilai *sum explained variance ratio*. Sedangkan metode Elbow untuk plot nilai *inertia* digunakan sebagai pertimbangan dalam menentukan jumlah kluster.
- v. Implementasi proses analisis sentimen.
Analisis sentimen yang dilakukan adalah untuk mendapatkan polaritas suatu opini dalam interval $[-1, 1]$. Aturan yang digunakan adalah aturan yang diusulkan oleh C. J. Hutto dan Eric Gilbert. Hal mengenai daftar kata dan polaritas *lexicon*, penulis menggunakan daftar kata dan polaritas *lexicon* dari Hutto dan Eric.
- vi. Implementasi proses perhitungan *opportunity* dan visualisasi dalam diagram.
Perhitungan *opportunity* adalah dengan memerhatikan parameter *importance* dan *satisfaction*. Dengan menggunakan rumus (2.17), akan dihitung *opportunity* dari topik yang merepresentasikan bidang suatu produk. Kemudian dari setiap pasangan *importance* dan

satisfaction akan divisualisasikan dalam diagram *product landscape map*.

vii. Implementasi desain antarmuka

Implementasi desain antarmuka ditujukan untuk memudahkan pengguna dalam menggunakan sistem yang dibangun.

5. Uji Coba dan Pembahasan

Pada tahap ini dilakukan uji coba terhadap fungsi-fungsi yang sudah diimplementasikan sebelumnya dan dilakukan pembahasan pada hasil yang sudah didapatkan sesuai dengan teori.

6. Penarikan Kesimpulan

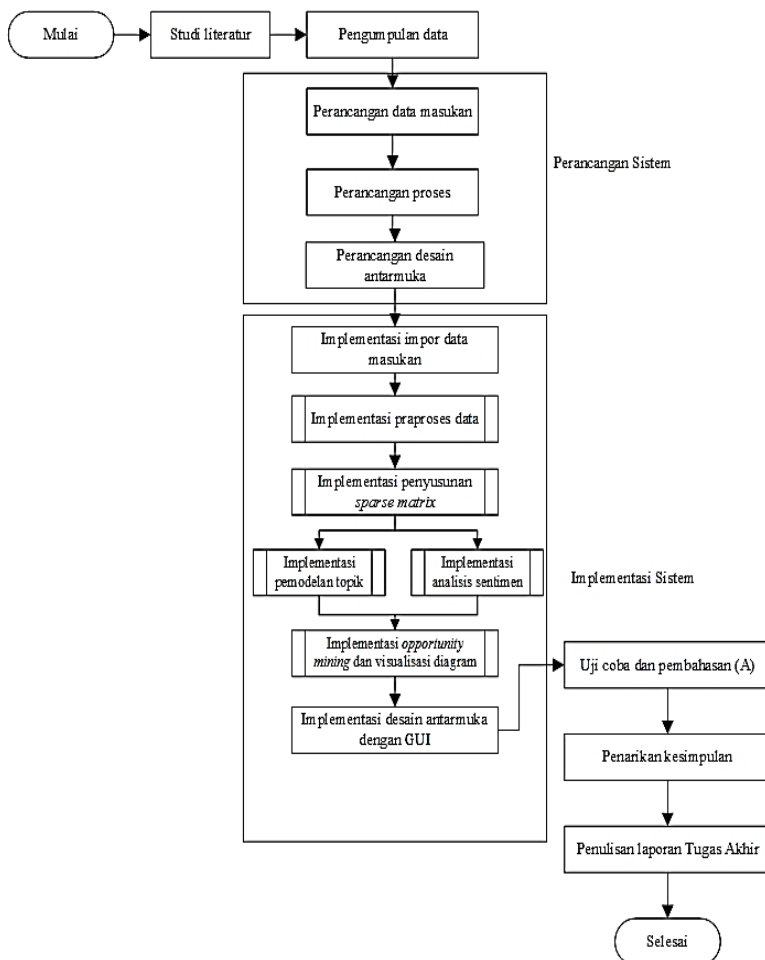
Pada tahap ini dilakukan penarikan kesimpulan dari pembahasan yang sudah dilakukan.

7. Pembuatan Laporan Tugas Akhir

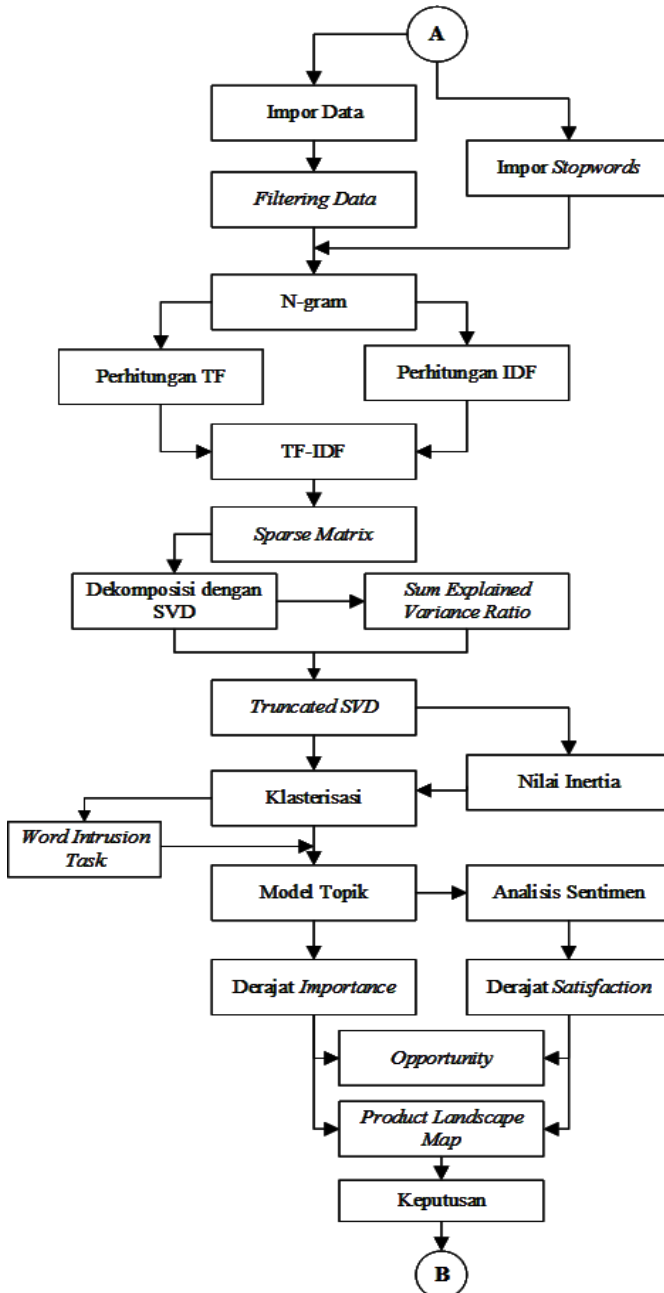
Pada tahap ini semua proses serta pembahasan dirangkum dan ditulis pada buku Tugas Akhir.

3.4 Diagram Blok

Diagram blok dari penelitian Tugas Akhir ini adalah sebagai berikut.



Perincian tahapan uji coba dan pembahasan dapat diperhatikan pada diagram blok berikut ini.



BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini diuraikan dan dijelaskan perancangan sistem serta implementasi sistem yang digunakan dalam penelitian Tugas Akhir ini. Selain itu juga diuraikan rancangan sistem secara manual. Adapun ilustrasi penjelasan proses adalah dengan menggunakan data fiktif (data contoh) untuk memudahkan dalam penjelasan.

4.1 Analisis Sistem

Dalam membangun sebuah perangkat lunak yang dapat menghitung *opportunity* dari bagian suatu produk dan memvisualisasikan bagian suatu produk dalam *product landscape map*, perangkat lunak memerlukan data masukan yang berasal dari Twitter, yakni kumpulan *tweet* berbentuk teks berbahasa Inggris yang kemudian akan diubah menjadi *sparse matrix* dengan TF-IDF. Namun, pada penelitian Tugas Akhir ini, data masukan sudah disimpan dalam suatu berkas berformat .csv dengan *sheet* dan kolom tunggal serta berjudul *header* “text”. Pengguna dapat memasukkan data *tweet* yang sudah disimpan ke dalam perangkat lunak melalui suatu tombol. Pada perhitungan *opportunity* dan visualisasi bagian suatu produk, diperlukan masukan lain pada beberapa parameter yakni jumlah komponen yang dikehendaki, jumlah kluster yang dikehendaki, jumlah kata kunci tiap kluster, dan jumlah iterasi yang dikehendaki. Untuk mendapatkan jumlah komponen dan jumlah kluster yang dikehendaki, pengguna dapat memanfaatkan menu untuk memeriksa jumlah komponen yang memenuhi yakni melalui *sum explained variance ratio* dan juga memeriksa jumlah kluster melalui nilai *inertia*. Adapun batas ambang minimum nilai *sum explained variance ratio* yang dijadikan peringatan adalah 0,3.

Jumlah komponen yang dimaksud adalah nilai k pada *truncated SVD* sehingga *sparse matrix* yang berasal dari data masukan akan memiliki ukuran yang lebih kecil daripada sebelum melalui *truncated SVD*. Setelah itu, sistem akan melakukan *truncated SVD* dan klasterisasi dengan K-Means. Setelah didapatkan klaster pada masing-masing *tweet*, sistem melakukan perhitungan rata-rata nilai polaritas pada masing-masing klaster (perhitungan dilakukan dengan menghitung nilai polaritas *tweet* pada klaster yang sama, kemudian dihitung rata-rata polaritas opini pada klaster tersebut). Kemudian sistem dapat melakukan perhitungan *opportunity* dan visualisasi bagian suatu produk dalam *product landscape map*.

4.1.1 Analisis Kebutuhan Fungsional

Berdasarkan analisis sistem seperti diuraikan di atas, diharapkan sistem mampu melakukan hal-hal sebagai berikut:

1. Sistem mampu mengubah data masukan berupa data *tweet* menjadi *sparse matrix*
2. Sistem mampu melakukan perhitungan *sum explained variance ratio* untuk mendapatkan jumlah komponen yang memiliki nilai *sum explained variance ratio* di atas batas ambang
3. Sistem mampu melakukan *truncated SVD* pada *sparse matrix*
4. Sistem mampu melakukan perhitungan nilai *inertia*
5. Sistem mampu melakukan klasterisasi dengan K-Means *cluster*
6. Sistem mampu melakukan perhitungan nilai polaritas *tweet*
7. Sistem mampu melakukan perhitungan *opportunity* dan memvisualisasikan dalam *product landscape map*

4.1.2 Analisis Kebutuhan Non-Fungsional

Perangkat lunak ini dibangun dengan menggunakan bahasa pemrograman Python dengan library sklearn, pandas, numpy, matplotlib, dan juga tkinter.

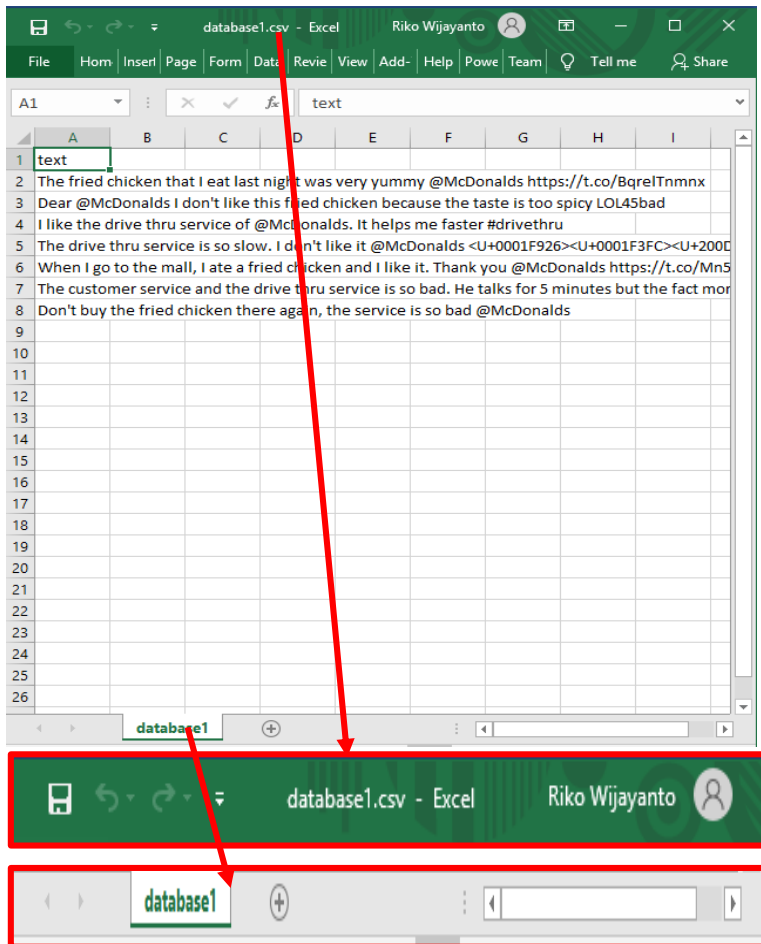
4.2 Perancangan Sistem

Dari hasil analisis sistem, kemudian dirancang suatu program yang diharapkan sesuai dengan tujuan Tugas Akhir ini.

4.2.1 Perancangan Data

Data pada sistem dapat dibedakan menjadi tiga bagian, yakni data masukan sistem, data proses, serta data hasil pengolahan.

1. Data masukan sistem merupakan data *tweet* berbahasa Inggris yang sudah disimpan dalam format .csv dengan *sheet* dan kolom tunggal dengan *header* berjudul “text”. Bila data masukan sudah sesuai, maka program akan berjalan sesuai dengan prosedur. Adapun contoh data masukan yang sudah disimpan ditunjukkan pada Gambar 4.1 pada halaman selanjutnya.
2. Data proses merupakan data yang digunakan dalam pengolahan data masukan hingga didapatkan hasil dari proses tersebut. Adapun data proses dapat diperhatikan pada Tabel 4.1.
3. Data hasil pengolahan pada sistem ini adalah nilai *opportunity* bagian suatu produk dan visualisasi bagian suatu produk dalam *product landscape map*.



Gambar 4.1 Contoh Data Masukan

Tabel 4.1 Tabel Data Proses

No.	Proses	Data Proses	Keterangan
1	Penyusunan <i>sparse matrix</i>	Daftar <i>stopwords</i>	Daftar <i>stopwords</i> adalah data yang berisi kumpulan <i>stopwords</i> yang disimpan dalam berkas dengan format .txt
2	<i>Truncated SVD</i>	<i>Sparse matrix</i>	Matriks yang merupakan representasi dari data masukan yang didapatkan dengan TF-IDF
3		Bilangan asli	Digunakan untuk melakukan <i>truncated SVD</i> serta menguji nilai <i>sum explained variance ratio</i> sehingga didapat matriks <i>tweet-principal components</i>
4	Klasterisasi	Matriks <i>tweet-principal components</i>	Matriks yang merupakan representasi <i>tweet</i> dan komponen utama yang menyusun matriks tersebut
5		Bilangan asli	Digunakan untuk melakukan klasterisasi <i>tweet</i> serta menguji nilai <i>inertia</i>

No.	Proses	Data Proses	Keterangan
6	Analisis Sentimen	Daftar <i>lexicon</i>	Daftar <i>lexicon</i> adalah daftar yang berisi kata perasa (<i>feeling words</i>) dan nilai polaritasnya
7	Perhitungan <i>opportunity mining</i>	Derajat <i>importance</i>	Derajat <i>importance</i> adalah derajat yang menyatakan tingkat kepentingan yang didapatkan dari perhitungan jumlah <i>tweet</i> dalam satu klaster
8		Derajat <i>satisfaction</i>	Derajat <i>satisfaction</i> adalah derajat yang menyatakan tingkat kepuasan yang didapatkan dari perhitungan rata-rata nilai polaritas dalam satu klaster yang sama

4.2.2 Perancangan Proses

Adapun perancangan proses sistem disajikan pada uraian berikut dengan menggunakan bantuan ilustrasi melalui kumpulan *tweet* sebagaimana ditampilkan pada Gambar 4.1 serta diperjelas pada Tabel 4.2 berikut.

Tabel 4.2 Tabel Data Fiktif *Tweet*

The fried chicken that I eat last night was very yummy @McDonalds https://t.co/BqrelTnmnx
Dear @McDonalds I don't like this fried chicken because the taste is too spicy LOL45bad
I like the drive thru service of @McDonalds. It helps me faster #drivethru
The drive thru service is so slow. I don't like it @McDonalds <U+0001F926><U+0001F3FC><U+200D><U+2640><U+FE0F>
When I go to the mall, I ate a fried chicken and I like it. Thank you @McDonalds https://t.co/Mn5vNfGNWj
The customer service and the drive thru service is so bad. He talks for 5 minutes but the fact more than 20 minutes. So poor. @McDonalds
Don't buy the fried chicken there again, the service is so bad @McDonalds

a. Perancangan Proses Impor Data

Proses pertama yang dilakukan adalah mengimpor atau *load* data masukan ke dalam sistem dengan *library* *pandas* menjadi bentuk *dataframe*. Setelah data masukan berhasil diimpor dalam sistem, data masukan dimanipulasi menjadi *list* dengan menggunakan sistem *looping for*. Hal ini dimaksudkan untuk memudahkan dalam mengakses per elemen *list* pada proses selanjutnya.

b. Perancangan Praproses Data

Praproses data yang dimaksud adalah proses menghilangkan beberapa kata, frasa, atau bagian kalimat yang menurut penulis

tidak penting. Adapun yang dimaksud adalah nama mention (@username), url (<https://t.co/BqrelTnmnx>), tagar atau *hashtag* (#tagar), gabungan kata dan angka (misalkan McD100), format lokasi (<U+0001F926><U+0001F3F>), karakter angka lebih dari satu karakter (12,14,456,700), karakter huruf satu digit, dan spasi berlebih. Hasil dari praproses data akan disimpan dalam list yang baru. Adapun ilustrasinya ditunjukkan pada Tabel 4.3 berikut.

Tabel 4.3 Contoh Data Sebelum dan Sesudah Praproses

Sebelum Praproses	Sesudah Praproses
The fried chicken that I eat last night was very yummy @McDonalds https://t.co/BqrelTnmnx	The fried chicken that eat last night was very yummy
Dear @McDonalds I don't like this fried chicken because the taste is too spicy LOL45bad	Dear don like this fried chicken because the taste is too spicy LOLbad
I like the drive thru service of @McDonalds. It helps me faster #drivethru	like the drive thru service of It helps me faster
The drive thru service is so slow. I don't like it @McDonalds <U+0001F926><U+0001F3FC><U+200D><U+2640><U+FE0F>	The drive thru service is so slow don like it
When I go to the mall, I ate a fried chicken and I like it.	When go to the mall ate fried chicken and like it Thank you

Thank you @McDonalds https://t.co/Mn5vNfGNWj	
The customer service and the drive thru service is so bad. He talks for 5 minutes but the fact more than 20 minutes. So poor. @McDonalds	The customer service and the drive thru service is so bad He talks for minutes but the fact more than minutes So poor
Don't buy the fried chicken there again, the service is so bad @McDonalds	Don buy the fried chicken there again the service is so bad

c. Perancangan Penyusunan *Sparse Matrix*

Tweet yang sudah dibersihkan pada tahap praproses data, kemudian dibobotkan sehingga dapat dibentuk *sparse matrix* dengan menggunakan TF-IDF yang dijelaskan pada sub bab 2.3 sebelumnya. Pertama, *tweet* yang sudah dibersihkan, diseleksi untuk diambil kata yang tidak termasuk dalam *stopwords* (*the, at, that, me, dsb*) dimana daftar *stopwords* didapat dari daftar *stopwords* yang juga diimpor dalam sistem menjadi sebuah *list stopwords*.

Tabel 4.4 Contoh Data Sebelum dan Sesudah Seleksi

Sebelum Seleksi <i>Stopwords</i>	Sesudah Seleksi <i>Stopwords</i>
The fried chicken that eat last night was very yummy	Fried chicken eat yummy
Dear don like this fried chicken because the taste is too spicy LOLbad	Fried chicken taste spicy LOLbad
like the drive thru service of It helps me faster	Drive thru service faster

The drive thru service is so slow don like it	Drive thru service slow
When go to the mall ate fried chicken and like it Thank you	Mall ate fried chicken
The customer service and the drive thru service is so bad He talks for minutes but the fact more than minutes So poor	Customer service drive thru service bad talks minutes fact minutes poor
Don buy the fried chicken there again the service is so bad	Buy fried chicken service bad bad

Berikutnya adalah penyusunan fitur kalimat berdasarkan N-gram. Penyusunan fitur dengan N-gram pada interval satu hingga tiga secara terurut dilakukan pada data sesudah diseleksi dari *stopwords*. Penentuan satu hingga tiga kata dalam penyusunan fitur adalah dipilih bebas sesuai dengan keinginan penulis untuk mengatasi kemungkinan keberadaan kata dalam suatu frasa yang tidak dapat dipisahkan, misalkan *drive thru* berbeda arti dengan hanya *drive* dan *thru* secara terpisah.

Tabel 4.5 Contoh Data Sebelum dan Sesudah N-gram

Sebelum N-gram	Sesudah N-gram
Fried chicken eat yummy	“fried”, “chicken”, “eat”, “yummy”, fried chicken”, “chicken eat”, “eat yummy”, “fried chicken eat”, “chicken eat yummy”
Fried chicken taste spicy LOLbad	“fried”, “chicken”, “taste”, “spicy”, LOLbad”, “fried chicken”, “chicken taste”, “taste spicy”, “spicy

	LOLbad", "fried chicken taste", "chicken taste spicy", "taste spicy LOLbad"
Drive thru service faster	"drive", "thru", "service", "faster", "drive thru", "thru service", "service faster", "drive thru service", "thru service faster"
Drive thru service slow	"drive", "thru", "service", "slow", "drive thru", "thru service", "service slow", "drive thru service", "thru service slow"
Mall ate fried chicken	"mall", "ate", "fried", "chicken", "mall ate", "ate fried", "fried chicken"
Customer service drive thru service bad talks minutes fact minutes poor	"customer", "service", "drive", "thru", "service", "bad", "talks", "minutes", "fact", "minutes", "poor", "customer service", "service drive", "drive thru", "thru service", "service bad", "bad talks", "talks minutes", "minutes fact", "fact minutes", "minutes poor", "customer service drive", "service drive thru", "drive thru service", "thru service bad", "service bad talks", "bad talks minutes", "talks minutes fact", "minutes fact minutes", "fact minutes poor"
Buy fried chicken service bad	"buy", "fried", "chicken", "service", "bad", "buy fried", "fried chicken", "chicken service", "service bad", "buy fried chicken", "fried

	chicken service”, ”chicken service bad”
--	---

Langkah terakhir adalah pembobotan *tweet* menjadi vektor baris dengan metode perhitungan TF-IDF. Perhitungan TF-IDF dimulai dengan penghilangan fitur yang hanya muncul sekali pada keseluruhan *tweet* karena dalam kondisi ini fitur dianggap tidak menonjol atau tidak penting. Oleh karena itu, fitur dari data fiktif yang sudah diproses dalam N-gram misalkan seperti di bawah ini.

“chicken”, “drive”, “drive thru”, “drive thru service”, “fried”, “fried chicken”, “service”, “thru”, “thru service”

Perhitungan TF atau *term frequency* dengan menggunakan rumus (2.1) yang diilustrasikan dengan *tweet* pertama adalah ditunjukkan pada Tabel 4.6 berikut.

Tabel 4.6 Tabel TF pada *Tweet* Pertama

Fitur	<i>tf</i>
Chicken	1
Drive	0
Drive thru	0
Drive thru service	0
Fried	1
Fried chicken	1
Service	0
Thru	0
Thru service	0

Perhitungan IDF dengan $m = 7$ dan menggunakan rumus (2.2) adalah tertera pada Tabel 4.7 di bawah ini.

Tabel 4.7 Tabel IDF pada *Tweet* Pertama

Fitur	doc_j	idf_j
Chicken	4	1,243038049
Drive	3	1,367976785
Drive thru	3	1,367976785
Drive thru service	3	1,367976785
Fried	4	1,243038049
Fried chicken	4	1,243038049
Service	4	1,243038049
Thru	3	1,367976785
Thru service	3	1,367976785

Berdasarkan rumus (2.3) menggunakan kombinasi nilai TF dari Tabel 4.6 dan nilai IDF dari Tabel 4.7, dihitung nilai bobot awal untuk fitur-fitur pada *tweet* pertama dan dibentuk sebuah vektor baris. Tabel 4.8 di bawah ini menunjukkan perhitungan nilai bobot awal pada *tweet* pertama.

Tabel 4.8 Tabel TF-IDF *Tweet* Pertama

Fitur	tf	idf_j	$w_j(pra)$
Chicken	1	1,243038049	1,243038049
Drive	0	1,367976785	0
Drive thru	0	1,367976785	0
Drive thru service	0	1,367976785	0
Fried	1	1,243038049	1,243038049
Fried chicken	1	1,243038049	1,243038049

Service	0	1,243038049	0
Thru	0	1,367976785	0
Thru service	0	1,367976785	0

Normalisasi bobot awal, $w(pra)$, untuk didapatkan bobot akhir, w , dengan rumus (2.4) diawali dengan perhitungan pembagi, yakni sebagai berikut.

$$\sum_{j=1}^n w_{j(pra)}^2 = 1,243038049^2 + 0 + 0 + 0 + 1,243038049^2 + 1,243038049^2 + 0 + 0 + 0 = 4,635430774$$

Sehingga bobot akhir untuk *tweet* pertama adalah pada Tabel 4.9 di bawah ini.

