

**SENTIMEN ANALISIS BERBASIS ASPEK TERHADAP *REVIEW*
APLIKASI *DIGITAL WALLET* MENGGUNAKAN METODE
*EXTREME GRADIENT BOOSTING***

SKRIPSI

MUHAMMAD KHAFFI IRWAN

181402102



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA MEDAN**

2023

**SENTIMEN ANALISIS BERBASIS ASPEK TERHADAP *REVIEW*
APLIKASI *DIGITAL WALLET* MENGGUNAKAN METODE
*EXTREME GRADIENT BOOSTING***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Teknologi Informasi

MUHAMMAD KHAFFI IRWAN

181402102



PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA MEDAN

2023

PERSETUJUAN

iii

PERSETUJUAN

Judul : SENTIMEN ANALISIS BERBASIS ASPEK
TERHADAP *REVIEW* APLIKASI *DIGITAL WALLET*
MENGUNAKAN METODE *EXTREME GRADIENT*
BOOST

Kategori : SKRIPSI

Nama Mahasiswa : MUHAMMAD KHAFFI IRWAN

Nomor Induk Mahasiswa : 181402102

Program Studi : SARJANA (S-1) TEKNOLOGI INFORMASI

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA

Medan, 22 November 2023

Komisi Pembimbing:

Pembimbing 2,



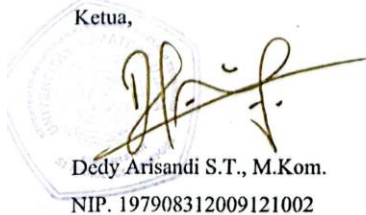
Fanindia Purnamasari S.TI., M.IT
NIP. 198908172019032023

Pembimbing 1,



Ainul Hizriadi S.Kom., M.Sc
NIP. 198510272017061001

Diketahui/disetujui oleh
Program Studi S-1 Teknologi Informasi
Ketua,



Dedy Arisandi S.T., M.Kom.
NIP. 197908312009121002

PERNYATAAN

SENTIMEN ANALISIS BERBASIS ASPEK TERHADAP *REVIEW* APLIKASI *DIGITAL WALLET* MENGGUNAKAN METODE *EXTREME GRADIENT BOOST*

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 22 November 2023

MUHAMMAD KHAFFI IRWAN

181402102

UCAPAN TERIMA KASIH

Segala pujian dan rasa syukur disampaikan kepada Allah SWT atas segala kebaikan dan kasih-Nya yang telah membantu penulis menyelesaikan skripsi yang diperlukan untuk meraih gelar sarjana komputer di Program Studi Teknologi Informasi Universitas Sumatera Utara. Penulis juga ingin mengucapkan terima kasih atas bimbingan, doa, dan dukungan yang diberikan oleh semua pihak selama penulis menyelesaikan skripsi ini. Dalam kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada:

1. Yang paling utama adalah rasa syukur penulis kepada Allah SWT yang sudah memperlancar penulis dalam menyelesaikan tugas akhir.
2. Keluarga penulis yaitu kedua Orangtua Ayah Irwan, S.T. dan Ibu Jeni Cancer, S.E. yang selalu memberikan nasihat dan dukungan kepada penulis untuk menyelesaikan tugas akhir.
3. Bapak Dr. Muryanto Amin, S.Sos., M. Si sebagai Rektor Universitas Sumatera Utara.
4. Ibu Dr. Maya Silvi Lydia, M.Sc sebagai Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara.
5. Bapak Dedy Arisandi ST., M.Kom., sebagai Ketua Program S1 Teknologi Informasi Universitas Sumatera Utara dan Bapak Ivan Jaya S.Si., M.Kom., sebagai Sekretaris Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
6. Bapak Ainul Hizriadi S.Kom., M.Sc., sebagai dosen pembimbing 1 dan Ibu Fanindia Purnamasari S.TI., M.IT., sebagai dosen pembimbing 2 yang telah meluangkan waktu mereka dan memberikan saran kepada penulis untuk menyelesaikan skripsi.
7. Bapak Baihaqi Siregar S.Si., M.T. sebagai dosen pembimbing 1 dan Ibu Rossy Nurhasanah S.Kom., M.Kom sebagai dosen pembimbing 2 yang telah memberikan saran kepada penulis untuk menyempurnakan skripsi.

8. Dosen pengajar di Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
9. Staff dan pegawai di Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
10. Teman Penulis Yaitu Arif Pati Lubis S.Kom. yang banyak membantu penulis selama masa perkuliahan dan terutama saat menyelesaikan tugas akhir.
11. Teman Penulis yaitu Rini Royanti S.Kom. dan Yehezkiel Simajuntak S.Kom. yang telah banyak mengajari penulis tentang teks processing.
12. Teman-teman selama masa perkuliahan penulis yaitu Farhan Al Zuhri S.Kom., Raymond Saragih, Alief Rizki Nuari, dan teman-teman lainnya yang termasuk di grup Tembak.
13. Teman-teman stambuk 2018 Teknologi Informasi yang menemani penulis dalam masa perkuliahan.

ABSTRAK

Aplikasi dompet digital memiliki peran penting dalam perkembangan finansial modern. LinkAja merupakan salah satu aplikasi *digital wallet* yang digunakan oleh masyarakat saat ini. Evaluasi masyarakat terhadap aplikasi ini mempengaruhi persepsi umum dan identifikasi aspek-aspek kunci yang memengaruhi kepuasan pengguna. Perusahaan perlu mengetahui pandangan pengguna terhadap aplikasi LinkAja agar dapat menjadi bahan evaluasi untuk meningkatkan performa aplikasi LinkAja. Analisis sentimen berbasis aspek adalah alat yang efektif untuk memahami pandangan pengguna dari beragam ulasan. Penelitian ini bertujuan mengembangkan model analisis sentimen yang fokus pada aspek-aspek dalam ulasan pengguna aplikasi dompet digital. Metode *Extreme Gradient Boosting* (XGBoost) dipilih karena kemampuannya dalam mengatasi masalah klasifikasi dan data berukuran besar. Penulis menggunakan 2000 data ulasan pengguna yang diambil dari hasil scrapping *Google Play Store* untuk penelitian ini. 2000 data tersebut akan di *split* menjadi data *training* dan data *testing* dengan perbandingan 70:30. Masing-masing data akan dilakukan tahap *preprocessing* agar data menjadi bersih. Selanjutnya penulis mengekstrak aspek menggunakan data *training* dengan algoritma LDA atau *Latent Dirichlet Allocation* lalu setelah aspek didapat maka kata-kata dari data *training* akan diubah menjadi vektor menggunakan fitur *Word2Vec*. Selanjutnya dilakukan lah pengklasifikasian analisis sentimen berbasis aspek menggunakan metode *Extreme Gradient Boosting* untuk menghasilkan model *XGBoost*. Setelah itu model akan diuji menggunakan data *testing* dan hasil evaluasi model disajikan dalam bentuk *confusion matrix* dan mendapatkan hasil rata-rata akurasi berdasarkan 4 aspek sebesar 90%.

