



TUGAS AKHIR - KM184801

**ANALISIS SENTIMEN PADA TEKS ULASAN
PELANGGAN *E-COMMERCE* BERDASARKAN
RATING MENGGUNAKAN *N-GRAM* DAN *NEURO-
FUZZY***

**AYU NI'MATUL FITRIYAH
NRP. 0611154000013**

Dosen Pembimbing :
Prof. Dr. Drs. 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

***SENTIMENT ANALYSIS OF CUSTOMER REVIEW
TEXT IN E-COMMERCE BASED ON RATING USING
N-GRAM AND NEURO-FUZZY***

**AYU NI'MATUL FITRIYAH
NRP. 0611154000013**

**Supervisors :
Prof. Dr. Drs. Mohammad Isa Irawan, M.T.
Muhammad Luthfi Shahab S.Si.,M.Si.**

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

LEMBAR PENGESAHAN

ANALISIS SENTIMEN PADA TEKS ULASAN PELANGGAN E-COMMERCE BERDASARKAN RATING MENGUNAKAN N-GRAM DAN NEURO-FUZZY

SENTIMENT ANALYSIS OF CUSTOMER REVIEW TEXT IN E-COMMERCE BASED ON RATING USING N-GRAM AND NEURO-FUZZY

TUGAS AKHIR

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

Oleh :

AYU NI'MATUL FITRIYAH
NRP. 0611154000013

Menyetujui,

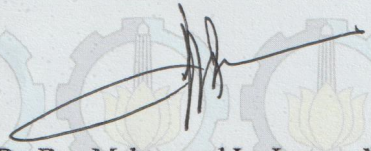
Dosen Pembimbing II,

Dosen Pembimbing I,



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

NIP. 199503312018031001



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

NIP. 196312251989031001

Mengetahui,
Kepala Departemen Matematika FMKSD ITS



Dr. Imam Mukhlash, S.Si, MT

NIP. 19700831 199403 1 003

Surabaya,

Juli 2019

ANALISIS SENTIMEN PADA TEKS ULASAN PELANGGAN *E-COMMERCE* BERDASARKAN RATING MENGGUNAKAN *N-GRAM* DAN *NEURO-FUZZY*

Nama : Ayu Ni'matul Fitriyah
NRP : 0611154000013
Departemen : Matematika
Dosen Pembimbing : 1. Prof. Dr. Drs. Mohammad Isa Irawan,
M.T.
2. Muhammad Luthfi Shahab, S.Si.,M.Si.

ABSTRAK

Perkembangan *e-commerce* di Indonesia cukup pesat bahkan diprediksi akan tumbuh sebesar 144.5% pada tahun 2022. Dalam perkembangannya *e-commerce* memberikan fasilitas pada pelanggan untuk memberikan pendapat mereka mengenai produk yang mereka jual berupa ulasan dan nilainya. Ulasan memiliki pengaruh dalam penentuan keputusan pembelian di masa depan dan memiliki hubungan dengan kepercayaan pelanggan, namun terkadang terjadi kesalahan berupa ketidakcocokan antara nilai dan ulasan yang diberikan dan mengakibatkan nilai ulasan yang tersimpan tidak akurat sehingga diperlukan suatu sistem untuk melakukan prediksi nilai dari suatu teks ulasan agar suatu ulasan memiliki kecocokan dengan nilai yang diberikan. Pada penelitian ini dilakukan prediksi sentimen teks ulasan berdasarkan *rating* secara otomatis menggunakan FCM-ANFIS untuk klasifikasi, penambahan metode *N-gram* pada pra proses data, dan pengurangan dimensi dengan *truncated SVD*. Penggunaan *n-gram* meliputi *unigram*, *bigram*, dan kombinasi *unigram bigram*. Pada pengujian sistem dengan 4 kasus diperoleh akurasi untuk *unigram* 0.657, 0.83 0.919, 0.87 sedangkan *bigram* 0.64, 0.75, 0.79, 0.77 dan kombinasi 0.69, 0.81, 0.929, 0.889. Hasil pengujian ini menunjukkan bahwa penggunaan kombinasi *unigram* dan *bigram* mampu meningkatkan hasil akurasi dari prediksi sentimen teks ulasan.

Kata kunci: ulasan, nilai ulasan, analisis sentimen, *N-Gram*, *Fuzzy C-Means*, *Adaptive Neuro Fuzzy Inference System*

**SENTIMENT ANALYSIS OF CUSTOMER REVIEW TEXT IN
E-COMMERCE BASED ON RATING USING N-GRAM AND
NEURO-FUZZY**

Name : Ayu Ni'matul Fitriyah
NRP : 0611154000013
Department : *Mathematics*
Supervisors : 1. Prof. Dr. Drs. Mohammad Isa Irawan, M.T.
2. Muhammad Luthfi Shahab, S.Si.,M.Si.

ABSTRACT

E-commerce in Indonesia is growing rapidly and projected will increase up to 144,5% in 2020. In its development, e-commerce provides facilities to customer to give their opinions about the product sold by them in the form of review and rating. Reviews have an influence in determining future purchasing and have a relationship with customer trust, but sometimes an error occurs between the rating and review given by customers which affects the stored rating being inaccurate. A system is needed to predict the rating of a review text automatically so the review match with the rating. This research is prediction of textual review sentiment based on rating using FCM-ANFIS for classification model, addition of N-gram in pre-processing data, and using truncated SVD for dimensionality reduction. N-gram that used are unigram, bigram, and combination both of them. The testing scenario use 4 cases that have results unigram 0.657, 0.83 0.919, 0.87 for unigram 0.64, 0.75, 0.79, 0.77 for bigram, and 0.69, 0.81, 0.929, 0.889 for combination both of them. The result indicates that use combination of unigram and bigram is able to improve the accuracy of sentiment prediction of textual review based on rating.

Keywords: *review, rating, sentiment analysis, N-Gram, Fuzzy C-Means, Adaptive Neuro Fuzzy Inference System*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillahirobbil'aalamiin, segala puji dan syukur bagi Allah SWT yang memiliki apa yang ada dilangit dan di bumi yang telah memberikan limpahan rahmat, petunjuk serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

“ANALISIS SENTIMEN PADA TEKS ULASAN PELANGGAN E-COMMERCE BERDASARKAN RATING MENGGUNAKAN N-GRAM DAN NEURO-FUZZY”

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika, Fakultas Matematika, Komputasi, dan Sains Data (FMKSD) Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dari berbagai pihak. Suatu kebahagiaan dan kewajiban bagi penulis untuk menyampaikan rasa terima kasih kepada semua pihak yang telah membantu dan mendukung atas terselesainya Tugas Akhir :

1. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Bapak Prof. Dr. Drs. Mohammad Isa Irawan, M.T. dan Bapak Muhammad Luthfi Shahab, S.Si., M.Si. selaku pembimbing Tugas Akhir yang telah memberikan bimbingan, arahan serta motivasi sehingga Tugas Akhir ini dapat terselesaian.
3. Bapak Dr. Didik Khusnul Arif, S. Si, M. Si selaku Dosen Wali sekaligus Ketua Program Studi S1 Departemen Matematika ITS yang telah memberian arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
4. Bapak Drs. Soetrisno, Ml.Komp, Ibu Dra. Wahyu Fistia Doctoriana, M. Si, dan Bapak Drs. Nurul Hidayat, M.Kom

selaku Dosen Penguji yang telah memberikan saran demi perbaikan Tugas Akhir ini.

5. Seluruh jajaran dan staf Departemen Matematika ITS.
6. Keluarga tercinta yang senantiasa memberikan dukungan dan do'a dengan ikhlas.
7. Keluarga Himpunan Mahasiswa Matematika ITS yang telah berusaha memfasilitasi saya untuk berkembang khususnya untuk teman-teman Kabinet SATU AKSI kepengurusan 2017/2018.
8. Teman-teman angkatan 2015 'M1550' yang saling mendukung dan memotivasi.
9. Semua pihak yang tidak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

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

Wassalamu'alaikum Wr. Wb.

Surabaya, Juli 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
<i>ABSTRACT</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu	7
2.2 <i>E-Commerce</i>	8
2.3 Ulasan	9
2.4 Analisis Sentimen	9
2.5 <i>Natural Language Processing (NLP)</i>	9
2.5.1 Tokenisasi	10
2.5.2 Filtrasi	10
2.5.3 <i>Stemming</i>	11
2.5.4 Pembobotan TF-IDF	11
2.6 <i>Metode N-Gram</i>	12

2.7	<i>Latent Semantic Indexing (LSI)</i>	12
2.8	<i>Fuzzy C-Means (FCM)</i>	14
2.9	<i>Adaptive Neuro Fuzzy Inference System (ANFIS)</i>	16
2.10	Proses Pembelajaran dengan RLSE	18
2.11	Proses Pembelajaran Propagasi Error	20
2.12	Ukuran Evaluasi Sistem	21
BAB III METODOLOGI PENELITIAN		25
3.1	Pengumpulan Data	25
3.2	Praproses Data	25
3.3	Ekstraksi Fitur	25
3.4	Implementasi FCM dan ANFIS	26
3.5	Pengujian dan Evaluasi	26
3.6	Penarikan Kesimpulan	26
BAB IV PERANCANGAN DAN IMPLEMENTASI		29
4.1	Pengumpulan Data	29
4.2	Praproses Data	32
4.2.1	Tokenisasi	33
4.2.2	<i>Spelling Normalization</i>	34
4.2.3	Filtrasi	36
4.2.4	<i>Stemming</i>	37
4.3	Pembangkitan <i>N-Gram</i>	39
4.4	Pembentukan Matriks <i>Term-by-doc</i>	42
4.5	Pembobotan TF-IDF	44
4.6	<i>Latent Semantic Indexing</i>	48
4.7	<i>Clustering</i> dengan <i>Fuzzy C-Means</i>	51
4.8	<i>Adaptive Neuro Fuzzy Inference System (ANFIS)</i>	53

4.9	Analisis Implementasi Sistem.....	59
4.9.1	<i>Use Case Diagram</i>	59
4.9.2	<i>Activity Diagram</i>	60
4.10	Tampilan Antar Muka GUI.....	61
BAB V HASIL DAN PEMBAHASAN		63
5.1	Deskripsi Data.....	63
5.2	Praproses Data	65
5.3	Hasil <i>N-Gram</i> dan TF-IDF.....	65
5.4	Hasil <i>Feature Extraction</i> dengan LSI	69
5.5	Hasil Prediksi Ulasan dengan FCM-ANFIS	71
BAB VI KESIMPULAN DAN SARAN		79
6.1	Kesimpulan	79
6.2	Saran	80
DAFTAR PUSTAKA.....		81
LAMPIRAN 1		83
LAMPIRAN 2		85
LAMPIRAN 3		91
LAMPIRAN 4		93
LAMPIRAN 5		99
LAMPIRAN 6		103
BIODATA PENULIS.....		105

DAFTAR GAMBAR

Gambar 2. 1 <i>Truncated</i> SVD.....	13
Gambar 2. 2 Arsitektur Jaringan ANFIS	17
Gambar 2. 3 <i>Confusion Matrix</i> Model Klasifikasi	22
Gambar 3. 1 Diagram Alir Metode Penelitian.....	27
Gambar 4. 1 Contoh <i>dictionary spelling normalization</i>	35
Gambar 4. 2 Matriks <i>term-by-doc Unigram</i>	43
Gambar 4. 3 Matriks <i>term-by-doc</i> Kombinasi	43
Gambar 4. 4 Matriks <i>doc-by-term Hasil Pembobotan</i>	47
Gambar 4. 5 Plot Nilai Singular Matriks Σ	50
Gambar 4. 6 Matriks <i>doc-by-dimension Hasil Truncated</i>	51
Gambar 4. 7 Arsitektur ANFIS pada Data Contoh.....	54
Gambar 4. 8 Plot RMSE Setiap <i>Epoch</i>	58
Gambar 4. 9 <i>Use Case Diagram</i> Sistem.....	59
Gambar 4. 10 <i>Activity Diagram</i> Sistem Untuk Prediksi Rating	60
Gambar 4. 11 Tampilan Awal GUI	61
Gambar 4. 12 Tampilan Tab Menu Prediksi <i>Rating</i>	62
Gambar 5. 1 Gambar Kutipan Data Ulasan	63
Gambar 5. 2 Prosentase Tiap Rating dalam Data Ulasan Produk....	64
Gambar 5. 3 Kutipan Hasil Praproses Data	65
Gambar 5. 4 Kutipan Fitur Terpilih Hasil Proses TF-IDF <i>Unigram</i> 66	
Gambar 5. 5 Kutipan Matriks TDM <i>Unigram</i>	67
Gambar 5. 6 Kutipan Fitur Terpilih Hasil Proses TF-IDF <i>Bigram</i> ..	67
Gambar 5. 7 Kutipan Matriks TDM <i>Bigram</i>	68
Gambar 5. 8 Kutipan Fitur Terpilih Hasil Proses TF-IDF <i>Bigram</i> ..	68
Gambar 5. 9 Kutipan Matriks TDM Kombinasi.....	69
Gambar 5. 10 Plot Rasio Nilai Eigen Skenario <i>Unigram</i>	69
Gambar 5. 11 Plot Rasio Nilai Singular Skenario <i>Bigram</i>	70
Gambar 5. 12 Plot Rasio Nilai Singular Skenario Kombinasi.....	70
Gambar 5. 13 Hasil Akurasi Pembelajaran Skenario <i>Unigram</i>	71
Gambar 5. 14 Hasil Akurasi Pembelajaran Skenario <i>Bigram</i>	73
Gambar 5. 15 Hasil Akurasi Pembelajaran Skenario Kombinasi	75
Gambar 5. 16 Perbandingan Performansi Skenario Toleransi 0.....	76
Gambar 5. 17 Perbandingan Performansi Skenario Toleransi 1	77
Gambar 5. 18 Perbandingan Performansi Skenario Sentimen 1	77
Gambar 5. 19 Perbandingan Performansi Skenario Sentimen 2.....	78

DAFTAR TABEL

Tabel 4. 1 Bentuk Dataset Ulasan Tokopedia	31
Tabel 4. 2 Contoh Data Inputan	32
Tabel 4. 3 <i>Array</i> Ulasan Hasil Tokenisasi	34
Tabel 4. 4 <i>Array</i> Ulasan Hasil <i>Spelling Normalization</i>	36
Tabel 4. 5 <i>Array</i> Ulasan Hasil Filtrasi Data	38
Tabel 4. 6 <i>Array</i> Ulasan Hasil Proses <i>Stemming</i>	39
Tabel 4. 7 Hasil <i>Array</i> Ulasan dengan <i>Bigram</i>	40
Tabel 4. 8 Hasil <i>Array</i> Ulasan dengan Kombinasi	41
Tabel 4. 9 Tabel TF Ulasan Pertama	44
Tabel 4. 10 Tabel IDF Ulasan Pertama	45
Tabel 4. 11 Tabel TF-IDF Ulasan Pertama	46
Tabel 4. 12 Tabel TF-IDF Hasil Normalisasi Ulasan Pertama	47
Tabel 4. 13 Hasil <i>Cluster</i> Ulasan	53
Tabel 4. 14 Nilai Mean Setiap Atribut pada <i>Cluster</i>	54
Tabel 4. 15 Nilai Deviasi Standar Setiap Atribut pada <i>Cluster</i>	54
Tabel 4. 16 <i>Output</i> Lapisan 1	55
Tabel 4. 17 <i>Output</i> Lapisan 2	56
Tabel 4. 18 <i>Output</i> Lapisan 3	56
Tabel 4. 19 Koefisien Parameter <i>w_{ixij}</i>	57
Tabel 5. 1 Rincian Jumlah Ulasan Tiap Produk	64
Tabel 5. 2 Hasil Performansi Skenario <i>Unigram</i>	72
Tabel 5. 3 Ulasan yang Salah Prediksi	72
Tabel 5. 4 Hasil Performansi Skenario <i>Unigram</i> Sentimen	73
Tabel 5. 5 Hasil Performansi Skenario <i>Bigram</i>	74
Tabel 5. 6 Hasil Performansi Skenario <i>Bigram</i> Sentimen	74
Tabel 5. 7 Hasil Performansi Skenario Kombinasi	75
Tabel 5. 8 Hasil Performansi Skenario Kombinasi Sentimen	76

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang melatarbelakangi munculnya permasalahan dari penelitian pada ini. Selanjutnya permasalahan tersebut disusun menjadi rumusan masalah yang akan dikerjakan. Kemudian dibuat beberapa batasan masalah untuk membatasi cakupan penelitian dan penentuan tujuan penelitian berdasarkan rumusan masalah yang telah dibuat.

1.1 Latar Belakang

Perkembangan teknologi memiliki pengaruh yang besar terhadap proses jual beli yang mana sebelumnya bersifat konvensional berubah ke arah modern. Hal inilah yang mendorong lahirnya perdagangan elektronik dengan memanfaatkan internet atau biasa disebut *e-commerce*. Pertumbuhan *e-commerce* di Indonesia meningkat cukup pesat dari tahun ke tahun. Berdasarkan survei Databoks 2018, penjualan *e-commerce* di Indonesia mencapai US\$ 5,8 miliar pada tahun 2016 dan US\$ 7,1 miliar pada 2017 serta diperkirakan akan tumbuh sebesar 133.5 % menjadi US\$ 16,5 miliar atau sekitar Rp 219 triliun pada 2022¹.

Dalam perkembangannya *e-commerce* telah memberikan fasilitas kepada pelanggannya untuk memberikan pendapat mereka mengenai produk atau pelayanan yang diberikan melalui ulasan serta penilaiannya secara numerik. Pendapat ini memiliki peranan yang penting dalam pembuatan keputusan pelanggan untuk pembelian di masa mendatang. Ulasan produk juga berguna untuk para penjual dalam memperbaiki strategi pemasaran dan perbaikan produk. Berdasarkan data dalam laporan *Global Online Consumer Report* 2017 [1], presentase pelanggan yang melihat ulasan online tentang

¹ Statista. 2018. **2022 Penjualan e-Commerce Indonesia Mencapai US\$ 16 Miliar**. katadata.co.id. [Online]. Tersedia : <https://databoks.katadata.co.id/datapublish/2018/02/12/2022-penjualan-e-commerce-indonesia-mencapai-rp-16-miliar>. (diakses 1 Februari 2019)

produk saat berbelanja adalah 49%. Selain itu dalam laporan ini juga disebutkan bahwa kepercayaan pada ulasan online dibandingkan terhadap informasi yang tersedia pada situs web perusahaan di beberapa benua yaitu Asia, Eropa Timur dan Rusia relatif lebih tinggi. Hal ini membuktikan bahwa ulasan dan penilaian dalam *e-commerce* memiliki hubungan terhadap minat pembelian dan menjadi salah satu fitur penting untuk meningkatkan kepercayaan pelanggan. Namun sering kali beberapa pelanggan memberikan nilai ulasan berkebalikan dengan teks ulasan yang dituliskan, hal ini mengakibatkan tidak akuratnya nilai ulasan suatu produk yang tersimpan. Sedangkan dalam prosesnya hal pertama yang dilihat oleh pelanggan dari suatu produk adalah nilainya, dimana nilai yang tersimpan merupakan akumulasi dari nilai-nilai ulasan yang telah diberikan.

Penting dalam tulisan pada suatu ulasan memiliki kecocokan dengan penilaian yang diberikan pelanggan secara numerik dengan tujuan agar sistem yang dimiliki konsisten. Dengan membuat prediksi penilaian secara numerik dari hasil ulasan yang diberikan diharapkan dapat meningkatkan akurasi data nilai ulasan yang tersimpan. Prediksi nilai suatu ulasan dilakukan dengan mengekstraksi informasi yang berguna dari ulasan pengguna yang disebut **analisis sentimen**. Analisis sentimen atau disebut juga *opinion mining* adalah bidang kajian untuk menganalisis pendapat, sentimen, evaluasi, sikap, dan emosi dari orang-orang terhadap suatu entitas seperti produk, layanan, organisasi, individu, permasalahan, peristiwa, topik dan atribut-atributnya [2]. Komponen utama dalam analisis sentimen adalah *Natural Language Processing* (NLP) yang mana berkaitan dengan perancangan dan pembangunan aplikasi dan sistem yang memungkinkan interaksi antara mesin dan bahasa alami yang dikembangkan oleh manusia [3]. Teknik NLP memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan menggunakannya lebih lanjut untuk diambil manfaatnya. Pendekatan menggunakan logika fuzzy merupakan salah satu metode yang dapat digunakan dalam proses analisis sentimen. Penyelesaian permasalahan semantik dari analisis sentimen menggunakan

pendekatan fuzzy akan meningkatkan proses ekstraksi, peringkasan dan penyajian dari opini-opini dengan bobot dan kekuatannya karena kemampuannya untuk menyelesaikan permasalahan yang memiliki parameter masukan berupa himpunan fuzzy menjadi nilai real atau *crisp* terdekat [4].

Pada penelitian Thiel yang berjudul *sentiment analysis with N-Gram* [5] menggunakan metode *N-Gram* dalam klasifikasi dokumen dengan sentimen positif dan negatif disebutkan bahwa *term* tunggal positif belum tentu sepenuhnya positif begitu pula sebaliknya *term* tunggal negatif belum tentu sepenuhnya negatif. Misalkan kata “buruk” merupakan *term* negatif namun “tidak buruk” merupakan *term* positif. Dengan menggunakan metode *N-Gram* dalam pemecahan kata dari teks ulasan produk diharapkan mampu meningkatkan akurasi dari prediksi nilai ulasan.

Penelitian mengenai prediksi nilai ulasan sebelumnya telah dilakukan oleh Georgina Cosma dan Giovanni Acampora [6] pada tahun 2016. Data yang digunakan merupakan data ulasan berbahasa inggris yang dipublikasikan secara terbuka untuk penelitian oleh Standford. Metode yang digunakan adalah *Singular Value Decomposition* (SVD) untuk mereduksi ukuran data dan pendekatan *Neuro-Fuzzy* yaitu *Fuzzy C-Means* (FCM) dan *Adaptive Neuro Fuzzy Inference System* (ANFIS) untuk memprediksi nilai ulasan. Pada penelitian dilakukan perbandingan antara metode FCM-ANFIS, FCM, *Artificial Neural Network*, dan *Support Vector Machine*, dimana hasilnya menunjukkan bahwa FCM-ANFIS memiliki nilai akurasi tertinggi yaitu 0.886.

Berdasarkan uraian diatas, pada Tugas Akhir ini dilakukan penelitian mengenai analisis sentimen untuk menentukan nilai dari teks ulasan secara otomatis yang diberikan oleh pelanggan *e-commerce*. Metode yang akan digunakan dalam penelitian ini adalah *N-Gram* dan *Neuro-Fuzzy*. Metode *Neuro-Fuzzy* yang digunakan adalah algoritma *Fuzzy C-Means* (FCM) dan *Adaptive Neuro-Fuzzy Inference System* (ANFIS).

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka rumusan masalah dari penelitian sebagai berikut :

1. Bagaimana melakukan analisis sentimen pada teks ulasan produk *e-commerce* berdasarkan *rating* menggunakan *N-gram* dan FCM-ANFIS?
2. Bagaimana analisis dari hasil proses analisis sentimen teks ulasan produk *e-commerce* berdasarkan *rating* menggunakan *N-gram* dan FCM-ANFIS?

1.3 Batasan Masalah

Batasan masalah yang ditentukan untuk menghindari perluasan dalam penelitian ini sebagai berikut :

1. Data yang digunakan merupakan data ulasan produk elektronik dari berbagai toko di situs www.tokopedia.com.
2. Ulasan yang digunakan berupa kalimat berbahasa Indonesia.
3. Ulasan memiliki skala *rating* 1-5.
4. Bahasa pemrograman yang digunakan adalah *python*

1.4 Tujuan

Tujuan yang ingin dicapai dari penelitian ini sebagai berikut :

1. Menerapkan analisis sentimen pada teks ulasan produk *e-commerce* berdasarkan *rating* menggunakan *N-gram* dan FCM-ANFIS.
2. Menganalisis hasil dari analisis sentimen pada teks ulasan produk *e-commerce* berdasarkan *rating* menggunakan *N-gram* dan FCM-ANFIS

1.5 Manfaat

Adapaun manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut :

1. Membantu pihak *e-commerce* untuk meningkatkan akurasi data nilai ulasan produk yang tersimpan.

2. Memberikan gambaran kepada pihak e-commerce mengenai sistem penilaian teks ulasan secara otomatis yang dapat dimanfaatkan sebagai contoh untuk verifikasi ulasan ataupun untuk melakukan *scoring* terhadap produk.

1.6 Sistematika Penulisan

Tugas akhir ini mempunyai susunan dalam penulisannya agar tersusun secara sistematis. Berikut sistematika penulisan Tugas Akhir ini :

1. BAB I PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang melatarbelakangi munculnya permasalahan dari penelitian ini. Selanjutnya permasalahan tersebut disusun menjadi rumusan masalah yang akan dikerjakan. Kemudian dibuat beberapa batasan masalah untuk membatasi cakupan penelitian dan penentuan tujuan penelitian berdasarkan rumusan masalah yang telah dibuat.