Tabel 4.9 Tabel Bobot Akhir pada *Tweet* Pertama

Fitur	$w_j(pra)$	w_j
Chicken	1,243038049	0,5773502692
Drive	0	0
Drive thru	0	0
Drive thru service	0	0
Fried	1,243038049	0,5773502692
Fried chicken	1,243038049	0,5773502692
Service	0	0
Thru	0	0
Thru service	0	0

Bila dilakukan perhitungan pada semua *tweet* dan disusun berdasarkan Tabel 2.1, maka akan didapatkan *sparse matrix* yang dimisalkan dengan matriks A berukuran 7×9 di bawah ini.

$$A = \begin{bmatrix} 0,58 & 0 & 0 & 0 & 0,58 & 0,58 & 0 & 0 & 0 \\ 0,58 & 0 & 0 & 0 & 0,58 & 0,58 & 0 & 0 & 0 \\ 0 & 0,42 & 0,42 & 0,42 & 0 & 0 & 0,35 & 0,42 & 0,42 \\ 0 & 0,42 & 0,42 & 0,42 & 0 & 0 & 0,35 & 0,42 & 0,42 \\ 0,58 & 0 & 0 & 0 & 0,58 & 0,58 & 0 & 0 & 0 \\ 0 & 0,36 & 0,36 & 0,36 & 0 & 0 & 0,60 & 0,36 & 0,36 \\ 0,5 & 0 & 0 & 0 & 0,5 & 0,5 & 0,5 & 0 & 0 \end{bmatrix}$$

d. Perancangan Proses Pemodelan Topik

Pemodelan topik dengan menggunakan kombinasi *Latent Semantic Analysis* dan *K-Means cluster* dimulai dengan mendekomposisi *sparse matrix* menjadi matriks SEV^T . Asumsikan *sparse matrix* yang digunakan adalah matriks $A_{7 \times 9}$, maka transpose dari matriks A adalah A^T dengan dimensi 9×7 , sehingga didapat $B = AA^T$ sesuai dengan penjelasan pada sub bab 2.4.

$$A^T = \begin{bmatrix} 0,58 & 0,58 & 0 & 0 & 0,58 & 0 & 0,5 \\ 0 & 0 & 0,42 & 0,42 & 0 & 0,36 & 0 \\ 0 & 0 & 0,42 & 0,42 & 0 & 0,36 & 0 \\ 0 & 0 & 0,42 & 0,42 & 0 & 0,36 & 0 \\ 0,58 & 0,58 & 0 & 0 & 0,58 & 0 & 0,5 \\ 0,58 & 0,58 & 0 & 0 & 0,58 & 0 & 0,5 \\ 0 & 0 & 0,35 & 0,35 & 0 & 0,60 & 0,5 \\ 0 & 0 & 0,42 & 0,42 & 0 & 0,36 & 0 \\ 0 & 0 & 0,42 & 0,42 & 0 & 0,36 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0,87 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0,87 \\ 0 & 0 & 1 & 1 & 0 & 0,96 & 0,18 \\ 0 & 0 & 1 & 1 & 0 & 0,96 & 0,18 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0,87 \\ 0 & 0 & 0,96 & 0,96 & 0 & 1 & 0,3 \\ 0,87 & 0,87 & 0,18 & 0,18 & 0,87 & 0,3 & 1 \end{bmatrix}$$

Nilai-nilai eigen dari matriks B dapat ditentukan dengan menghitung akar-akar persamaan karakteristiknya. Persamaan karakteristik dari matriks B dapat dicari dengan rumus $\det(\lambda I - B) = 0$.

$$\det(\lambda I - B) = \begin{vmatrix} \lambda - 1 & -1 & 0 & 0 & -1 & 0 & -0,87 \\ -1 & \lambda - 1 & 0 & 0 & -1 & 0 & -0,87 \\ 0 & 0 & \lambda - 1 & -1 & 0 & -0,96 & -0,18 \\ 0 & 0 & -1 & \lambda - 1 & 0 & -0,96 & -0,18 \\ -1 & -1 & 0 & 0 & \lambda - 1 & 0 & -0,87 \\ 0 & 0 & -0,96 & -0,96 & 0 & \lambda - 1 & -0,3 \\ -0,87 & -0,87 & -0,18 & -0,18 & -0,87 & -0,3 & \lambda - 1 \end{vmatrix} = 0$$

Maka didapat nilai-nilai eigen dari matriks B adalah 3,84; 2,96; 0,21; 0; 0; 0; 0 dimana 0 adalah pembulatan untuk suatu nilai yang kecil. Sehingga, nilai singular yang bersesuaian adalah 1,96; 1,72; 0,46; 0; 0; 0; 0. Kemudian dibentuk matriks diagonal Σ dengan elemen nilai-nilai singular sebagai berikut.

$$\Sigma = \begin{bmatrix} 1,96 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,72 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,46 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Selanjutnya akan dicari matriks singular kanan dan matriks singular kiri. Perhitungan dengan dimensi matriks yang lebih kecil dapat diperhatikan pada Lampiran B. Adapun matriks singular kiri dan kanan yang terbentuk dari proses sebelumnya adalah sebagai berikut.

$$S = \begin{bmatrix} 0,49 & -0,13 & -2,3 & 1,29 & -2,28 & 0,78 & -1 \\ 0,49 & -0,14 & -0,23 & -0,13 & -0,005 & -0,32 & -0,75 \\ 0,11 & 0,56 & -0,26 & -0,32 & 0,68 & -0,20 & 0,04 \\ 0,11 & 0,56 & -0,26 & -0,32 & -0,68 & -0,20 & -0,04 \\ -0,50 & -0,14 & -0,23 & 0,13 & 0,18 & -0,46 & -0,65 \\ 0,13 & 0,56 & 0,35 & 0,74 & 0 & 0 & 0 \\ 0,48 & 0,0008 & 0,76 & -0,45 & 0 & 0 & 0 \end{bmatrix}$$

$$V^T = \begin{bmatrix} 0,58 & 0,07 & 0,07 & 0,07 & 0,56 & 0,56 & 0,20 & 0,07 & 0,07 \\ -0,13 & 0,39 & 0,39 & 0,39 & -0,13 & -0,13 & 0,43 & 0,39 & 0,39 \\ -0,06 & -0,20 & -0,20 & -0,20 & -0,06 & -0,06 & 0,88 & -0,20 & -0,20 \\ -0,37 & 0,46 & -0,04 & -0,21 & -0,24 & 0,62 & 0 & 0,08 & 0,08 \\ -0,05 & 0,65 & 0,39 & -0,26 & -0,20 & -0,15 & 0 & -0,39 & -0,39 \\ 0,42 & 0,43 & -0,66 & 0,18 & 0,0007 & -0,42 & 0 & 0,003 & 0,003 \\ -0,06 & -0,004 & -0,25 & 0,29 & 0,70 & -0,11 & 0 & 0 & 0 \end{bmatrix}$$

Setelah didekomposisi menjadi matriks $S\Sigma V^T$, maka dilakukan *truncated SVD* yakni dengan memilih k *principal components* dengan $1 \leq k < 7$. Misalkan k yang dipilih adalah tiga, maka dipertahankan tiga baris dan tiga kolom pertama dari matriks Σ , dan dipertahankan tujuh baris pertama dan tiga kolom pertama dari matriks S sehingga menjadi matriks *tweet-principal components*. Perlu diperhatikan bahwa kriteria pengambilan jumlah k harus memenuhi ambang batas minimum *sum explained variance ratio* yang nilainya lebih dari 0,3. Pengambilan batas ambang 0,3 adalah didasarkan pada keputusan subyektif penulis.

Adapun dari matriks hasil *truncated SVD* sebagaimana di atas, misalkan Z , diperiksa apakah memiliki nilai *sum explained variance ratio* lebih dari 0,3 atau tidak. Bila tidak memenuhi, maka jumlah k ditambah dan dihitung kembali. Hal itu dilakukan pada batas interval tertentu dan dilakukan pengujian secara terus-menerus.

$$Z = \begin{bmatrix} 0,966689 & -0,232185 & -0,107712 \\ 0,966689 & -0,232185 & -0,107712 \\ 0,219048 & 0,968170 & -0,121097 \\ 0,219048 & 0,968170 & -0,121097 \\ 0,966689 & -0,232185 & -0,107712 \\ 0,247487 & 0,955256 & 0,161974 \\ 0,937797 & 0,013636 & 0,346916 \end{bmatrix}$$

Perhitungan secara manual *sum explained variance ratio* dari matriks Z di atas dengan rumus (2.6) hingga rumus (2.8) adalah sebagai berikut.

$$\sigma_{trunc_1}^2 = var(0,966689; \dots; \dots; 0,937797) = 0,152843$$

$$\sigma_{trunc_2}^2 = var(-0,232185; \dots; \dots; 0,013636) = 0,375374$$

$$\sigma_{trunc_3}^2 = var(-0,107712; \dots; \dots; 0,346916) = 0,035044$$

$$\text{Sehingga } \sigma_{trunc_1}^2 + \sigma_{trunc_2}^2 + \sigma_{trunc_3}^2 = 0,5 \quad (H1)$$

Selanjutnya,

$$\sigma_1^2 = var(0,58; 0; 0; 0; 0,58; 0,58; 0; 0; 0) = 0,08333$$

$$\sigma_2^2 = var(0,58; 0; 0; 0; 0,58; 0,58; 0; 0; 0) = 0,08333$$

...

$$\sigma_7^2 = var(0,5; 0; 0; 0; 0,5; 0,5; 0,5; 0,0) = 0,069444$$

$$\text{Sehingga } \sigma_1^2 + \sigma_2^2 + \dots + \sigma_7^2 = 0,5 \quad (H2)$$

Dari $H1$ dan $H2$ didapatkan nilai *sum explained variance ratio* seperti di bawah ini.

$$\sigma_{ratio}^2 = \frac{H1}{H2} = 1$$

Berdasarkan nilai *sum explained variance ratio* yang menunjukkan angka 1, maka *truncated SVD* dengan $k = 3$ diterima. Namun, dalam implementasinya, perhitungan *sum explained variance ratio* dilakukan lebih awal sebagai bahan pertimbangan pemilihan k .

Klasterisasi *tweet* untuk mengelompokkan *tweet* dengan menggunakan *K-Means cluster* yaitu dengan memanfaatkan matriks hasil *truncated SVD* (dalam proses di atas adalah matriks Z) sebagai masukan. Atribut yang digunakan adalah sel kolom dari matriks hasil *truncated SVD*.

Misalkan klasterisasi pada data fiktif dengan menggunakan matriks Z dan inisiasi awal dua klaster adalah sebagai berikut:

- I. Pada iterasi pertama. Misalkan diinisiasi dua klaster, maka diambil dua *tweet* secara acak sebagai titik pusat klaster awal. Misalkan *tweet 1* dan *tweet 3* adalah titik pusat klaster awal (*centroid*) 1 dan 2 secara berurutan.
- II. Tahap selanjutnya adalah perhitungan jarak masing-masing *tweet* dengan titik pusat klaster dengan menggunakan jarak *euclidean*.

Misalkan jarak *tweet 2* dengan *centroid 1* dan *centroid 2* berturut-turut sebagai berikut.

Dengan menggunakan rumus pada 2.9 didapat:

$$d_{2,1} = \sqrt{(0,966 - 0,966)^2 + (-0,232 - (-0,232))^2 + (-0,107 - (-0,107))^2} = 0$$

$$d_{2,2} = \sqrt{(0,966 - 0,219)^2 + (0,968 - (-0,232))^2 + (-0,121 - (-0,107))^2} = 1,073$$

Karena $d_{2,1} < d_{2,2}$, maka *tweet 2* akan dikelompokkan pada satu klaster dengan *centroid 1* atau *tweet 1*.

Selanjutnya adalah *tweet 4* dengan *centroid 1* dan *centroid 2* berturut-turut adalah sebagai berikut.

$$d_{4,1} = \sqrt{(0,219 - 0,966)^2 + (0,968 - (-0,232))^2 + (-0,121 - (-0,107))^2}$$

$$= 1,414$$

$$d_{4,2} = \sqrt{(0,219 - 0,219)^2 + (-0,232 - (-0,232))^2 + (-0,107 - (-0,107))^2}$$

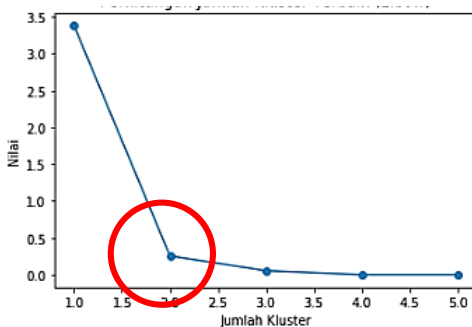
$$= 0$$

Karena $d_{4,2} < d_{4,1}$, maka *tweet* 4 akan dikelompokkan pada satu kluster dengan *centroid* 2 atau *tweet* 2.

Perhitungan dilakukan terus-menerus hingga *tweet* terakhir.

- III. Setelah semua *tweet* dikluster pada *centroid* terdekat, maka dilakukan perhitungan nilai rata-rata pada masing-masing atribut *centroid* dan nilai rata-rata dijadikan *centroid* baru untuk masing-masing kluster. Dan langkah b dilakukan kembali.
- IV. Bila tidak ada perubahan *centroid* atau telah memenuhi batas maksimum iterasi, maka proses selesai.

Pertimbangan dalam pemilihan jumlah kluster ditentukan dengan memerhatikan nilai *inertia* dengan metode *elbow* dengan menguji pada interval satu hingga lima kluster. Sehingga pada data fiktif tersebut, dapat digambarkan plot nilai perhitungan dengan grafik serta rumus (2.10) adalah sebagai berikut.



Gambar 4.2 Grafik Nilai *Inertia*

Tampak pada gambar 4.2 bahwa nilai *inertia* yang masuk dalam daerah perkiraan adalah ketika jumlah kluster adalah sebanyak dua kluster. Sehingga data fiktif tersebut akan dikluster dengan jumlah kluster sebanyak dua. Identifikasi topik tiap kluster dapat ditentukan dengan melihat n kata kunci yang memiliki n bobot terbesar pada tiap klasternya. Hasil klusterisasi pada data fiktif dapat diperhatikan pada Tabel 4.10 berikut.

Tabel 4.10 Tabel Hasil Klusterisasi Data Fiktif

<i>Tweet</i>	Nomor Kluster
The fried chicken that I eat last night was very yummy @McDonalds https://t.co/BqrelTnmnx	1
Dear @McDonalds I don't like this fried chicken because the taste is too spicy LOL45bad	1
I like the drive thru service of @McDonalds. It helps me faster #drivethru	2
The drive thru service is so slow. I don't like it @McDonalds <U+0001F926><U+0001F3FC><U+200D><U+2640><U+FE0F>	2
When I go to the mall, I ate a fried chicken and I like it. Thank you @McDonalds https://t.co/Mn5vNfGNWj	1
The customer service and the drive thru service is so bad. He talks for 5 minutes but the fact more than 20 minutes. So poor. @McDonalds	2

Don't buy the fried chicken there again, the service is so bad @McDonalds	1
---	---

Hasil akhir dari pemodelan topik adalah berupa kata kunci dan jumlah *tweet* tiap-tiap klaster yang dapat diperhatikan pada Tabel 4.11 berikut.

Tabel 4.11 Tabel Hasil Pemodelan Topik

Nomor Klaster	Jumlah <i>Tweet</i>	Kata Kunci
1	4	<i>Fried, chicken, fried chicken</i>
2	3	<i>Service, drive, thru service</i>

Makna dan pelabelan hasil pemodelan topik di atas adalah:

- Klaster 1 adalah kumpulan *tweet* yang mendiskusikan tentang topik *fried chicken* dengan jumlah *tweet* adalah sebanyak 4.
- Klaster 2 adalah kumpulan *tweet* yang mendiskusikan tentang topik *drive thru* dengan jumlah *tweet* adalah sebanyak 3.

c. Perancangan Proses Analisis Sentimen

Analisis sentimen berdasarkan *lexicon* dilakukan melalui beberapa proses. Pertama adalah dengan mengimpor *lexicon* ke dalam sistem. Daftar *lexicon* yang digunakan adalah dari C. J. Hutto dan Eric Gilbert dan disimpan dalam berkas berformat .txt. Adapun kutipan *lexicon* dapat dilihat pada Gambar 4.3. Terdapat sejumlah 7517 *lexicon* yang berada dalam daftar beserta nilai polaritasnya. Data-data *lexicon* kemudian disimpan dalam sebuah

kamus dengan atribut *lexicon* dan polaritasnya secara bersesuaian. Selain itu, kata *booster* seperti *very*, *most*, *amazingly*, *absolutely*, dan sebagainya serta kata berarti negasi seperti *don't*, *ain't*, dan sebagainya juga dideklarasikan pada awal sistem.

```

1 $: -1.5 0.80623 [-1, -1, -1, -1, -3, -1, -3, -1, -2, -1]
2 %: -0.4 1.0198 [-1, 0, -1, 0, 0, -2, -1, 2, -1, 0]
3 %: -1.5 1.43178 [-2, 0, -2, -2, -1, 2, -2, -3, -2, -3]
4 &: -0.4 1.42829 [-3, -1, 0, 0, -1, -1, -1, 2, -1, 2]
5 &: -0.7 0.64031 [0, -1, -1, -1, 1, -1, -1, -1, -1, -1]
6 ( '){ } 1.6 0.66332 [1, 2, 2, 1, 1, 2, 2, 1, 3, 1]
7 (% -0.9 0.9434 [0, 0, 1, -1, -1, -1, -2, -2, -1, -2]
8 (' -: 2.2 1.16619 [4, 1, 4, 3, 1, 2, 3, 1, 2, 1]
9 (' : 2.3 1.9 [1, 3, 3, 2, 2, 4, 2, 3, 1, 2]
10 ( -: 2.1 0.53852 [2, 2, 2, 1, 2, 3, 2, 2, 3, 2]
11 (* 1.1 1.3578 [2, 1, 1, -1, 1, 2, 2, -1, 2, 2]
12 (-% -0.7 1.26886 [-1, 2, 0, -1, -1, -2, 0, 0, -3, -1]
13 (-* 1.3 1.26886 [4, 1, 2, 0, 2, -1, 1, 2, 1, 1]
14 (-: 1.6 0.8 [2, 2, 1, 3, 1, 1, 1, 3, 1, 1]
15 (-:0 2.8 0.27178 [3, 2, 3, 4, 3, 2, 3, 1, 4, 3]
16 (-:< -0.4 2.15407 [-3, 3, -1, -1, 2, -1, -2, 3, -3, -1]
17 (-:o 1.5 0.67082 [3, 1, 1, 2, 2, 2, 1, 1, 1, 1]
18 (-:O 1.5 0.67082 [3, 1, 1, 2, 2, 2, 1, 1, 1, 1]

```

Gambar 4.3 Kutipan Daftar *Lexicon*

Berikutnya, dilakukan tokenisasi pada kalimat atau penguraian kalimat menjadi token-token secara independen dan disimpan dalam sebuah *list*. Tokenisasi secara standar dilakukan dengan mengidentifikasi dua token berbeda yang dipisahkan oleh keberadaan spasi di antara dua token tersebut.

Pengubahan kata yang diikuti oleh suatu tanda baca tanpa spasi (misal: kata!) atau tanda baca yang diikuti suatu kata tanpa spasi (misal: !kata) menjadi kata tanpa tanda baca dilakukan pada hasil tokenisasi. Inisiasi sebuah *list* untuk menyimpan nilai polaritas dilakukan untuk mempermudah dalam perhitungan, misalkan *list* diberi nama *sentimen*.

Langkah-langkah yang dilakukan sebagai berikut.

<p><i>1. Kelas utama</i></p> <ul style="list-style-type: none"> • <i>Untuk setiap token:</i> <ol style="list-style-type: none"> a. <i>Inisiasi nilai=0</i> b. <i>Simpan indeks token</i> c. <i>Ubah token menjadi huruf non-kapital, bila token ada dalam kata booster, masukkan nilai=0 dalam sentimen.</i> d. <i>Jika indeks < jumlah token-1 dan token = kind dan token ke indeks+1 = of, masukkan nilai=0 dalam sentimen</i> e. <i>Nilai= self.method hitung_valensi(nilai, tokenisasi, token, indeks, sentimen)</i> • <i>sentimen = self. Method konjungsi(tokenisasi, sentimen)</i> • <i>Polaritas = self. Method nilai_valensi(sentimen, kalimat-asli)</i> • <i>Kembalikan nilai polaritas</i>
<p><i>2. Method hitung_valensi</i></p> <ul style="list-style-type: none"> • <i>Cek jumlah token dalam tokenisasi yang ditulis dengan huruf kapital, jika $0 < \text{jumlah} < \text{jumlah token keseluruhan}$, simpan kapital = TRUE</i> • <i>Ubah token menjadi huruf non-kapital</i> <ol style="list-style-type: none"> a. <i>Jika, token berada dalam kamus lexicon,</i> <ul style="list-style-type: none"> ○ <i>nilai=nilai polaritas</i>

- Bila token yang diubah menjadi huruf kapital dan kapital = TRUE
 - Jika nilai > 0, maka nilai + 0,733
 - Selain itu, nilai - 0,733
- Untuk a dari 0 hingga 2:
 - Jika indeks > a dan token ke indeks-(a+1) tidak ada dalam kamus lexicon, jalankan method skalar_fungsi(token ke indeks-(a+1), nilai, kapital)
 - ✓ Jika a=1 dan skalar ≠ 0, maka skalar = skalar × 0,95
 - ✓ Jika a=2 dan skalar ≠ 0, maka skalar = skalar × 0,9
 - ✓ Nilai = nilai + skalar
 - ✓ Nilai = self.method apa_negatif(nilai, tokenisasi, a, indeks)
- Nilai = self.method apa_least (nilai, tokenisasi, indeks)
- b. Tambahkan nilai dalam sentimen
- c. Kembalikan sentimen ke fungsi awal

3. Method skalar_fungsi

- Inisiasi skalar = 0
- Ubah token menjadi non-kapital
- Jika token berada dalam kata booster:
 - a. Cek jenis kata booster (positif atau negatif)
 - b. Jika nilai < 0, maka skalar = skalar × -1
 - c. Jika token yang diubah menjadi kapital dan kapital=TRUE,
 - Jika nilai > 0, skalar = skalar + 0,733
 - Selain itu, skalar = skalar - 0,733
- Kembalikan skalar

4. Method *apa_negatif*

- *Ubah setiap token dalam tokenisasi menjadi huruf non-kapital*
- *Jika $a=0$, jalankan method *cek_negatif*(token ke indeks- $(a+1)$), jika keluaran adalah TRUE, maka nilai = nilai $\times -0,733$*
- *Jika $a = 1$,*
 - a. Jika token ke indeks-2 = “never”, dan (token ke indeks-1 = “so” atau token ke indeks-1 = “this”), maka nilai = nilai $\times 1,25$*
 - b. Jika tidak, jika token ke indeks-2 = “without” dan token ke indeks-1 = “doubt”, maka nilai=nilai*
 - c. Jika tidak, jalankan method *negatif*(token ke indeks- $(a+1)$), jika keluaran adalah TRUE, maka nilai = nilai $\times -0,733$*
- *Jika $a=2$*
 - a. Jika token ke indeks-3 = “never”, dan ((token ke indeks-2 = “so” atau token ke indeks-2 = “this”) atau (token ke indeks-1 = “so” atau token ke indeks-1 = “this”)), maka nilai = nilai $\times 1,25$*
 - b. Jika tidak, jika token ke indeks-2 = “without” dan (token ke indeks-2 = “doubt” atau token ke indeks-1 = “doubt”), maka nilai=nilai*
 - c. Jika tidak, jalankan method *negatif*(token ke indeks- $(a+1)$), jika keluaran adalah TRUE, maka nilai = nilai $\times -0,733$*
- *Kembalikan nilai*

<p>5. Method <i>apa_least</i></p> <ul style="list-style-type: none"> • Jika indeks > 1 dan token ke indeks-1 tidak dalam kamus <i>lexicon</i> dan token ke indeks-1 = "least", jika token ke indeks-2 ≠ "at" dan token ke indeks-2 ≠ "very", maka nilai = nilai × -0,733 • Jika tidak, jika indeks = 0 dan token ke indeks-1 tidak dalam kamus <i>lexicon</i> dan token ke indeks-1 = "least", maka nilai = nilai × -0,733 • Kembalikan nilai
<p>6. Method <i>cek_negatif</i></p> <ul style="list-style-type: none"> • Inisiasi status = TRUE • Jika token ada dalam negasi, kembalikan TRUE • Jika token mengandung "n't" kembalikan TRUE • Jika token = "least" dan didahului "at", kembalikan TRUE • Selain itu, kembalikan FALSE
<p>7. Method <i>konjungsi</i></p> <ul style="list-style-type: none"> • Ubah setiap token dalam tokenisasi menjadi huruf non-kapital • Jika kata "but" berada dalam tokenisasi <ul style="list-style-type: none"> a. Simpan indeks but b. Untuk setiap nilai dalam sentimen <ul style="list-style-type: none"> ○ Simpan indeks nilai ○ Jika indeks_nilai < indeks but, maka ganti nilai pada indeks_nilai dengan nilai × 0,5 ○ Jika tidak, jika indeks_nilai > indeks but, maka ganti nilai pada indeks_nilai dengan nilai × 1,5 • Kembalikan sentimen
<p>8. Method <i>nilai_valensi</i></p> <ul style="list-style-type: none"> • Jika sentimen memiliki elemen

<p>a. jumlahkan seluruh elemen sentimen, simpan sebagai nilai awal</p> <p>b. hitung polaritas tanda dengan method <i>tambah_tanda(kalimat-asli)</i></p> <p>c. Jika total awal > 0, maka total awal = total awal + polaritas tanda</p> <p>d. Jika tidak, jika total awal < 0, maka total awal = total awal – polaritas tanda</p> <p>e. hitung total akhir dengan method normalisasi(total awal)</p> <ul style="list-style-type: none"> • Jika tidak memiliki elemen, kembalikan 0 • Kembalikan total akhir
<p>9. Method <i>tambah_tanda</i></p> <ul style="list-style-type: none"> • <i>polaritas_seru</i> = jalankan method <i>tanda1_fungsi(kalimat_asli)</i> • <i>polaritas_tanya</i> = jalankan method <i>tanda2_fungsi(kalimat_asli)</i> • <i>hasil_tanda</i> = <i>polaritas_seru</i> + <i>polaritas_tanya</i> • kembalikan <i>hasil_tanda</i>
<p>10. Method <i>tanda1_fungsi</i></p> <ul style="list-style-type: none"> • Hitung jumlah tanda seru dalam kalimat • Jika jumlah tanda seru ≤ 4, maka polaritas=jumlah tanda seru $\times 0,292$ • Jika jumlah tanda seru > 4, maka polaritas = $4 \times 0,292$ • Kembalikan polaritas
<p>11. Method <i>tanda2_fungsi</i></p> <ul style="list-style-type: none"> • Hitung jumlah tanda tanya dalam kalimat • Bila tanda tanya berjumlah 1 hingga 3, maka polaritas = jumlah tanda tanya $\times 0,292$ • Bila tanda tanya berjumlah > 4. Polaritas = 0,96

<ul style="list-style-type: none"> • <i>Kembalikan polaritas</i>
<i>12. Method normalisasi</i>
<ul style="list-style-type: none"> • <i>Nilai normalisasi = $\frac{\text{total awal}}{\sqrt{(\text{total awal})^2 + 20}}$</i> • <i>Jika nilai normalisasi > 1, kembalikan 1</i> • <i>Jika nilai normalisasi < -1, kembalikan -1</i> • <i>Jika tidak, kembalikan nilai normalisasi</i>

Sehingga bila diterapkan pada data fiktif, akan didapatkan hasil seperti Tabel 4.12 berikut.