Kata Kunci: analisis sentimen berbasis aspek, *word2vec*, *extreme gradient boosting*, *latent dirichlet allocation*, *confusion matrix*

ASPECT BASED SENTIMENT ANALYSIS OF DIGITAL WALLET APPLICATION REVIEWS USING THE EXTREME GRADIENT BOOSTING METHOD

ABSTRACT

The digital wallet application plays a crucial role in the development of modern financial systems. LinkAja is among the digital wallet applications widely used by the community. Public evaluation of this application significantly influences general perceptions and the identification of key aspects affecting user satisfaction. It is imperative for companies to understand user perspectives on the LinkAja application to serve as an evaluation basis for enhancing its performance. Aspect-based sentiment analysis is an effective tool for comprehending user opinions from diverse reviews. This research aims to develop a sentiment analysis model focusing on aspects within user reviews of digital wallet applications. The Extreme Gradient Boosting (XGBoost) method was chosen for its proficiency in addressing classification problems and handling large datasets. The author utilized 2000 user review data extracted from Google Play Store scraping for this study. These 2000 data points will be split into training and testing data in a 70:30 ratio. Each dataset will undergo preprocessing stages to ensure data cleanliness. Subsequently, the author will extract aspects from the training data using the Latent Dirichlet Allocation (LDA) algorithm. Once aspects are identified, words from the training data will be transformed into vectors using Word2Vec features. Following this, aspect-based sentiment analysis classification will be conducted using the Extreme Gradient Boosting method to generate the XGBoost model. The model will then be tested on the testing data, and the evaluation results will be presented in the form of a confusion matrix. The average accuracy results based on four aspects are found to be 90%.

Keywords: Aspect-based sentiment analysis, *word2vec*, *extreme gradient boosting*, *latent dirichlet allocation*, *confusion matrix*

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN.....	iv
UCAPAN TERIMA KASIH	v
ABSTRAK.....	vii
ABSTRACT.....	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	5
1.3. Tujuan Penelitian.....	5
1.4. Batasan Masalah.....	5
1.5. Manfaat Penelitian.....	6
1.6. Metodologi Penelitian	6
1.7. Sistematika Penulisan.....	7
BAB 2 LANDASAN TEORI.....	9
2.1. Sentimen Analisis	9
2.2. <i>Aspect-Based Sentiment Analysis</i>	10
2.3. <i>Text Preprocessing</i>	11
2.4. <i>Word Embedding</i>	12
2.5. <i>Latent Dirichlet Allocation</i>	14
2.6. <i>Extreme Gradient Boosting</i>	15
2.7. Penelitian Terdahulu	17
2.8. Perbedaan Penelitian	24
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	25
3.1. Dataset.....	25
3.2. Arsitektur Umum.....	26
3.2.1. <i>Text Preprocessing</i>	27
3.2.2. <i>Latent Dirichlet Allocation (LDA)</i>	31
3.2.3. <i>Word Embedding</i>	33

3.2.4.	<i>Extreme Gradient Boosting (XGBoost)</i>	36
3.3.	Perancangan Sistem.....	44
3.3.1.	Desain Halaman Beranda.....	44
3.3.2.	Desain Halaman <i>Training</i>	45
3.3.3.	Desain Halaman Data <i>Testing</i>	46
3.3.4.	Desain Halaman Uji Coba <i>Real Time</i>	47
3.4.	Metode Evaluasi	48
BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM		50
4.1.	Implementasi Sistem	50
4.1.1.	Spesifikasi Perangkat Keras dan Perangkat Lunak.....	50
4.1.2.	Implementasi Perancangan Tampilan Antarmuka.....	51
4.2.	Impelementasi Model	55
4.2.1.	Menyematkan Vector <i>Word2Vec</i> ke Dalam Aspek	55
4.2.2.	Pelatihan Model <i>Extreme Gradient Boosting</i>	56
4.2.3.	Pengujian Model	61
4.3.	Evaluasi Model.....	62
BAB 5 KESIMPULAN DAN SARAN		73
5.1.	Kesimpulan.....	73
5.2.	Saran.....	73
DAFTAR PUSTAKA		74

DAFTAR TABEL

Tabel 2.1	Tabel Penelitian Terdahulu	19
Tabel 3.1	Dataset hasil <i>scrapping</i> ulasan pengguna LinkAja	25
Tabel 3.2	Tahapan <i>Punctual Removal</i>	28
Tabel 3.3	Tahapan <i>Case Folding</i>	29
Tabel 3.4	Tahapan <i>Stopword Removal</i>	29
Tabel 3.5	Tahapan <i>Normalization</i>	30
Tabel 3.6	Tahapan <i>Stemming</i>	30
Tabel 3.7	Interpretasi Aspek	31
Tabel 3.8	Dataset untuk membangun pohon XGBoost	37
Tabel 3.9	Perhitungan nilai <i>error</i> atau <i>residuals</i>	38
Tabel 3.10	Perhitungan Nilai Prediksi pada Model 1	43
Tabel 3.11	<i>Confusion Matrix</i>	48
Tabel 4.1	Implementasi kalkulasi prediksi awal dan <i>residual</i>	57
Tabel 4.2	<i>Hyperparameter Tuning</i> Menggunakan Metode Grid Search CV ...	59
Tabel 4.3	Tabel Hasil <i>Training Data</i>	59
Tabel 4.4	Hasil Pengujian Model Menggunakan Data <i>Testing</i>	61
Tabel 4.5	<i>Confusion Matrix</i> Sentimen Negatif Aspek <i>Customer Service</i>	63
Tabel 4.6	<i>Confusion Matrix</i> Sentimen Netral Aspek <i>Customer Service</i>	64
Tabel 4.7	<i>Confusion Matrix</i> Sentimen Positif aspek <i>Customer Service</i>	64
Tabel 4.8	<i>Confusion Matrix</i> Sentimen Negatif Aspek Fitur Aplikasi	65
Tabel 4.9	<i>Confusion Matrix</i> Sentimen Netral Aspek Fitur Aplikasi	65
Tabel 4.10	<i>Confusion Matrix</i> Sentimen Positif Aspek Fitur Aplikasi	65
Tabel 4.11	<i>Confusion Matrix</i> Sentimen Negatif Aspek User Experience	66
Tabel 4.12	<i>Confusion Matrix</i> Sentimen Netral Aspek User Experience	66
Tabel 4.13	<i>Confusion Matrix</i> Sentimen Positif Aspek User Experience	66
Tabel 4.14	<i>Confusion Matrix</i> Sentimen Negatif Aspek Verifikasi	67
Tabel 4.15	<i>Confusion Matrix</i> Sentimen Netral Aspek Verifikasi	67
Tabel 4.16	<i>Confusion Matrix</i> Sentimen Positif Aspek Verifikasi	68

DAFTAR GAMBAR

Gambar 2.1 <i>Word2Vec Training Model</i>	13
Gambar 2.2 Visualisasi Model Grafis LDA (Blei <i>et al.</i> , 2003).....	14
Gambar 3.1 Arsitektur Umum.....	27
Gambar 3.2 Representasi Vector kata ‘aplikasi’	36
Gambar 3.3 Membangun pohon XGBoost	39
Gambar 3.4 Perhitungan <i>similarity</i> dan <i>gain</i>	40
Gambar 3.5 Split pada Pohon XGBoost	40
Gambar 3.6 <i>Split</i> pada Turunan Percabangan	41
Gambar 3.7 Cara Perhitungan <i>similarity</i> dan <i>gain</i> pada <i>split</i> Lanjutan	41
Gambar 3.8 Tahap <i>Tree Pruning</i>	42
Gambar 3.9 Perhitungan <i>Output Value</i>	42
Gambar 3.10 Desain Halaman Beranda	44
Gambar 3.11 Desain Halaman <i>Training</i>	45
Gambar 3.12 Desain Halaman <i>Data Testing</i>	46
Gambar 3.13 Desain Halaman Uji Coba <i>Real Time</i>	47
Gambar 4.1 Halaman Beranda	51
Gambar 4.2 Halaman <i>Training</i>	52
Gambar 4.3 Halaman hasil <i>Training Data</i>	52
Gambar 4.4 Halaman <i>Testing</i>	53
Gambar 4.5 Halaman Hasil <i>Testing Data</i>	53
Gambar 4.6 Evaluasi Aspek <i>Customer Service</i>	54
Gambar 4.7 Evaluasi Aspek Fitur Aplikasi	54
Gambar 4.8 Evaluasi Aspek <i>User Experience</i>	54
Gambar 4.9 Evaluasi Aspek Verifikasi	54
Gambar 4.10 Halaman Uji Coba <i>Real Time</i>	55