2. BAB II TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa teori dasar untuk mendukung proses pengerjaan pada penelitian ini yang meliputi penelitian terdahulu, *Natural Language Processing* (NLP), *N-Gram*, *Latent Semantic Indexing* (LSI), *Fuzzy C-Means* (FCM), dan *Adaptive Neuro Fuzzy Inference System* (ANFIS), Proses Pembelajaran dengan RLSE dan Pembelajaran Propagasi Error.

3. BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahap-tahap pengerjaan dalam menyelesaikan penelitian ini dapat dilakukan secara sistematis dan diatur sebaik-baiknya.

4. BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan perancangan desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Perancangan

sistem menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga pembuatan model FCM-ANFIS. Langkah-langkah dalam pengimplementasian sistem berdasarkan desain sistem yang sudah dibentuk. Adapun ilustrasi dalam penjelasan tahapan akan menggunakan data contoh untuk memudahkan dalam proses interpretasi.

5. BAB V HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dan analisis terhadap hasil yang diperoleh dari proses implementasi penelitian pada data sebenarnya.

6. BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan dari semua proses penelitian yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

BAB II

TINJAUAN PUSTAKA

Pada bab ini dijelaskan beberapa teori dasar untuk mendukung proses pengerjaan pada penelitian ini yang meliputi penelitian terdahulu, *Natural Language Processing* (NLP), *N-Gram*, *Latent Semantic Indexing* (LSI), *Fuzzy C-Means* (FCM), dan *Adaptive Neuro Fuzzy Inference System* (ANFIS), Proses Pembelajaran dengan RLSE dan Pembelajaran Propagasi Error.

2.1 Penelitian Terdahulu

Berdasarkan penelitian sebelumnya, Georgina Cosma dan Giovanni Acampora [6] melakukan penelitian pada tahun 2016 tentang prediksi nilai ulasan berbahasa inggris pada *e-commerce* menggunakan data yang dipublikasikan secara terbuka untuk penelitian oleh Standford. Metode yang digunakan dalam penelitian ini adalah *Singular Value Decomposition* (SVD) untuk mereduksi ukuran data dan pendekatan *Neuro-Fuzzy* yaitu *Fuzzy C-Means* (FCM) dan *Adaptive Neuro Fuzzy Inference System* (ANFIS) untuk memprediksi nilai ulasan. Hasil dari penelitian dengan menggunakan data berturut-turut sejumlah 30.000, 40.000, dan 80.000 ulasan menunjukkan metode FCM-ANFIS memiliki rata-rata akurasi yang paling tinggi yaitu 0.886 jika dibandingkan dengan metode FCM, *Artificial Neural Network*, dan *Support Vector Machine* berturut-turut 0.872, 0.885, dan 0.764.

Pada tahun 2017 Wararat Songpan [7] melakukan penelitian mengenai analisis dan prediksi nilai ulasan pelanggan menggunakan *opinion mining*. Penelitian ini membandingkan penggunaan metode klasifikasi *Naïve Bayes* dan *Decision Tree* (C4.5). Dalam tahap pra-proses data dilakukan seleksi fitur untuk mencari kata negatif dan positif yang paling sering muncul untuk kemudian di analisis dan digunakan sebagai model klasifikasi. Dalam pengujiannya dilakukan pembagian menjadi 3 untuk jumlah kata yang digunakan sebagai model klasifikasi yaitu menggunakan 10 kata, 20 kata, dan 36 kata

dimana setiap pengujian berisi 50% kata positif dan 50% kata negatif. Hasil dari penelitian ini menunjukkan bahwa rata-rata akurasi algoritma *Naïve Bayes* lebih baik dibandingkan *Decision Tree*(C4.5) yaitu 93.61% dan 92.84%. Dengan hasil akurasi terbaik pada saat penggunaan 20 kata sebagai atribut untuk model klasifikasi.

Iswanto dan Poerwoto pada tahun 2018 [8] melakukan penelitian yang berjudul *Sentiment analysis on Bahasa Indonesia tweets using Unigram models and machine learning techniques*, dimana dalam penelitian ini dilakukan proses analisis sentimen terhadap *tweet* berbahasa Indonesia dengan berbagai metode *machine learning* yaitu *Naïve Bayes*, *Maximum Entropy*, dan *Support Vector Machines*. Dalam penelitian ini dilakukan percobaan menggunakan pemecahan kata dari *tweets* menggunakan metode *N-Gram* yaitu *unigram*, *bigram*, dan kombinasi *unigram-bigram*. Hasilnya menunjukkan bahwa penggunaan *unigram-bigram* memiliki dampak yang baik untuk meningkatkan nilai performansi sistem dibandingkan penggunaan *unigram* dengan peningkatan sebesar 14.41% untuk *Naïve Bayes*, 16.34% untuk *Maximum Entropy*, dan 5.07% untuk *Support Vector Machine*.

2.2 E-Commerce

Electronic Commerce atau *E-Commerce* tidak hanya berkaitan dengan pemesanan produk dari katalog *online*. *E-Commerce* bermakna penggunaan jaringan computer untuk meningkatkan kinerja organisasi berupa meningkatkan keuntungan, mendapatkan pangsa pasar, meningkatkan layanan pelanggan, dan mengirimkan produk dengan lebih cepat. Perdagangan elektronik melibatkan penggunaan teknologi informasi untuk meningkatkan transaksi dengan semua pemangku kepentingan yang meliputi pelanggan, pemasok, pemerintah, lembaga keuangan, manajer, karyawan dan masyarakat luas [9].

2.3 Ulasan

Ulasan dalam Kamus Besar Bahasa Indonesia memiliki makna kupasan, tafsiran, komentar. Menurut Mudambi dan Schuff [10] dalam penelitiannya yang berjudul *Customer Reviews on Amazon.com* bahwa ulasan pelanggan secara *online* dapat didefinisikan sebagai hasil evaluasi produk yang ditempatkan pada situs web perusahaan ataupun pihak ketiga. Situs web ritel memberikan kesempatan kepada pelanggan untuk mengirimkan ulasan produk dalam bentuk nilai bintang (biasanya berkisar antara 1 sampai 5) dan teks komentar.

2.4 Analisis Sentimen

Sentimen dalam Kamus Besar Bahasa Indonesia memiliki makna pendapat atau pandangan yang didasarkan pada perasaan yang berlebih-lebihan terhadap sesuatu atau juga dapat bermakna sebagai emosi yang berlebihan. Analisis sentimen atau yang juga disebut *opinion mining* adalah proses untuk menganalisa percakapan mengenai suatu peristiwa, topik atau sebuah produk berdasarkan sistem yang secara otomatis mengelola proses ini. Secara umum, sentimen analisis dibagi menjadi 3 level [2]:

Document Level : Tugas pada level ini adalah untuk mengklasifikasikan ungkapan dari dokumen yang berisi opini-opini apakah memiliki sentimen negatif atau positif .

Sentiment Level : Tugas pada level ini adalah untuk menentukan apakah sebuah kalimat mengungkapkan opini yang positif, negatif, atau netral.

Entity and Aspect Level : Tujuan analisis pada level ini adalah untuk menemukan sentimen pada entitas dan/atau aspeknya. Pada tingkat analisis ini dapat dihasilkan ringkasan opini yang terstruktur tentang entitas dan aspek-aspek yang dihasilkan yang dapat digunakan untuk semua jenis analisis baik kualitatif maupun kuantitatif.

2.5 *Natural Language Processing (NLP)*

NLP adalah bidang khusus dalam ilmu komputer dan kecerdasan buatan yang berakar pada komputasi linguistik. Hal ini terutama

berkaitan dengan perancangan dan pembangunan aplikasi dan sistem yang memungkinkan interaksi antara mesin dan bahasa alami yang dikembangkan oleh manusia. Teknik NLP memungkinkan komputer untuk memproses dan memahami bahasa alami manusia dan menggunakannya lebih lanjut untuk diambil manfaatnya. Dalam prosesnya NLP memiliki berbagai tahapan untuk pengolahannya. Beberapa diantaranya adalah tokenisasi, filtrasi, *stemming*, dan pembobotan TF-IDF [3].

2.5.1 Tokenisasi

Tokenisasi adalah proses pemisahan atau pengelompokan kalimat menjadi kata-kata pokok. Kalimat adalah sekumpulan kata dan dengan tokenisasi kalimat dapat dibaginya menjadi daftar yang dapat digunakan untuk merekonstruksi kalimat. Tokenisasi sangat penting dalam banyak proses terutama dalam proses filtrasi dan *stemming* karena kedua proses tersebut bekerja dalam satuan kata [3]. Sebagai contoh perhatikan ilustrasi berikut :

Kalimat : "Manajemen pengetahuan adalah sebuah konsep baru di dunia bisnis"

Hasil : 'manajemen', 'pengetahuan', 'adalah', 'sebuah', 'konsep', 'baru', 'di', 'dunia', 'bisnis'

2.5.2 Filtrasi

Filtrasi merupakan tahap pengambilan kata-kata penting dari hasil token. Salah satunya dengan *stopword removal*. *Stopword* adalah kata-kata yang memiliki arti sedikit atau bahkan tidak memiliki arti. Biasanya *stopword* akan dihapus dari teks selama pemrosesan untuk mempertahankan kata-kata yang memiliki pengaruh dan konteks maksimum. Tidak ada daftar *stopword* yang lengkap atau universal. Setiap domain atau bahasa mungkin memiliki kumpulan *stopword* masing-masing [3]. Sebagai contoh perhatikan ilustrasi berikut :

Kalimat : "Manajemen pengetahuan adalah sebuah konsep baru di dunia bisnis"

Hasil : 'manajemen', 'pengetahuan', 'konsep', 'baru', 'dunia', 'bisnis'

2.5.3 Stemming

Stemming adalah proses pengembalian kata dasar yang sudah mendapatkan imbuhan. Sebagai contoh kata pukul-pukulan dan memukul memiliki kata yang sama yaitu pukul. Algoritma *stemming* telah dikembangkan berdasarkan struktur morfologi Bahasa Indonesia yaitu dengan penghapusan prefiks, suffiks, dan infiks [11]. Sebagai contoh perhatikan ilustrasi berikut :

- membetulkan → betul
- berpegangan → pegang

2.5.4 Pembobotan TF-IDF

Pembobotan TF-IDF atau *Term Frequent-Inverse Document Frequency* memungkinkan untuk menangkap informasi mengenai pentingnya suatu kata untuk mendeskripsikan setiap kalimat. TF-IDF merupakan hasil perkalian antara dua matriks yang direpresentasikan dengan $tfidf = tf \times idf$ [3]. *Term frequency* adalah frekuensi *term* atau kata pada sebuah dokumen. Persamaan (2.1) dapat dinyatakan bahwa f_{wD} merupakan frekuensi kata w di dokumen D .

$$tf(w, D) = f_{wD} \quad (2.1)$$

Inverse document frequency atau *idf* adalah invers dari frekuensi dokumen dalam setiap kata. Persamaan (2.2) merupakan persamaan dari *idf* dari *term* t , C menunjukkan jumlah dari total seluruh dokumen, dan $df(t)$ menunjukkan frekuensi jumlah dokumen yang di dalamnya terdapat *term* t .

$$idf(t) = 1 + \log \frac{C}{1 + df(t)} \quad (2.2)$$

Selanjutnya pembobotan *tf-idf* dapat dihitung dengan mengalikan nilai *tf* dan *idf* yang telah dihitung. Selanjutnya nilainya akan di normalisasi menggunakan persamaan (2.3).

$$tfidf(i, t) = \frac{tf_i \times idf_t}{\sum_{i=1}^n tf_i \times idf_t}, n = \text{banyak ulasan} \quad (2.3)$$

2.6 Metode N-Gram

N-gram adalah potongan sejumlah n kata dari sebuah kalimat. *N-gram* merupakan metode yang diaplikasikan untuk pembangkitan kata atau karakter. Metode ini digunakan untuk mengambil potongan-potongan kata sejumlah n dari sebuah kalimat yang secara kontinuitas dari teks sumber hingga akhir dokumen. *N-gram* dibedakan berdasarkan jumlah potongan karakter sebesar n [3]. Untuk membantu dalam mengambil potongan-potongan kata berupa karakter huruf tersebut, maka dilakukan *padding* dengan *blank* diawal dan diakhir suatu kata. Berikut ilustrasi penggunaan *N-Gram* pada kalimat “aku suka banget pada produk cleanser ini”:

Unigram : ‘aku’, ‘suka’, ‘banget’, ‘produk’, ‘cleanser’, ‘ini’.

Bigram : ‘aku suka’, ‘suka banget’, ‘banget produk’, ‘produk cleanser’, ‘cleanser ini’

Kombinasi : ‘aku’, ‘suka’, ‘banget’, ‘produk’, ‘cleanser’, ‘ini’, ‘aku suka’, ‘suka banget’, ‘banget produk’, ‘produk cleanser’, ‘cleanser ini’.

2.7 Latent Semantic Indexing (LSI)

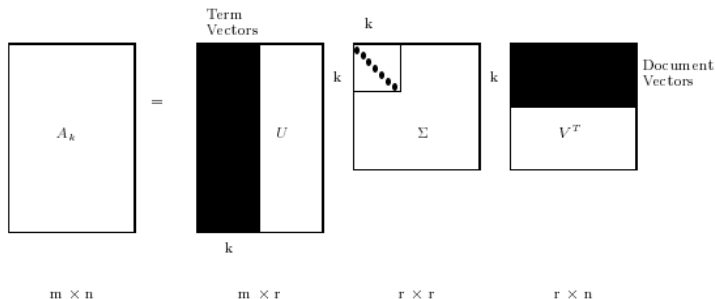
LSI adalah teknik dalam NLP khususnya untuk menganalisis hubungan antara seperangkat dokumen dan istilah-istilah yang ada didalamnya dengan menghasilkan serangkaian konsep yang berkaitan dengan dokumen dan istilah tersebut. Metode yang digunakan adalah *Singular Value Decomposition* (SVD). Misalkan diberikan matriks A berukuran $m \times n$ dan $rank(A) = r$. SVD dari A didefinisikan sebagai [12] :

$$A = U\Sigma V^T \quad (2.4)$$

dimana $U^T U = V^T V = I_n$ dan $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $\sigma_i > 0$ untuk $1 \leq i \leq r$, $\sigma_j = 0$ untuk $j \geq r + 1$. Kolom ke-1 sampai ke- r dari matriks orthogonal U dan V mendefinisikan vektor-vektor eigen ortonormal yang berkaitan dengan r nilai eigen tak nol dari $A^T A$. Kolom pada matriks U dan V masing-masing adalah vektor-vektor

singular kiri dan kanan sedangkan nilai singular dari A didefinisikan sebagai elemen diagonal dari matriks Σ yang merupakan akar kuadrat tak negatif dari n nilai eigen matriks $A^T A$.

SVD akan mengurai matriks *term-by-document* yang memiliki dimensi $m \times n$ dalam *space vector model* menjadi 3 matriks U, Σ , dan V yang memiliki dimensi lebih kecil berdasarkan persamaan (2.4). Matriks U merupakan matriks *term* berukuran $m \times r$, Σ matriks dengan elemen diagonal $d_{i,i} = \sigma_i, i \leq k$ berukuran $r \times r$, sedangkan V adalah matrik dokumen yang memiliki dimensi $n \times r$. Gambar 2.1 menunjukkan dekomposisi *truncated SVD* dimana r adalah rank dari A dan k adalah jumlah dari faktor.



Gambar 2. 1 *Truncated SVD*

SVD akan menggunakan pendekatan rank-k untuk pengurangan dimensi. Pendekatan seperti ini disebut *truncated SVD*. Matriks yang dihasilkan dari pendekatan rank-k SVD memiliki tingkat kemiripan dengan matriks *term-by-document*. Dimana untuk penentuan nilai k yang digunakan dengan melihat perbandingan rasio dari nilai eigen σ_i^2 sesuai pada persamaan (2.5).

$$rasio \approx \frac{\sum_{i=1}^k \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \quad (2.5)$$

dimana nilai rasio yang digunakan antara 80-90% yang menunjukkan rasio kemiripan dengan matriks awal.

2.8 Fuzzy C-Means (FCM)

Fuzzy C-Means (FCM) merupakan algoritma partisi (*clustering*) yang sangat dikenal. FCM adalah salah satu teknik *clustering* data yang mana keberadaan tiap-tiap data dalam suatu *cluster* ditentukan oleh nilai keanggotaannya. Konsep dasar FCM, pertama kali adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata untuk tiap *cluster*. Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk tiap-tiap *cluster*. Dengan cara memperbaiki pusat kluster dan nilai keanggotaan tiap-tiap data secara berulang, maka akan dapat dilihat bahwa pusat kluster akan bergerak menuju lokasi yang tepat. Perulangan ini didasarkan untuk meminimumkan fungsi obyektif [13]. Fungsi obyektif yang digunakan pada FCM adalah :

$$J_w(U, V; X) = \sum_{k=1}^n \sum_{i=1}^c (\mu_{ik})^w (d_{ik})^2 \quad (2.6)$$

dengan $w \in [1, \infty)$,

$$d_{ik} = d(x_k - v_i) = \left[\sum_{j=1}^m (x_{kj} - v_{ij})^2 \right]^{\frac{1}{2}} \quad (2.7)$$

X adalah data yang akan dikelompokkan :

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \quad (2.8)$$

V adalah matriks pusat *cluster* :

$$V = \begin{bmatrix} v_{11} & \cdots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{c1} & \cdots & v_{cm} \end{bmatrix} \quad (2.9)$$

U adalah matriks partisi :

$$U = \begin{bmatrix} \mu_{11}(x_1) & \mu_{12}(x_2) & \cdots & \mu_{1n}(x_n) \\ \mu_{21}(x_1) & \mu_{22}(x_2) & \cdots & \mu_{2n}(x_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{c1}(x_1) & \mu_{c2}(x_2) & \cdots & \mu_{cn}(x_n) \end{bmatrix} \quad (2.10)$$

Dimana nilai μ_{ic} menunjukkan nilai keanggotaan data ke- i pada *cluster* ke- c yang mana nilai ini akan menentukan keanggotaan data pada *cluster*. Nilai $z = \operatorname{argmax}(\mu_{ic})$ menunjukkan keanggotaan data terdapat pada *cluster* ke- z . Nilai J_w terkecil adalah yang terbaik, sehingga :

$$J_w^*(U^*, V^*; X) = \min_{M_k} J(U, V, X) \quad (2.11)$$

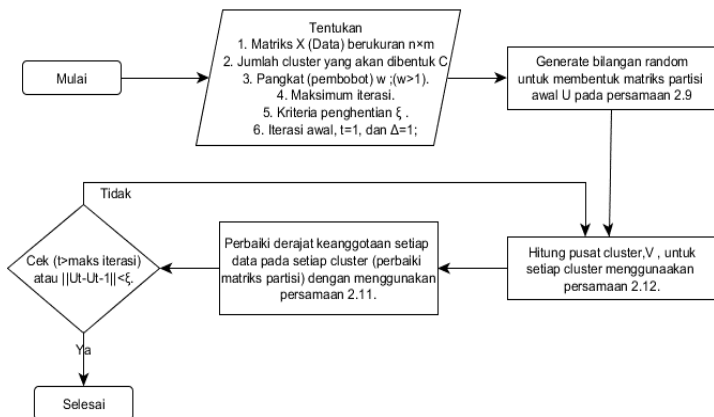
Jika $d_{ik} > 0, \forall i, k; w > 1$, dan X setidaknya memiliki c elemen, maka solusi optimal yang dapat meminimalisasi J_w :

$$\mu_{ik} = \left[\left(\frac{d_{ik}^2}{\sum_{j=1}^C d_{jk}^2} \right)^{\frac{1}{w-1}} \right]^{-1}; 1 \leq i \leq C; 1 \leq k \leq n \quad (2.12)$$

dimana d_{ik} merupakan jarak data ke- k dengan pusat *cluster* ke- i dan nilai pembagian d_{jk} merupakan jumlahan dari jarak data ke- k terhadap semua *cluster*.

$$v_{ij} = \frac{\sum_{k=1}^n (\mu_{ik})^w x_{kj}}{\sum_{k=1}^n (\mu_{ik})^w}; 1 \leq i \leq C; 1 \leq j \leq m \quad (2.13)$$

Berikut *flowchart* algoritma FCM :



2.9 Adaptive Neuro Fuzzy Inference System (ANFIS)

ANFIS adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Dapat dikatakan bahwa ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data [13]. Hasil *clustering* dari FCM akan digunakan sebagai parameter premis pada aturan model sugeno. Dimana setiap atribut dalam data dicari nilai mean dan standar deviasi yang berada dalam *cluster* yang sama, kemudian nilai tersebut akan digunakan sebagai parameter a dan c .

Misalkan terdapat m aturan model Sugeno yang digunakan adalah sebagai berikut [14] :

If x_1 is A_1 and x_2 is B_1, \dots, x_n is N_1 then $f_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n + c_{10}$

If x_1 is A_2 and x_2 is B_2, \dots, x_n is N_2 then $f_2 = c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}x_n + c_{20}$

.

.

.

If x_1 is A_m and x_2 is B_m, \dots, x_n is N_m then $f_m = c_{m1}x_1 + c_{m2}x_2 + \dots + c_{mn}x_n + c_{m0}$

Dimana $A_1, A_2, \dots, A_m, B_1, B_2, \dots, B_m$ dan N_1, N_2, \dots, N_m adalah himpunan fuzzy untuk setiap input x_1, x_2, \dots, x_n dengan $c_{11}, c_{12}, \dots, c_{1n}, c_{10}, c_{21}, c_{22}, \dots, c_{2n}, c_{20}, \dots, c_{m1}, c_{m2}, \dots, c_{mn}, c_{m0}$ adalah parameter linier pada bagian konsekuen dari model Sugeno.

Jika agregat untuk aturan di atas adalah w_1, w_2, \dots, w_m , maka dapat dihitung rata-rata terbobot :

$$f = \frac{\sum_{i=1}^m w_i f_i}{\sum_{i=1}^m w_i} = \sum_{i=1}^m \bar{w}_i f_i \quad (2.14)$$

Jaringan ANFIS seperti yang terlihat pada Gambar 2.1 terdiri dari lapisan-lapisan sebagai berikut :

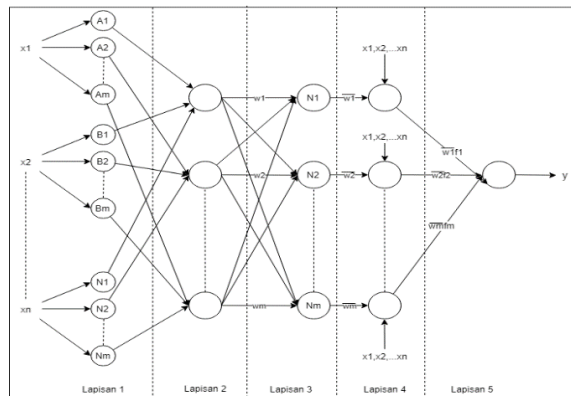
1. Tiap-tiap neuron i pada lapisan pertama adaptif terhadap parameter suatu fungsi aktivasi. Output dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan input yaitu : $\mu_{A_1}(x_1), \mu_{B_1}(x_2), \dots, \mu_{N_1}(x_n), \mu_{A_2}(x_1), \mu_{B_2}(x_2), \dots, \mu_{N_2}(x_n), \dots, \mu_{A_m}(x_1), \mu_{B_m}(x_2), \dots, \mu_{N_m}(x_n)$. Sebagai contoh, fungsi keanggotaan yang digunakan dapat berupa *generalized bell membership function* pada persamaan (2.15) atau *gaussian*

membership function untuk atribut pada persamaan (2.16). Dalam jurnal penelitian yang berjudul *An investigation of membership functions on performance of ANFIS for solving classification problems* [15] disebutkan bahwa kedua fungsi keanggotaan tersebut memiliki kurva yang halus. Hasil dari penelitian tersebut menunjukkan bahwa fungsi keanggotaan *gaussian* dan *generalized bell* memiliki error yang lebih kecil dibanding fungsi aktivasi yang lain.

$$\mu_{Ni}(x_n) = \frac{1}{1 + \left| \frac{x_n - c_i}{a_i} \right|^{2b}} \quad (2.15)$$

$$\mu_{Ni}(x_n) = \exp \left[-0.5 \left(\frac{x_n - c_i}{a_i} \right)^2 \right] \quad (2.16)$$

dimana $1 \leq n \leq \text{banyak atribut}$, $1 \leq m \leq \text{banyak aturan}$, $\{a_i, b, c_i\}$ adalah parameter-parameter dari fungsi keanggotaan yang biasanya disebut parameter premis. Nilai awal parameter a_i dan c_i diperoleh dari hasil *clustering* dari FCM dengan mencari nilai mean dan standar deviasi setiap atribut (x_1, x_2, \dots, x_n) dalam *cluster* $_i$. Sedangkan b merupakan suatu konstanta bilangan bulat positif



Gambar 2. 2 Arsitektur Jaringan ANFIS

2. Tiap-tiap neuron pada lapisan kedua berupa neuron tetap yang outputnya adalah hasil perkalian dari masukan. Biasanya digunakan operator AND. Tiap-tiap node mempresentasikan μ agregrat dari aturan ke- i .

$$w_i = \mu_{Ai}(x_1) * \mu_{Bi}(x_2) * \dots * \mu_{Ni}(x_n), i = 1, 2, \dots, m \quad (2.17)$$

3. Tiap-tiap neuron pada lapisan ketiga berupa node tetap yang merupakan hasil perhitungan rasio dari a agregrat(w), dari aturan ke- i terhadap jumlah dari keseluruhan a predikat.

$$\bar{w}_i = \frac{w_i}{\sum_i w_i}, i = 1, 2, \dots, m \quad (2.18)$$

Hasil ini dikenal dengan nama *normalized firing strength*.