Tabel 4.12 Tabel Hasil Analisis Sentimen

<i>Tweet</i>	Nilai Polaritas
The fried chicken that I eat last night was very yummy @McDonalds https://t.co/BqrelTnmnx	0,516
Dear @McDonalds I don't like this fried chicken because the taste is too spicy LOL45bad	-0,238
I like the drive thru service of @McDonalds. It helps me faster #drivethru	0,569
The drive thru service is so slow. I don't like it @McDonalds <U+0001F926><U+0001F3FC><U+200D><U+2640><U+FE0F>	-0,292
When I go to the mall, I ate a fried chicken and I like it. Thank you @McDonalds https://t.co/Mn5vNfGNWj	0,557

The customer service and the drive thru service is so bad. He talks for 5 minutes but the fact more than 20 minutes. So poor. @McDonalds	-0,812
Don't buy the fried chicken there again, the service is so bad @McDonalds	-0,615

f. Perancangan Proses *Opportunity Mining*

Tahap selanjutnya adalah perhitungan derajat *importance*, *satisfaction*, dan perhitungan nilai *opportunity*. Berdasarkan Tabel 4.11 dan Tabel 4.12, dan dengan aturan-aturan yang sudah dijelaskan pada sub bab 2.5 didapatkan suatu kesimpulan seperti yang dituliskan pada Tabel 4.13 dan Tabel 4.14 berikut.

Tabel 4.13 Tabel Hasil Perhitungan Awal

Nomor Klaster	Jumlah <i>Tweet</i>	Rata-Rata Polaritas
1	4	0,1422
2	3	-0,211

Dengan menggunakan rumus (2.18) dihitung derajat *importance* sebagai berikut.

$$importance_1 = 10 \times \frac{4 - 3}{4 - 3} = 10$$

$$importance_2 = 10 \times \frac{3 - 3}{4 - 3} = 0$$

Dengan menggunakan rumus (2.19) dihitung derajat *satisfaction* sebagai berikut.

$$satisfaction_1 = 10 \times \frac{0,1411 - (-0,211)}{0,1411 - (-0,211)} = 10$$

$$satisfaction_2 = 10 \times \frac{(-0,211) - (-0,211)}{0,1411 - (-0,211)} = 0$$

Dan dengan rumus (2.20) didapatkan nilai *opportunity* masing-masing topik.

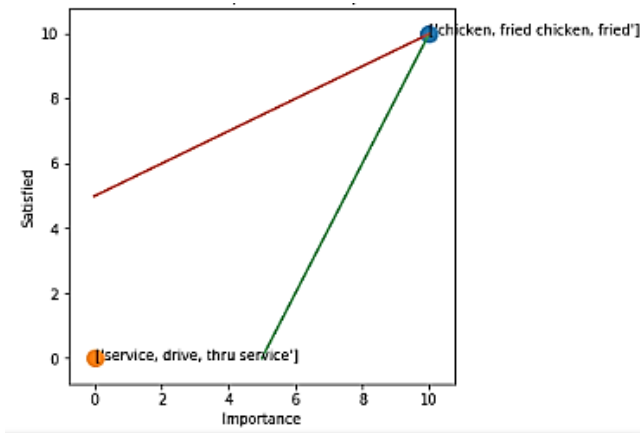
$$opportunity_1 = 10 + \max(10 - 10, 0) = 10$$

$$opportunity_2 = 0 + \max(0 - 0, 0) = 0$$

Tabel 4.14 Tabel Kesimpulan *Opportunity Mining*

Nomor Klaster	Derajat Importance	Derajat Satisfaction	Opportunity	Kata Kunci
1	10	10	10	<i>Fried, chicken, fried chicken</i>
2	0	0	0	<i>Service, drive, thru service</i>

Bila divisualisasikan dalam *product landscape map* didapatkan hasil seperti Gambar 4.4.



Gambar 4.4 Diagram *Product Landscape Map*

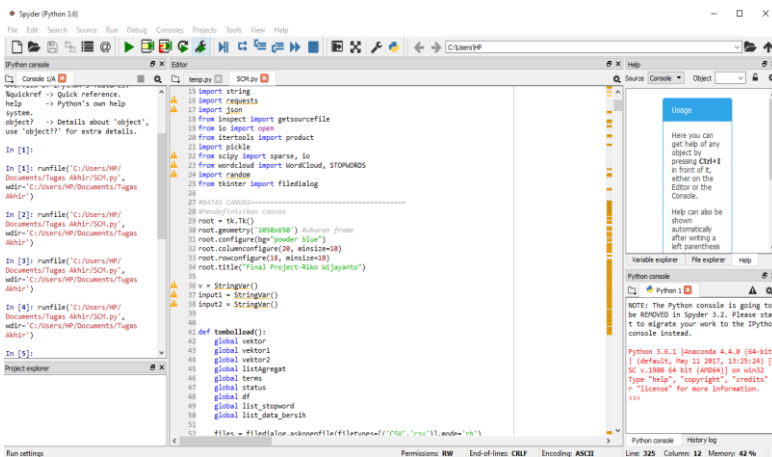
Tampak dari diagram bahwa kedua bagian produk berada dalam wilayah di antara dua garis walaupun salah satu berada di ujung batas wilayah. Ini mengindikasikan bahwa kedua bagian produk berada pada bagian *served-right* atau dilayani dengan tepat.

4.2.3 Perancangan Desain Antarmuka

Desain antarmuka sistem diperlukan untuk memudahkan pengguna dalam mengoperasikan perangkat lunak. Desain antarmuka pada python dapat dibangun dengan menggunakan *library* tkinter. Adapun desain antarmuka yang dirancang adalah dengan menampilkan beberapa pilihan.

1. Tombol *load* data melalui berkas .csv
2. Tombol *load model* yang memungkinkan pengguna memasukkan hasil *truncated SVD*, hasil seleksi fitur, hasil keluaran kata kunci, dan juga data asli dalam bentuk .txt.

3. Tombol *truncated* untuk memperoleh nilai *sum explained variance ratio* dengan memasukkan minimum, maksimum, dan langkah uji coba nilai *k*.
4. Tombol *cluster* untuk memperoleh nilai *inertia* dengan memasukkan minimum, maksimum, langkah uji coba jumlah kluster dan nilai *k* yang dipilih.
5. Tombol *predict* untuk menghitung *opportunity* dan memvisualisasikan hasil dalam *product landscape map* dengan memasukkan nilai *k*, jumlah kluster, jumlah kata kunci tiap kluster, dan jumlah iterasi maksimum.
6. Bagian khusus menyimpan hasil matriks hasil *truncated SVD*, fitur-fitur, kata kunci, dan juga data masukan dalam bentuk .txt.
7. Serta suatu menu *details* untuk menampilkan data yang digunakan, nilai *sum explained variance ratio*, nilai *inertia*, hasil kluster tiap dokumen, rata-rata nilai polaritas tiap kluster, hasil perhitungan *opportunity mining*, dan juga menu tambahan untuk menghitung polaritas suatu kalimat lain secara langsung.



Gambar 4.5 Perancangan Desain Antarmuka

4.3 Implementasi Sistem

Dari hasil perancangan sistem, akan diimplementasikan pada program untuk membangun sebuah perangkat lunak. Adapun program yang digunakan adalah Spyder.

4.3.1 Implementasi Proses Impor Data

Implementasi proses impor data adalah dengan menggunakan *pandas* untuk mengimpor berkas berformat *.csv* menjadi berbentuk *dataframe*, kemudian dilakukan *looping for* untuk memasukkan berkas ke dalam *list* kosong yang sudah diinisiasi pada awal program. Adapun *listing* program yang digunakan adalah sebagai berikut.

```
import pandas as pd
files =
filedialog.askopenfile(filetypes=[('CSV',
'csv')],mode='rb')
    df = pd.read_csv(files)

    v.set(files)

    listAgregat = [] #inisiasi list kosong
    for i in df.text:
        listAgregat.append(i)
#manipulasi data dalam list
```

4.3.2 Implementasi Praproses Data

Data masukan yang sudah diubah ke dalam format *list* kemudian melalui tahap praproses data. Dalam implementasinya, masing-masing kategori praproses dideklarasikan secara terpisah. Kombinasi masing-masing kategori digabungkan untuk dapat

mengolah data masukan. *Listing* program yang digunakan adalah sebagai berikut.

```
pat6 = r'[0-9]+' #menghapus karakter
angka dalam satu rangkaian lebih dari
satu
pattern = re.compile('\W') #menghapus
spasi berlebihan

combined_pat = r'|'.join((pat1, pat2,
pat3,pat4, pat5, pat6)) #pembuatan
kombinasi filter

list_data_bersih1 = []

for kata in listAgregat:
    dataBersih1 =
re.sub(combined_pat, '', str(kata)) #proses
filtering

    list_data_bersih1.append(dataBersih1)
#data bersih disimpan
list_data_bersih = []

for k in list_data_bersih1:
    dataBersih2 = re.sub(pattern, '
', str(k)) #menghapus karakter non
alphanumeric
    dataBersih3 = re.sub(r'\b\w{1,1}\b',
'', dataBersih2) #menghapus karakter 1
digit
    list_data_bersih.append(dataBersih3)
#menyimpan dalam list
```

4.3.3 Implementasi Penyusunan *Sparse Matrix*

Pada implementasi penyusunan *sparse matrix*, digunakan *library* sklearn dengan mengimpor terlebih dahulu daftar *stopwords* ke dalam program. Sebagaimana disebutkan pada perancangan proses, bahwa fitur yang muncul dalam satu dokumen dihilangkan, N-gram yang digunakan adalah satu hingga tiga, serta digunakan normalisasi pada bobot awal. *Listing* program penyusunan *sparse matrix* adalah sebagai berikut.

```
from sklearn.feature_extraction.text
import TfidfVectorizer

list_stopword=[]
    with
open('StopwordKataTugas.csv','r') as
stopword:
    for line in stopword:

list_stopword.append(line.strip())

vectorizer = TfidfVectorizer(stop_words =
list_stopword, #deklarasi stopwords
use_idf=True, #menginisiasi penggunaan
inverse document
ngram_range=(1,3), #deklarasi interval n-
gram
norm='l2', #menginisiasi penggunaan
normalisasi
smooth_idf=False, #menginiasi penggunaan
inverse document frequency
min_df=2 #menginisiasi minimal jumlah
kata dalam tweet)
```

```
X =
vectorizer.fit_transform(list_data_bersih
) #perhitungan TF-IDF
```

4.3.4 Implementasi Pemodelan Topik

Dalam implementasi pemodelan topik, pertama kali dilakukan adalah *truncated SVD*. Implementasi pemodelan topik pada penelitian Tugas Akhir ini adalah dengan menggabungkan proses LSA dan klasterisasi. Secara sederhana, pada bagian pengujian nilai *sum explained variance ratio*, *sparse matrix* akan di-*truncated* berdasarkan minimum, maksimum, dan langkah pengujian. Untuk memudahkan interpretasi, maka nilai *sum explained variance ratio* divisualisasikan dalam bentuk grafik. Adapun *listing* program untuk mendekomposisi *sparse matrix* dan *truncated SVD* adalah sebagai berikut.

```
lsa      =      TruncatedSVD(n_components=3,
algorithm='randomized')
lsa_matrix      =      lsa.fit_transform(X)
#melakukan pemotongan matriks
terms      =      vectorizer.get_feature_names()
#pengambilan kata bersesuaian
```

Jumlah k yang digunakan pada *truncated SVD* dapat diganti dengan mengubah nilai parameter `n_components` seperti kotak berwarna merah di atas. Adapun perhitungan nilai *sum explained variance ratio* dan visualisasi dalam bentuk grafik, diimplementasikan pada *listing* program berikut.

```
cek_range_bawah =
int(Entry_min_pca.get())
cek_range_atas = int(Entry_max_pca.get())
step = int(Entry_step_pca.get())
```

```

cobal = []
nomor = []

for i in
range(cek_range_bawah, (cek_range_atas+1),
step):
    lsa =
TruncatedSVD(n_components=i, algorithm='ra
ndomized')
    lsa.fit(vektor)
    var
=lsa.explained_variance_ratio_.sum()
    nomor.append(i)
    cobal.append(var)

figure1 = plt.Figure(figsize=(5,4),
    dpi=80)
ax = figure1.add_subplot(111)
ax.plot(nomor, cobal)
line = FigureCanvasTkAgg(figure1, root)
ax.set_xlabel('Number of PC')
ax.set_ylabel('Value')
ax.set_title('Truncated Sum Explained
    \nVariance Ratio')
ax.axhline(linewidth=2, color='r', y=0.3)
line.get_tk_widget().grid(row=4,
    rowspan=17, column=5, columnspan=11)
messagebox.showinfo("Message", "The plot
    of sum explained variance ratio done
    succesfully")

```

Matriks yang terbentuk setelah *truncated SVD* kemudian akan digunakan sebagai masukan dalam *K-Means cluster*. Adapun *listing* program yang digunakan untuk klasterisasi dengan *K-Means* adalah sebagai berikut.

```

from sklearn.cluster import KMeans
km_lsa = KMeans(n_clusters=50,
max_iter=5000,
random_state=100).fit(lsa_matrix_jadi) #
LSA-based k-means

```

Jumlah kluster yang dipilih dapat diganti dengan mengubah nilai parameter `n_clusters` seperti ditunjukkan pada kotak merah di atas. Adapun *listing* program untuk menghitung nilai *inertia* pada interval tertentu dan visualisasi pada grafik adalah sebagai berikut.

```

min_cluster = int(Entry_min_km.get())
max_cluster = int(Entry_max_km.get())
step_cluster = int(Entry_step_km.get())
jumlah_truncated =
int(Entry_max_trunc.get())

inertia = []
no = []
lsa =
TruncatedSVD(n_components=jumlah_truncated,
algorithm='randomized')
lsa_matrix = lsa.fit_transform(vektor1

for i in range (min_cluster,
max_cluster+1, step_cluster):
    model = KMeans(n_clusters=i,
max_iter=5000, random_state=100)
    model.fit(lsa_matrix)
    inertia.append(model.inertia_)
    no.append(i)

figure2 = plt.Figure(figsize=(5,4),
dpi=80)
ax1 = figure2.add_subplot(111)

```

```

ax1.plot(no,inertia, marker='o')
line = FigureCanvasTkAgg(figure2, root)
ax1.set_xlabel('Number of Cluster')
ax1.set_ylabel('Inertia')
ax1.set_title('Elbow Measurement')

```

Adapun tahap terakhir dalam pemodelan topik adalah perhitungan jumlah *tweet* dalam satu klaster dan penarikan kata kunci. *Listing* program yang digunakan adalah sebagai berikut.

```

centers =
lsa.inverse_transform(km_lsa.cluster_centers_).argsort()[:, :-1]

kata = []
for i in range(0,jumlah_cluster):
    l = []
    l.append(", ".join([terms[c] for c in
centers[i, :nilai_terms]]))
    kata.append(l)

```

4.3.5 Implementasi Analisis Sentimen

Implementasi analisis sentimen dimulai dengan mengimpor daftar *lexicon* berformat .txt ke dalam program dan disimpan dalam kamus *lexicon*. Adapun *listing*-nya adalah sebagai berikut.

```

def __init__(self, lexicon_database =
"lexicon1.txt",
emoji_database="lexicon2.txt"):
module_full =
os.path.abspath(getsourcefile(lambda:0))

```



```

lexicon_full =
os.path.join(os.path.dirname(module_full)
, lexicon_database)
with open(lexicon_full, encoding='utf-8')
as f:
    self.lexicon_full = f.read()
    self.lexicon = self.lexicon_dict()

emoji_full =
os.path.join(os.path.dirname(module_full)
, emoji_database)
with open(emoji_full, encoding='utf-8')
as f:
    self.emoji_full = f.read()
    self.emoji = self.emoji_dict()

def lexicon_dict(self):
    lexicon_kamus = { }
    for i in self.lexicon_full.split('\n'):
        (kata,          nilai) =
i.strip().split('\t')[0:2]
        lexicon_kamus[kata] = float(nilai)
    return lexicon_kamus

def emoji_dict(self):
    emoji_kamus = { }
    for i in self.emoji_full.split('\n'):
        (emoji,      kata) =
i.strip().split('\t')[0:2]
        emoji_kamus[emoji] = kata
    return emoji_kamus

```

Listing program untuk mendeklarasikan kata dan nilai *booster* serta negasi adalah sebagai berikut.

```

boster_pos = 0.293
boster_neg = -0.293

NEGATE =\
    ["aint", "arent", "cannot", "cant",
    "couldnt", "darent", "didnt", "doesnt",
    "ain't", "aren't", "can't", "couldn't",
    "daren't", "didn't", "doesn't", "dont",
    "hadnt", "hasnt", "havent", "isnt",
    "mightnt", "mustnt", "neither",
    "don't", "hadn't", "hasn't",
    "haven't", "isn't", "mightn't", "mustn't",
    "neednt", "needn't", "never", "none",
    "nope", "nor", "not", "nothing",
    "nowhere",
    "oughtnt", "shant", "shouldnt",
    "uhuh", "wasnt", "werent",
    "oughtn't", "shan't", "shouldn't",
    "uh-uh", "wasn't", "weren't",
    "without", "wont", "wouldnt",
    "won't", "wouldn't", "rarely", "seldom",
    "despite"] #jika kata negate lebih dari
    satu maka hanya dihitung satu kali

#mendefisikan kata-kata bososter dan
    nilainya
kata_boster =\
    {"absolutely": boster_pos,
    "amazingly": boster_pos, "awfully":
    boster_pos, "completely": boster_pos,
    "considerably": boster_pos,
    "decidedly": boster_pos, "deeply":
    boster_pos, "effing": boster_pos,
    "enormously": boster_pos,

```

```

    "entirely": boster_pos, "especially":
boster_pos, "exceptionally": boster_pos,
"extremely": boster_pos,
    "fabulously": boster_pos, "flipping":
boster_pos, "flippin": boster_pos,
    "fricking": boster_pos, "frickin":
boster_pos, "frigging": boster_pos,
"friggin": boster_pos, "fully":
boster_pos, "fucking": boster_pos,
    "greatly": boster_pos, "hella":
boster_pos, "highly": boster_pos,
"hugely": boster_pos, "incredibly":
boster_pos,
    "intensely": boster_pos, "majorly":
boster_pos, "more": boster_pos, "most":
boster_pos, "particularly": boster_pos,
    "purely": boster_pos, "quite":
boster_pos, "really": boster_pos,
"remarkably": boster_pos,
    "so": boster_pos, "substantially":
boster_pos,
    "thoroughly": boster_pos, "totally":
boster_pos, "tremendously": boster_pos,
    "uber": boster_pos, "unbelievably":
boster_pos, "unusually": boster_pos,
"utterly": boster_pos,
    "very": boster_pos,
    "almost": boster_neg, "barely":
boster_neg, "hardly": boster_neg, "just
enough": boster_neg,
    "kind of": boster_neg, "kinda":
boster_neg, "kindof": boster_neg, "kind-
of": boster_neg,
    "less": boster_neg, "little":
boster neg, "marginally": boster neg,

```

```

"occasionally": boster_neg, "partly":
boster_neg,
    "scarcely": boster_neg, "slightly":
boster_neg, "somewhat": boster_neg,
    "sort of": boster_neg, "sorta":
boster_neg, "sortof": boster_neg, "sort-
of": boster_neg}

```

Pada proses selanjutnya adalah pengubahan kata yang memiliki tanda baca menjadi hanya berisi kata. *Listing* program yang digunakan adalah sebagai berikut.

```

def __init__(self, kalimat):
    if not isinstance(kalimat, str):
        kalimat = str(kalimat).encode('utf-8')
    self.kalimat = kalimat
    self.kata_emoji =
self.kata_emoji_fungsi()
    self.apa_kapital =
cek_kapital(self.kata_emoji)

def cek_tanda(self):
    tanda = hapus_tanda.sub(' ',
self.kalimat)
    kata = tanda.split()
    sebelum = {''.join(p): p[1] for p in
product(daftar_tanda, kata) }
    sesudah = {''.join(p): p[0] for p in
product(kata, daftar_tanda) }
    transform = sebelum
    transform.update(sesudah)
    return transform

```

Adapun *listing* program kelas utama untuk mengembalikan nilai polaritas adalah sebagai berikut.

```
def nilai(self, kalimat):
    kata_token_list = kalimat.split()
    #split kalimat jadi token
    kata_token = []

    for i in kata_token_list:
        if i in self.emoji:
            hasil_ubah =
self.emoji[i]

kata_token.append(hasil_ubah)
        else:
            kata_token.append(i)
    kalimat = " ".join(kata for kata
in kata_token)

    sentikalimat =
SentiKalimat(kalimat)

    sentimen = []
    kata_emoji =
sentikalimat.kata_emoji

    #memberikan nilai valensi kata
    for i in kata_emoji:
        valensi = 0
        indeks = kata_emoji.index(i)

        if i.lower() in kata_boster:
            sentimen.append(valensi)
            continue
        if (indeks < len(kata_emoji)-
1 and i.lower() == "kind" and
```

```

kata_emoji[indeks+1].lower() == "of"):
    sentimen.append(valensi)
    continue

    sentimen =
self.hitung_valensi(valensi,
sentikalimat, i, indeks, sentimen) #sudah

    sentimen =
self.konjungsi(kata_emoji, sentimen)
#sudah

    valensi_akhir =
self.nilai_valensi(sentimen, kalimat)
#sudah

    return valensi_akhir

```

Adapun *listing* dari method-method yang digunakan adalah sebagai berikut.

a. Method `hitung_valensi`

Method `hitung_valensi` digunakan untuk memeriksa keberadaan kata dalam *lexicon*, memeriksa apakah merupakan kalimat negatif, dan mengandung kata “least”.

```

def hitung_valensi(self, valensi,
sentikalimat, i, indeks, sentimen):
    apa_kapital =
sentikalimat.apa_kapital

    kata_emoji =
sentikalimat.kata_emoji

    kata = i.lower()

    if kata in self.lexicon:

```

```

        #mendapatkan nilai valensi
standar lexicon
        valensi = self.lexicon[kata]

        if i.upper() and apa_kapital:
            if valensi > 0:
                valensi += caps_pos
            else:
                valensi -= caps_pos

        for a in range(0,3):
            if indeks > a and
kata_emoji[indeks-(a+1)].lower() not in
self.lexicon:
                s =
skalar_fungsi(kata_emoji[indeks-(a+1)],
valensi, apa_kapital)
                if a == 1 and s != 0:
                    s = s*0.95
                if a == 2 and s != 0:
                    s = s*0.9
                valensi = valensi + s
                valensi =
self.apa_negatif(valensi, kata_emoji, a,
indeks) #sudah

                if a == 2:

```

```

                valensi =
self.apa_idiom_special(valensi,
kata_emoji, indeks) #sudah

                valensi =
self.apa_least(valensi, kata_emoji,
indeks) #sudah
                sentimen.append(valensi)
                return sentimen

```

b. Method skalar

Method skalar digunakan untuk mentransformasi nilai polaritas kata *booster* bila diikuti suatu kata dengan nilai polaritas yang bertolak belakang serta memeriksa keberadaan suatu bentuk unik dari kata berhuruf kapital.

```

def skalar_fungsi(kata, valensi,
apa_kapital):

    skalar = 0.0
    kata_trans = kata.lower()

    if kata_trans in kata_boster:
        skalar = kata_boster[kata_trans]
        if valensi <0:

            skalar *= -1

        if kata.isupper() and
apa_kapital:
            if valensi >0:
                skalar += caps_pos
            else:

```



```

                skalar -= caps_pos
    return skalar

```

c. Method `apa_negatif`

Method `apa_negatif` digunakan untuk memeriksa keberadaan kata bermakna negatif dalam kalimat.

```

def apa_negatif(valensi, kata_emoji, a,
indeks):
    kata_emoji_trans =
[str(kata).lower() for kata in
kata_emoji]
    #kata_emoji_trans =
str(kata_emoji).lower()
    if a == 0:
        if
cek_negatif([kata_emoji_trans[indeks-
(a+1)]]):
            valensi = valensi *
caps_neg
        if a == 1:
            if kata_emoji_trans[indeks-
2]=="never" and \
(kata_emoji_trans[indeks-
1] == "so" or
kata_emoji_trans[indeks-
1]=="this"):
                valensi = valensi*1.25
            elif kata_emoji_trans[indeks-
2]=="without" and \

```

```

        kata_emoji_trans[indeks-
1] == "doubt":
            valensi = valensi
                elif
cek_negatif([kata_emoji_trans[indeks-
(a+1)]]):\
                    valensi =
valensi*caps_neg

            if a == 2:
                if kata_emoji_trans[indeks-
3]=="never" and \

(kata_emoji_trans[indeks-2] == "so" or
kata_emoji_trans[indeks-2]=="this") or \

(kata_emoji_trans[indeks-1] == "so" or
kata_emoji_trans[indeks-1]=="this"):
                    valensi = valensi*1.25

                elif kata_emoji_trans[indeks-
3]=="without" and \

(kata_emoji_trans[indeks-2] == "doubt"
or kata_emoji_trans[indeks-1] ==
"doubt"):
                    valensi = valensi
                        elif
cek_negatif([kata_emoji_trans[indeks-
(a+1)]]):\
                            valensi =
valensi*caps_neg
                                return valensi

```

d. Method `apa_least`

Method `apa_least` digunakan untuk memeriksa keberadaan kata "*least*", "*at least*", dan transformasi nilai polaritasnya.

```
def apa_least(self, valensi, kata_emoji,
indeks):
    if indeks > 1 and
kata_emoji[indeks-1].lower() not in
self.lexicon \
        and
kata_emoji[indeks-1].lower() == "least":
        if kata_emoji[indeks-
2].lower() != "at" and kata_emoji[indeks-
2].lower() != "very":
            valensi =
valensi*caps_neg
        elif indeks > 0 and
kata_emoji[indeks-1].lower() not in
self.lexicon \
            and kata_emoji[indeks-
1].lower() == "least":
                valensi = valensi*caps_neg
    return valensi
```

e. Method `cek_negatif`

Method `cek_negatif` digunakan untuk memeriksa keberadaan kata negasi dalam kalimat dan transformasi nilai polaritas.

```
def cek_negatif(kata_masukan,
status=True):
    #diubah ke huruf kecil
    kata_masukan = [str(i).lower() for i
in kata_masukan]
```

```

kata_neg = []
kata_neg.extend(NEGATE) #menambahkan
dictionary kata negatif

for kata in kata_neg:
    if kata in kata_masukan:
        return True

if status:
    for kata in kata_masukan:
        if "n't" in kata:
            return True
    if "least" in kata_masukan:
        z = kata_masukan.index("least")
#mencari index kata least
        if z >0 and kata_masukan [z-1] !=
"at":
            return True

return False

```

f. Method konjungsi

Method konjungsi digunakan untuk memeriksa keberadaan kata “*but*” dalam kalimat dan melakukan transformasi nilai polaritas kata sebelum dan sesudah kata “*but*”.

```

def konjungsi(kata_emoji, sentimen):
    kata_emoji_trans =
[str(kata).lower() for kata in
kata_emoji]
    #kata_emoji_trans =
str(kata_emoji).lower()

```

```

        if 'but' in kata_emoji_trans:
            indeks_but =
kata_emoji_trans.index('but') #mengambil
indeks but

            for sent in sentimen:
                indeks_a =
sentimen.index(sent)
                if indeks_a < indeks_but:

sentimen.pop(indeks_a)

sentimen.insert(indeks_a, sent*0.5)
                elif indeks_a >
indeks_but:

sentimen.pop(indeks_a)

sentimen.insert(indeks_a, sent*1.5)
            return sentimen

```

g. Method nilai_valensi

Method nilai_valensi digunakan untuk menjumlahkan nilai polaritas pada proses sebelumnya (disimpan dalam *list* sentimen), menambahkan nilai polaritas tanda baca, serta jembatan untuk melakukan normalisasi.

```

def nilai_valensi(self, sentimen,
kalimat):
    if sentimen:
        total_awal =
float(sum(sentimen))

```

```

        tambah_tanda =
self.tambah_tanda_fungsi(kalimat) #belum

        if total_awal > 0:
            total_awal +=
tambah_tanda
        elif total_awal < 0:
            total_awal -=
tambah_tanda

        total =
normalisasi(total_awal)
        else:
            total = 0
        return total

```

h. Method tambah_tanda_fungsi

Method tambah_tanda_fungsi digunakan sebagai jembatan dalam memeriksa keberadaan tanda baca seru dan tanda baca tanya dalam kalimat.

```

def tambah_tanda_fungsi(self,
kalimat):
    tambah_tanda1 =
self.tanda1_fungsi(kalimat)
    tambah_tanda2 =
self.tanda2_fungsi(kalimat)

    hasil =
tambah_tanda1+tambah_tanda2
    return hasil

```

i. Method `tanda1_fungsi`

Method `tanda1_fungsi` digunakan untuk menghitung jumlah tanda seru (!) dalam kalimat dan menghitung nilai polaritasnya.

```
def tanda1_fungsi(kalimat):  
    jumlah = kalimat.count("!")  
    if jumlah > 4:  
        jumlah = 4  
    hasil = jumlah*0.292  
    return hasil
```

j. Method `tanda2_fungsi`

Method `tanda1_fungsi` digunakan untuk menghitung jumlah tanda tanya (?) dalam kalimat dan menghitung nilai polaritasnya.

```
def tanda2_fungsi(kalimat):  
    jumlah = kalimat.count("?")  
    hasil = 0  
  
    if jumlah > 1:  
        if jumlah <= 3:  
            hasil = jumlah*0.18  
        else:  
            hasil = 0.96  
    return hasil
```

k. Method `normalisasi`

Method `normalisasi` digunakan untuk melakukan normalisasi pada nilai polaritas awal yang dilakukan pada proses-proses sebelumnya.

```

def normalisasi(nilai, alpha=20):
    nilai_norm =
nilai/math.sqrt((nilai*nilai)+20)

    if nilai_norm < -1.0: #jika nilai
normalisasi kurang dari -1
        return -1.0
    elif nilai_norm > 1.0: #jika nilai
normalisasi lebih dari 1
        return 1.0
    else:
        #nilai_norm = nilai
        return nilai_norm

```

4.3.6 Implementasi *Opportunity Mining*

Adapun *listing* program untuk mengimplementasikan *opportunity mining* meliputi perhitungan *importance*, *satisfaction*, *opportunity*, hingga visualisasi dalam *product landscape map* adalah sebagai berikut.

```

polaritas = nilai_kata()
nilai_trunc = int(Entry_trunc_p.get())
nilai_km = int(Entry_km_p.get())
nilai_terms = int(Entry_terms_p.get())
Epoch = int(Entry_epoch_p.get())
if ((nilai_trunc <1) or (nilai_trunc >
vektor2.shape[1]) or (nilai_km < 1) or
(nilai_km > vektor2.shape[0]) or (Epoch <
100) or (nilai_terms < 1) or (nilai_terms
> 10)):

```



```

        messagebox.showerror('Error', 'Please
make sure that input the correct value')
else:
    lsa =
TruncatedSVD(n_components=nilai_trunc,
algorithm='randomized')
    lsa_matrix =
lsa.fit_transform(vektor2) #melakukan
pemotongan matriks
    km_lsa = KMeans(n_clusters=nilai_km,
max_iter=Epoch,
random_state=100).fit(lsa_matrix) # LSA-
based k-means

    clust = km_lsa.labels_.tolist()
    idea={'Opinion':listAgregat,
'Cluster':clust} #Creating dict having
doc with the corresponding cluster
number.

frame_clust=pd.DataFrame(idea,index=[clus
t], columns=['Opinion','Cluster']) #
Converting it into a dataframe.

    hasil_sent = []
    for word in listAgregat:

hasil_sent.append(polaritas.nilai(word))
    hasil_sent1 =
list(zip(listAgregat,hasil_sent))

```

```

    tabel_sentimen =
pd.DataFrame(hasil_sent1,
columns=['Opinion','Polarity'])

    importance_awal = []
    satisfied_awal = []

    jumlah_cluster = nilai_km
    for i in range(jumlah_cluster):
        jumlah_importance = 0
        jumlah_satisfied = 0
        count_satisfied = 0
        for j in range(len(frame_clust)):
            if (frame_clust.iloc[j][1] ==
i):
                jumlah_importance =
jumlah_importance+1
                jumlah_satisfied =
jumlah_satisfied+polaritas.nilai(frame_cl
ust.iloc[j][0])
                count_satisfied =
count_satisfied+1

    importance_awal.append(jumlah_importance)

    satisfied_awal.append(jumlah_satisfied/co
unt_satisfied)

```

```

centers =
lsa.inverse_transform(km_lsa.cluster_centers_).argsort()[:, :-1]

kata = []
for i in range(0, jumlah_cluster):
    l = []
    l.append(", ".join([terms[c] for c in centers[i, :nilai_terms]]))
    kata.append(l)

importance = [(i - min(importance_awal)) / (max(importance_awal) - min(importance_awal)) * 10 for i in importance_awal]
satisfied = [(i - min(satisfied_awal)) / (max(satisfied_awal) - min(satisfied_awal)) * 10 for i in satisfied_awal]
opportunity = [(importance[i] + max((importance[i] - satisfied[i]), 0)) for i in range(0, jumlah_cluster)]
tabel =
list(zip(importance, satisfied, opportunity, kata))
tabel_frame = pd.DataFrame(tabel, columns=["Importance", "Satisfied", "Opportunity", "Key"]).sort_values("Importance", ascending=False)

```

```
a =
[sum(importance)/float(len(importance)),1
0]

b = [0,10]

a1 = [0,10]
b1 =
[sum(satisfied)/float(len(satisfied)),10]

z1 =
tabel_frame['Importance'].tolist()
z2 =
tabel_frame['Satisfied'].tolist()
z3 =
tabel_frame['Opportunity'].tolist()
z4 = list(tabel_frame.index+1)

window = Toplevel(root)
window.geometry('500x500')
figure3 = plt.Figure(figsize=(5,4),
dpi=100)
ax3 = figure3.add_subplot(111)
ax3.plot(a,b)
ax3.plot(a1,b1)
ax3.scatter(z1,z2, color='g')

for i, txt in enumerate(z4):
    ax3.annotate(txt, (z1[i],z2[i]))
```

```

        scatter3 = FigureCanvasTkAgg (figure3,
window)

scatter3.get_tk_widget().pack(side=tk.LEF
T, fill=tk.BOTH)
        ax3.legend()
        ax3.set_xlabel('Importance')
        ax3.set_ylabel('Satisfied')
        ax3.set_title('Product Landscape
Map')

```

4.3.7 Implementasi Tombol Simpan

Adapun *listing* program untuk menyimpan *sparse matrix*, fitur-fitur, data masukan dalam *list*, serta kata kunci dalam pemodelan topik adalah sebagai berikut.

```

def tombolsimpan():
    nama_matrix = str(entry_matrix.get())
    nama_kata = str(entry_key.get())
    nama_term = str(entry_fitur.get())
    nama_list = str(entry_list.get())

    nama_file_terms1 = nama_term + ".txt"
    nama_file_matrix1 = nama_matrix +
".mtx"
    nama_file_list1 = nama_list + ".txt"
    nama_file_katal = nama_kata + ".txt"

```

```

with open(nama_file_terms1, "wb") as
fp:    #Pickling
        pickle.dump(terms, fp)

        io.mmwrite(nama_file_matrix1,
lsa_matrix)

with open(nama_file_kata1, "wb") as
fp:    #Pickling
        pickle.dump(kata, fp)

with open(nama_file_list1, "wb") as
fp:    #Pickling
        pickle.dump(listAgregat, fp)

        messagebox.showinfo("Message", "Model
and data stored succesfully\n Caution :
Please make sure that you enter the
correct and the same cluster number as
predicted before to match the key")

```

4.3.8 Implementasi Tombol *Load* Model

Implementasi tombol *load* model untuk membuka kembali hasil proses adalah sebagai berikut.

```

def tombolmodel():
    global lsa_matrix_fix1
    global terms1

```

```

global kata1
global listAgregat1
global status

open_matrix = str(Entry_matrix.get())
open_matrix1 = open_matrix + ".mtx"
open_terms = str(Entry_terms.get())
open_terms1 = open_terms + ".txt"
open_kata = str(Entry_kata.get())
open_kata1 = open_kata + ".txt"
open_list = str(Entry_list.get())
open_list1 = open_list + ".txt"

lsa_matrix_fix =
io.mmread(open_matrix1)
lsa_matrix_fix1 =
np.asarray(lsa_matrix_fix.tolist())

with open(open_terms1, "rb") as fp:
# Unpickling
    terms1 = pickle.load(fp)

with open(open_kata1, "rb") as fp:
# Unpickling
    kata1 = pickle.load(fp)

with open(open_list1, "rb") as fp:
# Unpickling
    listAgregat1 = pickle.load(fp)

```

4.3.9 Implementasi Menu *Details*

Adapun implementasi menu *details* untuk menampilkan detail data hasil proses adalah pada *listing* di bawah ini.

```
def tomboldetails():

messagebox.showinfo("Message","Details
windows opened succesfully")

    window = Toplevel(root)
    window.geometry('500x300')
    window.configure(bg='powder blue')
    window.title('Details window')

    labeldetail = Label(window,
font=('calibri', 13, 'bold'),
text='Details Categories', fg='powder
blue',bg="powder blue", bd=10)
    labeldetail.grid(row=0, column=0)

    labeldetail1 = Label(window,
font=('calibri', 13, 'bold'),
text='Details Categories',
fg='black',bg="powder blue", bd=10)
    labeldetail1.grid(row=0, column=1)

    labeldetail2 = Label(window,
font=('calibri', 11, 'bold'), text='In
this windows, the detail process can \nbe
seen detailer', fg='black',bg="powder
blue", bd=10)
    labeldetail2.grid(row=1, column=0,
columnspan=3)
```



```
tombol_database = Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'), text='Open
Database', bg='tomato',

command=tombol_detail_database)
tombol_database.grid(row=2,column=0)

tombol_explained = Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'), text='Sum
Var', bg='tomato',

command=tombol_detail_explained)
tombol_explained.grid(row=2,column=1)

tombol_elbow = Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'), text='Show
Elbow', bg='tomato',

command=tombol_detail_elbow)
tombol_elbow.grid(row=2,column=2)

labeldetail3 = Label(window,
font=('calibri', 13, 'bold'),
text='Details Categories', fg='powder
blue',bg="powder blue", bd=10)
labeldetail3.grid(row=2, column=3)
```

```
tombol_clust= Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'), text='Cluster
Map', bg='tomato',
        command=tombol_detail_clust)
tombol_clust.grid(row=3,column=0)

tombol_sent = Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'),
text='Sentiment Map', bg='tomato',
        command=tombol_detail_sent)
tombol_sent.grid(row=3,column=1)

tombol_frame = Button(window,
width=15, height=1, fg="black",
font=('arial', 8, 'bold'), text='Fixed
Calc', bg='tomato',
        command=tombol_detail_fixed)
tombol_frame.grid(row=3,column=2)

labeldetail4 = Label(window,
font=('calibri', 13, 'bold'),
text='Details Categories', fg='powder
blue',bg="powder blue", bd=10)
labeldetail4.grid(row=3, column=3)

tombol_senti = Button(window,
width=15, height=1, fg="black",
```

```

font=('arial', 8, 'bold'), text='Sent.
App', bg='SpringGreen',
        command=tombol_sentimen1)
    tombol_senti.grid(row=4,column=1)

def tombol_detail_database():
    database = Toplevel(root)
    database.geometry('700x400')
    database.configure(bg='white')
    database.title('Details Tweet
Database')
    if (status == 1):
        list1 = []
        for kata in listAgregat1:
            list1.append(kata)
        txt = Text(database)
        txt.grid(row=0, column=0,
sticky="eswn")
        scroll_y = Scrollbar(database,
orient="vertical", command=txt.yview)
        scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)

        very_long_list =
"\n".join([str(i) for i in list1])
        txt.insert("1.0", very_long_list)

```

```

        txt.configure(state="disabled",
relief="flat", bg=database.cget("bg"))
    else:
        txt = Text(database)
        txt.grid(row=0, column=0,
sticky="eswn")
        scroll_y = Scrollbar(database,
orient="vertical", command=txt.yview)
        scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)

        very_long_list =
"\n".join([str(i) for i in listAgregat])
        txt.insert("1.0", very_long_list)
        txt.configure(state="disabled",
relief="flat", bg=database.cget("bg"))

def tombol_detail_explained():
    if (status == 1):

messagebox.showinfo("Message", "Explained
variance ratio can't be loaded on model
loading")
    else:
        databasel = Toplevel(root)
        databasel.geometry('600x200')
        databasel.configure(bg='white')

```

```

        database1.title('Details Sum
Explained Variance Ratio Score')
        txt = Text(database1)
        txt.grid(row=0, column=0,
sticky="eswn")
        scroll_y = Scrollbar(database1,
orient="vertical", command=txt.yview)
        scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)

        very_long_list =
"\n".join([str(i) for i in cobal])
        txt.insert("1.0", very_long_list)
        txt.configure(state="disabled",
relief="flat", bg=database1.cget("bg"))

def tombol_detail_elbow():
    database2 = Toplevel(root)
    database2.geometry('400x600')
    database2.configure(bg='white')
    database2.title('Details Elbow
Score')
    txt = Text(database2)
    txt.grid(row=0, column=0,
sticky="eswn")
    scroll_y = Scrollbar(database2,
orient="vertical", command=txt.yview)

```

```
        scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)

        very_long_list = "\n".join([str(i)
for i in inertia])
        txt.insert("1.0", very_long_list)
        txt.configure(state="disabled",
relief="flat", bg=database2.cget("bg"))

def tombol_detail_clust():
    database3 = Toplevel(root)
    database3.geometry('600x600')
    database3.configure(bg='white')
    database3.title('Cluster Map')

    tes1 =
frame_clust['Opinion'].tolist()
    tes2 =
frame_clust['Cluster'].tolist()
    tes3 = list(zip(tes1, tes2))

    txt = Text(database3)
    txt.grid(row=0, column=0,
sticky="eswn")
    scroll_y = Scrollbar(database3,
orient="vertical", command=txt.yview)
```

```

        scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)
        very_long_list = "\n".join([str(i)
for i in tes3])
        txt.insert("1.0", very_long_list)
            txt.configure(state="disabled",
relief="flat", bg=database3.cget("bg"))

def tombol_detail_fixed():
    database4 = Toplevel(root)
    database4.geometry('700x600')
    database4.configure(bg='powder blue')
    database4.title('Opportunity Mining
Table')
    frame = Text(database4)
    frame.pack()

    class printdata(object):
        def write(self, s):
            frame.insert(END, s)

    sys.stdout = printdata()
    print(tabel_frame)

def tombol_detail_sent():
    database5 = Toplevel(root)

```

```

database5.geometry('600x600')
database5.configure(bg='white')
database5.title('Opportunity Mining
Table')
tes1 =
tabel_sentimen['Opinion'].tolist()
tes2 =
tabel_sentimen['Polarity'].tolist()
tes3 = list(zip(tes1, tes2))
txt = Text(database5)
txt.grid(row=0, column=0,
sticky="eswn")
scroll_y = Scrollbar(database5,
orient="vertical", command=txt.yview)
scroll_y.grid(row=0, column=1,
sticky="ns")

txt.configure(yscrollcommand=scroll_y.set
)

very_long_list = "\n".join([str(i)
for i in tes3])
txt.insert("1.0", very_long_list)
txt.configure(state="disabled",
relief="flat", bg=database5.cget("bg"))

def tombol_check_senti():
katasenti = str(entry_sentimen.get())
polaritas = nilai_kata()
nilai1 = polaritas.nilai(katasenti)

```



```

input1.set(nilai1)
if (nilai1 > 0):
    input2.set('Positive')
elif (nilai1 < 0):
    input2.set('Negative')
else:
    input2.set('Neutral')

```

4.3.10 Implementasi Desain Antarmuka

Pada proses berjalannya sistem, ditampilkan desain antarmuka perangkat lunak. Adapun desain antarmuka yang dijalankan adalah sebagai berikut.

Final Project-Riko Wijayanto

Implementasi Social Media Mining untuk Pengambilan Keputusan dalam Perencanaan Produk berdasarkan Pemodelan Topik dan Analisis Sentimen pada Twitter
Implementation of Social Media Mining for Decision Making in Product Planning based on Topic Modelling and Sentiment Analysis on Twitter
by : Riko Wijayanto NRP. 0611154000010
Supervisor : Prof. Dr. Mohammad Isa Irawan, M.T. & Muhammad Luthfi Shahab, S.Si., M.Si.

The interface is organized into several functional areas:

- File Loader:** A text input field and a red "Load" button.
- Model Loader:** A text input field, a "Matrix" label, and a red "Load Model" button.
- Truncated SVD Parameter:** Three text input fields, a "Min Truncated" label, a "Max Truncated" label, a "Step Truncated" label, and a red "Truncated" button.
- Store Fixed Model:** Four text input fields labeled "Matrix Name", "Features Name", "Keyword Name", and "List Name", and two buttons: a green "Details" button and a green "Clear" button.
- KMeans Parameter:** Two text input fields, a "Min Cluster" label, a "Max Cluster" label, a "Step Cluster" label, a "Numb. Truncated" label, and a red "Cluster" button.
- Product Planning:** Three text input fields, a "Number of PC" label, a "Number of Cluster" label, a "Number of Epoch" label, a "Number of Key" label, and a red "Predict" button.

Gambar 4.6 Tampilan Desain Antarmuka Halaman Utama



Gambar 4.7 Tampilan Desain Antarmuka Halaman *Details*

BAB V

UJI COBA DAN PEMBAHASAN

Pada bab ini diuraikan uji coba program pada data uji coba, analisis statistik kuantitatif, serta pembahasan terkait hasil uji coba pada data *tweet*.