Gambar 4.11 <i>Vector</i> dari salah satu kalimat ulasan.....	56
Gambar 4.12 <i>Confusion Matrix</i> Aspek <i>Customer Service</i>	63
Gambar 4.13 <i>Confusion Matrix</i> Aspek Fitur Aplikasi	64
Gambar 4.14 <i>Confusion Matrix</i> Aspek <i>User Experience</i>	66
Gambar 4.15 <i>Confusion Matrix</i> Aspek Verifikasi.....	67

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Kemajuan teknologi dalam era digital telah mengubah semua hal jadi lebih efisien dan cepat. Revolusi digitalisasi telah memengaruhi berbagai aspek kehidupan masyarakat, termasuk layanan keuangan. Fenomena FinTech (*Financial Technology*), yang merupakan perpaduan antara keuangan dan teknologi, telah memungkinkan transaksi online yang terhubung dengan perangkat *smartphone* atau komputer, serta kartu yang terhubung ke sistem pembayaran digital yang aman (Manyika *et al.* 2016).

Dalam konteks ini, salah satu sektor yang mengalami perubahan adalah layanan keuangan, dengan peningkatan signifikan dalam penggunaan *digital wallet* atau dompet elektronik sebagai alat pembayaran (Uddin & Akhi, 2014). Keberhasilan *digital wallet* dalam menarik pengguna dapat dihubungkan dengan tingginya penetrasi *smartphone* dan internet di Indonesia, yang mendorong masyarakat beralih dari pembayaran konvensional ke uang elektronik (Bank Indonesia, 2019). Hal ini mencerminkan perubahan gaya hidup masyarakat yang lebih memilih menggunakan sistem pembayaran non-tunai.

Salah satu aplikasi *digital wallet* yang populer di Indonesia adalah LinkAja. Menurut riset yang dilakukan oleh lembaga penelitian *Snapcart* terkait perilaku konsumen saat bertransaksi menggunakan aplikasi pembayaran digital, LinkAja menempati peringkat terbawah di antara tiga pesaingnya dalam kategori *e-money* (Fajar, 2019). Data pengguna *e-money* pada tahun 2019 juga menunjukkan bahwa LinkAja menduduki peringkat terakhir dengan jumlah pengguna mencapai 23 juta

(Pusparisa, 2019). LinkAja menempati peringkat kelima dalam hasil survei konsumen *online*, dan terdapat perbedaan yang signifikan atau jarak yang cukup besar antara LinkAja dengan *e-wallet* pesaing di peringkat di atasnya seperti OVO, Gopay, ShopeePay, dan Dana (Pahlevi, 2021). Menurut penelitian yang dilakukan oleh (Meileny & Wijaksana, T.I. 2020) ada beberapa keluhan pengguna yang membuat LinkAja kurang diminati. Keluhan yang disampaikan berkaitan dengan jumlah *cashback* dan promo, kelengkapan serta keunggulan fitur layanan, performa aplikasi, mutu layanan pelanggan LinkAja, dan prosedur pengembalian dana. Berdasarkan data yang dijelaskan, penulis tertarik untuk menjadikan LinkAja sebagai objek yang diteliti pada penelitian ini.

Untuk mengumpulkan data ulasan pengguna tentang LinkAja yang bisa dilihat melalui *Google Play Store*, di mana terdapat ulasan positif dan negatif. Mengelola dan mengklasifikasikan ulasan dengan cara manual menjadi tugas yang sulit karena jumlah ulasan yang besar. Oleh karena itu, teknik *web scraping* digunakan untuk mengumpulkan data ini (Marres, Joan, & Wilson, 2013). Dalam penelitian ini, analisis sentimen berbasis aspek (ABSA) diterapkan menggunakan algoritma *Extreme Gradient Boosting* (XGBoost). Pendekatan ini telah terbukti efektif dalam identifikasi sentimen dan pengembangan produk layanan.

Analisis sentimen berbasis aspek dilakukan oleh Mohamad et al. (2017) untuk menilai produk menggunakan *Naïve Bayes*. Ada tiga komponen utama dalam penelitian ini: prapemrosesan data secara *part-of-speech* (POS), pemilihan fitur menggunakan metode *Chi Square*, dan klasifikasi aspek target berdasarkan polaritas menggunakan metode *Naive Bayes*. Hasil penelitian menunjukkan bahwa sistem dapat melakukan analisis sentimen berdasarkan aspek mencapai nilai *F1-Measure* sekitar 78,12%.

Turjaman & Budi (2022) melakukan penelitian terhadap aplikasi LinkAja pada twitter. Tujuan dari penelitian ini adalah melakukan analisis sentimen secara komprehensif yang berfokus pada beberapa aspek guna mengidentifikasi implikasi positif, negatif, dan netral dari ulasan yang diberikan kepada konsumen. Dataset yang digunakan berasal dari Twitter akun @linkaja selama periode Januari hingga Mei 2022. Penelitian ini menggunakan pencocokan *string* dengan bantuan paket *open-*

source Thefuzz untuk mengklasifikasikan aspek sebelum melanjutkan ke klasifikasi sentimen dengan bantuan algoritma *Support Vector Machine* (SVM). Hasil klasifikasi menunjukkan bahwa aplikasi LinkAja mendapat *feedback* negatif, terutama pada aspek kualitas produk (98% dari total ulasan) dan aspek tempat (100% total ulasan). Selain itu, aspek harga menghasilkan sentimen netral sekitar 89% dari total ulasan, dan aspek promosi menghasilkan sentimen positif sekitar 98% dari total ulasan.

Penelitian Ailiyya, S. (2020) adalah penelitian ABSA menggunakan algoritma *Support Vector Machine*. Data yang dianalisis diambil ulasan pengguna aplikasi Tokopedia di *Google Play Store*. Penelitian ini mencakup dua tahap klasifikasi, yaitu klasifikasi sentimen dan klasifikasi aspek menggunakan algoritma *Support Vector Machine*. Aspek yang dianalisis meliputi layanan, sistem, dan kebermanfaatan. Hasil penelitian menunjukkan bahwa parameter terbaik untuk model klasifikasi sentimen dan aspek adalah kernel linear dengan nilai $c=1$. Akurasi yang dihasilkan adalah 69,6% untuk klasifikasi sentimen dan 74,2% untuk klasifikasi aspek.

Ikegami & Darmawan (2022) melakukan studi pada ulasan aplikasi Noice. Analisis sentimen berbasis aspek dilakukan terhadap dataset ulasan aplikasi Noice sebagai respons terhadap peningkatan yang signifikan dalam konsumsi konten *audio* dalam beberapa tahun terakhir, terutama akibat dari pandemi COVID-19. Dalam upaya untuk menjaga kualitas layanan mereka, sangat penting untuk mengevaluasi ulasan yang diberikan oleh pengguna terhadap aplikasi mereka. Untuk melakukan analisis terhadap ulasan-ulasan tersebut, mereka menerapkan pendekatan analisis sentimen dan pemodelan topik untuk mengekstraksi polaritas sentimen dan topik-topik yang ada pada masing-masing polaritas sentimen. Pada penelitian ini, mereka menggunakan algoritma *XGBoost* dan *Latent Dirichlet Allocation* (LDA) untuk menganalisis ulasan yang dipublikasikan di *Google Play Store*. Hasil dari analisis sentimen mencapai metrik akurasi sebesar 86,8%, presisi sebesar 83,9%, *recall* sebesar 77,9%, dan *F1-score* sebesar 80,2%. Selain itu, melalui proses pemodelan topik, mereka berhasil mengidentifikasi 3 topik untuk ulasan yang bersentimen positif dan 6 topik untuk ulasan yang bersentimen negatif.