4. Tiap-tiap neuron pada lapisan keempat merupakan node adaptif terhadap suatu output

$$\bar{w}_i f_i = c_{i1}x_1 + c_{i2}x_2 + \dots + c_{in}x_n + c_{i0}, i = 1, 2, \dots, m \quad (2.19)$$

Dengan $(c_{i1}x_1 + c_{i2}x_2 + \dots + c_{in}x_n + c_{i0})$ adalah parameter-parameter pada neuron tersebut. Parameter-parameter tersebut disebut parameter konsekuen.

5. Tiap-tiap neuron pada lapisan kelima adalah node tetap yang merupakan jumlahan semua masukan.

$$\sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, i = 1, 2, \dots, m \quad (2.20)$$

2.10 Proses Pembelajaran dengan RLSE

Pada langkah maju (*forward*), input jaringan akan merambat maju sampai pada lapisan keempat dimana parameter-parameter konsekuen akan diidentifikasi menggunakan *Recursive Least Square Estimator* (RLSE) [14]. Pada saat parameter premis ditemukan maka nilainya merupakan kombinasi linier dari dari parameter konsekuen. Dari persamaan (2.10) diperoleh

$$\begin{aligned}
y &= \overline{w_1}f_1 + \overline{w_2}f_2 \\
&= \overline{w_1}(p_1x + q_1y + r_1) + \overline{w_2}(p_2y + q_2y + r_2) \\
&= \frac{\overline{w_1}x}{\overline{w_2}y}p_1 + \frac{\overline{w_1}y}{\overline{w_2}y}q_1 + \frac{\overline{w_1}}{\overline{w_2}}r_1 + \frac{\overline{w_2}x}{\overline{w_2}y}p_1 + \\
&\quad \frac{\overline{w_2}y}{\overline{w_2}y}q_2 + \frac{\overline{w_2}}{\overline{w_2}}r_2
\end{aligned} \tag{2.21}$$

Jika sejumlah N data dengan parameter input x dan y untuk pembelajaran diterapkan pada persamaan (2.16) didapat :

$$\begin{aligned}
\frac{\overline{w_1}x}{\overline{w_2}y}p_1 + \frac{\overline{w_1}y}{\overline{w_2}y}q_1 + \frac{\overline{w_1}}{\overline{w_2}}r_1 + \frac{\overline{w_2}x}{\overline{w_2}y}p_1 + \frac{\overline{w_2}y}{\overline{w_2}y}q_2 + \frac{\overline{w_2}}{\overline{w_2}}r_2 &= f_1 \\
&\vdots \\
&\vdots \\
&\vdots
\end{aligned} \tag{2.22}$$

$$\begin{aligned}
\frac{\overline{w_1}x}{\overline{w_2}y}p_1 + \frac{\overline{w_1}y}{\overline{w_2}y}q_1 + \frac{\overline{w_1}}{\overline{w_2}}r_1 + \frac{\overline{w_2}x}{\overline{w_2}y}p_1 + \\
\frac{\overline{w_2}y}{\overline{w_2}y}q_2 + \frac{\overline{w_2}}{\overline{w_2}}r_2 &= f_n
\end{aligned}$$

Jika dinyatakan dalam persamaan berbentuk matriks adalah :

$$A\theta = y \tag{2.23}$$

Dengan θ adalah vektor yang elemen-elemennya merupakan himpunan parameter konsekuen dengan ukuran $M \times 1$ (M menunjukkan jumlah parameter konsekuen). A merupakan matriks $P \times M$, dengan P adalah jumlah dari N data pelatihan yang disediakan untuk jaringan adaptif dan y merupakan vektor keluaran berukuran $P \times 1$ yang elemen-elemennya berupa N buah data keluaran. Pada umumnya jumlah dari data pelatihan lebih besar dari jumlah parameter konsekuen, sehingga solusi terbaik untuk θ adalah meminimumkan jumlah kuadrat error $\sum_{i=1}^N (y_i - a_i^T \theta)^2$, dengan a_i^T merupakan vektor baris dari matriks A , dengan menggunakan metode LSE untuk estimasi parameter θ diperoleh:

$$\theta^* = (A^T A)^{-1} A^T y \tag{2.23}$$

Dengan menggunakan metode rekursif LSE maka dari persamaan (2.20) diperoleh

$$\begin{cases} P_{i+1} = P_i - \frac{P_i a_{i+1} a_{i+1}^T P_i}{1 + a_{i+1}^T P_i a_{i+1}} \\ \theta_{i+1} = \theta_i + P_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T \theta_i) \end{cases} \quad (2.24)$$

Dengan a_i^T adalah vektor baris dari matriks A . P_i disebut matriks kovariansi yang didefinisikan dengan :

$$P_i = (A^T A)^{-1} \quad (2.25)$$

2.11 Proses Pembelajaran Propagasi Error

Parameter premis $\{a, b, c\}$ merupakan parameter adaptif yang dapat dilatih untuk mendapatkan nilai parameter yang sesuai dengan lingkungannya. Misalkan sebuah sistem jaringan adaptif dinyatakan dengan L lapisan dan lapisan ke- i mempunyai $N(i)$ simpul. Keluaran dan fungsi simpul ke- ℓ pada lapisan ke- ℓ dinyatakan dengan $x_{\ell,i}$ dan $f_{\ell,i}$. Keluaran simpul merupakan fungsi dari sinyal yang masuk dan parameter sistem seperti pada persamaan (2.26).

$$x_{\ell,i} = f_{\ell,i}(x_{\ell-1,1}, \dots, x_{\ell-1,N(\ell-1)}, a, b, c) \quad (2.26)$$

dengan a, b, c adalah parameter simpul.

Maka sejumlah p pasangan data untuk proses belajar jaringan adaptif, selanjutnya dapat didefinisikan pengukuran kesalahan data latih ke- p adalah jumlah kuadrat error pada lapisan ke- L pada data pelatihan ke- p ($1 \leq p \leq N$) yaitu [14]:

$$E_p = \sum_{k=1}^{N(L)} (d_k - X_{k,p}^L)^2 \quad (2.27)$$

Dimana d_k adalah komponen ke- k dari vektor keluaran yang diinginkan(target). Sedangkan $X_{i,p}^L$ adalah vektor keluaran aktual yang dihasilkan oleh sistem jaringan adaptif dengan masukan dari vektor masukan ke- p dari P data *training*. Tahap awal pembelajaran dimulai dengan menghitung tingkat kesalahan pada keluaran node ke- i lapisan ke- L dengan persamaan derivasi sebagai berikut :

$$\xi_{L,i} = \frac{\partial E_p}{\partial x_{i,p}^L} = -2(d_{i,p} - x_{i,p}^L) \quad (2.28)$$

Untuk simpul dalam pada lapisan ke- L posisi ke- i , error dapat dihitung dengan aturan rantai sebagai berikut :

$$\frac{\partial E_p}{\partial X_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial X_{m,p}^{l+1}} \frac{\partial X_{m,p}^{l+1}}{\partial X_{l,i}} \quad (2.29)$$

Dengan $0 \leq l \leq L - 1$. Nilai error simpul dalam dapat dinyatakan sebagai kombinasi linier dari nilai error lapisan simpul ke $l(l + 1)$. Persamaan (2.29) digunakan untuk menghitung sinyal error pada simpul lapisan ke- i sampai ke $l(l < L)$. Jika α adalah parameter yang digunakan pada beberapa simpul, maka persamaan yang diperoleh adalah :

$$\frac{\partial E_p}{\partial \alpha} = \sum_{x^* \in S} \frac{\partial E_p}{\partial x^*} \frac{\partial f^*}{\partial \alpha} = \varepsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \quad (2.30)$$

Dimana $f_{l,i}$ merupakan fungsi keanggotaan yang digunakan dan S adalah himpunan simpul yang mengandung parameter α , sehingga keseluruhan keluaran error dari α sebagai berikut :

$$\frac{\partial^* E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad (2.31)$$

dengan metode *steepest gradient descent*, persamaan untuk memperbaiki parameter α adalah :

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \quad (2.32)$$

Dengan η adalah nilai *learning rate*.

2.12 Ukuran Evaluasi Sistem

Sistem yang telah dibentuk akan dinilai dan dievaluasi dengan sejumlah ukuran antara lain akurasi, presisi, dan *recall*. Untuk melakukan evaluasi dalam model klasifikasi teks dimulai dengan pembentukan *confusion matrix* pada Gambar 2.3 [16].

		Predicted class	
		<i>P</i>	<i>N</i>
Actual Class	<i>P</i>	True Positives (TP)	False Negatives (FN)
	<i>N</i>	False Positives (FP)	True Negatives (TN)

Gambar 2. 3 *Confusion Matrix* Model Klasifikasi

Akurasi menunjukkan persentase dari semua data yang diklasifikasikan dengan benar oleh sistem yang dibuat. Meskipun akurasi terlihat cukup bagus untuk menilai suatu model klasifikasi namun akurasi tidak bekerja dengan baik pada data yang memiliki jumlah data tiap kelas tidak seimbang (*unbalance dataset*) oleh karena itu diperlukan ukuran performansi lain yaitu presisi dan *recall*. Presisi mengukur persentase dari data yang dideteksi sistem adalah benar pada kelas tersebut. Sedangkan *recall* mengukur persentase dari data dari suatu kelas teridentifikasi benar oleh sistem. Persamaan (2.33), (2.34), dan (2.35) menunjukkan rumus yang digunakan dalam perhitungan akurasi, presisi, dan *recall*.

$$akurasi = \frac{TP + FP}{TP + FP + FN + FP} \quad (2.33)$$

$$presisi = \frac{TP}{TP + FP} \quad (2.34)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.35)$$

Keterangan :

TP : Jumlah teks yang diprediksi sebagai kelas i dan kelas target sebenarnya adalah i

FP : Jumlah teks yang diprediksi sebagai kelas i, namun kelas target sebenarnya bukan i

TN : Jumlah teks yang diprediksi bukan kelas i dan kelas target sebenarnya bukan i

FN : Jumlah teks yang diprediksi bukan kelas i, namun kelas target sebenarnya adalah i

BAB III

METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahap-tahap pengerjaan dalam menyelesaikan penelitian ini dapat dilakukan secara sistematis dan diatur sebaik-baiknya.

3.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan melakukan *web scraping* pada web *e-commerce* www.tokopedia.com untuk mengambil data ulasan beserta nilainya dari produk-produk elektronik dari berbagai toko.

3.2 Praproses Data

Pada tahap ini akan dilakukan pra-proses data ulasan yang telah diperoleh. Pra proses data akan dilakukan menggunakan modul NLP yang meliputi tokenisasi, *spelling* normalization, filtrasi, *stemming*, dan penerapan *N-gram* untuk pemecahan kata pada teks ulasan. *N-Gram* yang akan digunakan yaitu *unigram*, *bigram*, dan kombinasi *unigram bigram*. Selanjutnya akan dilakukan pembentukan *space vector model* sehingga berbentuk matrik *term-by-document* hasil dari matriks selanjutnya akan di lakukan pembobotan menggunakan TF-IDF.

3.3 Ekstraksi Fitur

Pada tahap ini akan dilakukan ekstraksi fitur dari matrik dari hasil pembentukan *space vector model*. Metode yang digunakan adalah *truncated SVD* dimana matrik *term-by-document* akan didekomposisi menjadi matriks U , Σ , dan V . Selanjutnya matriks tersebut akan direduksi jumlahnya atributnya. Matriks V (matriks dokumen) nantinya kan digunakan sebagai input dalam pembelajaran FCM-ANFIS.

3.4 Implementasi *Fuzzy C-Means* dan *Adaptive Neuro Fuzzy Inference System*

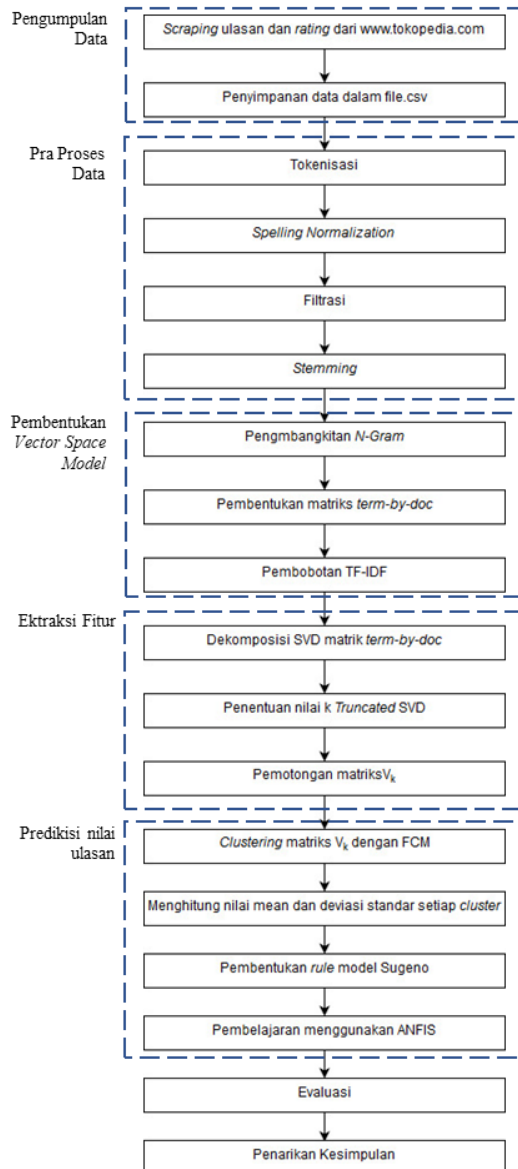
Selanjutnya akan dilakukan proses analisis sentimen terhadap data ulasan. Proses analisis sentimen mirip dengan proses klasifikasi data dengan kelas setiap dokumen merupakan *rating* yang diperoleh dari pelanggan. Proses analisis sentimen berdasarkan pada skala rating dimana secara berurutan skala 1-5 memiliki sentimen sangat buruk, buruk, cukup, baik, dan sangat baik berdasarkan jenis rating yang didefinisikan oleh Tokopedia. Pada tahap ini kumpulan data ulasan yang telah dibentuk dalam matiks akan dikelompokkan menggunakan metode FCM dengan jumlah *cluster* sebanyak 5. Hasil dari metode FCM yaitu nilai mean dan deviasi standar dari tiap-tiap atribut dalam setiap *cluster* akan digunakan sebagai nilai awal dari parameter-parameter premis dalam basis aturan model Sugeno. Selanjutnya aturan-aturan yang telah terbentuk akan digunakan dalam proses pembelajaran jaringan ANFIS untuk proses prediksi nilai ulasan produk. ANFIS akan melakukan perbaikan terhadap parameter-parameter pada bagian premis dan konsekuen agar output dari ANFIS memiliki error yang seminimum mungkin dengan output data yang sesungguhnya (*rating* ulasan).

3.5 Pengujian dan Evaluasi

Selanjutnya akan dilakukan proses pengujian dan evaluasi dari hasil implementasi. Akan dilihat bagaimana performansi dari metode yang digunakan untuk proses prediksi nilai ulasan.

3.6 Penarikan Kesimpulan

Setelah evaluasi selesai dilakukan, akan dilakukan penarikan kesimpulan dari sekian tahapan yang telah dilakukan sebelumnya.



Gambar 3. 1 Diagram Alir Metode Penelitian

BAB IV

PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan perancangan desain sistem yang digunakan sebagai acuan untuk implementasi sistem. Perancangan sistem menggambarkan proses rancang bangun secara terperinci dari awal tahap pengumpulan data hingga pembuatan model FCM-ANFIS. Langkah-langkah dalam pengimplementasian sistem berdasarkan desain sistem yang sudah dibentuk. Adapun ilustrasi dalam penjelasan tahapan akan menggunakan data contoh untuk memudahkan dalam proses interpretasi.

4.1 Pengumpulan Data

Dalam penelitian ini dibutuhkan data sebagai bahan masukan dari algoritma yang akan dijalankan. Data yang digunakan dalam penelitian ini adalah data ulasan beserta nilainya dari produk elektronik di website Tokopedia. Pengambilan data dilakukan dengan proses *scraping* dari website www.tokopedia.com. Proses *scraping* yang dibuat menggunakan *library* request untuk mengirim *request* pada HTTP dan *library* BeautifulSoup untuk membaca *tag* HTML dari url yang diminta. Dalam menampilkan ulasan, Tokopedia menggunakan API yang mengembalikan data ulasan berupa data .json dari setiap produk, sehingga dalam proses *scraping* akan dilakukan *request* dari url API ulasan menggunakan nomor id produk. Hasil *request* akan dikembalikan berupa data ulasan .json yang berisi *review_id*, *message*, *rating*, *update_time*, *attachments*, *user_id*, dan *full_name*. Data .json yang diperoleh selanjutnya diolah agar dapat diubah dalam bentuk .csv untuk mempermudah penyimpanan dan penggunaan. Dalam proses penyimpanan dalam bentuk .csv atribut yang akan diambil disesuaikan dengan kebutuhan dalam penelitian. Atribut yang akan disimpan adalah *user_id*, *full_name*, *message*, dan *rating* seperti yang terlihat dalam Tabel 4.1.

Berikut disajikan syntax program untuk *scraping* :

```
import pandas as pd
import json
import requests
import math
from urllib.request import urlopen
from bs4 import BeautifulSoup
import re
url = https://www.tokopedia.com/px/fs-px-digital-tv-indoor-antenna-da-3920n/review

def scraping_review(url):
    url_new = urlopen(url).read()
    soup=BeautifulSoup(url_new, "html.parser")

    product_name=punctreplace(str(soup.find('title').get_text()))
    id1=soup.find('div', 'container container-product')
    product_id=re.findall('\d+', id1.get('id'))[0]
    number_rev=int(soup.find('meta',
itemprop="ratingCount").get('content'))
    num_page=math.ceil(number_rev/20)
    url_all =
['https://www.tokopedia.com/reputationapp/review/api/v2/
product/'+product_id+'?page='+str(i)+'&total=20' for i in
range(1, num_page+1)]
    data_full=[]
    for i, v in enumerate(url_all):
        data_json={}
        req = requests.get(v)
        data_json = req.json()
        for j in range(20):
            try :

komentar=remove_emoji(data_json['data']['list'][j]['message'])
            except :
                print('No More Review')
                break

reviewer=data_json['data']['list'][j]['reviewer']['user_id']

full_name=data_json['data']['list'][j]['reviewer']['full_name']
]
        rating = data_json['data']['list'][j]['rating']
        if komentar!='':
            data=[reviewer, full_name, komentar, rating]
            data_full.append(data)
    return data_full
```

Tabel 4. 1 Bentuk Dataset Ulasan Tokopedia

<i>user_id,</i>	<i>full_name</i>	<i>message</i>	<i>rating</i>
5521905	Fikri Yasir	ga rekomended banget buat dropship, bukan untung malah rugi, pengiriman lama sampe angus di transaksi lain	1
608626	Anindita Saktiaji	Barang yang saya terima sepertinya cacat produksi. Angin bukannya berhembus ke arah depan, tapi malah ke belakang kipas. Aneh...	2
42741601	gohan yuan	Barang,.sudah sampai tujuan,tapi agak sayang.kardusnya rusak,.	3
3231312	Setiadi Wedasmara	barang sudah di terima kemarin,warna sesuai & berfungsi dgn baik :),cuma ada 1 baling2 nya yg bengkok..tp udh fix. thx	4
50133084	Pinku Artshop	fast respon, recommended seller :)	5

Untuk mempermudah dalam interpretasi jalannya program pada data masukkan secara langsung, maka akan digunakan data contoh yang jumlahnya adalah sebelas ulasan beserta nilainya yang disajikan dalam Tabel 4.2.

Tabel 4. 2 Contoh Data Inputan

Ulasan	Rating
Brang sesuai deskripsi, seller recommended banget suka aku. Pengiriman cepat. Suara juga keras	5
Brng MANTAP, pengiriman cpt. Mantap gan brng sampe tanpa komplain.	5
Brang bagus sesuai deskripsi tp pengiriman agak lama. Suka aku. Oke good	4
Lumayan lah barangnya tapi suara mic agak aneh	3
Psan warna hitam dateng silver. Baterai gampang drop. Respon seller juga lama	2
Brang banyak minusnya gak bisa digunakan. Gak original	1
Packaging dan pengiriman cukup bgus minus seller respon lama	3
Barng okee banget, gak nyesel beli disini. Suara mantap pengiriman cepat. Mantap pokok	5
Barang super bagus pake banget. Oke deh pokok. Suka banget deh, Seller oke	5
Sesuai deskripsi, Oke deh boleh juga!!	4
Kualitas produk mantappp bagus, sesuai harga.	4

Dari data contoh di atas dapat dilihat beberapa opini tidak tersusun secara rapi misalnya masih terdapat tanda baca, kata yang berupa singkatan, huruf besar, dan lain-lain. Data tersebut akan disimpan dalam csv untuk selanjutnya melalui tahap praproses data.

4.2 Praproses Data

Data ulasan yang sudah tersimpan belum sepenuhnya siap untuk dimasukkan ke dalam tahap prediksi nilai *rating* karena data masih tidak terstruktur dan mengandung banyak *noise* yang dapat merusak performa dari model yang akan dibuat. Data ulasan masih memuat angka, tanda baca, kata-kata yang berbahasa inggris, kata dengan penulisan yang salah maupun yang kurang bermakna untuk dijadikan fitur. Maka dari itu perlu dilakukan prapemrosesan data dengan

tujuan untuk membersihkan data yang akan digunakan dalam pembelajaran model. Proses ini sangat penting karena akan mempengaruhi performansi dari model yang akan digunakan dalam prediksi nilai ulasan.

Berikut merupakan tahap-tahap dalam praproses data pada penelitian ini :

4.2.1 Tokenisasi

Pada tahap ini dilakukan proses pemisahan kalimat menjadi satuan kata yang saling independen. Hal ini berguna untuk proses selanjutnya yaitu filtrasi dan *stemming* dimana keduanya bekerja dalam satuan kata. Sebelum dilakukan tokenisasi setiap kalimat akan dilakukan *case folding* untuk mengubah setiap huruf dalam kalimat ulasan berubah menjadi huruf kecil (*lower case*). Selanjutnya setiap kalimat juga akan dibersihkan dari tanda baca dan angka selanjutnya kalimat ulasan siap untuk dilakukan proses tokenisasi. Berikut *syntax* dalam proses tokenisasi.

```
def punctreplace(sentence):
    punct =set(['\r', '\n', ',', '.', '!', '"', '#', '$', '%', '&', '"', '(', ')', '*', '+', '-', '/', ':', ';', '<', '=', '>', '?', '@', '~', '_', '^', '[', ']', '/', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0'])
    for i in punct:
        sentence=sentence.replace(i, ' ')
    return sentence

# melakukan pemotongan kata
def tokenisasi(review):
    review_token=[]
    for i in review:
        i=i.lower()
        a=punctreplace(i)
        token=word_tokenize(a)
        review_token.append(token)
    return review_token
```

Hasil proses tokenisasi yang diterapkan pada data contoh dapat dilihat pada Tabel 4.3 yang setiap array yang berisi kata-kata pada setiap ulasan.