5.1 Deskripsi Data Uji Coba

Sebagaimana disampaikan pada bab sebelumnya, bahwa data yang digunakan untuk uji coba adalah data *tweet* dengan *tweet* yang ditujukan kepada akun @McDonalds. Jumlah *tweet* yang digunakan adalah sebanyak 2000 *tweet*. Kutipan data yang diambil dalam bentuk .csv ditunjukkan oleh Gambar 5.1 berikut.

text	favorited	favoriteCount	created_at	truncated_replyToSId	replyToUId	statusSource	screenName	retweeted	retweetCount	retweetUser	longitude	latitude
@McDonalds I realize McDonald's is meant	FALSE	1	#####	TRUE	1.10E+18	1.10E+18	2.68E+09 WMonnat	0	FALSE	FALSE	NA	NA
Mmmmm ipads in @McDonalds's... Soooo	FALSE	1	#####	FALSE	1.10E+18	1.10E+18	71026122 initsua	0	FALSE	FALSE	NA	NA
@McDonalds Only 2 reviews for a McDonald	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	71026122 OwOWha	0	FALSE	FALSE	NA	NA
@McDonalds I was in the Logan Sq. area an	FALSE	0	#####	FALSE	NA	1.10E+18	71026122 rtccatkrk	0	FALSE	FALSE	NA	NA
@McDonalds Something is wrong with this	FALSE	0	#####	TRUE	1.10E+18	1.10E+18	44945327 happicous	0	FALSE	FALSE	NA	NA
@McDonalds Typical McDonalds food not r	FALSE	3	#####	FALSE	NA	1.10E+18	NA awebreea	0	FALSE	FALSE	NA	NA
@McDonalds I was on my way home a littl	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	9.75E+17 Jat3153	0	FALSE	FALSE	NA	NA
One of the early EPOS (Electronic Point-of	FALSE	1	#####	FALSE	1.10E+18	1.10E+18	8.95E+17 Jutsiin	0	FALSE	FALSE	NA	NA
@McDonalds This McDonald's has made a r	FALSE	1	#####	TRUE	1.10E+18	1.10E+18	2.68E+09 WMonnat	0	FALSE	FALSE	NA	NA
@McDonalds Y'all want \$15 an hour but I'm	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	9.75E+17 Jat3153	0	FALSE	FALSE	NA	NA
@McDonalds They've remodeled, the plac	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	1.5E+08 geonadj	0	FALSE	FALSE	NA	NA
I'm at @McDonalds in Palatka, FL https://t	FALSE	0	#####	TRUE	1.10E+18	1.10E+18	42697716 A3095Roa	0	FALSE	FALSE	NA	NA
@McDonalds My mom and I went here for	FALSE	2	#####	FALSE	1.10E+18	1.10E+18	4.55E+08 DandyAnc	0	FALSE	FALSE	NA	NA
@McDonalds I don't understand why an in	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	19402541 Eric_Bridg	0	FALSE	FALSE	NA	NA
@McDonalds The strictest mcDonalds I've	FALSE	0	#####	FALSE	NA	1.10E+18	NA amanee151	0	FALSE	FALSE	-112.358	33.46514
@McDonalds Went through the drive-thro	FALSE	0	#####	FALSE	NA	1.10E+18	71026122 amaneejav	0	FALSE	FALSE	NA	NA
@McDonalds Went shamrock shake thro	FALSE	1	#####	TRUE	NA	1.10E+18	NA SCFJobs	0	FALSE	FALSE	NA	NA
RECORDING TONIGHT! Last chance to buy a	FALSE	1	#####	FALSE	NA	1.10E+18	NA Mitchalle	0	FALSE	FALSE	NA	NA
@McDonalds I stopped at this McDonald's	FALSE	1	#####	FALSE	1.10E+18	1.10E+18	71026122 madonna	0	FALSE	FALSE	NA	NA
@McDonalds I stopped here to use the Rec	FALSE	1	#####	TRUE	1.10E+18	1.10E+18	4.89E+09 CarneHan	0	FALSE	FALSE	NA	NA
@McDonalds Sorr, am I bothering you? V	FALSE	0	#####	FALSE	1.10E+18	1.10E+18	3.75E+08 JoseGarza	0	FALSE	FALSE	NA	NA

Gambar 5.1 Gambar Kutipan Data *Tweet*

Detail jumlah data *tweet* tersebut berdasarkan urutan waktu adalah ditunjukkan pada Tabel 5.1.

Tabel 5.1 Jumlah Data *Tweet* per-Tanggal

Tanggal	Jumlah <i>Tweet</i>
4 Maret 2019	440
5 Maret 2019	520
6 Maret 2019	520
7 Maret 2019	520

Contoh data *tweet* digunakan sebagai uji adalah seperti di bawah ini.

@McDonalds Very horrible Micky Ds! Very dirty and just a horrible place to take the kids too!
The “super size” idea has barged its way into our homes (and has influenced portion sizes). Well done @McDonalds fo... https://t.co/86dPspB38V
@McDonalds Wow, this place was so good after the night was over!
@McDonalds Terrible service, plain and simple!

5.2 Proses Uji Coba

Untuk menampilkan detail dari berjalannya sistem yang sudah diimplementasikan pada data uji coba, maka ditampilkan hasil uji coba pada masing-masing proses.

5.2.1 Uji Coba Impor Data

Uji coba impor data *tweet* dengan *library* *pandas* dalam bentuk *dataframe* adalah sebagai berikut.

	text
0	@McDonalds I realize McDonald's is meant to be...
1	Mmmmm ipads in @McDonalds ... Soooo hygienic...
2	@McDonalds Only 2 reviews for a McDonalds, hmm...
3	@McDonalds I was in the Logan Sq. area and sto...
4	@McDonalds Something is wrong with this Mc Don...
5	@McDonalds Typical McDonalds food not much nut...
6	@McDonalds I was on my way home a little after...
7	One of the early EPOS (Electronic-Point-of-Sal...
8	@McDonalds This McDonald's has made a radical ...
9	@McDonalds Y'all want \$15 an hour but I'm wait...
10	@McDonalds They've remodeled, the place is gre...
11	I'm at @McDonalds in Palatka, FL https://t.co/...
12	@McDonalds My mom and I went here for sodas ar...
13	@McDonalds I don't understand why an intellige...

Gambar 5.2 Dataframe dari Data Tweet

Hasil manipulasi *dataframe* ke dalam bentuk *list* pada data uji coba adalah sebagai berikut.

```
@McDonalds ew. there are plenty of more interesting places to eat.
"@McDonalds This is the worst McDonald's I've ever had. I've gone twice since moving to the area and the fries were always soggy (the first time they were borderline uncooked) and the burgers were always poorly made and greasy. Please avoid this McDonalds. It blows.",
'First the @McDonalds on homer Watson doesn't take the cheese off my burger when it says remove cheese, then they give me a free drink. https://t.co/s0w8n5EDLa',
"@McDonalds Does this look cooked to you?!" <u+0001F633> #quarterpounder <u+0001F922> https://t.co/gUTjBZ0Vij',
"@McDonalds The Mickey D's breakfast vibe was strong in the work space and several of us were looking for some extra calories to fuel the early morning mental dash. Given there was four of us with one getting a meal for another, we opted to go inside. Chaos on only describe the inside. The wait, the ordering process, the retrieval of food seemed very disorganized and confusing. When my order did pop up I did a quick cursory inspection of the two bags and was out the door. We orbited the drive-thru (which is long) and noticed it was full of cars. I was initially thankful we went indoors to make our orders. It was only when we arrived back at the office did I notice that I was not f&#xed at drive-thru but f&#xed in the interior. My order was wrong and.....I am ashamed to admit...I did not notice they did not give me my coffee (no exclamation mark needed). And so began the internal debate with myself, just whose responsibility was it to make sure I got my coffee? I then decided it was both mine and theirs,,but mostly theirs. When I am starving in the morning from here on out, I will just wait for lunch.",
"@McDonalds on occasion I may exaggerate. But this place gets my order wrong literally every single time.. the sad thing is that I dont even like their food, it's just the ONLY place open when I get off, or am going to work. The only saving grace is the drive thru team. They are constantly sweet and apologetic for their kitchen staffs ineptitudes.",
"@McDonalds worst mcdonalds store. All employees are not happy.. not enthusiastic to work. The cashier was not even smiling.."
```

Gambar 5.3 List dari Data Tweet

5.2.2 Uji Coba Praproses Data

Hasil uji coba praproses data sebagaimana proses berjalannya dijelaskan pada bab sebelumnya dan disimpan dalam *list* baru adalah sebagai berikut.

```
[I realize McDonald is meant to be fast food but to call this one particular location fast food is an oxymoron unless waiting minutes to get your items and half the time they are incorrect. I can pass as fast food now. I don't know why my friend keeps insisting on stopping here for drinks. It'd be faster to stop at Walgreens, get out of the car, get a bottle of coke, pay and get back in the car to drive off. I believe the addition of a second drive thru lane has created a bigger problem. This is causing more congestion as the one right lane blocks people who have ordered in the left lane from being able to advance and join the main paying lane which forces both lanes to immediately merge especially if the ones in the right lane tailgate a nd won't let you in even though you placed your order before them which would imply your order would come up first. You can easily sit in that left lane for minutes while the people behind you are helpless and hunk uselessly trying to move up to place their order. In addition the cashier has a hard time dispensing the correct items and keeping track of who ordered what we ordered soda but were handed bag with burgers and fries which we promptly returned. The associate then asked us if we were sure we hadn't ordered that uh what seriously way to take responsibility for your mistakes. Next time this happens we'll just take the burger and fries and get drinks elsewhere instead of being subjected to this stupid questioning and terminal waits.
' Hmmm ipads in Sooooo hygienic '
' Only reviews for McDonalds hmmm make it Let just say not the best we've been times to this location of the times they had my order wrong. Once when ordering chicken McNuggets via the Drive Thru they asked me to pull around the corner and park and they would bring it out to me. I waited minutes. On positive note if you order and decide to eat there the inside is actually pretty big and kept clean. It also has an offstrip and has easy parking. In and Out is just one more block up the street. '
' I was sad to hear that the drive thru behind the drive thru was just a drive thru. Some people really like customer
```

Gambar 5.4 List Hasil Praproses Data

5.2.3 Uji Coba Penyusunan *Sparse Matrix*

Pada tahap penyusunan *sparse matrix* dengan bantuan sklearn, dapat ditampilkan *sparse matrix* dalam bentuk format numpy, namun untuk memudahkan pengujian maka juga ditampilkan *sparse matrix* dalam format *dataframe*. Hal ini dimaksudkan untuk dapat dikenali identitas *tweet*, fitur, dan juga bobotnya. Adapun hasil uji coba adalah sebagai berikut.

```
'amn',
'amn't',
'among',
'amongst',
'amoungst',
'amount',
'amp',
'an',
'and',
'announce',
'another',
'any',
'anybody',
'anyhow',
'anymore',
'anyone',
'anything',
'anyway',
'anyways',
'anywhere',
```

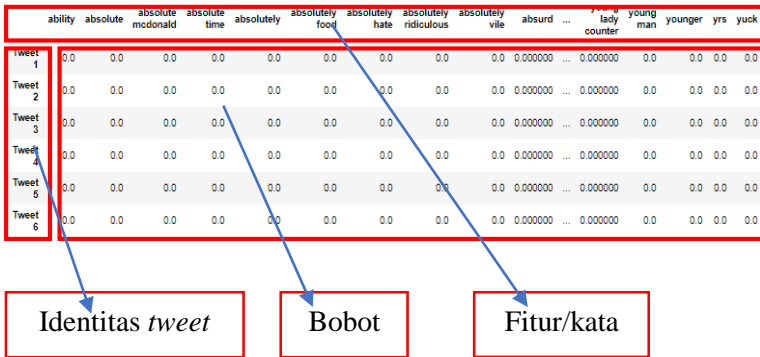
(a)

```
'altercation',
'alternated',
'alternative',
'alway',
'amazed',
'amazingly',
'america',
'american',
'amok',
'ample',
'ample seating',
'angel',
'angry',
'angus',
'angus bacon',
'angus burger',
'angus burgers',
'angus burger',
```

(b)

Gambar 5.5 (a) Daftar *Stopwords*, (b) Hasil N-gram

Sparse matrix yang didapatkan pada uji coba data *tweet* adalah sebagai berikut.



Gambar 5.6 Kutipan *Sparse Matrix*

5.2.4 Uji Coba Pemodelan Topik

Sparse matrix sebagaimana didapatkan dari uji coba pada proses sebelumnya, didekomposisi dengan *singular value decomposition* menjadi matriks singular kiri, matriks singular kanan, dan matriks berisi nilai singular. Adapun hasil penguraian *sparse matrix* adalah sebagai berikut.

	0	1	2	3	4	5	6	7	8	9 ...	1990	1991		
Tweet 1	0.041675	-0.001897	0.010016	-0.005739	-0.001644	-0.006998	-0.040295	-0.013235	-0.026902	-0.023357	...	-0.000000e+00	0.000000e+00	-0.000000
Tweet 2	0.000415	0.000010	-0.001041	-0.000185	-0.000093	-0.000777	-0.002008	-0.002544	-0.000410	0.000319	...	4.872911e-17	4.338500e-19	1.214308
Tweet 3	0.034243	0.005101	0.025813	-0.011699	0.011897	-0.014955	-0.005795	-0.010142	-0.008401	-0.016257	...	-1.8653581e-17	3.470359e-17	2.153810
Tweet 4	0.027895	-0.000282	-0.037895	-0.006847	-0.013838	0.038689	0.016788	-0.000793	0.000840	0.007285	...	8.815500e-17	1.028669e-16	-7.520687
Tweet 5	0.017119	-0.000032	-0.013584	-0.003132	-0.003152	-0.007062	-0.023777	0.011261	-0.010480	0.057385	...	7.054420e-17	-8.078706e-18	2.045005
Tweet 6	0.035641	-0.001315	-0.050630	-0.004848	-0.022810	-0.013198	0.082992	0.002498	0.019239	0.028509	...	1.829875e-16	2.298867e-17	5.617457
Tweet 7	0.024758	-0.001127	0.002646	-0.003027	-0.002216	0.022674	-0.014655	-0.003981	-0.003259	-0.004908	...	4.873432e-17	1.195564e-17	-5.713428

Gambar 5.7 Kutipan Matriks *S*

	Fitur 1	Fitur 2	Fitur 3	Fitur 4	Fitur 5	Fitur 6	Fitur 7	Fitur 8	Fitur 9	Fitur 10	...	Fitur 7271	Fitur 7272	Fitur 7273	Fitur 7274
0	0.001956	0.007987	0.002229	0.001757	0.018516	0.001490	0.002163	0.001424	0.001793	0.003941	...	0.001623	0.001425	0.003096	0.001228
1	-0.000008	0.000239	-0.000170	-0.000184	0.000646	-0.000095	-0.000203	-0.000007	-0.000110	-0.000307	...	-0.000048	0.000101	-0.000160	-0.000080
2	0.002082	-0.002223	-0.000656	0.000892	-0.006913	-0.000579	-0.000226	-0.001902	-0.003083	-0.001131	...	-0.002872	-0.000768	-0.003960	-0.000386
3	0.000015	-0.000142	-0.000676	0.000020	-0.001834	-0.000415	-0.000525	0.000043	0.000696	-0.001174	...	0.000177	-0.000125	0.000173	-0.000259
4	0.000658	0.000893	-0.001116	-0.000168	-0.000748	-0.000366	-0.001035	-0.000047	-0.000805	-0.001692	...	-0.000778	0.000241	-0.001515	-0.000446
5	0.001593	-0.001685	0.004587	-0.001814	-0.007380	-0.000985	-0.000168	-0.000030	-0.004279	0.001843	...	0.002850	-0.001235	-0.004911	-0.000423
6	-0.000258	-0.002003	-0.000159	0.000274	0.002777	-0.002809	0.001142	-0.000957	0.003812	0.000600	...	0.000799	-0.002162	0.001931	0.000348
7	0.001179	-0.003507	0.001112	-0.000223	0.001226	-0.000011	0.002404	0.000033	0.001305	0.004375	...	0.002510	-0.000011	0.000902	-0.000282
8	0.001831	-0.000644	-0.002171	-0.002219	0.007089	-0.003141	-0.003148	-0.002024	0.007775	-0.004440	...	0.000596	0.000241	0.002839	0.001711
9	0.001210	-0.005581	-0.003322	-0.001550	0.019521	-0.000783	0.004041	0.002015	0.000394	0.001956	...	0.004020	-0.000823	-0.001431	0.000806
10	-0.001195	0.000623	0.001731	0.002404	0.006728	0.000987	0.004852	-0.002390	-0.001278	0.002676	...	0.000704	-0.000457	0.000749	-0.000494
11	-0.000032	-0.000196	-0.000199	-0.000142	0.000454	-0.000154	-0.000074	0.000113	0.000086	-0.000371	...	-0.000047	-0.000191	-0.000467	-0.000031
12	0.000782	-0.001672	-0.004482	0.001535	-0.015591	0.001849	-0.000802	-0.001401	0.000470	-0.001793	...	0.001773	-0.000700	0.000213	0.002831
13	0.001656	0.003982	-0.002292	0.005566	-0.000860	-0.001751	-0.001491	0.003971	0.001444	-0.000948	...	-0.003079	0.001027	0.000212	0.001671

Gambar 5.8 Kutipan Matriks V^T

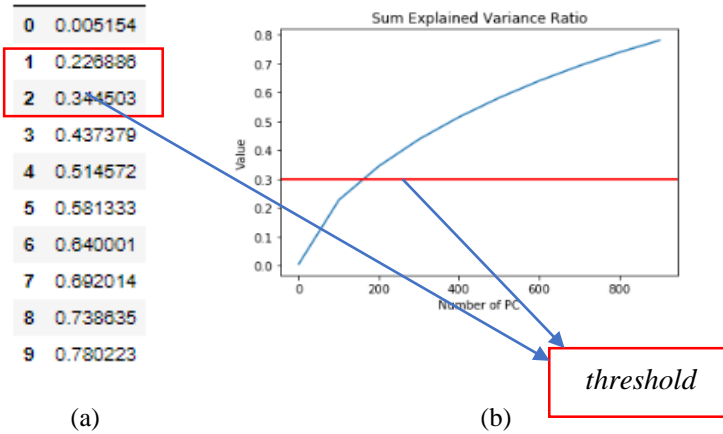
Adapun matriks Σ pada uji coba di bawah ini ditampilkan dalam bentuk urutan menurun dari nilai singular yang terbentuk. Nilai-nilai singular yang terbentuk adalah sebagai berikut.

	0
0	5.910362e+00
1	4.060972e+00
2	3.629665e+00
3	3.434922e+00
4	3.408183e+00
5	3.243396e+00
6	2.989503e+00
7	2.891923e+00
8	2.798509e+00
9	2.703568e+00
10	2.656280e+00
11	2.652919e+00

Nilai singular dituliskanurut
menurun

Gambar 5.9 Kutipan Nilai-Nilai Singular

Berikutnya adalah pengujian pada nilai *sum explained variance ratio*. Pada interval [0,1000] dengan langkah 100 didapatkan hasil pengujian seperti di bawah ini.



Gambar 5.10 (a) Nilai *Sum Explained Variance Ratio*, (b) Plot Grafik *Sum Explained Variance Ratio*

Misalkan dipilih $k = 300$, dimana memiliki nilai *sum explained variance ratio* lebih dari tiga, maka akan didapatkan matriks S_k dan Σ_k serta didapatkan matriks *tweet-principal components*-nya sebagai berikut.

	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	PC 9	PC 10	...	PC 291	PC 292	PC 293	PC 2
Tweet 1	0.548317	-0.007583	0.038354	-0.019713	-0.005904	-0.029079	-0.120371	-0.038276	-0.075297	-0.093144	...	0.007058	0.030390	-0.008259	0.0173
Tweet 2	0.002450	0.000042	-0.003780	-0.000037	-0.000316	-0.002519	-0.000599	-0.007385	-0.001151	0.000887	...	-0.028288	0.008295	-0.012150	0.0550
Tweet 3	0.002388	0.020714	0.098692	-0.040083	0.040545	-0.048506	-0.017325	-0.029930	-0.023507	-0.043956	...	-0.032722	0.027369	-0.030348	-0.0276
Tweet 4	0.004870	-0.001145	-0.137547	-0.023510	-0.047161	0.126457	0.050191	-0.022295	0.002358	0.019683	...	-0.015387	0.018490	-0.060220	0.0085
Tweet 5	0.010182	-0.000128	-0.049305	-0.010757	0.010743	-0.022903	-0.071082	0.032595	-0.026290	0.155146	...	-0.035238	-0.007612	-0.024216	0.0460
Tweet 6	0.012426	-0.005341	-0.183771	-0.019654	-0.077058	-0.042807	0.248102	0.007188	0.053838	0.077086	...	-0.038024	0.010916	0.054028	0.0335
Tweet 7	0.005593	-0.004578	0.009806	-0.010388	-0.007555	0.073639	-0.044706	-0.011509	-0.009125	-0.010110	...	-0.038650	0.044474	-0.017575	-0.0083

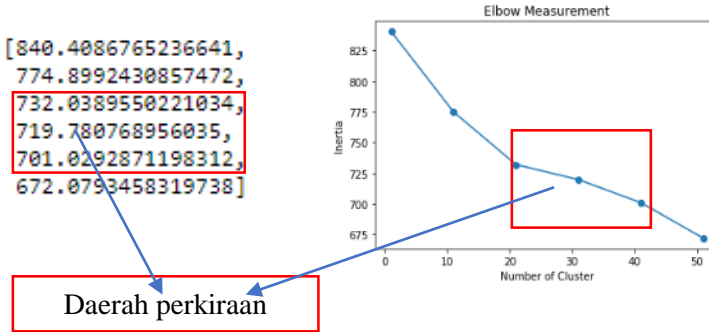
Identitas tweet

Fitur PC

Bobot

Gambar 5.11 Kutipan Matriks *Tweet-Principal Components*

Matriks seperti pada Gambar 5.11 kemudian digunakan sebagai masukan pada tahap klusterisasi dengan K-Means *cluster*. Sebagai pertimbangan dalam menentukan perkiraan jumlah kluster yang baik pada interval tertentu, maka diperhatikan nilai *inertia* dengan metode *elbow*. Hasil uji coba dengan matriks tersebut adalah sebagai berikut.



Gambar 5.12 (a) Nilai *Inertia*, (b) Plot Nilai *Inertia*

Asumsikan, jumlah kluster yang dipilih adalah 30, maka didapatkan hasil kluster sebagai berikut.

Tweet	Cluster
@McDonalds I realize McDonald's is meant to be...	50
Mmmmm ipads in @McDonalds . . . Socooo hygienic...	2
@McDonalds Only 2 reviews for a McDonalds, hmm...	34
@McDonalds I was in the Logan Sq. area and sto...	45
@McDonalds Something is wrong with this Mc Don...	27
@McDonalds Typical McDonalds food not much nut...	29
@McDonalds I was on my way home a little after...	1
One of the early EPOS (Electronic-Point-of-Sal...	2
@McDonalds This McDonald's has made a radical ...	30
@McDonalds Y'all want \$15 an hour but I'm wait...	2
@McDonalds They've remodeled, the place is gre...	2

Two red boxes with arrows point to the table. One box labeled 'Nomor Kluster' points to the 'Cluster' column. Another box labeled 'Tweet' points to the 'Tweet' column.

Gambar 5.13 Kutipan Hasil Kluster

Sehingga didapatkan hasil pemodelan topik adalah sebagai berikut.

	Jumlah tweet	Key
0	33	[nuggets, chicken, chicken nuggets, food, modon...
1	130	[drive, thru, drive thru, modonald, food]
2	434	[food, people, time, service, modonald]
3	24	[cold, poor, food, service, food cold]
4	41	[slow, service, slow service, drive, modonalds]
5	16	[chickens, living chicken, deserve lives livin...
6	67	[fries, fresh, french, cold, food]
7	7	[align actions, trust rhetoric, trust rhetoric...
8	5	[price, cashier, manager care, sour sauce]

Nomor klaster	Jumlah <i>tweet</i>	Kata kunci
---------------	---------------------	------------

Gambar 5.14 Hasil Pemodelan Topik

5.2.5 Uji Coba Analisis Sentimen

Uji coba analisis sentimen pada masing-masing *tweet* didapatkan hasil sebagai berikut.

Tweet	Polaritas
@McDonalds I realize McDonald's is meant to be ...	-0.930184
Mmmmm ipads in @McDonalds ... Soooo hygienic	0.600000
@McDonalds Only 2 reviews for a McDonalds, hmm ...	0.623458
@McDonalds I was in the Logan Sq. area and sto ...	0.914270
@McDonalds Something is wrong with this Mo Don ...	-0.559194
@McDonalds Typical McDonalds food not much nut ...	0.647518
@McDonalds I was on my way home a little after ...	0.952191
One of the early EPOS (Electronic-Point-of-Sal ...	0.000000
@McDonalds This McDonald's has made a radical ...	0.949464
@McDonalds Y'all want \$15 an hour but I'm wait ...	0.033522

Tweet	Nilai polaritas
-------	-----------------

Gambar 5.15 Kutipan Hasil Uji Coba Analisis Sentimen Pada *Tweet*

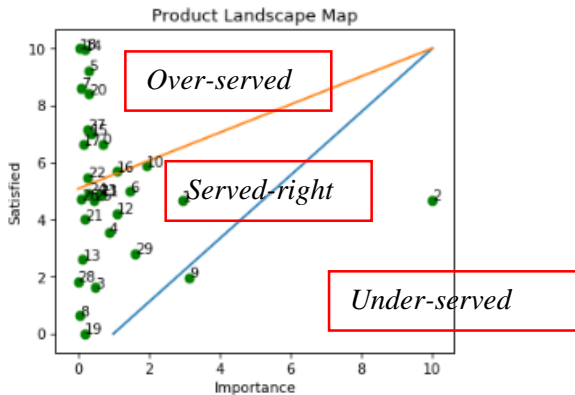
Sedangkan kutipan hasil uji coba analisis sentimen pada kluster yakni dengan menghitung rata-rata nilai polaritas sentimen pada masing-masing kluster didapatkan hasil sebagai berikut.