Ginanjari & Mukti (2019) melakukan penelitian tentang pengklasifikasian *Parkinson Disease* (PD) menggunakan metode *XGBoost*. Pada awalnya seluruh fitur

dasar digunakan dalam algoritma *XGBoost* digunakan lalu diperoleh skor akurasi model 84,80% lalu untuk meningkatkan model peneliti melakukan pemilihan fitur dengan memplot kepentingan fitur yang menyebabkan fitur *IocShimmer* ($Fscore=3$) dikeluarkan dari model. Setelah pemilihan fitur dilakukan, skor akurasi dari model telah meningkat menjadi 85,60%.

Giffari, M.R.A. (2022) melakukan penelitian Analisis Sentimen Berbasis Aspek pada ulasan aplikasi Tangerang *Live* karena meskipun aplikasi Tangerang *LIVE* mendapatkan *rating* yang cukup baik di *play store*, layanannya belum mencapai tingkat optimal. Penelitian ini menggunakan kumpulan data ulasan aplikasi Tangerang *LIVE* yang diperoleh melalui proses *scraping* di *Google Play Store*. Dalam penelitian ini, metode *Latent Dirichlet Allocation* (LDA) digunakan untuk pemodelan topik, sementara metode *Naïve Bayes* digunakan untuk mengklasifikasikan sentimen. Hasil pemodelan topik menggunakan LDA mengidentifikasi empat aspek utama, yaitu tampilan pengguna (*user interface*), kinerja dan performa (*functionality and performance*), dukungan (*supports*), dan pembaruan (*updates*). Penelitian ini berhasil mencapai tingkat akurasi sekitar 87,80% menggunakan algoritma *Naïve Bayes*, dan evaluasi kinerja algoritma menggunakan kurva ROC menghasilkan nilai AUC sekitar 0,94.

Haumahu et al. (2020) mengklasifikasikan antara berita *hoax* dan *valid* menggunakan metode *XGBoost*. Dataset yang digunakan dalam penelitian ini adalah berita Indonesia tentang Indonesia sendiri dan dunia dari 2015 hingga awal 2020. Penelitian menggunakan 500 data berita yang terdiri dari 250 berita *valid* dan 250 berita *hoax* dibagi menjadi 80% *data train* dan 20% *data test*. Hasil dari penelitian ini menyimpulkan bahwa mesin model pembelajaran yang dibuat menggunakan *XGBoost* memiliki nilai akurasi sebesar 89% dengan nilai presisi 90% dan *recall* 80%.

Dengan mengacu pada penelitian terdahulu yang sudah dijelaskan, peneliti memutuskan untuk membuat sebuah penelitian yang berjudul **"ANALISIS SENTIMEN BERBASIS ASPEK TERHADAP REVIEW APLIKASI DIGITAL WALLET MENGGUNAKAN METODE EXTREME GRADIENT BOOSTING."**

1.2. Rumusan Masalah

Dalam aplikasi tertentu, ulasan atau *feedback* dari pengguna akan selalu ada yang berkaitan dengan masalah seperti lambatnya kinerja di halaman tertentu, fitur yang tidak berfungsi, bug, tata letak menu yang membingungkan, dan lain sebagainya. Namun, sebagian ulasan seperti ini belum memberikan petunjuk yang cukup konkret untuk perbaikan dan pengembangan aplikasi karena klasifikasi aspeknya masih terlalu umum. Oleh karena itu, diperlukan analisis sentimen yang lebih fokus pada aspek-aspek tertentu yang diekstraksi dari ulasan pengguna pada aplikasi dompet digital menggunakan metode *Extreme Gradient Boosting*.

1.3. Tujuan Penelitian

Penelitian yang dikerjakan memiliki tujuan mengklasifikasikan kalimat-kalimat dalam ulasan pengguna aplikasi *digital wallet* LinkAja berdasarkan berbagai aspek yang telah diekstrak menggunakan *Latent Dirichlet Allocation* lalu diidentifikasi menggunakan metode *Extreme Gradient Boosting* dengan pemanfaatan *word embedding Word2Vec*.

1.4. Batasan Masalah

Batasan-batasan yang diterapkan dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya menggunakan ulasan yang ditulis dalam Bahasa Indonesia.
2. Aplikasi *digital wallet* yang menjadi subjek penelitian adalah LinkAja.
3. Analisis sentimen berdasarkan aspek akan dilakukan dengan mempertimbangkan kata-kata dalam ulasan, sedangkan angka, emoji, dan gambar tidak akan dipertimbangkan.
4. Dataset ulasan yang digunakan diambil dari *Google Play Store* melalui proses pengambilan data (*scrapping*) dan disimpan dalam format *Comma Separated Value* (CSV).
5. Metode *Latent Dirichlet Allocation* (LDA) akan digunakan untuk mengekstrak empat aspek yang akan menjadi fokus analisis.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini meliputi:

1. Memberikan pengetahuan yang berharga kepada peneliti dan pembaca tentang performa metode *Word2vec*, *Latent Dirichlet Allocation*, dan algoritma *Extreme Gradient Boosting* dalam melakukan Analisis Sentimen Berbasis Aspek terhadap ulasan pengguna pada aplikasi dompet digital.
2. Menyediakan sumber referensi yang berharga bagi penelitian masa depan dalam bidang Analisis Sentimen Berbasis Aspek menggunakan *Extreme Gradient Boosting*.

1.6. Metodologi Penelitian

Tahapan penelitian yang akan dikerjakan adalah:

1.6.1. Studi Literatur

Dalam fase ini, peneliti menghimpun data dan bahan referensi yang relevan dengan penelitian ini. Ini termasuk jurnal, artikel, skripsi, serta penelitian terdahulu lainnya yang membahas topik terkait, seperti pemrosesan teks, analisis sentimen berbasis aspek, teknik *word embedding word2vec*, metode *latent dirichlet allocation*, dan algoritma *extreme gradient boosting*.

1.6.2. Analisis Permasalahan

Jika sudah selesai tahap Studi literatur, maka langkah berikutnya ialah menganalisis permasalahan. Analisis permasalahan merupakan langkah yang ditempuh oleh peneliti untuk memahami konsep penggunaan *Extreme Gradient Boosting* dalam penelitian ini, terutama dalam konteks menganalisis kalimat ulasan dari pengguna aplikasi dompet digital.

1.6.3. Perancangan

Setelah menganalisis permasalahan, langkah selanjutnya adalah merancang arsitektur umum sistem, memilih *dataset* yang akan dipakai, dan mendesain antarmuka sistem yang akan digunakan dalam bagian implementasi.

1.6.4. Implementasi

Setelah proses perancangan, langkah berikutnya adalah implementasi. Dalam tahap ini, peneliti akan melaksanakan implementasi dari rancangan yang telah disiapkan sebelumnya. Implementasi akan melibatkan penerapan metode *Extreme Gradient Boosting* dengan menggunakan bahasa pemrograman Python.

1.6.5. Pengujian

Setelah proses implementasi selesai, langkah berikutnya adalah tahap pengujian. Dalam tahap ini, peneliti akan menguji sistem untuk mengevaluasi tingkat akurasi penggunaan metode word embedding *Word2vec* dan algoritma *Extreme Gradient Boosting*.

1.6.6. Penyusunan Laporan

Setelah proses pengujian selesai, langkah selanjutnya adalah penyusunan laporan. Pada tahap ini, peneliti melakukan penyusunan laporan yang mencakup seluruh rangkaian proses penelitian sampai hasil akhirnya.