Tabel 4. 3 *Array* Ulasan Hasil Tokenisasi

No	Array Ulasan
1	['brang', 'sesuai', 'deskripsi', 'seller', 'recommended', 'banget', 'suka', 'aku', 'pengiriman', 'cepat', 'suara', 'juga', 'keras']
2	['brng', 'mantap', 'pengiriman', 'cpt', 'mantap', 'gan', 'brng', 'sampe', 'tanpa', 'komplain']
3	['brang', 'bagus', 'sesuai', 'deskripsi', 'tp', 'pengiriman', 'agak', 'lama', 'suka', 'aku', 'oke', 'good']
4	['lumayan', 'lah', 'barangnya', 'tapi', 'suara', 'mic', 'agak', 'aneh']
5	['psan', 'warna', 'hitam', 'dateng', 'silver', 'baterai', 'gampang', 'drop', 'respon', 'seller', 'juga', 'lama']
6	['brang', 'banyak', 'minusnya', 'gak', 'bisa', 'digunakan', 'gak', 'original']
7	['packaging', 'dan', 'pengiriman', 'cukup', 'bgus', 'minus', 'seller', 'respon', 'lama']
8	['barng', 'okee', 'banget', 'gak', 'nyesel', 'beli', 'disini', 'suara', 'mantap', 'pengiriman', 'cepat', 'mantap', 'pokok']
9.	['barang', 'super', 'bagus', 'pake', 'banget', 'oke', 'deh', 'pokok', 'suka', 'banget', 'deh', 'seller', 'oke']
10.	['sesuai', 'deskripsi', 'oke', 'deh', 'boleh', 'juga']
11.	['kualitas', 'produk', 'mantappp', 'bagus', 'sesuai', 'harga']

4.2.2 *Spelling Normalization*

Spelling normalization adalah tahap yang dilakukan dengan tujuan agar kualitas dokumen menjadi baik, dimana akan dilakukan proses perbaikan kata-kata yang penulisannya salah, kata singkatan, dan kata gaul. Selain itu dalam proses ini juga dilakukan proses pengubahan kata berbahasa inggris menjadi kata berbahasa Indonesia namun kata yang diubah hanya kata bahasa inggris yang sering muncul dalam ulasan produk seperti *'seller'*, *'recommended'*, dan *'good'* karena kata-kata tersebut cukup mempengaruhi untuk membedakan dokumen antar kelas *rating*. Dalam proses *spelling normalization* ini di buat korpus tersendiri dalam file teks yang kemudian diubah menjadi tipe data *dictionary* dalam python. Korpus dalam tahap ini dapat dilihat dalam Lampiran 1, dimana Gambar 4.1

menunjukkan contoh *dictionary* dari beberapa kata yang digunakan dalam proses ini.

```
{'barang': ['brg', 'brang', 'brng'],
'direkomendasikan': ['recommended', 'rekomended',
'recomended', 'recommender', 'recommend',
'recomend', 'rekomen', 'recomendeed', 'recomm'],
'pengemasan': ['packing', 'pckng', 'packingan',
'paking', 'packaging', 'packagingnya']}
```

Gambar 4. 1 Contoh *dictionary spelling normalization*

Tipe data *dictionary* sendiri terdiri dari dua bagian yaitu *key* dan *value* dimana dalam proses ini *key* menunjukkan kata yang benar sedangkan *value* berisikan *list* kata-kata yang kurang tepat dalam menuliskan kata dari *key* tersebut. Berikut merupakan *syntax* dalam proses *spelling normalization*.

```
def load_corpus():
    with open('repairword1.txt', 'r') as document:
        answer = {}
        for line in document:
            if line.strip():
                key, value = line.split(None, 1)
                answer[key] = value.split()
    return answer
def typoremoval(kata):
    for key in answer:
        indeks=key
        list_temp=answer.get(key)
        if kata in list_temp:
            return key
    return kata
# membenarkan list seluruh review dari typo
def removetypo(review2):
    kalimat_temp=[]
    for kalimat in review2:
        token_temp=[]
        for kata in kalimat :
            if len(kata)==1:
                continue
            else:
                temp=typoremoval(kata)
                token_temp.append(temp)
        kalimat_temp.append(token_temp)
    return kalimat_temp
```

Tabel 4.4 menunjukkan hasil dari data pada Tabel 4.3 setelah melalui tahap *spelling normalization*, dimana hasilnya berupa *array* ulasan yang sudah diperbaiki setiap katanya.

Tabel 4. 4 *Array Ulasan Hasil Spelling Normalization*

No	Array Ulasan
1	['barang', 'sesuai', 'deskripsi', 'penjual', 'direkomendasikan', 'banget', 'suka', 'saya', 'pengiriman', 'cepat', 'suara', 'juga', 'keras']
2	['barang', 'mantap', 'pengiriman', 'cepat', 'mantap', 'gan', 'barang', 'sampai', 'tanpa', 'komplain']
3	['barang', 'bagus', 'sesuai', 'deskripsi', 'tapi', 'pengiriman', 'agak', 'lama', 'suka', 'saya', 'oke', 'bagus']
4	['lumayan', 'lah', 'barangnya', 'tapi', 'suara', 'mic', 'agak', 'aneh']
5	['pesan', 'warna', 'hitam', 'datang', 'silver', 'baterai', 'gampang', 'drop', 'respon', 'penjual', 'juga', 'lama']
6	['barang', 'banyak', 'minusnya', 'tidak', 'bisa', 'digunakan', 'tidak', 'original']
7	['pengemasan', 'dan', 'pengiriman', 'cukup', 'bagus', 'minus', 'penjual', 'respon', 'lama']
8	['barang', 'oke', 'banget', 'tidak', 'nyesel', 'beli', 'disini', 'suara', 'mantap', 'pengiriman', 'cepat', 'mantap', 'pokok']
9	['barang', 'super', 'bagus', 'pakai', 'banget', 'oke', 'deh', 'pokok', 'suka', 'banget', 'deh', 'penjual', 'oke']
10	['sesuai', 'deskripsi', 'oke', 'deh', 'boleh', 'juga']
11	['kualitas', 'produk', 'mantap', 'bagus', 'sesuai', 'harga']

4.2.3 Filtrasi

Proses selanjutnya adalah filtrasi atau biasa disebut *stopword removal*. Pada tahap ini dibuat korpus yang berisi kata-kata yang merupakan *stopwords* dalam bahasa Indonesia yang dapat dilihat pada Lampiran 2. Adapun kata-kata yang dihilangkan yaitu :

- a. Penghubung antar kata seperti : dan, atau

- b. Preposisi, contoh : ke, di, pada
- c. Kata-kata yang tidak diinginkan namun memiliki frekuensi tinggi seperti : lah, nah, nih, gan

Umumnya dalam bahasa Indonesia kata seperti agak, sangat, dan tetapi merupakan bagian dari *stopwords*, namun dalam penelitian ini kata-kata tersebut akan tetap di pergunakan karena cukup berpengaruh terhadap sentimen dari *rating* yang diberikan. Berikut *syntax* dari proses filtrasi.

```
def load_corpus():
    with open('stopwords.csv', newline='') as csvfile:
        spamreader = csv.reader(csvfile, delimiter=',')
        for row in spamreader:
            stopwords=row
    return set(stopwords)
def remove_stopword(review1):
    review=list(review1)
    dok_temp=[]
    for kalimat in review:
        list_1=list(set(kalimat) & set(stopword))
        for i in list_1:
            kalimat=list(filter(lambda a: a!=i,kalimat))
            dok_temp.append(kalimat)
    return dok_temp
```

Tabel 4.5 menunjukkan hasil dari proses filtrasi dari data *array* ulasan pada Tabel 4.4 berupa ulasan yang telah dihapus kata yang merupakan *stopword*. Data ulasan yang udah bersih dari *stopword* selanjutnya akan melalui proses *stemming*.

4.2.4 *Stemming*

Stemming merupakan tahap untuk merubah kata menjadi kata dasarnya. Setiap kata akan dihapuskan bagian *prefix*, *infix*, dan *suffix* sesuai kaidah penulisan imbuhan dalam bahasa Indonesia. Kata dasar yang dihasilkan sesuai dengan kamus besar bahasa Indonesia. Dalam proses ini digunakan *package* Sastrawi dalam bahasa Indonesia. Dalam proses *stemming* akan dilakukan pengecekan pada setiap kata dalam ulasan sehingga pada proses ini membutuhkan waktu yang cukup lama.

Tabel 4. 5 Array Ulasan Hasil Filtrasi Data

No	Array Ulasan
1	['barang', 'sesuai', 'deskripsi', 'penjual', 'direkomendasikan', 'banget', 'suka', 'pengiriman', 'cepat', 'suara', 'keras']
2	['barang', 'mantap', 'pengiriman', 'cepat', 'mantap', 'barang', 'komplain']
3	['barang', 'bagus', 'sesuai', 'deskripsi', 'pengiriman', 'agak', 'lama', 'suka', 'oke', 'bagus']
4	['lumayan', 'barangnya', 'suara', 'mic', 'agak', 'aneh']
5	['pesan', 'warna', 'hitam', 'datang', 'silver', 'baterai', 'gampang', 'drop', 'respon', 'penjual', 'lama']
6	['barang', 'banyak', 'minusnya', 'tidak', 'bisa', 'digunakan', 'tidak', 'original']
7	['pengemasan', 'dan', 'pengiriman', 'cukup', 'bagus', 'minus', 'penjual', 'respon', 'lama']
8	['barang', 'oke', 'banget', 'tidak', 'nyesel', 'beli', 'disini', 'suara', 'mantap', 'pengiriman', 'cepat', 'mantap', 'pokok']
9	['barang', 'super', 'bagus', 'pakai', 'banget', 'oke', 'deh', 'pokok', 'suka', 'banget', 'deh', 'penjual', 'oke']
10	['sesuai', 'deskripsi', 'oke', 'deh', 'boleh', 'juga']
11	['kualitas', 'produk', 'mantap', 'bagus', 'sesuai', 'harga']

Berikut merupakan *syntax* dalam proses *stemming* :

```
def stemming_word(review):
    review_temp=[]
    for kalimat in review:
        kalimat_temp=[]
        for i,katal in enumerate(kalimat):
            kalimat_temp.append(stemmer.stem(katal))
        review_temp.append(kalimat_temp)
    return review_temp
```

Tabel 4.6 merupakan hasil keluaran pada tahap *stemming* berupa array ulasan dengan elemen berupa kata yang sudah dihapuskan imbuhan di dalamnya.

Tabel 4. 6 Array Ulasan Hasil Proses *Stemming*

No	Array Ulasan
1	['barang', 'sesuai', 'deskripsi', 'jual', 'rekomendasi', 'banget', 'suka', 'kirim', 'cepat', 'suara', 'keras']
2	['barang', 'mantap', 'kirim', 'cepat', 'mantap', 'barang', 'komplain']
3	['barang', 'bagus', 'sesuai', 'deskripsi', 'kirim', 'agak', 'lama', 'suka', 'oke', 'bagus']
4	['lumbayan', 'barang', 'suara', 'mic', 'agak', 'aneh']
5	['pesan', 'warna', 'hitam', 'datang', 'silver', 'baterai', 'gampang', 'drop', 'respon', 'jual', 'lama']
6	['barang', 'banyak', 'minus', 'tidak', 'guna', 'tidak', 'original']
7	['kemas', 'kirim', 'cukup', 'bagus', 'minus', 'jual', 'respon', 'lama']
8	['barang', 'oke', 'banget', 'tidak', 'nyesel', 'beli', 'suara', 'mantap', 'kirim', 'cepat', 'mantap', 'pokok']
9	['barang', 'super', 'bagus', 'pakai', 'banget', 'oke', 'pokok', 'suka', 'banget', 'jual', 'oke']
10	['sesuai', 'deskripsi', 'oke']
11	['kualitas', 'produk', 'mantap', 'bagus', 'sesuai', 'harga']

4.3 Pembangkitan *N-Gram*

Sebelum kata *array* ulasan di ubah dalam bentuk matriks *term-by-doc* dalam proses *Vector Space Model*, data hasil proses *stemming* pada Tabel 4.6 yang masih merupakan data array dengan *unigram* akan dibangkitkan agar memperoleh data ulasan berupa *bigram* serta kombinasi dari *unigram bigram*. Berikut merupakan *syntax* pembangkitan untuk *bigram* :

```
#untuk membuat bigram
def bigrams(review):
    review_temp=[]
    for kalimat in review:
        kalimat_temp=[]
        for i in range(len(kalimat)-1):
            temp=kalimat[i]+' '+kalimat[i+1]
            kalimat_temp.append(temp)
        review_temp.append(kalimat_temp)
    return review_temp
```

Tabel 4.7 merupakan hasil pembangkitan *bigram* dengan setiap elemen berisi dua kata kombinasi dari kata ke- i dengan kata ke- $i + 1$ dengan $1 \leq i \leq n - 1$, $n =$ jumlah kata.

Tabel 4. 7 Hasil Array Ulasan dengan *Bigram*

No	Array Ulasan
1	['barang sesuai', 'sesuai deskripsi', 'deskripsi jual', 'jual rekomendasi', 'rekomendasi banget', 'banget suka', 'suka kirim', 'kirim cepat', 'cepat suara', 'suara keras']
2	['barang mantap', 'mantap kirim', 'kirim cepat', 'cepat mantap', 'mantap barang', 'barang komplain']
3	['barang bagus', 'bagus sesuai', 'sesuai deskripsi', 'deskripsi kirim', 'kirim agak', 'agak lama', 'lama suka', 'suka oke', 'oke bagus']
4	['lumayan barang', 'barang suara', 'suara mic', 'mic agak', 'agak aneh']
5	['pesan warna', 'warna hitam', 'hitam datang', 'datang silver', 'silver baterai', 'baterai gampang', 'gampang drop', 'drop respon', 'respon jual', 'jual lama']
6	['barang banyak', 'banyak minus', 'minus tidak', 'tidak guna', 'guna tidak', 'tidak original']
7	['kemas kirim', 'kirim cukup', 'cukup bagus', 'bagus minus', 'minus jual', 'jual respon', 'respon lama']
8	['barang oke', 'oke banget', 'banget tidak', 'tidak nyesel', 'nyesel beli', 'beli suara', 'suara mantap', 'mantap kirim', 'kirim cepat', 'cepat mantap', 'mantap pokok']
9	['barang super', 'super bagus', 'bagus pakai', 'pakai banget', 'banget oke', 'oke pokok', 'pokok suka', 'suka banget', 'banget jual', 'jual oke']
10	['sesuai deskripsi', 'deskripsi oke']
11	['kualitas produk', 'produk mantap', 'mantap bagus', 'bagus sesuai', 'sesuai harga']

Tabel 4.8 merupakan hasil pembangkitan *bigram* dan kombinasi (*unigram bigram*) dari data array ulasan pada Tabel 4.6, dengan pembangkitan kombinasi merupakan hasil gabungan dari array *unigram* dan array *bigram*.

Tabel 4. 8 Hasil *Array Ulasan* dengan Kombinasi

No	<i>Array Ulasan</i>
1	['barang', 'sesuai', 'deskripsi', 'jual', 'rekomendasi', 'banget', 'suka', 'kirim', 'cepat', 'suara', 'keras', 'barang sesuai', 'sesuai deskripsi', 'deskripsi jual', 'jual rekomendasi', 'rekomendasi banget', 'banget suka', 'suka kirim', 'kirim cepat', 'cepat suara', 'suara keras']
2	['barang', 'mantap', 'kirim', 'cepat', 'mantap', 'barang', 'komplain', 'barang mantap', 'mantap kirim', 'kirim cepat', 'cepat mantap', 'mantap barang', 'barang komplain']
3	['barang', 'bagus', 'sesuai', 'deskripsi', 'kirim', 'agak', 'lama', 'suka', 'oke', 'bagus', 'barang bagus', 'bagus sesuai', 'sesuai deskripsi', 'deskripsi kirim', 'kirim agak', 'agak lama', 'lama suka', 'suka oke', 'oke bagus']
4	['lumayan', 'barang', 'suara', 'mic', 'agak', 'aneh', 'lumayan barang', 'barang suara', 'suara mic', 'mic agak', 'agak aneh']
5	['pesan', 'warna', 'hitam', 'datang', 'silver', 'baterai', 'gampang', 'drop', 'respon', 'jual', 'lama', 'pesan warna', 'warna hitam', 'hitam datang', 'datang silver', 'silver baterai', 'baterai gampang', 'gampang drop', 'drop respon', 'respon jual', 'jual lama']
6	['barang', 'banyak', 'minus', 'tidak', 'guna', 'tidak', 'original', 'barang banyak', 'banyak minus', 'minus tidak', 'tidak guna', 'guna tidak', 'tidak original']
7	['kemas', 'kirim', 'cukup', 'bagus', 'minus', 'jual', 'respon', 'lama', 'kemas kirim', 'kirim cukup', 'cukup bagus', 'bagus minus', 'minus jual', 'jual respon', 'respon lama']

No	Array Ulasan
8	['barang', 'oke', 'banget', 'tidak', 'nyesel', 'beli', 'suara', 'mantap', 'kirim', 'cepat', 'mantap', 'pokok', 'barang oke', 'oke banget', 'banget tidak', 'tidak nyesel', 'nyesel beli', 'beli suara', 'suara mantap', 'mantap kirim', 'kirim cepat', 'cepat mantap', 'mantap pokok']
9	['barang', 'super', 'bagus', 'pakai', 'banget', 'oke', 'pokok', 'suka', 'banget', 'jual', 'oke', 'barang super', 'super bagus', 'bagus pakai', 'pakai banget', 'banget oke', 'oke pokok', 'pokok suka', 'suka banget', 'banget jual', 'jual oke']
10	['sesuai', 'deskripsi', 'oke', 'sesuai deskripsi', 'deskripsi oke']
11	['kualitas', 'produk', 'mantap', 'bagus', 'sesuai', 'harga', 'kualitas produk', 'produk mantap', 'mantap bagus', 'bagus sesuai', 'sesuai harga']

4.4 Pembentukan Matriks *Term-by-doc*

Data array ulasan masih perlu dirubah dalam bentuk numerik agar dapat dipelajari oleh komputer. Pada *Vector Space Model* data akan diubah ke dalam numerik yaitu dengan mengubah *array* ulasan dalam matriks *term_by_doc* dengan menggunakan *CountVectorizer* dari *library sklearn*. Pada tahap pembentukan matriks ini minimum *document frequency* yang digunakan adalah 2, yang artinya kata yang akan dijadikan fitur dalam kolom harus setidaknya terdapat dalam 2 dokumen jika tidak maka kata akan dihapuskan, hal ini ditujukan untuk mereduksi jumlah fitur yang akan digunakan dalam membentuk model. Gambar 4.2 dan Gambar 4.3 merupakan matriks *term-by-doc* dari data ulasan dengan pembangkitan *unigram* dan kombinasi *unigram bigram*.

Elemen dari matriks *term-by-doc* A_{ij} yang berukuran $m \times n$ dimana i menunjukkan indeks fitur dan j menunjukkan indeks ulasan merepresentasikan frekuensi fitur ke- i dalam ulasan ke- j .

	0	1	2	3	4	5	6	7	8	9	10
agak	0	0	1	1	0	0	0	0	0	0	0
bagus	0	0	2	0	0	0	1	0	1	0	1
banget	1	0	0	0	0	0	0	1	2	0	0
barang	1	2	1	1	0	1	0	1	1	0	0
cepat	1	1	0	0	0	0	0	1	0	0	0
deskripsi	1	0	1	0	0	0	0	0	0	1	0
jual	1	0	0	0	1	0	1	0	1	0	0
kirin	1	1	1	0	0	0	1	1	0	0	0
lama	0	0	1	0	1	0	1	0	0	0	0
mantap	0	2	0	0	0	0	0	2	0	0	1
minus	0	0	0	0	0	1	1	0	0	0	0
oke	0	0	1	0	0	0	0	1	2	1	0
pokok	0	0	0	0	0	0	0	1	1	0	0
respon	0	0	0	0	1	0	1	0	0	0	0
sesuai	1	0	1	0	0	0	0	0	0	1	1
suara	1	0	0	1	0	0	0	1	0	0	0
suka	1	0	1	0	0	0	0	0	1	0	0
tidak	0	0	0	0	0	2	0	1	0	0	0

Gambar 4. 2 Matriks *term-by-doc Unigram*

	0	1	2	3	4	5	6	7	8	9	10
agak	0	0	1	1	0	0	0	0	0	0	0
bagus	0	0	2	0	0	0	1	0	1	0	1
bagus sesuai	0	0	1	0	0	0	0	0	0	0	1
banget	1	0	0	0	0	0	0	1	2	0	0
barang	1	2	1	1	0	1	0	1	1	0	0
cepat	1	1	0	0	0	0	0	1	0	0	0
cepat mantap	0	1	0	0	0	0	0	1	0	0	0
deskripsi	1	0	1	0	0	0	0	0	0	1	0
jual	1	0	0	0	1	0	1	0	1	0	0
kirin	1	1	1	0	0	0	1	1	0	0	0
kirin cepat	1	1	0	0	0	0	0	1	0	0	0
lama	0	0	1	0	1	0	1	0	0	0	0
mantap	0	2	0	0	0	0	0	2	0	0	1
mantap kirin	0	1	0	0	0	0	0	1	0	0	0
minus	0	0	0	0	0	1	1	0	0	0	0
oke	0	0	1	0	0	0	0	1	2	1	0
pokok	0	0	0	0	0	0	0	1	1	0	0
respon	0	0	0	0	1	0	1	0	0	0	0
sesuai	1	0	1	0	0	0	0	0	0	1	1
sesuai deskripsi	1	0	1	0	0	0	0	0	0	1	0
suara	1	0	0	1	0	0	0	1	0	0	0
suka	1	0	1	0	0	0	0	0	1	0	0
tidak	0	0	0	0	0	2	0	1	0	0	0

Gambar 4. 3 Matriks *term-by-doc Kombinasi*

4.5 Pembobotan TF-IDF

Selanjutnya akan dilakukan pembobotan kata menggunakan skema *Term Weighting* TF-IDF. Untuk mempermudah contoh dalam menginterpretasikan tahapan dalam proses pembobotan TF-IDF, matriks *term-by-doc* yang digunakan adalah matriks *unigram* pada Gambar 4.3. Kata yang terseleksi dari matriks *term-by-doc* adalah :

'agak', 'bagus', 'banget', 'barang', 'cepat', 'deskripsi', 'jual', 'kirin', 'lama', 'mantap', 'minus', 'oke', 'pokok', 'respon', 'sesuai', 'suara', 'suka', 'tidak'

Berikut merupakan *syntax* dari proses pembobotan TF-IDF:

```
tfidf=TfidfVectorizer(min_df=2, smooth_idf=False,
                      norm='l1')
X=tfidf.fit_transform(data_clean['Review Clean'])
dtm=pd.DataFrame(X.toarray(),
                 columns=tfidf.get_feature_names())
```

Perhitungan *term frequency* atau TF dengan menggunakan rumus pada persamaan (2.1) dari ulasan ke-1 dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Tabel TF Ulasan Pertama

Kata	<i>tf</i>
agak	0
bagus	0
banget	1
barang	1
cepat	1
deskripsi	1
jual	1
kirin	1
lama	0
mantap	0
minus	0
oke	0
pokok	0
respon	0
sesuai	1
suara	1
suka	1
tidak	0

Perhitungan dari *Inverse Document Frequency* atau IDF setiap kata dalam matriks *term-by-doc* pada data contoh dengan menggunakan persamaan (2.2) tertera pada Tabel 4.10. Kolom “*Doc_j*” menunjukkan jumlah dokumen yang didalamnya terdapat kata tersebut.

Tabel 4. 10 Tabel IDF Ulasan Pertama

Kata	<i>Doc_j</i>	<i>Idf_j</i>
agak	2	2.704748092
bagus	4	2.011600912
banget	3	2.299282984
barang	7	1.451985124
cepat	3	2.299282984
deskripsi	3	2.299282984
jual	4	2.011600912
kirim	5	1.78845736
lama	3	2.299282984
mantap	3	2.299282984
minus	2	2.704748092
oke	4	2.011600912
pokok	2	2.704748092
respon	2	2.704748092
sesuai	4	2.011600912
suara	3	2.299282984
suka	3	2.299282984
tidak	2	2.704748092

Dengan menggunakan kombinasi nilai TF dari Tabel 4.9 dan nilai IDF Tabel 4.10 maka nilai pembobotan TF-IDF awal dari ulasan pertama ditunjukkan pada Tabel 4.11. Selanjutnya dilakuakn normalisasi bobot $w(p_{ra})$ dengan persamaan (2.3) dengan menghitung pembagi nilai bobot awal sebagai berikut :

$$\sum_{j=1}^n w_{i,j} = 0 + 0 + 2.299282984 + 1.451985124 + 2.299282984$$

$$+ 2.299282984 + 2.011600912 + 1.78845736 + 0 + 0 + 0$$

$$+ 0 + 0 + 0 + 2.011600912 + 2.299282984 + 2.299282984 + 0$$

$$= 18.76005923$$

Sehingga hasil TF-IDF setelah dilakukan normalisasi untuk ulasan pertama seperti yang ada pada Tabel 4.12.