	Polaritas
0	-0.008874
1	-0.004801
2	-0.368054
3	-0.140461
4	0.530869
5	0.030171
6	0.457355
7	-0.481888
8	-0.322951
9	0.137022

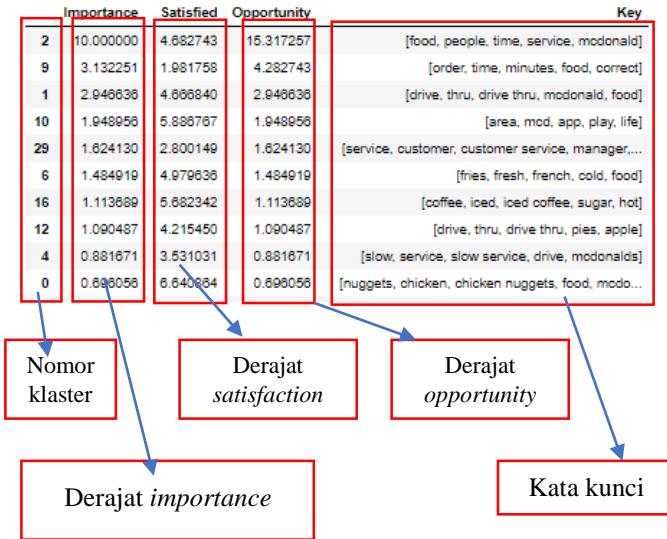
Gambar 5.16 Kutipan Hasil Analisis Sentimen pada Kluster

5.2.6 Uji Coba *Opportunity Mining*

Hasil *opportunity mining* dapat ditunjukkan pada Gambar 5.17 berikut, sedangkan rincian masing-masing derajat dapat diperhatikan pada Gambar 5.18.



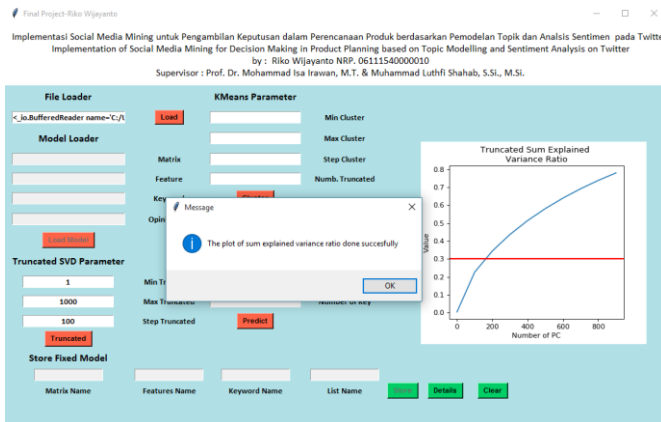
Gambar 5.17 Hasil *Opportunity Mining*



Gambar 5.18 Kutipan Nilai Derajat Hasil *Opportunity Mining*

5.2.7 Uji Coba Desain Antarmuka

Hasil uji coba pada desain antarmuka dengan menjalankan salah satu fungsi adalah sebagai berikut.



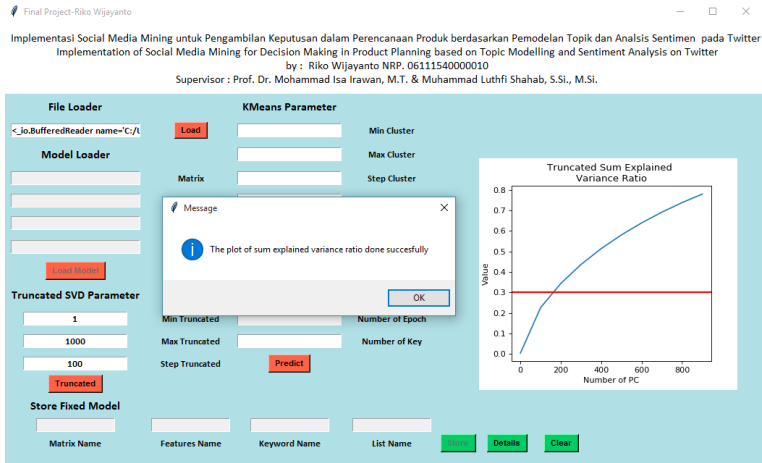
Gambar 5.19 Gambar Uji Coba Desain Antarmuka

5.3 Penentuan Skenario dan Uji Coba

Dalam menentukan skenario yang digunakan dan uji coba, data diimpor melalui desain antarmuka. Kemudian diperhatikan nilai *sum explained variance ratio* dengan memasukkan parameter-parameter pada Truncated SVD Parameter. Penulis menetapkan pengujian pertama *sum explained variance ratio* adalah pada interval [0,1000] dengan langkah sejauh 100 langkah. Sehingga didapatkan hasil seperti pada Tabel 5.2 dan Gambar 5.20 di bawah.

Tabel 5.2 Tabel Nilai *Sum Explained Variance Ratio*

Jumlah Principal Components	Nilai Sum Explained Variance Ratio
100	0,225863
200	0,343692
300	0,436430
400	0,513897
500	0,580706
600	0,639341
700	0,691553
800	0,738228
900	0,779831
1000	0,817143

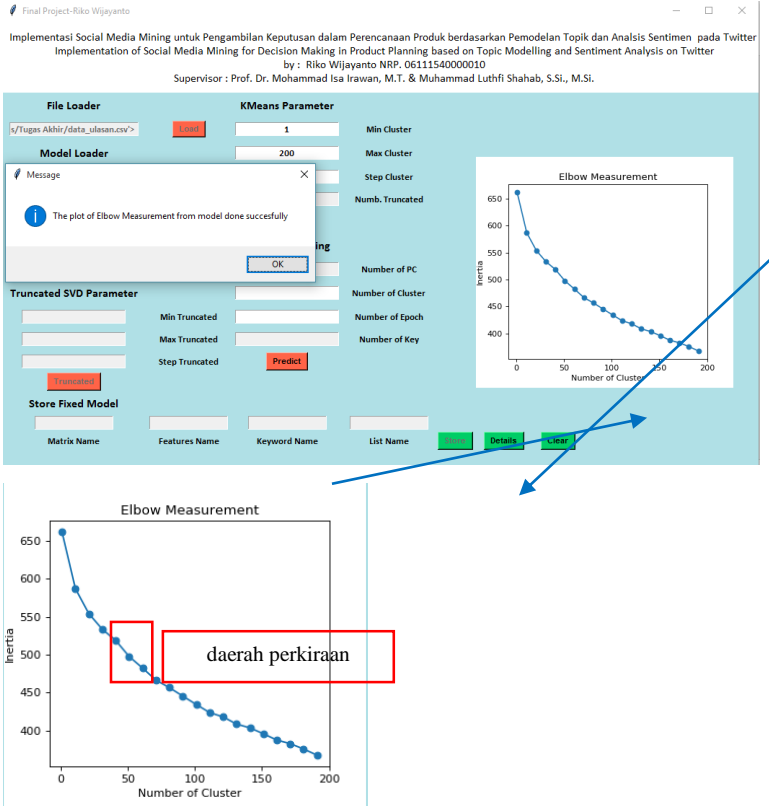


Gambar 5.20 Plot Grafik Nilai *Sum Explained Variance Ratio*

Berdasarkan hasil pengujian nilai *sum explained variance ratio*, didapatkan hasil bahwa batas ambang minimum yakni 0,3 berada pada interval nilai k sekitar interval [150,200]. Karena pemilihan k adalah sembarang di atas batas ambang, maka penulis secara subyektif dengan memerhatikan batas ambang tersebut menetapkan dua skenario dalam pemilihan nilai k , yakni $k = 200$ dan $k = 1000$. Ini berarti bahwa *principal components* yang penulis kehendaki adalah 200 dan 1000 untuk masing-masing kasus. Sehingga dalam penelitian Tugas Akhir ini, uji coba pemodelan topik akan dikerucutkan pada *truncated SVD* sebanyak 200 dan 1000 serta akan dipilih satu skenario yang lebih baik hasil pengujiannya. Namun, perlu diperhatikan bahwa pada tahap analisis sentimen hingga penarikan kesimpulan hanya akan menggunakan satu skenario berdasarkan uji *word intrusion task*.

5.3.1 Uji Coba Skenario I

Skenario I yakni *truncated SVD 200 principal components* menghasilkan hasil plot grafik nilai *inertia* sebagai berikut.



Gambar 5.21 Plot Grafik Nilai *Inertia* Skenario I

Berdasarkan Gambar 5.21, daerah perkiraan jumlah kluster adalah pada interval [40,50]. Sehingga penulis memilih jumlah kluster yang dikehendaki adalah sebanyak 42 kluster. Hasil

pemodelan topik pada skenario I dengan lima kata kunci ditunjukkan pada Tabel 5.3 berikut ini.

Tabel 5.3 Hasil Pemodelan Topik Skenario I

Klaster	Jumlah <i>Tweet</i>	Kata Kunci
1	70	<i>['service, customer, customer service, manager, order']</i>
2	57	<i>['coffee, iced, iced coffee, work, sugar']</i>
3	49	<i>['chicken, sandwich, nuggets, chicken nuggets, food']</i>
4	51	<i>['window, order, drive, thru, car']</i>
5	12	<i>['food, drive, thru, drive thru, work']</i>
6	152	<i>['mcdonalds, food, service, people, staff']</i>
7	637	<i>['food, service, breakfast, mcd, eat']</i>
8	7	<i>['drive, order, avoid, mcdonald, hard']</i>
9	156	<i>['drive, thru, drive thru, order, mcdonald']</i>
10	8	<i>['mcchicken, spicy, ordered, spicy mcchicken, dollar']</i>
11	5	<i>['cruelty, supply, cruelty supply, chicken cruelty supply, chicken cruelty']</i>
12	16	<i>['thru, drive thru, drive, order, drive thru order']</i>
13	129	<i>['mcdonald, food, service, people, eat']</i>
14	15	<i>['happy, meal, happy meal, toys, meal toys']</i>

15	14	<i>['pounder, quarter, quarter pounder, cheese, double quarter']</i>
16	11	<i>['sandwiches, menu, chicken, sandwich, chicken sandwiches']</i>
17	9	<i>['dirty, kids, hope, staff, table']</i>
18	19	<i>['job, hate, friendly, bag, employees']</i>
19	16	<i>['chickens, chickens deserve, deserve lives, lives living, lives living chicken']</i>
20	180	<i>['order, time, food, times, mcdonald']</i>
21	4	<i>['close, mcdonald, dirty, experience, eating']</i>
22	13	<i>['ice cream, machine, cream, ice, ice cream machine']</i>
23	8	<i>['dead, man, raccoon, waited, video']</i>
24	20	<i>['coke, diet, diet coke, drink, soda']</i>
25	17	<i>['coffee, cup, manager, milk, job']</i>
26	20	<i>['cheese, bacon, egg, egg cheese, biscuit']</i>
27	11	<i>['commitment, chicken commitment, chicken, store, expect']</i>
28	18	<i>['ice cream, ice, cream, cone, cream cone']</i>
29	23	<i>['egg, mcmuffin, sausage, egg mcmuffin, sausage egg']</i>
30	15	<i>['fish, filet, filet fish, cheese, fillet']</i>
31	7	<i>['align actions, align, rhetoric, trust rhetoric align, rhetoric align']</i>
32	9	<i>['apple, pies, apple pies, pie, apple pie']</i>

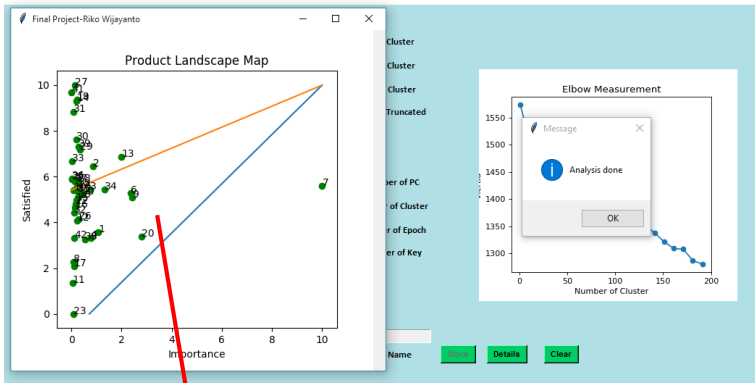
33	3	<i>['burger, king, veggie, burger king, wanted']</i>
34	86	<i>['fries, cold, french, french fries, food']</i>
35	19	<i>['burger, king, cheese, burger king, bun']</i>
36	4	<i>['dinner, messed, fries, messed order, friend']</i>
37	11	<i>['wrap, chicken, missing, lettuce, hope']</i>
38	37	<i>['minutes, waited, waiting, waited minutes, order']</i>
39	21	<i>['free, cheap, food, fries, buy']</i>
40	29	<i>['sauce, asked, nuggets, sour, sour sauce']</i>
41	2	<i>['window, coffee, hope, wait, mcdonalds']</i>
42	10	<i>['order, mess, mess order, messed, staff']</i>

Setelah dilakukan analisis sentimen secara tertutup pada uji coba skenario I, dan dilakukan perhitungan *opportunity mining* didapatkan visualisasi bagian produk pada *product landscape map* sebagai berikut.

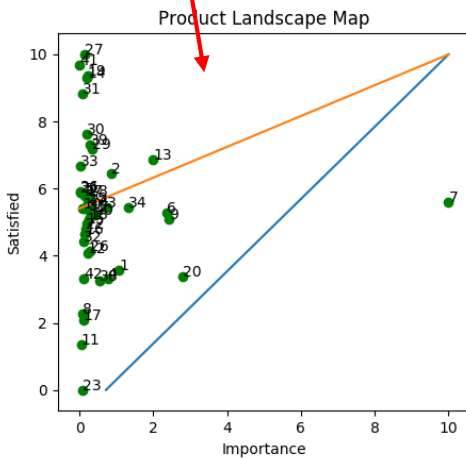
Final Project-Riko Wijayanto

Implementasi Social Media Mining untuk Pengambilan Keputusan dalam Perencanaan Produk berdasarkan Pemodelan Topik dan Analisis Sentimen pada Twitter
 Implementation of Social Media Mining for Decision Making in Product Planning based on Topic Modelling and Sentiment Analysis on Twitter

Supervisor : Prof. Dr. Mohammad Isa Irawan, M.T. & Muhammad Luthfi Shahab, S.Si., M.Si.



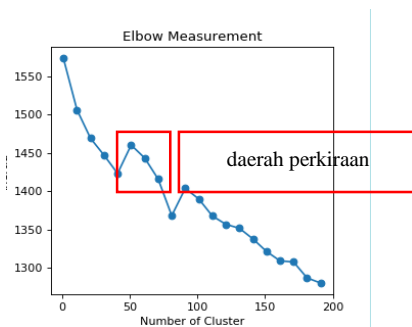
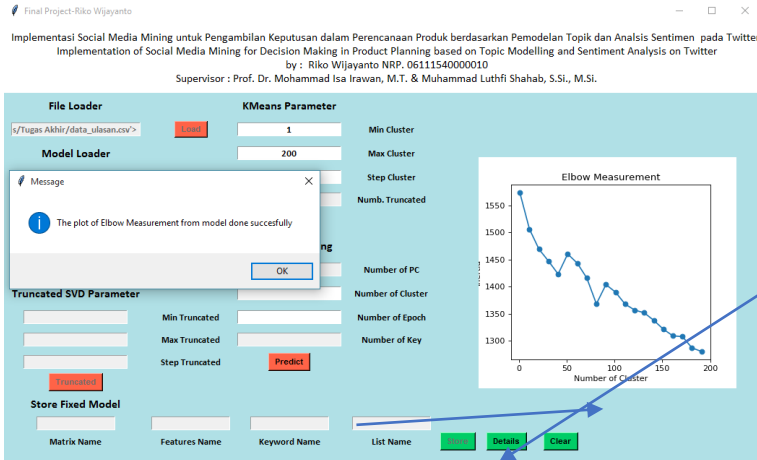
Final Project-Riko Wijayanto



Gambar 5.22 Visualisasi *Product Landscape Map* Skenario I

5.3.2 Uji Coba Skenario II

Skenario II yakni *truncated SVD* 1000 *principal components* menghasilkan hasil plot grafik nilai *inertia* sebagai berikut.



Gambar 5.23 Plot Grafik Nilai *Inertia* Skenario II

Berdasarkan Gambar 5.22, daerah perkiraan jumlah kluster adalah pada interval $[45,55]$. Sehingga penulis memilih jumlah kluster yang dikehendaki adalah sebanyak 49 kluster. Hasil

pemodelan topik pada skenario II dengan lima kata kunci ditunjukkan pada Tabel 5.4 berikut ini.

Tabel 5.4 Hasil Pemodelan Topik Skenario II

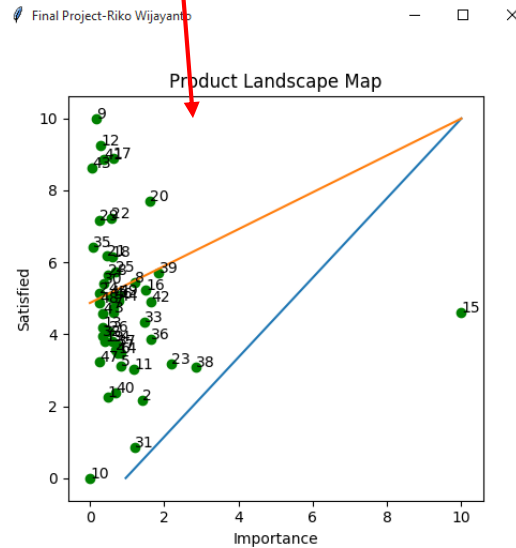
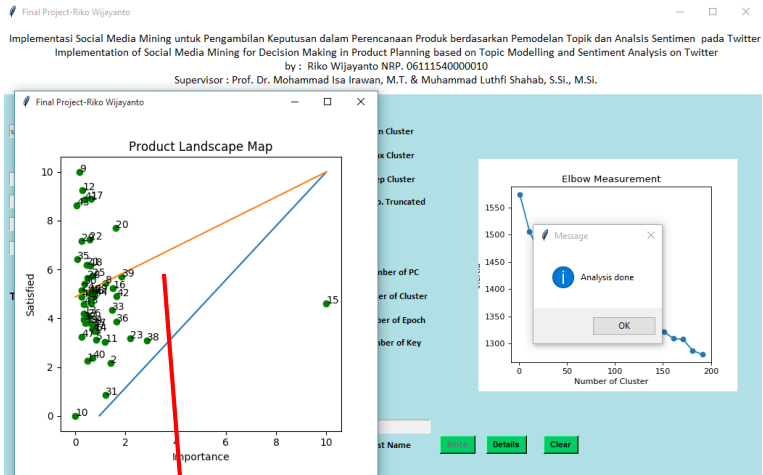
Klaster	Jumlah Tweet	Kata Kunci
1	23	<i>['check, order, check order, neighborhood, leave']</i>
2	57	<i>['window, drive, order, thru, drive thru']</i>
3	29	<i>['food, homeless, mcdonald, man, people']</i>
4	18	<i>['care, price, breakfast, person, difference']</i>
5	36	<i>['minutes, waited, waited minutes, food, time']</i>
6	36	<i>['slow, service, slow service, mcdonalds, food']</i>
7	28	<i>['ice cream, cream, ice, cones, ice cream cone']</i>
8	50	<i>['chicken, nuggets, chicken nuggets, sandwich, french']</i>
9	11	<i>['chicken commitment, commitment, chicken, sees, laughed']</i>
10	5	<i>['cruelty, supply, cruelty supply, chicken cruelty supply, chicken cruelty']</i>
11	49	<i>['people, life, change, late, mcdonald']</i>

12	16	<i>['chickens, farmed chickens, lives living chicken, lives living, farmed']</i>
13	18	<i>['cup, coffee, cup coffee, polite, extra']</i>
14	33	<i>['times, order, order times, single, time']</i>
15	376	<i>['mcdonalds, food, order, mcdonald, disgusting']</i>
16	61	<i>['coffee, iced, iced coffee, sugar, order']</i>
17	28	<i>['clean, open, yay, ketchup, hours']</i>
18	27	<i>['dinner, breakfast, lunch, mcdonald, ct']</i>
19	20	<i>['soda, drink, drink soda, mcdonald, diet']</i>
20	65	<i>['area, staff, food, mcdonalds, friendly']</i>
21	22	<i>['egg, mcmuffin, egg mcmuffin, sausage, sausage egg']</i>
22	26	<i>['hear, nice, thru, drive thru, girls']</i>
23	86	<i>['service, customer, customer service, mcdonald, food']</i>
24	15	<i>['wrap, chicken, veggie, lettuce, buffalo wrap']</i>
25	31	<i>['kids, enjoyed, playground, remember, enjoying']</i>
26	24	<i>['bacon, biscuit, egg, cheese, bacon egg']</i>
27	31	<i>['tea, drink, office, time, corporate']</i>
28	23	<i>['sauce, nuggets, sour, sour sauce, asked']</i>

29	15	<i>['fish, filet, filet fish, jeez, double']</i>
30	19	<i>['cool, ronald, interesting, job, breakfast']</i>
31	50	<i>['time, missing, food, order, order time']</i>
32	18	<i>['sandwiches, menu, bread, mustard, compared']</i>
33	60	<i>['thru, drive thru, drive, order, drive thru order']</i>
34	27	<i>['ghetto, ass, talk, mcdonalds, personal']</i>
35	8	<i>['heads, corporate, roll, boycott, breakfast']</i>
36	66	<i>['wait, order, drive, cars, times']</i>
37	33	<i>['street, mcdonalds, drive, service, people']</i>
38	111	<i>['food, drive, thru, drive thru, mcdonalds']</i>
39	74	<i>['mcdonald, food, review, drive, mcdonald mcdonald']</i>
40	31	<i>['mins, wait, pull, tonight, order']</i>
41	19	<i>['happy, meal, happy meal, meal toys, happy meal toys']</i>
42	66	<i>['fries, cold, mcdonalds, food, burgers']</i>
43	7	<i>['rhetoric align actions, trust rhetoric align, trust rhetoric, rhetoric, rhetoric align']</i>
44	34	<i>['burger, shake, king, shamrock, shamrock shake']</i>

45	25	<i>['mac, sauce, bun, mac sauce, lettuce']</i>
46	29	<i>['pounder, quarter, quarter pounder, apple, pies']</i>
47	15	<i>['ice cream machine, cream machine, ice cream, machine, cream']</i>
48	15	<i>['cheeseburger, double cheeseburger, double, mcdouble, double cheeseburger mcdouble']</i>
49	34	<i>['eat, cheap, food, restaurant, mcdonald']</i>

Setelah dilakukan analisis sentimen secara tertutup pada uji coba skenario II sebagaimana sub bab sebelumnya, dan dilakukan perhitungan *opportunity mining* didapatkan visualisasi bagian produk pada *product landscape map* sebagai berikut.



Gambar 5.24 Visualisasi *Product Landscape Map* Skenario II

5.4 Pemilihan Skenario Terbaik dan Pembahasan

Pemilihan salah satu skenario yang lebih baik dan pengambilan keputusan terkait bagian produk yang dikembangkan adalah didasarkan pada hasil pengujian *word intrusion task* serta teori-teori dalam Anthony Ulwick dalam “What Customers Want”.

5.4.1 *Word Intrusion Task*

Pada *word intrusion task*, dilakukan untuk mengukur kemudahan interpretasi hasil pemodelan topik kepada masyarakat dengan menggunakan uji sampel dari kedua skenario di atas. Pengukuran yang dilakukan adalah dengan membandingkan rata-rata nilai benar responden pada masing-masing skenario dengan uji statistik. Penulis tidak memperhitungkan proporsi dari populasi dengan mengambil minimum sampel besar yakni sebanyak 30 responden.

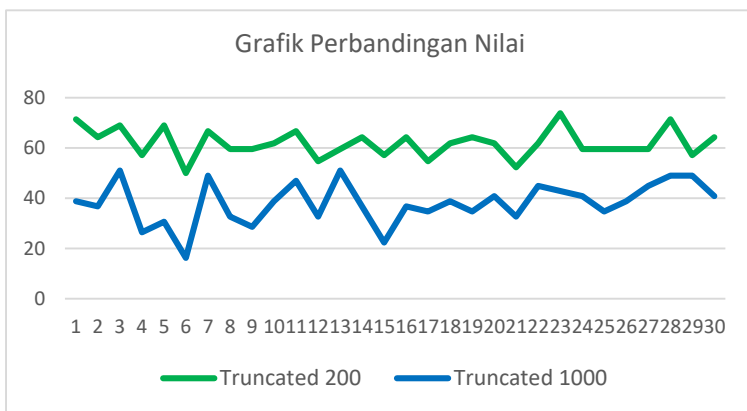
Uraian perancangan eksperimen *word intrusion task* yang penulis lakukan adalah sebagai berikut:

- a. Pada masing-masing skenario, empat kata kunci tiap klaster akan dikombinasikan dengan satu kata kunci dari klaster lain
- b. Responden akan menebak kata kunci mana yang bukan merupakan berasal dari klaster yang sama dengan kata kunci lain (menebak kata kunci yang paling berbeda).
- c. Penilaian dilakukan dengan menghitung nilai responden pada masing-masing skenario secara terpisah yakni dengan menghitung persentase jawaban benar dengan rumus $(\frac{\text{jumlah jawaban benar}}{\text{jumlah soal}} \times 100\%)$ yang kemudian disebut dengan persentase kemudahan interpretasi responden.
- d. Pengukuran dilakukan dengan membandingkan rata-rata nilai responden pada masing-masing skenario secara

berpasangan melalui uji rata-rata *random sampling* dengan σ^2 tidak diketahui dan pada level $\alpha = 0,05$.

- e. Skenario terbaik yang diambil adalah skenario yang memiliki persentase kemudahan interpretasi lebih tinggi secara statistik.