1.7. Sistematika Penulisan

Bab 1: Pendahuluan

Bagian pendahuluan mencakup rangkuman tentang alasan di balik pemilihan judul "Analisis Sentimen Berbasis Aspek Terhadap *Review Aplikasi Digital Wallet Menggunakan Metode Extreme Gradient Boosting*," dan tahap penelitian yang akan dilakukan.

Bab 2: Landasan Teori

Bagian ini berfokus pada fondasi teoritis pembelajaran, yang mencakup konsep-konsep dasar yang relevan dalam menyelesaikan masalah. Ini akan digunakan untuk memeriksa isu-isu yang terkait dengan penelitian ini, seperti analisis sentimen berbasis aspek, teknik *word embedding Word2Vec*, algoritma *Extreme Gradient Boosting*, dan proses ekstraksi aspek dengan menggunakan metode LDA.

Bab 3: Analisis dan Perancangan Sistem

Dalam bab ini, akan dibahas analisis masalah penelitian dan perancangan sistem yang mengaplikasikan algoritma *Extreme Gradient Boosting* untuk mengkategorikan ulasan pengguna berdasarkan aspek yang mengandung sentimen positif, netral, atau negatif. Bab ini juga akan berisi penjelasan arsitektur umum dari sistem yang dikembangkan.

Bab 4: Implementasi dan Pengujian Sistem

Bagian ini menjelaskan tahap pelaksanaan implementasi berdasarkan rancangan sistem yang telah dirancang sebelumnya. Selain itu, bab ini juga akan memaparkan hasil pengujian sistem yang telah dibuat dan akan dilakukan evaluasi terhadap hasil yang sudah diperoleh.

Bab 5: Kesimpulan dan Saran

Di bagian terakhir ini, akan disajikan ringkasan dari hasil penelitian yang telah berhasil dilakukan, termasuk apakah target penelitian berhasil dicapai atau tidak, serta akan dibahas saran-saran yang diajukan oleh penulis mengenai potensi pengembangan penelitian di masa selanjutnya.

BAB 2

LANDASAN TEORI

2.1. Sentimen Analisis

Analisis sentimen dan eksplorasi opini merupakan studi yang befokus pada pendapat, perasaan, penilaian, pandangan, sikap, dan ekspresi emosi yang ditunjukkan oleh individu terhadap berbagai entitas, seperti produk, layanan, organisasi, individu, isu, peristiwa, topik, dan atribut-atribut yang terkait (Liu & Zhang, 2012).

Analisis sentimen umumnya memproses sebuah teks (kalimat, paragraf, buku, dll) dan menghasilkan skor kuantitatif atau klasifikasi untuk menunjukkan apakah algoritma menganggap teks tersebut menyampaikan sentimen positif atau negatif (Saldana, 2018).

Ada beberapa jenis analisis sentimen yang umum digunakan. Beberapa di antaranya meliputi:

1. Analisis Sentimen Berbasis Kata (*Lexicon-Based Sentiment Analysis*): Metode ini menggunakan leksikon atau daftar kata-kata yang memiliki nilai sentimen yang telah ditentukan sebelumnya. Setiap kata dalam teks diberi skor sentimen, dan hasilnya dijumlahkan untuk menghitung sentimen keseluruhan dalam teks.
2. Analisis Sentimen yang Mengandalkan Pembelajaran Mesin (*Machine Learning-Based Sentiment Analysis*): Pendekatan ini melibatkan pemanfaatan algoritma pembelajaran mesin untuk mengategorikan teks ke dalam kategori sentimen tertentu, seperti positif, negatif, atau netral. Algoritma pembelajaran mesin ini dilatih melalui penggunaan dataset yang telah terklasifikasi sebelumnya.

3. Analisis Sentimen Berbasis Aspek (*Aspect-Based Sentiment Analysis*): metode analisis ini fokus pada aspek-aspek tertentu dari suatu subjek atau entitas. Misalnya, dalam ulasan produk, analisis ini membedakan sentimen terkait kualitas produk, layanan pelanggan, harga, dan lainnya.
4. Analisis Sentimen Berbasis Emosi (*Emotion-Based Sentiment Analysis*): Pendekatan ini mengevaluasi emosi yang terkandung dalam teks, seperti senang, sedih, marah, atau bahagia. Ini melibatkan identifikasi dan klasifikasi emosi berdasarkan kata-kata atau ungkapan tertentu.
5. Analisis Sentimen Berbasis Jaringan Sosial (*Social Network-Based Sentiment Analysis*): Jenis analisis ini melibatkan pengumpulan data dari platform media sosial untuk menganalisis sentimen publik terhadap suatu topik. Data yang dikumpulkan dapat berupa *tweet*, postingan Facebook, atau komentar di *platform* lainnya.

2.2. *Aspect-Based Sentiment Analysis*

Analisis Sentimen Berbasis Aspek (*Aspect-Based Sentiment Analysis* atau ABSA) adalah teknik analisis berkehalusan tinggi yang membedah sentimen yang terungkap dalam teks menjadi unit opini yang lebih kecil, di mana setiap unit opini berkorespondensi dengan aspek spesifik dari entitas yang dibahas (Schouten et al., 2016).

Analisis sentimen berbasis aspek (ABSA) telah menjadi fokus penelitian analisis sentimen dan telah diterapkan secara luas dalam memahami pendapat pengguna serta meningkatkan produk dan layanan (Fan et al., 2020)

Dalam ABSA, fokus utama adalah pada pemahaman sentimen yang dihubungkan dengan atribut-atribut khusus yang ada dalam teks, seperti fitur produk, layanan pelanggan, harga, kualitas, dan lain sebagainya. Pentingnya ABSA terletak pada kemampuannya untuk memberikan pandangan yang lebih rinci dan mendalam tentang bagaimana masing-masing aspek mempengaruhi pandangan umum atau opini. Hal ini sangat bermanfaat dalam pemahaman tentang kekuatan dan kelemahan suatu produk atau layanan, serta meningkatkan performa produk atau layanan.

Dalam ABSA, beberapa langkah yang umumnya diikuti meliputi:

1. Pengenalan Aspek: Identifikasi aspek-aspek atau fitur-fitur tertentu yang ingin dianalisis dalam teks.
2. Pemberian Sentimen: Penentuan sentimen negatif, netral, atau positif yang terkait dengan masing-masing aspek.
3. Klasifikasi Sentimen: Pengelompokan teks ke dalam kategori sentimen berdasarkan aspek-aspek yang telah ditentukan.

2.3. Text Preprocessing

Text Preprocessing adalah langkah penting dalam pengolahan bahasa alami yang melibatkan serangkaian teknik untuk membersihkan, mengubah, dan mempersiapkan teks mentah sebelum dijalani oleh algoritma analisis. Tujuan utama dari *text preprocessing* adalah untuk meningkatkan kualitas dan akurasi analisis teks dengan menghilangkan *noise*, mengatasi variasi kata-kata, serta menghasilkan representasi teks yang lebih sesuai untuk pemrosesan lebih lanjut.

Text preprocessing melibatkan beragam tugas yang mempersiapkan data teks untuk analisis. Ini dapat mencakup tugas-tugas seperti menghapus informasi yang tidak relevan, mengubah teks menjadi format yang sesuai untuk analisis, dan mengurangi ruang fitur berdimensi tinggi (Manning et al., 2008).

Text preprocessing adalah langkah penting untuk sebagian besar tugas pemrosesan bahasa alami. Ini melibatkan membersihkan dan mengubah teks mentah menjadi format yang lebih dapat digunakan (Bird et al., 2009).

Beberapa teknik umum dalam *text preprocessing* meliputi:

1. Tokenisasi: Proses pembagian teks menjadi unit-unit terpisah, seperti kata-kata atau frasa-frasa.
2. *Case Folding*: Merubah seluruh huruf dalam teks menjadi huruf kecil guna menghindari perbedaan yang tidak diperlukan dalam kata yang sama.