Tabel 4. 11 Tabel TF-IDF Ulasan Pertama

Kata	tf	idfj	w(pra)
agak	0	2.704748092	0
bagus	0	2.011600912	0
banget	1	2.299282984	2.299282984
barang	1	1.451985124	1.451985124
cepat	1	2.299282984	2.299282984
deskripsi	1	2.299282984	2.299282984
jual	1	2.011600912	2.011600912
kirim	1	1.78845736	1.78845736
lama	0	2.299282984	0
mantap	0	2.299282984	0
minus	0	2.704748092	0
oke	0	2.011600912	0
pokok	0	2.704748092	0
respon	0	2.704748092	0
sesuai	1	2.011600912	2.011600912
suara	1	2.299282984	2.299282984
suka	1	2.299282984	2.299282984
tidak	0	2.704748092	0

Sehingga apabila dilakukan perhitungan pada seluruh ulasan akan diperoleh matriks *term-by-document* hasil pembobotan seperti pada Gambar 4.4

Tabel 4. 12 Tabel TF-IDF Hasil Normalisasi Ulasan Pertama

Kata	$w(pra)$	w
agak	0	0
bagus	0	0
banget	2.299282984	0.122562672
barang	1.451985124	0.077397683
cepat	2.299282984	0.122562672
deskripsi	2.299282984	0.122562672
jual	2.011600912	0.107227855
kirim	1.78845736	0.095333247
lama	0	0
mantap	0	0
minus	0	0
oke	0	0
pokok	0	0
respon	0	0
sesuai	2.011600912	0.107227855
suara	2.299282984	0.122562672
suka	2.299282984	0.122562672
tidak	0	0

	0	1	2	3	4	5	6	7	8	9	10
agak	0.000000	0.000000	0.129479	0.418950	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
bagus	0.000000	0.000000	0.192595	0.000000	0.000000	0.000000	0.148782	0.000000	0.105314	0.000000	0.318166
banget	0.122563	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.103768	0.240750	0.000000	0.000000
barang	0.077398	0.250552	0.069508	0.224904	0.000000	0.151782	0.000000	0.065529	0.076016	0.000000	0.000000
cepat	0.122563	0.198380	0.000000	0.000000	0.000000	0.000000	0.000000	0.103768	0.000000	0.000000	0.000000
deskripsi	0.122563	0.000000	0.110069	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.363668	0.000000
jual	0.107228	0.000000	0.000000	0.000000	0.286731	0.000000	0.148782	0.000000	0.105314	0.000000	0.000000
kirim	0.095333	0.154307	0.085615	0.000000	0.000000	0.000000	0.132278	0.080714	0.000000	0.000000	0.000000
lama	0.000000	0.000000	0.110069	0.000000	0.327737	0.000000	0.170060	0.000000	0.000000	0.000000	0.000000
mantap	0.000000	0.398781	0.000000	0.000000	0.000000	0.000000	0.000000	0.207536	0.000000	0.000000	0.363668
minus	0.000000	0.000000	0.000000	0.000000	0.000000	0.282739	0.200049	0.000000	0.000000	0.000000	0.000000
oke	0.000000	0.000000	0.096297	0.000000	0.000000	0.000000	0.000000	0.090785	0.210828	0.318166	0.000000
pokok	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.122067	0.141603	0.000000	0.000000
respon	0.000000	0.000000	0.000000	0.000000	0.385532	0.000000	0.200049	0.000000	0.000000	0.000000	0.000000
sesuai	0.107228	0.000000	0.096297	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.318166	0.318166
suara	0.122563	0.000000	0.000000	0.356146	0.000000	0.000000	0.000000	0.103768	0.000000	0.000000	0.000000
suka	0.122563	0.000000	0.110069	0.000000	0.000000	0.000000	0.000000	0.000000	0.120375	0.000000	0.000000
tidak	0.000000	0.000000	0.000000	0.000000	0.000000	0.565478	0.000000	0.122067	0.000000	0.000000	0.000000

Gambar 4. 4 Matriks *doc-by-term* Hasil Pembobotan

Selanjutnya akan dicari matriks singular kanan U dan matriks singular kiri V^T dengan hasil sebagai berikut :

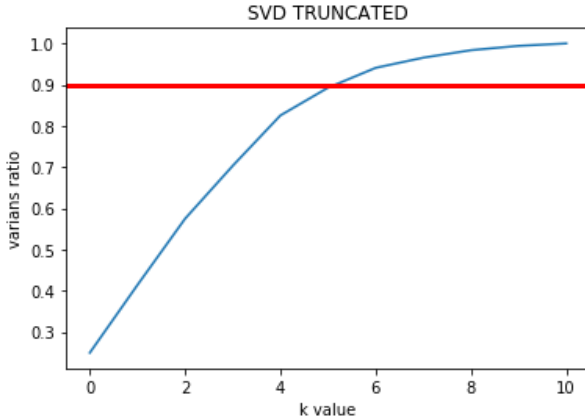
$$U = \begin{bmatrix} -0.245 & 0.035 & -0.039 & 0.119 & -0.136 & 0.295 & 0.055 & -0.361 & -0.820 & 0.109 & 0.028 \\ -0.406 & 0.070 & 0.231 & 0.071 & 0.526 & 0.307 & 0.406 & -0.229 & 0.285 & 0.146 & -0.285 \\ -0.286 & 0.073 & -0.121 & 0.077 & -0.177 & -0.175 & -0.219 & -0.472 & 0.344 & 0.460 & 0.479 \\ -0.257 & -0.114 & 0.281 & 0.805 & -0.245 & -0.276 & 0.005 & 0.165 & 0.025 & -0.106 & -0.137 \\ -0.178 & -0.451 & -0.661 & 0.123 & 0.113 & 0.030 & 0.235 & 0.381 & -0.034 & 0.311 & 0.036 \\ -0.258 & -0.640 & 0.462 & -0.451 & -0.241 & -0.129 & 0.008 & 0.028 & -0.039 & 0.148 & -0.086 \\ -0.214 & -0.325 & -0.351 & -0.026 & 0.090 & -0.133 & -0.127 & -0.454 & 0.098 & -0.675 & -0.106 \\ -0.299 & -0.028 & 0.183 & -0.002 & 0.119 & 0.355 & -0.021 & 0.329 & 0.016 & -0.367 & 0.702 \\ -0.225 & 0.032 & -0.098 & 0.005 & -0.177 & 0.526 & -0.657 & 0.167 & 0.173 & 0.075 & -0.369 \\ -0.341 & 0.379 & -0.185 & -0.243 & -0.606 & 0.013 & 0.450 & 0.111 & 0.152 & -0.170 & -0.118 \\ -0.476 & 0.334 & -0.036 & -0.220 & 0.351 & -0.523 & -0.279 & 0.257 & -0.252 & 0.024 & -0.086 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.174 & -0.057 & 0.153 & 0.583 & -0.216 & -0.320 & -0.072 & 0.031 & 0.245 & 0.088 & 0.036 \\ -0.316 & 0.112 & -0.146 & -0.098 & 0.124 & -0.381 & -0.601 & -0.226 & 0.084 & 0.021 & 0.080 \\ -0.139 & 0.014 & -0.014 & 0.026 & -0.081 & 0.463 & -0.421 & 0.115 & -0.256 & -0.039 & -0.098 \\ -0.331 & -0.144 & 0.276 & 0.244 & 0.020 & 0.161 & 0.115 & -0.163 & 0.204 & 0.336 & -0.487 \\ -0.171 & 0.023 & 0.090 & 0.048 & 0.172 & 0.311 & 0.233 & -0.213 & -0.189 & 0.025 & 0.155 \\ -0.224 & 0.222 & -0.128 & -0.109 & -0.442 & 0.050 & 0.401 & -0.215 & -0.034 & 0.013 & 0.105 \\ -0.160 & -0.253 & -0.384 & 0.075 & 0.022 & 0.176 & -0.040 & 0.078 & -0.289 & 0.049 & -0.325 \\ -0.197 & -0.037 & -0.015 & 0.043 & 0.128 & 0.166 & 0.084 & -0.551 & 0.042 & -0.273 & 0.333 \\ -0.152 & -0.289 & -0.434 & 0.075 & 0.057 & -0.074 & 0.086 & -0.017 & 0.194 & 0.222 & 0.366 \\ -0.478 & 0.213 & 0.174 & -0.087 & 0.621 & 0.012 & 0.152 & 0.272 & 0.110 & -0.056 & 0.009 \\ -0.140 & -0.365 & 0.090 & -0.223 & -0.086 & -0.146 & -0.063 & -0.317 & 0.039 & -0.547 & -0.359 \\ -0.254 & 0.195 & -0.112 & -0.116 & -0.407 & 0.302 & -0.050 & 0.210 & 0.532 & -0.161 & -0.042 \\ -0.082 & 0.002 & 0.013 & 0.001 & -0.018 & 0.273 & -0.262 & 0.245 & 0.118 & -0.201 & 0.263 \\ -0.135 & -0.354 & -0.488 & 0.071 & 0.106 & -0.034 & 0.179 & 0.214 & 0.030 & -0.088 & -0.058 \\ -0.378 & 0.352 & -0.129 & -0.213 & -0.194 & -0.342 & 0.107 & 0.126 & -0.387 & 0.056 & -0.125 \\ -0.184 & -0.058 & 0.171 & 0.506 & -0.158 & -0.058 & 0.017 & 0.187 & -0.402 & -0.367 & 0.216 \\ -0.107 & 0.024 & -0.045 & 0.040 & -0.099 & 0.186 & -0.264 & -0.292 & -0.187 & 0.429 & 0.093 \\ -0.220 & -0.542 & 0.425 & -0.429 & -0.210 & -0.068 & 0.006 & 0.215 & -0.089 & 0.228 & 0.291 \end{bmatrix}$$

Setelah didekomposisi menjadi matriks U, Σ, V^T selanjutnya akan dilakukan pengurangan dimensi menggunakan *truncated SVD* atau pemilihan k komponen dimana $1 \leq k \leq 11$. Pemilihan k dilakukan sesuai dengan persamaan (2.5) dengan nilai rasio 80%. Berikut syntax dari pemilihan nilai k untuk proses *truncated*:

```
def plot_sum_s(s, rasio, title_plot):
    sum_s=np.sum(np.square(s))
    cumsum_s=np.cumsum(np.square(s))
    plot_var=cumsum_s/sum_s
    for i,v in enumerate(plot_var):
        if v>=rasio:
            trun_k=i
            break
    plt.plot(plot_var)
    plt.title(title_plot)
    plt.ylabel('varians ratio')
    plt.xlabel('k value')
    plt.axhline(linewidth=3,color='r', y=rasio)
    plt.show()
    return trun_k
```

Hasil dari plot nilai singular pada matriks Σ dapat dilihat pada Gambar 4.5 dimana jika diambil rasio sebesar 90% maka diperoleh nilai k untuk proses truncated yaitu 6.



Gambar 4. 5 Plot Nilai Singular Matriks Σ

Setelah penentuan nilai k dilakukan proses *truncated* dengan menggunakan *library* scikit-learn dengan *syntax* sebagai berikut :

```
def truncated_svd(matriks, num_k):
    svd = skd.TruncatedSVD(n_components=num_k)
    X_transformed = svd.fit_transform(matriks)
    U = X_transformed / svd.singular_values_
    Sigma_matrix = np.diag(svd.singular_values_)
    VT = svd.components_
    return U, Sigma_matrix, VT
```

Hasil *truncated* SVD diambil matriks V yaitu matriks *doc-by-dimension* dengan ukuran 11×6 seperti pada Gambar 4.6.

	Dim 1	Dim 2	Dim 3	Dim 4	Dim 5	Dim 6
Ulasan 1	0.244942	-0.034963	0.039179	0.118541	-0.136197	0.294789
Ulasan 2	0.406028	-0.070183	-0.230738	0.071443	0.525623	0.307266
Ulasan 3	0.286203	-0.072764	0.120844	0.077485	-0.178902	-0.175007
Ulasan 4	0.256884	0.114054	-0.280642	0.805173	-0.244550	-0.275769
Ulasan 5	0.178260	0.450900	0.660734	0.123029	0.113151	0.030402
Ulasan 6	0.258402	0.640372	-0.461690	-0.451237	-0.241067	-0.128894
Ulasan 7	0.214313	0.324862	0.351406	-0.026147	0.090032	-0.132559
Ulasan 8	0.299019	0.027722	-0.182887	-0.002323	0.118612	0.355298
Ulasan 9	0.225309	-0.032360	0.098229	0.004783	-0.176848	0.526101
Ulasan 10	0.340586	-0.378581	0.184988	-0.242625	-0.605656	0.013271
Ulasan 11	0.476025	-0.333897	0.036439	-0.219558	0.350552	-0.523169

Gambar 4. 6 Matriks *doc-by-dimension* Hasil *Truncated*

4.7 Clustering dengan *Fuzzy C-Means*

Clustering ulasan digunakan untuk mengelompokkan data ulasan yang memiliki kemiripan dimana hasil dari proses ini akan digunakan sebagai nilai awal dari proses klasifikasi dalam metode ANFIS. Algoritma *clustering* yang digunakan adalah *Fuzzy C-Means* (FCM). Matriks yang akan dijadikan input dalam proses *cluster* merupakan matriks hasil *truncated* pada Gambar 4.6. Berikut langkah-langkah proses *cluster* dari data contoh yang digunakan :

a. Tahap pertama adalah tentukan :

- 1) Matriks data yang digunakan merupakan matriks pada Gambar 4.6 yang berukuran 11x6 dengan 11 merupakan jumlah ulasan dan 6 merupakan dimensi atau digunakan sebagai atribut dalam proses *clustering*.
- 2) Jumlah *cluster* yang digunakan misalkan digunakan $C = 2$, pemilihan jumlah ini dikarenakan jumlah data yang sedikit sehingga jika jumlah *cluster* terlalu banyak pengelompokan data kurang tampak
- 3) Pangkat pembobot $w = 2$, merupakan parameter pembobotan yang disarankan pada *clustering* dengan fuzzy [17]
- 4) Maksimum Iterasi $epoch_{max} = 1000$
- 5) Kriteria Penghentian $\xi = 10^{-6}$

b. Menentukan matriks partisi U_0 dengan generate secara random.

Misalkan matriks U_0 awal adalah sebagai berikut :

	1	2	3	4	5	6	7	8	9	10	11
1	0.49	0.39	0.79	0.58	0.49	0.70	0.85	0.56	0.47	0.38	0.29
2	0.51	0.61	0.21	0.42	0.51	0.30	0.15	0.44	0.53	0.62	0.71

Dimana μ_{ij} menunjukkan derajat keanggotaan data ke- j pada cluster ke- i .

c. Selanjutnya dengan persamaan (2.2) akan dihitung matriks V yang merupakan pusat cluster ke- i atribut ke- j . Untuk contoh penghitungan data pada atribut 1 cluster ke 1 sebagai berikut :

$$v_{11} = ((0.49)^2(0.245) + (0.39)^2(0.406) + (0.79)^2(0.286) + (0.58)^2(0.257) + (0.49)^2(0.178) + (0.70)^2(0.258) + (0.85)^2(0.214) + (0.56)^2(0.299) + (0.47)^2(0.225) + (0.38)^2(0.341) + (0.29)^2(0.476)) : (((0.49)^2 + (0.39)^2 + (0.79)^2 + (0.58)^2 + (0.49)^2 + (0.70)^2 + (0.85)^2 + (0.56)^2 + (0.47)^2 + (0.38)^2 + (0.29)^2) = 0.26423216$$

Sehingga diperoleh matriks V sebagai berikut :

	1	2	3	4	5	6
1	0.26423216	0.153127	0.038304	0.027143	-0.06447	-0.01269
2	0.324599437	-0.05997	0.038294	-0.00433	0.013912	0.030778

d. Lalu akan dilakukan perbaikan derajat keanggotaan cluster dari setiap data menggunakan persamaan (2.11) dan diperoleh U_1 sebagai berikut :

	1	2	3	4	5	6	7	8	9	10	11
1	0.44	0.42	0.48	0.54	0.57	0.58	0.63	0.42	0.46	0.45	0.42
2	0.56	0.58	0.52	0.46	0.43	0.42	0.37	0.58	0.54	0.55	0.58

e. Kemudian akan dilakukan pengecekan $\max(|U_1 - U_0|)$, nilai maksimum dari selisih absolut U_1 dan U_0 adalah 0.451, karena nilainya masih lebih besar dari ξ maka langkah ke c,d,e akan diulangi hingga iterasi maksimum atau nilai eror terpenuhi.

f. Setelah dilakukan iterasi sebanyak 17 kali diperoleh nilai matriks partisi U sebagai berikut :

	1	2	3	4	5	6	7	8	9	10	11
1	0.155	0.354	0.811	0.530	0.588	0.517	0.722	0.159	0.252	0.518	0.604
2	0.845	0.646	0.189	0.470	0.412	0.483	0.278	0.841	0.748	0.482	0.396

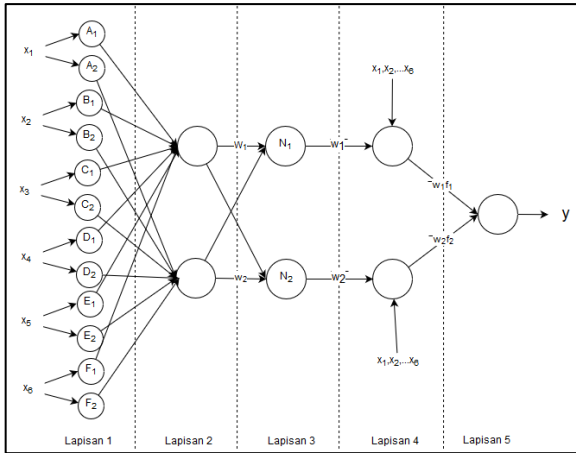
Dari setiap data dapat dicari nilai keanggotaan terbesar setiap data berada di cluster mana. Sehingga dari matriks U iterasi terakhir maka cluster setiap data tertulis pada Tabel 4.13.

Tabel 4. 13 Hasil *Cluster* Ulasan

No	Ulasan Bersih	Cluster
1	barang sesuai deskripsi jual rekomendasi banget suka kirim cepat suara keras	1
2	barang mantap kirim cepat mantap barang komplain	1
3	barang bagus sesuai deskripsi kirim agak lama suka oke bagus	0
4	lumayan barang suara mic agak aneh	0
5	pesan warna hitam datang silver baterai gampang drop respon jual lama	0
6	barang banyak minus tidak guna tidak original	0
7	packing kirim cukup bagus minus jual respon lama	0
8	barang oke banget tidak nyesel beli suara mantap kirim cepat mantap pokok	1
9	barang super bagus pakai banget oke pokok suka banget jual oke	1
10	sesuai deskripsi oke	0
11	kualitas produk mantap bagus sesuai harga	0

4.8 Adaptive Neuro Fuzzy Inference System (ANFIS)

Model ANFIS akan digunakan dalam proses prediksi nilai dari teks ulasan. Dari proses *clustering* pada tahapan sebelumnya sebagai input awal parameter premis dalam aturan Sugeno yang akan digunakan dalam aturan Sugeno dimana jumlah aturan yang digunakan sesuai dengan jumlah *cluster* yang didefinisikan. Arsitektur dari jaringan ANFIS yang digunakan dapat dilihat pada Gambar 4.7 yang mana aturan pada fungsi aktivasi di lapisan 1 pada masing-masing atribut, jumlah node pada lapisan 2,3, dan 4 sesuai dengan jumlah *cluster* yang digunakan. x_1, x_2, x_3, x_4, x_5 , dan x_6 merupakan atribut dimensi 1 hingga 6 pada matriks V_k hasil *truncated* SVD pada Gambar 4.6.



Gambar 4. 7 Arsitektur ANFIS pada Data Contoh

Sebagaimana dijelaskan pada sub bab 3.4 akan dicari nilai mean dan standar deviasi dari setiap atribut pada masing-masing *cluster*. Nilai mean dan standar dari setiap data pada matriks *term-by-dim* pada Gambar 4.6 ada pada Tabel 4.14.

Tabel 4. 14 Nilai Mean Setiap Atribut pada *Cluster*

Mean	1	2	3	4	5	6
<i>Cluster 1</i>	0.287	0.106	0.087	0.009	-0.102	-0.170
<i>Cluster 2</i>	0.294	-0.027	-0.069	0.048	0.083	0.371

Tabel 4. 15 Nilai Deviasi Standar Setiap Atribut pada *Cluster*

Standar Deviasi	1	2	3	4	5	6
<i>Cluster 1</i>	0.098	0.390	0.376	0.404	0.313	0.188
<i>Cluster 2</i>	0.081	0.041	0.162	0.058	0.323	0.107

Dari data contoh yang digunakan dengan jumlah *cluster* adalah 2 maka didapat 2 aturan Sugeno yaitu:

Jika x_1 adalah A_1, x_2 adalah B_1, x_3 adalah C_1, x_4 adalah D_1, x_5 adalah E_1 dan x_6 adalah F_1 maka $y_1 = c_{11}x_1 + c_{12}x_2 + c_{13}x_3 + c_{14}x_4 + c_{15}x_5 + c_{16}x_6 + c_{10}$

Jika x_1 adalah A_2 , x_2 adalah B_2 , x_3 adalah C_2 , x_4 adalah D_2 , x_5 adalah E_2 dan x_6 adalah F_2 maka $y_1 = c_{21}x_1 + c_{22}x_2 + c_{23}x_3 + c_{24}x_4 + c_{25}x_5 + c_{26}x_6 + c_{20}$

Sehingga didapat rata-rata terbobot sesuai pada persamaan (2.13). Fungsi aktivasi yang digunakan adalah *generalized bell* sesuai dengan persamaan (2.14) dimana nilai a dan c sesuai dengan Tabel 4.14 dan 4.15. Untuk memudahkan dalam pemberian contoh maka akan dituliskan fungsi keanggotaan yang ada pada neuron 1 berdasarkan persamaan (2.14).

$$\begin{aligned}\mu_{A_1}(x_1) &= \frac{1}{1 + \left| \frac{x_1 - 0.287}{0.098} \right|} & \mu_{B_1}(x_2) &= \frac{1}{1 + \left| \frac{x_2 - 0.106}{0.039} \right|} \\ \mu_{C_1}(x_3) &= \frac{1}{1 + \left| \frac{x_3 - 0.087}{0.376} \right|} & \mu_{D_1}(x_4) &= \frac{1}{1 + \left| \frac{x_4 - 0.009}{0.404} \right|} \\ \mu_{E_1}(x_5) &= \frac{1}{1 + \left| \frac{x_5 - (-0.102)}{0.313} \right|} & \mu_{F_1}(x_6) &= \frac{1}{1 + \left| \frac{x_6 - (-0.17)}{0.188} \right|}\end{aligned}$$

Selanjutnya seperti arsitektur ANFIS pada Gambar 4.7 dan telah dijelaskan *output* setiap lapisan pada Sub Bab 2.9 akan dihitung *output* lapisan pertama yang berupa derajat keanggotaan setiap data dimana hasilnya terdapat pada Tabel 4.16 dimana jumlah *output* pada lapisan 1 sebanyak jumlah atribut dikalikan jumlah *cluster*.

Tabel 4. 16 *Output* Lapisan 1

Data	μ_{A_1}	μ_{B_1}	μ_{C_1}	μ_{D_1}	μ_{E_1}	μ_{F_1}	μ_{A_2}	μ_{B_2}	μ_{C_2}	μ_{D_2}	μ_{E_2}	μ_{F_2}
1	0.843	0.733	0.884	0.967	0.984	0.692	0.932	0.400	0.988	0.685	0.141	0.663
2	0.404	0.343	0.830	0.475	0.583	0.501	0.977	0.859	0.199	0.347	0.135	0.738
3	1.000	0.991	0.826	0.446	0.992	0.421	0.972	0.793	0.946	0.607	0.999	0.037
4	0.912	0.828	1.000	0.076	0.511	0.370	0.205	0.006	0.828	0.493	0.761	0.027
5	0.446	0.330	0.562	0.007	0.301	0.047	0.927	0.371	0.679	0.991	0.468	0.089
6	0.920	0.840	0.348	0.004	0.319	0.146	0.435	0.013	0.835	0.498	0.954	0.044
7	0.643	0.509	0.761	0.013	0.670	0.129	0.992	0.375	0.726	0.999	0.961	0.043
8	0.986	0.996	0.961	0.352	0.659	0.670	0.999	0.565	0.668	0.988	0.114	0.979
9	0.714	0.583	0.888	0.986	0.999	0.484	1.000	0.638	0.946	0.607	0.068	0.321
10	0.771	0.750	0.393	0.013	0.937	0.289	0.720	0.038	0.278	0.180	0.513	0.082
11	0.212	0.165	0.440	0.017	0.982	0.702	0.757	0.044	0.323	0.593	0.222	0.014

Pada lapisan kedua akan dikalikan derajat keanggotaan yaitu: $w_1 = (\mu_{A1})(\mu_{B1})(\mu_{C1})(\mu_{D1})(\mu_{E1})(\mu_{F1})$ begitu pula untuk w_2 . *Output* pada lapisan ke-2 seperti yang terlihat pada Tabel 4.17

Tabel 4. 17 *Output* Lapisan 2

	w_1	w_2
1	0.140884	0.400119
2	0.134595	0.342799
3	0.825725	0.036820
4	0.205091	0.005740
5	0.300861	0.007160
6	0.319280	0.003687
7	0.642874	0.013121
8	0.113789	0.351576
9	0.068151	0.320966
10	0.278348	0.013207
11	0.211719	0.014051

Pada lapisan ke tiga setiap bobot dinormalisasi untuk mendapatkan nilai \bar{w}_1 dan \bar{w}_2 . Sesuai dengan persamaan (2.17) maka diperoleh output lapisan ke 3 seperti pada Tabel 4.18.

Tabel 4. 18 *Output* Lapisan 3

	\bar{w}_1	\bar{w}_2
1	0.260412	0.739588
2	0.281937	0.718063
3	0.957312	0.042688
4	0.972777	0.027223
5	0.976754	0.023246
6	0.988585	0.011415
7	0.979999	0.020001
8	0.244515	0.755485
9	0.175143	0.824857
10	0.954700	0.045300
11	0.937764	0.062236

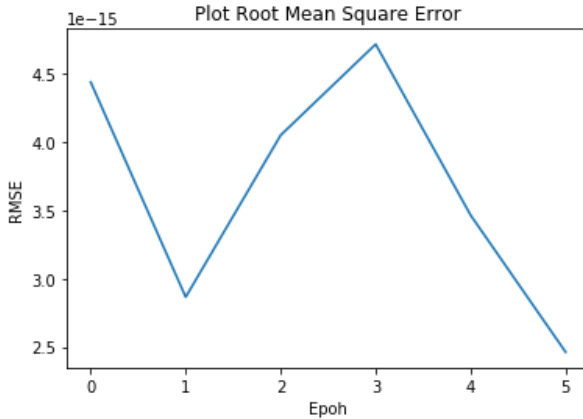
Terakhir pada lapisan keempat dapat dihitung nilai $\bar{w}_i x_{ij}$ dengan $i = 1,2$ dan $j = 1,2, \dots, 6$ sebagai koefisien dari c_{ij} , $i = 1,2$ dan $j = 0,1,2,3,4,5,6$ pada persamaan (2.8). Hasilnya dapat dilihat pada Tabel 4.19 (untuk hasil lengkap dapat dilihat pada Lampiran 3).