Setelah dilakukan eksperimen pada tiga puluh responden dengan kuesioner *word intrusion task* (lihat Lampiran C) didapatkan hasil seperti pada Lampiran D. Adapun grafik perbandingan hasil *word intrusion task* pada masing-masing skenario adalah sebagai berikut.



Gambar 5.25 Grafik Perbandingan Nilai WIT

Secara visual seperti pada Gambar 5.25, hasil *word intrusion task* pada pemodelan topik dengan skenario I lebih baik daripada skenario II. Secara rata-rata, skenario I dapat dikategorikan lebih baik daripada skenario II. Rata-rata skenario I adalah sebesar 61,9%, sedangkan rata-rata skenario II adalah 38,1%. Pemahaman secara sederhana menyatakan bahwa

pemodelan topik pada skenario I lebih mudah diinterpretasi daripada skenario II. Untuk itu, uji rata-rata secara statistik dengan *random sampling* berpasangan dilakukan untuk memvalidasi hasil tersebut dengan rumus (2.21) dan rumus (2.22).

Hipotesis:

H_0 : skenario I tidak lebih baik dari skenario II

H_1 : skenario I lebih baik dari skenario II

Statistik uji:

$$S = \sqrt{\frac{\sum_{i=1}^{30}(X_i - 61,9)^2 + \sum_{j=1}^{30}(Y_j - 38,1)^2}{30 + 30 - 2}} = 7,163$$

$$t^* = \frac{23,8}{7,163 \sqrt{\frac{1}{30} + \frac{1}{30}}} = 12,87$$

$$p_{value} = P(t > t^*) = 0$$

Kriteria pengujian:

Dengan $\alpha = 0,05$, $p_{value} < \alpha$ sehingga H_0 ditolak. Ini memberikan suatu kesimpulan bahwa benar skenario I lebih baik dari skenario II dan skenario I lebih mudah diinterpretasikan oleh masyarakat daripada skenario II. Perhitungan secara lengkap dapat dilihat pada Lampiran E.

Untuk menambah kajian ilmu pada metode ini, akan dilakukan pengujian pengaruh *truncated SVD* pada hasil

pemodelan topik. Pengujian dengan Uji ANOVA satu arah dengan Rancangan Acak Lengkap akan memberikan suatu kesimpulan apakah *truncated SVD* memberikan pengaruh yang berbeda secara signifikan pada hasil pemodelan topik atau tidak.

Dengan menganalisis tabel pengamatan seperti dapat dilihat pada Lampiran F, didapatkan Tabel ANOVA sebagai berikut.

Tabel 5.5 Tabel ANOVA Rancangan Acak Lengkap

Sumber Variasi	Derajat Kebebasan	Sum Squared	Mean Squared	F-Measure
<i>Truncated SVD</i>	1	8503,40	8503,40	165,71
Galat	58	2976,31	51,32	
Total	59	11479,71		

Untuk menguji apakah perlakuan *truncated SVD* memberikan pengaruh yang berbeda secara signifikan pada hasil pemodelan topik atau tidak, akan digunakan asumsi model acak.

Hipotesis:

H_0 : *truncated SVD* memberikan hasil yang sama

H_1 : *truncated SVD* memberikan hasil yang berbeda

Statistik uji:

$F - measure = 165,71$

$F - score = f_{1,58,0,05} = 4$

Kriteria pengujian:

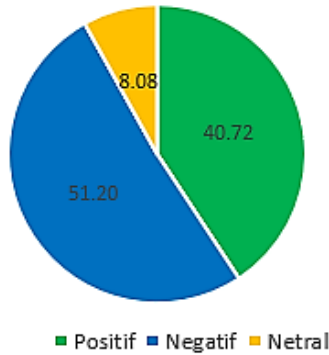
Karena $F - measure > F - score$ maka H_0 ditolak dan disimpulkan bahwa *truncated SVD* memberikan hasil yang berbeda secara signifikan.

Dari uraian pembahasan pengujian secara statistik di atas, cukup memberikan suatu alasan yang valid bahwa skenario I memberikan hasil yang lebih baik daripada skenario II. Hal ini juga diimbangi dengan suatu kesimpulan yang menyatakan bahwa pemodelan topik yang dihasilkan juga memiliki perbedaan. Kesimpulannya adalah bagian pemodelan topik yang akan dibahas untuk penentuan hasil perencanaan produk adalah dengan menggunakan skenario I.

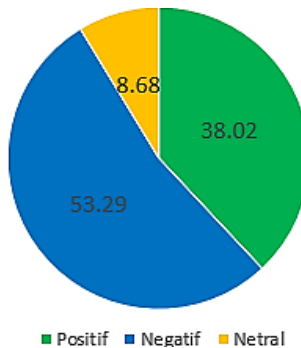
5.4.2 Pembahasan Analisis Sentimen

Sebagaimana dijelaskan pada bagian tinjauan pustaka bahwa hasil analisis sentimen dibagi menjadi tiga kelas, yakni positif, negatif, dan netral. Validasi hasil analisis sentimen pada data *tweet* yang digunakan adalah dengan menggunakan rumus (2.14), (2.15), dan rumus (2.16). Untuk melakukan analisis yang merepresentasikan dari 2000 data *tweet* yang digunakan, akan dianalisis 334 *tweet* yang dipilih secara acak berdasarkan angka acak serta metode penarikan sampel Slovin dengan mengambil $\alpha = 0,05$. Pengukuran dilakukan dengan membandingkan hasil antara analisis sentimen yang dilakukan sistem dengan aktual dengan pada kelas target positif, negatif, dan netral.

Diagram persebaran hasil analisis sentimen yang dilakukan sistem dan sentimen aktual ditampilkan pada Gambar 5.26 dan Gambar 5.27 berikut.



Gambar 5.26 Persentase Persebaran Hasil Analisis Sentimen oleh Sistem



Gambar 5.27 Persentase Persebaran Hasil Analisis Sentimen secara Aktual

Dari perhitungan pada persebaran hasil analisis sentimen didapatkan *confusion matrix* untuk kategori positif dan non-positif adalah sebagai berikut.

		Aktual	
		Positif	Non-positif
Prediksi	Positif	120	16
	Non-positif	7	191

Sehingga didapatkan nilai hasil pengukuran dengan rumus (2.14) hingga rumus (2.16).

$$accuracy = 93\%$$

$$recall = 0,94\%$$

$$precision = 0,88\%$$

Accuracy pada level 93% memberikan suatu arti bahwa persentase *tweet* yang benar diprediksi kelas sentimen secara positif non-positif oleh sistem adalah sebesar 93%. Nilai *recall* yang ditunjukkan memberikan arti bahwa persentase *tweet* positif yang benar diprediksi oleh ke dalam kelas-kelas targetnya adalah sebesar 94%. Terakhir adalah *precision*, *precision* menunjukkan bahwa 88% *tweet* berhasil diprediksi kelas targetnya dengan benar ke dalam kelas target positif.

5.4.3 Perincian *Opportunity Mining*

Pada skenario I, khususnya pada visualisasi dalam *product landscape map*, bila ditinjau secara numerik terhadap hasil perhitungan *opportunity*, didapatkan hasil seperti Tabel 5.6 berikut.

Tabel 5.6 Hasil Perhitungan *Opportunity Mining*

Klaster	<i>Importance</i>	<i>Satisfaction</i>	<i>Opportunity</i>
1	1,07	3,55	1,07
2	0,87	6,46	0,87
3	0,74	5,44	0,74
4	0,77	3,32	0,77
5	0,16	5,85	0,16
6	2,36	5,27	2,36
7	10,00	5,60	14,40
8	0,08	2,28	0,08
9	2,43	5,10	2,43
10	0,09	5,40	0,09
11	0,05	1,34	0,05
12	0,22	4,09	0,22
13	2,00	6,87	2,00
14	0,20	9,29	0,20
15	0,19	4,97	0,19
16	0,14	4,66	0,14
17	0,11	2,09	0,11
18	0,27	5,08	0,27
19	0,22	9,36	0,22
20	2,80	3,37	2,80
21	0,03	5,88	0,03
22	0,17	4,80	0,17
23	0,09	0,00	0,19
24	0,28	5,51	0,28
25	0,24	5,55	0,24
26	0,28	4,14	0,28
27	0,14	10,00	0,14
28	0,25	5,80	0,25

29	0,33	7,18	0,33
30	0,20	7,64	0,20
31	0,08	8,81	0,08
32	0,11	4,42	0,11
33	0,02	6,66	0,02
34	1,32	5,43	1,32
35	0,27	5,35	0,27
36	0,03	5,93	0,03
37	0,14	5,80	0,14
38	0,55	3,26	0,55
39	0,30	7,30	0,30
40	0,43	5,23	0,43
41	0,00	9,69	0,00
42	0,13	3,30	0,13

Bila masing-masing bagian diberikan label kelas sesuai dengan topik yang bersesuaian, akan didapatkan label seperti ditampilkan pada Tabel 5.7 di bawah ini. Pelabelan dilakukan dengan menganalisis kata kunci pada tiap-tiap kluster serta mempertimbangkan data *tweet* yang sebenarnya. Hal ini dilakukan karena dimungkinkan terdapat kluster dengan kata kunci yang sama, namun secara semantik dalam *tweet* memiliki suatu makna yang berbeda. Oleh karena itu, diidentifikasi langsung pada data yang sebenarnya perlu dilakukan.

Tabel 5.7 Pelabelan Hasil Pemodelan Topik

Kluster	Kata Kunci	Nama Label
1	<i>['service, customer, customer service, manager, order']</i>	Manajer dan customer service

2	<i>['coffee, iced, iced coffee, work, sugar']</i>	Minuman kopi
3	<i>['chicken, sandwich, nuggets, chicken nuggets, food']</i>	<i>Chicken nugget</i>
4	<i>['window, order, drive, thru, car']</i>	Layanan <i>drive thru</i> untuk mobil
5	<i>['food, drive, thru, drive thru, work']</i>	Pelayan di bagian <i>drive thru</i>
6	<i>['mcdonalds, food, service, people, staff']</i>	Pelayanan konsumen (pelayan)
7	<i>['food, service, breakfast, mcd, eat']</i>	Pelayanan di pagi hari
8	<i>['drive, order, avoid, mcdonald, hard']</i>	Layanan pesan antar
9	<i>['drive, thru, drive thru, order, mcdonald']</i>	Pemesanan makanan dengan <i>drive thru</i>
10	<i>['mcchicken, spicy, ordered, spicy mcchicken, dollar']</i>	Menu <i>spicy mcchicken</i>
11	<i>['cruelty, supply, cruelty supply, chicken cruelty supply, chicken cruelty']</i>	Pasokan daging ayam
12	<i>['thru, drive thru, drive, order, drive thru order']</i>	Sistem pemesanan di <i>drive thru</i>
13	<i>['mcdonald, food, service, people, eat']</i>	Pelayanan makanan
14	<i>['happy, meal, happy meal, toys, meal toys']</i>	Bonus mainan anak
15	<i>['pounder, quarter, quarter pounder, cheese, double quarter']</i>	Porsi makanan (keju)

16	<i>['sandwiches, menu, chicken, sandwich, chicken sandwiches']</i>	<i>Chicken sandwich</i>
17	<i>['dirty, kids, hope, staff, table']</i>	Kawasan bermain anak
18	<i>['job, hate, friendly, bag, employees']</i>	Keramahan pegawai
19	<i>['chickens, chickens deserve, deserve lives, lives living, lives living chicken']</i>	Penyedia daging ayam
20	<i>['order, time, food, times, mcdonald']</i>	Durasi pemesanan makanan
21	<i>['close, mcdonald, dirty, experience, eating']</i>	Kebersihan tempat
22	<i>['ice cream, machine, cream, ice, ice cream machine']</i>	Es krim dan mesin es krim
23	<i>['dead, man, raccoon, waited, video']</i>	Kebersihan makanan
24	<i>['coke, diet, diet coke, drink, soda']</i>	Minuman bersoda
25	<i>['coffee, cup, manager, milk, job']</i>	Porsi minuman kpi
26	<i>['cheese, bacon, egg, egg cheese, biscuit']</i>	<i>Bacon dan telur keju</i>
27	<i>['commitment, chicken commitment, chicken, store, expect']</i>	Lain-lain
28	<i>['ice cream, ice, cream, cone, cream cone']</i>	Cone es krim dan porsi es krim
29	<i>['egg, mcmuffin, sausage, egg mcmuffin, sausage egg']</i>	<i>Mcmuffin</i>
30	<i>['fish, filet, filet fish, cheese, fillet']</i>	Menu ikan fillet

31	<i>['align actions, align, rhetoric, trust rhetoric align, rhetoric align']</i>	Lain-lain
32	<i>['apple, pies, apple pies, pie, apple pie']</i>	<i>Pie</i> apel
33	<i>['burger, king, veggie, burger king, wanted']</i>	<i>Burger</i> isi sayur
34	<i>['fries, cold, french, french fries, food']</i>	Kentang goreng
35	<i>['burger, king, cheese, burger king, bun']</i>	Burger isi keju
36	<i>['dinner, messed, fries, messed order, friend']</i>	Menu makan malam
37	<i>['wrap, chicken, missing, lettuce, hope']</i>	<i>Packaging</i>
38	<i>['minutes, waited, waiting, waited minutes, order']</i>	Lama menunggu pelayanan
39	<i>['free, cheap, food, fries, buy']</i>	Harga
40	<i>['sauce, asked, nuggets, sour, sour sauce']</i>	Saos
41	<i>['window, coffee, hope, wait, mcdonalds']</i>	Kedisiplinan pekerja
42	<i>['order, mess, mess order, messed, staff']</i>	Pelayanan bagian pencatatan pesanan

Visualisasi pada *product landscape map* akan dihasilkan pemetaan bagian-bagian produk yang dalam Tabel 5.8 disebut dengan label. Adapun posisi bagian produk yang dirumuskan adalah *under-served*, *right-served*, dan *over-served* dimana kajian pemetaan ini didasarkan pada penelitian Anthony Ulwick.

Tabel 5.8 Pemetaan Posisi Bagian Produk

No.	Bagian produk	Posisi
1	Manajer dan <i>customer service</i>	<i>Served right</i>
2	Minuman kopi	<i>Overserved</i>
3	<i>Chicken nugget</i>	<i>Served right</i>
4	Layanan <i>drive thru</i> untuk mobil	<i>Served right</i>
5	Pelayan di bagian <i>drive thru</i>	<i>Overserved</i>
6	Pelayanan konsumen (pelayan)	<i>Served right</i>
7	Pelayanan di pagi hari	<i>Underserved</i>
8	Layanan pesan antar	<i>Served right</i>
9	Pemesanan makanan dengan <i>drive thru</i>	<i>Served right</i>
10	Menu <i>spicy mcchicken</i>	<i>Overserved</i>
11	Pasokan daging ayam	<i>Served right</i>
12	Sistem pemesanan di <i>drive thru</i>	<i>Served right</i>
13	Pelayanan makanan	<i>Overserved</i>
14	Bonus mainan anak	<i>Overserved</i>
15	Porsi makanan (keju)	<i>Served right</i>
16	<i>Chicken sandwich</i>	<i>Served right</i>
17	Kawasan bermain anak	<i>Served right</i>
18	Keramahan pegawai	<i>Served right</i>
19	Penyedia daging ayam	<i>Overserved</i>
20	Durasi pemesanan makanan	<i>Served right</i>
21	Kebersihan tempat	<i>Overserved</i>
22	Es krim dan mesin es krim	<i>Served right</i>
23	Kebersihan makanan	<i>Served right</i>
24	Minuman bersoda	<i>Overserved</i>

25	Porsi minuman kpi	<i>Overserved</i>
26	<i>Bacon</i> dan telur keju	<i>Served right</i>
27	Lain-lain	<i>Overserved</i>
28	<i>Cone</i> es krim dan porsi es krim	<i>Overserved</i>
29	<i>Mcmuffin</i>	<i>Overserved</i>
30	Menu ikan fillet	<i>Overserved</i>
31	Lain-lain	<i>Overserved</i>
32	<i>Pie</i> apel	<i>Served right</i>
33	<i>Burger</i> isi sayur	<i>Overserved</i>
34	Kentang goreng	<i>Served right</i>
35	Burger isi keju	<i>Served right</i>
36	Menu makan malam	<i>Overserved</i>
37	<i>Packaging</i>	<i>Overserved</i>
38	Lama menunggu pelayanan	<i>Served right</i>
39	Harga	<i>Overserved</i>
40	Saos	<i>Served right</i>
41	Kedisiplinan pekerja	<i>Overserved</i>
42	Pelayanan bagian pencatatan pesanan	<i>Served right</i>

5.4.4 Pembahasan Keputusan

Nilai *opportunity* dari keseluruhan bagian produk dan visualisasi yang dapat diperhatikan pada Tabel 5.7 dan Tabel 5.8, jumlah bagian produk yang berada pada wilayah *under-served* adalah satu bagian, yakni pelayanan di pagi hari dan juga memiliki *opportunity* tertinggi yakni sebesar 14,40. Posisi *under-served* pada pelayanan di pagi hari menggambarkan bahwa bagian penting dari McDonald's yang seharusnya diutamakan,

namun memiliki tingkat kepuasan yang relatif rendah dibanding kepentingannya dalam masyarakat. Ini mengindikasikan bahwa pelanggan kurang puas pada hal tersebut. Nilai *opportunity* sebesar 14,40 memberikan suatu tinjauan menarik lain. Ulwick, dalam bukunya, menjelaskan bahwa suatu bagian produk dalam pangsa pasar yang memiliki nilai *opportunity* pada rentang $12 \leq opportunity \leq 15$ merupakan bagian yang berpeluang bila direncanakan dengan baik.

Bila memerhatikan kondisi saat ini di masyarakat, terkadang banyak dijumpai restoran yang tidak memberikan pelayanan dengan optimal di pagi hari atau bahkan belum buka. Hal ini tentunya menjadi salah satu trik menarik untuk memperoleh keuntungan yang lebih baik dibandingkan dengan kompetitor.

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini diuraikan kesimpulan dari serangkaian penelitian Tugas Akhir yang dilakukan. Kesimpulan dari hasil pengujian serta analisis dirangkum dan diwujudkan dalam bab ini. Selain itu saran dan pengembangan untuk penelitian di masa mendatang akan dituliskan untuk mempermudah peneliti selanjutnya dalam mengisi celah-celah penelitian Tugas Akhir ini.

6.1 Kesimpulan

Setelah melalui serangkaian tahap, dari perancangan, pengujian, hingga analisis didapatkan kesimpulan sebagai berikut.

1. Implementasi pemodelan topik pada kumpulan *tweet* dengan menggunakan *Latent Semantic Analysis* dan *K-Means cluster* dilakukan dengan mereduksi *sparse matrix* menjadi matriks *tweet-principal components* terlebih dahulu. Setelah didapatkan matriks *tweet-principal components* kemudian dilakukan klasterisasi dengan menggunakan *K-Means Cluster*. Salah satu pertimbangan penentuan k dan berapa jumlah klaster dapat diperhatikan dari nilai *sum explained variance ratio* dan juga nilai *inertia*. Dalam dua skenario pada penelitian Tugas Akhir ini, didapatkan hasil bahwa jumlah *truncated SVD* dan jumlah klaster yang lebih baik berdasarkan *word intrusion task* adalah sebanyak 200 *principal components* dengan 42 klaster (topik). Hasil ini juga didukung oleh validasi secara statistik yang menyatakan bahwa hasil intrusi pada skenario tersebut lebih baik dari skenario yang lain.

2. Implementasi analisis sentimen pada kumpulan *tweet* dilakukan dengan melakukan pencocokan *tweet* dengan kata perasa (*feeling words*) yang sudah dibobotkan pada *lexicon* C. J. Hutto dan Eric Gilbert. Pengukuran dilakukan dengan memeriksa apakah kata tersebut berada dalam daftar *lexicon* atau tidak, memiliki unsur kata *booster* atau tidak, serta secara umum memiliki tanda baca berlebih atau tidak. Sedemikian hingga, analisis sentimen menghasilkan polaritas opini yang dapat dinormalisasi pada interval $[-1,1]$ untuk menghilangkan efek ragam yang tinggi. Adapun nilai pengukuran kualitas sistem dengan *accuracy*, *recall*, dan *precision* berturut-turut adalah 93%, 94%, dan 88%.
3. Implementasi *social media mining* dalam perencanaan produk dengan *opportunity mining* dilakukan dengan menghitung derajat *importance* yang berasal dari jumlah *tweet* per klaster, *derajat satisfaction* yang berasal dari rata-rata polaritas opini tiap-tiap klaster, dan yang terakhir adalah perhitungan *opportunity*. Hasil analisis dalam penelitian Tugas Akhir ini adalah sebuah keputusan yang menyatakan bahwa “pelayanan di pagi hari” memiliki *opportunity* tertinggi yakni sebesar 14,40 dengan posisi pada *under-served*. Sehingga perencanaan dalam bidang ini perlu dilakukan dengan matang dan dengan baik.

6.2 Saran

Adapun saran yang dapat diberikan untuk penelitian *social media mining* dikemudian hari, dirangkum dalam pernyataan di bawah ini.

1. Sebaiknya menggunakan metode pemodelan topik yang lebih mempertimbangkan konvergensi hasil pemodelan serta berpatokan pada peluang suatu kata terhadap keseluruhan dokumen.

2. Hendaknya menggunakan metode yang lebih mempertimbangkan kalimat bergaya bahasa satire dalam perhitungan polaritas opini. Misalnya adalah dalam kondisi ruangan yang kotor, pelanggan mengatakan bahwa ruangan tersebut bersih sebagai salah satu kalimat sindiran.
3. Pemrosesan data secara *real time* atau daring dapat dijadikan sebagai salah satu alternatif sehingga lebih dinamis pada kebutuhan perusahaan.
4. Analisis pengambilan keputusan dengan menambahkan usulan strategi terbaik dengan berpatokan pada *opportunity mining* perlu dipertimbangkan untuk melengkapi analisis yang sudah dipaparkan.

DAFTAR PUSTAKA

- [1] Silverstein, David, Philip Samuel, dan Neil DeCarlo. 2013. *The Innovator's toolkit: 50+ techniques for predictable and sustainable organic growth*. John Wiley & Sons.
- [2] Park, Hyunseok. dan Janhyeok Yoon. 2015. *A Chance Discovery based Approach for New Product Service System (PSS) Concepts*. Service Business Journal. Vol. I, No. 9 : 115-135.
- [3] Brooks, Stoney. 2015. *Does Personal Social Media Usage Affect Efficiency and Well Being?*. Journal of Computers in Human Behaviour. Vol. 46, No. 1: 26-37.
- [4] Ulwick, Anthony. 2005. *What Customers want : Using Outcome Driven Innovation to Create Breakthrough Products and Services*. New York : McGraw-Hill Companies, Inc.
- [5] Jeong, Byeongki, Janhyeok Yoon dan Jae-Min Lee. 2017. *Social Media Mining for Product Planning : A Product Opportunity Mining Approach Based on Topic Modelling and Sentiment Analysis*. International Journal of Information Management. DOI : <http://dx.doi.org/10.1016/j.ijinfomgt.2017.09.009>.
- [6] Seo, Wonchul dkk. 2016. *Product Opportunity Identification based on Internal Capabilities Using Text Mining and Association Rule*. Journal of Technological Forecasting and Social Change. Vol. 105 : 94-104,
- [7] Kou, Gang. 2015. *An Application of Latent Semantic Analysis for Text Categorization*. International Journal of

- Computers, Communications & Control. Vol. 10, No. 3 : 357-369.
- [8] Chauhan, Vipul Kumar, Anshish Bansal dan Amita Goel. 2018. *Twitter Sentiment Analysis Using Vader*. International Journal of Advance Research, Ideas and Innovations in Technology. Vol. 4, No. 1 : 485-489.
- [9] Sajithra, K., dan Rajindra Patil. 2013. *Social Media-History and Components*. IOSR Journal of Business and Management. Vol. 7, No. 1 : 69-74.
- [10] Zafarani, Reza, Mohammad Ali Abbasi, dan Huan Liu. 2014. *Social Media Mining : An Introduction*. London: Cambridge University Press.
- [11] Manning, Christopher D., Prabhakar Raghavan dan Hinrich Schutze. 2008. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press.
- [12] Dennis, Simon. 2007. *Handbook of Latent Semantic Analysis*. London: Routledge.
- [13] Han, Jiawei, Micheline Kamber, dan Jian Pei. 2012. *Data Mining : Concepts and Techniques 3rd edition*. Waltham: Morgan Kauffman Publisher.
- [14] Hutto, Clayton J. dan Eric Gilbert. 2014. *VADER : A Parsinomous Rule-Based Model for Sentiment Analysis of Social Media Text*. ICWSM, The AAI Press.
- [15] Chang, Jonathan dkk. 2009. *Reading Tea Leaves: How Humans Interpret Topic Models*. Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Vancouver, British Columbia, Canada.

- [16] Box, George E.P., William G. Hunter, dan J. Stuart Hunter. 1978. *Statistics for Experiments : An Introduction to Design, Data Analysis, and Model Building*. USA: John Willey & Sons.