3. Pembersihan Teks: Menghilangkan karakter khusus, tanda baca, dan simbol yang tidak perlu.
4. *Stemming* dan *Lemmatization*: Merubah kata-kata kembali kepada bentuk dasarnya untuk mengatasi variasi kata yang sebenarnya memiliki makna yang sama.
5. *Stopword Removal*: Menghilangkan kata-kata yang bersifat umum yang tidak memberikan makna penting seperti "dan," "atau," "di," dan sejenisnya.
6. Normalisasi: Merubah bentuk-bentuk khusus kata menjadi bentuk standar (misalnya, mengubah "dr" menjadi "dari").

2.4. *Word Embedding*

Word Embedding adalah teknik dalam pengolahan bahasa alami yang bertujuan untuk mengubah kata-kata menjadi representasi vektor dalam ruang multidimensional. Representasi ini memungkinkan komputer untuk memahami hubungan semantik antara kata-kata berdasarkan kedekatan vektor mereka. *Word embedding* membantu mengatasi masalah representasi kata yang kaku dan memperkaya pemahaman tentang makna kata-kata dalam konteks.

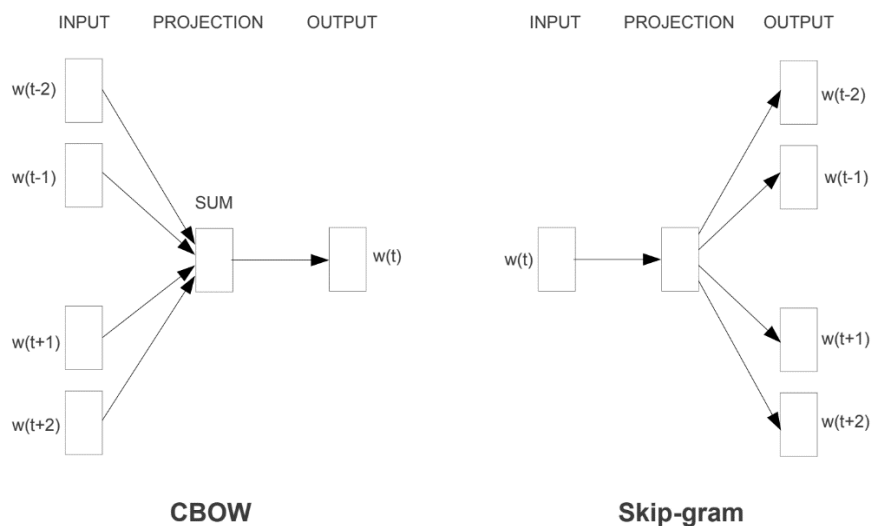
Menurut Mikolov *et al* (2013) *Word embedding* adalah representasi vektor padat dari kata-kata, biasanya dalam beberapa ratus dimensi. Vektor-vektor ini menangkap makna semantik dan digunakan untuk meningkatkan kinerja berbagai tugas pemrosesan bahasa alami.

Word2Vec adalah salah satu teknik *word embedding* yang populer dalam pengolahan bahasa alami. Tujuannya adalah untuk mengkonversi kata-kata menjadi representasi vektor dalam ruang *multidimensional*, di mana kata-kata yang sering muncul bersama memiliki representasi vektor yang mendekat satu sama lain. *Word2Vec* adalah kerangka kerja yang efisien secara komputasi untuk mempelajari *word embedding* dari data teks berskala besar.

Word2Vec menggunakan metode pelatihan yang melibatkan jaringan saraf tiruan (*neural network*) dan optimisasi untuk menghasilkan vektor representasi kata. Hasil dari *Word2Vec* adalah vektor berdimensi rendah yang merepresentasikan kata-

kata dalam ruang kontinu. Vektor-vektor ini dapat digunakan untuk masing-masing kata dalam berbagai tugas pemrosesan bahasa alami, seperti klasifikasi teks, pencarian semantik, dan pemahaman teks. Menurut Mikolov *et al* (2013) *Word2Vec* adalah kerangka kerja yang efisien secara komputasi untuk mempelajari *word embedding* dari data teks berskala besar,

Word2Vec memiliki dua arsitektur utama: *Skip-gram* dan *Continuous Bag of Words* (CBOW). Dalam arsitektur *Skip-gram*, model berusaha memprediksi kata-kata di sekitar suatu kata yang diberikan, sementara dalam arsitektur CBOW, model mencoba memprediksi kata yang diberikan berdasarkan kata-kata di sekitarnya. Proses pelatihan ini memungkinkan model untuk memahami hubungan semantik dan sintaktik antara kata-kata.



Gambar 2.1 *Word2Vec Training Model* (Sumber:

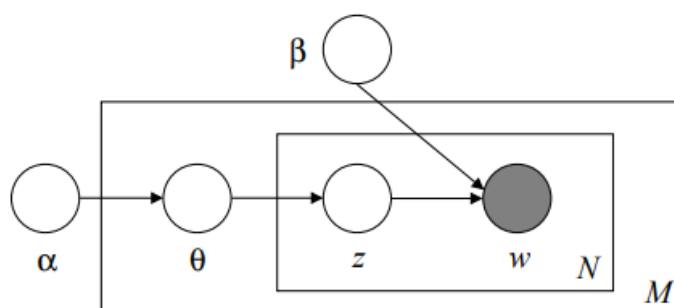
<https://machinelearningmastery.com/what-are-word-embeddings/>)

2.5. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) adalah model probabilistik yang digunakan dalam pengolahan bahasa alami dan analisis topik. Tujuannya adalah untuk mengidentifikasi topik-topik yang tersembunyi dalam koleksi dokumen dan bagaimana kata-kata terkait dengan topik-topik tersebut. LDA beroperasi berdasarkan asumsi bahwa setiap dokumen adalah campuran dari beberapa topik, dan setiap kata dalam dokumen berasal dari salah satu topik dengan probabilitas tertentu.

Proses LDA melibatkan langkah-langkah seperti inisialisasi topik, pelatihan model, dan inferensi topik. Model LDA menerima matriks dokumen-kata sebagai masukan dan menghasilkan dua distribusi: distribusi topik per dokumen dan distribusi kata per topik. Dengan demikian, LDA memetakan dokumen dan kata-kata dalam ruang topik, memungkinkan analisis yang lebih baik tentang struktur topik dalam koleksi dokumen.

Menurut Griffiths *et al* (2004) LDA mewakili dokumen sebagai campuran topik yang menghasilkan kata-kata dengan probabilitas tertentu. Pada gambar 2.2 menampilkan representasi model grafis LDA.



Gambar 2.2 Visualisasi Model Grafis LDA (Blei *et al.*, 2003)

Menurut (Astuti, 2020) LDA didefinisikan dalam notasi sebagai berikut:

- Kata dianggap sebagai elemen dasar dalam data yang bersifat diskrit.
- Sebuah dokumen dilihat sebagai urutan kata-kata N yang dinotasikan dengan $w = (w_1, w_2, \dots, w_N)$ dimana w_N merupakan barisan kata ke- n .
- Sebuah korpus merujuk pada koleksi M dokumen, yang dinotasikan dengan $D = (w_1, w_2, \dots, w_M)$.

LDA mengasumsikan tahapan generatif berikut untuk setiap dokumen w dalam korpus D (Blei *et al.*, 2003):

- Pilih $N \sim \text{Poisson}(\beta)$.
- Pilih $\theta \sim \text{Dir}(\alpha)$.
- Untuk setiap N kata w_n :
 - i. Pilih topik $z_n \sim \text{Multinomial}(\theta)$.
 - ii. Pilih kata w_n dari $p(w_n | z_n, \beta)$, probabilitas *multinomial* yang dikondisikan pada topik z_n .