Dengan menggunakan metode RLSE seperti yang telah dijelaskan pada sub bab 2.10 maka didapat parameter $c_{11}, c_{12}, \dots, c_{20}$ yaitu [-21.392044, -7.292261, -1.752757, -1.446219, 2.268875, -1.442984, 10.038580, -1.058757, 6.056786, 1.791536, 3.310070, 2.277861, 0.823378, 5.400251].

Sehingga diperoleh output nilai prediksi y atau \hat{y} dari lapisan ke empat dan ke lima sebagai berikut [4.9999999999999964, 4.9999999999999994, 4.0000000000000036, 2.999999999999997, 2.0000000000000013, 0.9999999999999928, 3.0000000000000004, 4.999999999999995, 4.999999999999995, 4.0000000000000004, 3.9999999999999964] dari hasil iterasi pertama. Selanjutnya dilakukan *epoch* sebanyak 5 kali dan dapat dilihat nilai *Root Mean Square Error* (RMSE) dari setiap *epoch* pada Gambar 4.8. Untuk *syntax* lengkap pada proses FCM-ANFIS diberikan pada Lampiran 4.

Tabel 4. 19 Koefisien Parameter $\bar{w}_i x_{ij}$

	c_{11}	c_{12}	c_{25}	c_{26}	c_{20}
1	0.063786	-0.009105	-0.100729	0.218022	0.739588
2	0.114474	-0.019787	0.377430	0.220636	0.718063
3	0.273985	-0.069658	-0.007552	-0.007471	0.042688
4	0.249871	0.110949	-0.006657	-0.007507	0.027223
5	0.174116	0.440418	0.002630	0.000707	0.023246
6	0.255452	0.633063	-0.002752	-0.001471	0.011415
7	0.210027	0.318364	0.001801	-0.002651	0.020001
8	0.073114	0.006779	0.089609	0.268423	0.755485
9	0.039461	-0.005668	-0.145875	0.433958	0.824857
10	0.325157	-0.361431	-0.027436	0.000601	0.045300
11	0.446400	-0.313116	0.021817	-0.032560	0.062236



Gambar 4. 8 Plot RMSE Setiap Epoch

Dari grafik pada Gambar 4.8 dapat dilihat bahwa setelah epoch ke-6 RMSE dari model sudah sangat kecil . Begitu pula pada Gambar 4.8

Hasil perbandingan nilai y dan \hat{y} setelah epoch ke-5 dapat dilihat sebagai berikut :

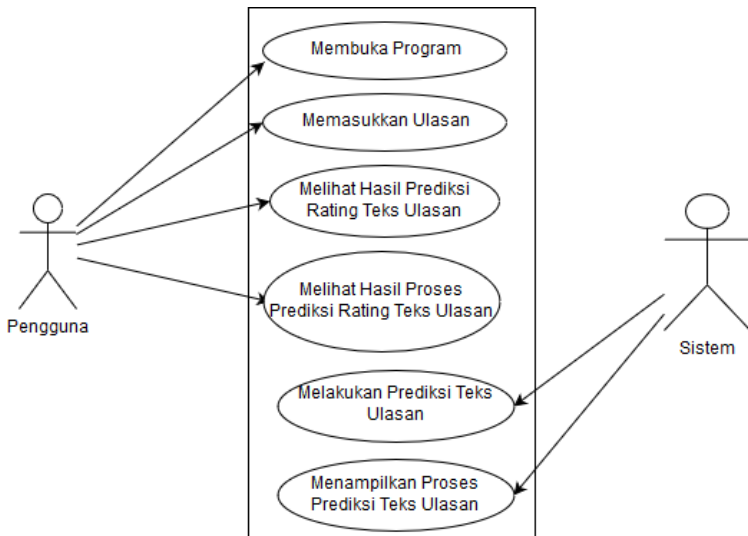
Ulasan ke	y	\hat{y}
1	5	4.999999999999996
2	5	4.999999999999997
3	4	4.0
4	3	3.0000000000000004
5	2	1.999999999999998
6	1	0.9999999999999992
7	3	3.0000000000000004
8	5	4.999999999999996
9	5	4.9999999999999964
10	4	3.999999999999999
11	4	3.9999999999999987

4.9 Analisis Implementasi Sistem

Perancangan sistem yang selanjutnya diimplementasikan dalam suatu program yang mana dalam proses pembuatannya diperlukan analisis agar sistem yang dibuat berjalan sebagaimana seharusnya. Analisis suatu sistem dalam implementasi berbasis perangkat lunak, dapat digambarkan dalam *use case diagram* dan *activity diagram*.

4.9.1 Use Case Diagram

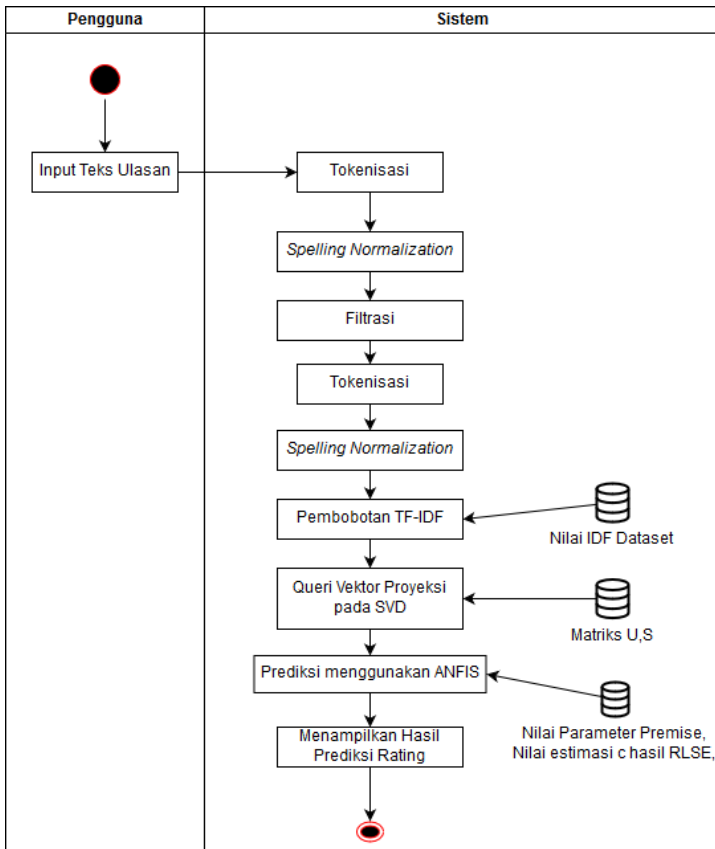
Use case diagram merupakan suatu diagram yang menjelaskan interaksi yang terjadi pada suatu sistem dengan aktor sistem. Adapun *use case diagram* dari Sistem dalam penelitian ini seperti pada Gambar 4.8.



Gambar 4. 9 Use Case Diagram Sistem

4.9.2 Activity Diagram

Untuk menjelaskan proses jalannya sistem dari awal sampai akhir, dapat digunakan *activity diagram* yang menggambarkan cara kerja sistem yang dibuat untuk penelitian ini seperti yang terlihat pada Gambar 4.9



Gambar 4. 10 Activity Diagram Sistem Untuk Prediksi Rating

4.10 Tampilan Antar Muka GUI

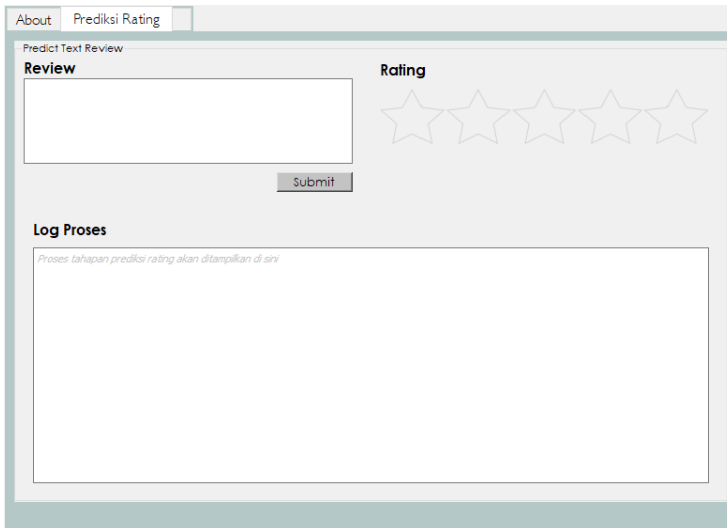
Untuk memudahkan pengguna dalam mengoperasikan sistem maka dirancang desain GUI untuk kemudahan dalam implementasi sistem. Dengan implementasi tersebut pengguna dapat melakukan prediksi nilai ulasan yang diberikan serta mencoba sistem verifikasi ulasan yang diterapkan. Pembuatan GUI menggunakan *library* PyQt5 dengan generator menggunakan aplikasi *designer*. Tampilan GUI dapat dilihat pada Gambar 4.10 yang merupakan tampilan awal program sedangkan untuk sistem prediksi nilai ulasan dapat dilihat pada Gambar 4.12.



Gambar 4. 11 Tampilan Awal GUI

Seperti yang terlihat pada Gambar 4.11 bahwa tampilan pada tab menu *Prediksi Rating* memiliki beberapa komponen. Pada panel bagian atas digunakan *text edit review* digunakan untuk memasukkan teks ulasan oleh. Selanjutnya tombol *submit* digunakan untuk melakukan prediksi nilai dari teks ulasan yang diberikan, dimana pada bagian kanan akan muncul hasil prediksi ulasan sesuai jumlah bintang yang berwarna kuning. Kemudian pada bagian *Log Process* akan menampilkan tahap-tahap yang digunakan dalam proses prediksi nilai

ulasan dari parameter-parameter yang nilainya sudah didapat dari proses pembelajaran. Untuk *syntax* dan hasil *running* dari program untuk prediksi nilai sentimen terdapat pada Lampiran 5.



The image shows a software application window titled "Predict Text Review". At the top, there are two tabs: "About" and "Prediksi Rating". The main interface is divided into three sections:

- Review:** A large text input field for entering text, with a "Submit" button below it.
- Rating:** A section labeled "Rating" containing five empty star icons for rating.
- Log Proses:** A section labeled "Log Proses" containing a large text area with the placeholder text "Proses tahapan prediksi rating akan ditampilkan di sini".

Gambar 4. 12 Tampilan Tab Menu Prediksi *Rating*

BAB V HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dan analisis terhadap hasil yang diperoleh dari proses implementasi penelitian pada data sebenarnya.

5.1 Deskripsi Data

Sebagaimana telah dijelaskan pada bab sebelumnya, bahwa data yang digunakan merupakan data produk dengan kategori elektronik di *website* www.tokopedia.com. Jumlah data ulasan yang diambil adalah 5904 ulasan.

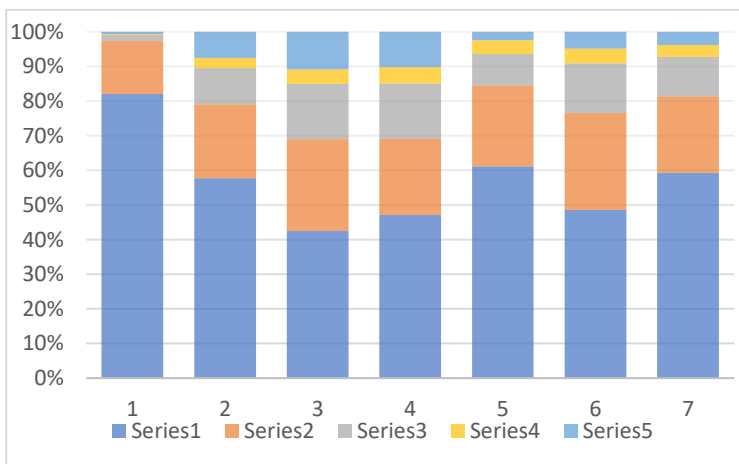
Id	Nama	Review	Rating
11383775	Agung Pangestu Aji	Mantap... maju terus.....	5
2908041	eqi putra	Nice produk... Sesuai etalase	5
16727412	Troy Lee Faredo	Semoga packingnya bisa di tingkatkan lagi walaupun simpel, barang ba	4
21450714	Imam Abdul Malik	Barang pesanan sesuai dan cepat	5
0 A***o		Barang sesuai tp harus teliti kalo pesen di toko ini salah nomer sepatu	3
24528988	Bayu Listianto	oke utk harga 40 ribuan...keluhan kurang ringan	3
2149977	Dodoy Mnf	Oke,,sdh diterima dg kondisi yg baik	5
11053508	maritha devi	Produk ok sesuai deskripsi. Sellernya responsif. Terima kasih.	5
0 R***o		Produk sesuai gambar,,bagus	5
9204005	Willy Kusuma	Barang sesuai deskripsi. Mudah2an awet, cocok pas musin hujan.	5
21131850	Syahril Reza	Barang sudah sampai, sesuai dengan gambar, pengiriman cepat, Recor	5
34645210	Alpian Budi Kusuma	Tksssssssssssssss	5
1344156	Rendy Agus Nur Sasongko	Barang oke. Respon cepat.. Recommended seller..	5
2432293	Tarmo Purgana	Sepatu sudah sampai dan sesuai pesanan. Thanks	5
9574908	Supriadi	mantaapp... ukuran nya pas.	5
4402619	Suryadi	Oke barang bagus sesuai dengan gambar, dan dikirim tepat waktu	4
0 J***g		great vendor very fast delivery	5
33717740	Idi Winarno	Barang sesuai deskripsi. Pengiriman cepat dan sampai tujuan dalam kc	4
8470760	Niesin	Good quality, fast rensponse. TOPP....	5
4699819	Martian	Salah npmorminta ganti gak direspon... berkunjung juga ke toko d	5

Gambar 5. 1 Gambar Kutipan Data Ulasan

Rincian data tiap produk dapat dilihat pada Tabel 5.1, dimana prosentase tiap-tiap *rating* dalam masing-masing produk dapat dilihat pada Gambar 5.2.

Tabel 5. 1 Rincian Jumlah Ulasan Tiap Produk

Produk	Rating					Jumlah
	5	4	3	2	1	
Handsfree Bluetooth 41 Earbud S530	1223	229	25	4	10	1491
Headset Bluetooth I7S	337	125	61	17	44	584
Headset Samsung HS330	141	88	53	14	36	332
Paket Ngebul Tesla Baterai Liquid	253	118	86	25	55	838
Lampu Nyamuk Atau Perangkap Nyamuk	201	77	30	13	8	329
Premium Eliquid Vaporizer	890	512	262	79	88	1831
TV Indoor Antenna DA3920N	296	110	57	17	19	499
Jumlah	3341	1259	574	169	260	5904



Gambar 5. 2 Presentase Tiap Rating dalam Data Ulasan Produk

Dataset yang sudah tersimpan nantinya akan dibagi menjadi data latih dan data uji dengan perbandingan 80% untuk data latih dan 20% untuk data uji.

5.2 Praproses Data

Data ulasan yang didapat dari proses *scraping* pada *e-commerce* Tokopedia akan dilakukan proses pembersihan untuk menghilangkan *noise* pada data. Langkah-langkah pada proses ini seperti yang telah dijelaskan dalam sub bab 4.2 mulai tokenisasi hingga proses *stemming*. Kutipan data hasil praproses ditunjukkan pada Gambar 5.3. Jumlah data hasil pra proses bersisa 5579 data ulasan dari 5904 data hasil *scraping*. Pengurangan data ini terjadi setelah data melalui proses pembersihan dimana data ulasan yang ulasannya kosong atau *null* setelah pembersihan akan dihapus. Ulasan yang terhapus merupakan ulasan yang hanya berisi tanda baca dan *stopwords*.

	Nama	Review	Review Clean	Rating
0	Agung Pangestu Aji	Mantap... maju terus.....	mantap maju terus	5
1	eqi putra	Nioe produk.. Sesuai etalase	bagus produk sesuai etalase	5
2	Troy Lee Faredo	Semoga packingnya bisa di tingkatkan lagi wala... moga packing tingkat walaupun simpel barang ba...		4
3	Imam Abdul Malik	Barang pesanan sesuai dan cepat	barang pesan sesuai cepat	5
4	A***o	Barang sesuai tp harus teliti kalo pesen di to...	barang sesuai teliti sen toko salah nomor sepa...	3
5	Bayu Listianto	oke utk harga 40 ribuan...keluhan kurang ringan	oke harga ribu keluh kurang ringan	3
6	Dodoy Mnf	Oke,,sdh diterima dg kondisi yg baik	oke terima kondisi baik	5
7	maritha devi	Produk ok sesuai deskripsi. Sellernya responsi...	produk oke sesuai deskripsi sellernya respon t...	5
8	R***o	Produk sesuai gambar, bagus	produk sesuai gambar bagus	5
9	Willy Kusuma	Barang sesuai deskripsi. Mudah2an awet, cocokk ...	barang sesuai deskripsi mudah awet cocok pas m...	5

Gambar 5. 3 Kutipan Hasil Praproses Data

5.3 Hasil *N-Gram* dan TF-IDF

Dalam proses perubahan ulasan bersih menjadi matriks *term-by-doc* dan pembobotan TF-IDF akan digunakan skenario *N-Gram* dalam pemilihan fitur yang digunakan seperti yang dijelaskan pada sub bab 4.5 dan 4.6.

a. *Unigram*

Pada hasil TF-IDF untuk *term* tunggal atau *unigram* dengan nilai df adalah 3 sebanyak 933 fitur. Hasil dari proses TF-IDF ini

merupakan matriks *term-by-doc* yang setiap elemen didalamnya sudah dibobotkan seperti yang dijelaskan pada bab sebelumnya. Gambar 5.4 menunjukkan fitur-fitur yang terpilih dalam proses TF-IDF.

Fitur Terpilih		919	wajib
no		920	waktu
1	ada	921	walaupun
2	admin	922	wangi
3	after	923	warna
4	agak	924	waspada
5	agen	925	watermelon
6	air	926	watt
7	airpod	927	well
8	ajib	928	wilayah
9	akhir	929	work
10	akurasi	930	working
11	akurat	931	worth
12	alamat	932	yakin
13	alat	933	yes
14	alhamdulillah		

Gambar 5. 4 Kutipan Fitur Terpilih Hasil Proses TF-IDF *Unigram*

Matriks yang akan digunakan dalam proses selanjutnya berbentuk *term-by-doc* seperti yang dijelaskan dalam bab sebelumnya dimana hasilnya dapat dilihat pada Gambar 5.5 dimana baris menunjukkan *term* dan kolom merupakan indeks dari ulasan.

	ulasan 1	ulasan 2	ulasan 3	ulasan 4	ulasan 5	ulasan 6	ulasan 7	ulasan 8	ulasan 9	ulasan 10	...
ada	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
admin	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
after	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agak	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agen	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
air	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
airpod	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
ajib	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
akhir	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
akurasi	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
akurat	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
alamat	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

Gambar 5. 5 Kutipan Matriks TDM *Unigram*

b. *Bigram*

Pada hasil TF-IDF dalam *bigram* atau pemilihan kata dengan fitur dengan jumlah kata sebanyak 2 memiliki jumlah fitur 1237 terjadinya peningkatan jumlah fitur dibandingkan pada saat pemilihan fitur dengan *unigram* disebabkan oleh banyaknya jenis kombinasi kata yang terjadi di dalam dokumen. Gambar 5.6 merupakan fitur-fitur hasil seleksi TF-IDF dengan minimum df adalah 3.

No	Fitur Terpilih	Fitur Terpilih	
		1228	warna barang
1	ada baik	1229	warna hitam
2	after taste	1230	warna liquid
3	agak aneh	1231	warna merah
4	agak besar	1232	warna putih
5	agak kecewa	1233	warna sesuai
6	agak kurang	1234	warna silver
7	agak lama	1235	warna tidak
8	agak lambat	1236	waspada jangan
9	agak miring	1237	yes sampai

Gambar 5. 6 Kutipan Fitur Terpilih Hasil Proses TF-IDF *Bigram*

Sama seperti pada proses *unigram* matriks yang akan dijadikan masukan merupakan matriks *doc_by_term* seperti yang terlihat pada Gambar 5.7.

	ulasan 1	ulasan 2	ulasan 3	ulasan 4	ulasan 5	ulasan 6	ulasan 7	ulasan 8	ulasan 9	ulasan 10
ada baik	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
after taste	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak aneh	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak besar	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak kecewa	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak kurang	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak lama	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak lambat	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
agak miring	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Gambar 5. 7 Kutipan Matriks TDM *Bigram*

c. Kombinasi

Pada pemilihan fitur dengan kombinasi *unigram bigram* jumlah fitur yang terpilih sebanyak 2514. Jumlah fitur menjadi semakin banyak disebabkan setiap ulasan memiliki *term* dengan pemotongan kata satu dan dua sehingga kombinasi yang diperoleh menjadi semakin banyak. Gambar 5.8 menunjukkan hasil seleksi fitur dalam skenario pemilihan kata dalam kombinasi *unigram bigram*.

No	Fitur Terpilih	2503	waspada
		2504	waspada jangan
1	ada	2505	watermelon
2	ada baik	2506	watt
3	admin	2507	well
4	after	2508	wilayah
5	after taste	2509	work
6	agak	2510	working
7	agak aneh	2511	worth
8	agak besar	2512	yakin
9	agak gede	2513	yes
10	agak kecewa	2514	yes sampai

Gambar 5. 8 Kutipan Fitur Terpilih Hasil Proses TF-IDF *Bigram*

Hasil matriks *doc-by-term* hasil proses TF-IDF dengan pemotongan *n-gram* memiliki rentang (1,2) dapat dilihat pada Gambar 5.9.

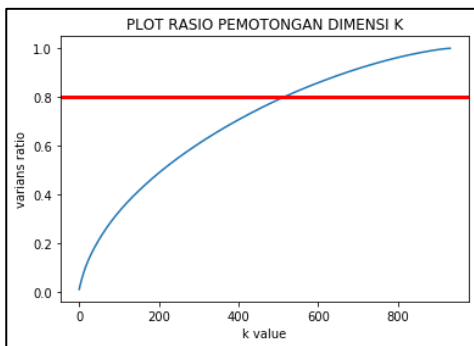
	ulasan 1	ulasan 2	ulasan 3	ulasan 4	ulasan 5	ulasan 6	ulasan 7	ulasan 8	ulasan 9	ulasan 10	...
ada	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
ada baik	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
admin	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
after	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
after taste	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agak	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agak aneh	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agak besar	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
agak gede	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

Gambar 5. 9 Kutipan Matriks TDM Kombinasi

5.4 Hasil *Feature Extraction* dengan LSI

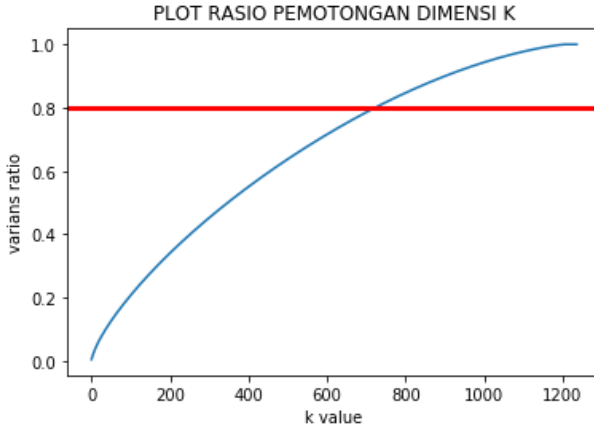
Selanjutnya akan dilakukan *Feature Extraction* dari matriks *doc-by-term* pada tahap sebelumnya dimana akan dilakukan *truncated* SVD dalam proses ini. Seperti yang telah dijelaskan pada bab sebelumnya pemilihan nilai k yang digunakan yaitu rasio 80% dari hasil plot nilai singular.

Hasil dari plot nilai singular dari matriks TF-IDF untuk fitur *unigram* dapat dilihat pada Gambar 5.10 dimana garis merah vertikal menunjukkan pemotongan dimensi k . Nilai k yang digunakan yaitu 514.



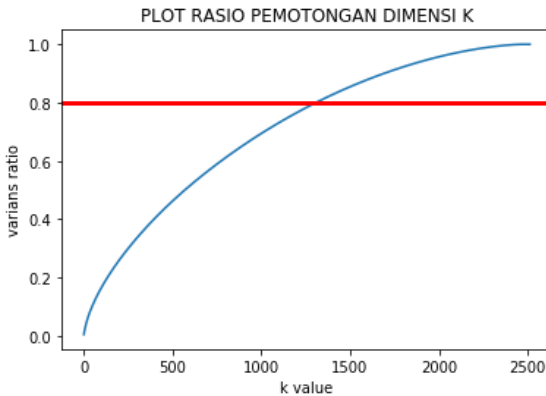
Gambar 5. 10 Plot Rasio Nilai Eigen Skenario *Unigram*

Sedangkan untuk matriks pada Gambar 5.7 untuk pemotongan fitur dengan *bigram* nilai k yang diperoleh adalah 723. Grafik plot rasio nilai singular dapat dilihat pada Gambar 5.11.