LAMPIRAN A. Tabel Perbandingan Penelitian Terdahulu

No.	Judul Penelitian	Penulis (Tahun)	Metode yang Digunakan	Obyek Penelitian
1	<i>Product opportunity identification based on internal capabilities using text mining and association rule mining</i>	Wonchul, Seo dkk (2016)	<i>Association rule mining</i>	Canon Inc.
2	<i>An Application of Latent Semantic Analysis for Text Categorization</i>	Kou, Gang (2015)	<i>Latent Semantic Analysis</i>	Dokumen dengan tema Multicriteria Decision Making
3	<i>Social Media Mining for Product Planning : A Product Opportunity Mining Approach Based on Topic Modelling and Sentiment Analysis.</i>	Jeong, Byeongki, Janhyeok Yoon dan Jae-Min Lee (2017)	<i>Latent Dirichlet Allocation dan AlchemyAPI</i>	Samsung Galaxy Note-5

Lanjutan Lampiran A

No.	Judul Penelitian	Penulis (Tahun)	Metode yang Digunakan	Obyek Penelitian
4	<i>Twitter Sentiment Analysis Using Vader</i>	Chauhan, Vipul Kumar, Anshish Bansal dan Amita Goel (2015)	VADER	Data <i>tweet</i> bertajuk ekonomi, kesehatan, sosial, dan sebagainya

LAMPIRAN B. Perhitungan *Singular Value Decomposition*

Misalkan diberikan matriks $A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix}$, matriks A memiliki

dimensi $m \times n$, yakni 3×2 . Tranpose matriks A , dilambangkan dengan A^T , didapat $A^T = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix}$.

Karena $m > n$, sehingga didapatkan

$$B = A^T A = \begin{bmatrix} 1 & 1 & -1 \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix}$$

Persamaan karakteristik, $P(\lambda)$, matriks B dapat dicari dengan $\det(\lambda I - B) = 0$, sehingga didapat:

$$P(\lambda) = \det(\lambda I - B) = \begin{vmatrix} \lambda - 3 & -1 \\ -1 & \lambda - 3 \end{vmatrix} = (\lambda - 4)(\lambda - 2) = 0$$

Apabila dicari akar-akarnya, maka didapat $\lambda_1 = 4, \lambda_2 = 2$. Vektor-vektor eigen yang bersesuaian dapat dicari dengan langkah sebagai berikut:

$$\text{Untuk } \lambda = 4, \text{ didapat } \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = a \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\text{Untuk } \lambda = 2, \text{ didapat } \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = b \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Kemudian, normalisasi vektor eigen menjadi matriks orthogonal V sebagai berikut:

Lanjutan Lampiran B

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Maka didapatkan vektor-vektor singular kanan adalah $v_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$,

dan $v_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$,

Berikutnya adalah pencarian nilai singular yang bersesuaian dengan nilai eigen,

$$\delta_1 = \sqrt{4} = 2, \delta_2 = \sqrt{2}$$

sehingga didapat matriks diagonal $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & \sqrt{2} \end{bmatrix}$,

Langkah selanjutnya adalah pencarian matriks singular kiri sebagai berikut:

$$s_1 = \frac{1}{\delta_1} A v_1 = \frac{1}{2} \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix}$$

$$s_2 = \frac{1}{\delta_2} A v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Lanjutan Lampiran B

Kemudian didapat matriks singular kiri S yakni:

$$S = \begin{bmatrix} 0 & -1 \\ \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 \end{bmatrix}$$

Langkah *truncated SVD* dengan mengambil $k=2$ yakni:

Dan didapatkan matriks

$$S \times \Sigma = \begin{bmatrix} 0 & -1 \\ \frac{1}{2} & 0 \\ -\frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 & -\sqrt{2} \\ \frac{2}{\sqrt{2}} & 0 \\ -\frac{2}{\sqrt{2}} & 0 \end{bmatrix}$$

Adapun *truncated SVD* dengan $k = 1$, didapat

$$S_1 \Sigma_1 = \begin{bmatrix} 0 \\ 2 \\ \frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix}$$

LAMPIRAN C. Kuesioner *Word Intrusion Task*



Kuisisioner *Word Intrusion Task*
Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Informasi Responden

Nama : NRP :

Pilihlah satu kata dari lima pilihan kata pada masing-masing pertanyaan berikut yang menurut Anda PALING BERBEDA, SUSUNAN KATANYA BERBEDA, PENULISAN BERBEDA, ATAU TIDAK COCOK MENJADI SATU KESATUAN TOPIK DENGAN YANG LAJN.

Contoh : Diantara kata *apple, mango, orange, banana, dan train*, *train* yang paling berbeda dengan kata lain karena berasal dari unsur transportasi. Sedang keempat yang lain adalah dari unsur buah.

BAGIAN I

- | | | | | |
|------------------|---------------|---------------------|----------------------|--------------------------|
| 1. a. service | b. customer | c. customer service | d. manager | e. sugar |
| 2. a. windows | b. coffe | c. sugar | d. iced | e. iced coffe |
| 3. a. chicken | b. sandwich | c. nuggets | d. chicken nuggets | e. sugar |
| 4. a. order | b. thru | c. drive | d. window | e. breakfast |
| 5. a. food | b. drive | c. happy | d. drive thru | e. work |
| 6. a. mcdonalds | b. toys | c. service | d. people | e. staff |
| 7. a. food | b. breakfast | c. drive thru | d. mcd | e. eat |
| 8. a. drive | b. supply | c. order | d. mcdonald | e. hard |
| 9. a. drive thru | b. order | c. drive | d. thru | e. spicy |
| 10. a. mcchicken | b. spicy | c. spicy mcchicken | d. bag | e. ordered |
| 11. a. supply | b. cruelty | c. chicken cruelty | d. table | e. cruelty supply |
| 12. a. thru | b. sandwich | c. order | d. drive thru order | e. drive thru |
| 13. a. food | b. close | c. mcdonald | d. service | e. eat |
| 14. a. happy | b. toys | c. experience | d. happy meal | e. meal toys |
| 15. a. hope | b. quarter | c. pounder | d. double quarter | e. quarter pounder |
| 16. a. dirty | b. sandwiches | c. menu | d. chicken | e. chicken sandwiches |
| 17. a. dirty | b. kids | c. windows | d. staff | e. table |
| 18. a. job | b. chicken | c. employess | d. friendly | e. hate |
| 19. a. dollar | b. chickens | c. chickens deserve | d. lives living | e. lives living chickens |
| 20. a. order | b. time | c. eating | d. food | e. mcdonalds |
| 21. a. close | b. mcdonalds | c. eating | d. experience | e. spicy |
| 22. a. ice cream | b. ice | c. manager | d. ice cream machine | e. machine |
| 23. a. dead | b. raccoon | c. man | d. waited | e. cone |
| 24. a. diet | b. align | c. coke | d. drink | e. soda |
| 25. a. coffe | b. sausage | c. cup | d. job | e. manager |

Lanjutan Lampiran C

26. a. cheese	b. fish	c. egg	d. egg cheese	e. biscuit
27. a. store	b. fish	c. commitment	d. chicken	e. chicken commitment
28. a. cone	b. ice cream	c. cream	d. cream cone	e. egg
29. a. mcmuffin	b. cream	c. sausage	d. sausage egg	e. egg mcmuffin
30. a. burger	b. filet	c. fish	d. filet fish	e. cheese
31. a. rhetoric	b. align actions	c. apple	d. rhetoric align	e. trust rhetoric align
32. a. pies	b. hope	c. apple	d. apple pie	e. pies
33. a. burger	b. wrap	c. king	d. veggie	e. burger king
34. a. wrap	b. cold	c. fries	d. food	e. french fries
35. a. dinner	b. burger	c. king	d. cheese	e. burger king
36. a. dinner	b. buy	c. messed	d. messed order	e. ice cream cone
37. a. missing	b. lettuce	c. cone	d. chicken	e. wrap
38. a. free	b. minutes	c. waited	d. order	e. waiting
39. a. bacon	b. free	c. food	d. fried	e. buy
40. a. sour	b. machine	c. sauce	d. nuggets	e. sour sauce
41. a. mcdonalds	b. ice	c. coffe	d. wait	e. coffe
42. a. mess	b. mess order	c. messed	d. messed	e. video

BAGIAN II

1. a. check	b. order	c. check order	d. difference	e. leave
2. a. window	b. cone	c. drive	d. drive thru	e. order
3. a. food	b. mcdonald	c. man	d. people	e. change
4. a. care	b. price	c. breakfast	d. person	e. disgusting
5. a. minutes	b. extra	c. food	d. time	e. waited
6. a. slow	b. extra	c. slow service	d. food	e. mcdonalds
7. a. ice cream	b. cream	c. cones	d. coffe	e. ice cream cone
8. a. coffe	b. nuggets	c. chicken	d. sandwich	e. french
9. a. chicken	b. peas	c. single	d. laughed	e. commitment
10. a. supply	b. chicken	c. chicken cruelty	d. cruelty supply	e. chicken cruelty supply
11. a. people	b. late	c. change	d. time	e. mcdonald
12. a. chicken	b. farmed chicken	c. farmed	d. open	e. lives living
13. a. cup	b. ketchup	c. polite	d. extra	e. coffe
14. a. mcdonalds	b. disgusting	c. iced coffe	d. order	e. food
15. a. iced	b. iced coffe	c. sugar	d. order	e. ketchup
16. a. dinner	b. clean	c. open	d. yay	e. hours
17. a. dinner	b. breakfast	c. lunch	d. ct	e. chicken
18. a. soda	b. food	c. drink	d. drink soda	e. diet

Lanjutan Lampiran C

- | | | | | |
|-----------------------|-------------------|---------------------|-------------------|-------------------------|
| 19. a. area | b. staff | c. egg | d. friendly | e. food |
| 20. a. McMuffin | b. sausage | c. egg | d. sausage egg | e. McDonald |
| 21. a. order | b. times | c. single | d. order time | e. drive thru |
| 22. a. nice | b. thru | c. single | d. drive thru | e. girls |
| 23. a. service | b. customer | c. lettuce | d. food | e. McDonald |
| 24. a. tea | b. chicken | c. buffalo wrap | d. veggie | e. wrap |
| 25. a. enjoyed | b. order | c. remember | d. playground | e. enjoying |
| 26. a. nuggets | b. egg | c. cheese | d. biscuit | e. bacon |
| 27. a. tea | b. drink | c. corporate | d. time | e. bread |
| 28. a. asked | b. sour | c. sausage | d. fish | e. nuggets |
| 29. a. filet | b. filet fish | c. jazz | d. nuggets | e. double |
| 30. a. job | b. interesting | c. breakfast | d. Ronald | e. McDonald |
| 31. a. drive | b. order | c. order time | d. missing | e. food |
| 32. a. sandwiches | b. compared | c. bread | d. menu | e. dinner |
| 33. a. order | b. drive | c. thru | d. drive thru | e. breakfast |
| 34. a. ghetto | b. ass | c. personal | d. McDonald | e. wait |
| 35. a. roll | b. corporate | c. order | d. boycott | e. breakfast |
| 36. a. times | b. cars | c. McDonalds | d. drive | e. wait |
| 37. a. street | b. drive | c. McDonalds | d. service | e. bacon |
| 38. a. food | b. thru | c. drive | d. double | e. McDonalds |
| 39. a. order | b. review | c. McDonald | d. food | e. drive |
| 40. a. mins | b. wait | c. meal | d. pull | e. tonight |
| 41. a. happy | b. meal | c. happy meal | d. meal toys | e. lettuce |
| 42. a. burgers | b. cream | c. cold | d. McDonald | e. fries |
| 43. a. rhetoric | b. trust rhetoric | c. cold | d. rhetoric align | e. trust rhetoric align |
| 44. a. shake | b. king | c. shamrock | d. shamrock shake | e. lettuce |
| 45. a. sauce | b. mac | c. lettuce | d. bun | e. food |
| 46. a. driver | b. pounder | c. quarter | d. apple | e. pies |
| 47. a. ice cream | b. machine | c. cream | d. drink | e. cream machine |
| 48. a. cheeseburger | b. double | c. cheap | d. mcdouble | e. double cheeseburger |
| 49. a. wait | b. cheap | c. eat | d. McDonald | e. restaurant |

Bagian mana yang lebih susah Anda mengerti dalam menginterpretasikan topik?

- a. BAGIAN I b. BAGIAN II

— TERIMA KASIH. SEMOGA TUHAN MEMBALAS KEBAIKAN ANDA —

LAMPIRAN D. Hasil Penilaian *Word Intrusion Task*

No.	Nama Responden	NRP	Jumlah Soal Benar		Nilai		Kesulitan	
			I	II	I	II	I	II
1	Diki Enggar S	0611154000027	30	19	71.43	38.78	0	1
2	Rudat Iaina	0611154000058	27	18	64.29	36.73	0	1
3	Rachmat Wahyudi I.	0611154000028	29	25	69.05	51.02	0	1
4	Windya Herieska	0611164000030	24	13	57.14	26.53	0	1
5	Yuda Indra Pratiwi	06111640000135	29	15	69.05	30.61	0	1
6	Adrian Maulana M.	0611154000099	21	8	50.00	16.33	0	1
7	Athyah Danni	0611154000061	28	24	66.67	48.98	0	1
8	Maslahatul Ummah	0611164000032	25	16	59.52	32.65	1	0
9	Fatimah Azzahra	06111640000111	25	14	59.52	28.57	1	0
10	Li'izza Diana Manzil	0611154000049	26	19	61.90	38.78	0	1
12	Alvinda Nisma Yusniar	0611154000003	23	16	54.76	32.65	0	1

--	--	--	--	--	--	--	--	--

Lanjutan Lampiran D

No.	Nama Responden	NRP	Jumlah Soal Benar		Nilai		Kesulitan	
			I	II	I	II	I	II
13	Ajeng Puspitaningrum	06111540000005	25	25	59.52	51.02	1	0
14	Ayu Ni'matul Fitriyah	06111540000013	27	18	64.29	36.73	0	1
15	Al Farizi Akhtiar Daniarta	06111540000117	24	11	57.14	22.45	0	1
16	Arif Fadhilah	06111540007003	27	18	64.29	36.73	0	1
17	Hayu Marta Febrian	06111540000100	23	17	54.76	34.69	0	1
18	Tommy Ferdinand	06111540000019	26	19	61.90	38.78	0	1
19	Lingga Ayu Anggaraini	06111540000050	27	17	64.29	34.69	0	1
20	Anindya R.	06111540000111	26	20	61.90	40.82	0	1
21	Kirana Damayanti	06111640000014	22	16	52.38	32.65	0	1
22	Isabella Anjani Cesarethai	06111540000029	26	22	61.90	44.90	0	1

23	Cindy Chelia	06111540000084	31	21	73.81	42.86	0	1
----	--------------	----------------	----	----	-------	-------	---	---

Lanjutan Lampiran D

No.	Nama Responden	NRP	Jumlah Soal Benar		Nilai		Kesulitan	
			I	II	I	II	I	II
24	Siti Zumrotul Khasanah	06111540000059	25	20	59.52	40.82	0	1
25	Nida Nabilah Nuzul L.	06111540000007	25	17	59.52	34.69	0	1
26	Evika Rachma Yuniasari	06111540000037	25	19	59.52	38.78	0	1
27	Ramadhani Prasyanto	06111540000079	25	22	59.52	44.90	0	1
28	M. Aditya Dharmawan	06111540000104	30	24	71.43	48.98	0	1
29	Izah Amalia	06111540000068	24	24	57.14	48.98	0	1
30	Vira Diana Ulnazilla	06111540000067	27	20	64.29	40.82	0	1
Rata-Rata / Jumlah (Khusus Kesulitan)			26	18.66	61.90	38.10	3	27

LAMPIRAN E. Perhitungan Uji Wilcoxon

No.	Nilai Truncated		$X - \bar{X}$	$Y - \bar{Y}$	$(X - \bar{X})^2$	$(Y - \bar{Y})^2$
	200	1000				
1	71.43	38.78	9.52	0.68	90.70	0.46
2	64.29	36.73	2.38	-1.36	5.67	1.85
3	69.05	51.02	7.14	12.93	51.02	167.06
4	57.14	26.53	-4.76	-11.56	22.68	133.74
5	69.05	30.61	7.14	-7.48	51.02	56.00
6	50.00	16.33	-11.90	-21.77	141.72	473.88
7	66.67	48.98	4.76	10.88	22.68	118.47
8	59.52	32.65	-2.38	-5.44	5.67	29.62
9	59.52	28.57	-2.38	-9.52	5.67	90.70
10	61.90	38.78	0.00	0.68	0.00	0.46
11	66.67	46.94	4.76	8.84	22.68	78.21
12	54.76	32.65	-7.14	-5.44	51.02	29.62
13	59.52	51.02	-2.38	12.93	5.67	167.06
14	64.29	36.73	2.38	-1.36	5.67	1.85
15	57.14	22.45	-4.76	-15.65	22.68	244.81
16	64.29	36.73	2.38	-1.36	5.67	1.85
17	54.76	34.69	-7.14	-3.40	51.02	11.57
18	61.90	38.78	0.00	0.68	0.00	0.46
19	64.29	34.69	2.38	-3.40	5.67	11.57
20	61.90	40.82	0.00	2.72	0.00	7.40
21	52.38	32.65	-9.52	-5.44	90.70	29.62
22	61.90	44.90	0.00	6.80	0.00	46.28
23	73.81	42.86	11.90	4.76	141.72	22.68
24	59.52	40.82	-2.38	2.72	5.67	7.40

Lanjutan Lampiran E

No.	Nilai Truncated		$X - \bar{X}$	$Y - \bar{Y}$	$(X - \bar{X})^2$	$(Y - \bar{Y})^2$
	200	1000				
25	59.52	34.69	-2.38	-3.40	5.67	11.57
27	59.52	44.90	-2.38	6.80	5.67	46.28
28	71.43	48.98	9.52	10.88	90.70	118.47
29	57.14	48.98	-4.76	10.88	22.68	118.47
30	64.29	40.82	2.38	2.72	5.67	7.40
25	59.52	34.69	-2.38	-3.40	5.67	11.57
N	30	30	SUM		941.04	2035.26
Mean	61.9	38.10				
df	58.0					

Uji dengan alternatif Minitab, menghasilkan p_{value} sama dengan perhitungan manual, yakni 0.

Wilcoxon Signed Rank Test: selisih

Test of median = 0.000000 versus median > 0.000000

	N	for Test	Wilcoxon Statistic	P	Estimated Median
selisih	30	30	465.0	0.000	23.81

Ketika p_{value} kurang dari α maka H_0 ditolak.

Sehingga dalam kasus ini H_0 pada $\alpha = 0,05$ ditolak dan benar bahwa *truncated SVD* 200 lebih daripada *truncated SVD* 1000 secara rata-rata.

LAMPIRAN F. Tabel Pengamatan Uji ANOVA

Eksperimen	Truncated		Jumlah Baris	
	200	1000		
Perulangan	1	71.43	38.78	110.20
	2	64.29	36.73	101.02
	3	69.05	51.02	120.07
	4	57.14	26.53	83.67
	5	69.05	30.61	99.66
	6	50.00	16.33	66.33
	7	66.67	48.98	115.65
	8	59.52	32.65	92.18
	9	59.52	28.57	88.10
	10	61.90	38.78	100.68
	11	66.67	46.94	113.61
	12	54.76	32.65	87.41
	13	59.52	51.02	110.54
	14	64.29	36.73	101.02
	15	57.14	22.45	79.59
	16	64.29	36.73	101.02
	17	54.76	34.69	89.46
	18	61.90	38.78	100.68
	19	64.29	34.69	98.98
	20	61.90	40.82	102.72
	21	52.38	32.65	85.03
	22	61.90	44.90	106.80
	23	73.81	42.86	116.67
	25	59.52	34.69	94.22
	26	59.52	38.78	98.30

Lanjutan Lampiran F

Eksperimen		Truncated		Jumlah Baris
		200	1000	
Perulangan	27	59.52	44.90	104.42
	28	71.43	48.98	120.41
	29	57.14	48.98	106.12
	30	64.29	40.82	105.10
Jumlah Kolom		1857.14	1142.86	3000.00

$$FK = \frac{3000^2}{60} = \frac{9000000}{60} = 150000$$

$$SS_T = (71,43^2 + 38,78^2 + \dots + 40,82^2 + 105,10^2) - 150000 \\ = 11479,70$$

$$SS_P = \frac{(110,20^2 + 101,02^2 + \dots + 106,12^2 + 105,10^2)}{30} \\ - 150000 = 8503,40$$

$$SS_E = 11479,70 - 8503,40 = 2976,30$$

$$MS_P = \frac{8503,40}{2 - 1} = 8503,40$$

$$MS_E = \frac{2976,30}{58} = 51,31$$

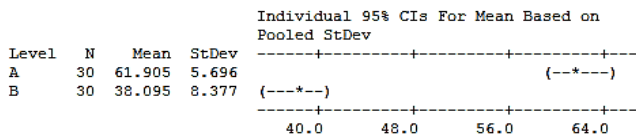
$$F - \text{measure} = \frac{8503,40}{51,31} = 165,70$$

Pengujian dengan menggunakan Minitab menghasilkan Tabel ANOVA yang hampir sama dengan perhitungan manual.

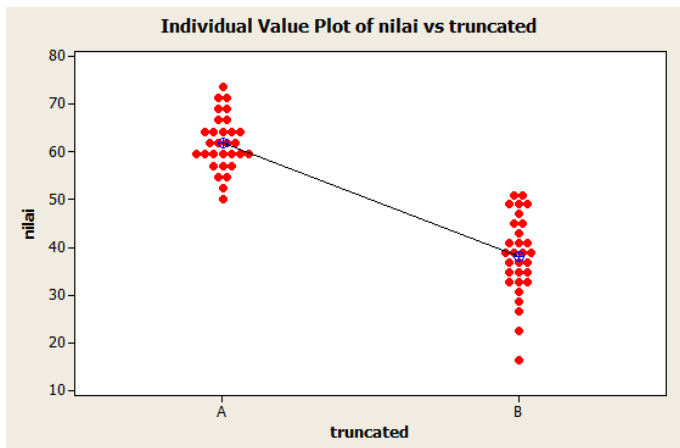
One-way ANOVA: nilai versus truncated

Source	DF	SS	MS	F	P
truncated	1	8503.4	8503.4	165.71	0.000
Error	58	2976.3	51.3		
Total	59	11479.7			

$\bar{s} = 7.163$ $R-Sq = 74.07\%$ $R-Sq(adj) = 73.63\%$



Adapun hasil plot data individu adalah sebagai berikut:



BIODATA PENULIS



Penulis memiliki nama lengkap Riko Wijayanto. Penulis dilahirkan pada tanggal 15 April 1997 di Kabupaten Tulungagung, Jawa Timur. Penulis adalah anak tunggal dari Alm. Gopar dan Sutarmi. Penulis telah menempuh sekolah formal sejak usia lima tahun, yakni di RA Al Hikmah Doroampel (2002-2003), SDN I Doroampel (2003-2009), SMPN 3 Tulungagung (2009-2012), dan SMAN 1 Boyolangu Tulungagung (2012-2015). Penulis menempuh pendidikan Strata-1 di Departemen Matematika ITS dengan NRP 0611154000010 melalui jalur SNMPTN tahun 2015-2019. Penulis mengambil konsentrasi minat pada bidang campuran, yakni Ilmu Komputer dan juga ROPD (Riset Operasi dan Pengolahan Data). Selain kuliah, penulis juga aktif dalam beberapa kegiatan organisasi, yakni sebagai staf *Internal Affair* HIMATIKA ITS (2016-2017), staf Eksternal LMB ITS (2016-2017), staff Pengembangan Sumber Daya Mahasiswa Ibnu Muqalah (2016-2017), dan juga di PSDM KSR PMI ITS (2016-2017). Di tahun berikutnya penulis juga aktif sebagai staf ahli di *Media and Information* HIMATIKA ITS (2017-2018), staf magang Bakor Pemandu ITS (2017-2018), dan Ketua Divisi Hubungan Luar di BEM FMKSD ITS (2018). Penulis juga aktif dalam beberapa kegiatan kemahasiswaan, seperti Penanggung Jawab Regional OMITS 2017, *Steering Committee* LKMW TL FMIPA ITS 2017, *Steering Committee* LOMT HIMATIKA ITS 2017, Fasilitator Konsumsi GERIGI ITS 2016, Mentor GERIGI ITS 2017, Sie Soal OMITS 2018, dan lainnya. Untuk kegiatan berbasis internasionalisasi, penulis pernah mengikuti ITS Goes Global : Singapore Batch 1 tahun

2017, menjadi Tutor Bahasa Indonesia untuk mahasiswa asing saat Commtech, dan juga kegiatan pelatihan lain yang diadakan ITS International Office. Beberapa beasiswa juga pernah penulis dapatkan, salah satunya adalah beasiswa Project READI dari *Waterloo University of Canada* selama satu tahun. Beberapa proyek yang pernah penulis ikuti dan terlibat langsung adalah kajian perpustakaan Pemerintah Kota Surabaya, survei tenaga kerja, pengukuran kemampuan *rewrite* masyarakat Surabaya, dan lainnya. Adapun informasi lebih lanjut terkait laporan Tugas Akhir ini, kritik dan juga saran dapat disampaikan melalui e-mail penulis di mr.rikowijayanto@gmail.com.