2.6. *Extreme Gradient Boosting*

Extreme Gradient Boosting (XGBoost) adalah metode *ensemble learning* yang populer dalam analisis prediksi dan klasifikasi. Ini adalah variasi dari algoritma gradient boosting yang dikembangkan untuk mengatasi beberapa masalah dan meningkatkan kinerja. XGBoost memadukan konsep *boosting* dan regularisasi untuk menghasilkan model yang kuat dan tahan terhadap *overfitting*. XGBoost adalah implementasi *framework gradient boosting* yang efisien dan dapat ditingkatkan (Chen & Guestrin, 2016)

XGBoost bekerja dengan cara menggabungkan beberapa model lemah (misalnya, pohon keputusan) menjadi model yang lebih kuat. Pada setiap iterasi, XGBoost akan mengevaluasi performa model yang ada dan memberikan bobot yang lebih tinggi pada data yang sulit dijelaskan oleh model sebelumnya. Ini membantu model berfokus pada data yang paling kompleks untuk ditingkatkan dalam iterasi berikutnya.

Adapun tahap-tahap proses klasifikasi dengan algoritma XGBoost dijelaskan oleh (Ikegami & Darmawan, 2022):

1. Membuat perkiraan awal

Langkah pertama yang dilakukan adalah membuat perkiraan awal, yaitu memberikan setiap objek data probabilitas sebesar 0,5 untuk klasifikasi ke dalam kelas tertentu.

2. Menghitung residual

Setelah melakukan prediksi awal, langkah berikutnya adalah membangun *tree* berdasarkan residual dari prediksi awal. Residual dapat dihitung menggunakan rumus berikut.

$$Residual = Nilai Observasi - Nilai Prediksi \quad (2,1)$$

3. Menghitung gain

Sebelum menghitung *gain*, nilai *similarity score* dari setiap *node* perlu dihitung menggunakan rumus.

$$Similarity\ score = \frac{(\sum r_i)^2}{\sum |(p)x(1-p)| + \lambda} \quad (2,2)$$

Keterangan:

r_i = residual ke-i, p = probabilitas sebelumnya, λ = parameter regularisasi

Jika sudah menghitung nilai *Similarity score*, maka nilai *gain* dapat dihitung menggunakan rumus. Fitur dan *split point* terbesar dari fitur tersebut akan dipilih untuk melakukan *splitting* jika didapat nilai *gain* terbesar.

$$Gain = Left\ Similarity + Right\ Similarity - Root\ Similarity \quad (2,3)$$

4. Menghitung cover

Nilai *cover* berguna untuk menetapkan jumlah minimum *residual* pada *leaf* dari suatu *tree*. Nilai *cover* dapat dihitung menggunakan rumus.

$$Cover = \sum |p \times (1 - p)| \quad (2,4)$$

5. Menghitung probabilitas kelas

Untuk memperkirakan probabilitas kelas suatu objek data, maka perlu dihitung *output* dari *node*-nya menggunakan rumus.

$$Output = \frac{Residual}{\sum |(p)x(1-p)| + \lambda} \quad (2,5)$$

Selanjutnya, nilai perkiraan probabilitas awal yang dibuat pada langkah pertama harus diubah menjadi bentuk $\log(odds)$ untuk itu, digunakan rumus

$$\log(odds) = \log\left(\frac{p}{1-p}\right) \quad (2,6)$$

Setelah itu, akan dihitung $\log(odds)$ menggunakan persamaan

$$\log(odds)Prediction = \log(odds) + (\varepsilon \times Output) \quad (2,7)$$

Untuk mendapatkan probabilitas kelas dari objek data, ubah kembali $\log(odds)$ menjadi probabilitas dengan rumus.

$$Probability = \frac{e^{\log(odds)}}{1+e^{\log(odds)}} \quad (2,8)$$

Keterangan:

P = prediksi, ε = *learning rate*

Probabilitas ini menggambarkan peluang objek data masuk ke dalam suatu kelas.

Kalkulasi ini akan diulang dari langkah kedua untuk membuat *tree* tambahan.

2.7. Penelitian Terdahulu

Analisis sentimen berbasis aspek dilakukan oleh Mohamad et al. (2017) untuk menilai produk menggunakan *Naïve Bayes*. Ada tiga komponen utama dalam penelitian ini: prapemrosesan data secara *part-of-speech* (POS), pemilihan fitur menggunakan metode *Chi Square*, dan klasifikasi aspek target berdasarkan polaritas menggunakan metode *Naive Bayes*. Hasil penelitian menunjukkan sistem dapat melakukan analisis sentimen berdasarkan aspek mencapai nilai *F1-Measure* sekitar 78,12%.

Turjaman & Budi (2022) melakukan penelitian terhadap aplikasi LinkAja pada twitter. Tujuan dari penelitian ini adalah melakukan analisis sentimen secara komprehensif yang berfokus pada beberapa aspek guna mengidentifikasi implikasi positif, negatif, dan netral dari ulasan yang diberikan kepada konsumen. *Dataset* dalam penelitian ini berasal dari Twitter akun @linkaja selama periode Januari hingga Mei 2022. Pencocokan *string* dengan bantuan paket *open-source Thefuzz* digunakan untuk mengklasifikasikan aspek sebelum melanjutkan ke klasifikasi sentimen dengan bantuan algoritma *Support Vector Machine* (SVM). Hasilnya menunjukkan aplikasi LinkAja mendapat feedback negatif, terutama pada aspek kualitas produk (98% dari total ulasan) dan aspek tempat (100% total ulasan). Selain itu, aspek harga menghasilkan sentimen netral sekitar 89% dari total ulasan, dan aspek promosi menghasilkan sentimen positif sekitar 98% dari total ulasan.

Penelitian Ailiyya, S. (2020) adalah penelitian ABSA menggunakan algoritma *Support Vector Machine*. Data yang dianalisis diambil ulasan pengguna aplikasi

Tokopedia di *Google Play Store*. Penelitian ini mencakup dua tahap klasifikasi, yaitu klasifikasi sentimen dan klasifikasi aspek menggunakan algoritma *Support Vector Machine*. Aspek yang dianalisis meliputi layanan, sistem, dan kebermanfaatan. Hasil penelitian menunjukkan bahwa parameter terbaik untuk model klasifikasi sentimen dan aspek adalah *kernel linear* dengan nilai $c=1$. Akurasi yang dihasilkan adalah 69,6% untuk klasifikasi sentimen dan 74,2% untuk klasifikasi aspek.

Ginanjari & Mukti (2019) melakukan suatu studi mengenai pengklasifikasian *Penyakit Parkinson* (PD) menggunakan pendekatan *XGBoost*. Pada tahap awal, semua fitur dasar diterapkan dalam algoritma *XGBoost*, menghasilkan tingkat akurasi model sebesar 84,80%. Selanjutnya, dengan tujuan meningkatkan kinerja model, peneliti melakukan seleksi fitur dengan menganalisis tingkat kepentingan fitur-fitur tersebut. Langkah ini menghasilkan penghilangan fitur *IocShimmer* (dengan $Fscore=3$) dari model. Setelah tahap seleksi fitur, model mencapai tingkat akurasi yang lebih tinggi, yaitu sekitar 85,60%.

Haumahu et al. (2020) mengklasifikasikan antara berita *hoax* dan *valid* menggunakan metode *XGBoost*. Dataset yang digunakan dalam penelitian ini adalah berita Indonesia tentang Indonesia sendiri dan dunia dari 2015 hingga awal 2020. Penelitian menggunakan 500 data berita yang terdiri dari 250 berita *valid* dan 250 berita *hoax* dibagi menjadi 80% data *train* dan 20% data *test*. Hasil dari penelitian ini menyimpulkan bahwa mesin model pembelajaran yang dibuat menggunakan *XGBoost* memiliki nilai akurasi sebesar 89% dengan nilai presisi 90% dan *recall* 80%.