Gambar 5. 11 Plot Rasio Nilai Singular Skenario *Bigram*

Pada pemilihan fitur kombinasi pemilihan nilai k untuk proses *Truncated SVD* yaitu sebesar 1307. Plot dari rasio nilai singular matriks *doc-by-term* pada Gambar 5.9 dapat dilihat pada Gambar 5.12



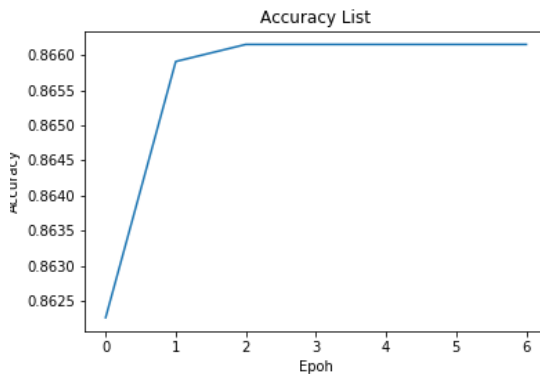
Gambar 5. 12 Plot Rasio Nilai Singular Skenario Kombinasi

5.5 Hasil Prediksi Ulasan dengan FCM-ANFIS

Berikut akan dijelaskan hasil dari proses pembelajaran dan pengujian dari algoritma FCM-ANFIS yang digunakan untuk prediksi nilai ulasan. Input yang digunakan merupakan matriks V hasil truncated sebagaimana dijelaskan dalam bab sebelumnya. Pemilihan jumlah *cluster* didasarkan pada proses percobaan dengan pertimbangan jika jumlah *cluster* dinaikkan maka data akan mengalami *overfitting* dimana pada saat proses pembelajaran nilai akurasi sangat tinggi namun pada saat pengujian nilai akurasi data rendah, hal ini disebabkan karena model mempelajari data terlalu baik namun kurang bagus untuk mengenali data baru yang. Selain itu jumlah *cluster* yang terlalu banyak dapat meningkatkan waktu komputasi karena jumlah parameter konsekuensi dan jumlah aturan pada aturan model Sugeno juga akan bertambah.

a. Skenario *Unigram*

Pada proses ini nilai *cluster* yang dipilih adalah 5 dimana Gambar 5.13 menunjukkan nilai akurasi pada proses pembelajaran tiap *epoch*. Pada *epoch* ke 3 (karena *python* dimulai dari 0) nilai akurasi sudah konstan dan tidak terjadi perubahan dan nilai akurasinya sudah baik oleh karena itu proses pembelajaran digunakan hingga 6 *epoch*.



Gambar 5. 13 Hasil Akurasi Pembelajaran Skenario *Unigram*.

Nilai hasil akurasi dari proses pembelajaran adalah 0.866 dimana nilai RMSE yang diperoleh adalah 0.3984. Sedangkan nilai akurasi untuk proses testing yaitu 0.657. Selanjutnya dilakukan percobaan dimana melakukan toleransi sebesar 1 pada data hasil prediksi yaitu model pengujian yang mentolerir hasil prediksi yang memiliki selisih rentang 1 dimana menunjukkan akurasi sebesar 0.83. Hal ini menunjukkan bahwa sistem banyak salah memprediksi dengan rentang *rating* sebesar 1 yang disebabkan sulit membedakan ulasan antar *rating*. Tabel 5.2 menunjukkan nilai evaluasi dari algoritma dari beberapa variabel pengukuran performansi.

Tabel 5. 2 Hasil Performansi Skenario *Unigram*

	Akurasi	Presisi	Recall
<i>Training</i>	0.87	0.87	0.87
<i>Testing</i> Toleransi 0	0.657	0.68	0.66
<i>Testing</i> Toleransi 1	0.83	0.89	0.84

Penghitungan akurasi, presisi, *recall*, serta matriks *confusion* hasil *testing* sesuai dengan persamaan (2.33), (2.34), dan (2.34) dapat dilihat di Lampiran 6. Tabel 5.4 menunjukkan beberapa ulasan yang salah diprediksi ke nilai *rating* yang berbeda.

Tabel 5. 3 Ulasan yang Salah Prediksi

Ulasan	<i>Rating</i>	Prediksi
barang sudah sampai, terimakasih banyak...	4	5
Ga fungsi sama sekali tuh.	2	3
bagus barangnya....fast respon juga...	4	5
kurang memuaskan sorry	1	2
Banyak channel yg gambarnya tidak bagus	2	1

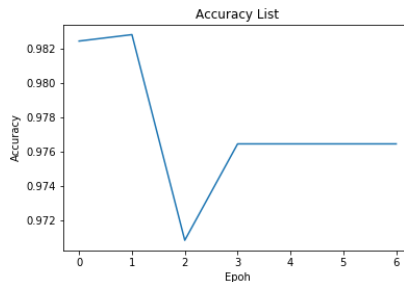
Selain itu juga dilakukan percobaan dengan menggunakan sentimen dari ulasan dimana dilakukan 2 kasus pengelompokan data sentimen yaitu sentimen negatif dengan rating 1 dan 2 sedangkan sentimen positif untuk rating 3,4,dan 5. Untuk kasus ke dua sentimen negatif 1, 2, dan 3 serta 4 dan 5 untuk sentimen positif, Hal ini dilakukan karena ulasan pada rating 3 cukup ambigu untuk dimasukkan sebagai sentimen positif ataupun negatif. Jumlah *cluster* yang digunakan adalah 2. Hasil dari proses pembelajaran dan pengujian dapat dilihat pada Tabel 5.4.

Tabel 5. 4 Hasil Performansi Skenario *Unigram* Sentimen

	Akurasi	Presisi	Recall
<i>Training</i> Sentimen 1	0.967	0.94	0.84
<i>Testing</i> Sentimen 1	0.919	0.92	0.92
<i>Training</i> Sentimen 2	0.933	0.93	0.93
<i>Testing</i> Sentimen 2	0.87	0.87	0.87

b. Skenario *Bigram*

Jumlah *cluster* yang digunakan yaitu 5 dan Gambar 5.14 menunjukkan nilai akurasi pada proses pembelajaran tiap *epoch*. Nilai akurasi dan RMSE dari proses pembelajaran berturut-turut adalah 0.976 dan 0.141. Pada hasil *epoch* ke-4 dan seterusnya nilai akurasi sudah konstan namun dalam proses pembelajaran nilai akurasi sempat naik turun, namun karena dirasa nilai 0.976 masih bagus maka jumlah *epoch* dihentikan pada saat *epoch*=6.



Gambar 5. 14 Hasil Akurasi Pembelajaran Skenario *Bigram*

Sama seperti sebelumnya akan dilakukan evaluasi dengan toleransi 0 dan toleransi 1. Hasil dari evaluasi performansi pada skenario ini dapat dilihat pada Tabel 5.5.

Tabel 5. 5 Hasil Performansi Skenario *Bigram*

	Akurasi	Presisi	Recall
<i>Training</i>	0.976	0.98	0.98
<i>Testing</i> Toleransi 0	0.64	0.80	0.64
<i>Testing</i> Toleransi 1	0.75	0.90	0.75

Sedangkan untuk percobaan sentimen kasus 1 dan kasus 2 seperti sebelumnya dapat dilihat pada Tabel 5.6 dengan pengambilan jumlah *cluster* yang digunakan adalah 2.

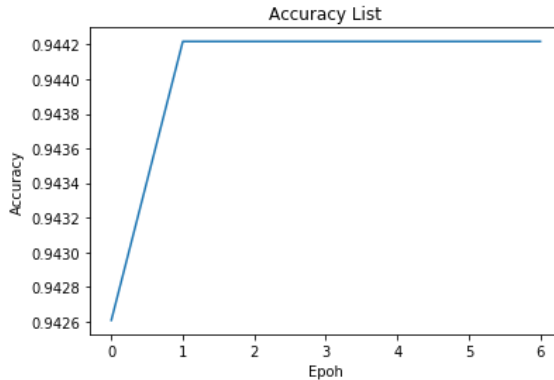
Tabel 5. 6 Hasil Performansi Skenario *Bigram* Sentimen

	Akurasi	Presisi	Recall
<i>Training</i> Sentimen 1	0.99	1.00	1.00
<i>Testing</i> Sentimen 1	0.79	0.90	0.79
<i>Training</i> Sentimen 2	0.99	1.00	0.99
<i>Testing</i> Sentimen 2	0.77	0.83	0.77

c. Skenario Kombinasi

Jumlah *cluster* yang digunakan dalam skenario ini adalah 4, dengan hasil pada proses pembelajaran memiliki akurasi dan RMSE yaitu 0.944 dan 0.140. Sedangkan hasil akurasi pada proses pengujian adalah 0.69. Tabel 5.7 menunjukkan hasil performansi dari skenario

pemotongan fitur kombinasi *unigram* dan *bigram*. Gambar 5.15 menunjukkan nilai akurasi pada proses pembelajaran tiap *epoch*.



Gambar 5. 15 Hasil Akurasi Pembelajaran Skenario Kombinasi

Sama seperti sebelumnya akan dilakukan evaluasi dengan toleransi 0 dan toleransi 1. Hasil dari evaluasi performansi pada skenario ini dapat dilihat pada Tabel 5.7.

Tabel 5. 7 Hasil Performansi Skenario Kombinasi

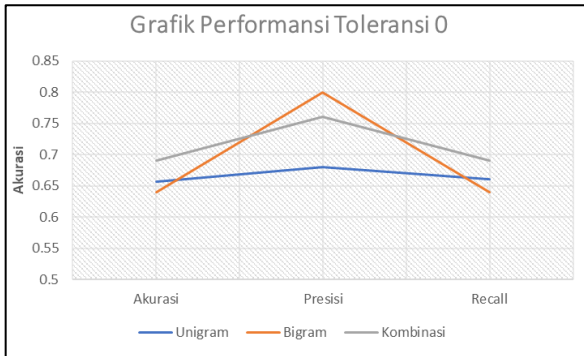
	Akurasi	Presisi	Recall
<i>Training</i>	0.94	0.95	0.94
<i>Testing</i> Toleransi 0	0.69	0.76	0.69
<i>Testing</i> Toleransi 1	0.81	0.89	0.81

Sedangkan untuk skenario sentimen 1 dan sentimen 2 seperti yang dilakukan pada sebelumnya dapat dilihat performansi metode pada Tabel 5.8.

Tabel 5. 8 Hasil Performansi Skenario Kombinasi Sentimen

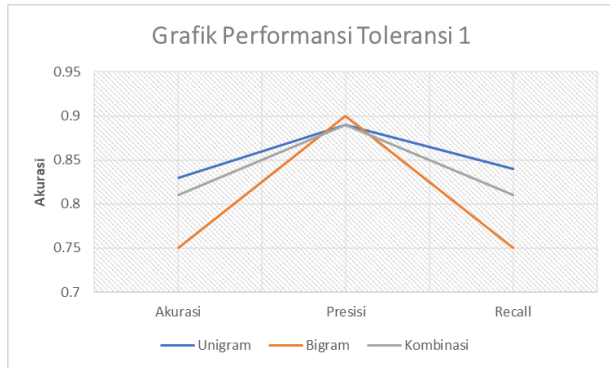
	Akurasi	Presisi	Recall
<i>Training</i> Sentimen 1	0.993	0.99	0.99
<i>Testing</i> Sentimen 1	0.928	0.93	0.93
<i>Training</i> Sentimen 2	0.978	0.98	0.98
<i>Testing</i> Sentimen 2	0.889	0.89	0.89

Grafik dari performansi untuk model yang digunakan untuk prediksi nilai ulasan dengan toleransi 0 dapat dilihat pada Gambar 5.16. Dari grafik dapat dilihat bahwa penggunaan Kombinasi memiliki performansi akurasi dan recall yang paling baik. Nilai *unigram* untuk akurasi dan *recall* jauh lebih baik dari pada skenario *bigram*.



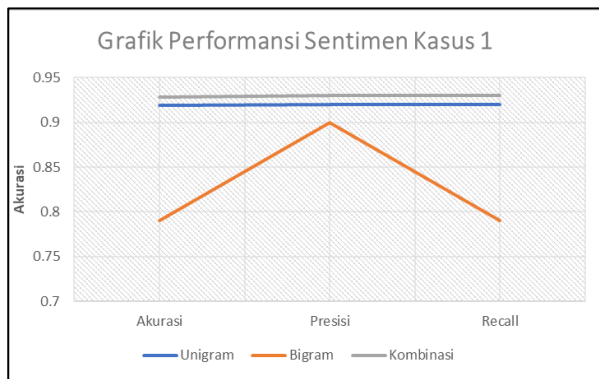
Gambar 5. 16 Perbandingan Performansi Skenario Pada Toleransi 0

Sedangkan grafik dari performansi untuk model yang digunakan untuk prediksi nilai ulasan dengan toleransi 1 dapat dilihat pada Gambar 5.16. Dari grafik dapat dilihat bahwa penggunaan Kombinasi memiliki performansi lebih baik dari yang lain. Sedangkan nilai performansi untuk skenario *unigram* jauh lebih baik dari pada skenario *bigram*.

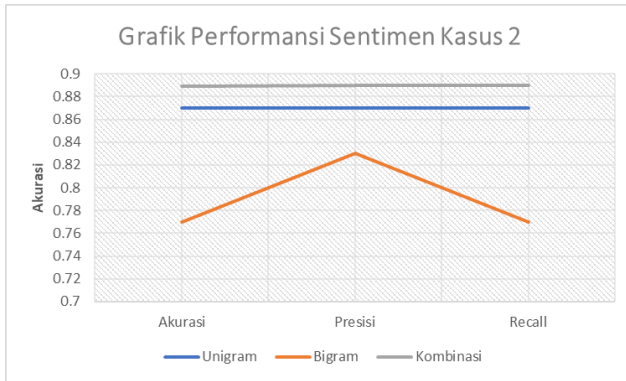


Gambar 5. 17 Perbandingan Performansi Skenario Pada Toleransi 1

Pada bagian skenario sentimen tipe 1 dan 2 pun nilai performansi dengan menggunakan kombinasi lebih baik dari pada skenario *unigram* dan *bigram*. Sedangkan yang memiliki performansi paling baik adalah skenario *bigram*. Grafik performansi dapat dilihat pada Gambar 5.18 dan 5.19.



Gambar 5. 18 Perbandingan Performansi Skenario Pada Sentimen 1



Gambar 5. 19 Perbandingan Performansi Skenario Pada Sentimen 2

Performansi pada metode dengan pembangkitan *bigram* memiliki performansi paling buruk. Hal ini dikarenakan banyaknya fitur yang kurang relevan menjadi fitur terpilih dalam proses pembentukan TF-IDF. Misalnya fitur ‘barang maaf’, ‘barang tadi’, ‘bau lilin’ kurang relevan untuk dijadikan fitur namun fitur seperti ‘agak kecewa’, ‘agak kurang’, dan ‘agak lama’ cukup bagus untuk dijadikan fitur karena dapat menjadi pembeda antar kelas sentimen sehingga pada saat penggunaan kombinasi antara *unigram* dan *bigram* nilai performansi menjadi lebih bagus namun jumlah fitur yang dijadikan sebagai fitur masih terlalu banyak sehingga dapat memperlambat kinerja sistem.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan dibahas mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik.

6.1 Kesimpulan

Kesimpulan yang didapat dari proses pengerjaan pada penelitian yang telah dilakukan antara lain:

1. Tahap-tahap yang dilakukan dalam proses analisis sentimen pada teks ulasan produk *e-commerce* menggunakan metode *N-Gram* dan *Neuro-Fuzzy* meliputi:
 - a. Pengambilan data pada *website* www.tokopedia.com dengan menggunakan library *beautifulsoup* dan *json*.
 - b. Melakukan pra proses data yang meliputi tokenisasi, *spelling normalization* dimana kata-kata yang tidak baku namun memiliki frekuensi tinggi akan diperbaiki ke dalam kata yang baku dengan menggunakan *corpus* yang dibuat, selanjutnya filtrasi untuk menghilangkan kata *stopword* dimana kata 'tidak', 'agak', 'sangat', 'bukan', 'kurang', 'amat', dan 'cukup' tidak perlu dihapus dari teks ulasan karena cukup mempengaruhi sentimen ulasan, dan terakhir adalah proses *stemming* untuk penghapusan imbuhan kata.
 - c. Proses ekstraksi fitur untuk reduksi dimensi data yang akan digunakan dalam proses klasifikasi menggunakan *truncated SVD* dengan rasio kemiripan 80%.
 - d. Proses pembelajaran menggunakan FCM-ANFIS dimana FCM digunakan sebagai nilai awal dalam parameter premis dalam aturan Sugeno yang digunakan dalam pembelajaran ANFIS dimana nilai *cluster* dan *learning rate* yang digunakan disesuaikan untuk mendapatkan nilai akurasi terbaik antara data latih dan data uji.
2. Hasil terbaik untuk proses prediksi sentimen dari teks ulasan berdasarkan nilai rating adalah skenario kombinasi antara *unigram*

dan *bigram* dengan akurasi sebesar 0.69, sedangkan untuk toleransi 1 akurasi terbaik yaitu skenario *unigram* dengan nilai 0.83. Hasil pengujian dengan menggunakan sentimen diperoleh yang terbaik dengan akurasi pada skenario *unigram bigram* dengan nilai 0.928 untuk skenario sentimen tipe 1 dan nilai terbaik menggunakan skenario kombinasi dengan akurasi sebesar 0.893 skenario sentimen tipe 2. Hal ini menunjukkan *N-gram* kombinasi antara *unigram* dan *bigram* cukup efektif untuk meningkatkan hasil akurasi untuk analisis sentimen berdasarkan nilai rating maupun analisis sentimen untuk kelas negatif dan positif.

6.2 Saran

Saran yang diberikan untuk perbaikan pada penelitian selanjutnya adalah:

1. Menambahkan metode untuk mengatasi *overfitting* dari model yang digunakan.
2. Melakukan *post tagging* untuk memiliki fitur *bigram* yang sesuai karena penggunaan *bigram* untuk pemilihan fitur tidak cukup bagus digunakan untuk pembentukan model dikarenakan banyaknya fitur yang kurang relevan menjadi atribut dalam proses pembelajaran. Misalnya fitur 'barang maaf', 'barang tadi', 'bau lilin' kurang relevan untuk dijadikan fitur namun fitur seperti 'agak kecewa', 'agak kurang', dan 'agak lama' cukup bagus untuk dijadikan fitur karena dapat menjadi pembeda antar kelas sentimen.

DAFTAR PUSTAKA

- [1] PMG International. 2017. **The truth about online consumers.** KPMG International Cooperative. Swiss.
- [2] Liu, Bing. 2012. **Sentiment Analysis and Opinion Mining.** Morgan & Claypol Publisher.
- [3] Sarkar, Dipanjan. 2016. **Text Analytics with Python.** New York: Springer Science+Business Media Inc.
- [4] Ranjana, Rajnish. 2016. **Fuzzy Aspects in Sentiment Analysis and Opinion Mining.** *International Journal of Innovative Research in Science, Engineering and Technology.* Vol. 5. No. 5 : 7750-7755.
- [5] Thiel, Kilian. 2016. **Sentiment Analysis with N-grams.** [Online]tersedia di: <https://www.knime.com/blog/sentiment-analysis-with-n-grams>. [diakses 15 Januari 2019].
- [6] Cosma, Georgina dan Acampora, Giovanni. 2016. **A computational intelligence approach to efficiently predicting review ratings in e-commerce.** *Applied Soft Computing.* Vol. 44:153-162.
- [7] Songpan, Wararat. 2017. **The Analysis and Prediction of Customer of Review Rating Using Opinion Mining.** *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA).* Hal 71-77.
- [8] Iswanto, B. H., Poerwoto, V. 2018. **Sentiment analysis on Bahasa Indonesia tweets using Unigram models and machine learning techniques.** *IOP Conference Series: Materials Science and Engineering.* Vol. 434.
- [9] Watson, Richard. 2008. **Electronic Commerce: The Strategic Perspective.** Zurich: Global Text Project.
- [10] Mudambi, Susan dan Schuff, David. 2010. **What Makes A Helpful Online Review? A Study of Customer Review on Amazon.com.** *MIS Quarterly.* Vol. 34. No. 1:185-200.
- [11] Adriani, Mirna dan Manurung, Ruli. 2018. **A survey of bahasa Indonesia NLP research conducted at the University of Indonesia.**

- [12] Berry dkk. 1994. **Using Linear Algebra for Intelligent Information Retrieval**. Computer Science Department CS-94-270.
- [13] Kusumadewi, Sri dan Hartati, Sri. 2016. **Neuro-Fuzzy**. Yogyakarta : Graha Ilmu.
- [14] Suparta, W. dan Alhasa, K. M. 2016. **Modeling of Tropospheric Delays Using ANFIS**. Springer International.
- [15] Talpur, Noureen, Salleh, Mohd dan Hussain, Kashif. 2017. **An investigation of membership functions on performance of ANFIS for solving classification problems**. IOP Conference Series: Materials Science and Engineering. 226. 012103. 10.1088/1757-899X/226/1/012103.
- [16] Daniel, Jurafsky dan Martin, James H. 2018. **Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition**. New Jersey: Pearson Prentice Hall.
- [17] Azzeh, Mohammad, Cowling, Peter I., dan Neagu, Daniel. 2008. **Software Project Similarity Measurement Based on Fuzzy C-Means**. In: Wang Q., Pfahl D., Raffo D.M. (eds) Making Globally Distributed Software Development a Success Story. ICSP 2008. Lecture Notes in Computer Science. Vol. 5007:123-132.

LAMPIRAN 1

Daftar *Corpus Stopwords*

yang, untuk, pada, ke, para, menurut, antara, dia, ia, seperti, jika, jika, sehingga, kembali, dan, ini, karena, kepada, oleh, saat, harus, sementara, setelah, belum, kami, sekitar, bagi, serta, di, dari, telah, sebagai, masih, hal, ketika, adal ah, itu, dalam, bisa, bahwa, atau, hanya, kita, dengan, akan, juga, ada, mereka, sudah, saya, terhadap, secara, agar, lain, anda, begitu, mengapa, kenapa, yaitu, yakni, daripada, itulah, lagi, maka, tentang, demi, dimana, kemana, pula, sambil, sebelum, sesudah, supaya, guna, kah, pun, sampai, sedangkan, selagi, sementara, apakah, kecuali, sebab, selain, seolah, seraya, seterusnya, tanpa, boleh, dapat, dsb, dst, dll, dahulu, dulunya, anu, demikian, tapi, ingin, juga, nggak, mari, nanti, melainkan, oh, seharusnya, sebetulnya, setiap, setidaknya, sesuatu, pasti, saja, toh, ya, walau, tentu, amat, apalagi, bagaimana pun, aku, kamu, jadi, ke, nya, wah, di, dan, saja, mau, saya, deh, disini, kan, yah, nah, dengan, di, saya, iya, hehe, nih, gan, sis, se, bos, euy, mas, agan, hi, juragan, sih, dari, sama, lah, untuk, apa, kok, an, kak, kalau, sini, bro, sis, tuh, eh, it, ka, pak, neng, om, haha, ml, cm, s, ni, ko, ama, the, btw, broo, bang, mah, mas, mbak, nge, lh, guys, laah, laaah, brooo, ah, dahh, dah, dahhh, yak, amp, as, boss, bosss, boz, gaan, gannnn, gannnnn, gans, lu, man, masbro, my, si, sii, so, wkwkw, yahh, yaaa, an, ana, dong, gaaan, gaes, is, kakak, kk, la, lahhhh, lahhhh, min, ndan, nihh, reseller, sist, this, uuuuuu.

LAMPIRAN 2

Daftar Corpus *Spelling Normalization*

- ada ad da
- agak agk aga rada agk rda
- alhamdulillah alhamdulillah
- antar anter
- antena antenna
- apa pa ap
- asap asep asepnnya asepnnyaa
- atau ato
- bagaimana bgmn gmn
gimana bgaimn gmana
- bagus good nice bgus
bagusssssssssss baguss bgs
bagussss nicee baguss
baguuus good
- balas bales bls
- banget bngt bgt banget bngt
bangetttttt bngit
- banyak byk bnyk
- barang brg brang brng
baramg barangg baraaang
baranf barng
- barangnya barangny barangx
baranh brangx brgnya
- baru bru br
- baterai batre baterai
baterainya baterenya
batrenya batere batrenya
baterynya battery btire
- bayar byr
- beberapa bbrp bbrapa
- begitu gitu gtu gt gituu
- belanja blnj blanja
- beli bli
- belum blum blm blom belum
- benar bener bnar bner bnr
- berasa brasa
- besar gede gde bsar
- besok bsk bsok
- bintang bingtang
- bisa bsa bs bnyk
- bohong boong bhong
- bro broo brooo
- buat bt buwat bwt
- bukan bkn
- buram burem
- buruk bad bruk
- cepat fast cpt cpat cpet
cepaatt cepet cepatt cept
- charger charge
- coba nyoba
- dah daah dach deeh
- dalam dlm dlam dalem
- dan and n dn
- dapat dapet dpet dpt
- dari dri dr
- datang dtg dateng dtang datng
dtng datengg dteng
- datangnya datengnya
dtengnya
- deh dech dehh
- dengan dgn dg dng dngan
dngn
- dengar denger dengerr
- deskripsi description deskrips
diskripsi
- dicoba dcoba
- dikirim dkirim

- dikit dkit
- dipakai dipakek dipake
- direkomendasikan
recommended rekomended
recomended recommender
recommend recomend
rekomen recomendeed
recomm reccomended
rekomend
- disini dsni dsini
- diterima dterima
- doang doank
- dong donk
- dulu dlu
- ekspedisi expedisi
- ekspektasi ekspetasi
ekpektasi ekspetasi expectasi
- ekstra extra
- enak enakk enakkk enk
- entah ntah
- eror error
- flavor flavour flavornya
flavournya
- gaada gada
- gambar gmbr gmbar pict pic
gmbr gbr
- gan gann gannn gannnnnn
gannnnnn
- gila gile gilee
- habis abis abisin abissss hbis
- haduh hadehh hadeh hduh
- haha hehehe wkwk wkwwk
wkk kwkw hehe hahahaha
kwkwkwkw hahahahha
hahah hahaha hee heee hehee
hehehehe
- halus alus
- hancur ancur hncur hncr
- hangus angus hngus
- hanya cuma cma hanya cuman
cmn
- harga hrg hrga
- hari hri hr
- harus hrs hrus
- hidup idup hdp hdup
- hilang hlang ilang ilangg
- impor import
- iya ya iy y yaa
- jadi jd jdi
- jangan jgn jngn
- jawaban jwaban jwb
- job jobs
- jos joss jossss josssss
josssssss jooss
- juga jg jag jga
- kakak kaka kak
- kalau klo klau kalo kl klu klw
klow
- kamu kmu you
- kapan kpan
- kapastitas capacity kpasitas
kpsitas
- karena krn soalnya soale
karna krna coz
- kasih kasihh
- kasih ksh kasi ksh
- kayak kyk
- kemana kmn
- kemarin kemaren kmren
kmarin kemrin kmrn
- kenapa knp knapa
- kental kentel kntel
- keren kren
- kilat kilatzzz

- kirim krim girim kirm send
sent krm krim
- kok koq kog
- komentar komen comment
coment
- komplain komplem
- komplit complete complit
kmpelit compelete kumplit
- kualitas kualitas kwalitas
quality kualiatas
- kurang krg krang krng krg
- label lebel lbel
- lagi lg lgi lagii lagiii
- lah lahh
- lain laen
- lama lm lma luamaa
- lancar lncr lncarr
- langsung lngsung lgsg lgs
- lebih lbh lrbih lbih
- lihat liat lihat
- lumayan lmayan lmyan
lumayann lumayannnn
mayan
- maaf maap sori sorry sory
- main maen
- malam malem mlm
- malas males mles
- mantap mantul mantp mntap
mntr mantaappp mantp
mantapp maknyuussss
maknyus mantaff mantaf
mntap maknyuss mantab
mantabb mantaap mantabs
mantaps manteb mantep
mantepp mantul mntp mantp
mntap mantaap mantaappp
mantep mantepp mntab ntap
mantaaaap mantaap
- mantaab mantabss mantappp
mantav
- masih msih msh
- masuk msuk
- mau mao mw
- memang emang emg mmg
- mesan mesen
- mintanya mintnya minty
- mudah mdh mdah
- mungkin mgkn mungkn
mngkin
- menangkap nangkep
- nanti ntar
- ngebul ngebull ngebulll
ngebulnya
- no nooo noo
- nomor nmr nomer
- nuhun nuwun
- nya ny
- oke ok okelah okeee okee
okeh okey okay oc oce
okeeee okeeeee okeeeeeee
okkk
- original ori orgnal originl orii
- overall overal
- packing pckng packingan
paking packaging
packagingnya packingannya
packingnya pengemasan
peking
- padahal pdhl pdahal pdahl
pdhal
- pakai pkai pake pke make
- pas pass passs
- pelayanan pelayaman
playanan service servis
services

- pembeli buyer customer
- pengiriman pngiriman
delivery
- penjual seller sellr
- pesan psan
- pesanan order pesana psanan
mesan
- pokok pkok poko pokonya
- produk prodk peoduk product
prodak
- puas puass puassss
- rapat rapet
- rapi rapih rpi
- rata rta
- reguler regular
- respon respond rson respn
response repon respons
responsif
- responsif responsive
- ribu rbu
- rumah rmh
- rupa rupo
- saja aja sja aj aje ajaaa
- sama sm sma ma
- sampai nyampe smpai
nyampai nyampek sampe
smp ampe smpai sampe
nyampe nympe smpe ampe
amp smpenya nympe
- sampainya nyampainya
nyampainya samapai sampei
sampek sampenya smpe
- sangat sngt sngat sgt
- sangat very
- saya sya sy gue aku ak gw aq
ane ku
- sayang syg syang sayng
- sebelum sblum sblm
- sebentar sbntr sbentr sbntar
- sedikit sdkit sedikit
- segar seger sgar
- sekali skali skli sekli
- sekarang skarang skrng skrag
- selalu slalu
- selamat slamat slmat
- semoga smoga smga moga
smg mg
- sempat sempet smpet smpat
- semua smua
- seperti sprti spt sperti
- sesuai ssuai sesuai
- setelah stelah stlh
- sih sh
- simpel simple simpl
- sip sipp siip sipp siiiippppp
siiip sipp siiiip
- standar standard stndar
standart
- stok stock
- sudah sdh sdah udah dah sudh
udahh uda ud syudah sydah
udah udh sdh sdah uda dah dh
sudh ud
- tadi td
- tambah tmbh
- tapi tp tpi
- tebal tebal tbel tbal
- telepon tlp tlepon telpon
telepeon tlp tlepon telp
- teman tmen temen
- terima trima trm trma
- terimakasih thanks tanks thx
thank tnks makasih mksh
trimakasih tq tnks tx trims

makasihhh makasi thanx
 tenkyu mkasih tks trmksh
 thankyou mkasih terimakasih
 trim mkasih thxs makasii thnks
 terimakasih thankss
 haturnuhun makasihh
 tengkyu tengkyu

- ternyata trnyata
- tersedia ready
- terus trs trus
- tes testing test
- tetap tetep
- tetapi but tpi
- tidak gak gk tdk gx ga tidal
tdak kagak kgak nggak ngak

no not nga g ngga ngak ngk
ndak engga tidk kaga

- tokopedia tokped tokpd toped
- tolong tlng tlong
- ulasan review
- untuk utk
- wah wa
- waktu wkt wktu
- walaupun walopun wlopun
wlpn wlpun
- warna wa4na wrna
- worth wort
- yang yg
- yang yng yg

LAMPIRAN 3

Matriks Koefisien $\bar{w}_i x_{ij}$ Halaman 52

	0	1	2	3	4
0	0.063786	-0.009105	0.010203	0.030869	-0.035467
1	0.114474	-0.019787	-0.065054	0.020142	0.148193
2	0.273985	-0.069658	0.115685	0.074177	-0.169350
3	0.249871	0.110949	-0.273002	0.783254	-0.237892
4	0.174116	0.440418	0.645374	0.120169	0.110521
5	0.255452	0.633063	-0.456420	-0.446087	-0.238316
6	0.210027	0.318364	0.344377	-0.025624	0.088231
7	0.073114	0.006779	-0.044719	-0.000568	0.029002
8	0.039461	-0.005668	0.017204	0.000838	-0.030974
9	0.325157	-0.361431	0.176617	-0.231635	-0.578220
10	0.446400	-0.313116	0.034171	-0.205894	0.328735

	5	6	7	8	9
0	0.076767	0.260412	0.181156	-0.025858	0.028976
1	0.086630	0.281937	0.291553	-0.050396	-0.165684
2	-0.167537	0.957312	0.012217	-0.003106	0.005159
3	-0.268261	0.972777	0.006993	0.003105	-0.007640
4	0.029695	0.976754	0.004144	0.010482	0.015360
5	-0.127423	0.988585	0.002950	0.007310	-0.005270
6	-0.129907	0.979999	0.004287	0.006498	0.007029
7	0.086876	0.244515	0.225904	0.020944	-0.138168
8	0.092143	0.175143	0.185848	-0.026693	0.081025
9	0.012669	0.954700	0.015428	-0.017150	0.008380
10	-0.490609	0.937764	0.029626	-0.020780	0.002268

	10	11	12	13
0	0.087671	-0.100729	0.218022	0.739588
1	0.051301	0.377430	0.220636	0.718063
2	0.003308	-0.007552	-0.007471	0.042688
3	0.021919	-0.006657	-0.007507	0.027223
4	0.002860	0.002630	0.000707	0.023246
5	-0.005151	-0.002752	-0.001471	0.011415
6	-0.000523	0.001801	-0.002651	0.020001
7	-0.001755	0.089609	0.268423	0.755485
8	0.003946	-0.145875	0.433958	0.824857
9	-0.010991	-0.027436	0.000601	0.045300
10	-0.013664	0.021817	-0.032560	0.062236

LAMPIRAN 4

Syntax ANFIS

```
import numpy as np
import pandas as pd
import skfuzzy as fuzz
import time
from sklearn.model_selection import train_test_split
import copy
import matplotlib.pyplot as plt

data1=pd.read_csv('train_test_data_tfidfummy.csv',
index_col=0)
print(data1.shape)
X_train=data1.drop(columns='label')
Y_train=data1['label']

data_attr=X_train.values
data_target=Y_train.values
arr_attr=data_attr
arr_target=np.matrix(data_target)

n_data,n_attr=arr_attr.shape
n_c,E,Eps,epoh,epoh_max,lr=2,1,0.01,1,30,0.4

def gbell(x,c,a):
    y=1/(1+abs((x-c)/a)**2)
    return y
def RMSE(y, y_cap):
    rmse=np.sqrt(np.sum(np.square(arr_target-
np.transpose(y_cap)))*(1/len(y_cap)))
    return rmse

loop=True
while loop:
    cntr, u, u0, d, jm, p, fpc =
fuzz.cluster.cmeans(np.transpose(arr_attr),n_c,2,error=0.
001,maxiter=2000,init=None,seed=123)
    distance=np.transpose(d)
    data_cluster=[]
    for i in distance:
```

```

mean_arr=[]
std_arr=[]
for i in range(n_c):
    temp=X_train[X_train['Cluster']==i]
    mean_arr.append(list(np.mean(temp)[0:n_attr]))
    std_arr.append(list(np.std(temp, ddof=1)[0:n_attr]))

E_list=[]
epoh=0
E=1
da=[]
dc=[]
dc_lama=[]
da_lama=[]
rmse_list=[]
acc_list=[]
while (epoh<epoh_max):
    print('-----')
    '+str(epoh)+'-----')
    -----')
    epoh+=1
    E=0
    node_1=n_c*n_attr
    output_1=[]
    for i in range(n_data):
        temp_col=[gbell(data_attr[i][j],mean_arr[k][j],
            std_arr[k][j]) for j in range(n_attr) for k in
            range(n_c)]
        output_1.append(temp_col)
    if(epoh==1):
        lapisan1=output_1

#ouput lapisan ke-2
start=time.time()
output_2=[]
for row in output_1:
    temp=[]
    for j in range(n_c):
        temp.append(np.min(row[j::n_c]))
#        temp.append(np.prod(row[j::n_c]))
    output_2.append(temp)
if(epoh==1):
    lapisan2=output_2

```

```

#ouput lapisan ke-3
start=time.time()
output_3=[]
for row in output_2:
    row=np.array(row)
    output_3.append(list(row/np.sum(row)))
if(epoh==1):
    lapisan3=output_3

#mencari matriks koefisien cij --> wi*xij
arr_attr_bias=np.append(arr_attr,
np.ones((n_data,1)), axis=1)
arr_attr_bias.shape
koef_A=[]
for indeks,data_bias in enumerate(arr_attr_bias):
    temp=np.array(data_bias*output_3[indeks][0])
    for i in range(1,n_c):
        z=np.array(data_bias*output_3[indeks][i])
        temp=np.append(temp,z)
    koef_A.append(temp)
koef_A=np.matrix(koef_A)
if(epoh==1):
    matriks_koefA=koef_A
#LSE
num_kof=(n_attr+1)*n_c
c_cap, residuals,
rank,s=np.linalg.lstsq(koef_A,np.transpose(arr_target))
if(epoh==1):
    matriks_c_cap=c_cap

#ouput lapisan ke-4 & 5
matrix_attr_bias=np.matrix(arr_attr_bias)
for i in range(0,num_kof-1,n_attr+1):
    ini=koef_A[:,i:i+n_attr+1]*c_cap[i:i+n_attr+1]
    if(i==0):
        y_cap_k=ini
    else:
        y_cap_k=np.append(y_cap_k,ini, axis=1)
y_cap=koef_A*c_cap
if(epoh==1):
    matriks_y_cap=y_cap

```

```

if (epoh < epoh_max):
    print(epoh, '-->propagasi error')
    #propagasi error
    start=time.time()
    for i in range(n_data):
        print('.',end=" ")
        #propagasi error lapisan ke-5 (E5)
        E5=-2*(arr_target[0,i]-y_cap[i])

        #propagasi error lapisan ke-4 (E4)
        E4=np.array([E5 for j in range(n_c)])

        #propagasi error lapisan ke-3
        E3=[]
        for j in range(n_c):
            E3=np.append(E3, y_cap_k[i,j]*E4[j])

        #propagasi error lapisan ke-2
        E2=np.array([])
        for j in range(n_c):
            in_k=1 #pembilang untuk error kelas yg sma
            for k in range(n_c):
                if (k!=j):
                    in_k=in_k*output_3[i][k]
            temp_e2=0
            for k in range(n_c):
                if(k!=j):
                    temp_e2=temp_e2+E3[k]*((-
output_2[i][k])/sum(output_2[i])**2)
                else:
                    temp_e2=temp_e2+E3[k]*(in_k/sum(output_2[i])**2)
            E2=np.append(E2, temp_e2)

        #propagasi error lapisan ke-1
        E1=np.array([])
        for j in range(n_attr*n_c):
            tt=1
            for k in range(j%n_c,n_attr*n_c,n_c):
                if(k!=j):
                    tt=tt*output_1[i][k]
            E1=np.append(E1, tt*E2[j%n_c])

```

```

#perbaiki bobot a dan c
    for j in range(n_attr):
        dstd=[]
        dmean=[]
        dstd_lama=[]
        dmean_lama=[]
        for k in range(n_c):
            L=arr_attr[i][j]-mean_arr[k][j]
            H=(1+(L/std_arr[k][j])**2)**2

temp_dstd=((2*L**2)/((std_arr[k][j]**3)*H))*(E1[j*k]*lr*arr_attr[i][j])

temp_dmean=((2*L)/((mean_arr[k][j]**2)*H))*(E1[j*k]*lr*arr_attr[i][j])

        if(epoh>1):
            temp_dstd=0.9*da_lama[j][k]+temp_dstd

temp_dmean=0.9*dc_lama[j][k]+temp_dmean
            mean_arr[k][j]=mean_arr[k][j]+temp_dmean
            std_arr[k][j]=std_arr[k][j]+temp_dstd
            temp_dstd_lama=temp_dstd
            temp_dmean_lama=temp_dmean
            dstd.append(temp_dstd)
            dmean.append(temp_dmean)
            dstd_lama.append(temp_dstd_lama)
            dmean_lama.append(temp_dmean_lama)
        da.append(dstd)
        dc.append(dmean)
        dc_lama.append(dstd_lama)
        da_lama.append(dmean_lama)

#hitung SSE
E=E+(arr_target[0,i]-y_cap[i])**2
print('epoh ke-', epoh, 'error', float(E))
rmse_temp=RMSE(arr_target, y_cap)
rmse_list.append(rmse_temp)
print('epoh ke-', epoh, 'rmse', rmse_temp)

```


LAMPIRAN 5

```
from coba import *
from rating import
class MyWindow(QMainWindow):
    def __init__(self, parent=None):
        super().__init__()
        self.ui=Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.butt_submit.clicked.connect(self.prediksi)
        self.answer, self.stopword = self.load_corpus()
        self.aa=rating()

    def prediksi_anfis(self, teks):
        print('token')
        hasil_token=self.tokenisasi(teks)
        print(hasil_token[0][0])
        self.ui.plainTextEdit.setStyleSheet("background
        -color: rgb(255, 255, 255);\n"
        "color: rgb(0,0,0);")
        aa='HASIL TOKEN -> '
        for i,v in enumerate(hasil_token[0]):
            if(i==0):
                aa=aa+' '+v
            else:
                aa=aa+', '+v
        teks22=aa
        self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
        print('remove typo')
        hasil_remove typo=self.remove typo(hasil_token)
        aa='HASIL SPELLING NORMALIZATION -> '
        for i,v in enumerate(hasil_remove typo[0]):
            if(i==0):
                aa=aa+' '+v
            else:
                aa=aa+', '+v
        teks22=teks22+'\n'+aa
        self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
        print('filtrasi')
        hasil_filtrasi=self.remove_stopword(hasil_remove
        etypo)
        aa='HASIL FILTRASI -> '
        for i,v in enumerate(hasil_filtrasi[0]):
            if(i==0):
                aa=aa+' '+v
            else:
```

```

        aa=aa+', '+v
    teks22=teks22+'\n'+aa
    self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
    print('stem')
    aa='HASIL STEMMING -> '
    hasil_stemming=self.stemming_word(hasil_filtras
    i)

        for i,v in
    enumerate(hasil_stemming[0]):
            if(i==0):
                aa=aa+' '+v
            else:
                aa=aa+', '+v
        teks22=teks22+'\n'+aa

    self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
    aa=self.wordtosentence(hasil_stemming)
    df_clean=pd.read_csv(r'C:\Users\HP\KUN FAYAKUN
    2\dataset\UNIGRAM\df_clean_4583.csv",
    index_col=0)
    tfidf=TfidfVectorizer(min_df=3,
    ngram_range=(1,1))
    X=tfidf.fit_transform(df_clean['Review Clean'])
    brs_1=aa
    tf_idf_1=pd.DataFrame(tfidf.transform(brs_1).to
    array())
    wew='HASIL TF IDF\n'+str(tf_idf_1.values)
    teks22=teks22+'\n'+wew
    self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
    S=pd.read_csv(r'C:\Users\HP\KUN FAYAKUN
    2\dataset\UNIGRAM\Sk_unigram.csv', index_col=0)
    U=pd.read_csv(r'C:\Users\HP\KUN FAYAKUN
    2\dataset\UNIGRAM\U_unigram.csv', index_col=0)
    mat_u=U.values
    mat_s=S.values
    row_1=tf_idf_1.values
    q_u=np.dot(row_1,mat_u)
    q_proyeksi=np.dot(q_u, np.linalg.inv(mat_s))
    wew='HASIL PROYEKSI SVD\n'+str(q_proyeksi)
    teks22=teks22+'\n'+wew
    self.ui.plainTextEdit.setPlainText(self._translate("MainWindow", teks22))
    rating_a=self.aa.haha(q_proyeksi)
    print(rating_a)
    return rating_a[0]

```



```

def prediksi(self):
    teks = self.ui.textReview1.toPlainText()
    teks=[teks]
    rate=self.prediksi_anfis(teks)
    print(rate)
    if rate==1:
        self.ui.star_img.setStyleSheet("image:
            url(/gambar/1star.png);")
    elif rate==2:
        self.ui.star_img.setStyleSheet("image:
            url(/gambar/2star.png);")
    elif rate==3:
        self.ui.star_img.setStyleSheet("image:
            url(/gambar/3star.png);")
    elif rate==4:
        self.ui.star_img.setStyleSheet("image:
            url(/gambar/4star.png);")
    else:
        self.ui.star_img.setStyleSheet("image:
            url(/gambar/5star.png);")

class rating():
    def haha(self, q_proyeksi):
        self.q_proyeksi=q_proyeksi
        mean_csv=pd.read_csv(r"C:\Users\HP\KUN FAYAKUN
            2\UNIGRAM\5\mean_arr_train.csv",
            index_col=0)
        std_csv=pd.read_csv(r"C:\Users\HP\KUN FAYAKUN
            2\UNIGRAM\5\std_arr_train.csv",
            index_col=0)
        c_cap_csv=pd.read_csv(r"C:\Users\HP\KUN FAYAKUN
            2\UNIGRAM\5\c_cap_train.csv", index_col=0)
        mean_arr=mean_csv.values
        std_arr=std_csv.values
        c_cap_arr=c_cap_csv.values
        data_attr1=self.q_proyeksi
        arr_attr1=data_attr1
        n_attr1=arr_attr1.shape[1]
        n_c1=5
        n_data1=1
        num_kof=(n_attr1+1)*n_c1
        print('Lapisan 1')
        output_1=[]
        for i in range(n_data1):
            temp_col=[self.gbell(data_attr1[i][j],mean_
                arr[k][j],std_arr[k][j]) for j in
                range(n_attr1) for k in range(n_c1)]
            output_1.append(temp_col)

```

```

# #ouput lapisan ke-2
output_2=[]
print('Lapisan 2')
for row in output_1:
    temp=[]
    for j in range(n_c1):
        temp.append(np.min(row[j::n_c1]))
    output_2.append(temp)
#ouput lapisan ke-3
output_3=[]
print('Lapisan 3')
for row in output_2:
    row=np.array(row)
    output_3.append(list(row/np.sum(row)))
# #mencari matriks koefisien cij --> wi*xij
print('mencari matriks koefisien cij -->
wi*xij')
arr_attr_bias=np.append(arr_attr1,
np.ones((1,1)))
koef_A=[]
temp=np.array([])
for i in range(0,n_c1):

z=np.array(arr_attr_bias*output_3[0][i])
temp=np.append(temp,z)
koef_A.append(temp)
koef_A=np.matrix(koef_A)

print('Lapisan 4&5')
matrix_attr_bias=np.matrix(arr_attr_bias)
y_cap=koef_A*c_cap_arr

y_pred=self.label(y_cap)
print(y_cap)
print(y_pred)
return y_pred

```

LAMPIRAN 6

- Matriks *Confusion*

		LABEL PREDIKSI					SUPPORT
		1	2	3	4	5	
LABEL ASLI	1	26	1	5	3	9	44
	2	3	17	4	0	6	30
	3	12	5	50	18	22	107
	4	27	4	5	83	80	199
	5	38	12	9	51	427	537

Penghitungan Akurasi :

$$\begin{aligned}
 \text{Akurasi} &= \frac{TP + TN}{TP + TN + FP + FN} = \frac{26 + 17 + 50 + 83 + 427}{44 + 30 + 107 + 199 + 537} \\
 &= 0.657579062
 \end{aligned}$$

Penghitungan *Recall* :

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

Rating 1	$\frac{26}{26 + 1 + 5 + 3 + 9} = 0.590$
Rating 2	$\frac{17}{3 + 17 + 4 + 0 + 6} = 0.567$
Rating 3	$\frac{50}{12 + 5 + 50 + 18 + 22} = 0.467$
Rating 4	$\frac{83}{27 + 4 + 5 + 83 + 80} = 0.417$

Rating 5	$\frac{427}{38 + 12 + 9 + 51 + 427} = 0.795$
----------	--

$$\begin{aligned} \text{Weighted Avg Recall} &= \frac{44}{917} \times 0.590 + \frac{30}{917} \times 0.567 + \\ &\frac{107}{917} \times 0.467 + \frac{199}{917} \times 0.417 + \frac{537}{917} \times 0.590 = \mathbf{0.6576} \end{aligned}$$

Penghitungan *Precision* :

$$\text{Precision} = \frac{TP}{TP + FP}$$

Rating 1	$\frac{26}{26 + 3 + 12 + 27 + 38} = 0.245$
Rating 2	$\frac{17}{1 + 17 + 5 + 4 + 12} = 0.436$
Rating 3	$\frac{50}{5 + 4 + 50 + 5 + 9} = 0.685$
Rating 4	$\frac{83}{3 + 0 + 18 + 83 + 51} = 0.535$
Rating 5	$\frac{427}{9 + 6 + 22 + 80 + 427} = 0.785$

$$\begin{aligned} \text{Weighted Avg Precision} &= \frac{44}{917} \times 0.245 + \frac{30}{917} \times 0.436 + \\ &\frac{107}{917} \times 0.685 + \frac{199}{917} \times 0.535 + \frac{537}{917} \times 0.785 = \mathbf{0.6818} \end{aligned}$$

BIODATA PENULIS



Ayu Ni'matul Fitriyah atau biasa dipanggil Ayu lahir di Kediri tanggal 24 Mei 1997. Pendidikan formal yang pernah ditempuh yaitu TK Kusuma Mulia An Nur Ngletih, MI TPIM Ngletih, SMP Negeri 1 Wates dan SMA Negeri 3 Kediri. Sekarang penulis menempuh pendidikan S1 di Departemen Matematika Fakultas

Matematika, Komputasi dan Sains Data Institut Teknologi Sepuluh Nopember Surabaya dengan bidang minat Ilmu Komputer. Selama kuliah, penulis aktif di organisasi tingkat jurusan yakni HIMATIK ITS. Pada tahun 2016-2017 penulis menjadi anggota aktif HIMATIKA ITS sebagai *Assistant Treasurer* dan tahun 2017-2018 penulis tetap aktif di organisasi tersebut dengan mengemban amanah sebagai *Secretary of HIMATIKA ITS*. Demikian biodata tentang penulis. Jika ingin memberikan saran, kritik, dan diskusi mengenai laporan Tugas Akhir ini, dapat dikirimkan melalui email ayu.nimatulf@gmail.com. Terimakasih.