Ikegami & Darmawan (2022) melakukan studi pada ulasan aplikasi Noice. Analisis sentimen berbasis aspek dilakukan terhadap dataset ulasan aplikasi Noice sebagai respons terhadap peningkatan yang signifikan dalam konsumsi konten *audio* dalam beberapa tahun terakhir, terutama akibat dari pandemi COVID-19. Dalam upaya untuk menjaga kualitas layanan mereka, sangat penting untuk mengevaluasi ulasan yang diberikan oleh pengguna terhadap aplikasi mereka. Untuk melakukan analisis terhadap ulasan-ulasan tersebut, mereka menerapkan pendekatan analisis sentimen dan pemodelan topik untuk mengekstraksi polaritas sentimen dan topik-topik yang ada pada masing-masing polaritas sentimen. Pada penelitian ini, mereka menggunakan algoritma *XGBoost* dan *Latent Dirichlet Allocation* (LDA) untuk

menganalisis ulasan yang dipublikasikan di *Google Play Store*. Hasil dari analisis sentimen mencapai metrik akurasi sebesar 86,8%, presisi sebesar 83,9%, *recall* sebesar 77,9%, dan *F1-score* sebesar 80,2%. Selain itu, melalui proses pemodelan topik, mereka berhasil mengidentifikasi 3 topik untuk ulasan yang bersentimen positif dan 6 topik untuk ulasan yang bersentimen negatif.

Giffari, M.R.A. (2022) melakukan penelitian Analisis Sentimen Berbasis Aspek pada ulasan aplikasi Tangerang *Live* karena meskipun aplikasi Tangerang *LIVE* mendapatkan *rating* yang cukup baik di *play store*, layanannya belum mencapai tingkat optimal. Penelitian ini menggunakan kumpulan data ulasan aplikasi Tangerang *LIVE* yang diperoleh melalui proses *scraping* di *Google Play Store*. Dalam penelitian ini, metode *Latent Dirichlet Allocation* (LDA) digunakan untuk pemodelan topik, sementara metode *Naïve Bayes* digunakan untuk mengklasifikasikan sentimen. Hasil pemodelan topik menggunakan LDA mengidentifikasi empat aspek utama, yaitu tampilan pengguna (*user interface*), kinerja dan performa (*functionality and performance*), dukungan (*supports*), dan pembaruan (*updates*). Penelitian ini berhasil mencapai tingkat akurasi sekitar 87,80% menggunakan algoritma *Naïve Bayes*, dan evaluasi kinerja algoritma menggunakan kurva ROC menghasilkan nilai AUC sekitar 0,94.

Tabel 2.1 Tabel Penelitian Terdahulu

No.	Peneliti	Judul	Algoritma	Keterangan
1	Mohamad et al. (2022)	<i>Aspect-based sentiment analysis to review products using Naïve Bayes</i>	<i>Naïve Bayes</i>	Penelitian terdiri dari tiga langkah utama, yaitu tahap pra-pemrosesan data yang mencakup tanda-tanda <i>part-of-speech</i> (POS), seleksi fitur melalui penggunaan <i>Chi</i>

				<p><i>Square</i>, dan klasifikasi polaritas sentimen aspek dengan <i>Naïve Bayes</i> yang menghasilkan <i>F1-Measure</i> tertinggi sebesar 78,12%.</p>
2	Turjaman & Budi (2022)	<p>Analisis Sentimen berbasis aspek <i>marketing mix</i> terhadap ulasan aplikasi dompet digital (studi kasus: aplikasi LinkAja pada twitter).</p>	<p><i>Support Vector Machine</i></p>	<p>Penelitian Analisis sentimen berbasis aspek pada aplikasi di twitter yang menghasilkan sentimen negatif terutama pada aspek produk, dengan persentase 98% dari total ulasan, serta pada aspek tempat dengan persentase 100% dari total ulasan. Di sisi lain, aspek harga menerima sentimen netral</p>

				sebanyak 89% dari total ulasan, dan aspek promosi memperoleh sentimen positif sebanyak 98% dari total ulasan.
3	Ailiyya, S. (2020)	Analisis Sentimen Berbasis Aspek pada Ulasan Aplikasi Tokopedia Menggunakan <i>Support Vector Machine</i>	<i>Support Vector Machine</i>	Penelitian analisis sentimen berdasarkan aspek <i>usefulness</i> , <i>system</i> , dan <i>service</i> ulasan pengguna Tokopedia yang menghasilkan akurasi sebesar 69,6% untuk klasifikasi sentimen dan 74,2% untuk klasifikasi aspek.
4	Ginanjari & Mukti (2019)	<i>Implementation of xgboost for classification of parkinson's disease</i>	<i>Extreme gradient boost</i>	Penelitian tentang penyakit <i>Parkinson</i> yang menghasilkan tingkat akurasi model sebesar

				<p>84,80%.</p> <p>Selanjutnya, dengan tujuan meningkatkan kinerja model, peneliti melakukan seleksi fitur dengan menganalisis tingkat kepentingan fitur-fitur tersebut. Setelah tahap seleksi fitur, akurasi menjadi 85,60%.</p>
5	Haumahu et al., (2020)	<p><i>Classification of Fake News for Indonesian News using Extreme Gradient Boosting (XGBoost).</i></p>	<i>Extreme gradient boost</i>	<p>Penelitian tentang klasifikasi berita <i>hoax</i> Bahasa Indonesia yang menghasilkan akurasi sebesar 89%, dengan nilai presisi mencapai 90% dan <i>recall</i> mencapai 80%.</p>
6	Ikegami & Darmawan	Analisis Sentimen dan	<i>Extreme gradient</i>	Penelitian analisis

	(2022)	Pemodelan Ulasan Aplikasi Noice Menggunakan <i>XGBoost</i> dan LDA	<i>boost, latent dirichlet allocation.</i>	sentimen berbasis aspek ulasan aplikasi noice yang menghasilkan analisis metrik akurasi, presisi, <i>recall</i> , serta <i>F1- score</i> masing- masing sebesar 86,8%, 83,9%, 77,9%, dan 80,2%. Pemodelan topik berhasil mengidentifikasi 3 topik untuk ulasan positif dan 6 topik untuk ulasan negatif.
7	Giffari, M.R.A. (2022)	Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Tangerang Live Menggunakan <i>Latent Dirichlet Allocation</i> dan <i>Naïve Bayes</i>	<i>Naïve Bayes, Latent Dirichlet Allocation</i>	Penelitian analisis sentimen berbasis aspek dimana Pemodelan topik menggunakan LDA menghasilkan 4 aspek yaitu:

				<i>user interface, functionality</i> dan <i>performance, supports, dan updates.</i> Penelitian ini menghasilkan akurasi 87,80% menggunakan algoritma <i>Naïve Bayes</i> dan evaluasi kinerja algoritma menggunakan kurva ROC menghasilkan nilai AUC 0,94
--	--	--	--	---

2.8. Perbedaan Penelitian

Terdapat hal yang membedakan penelitian ini dengan penelitian lain sebelumnya yang sudah pernah dilakukan seperti sejumlah penelitian analisis sentimen berbasis aspek belum menggunakan algoritma *Extreme Gradient Boost*. Selain itu penelitian ini juga menggunakan *word embedding Word2Vec* sedangkan penelitian lain nya kebanyakan menggunakan TF-IDF sebagai pembobot kata. Hal terakhir yang membuat penelitian ini beda dengan penelitian lain adalah aspek nya yang diekstrak sendiri menggunakan algoritma pemodelan topik *Latent Dirichlet Allocation*.

BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1. Dataset

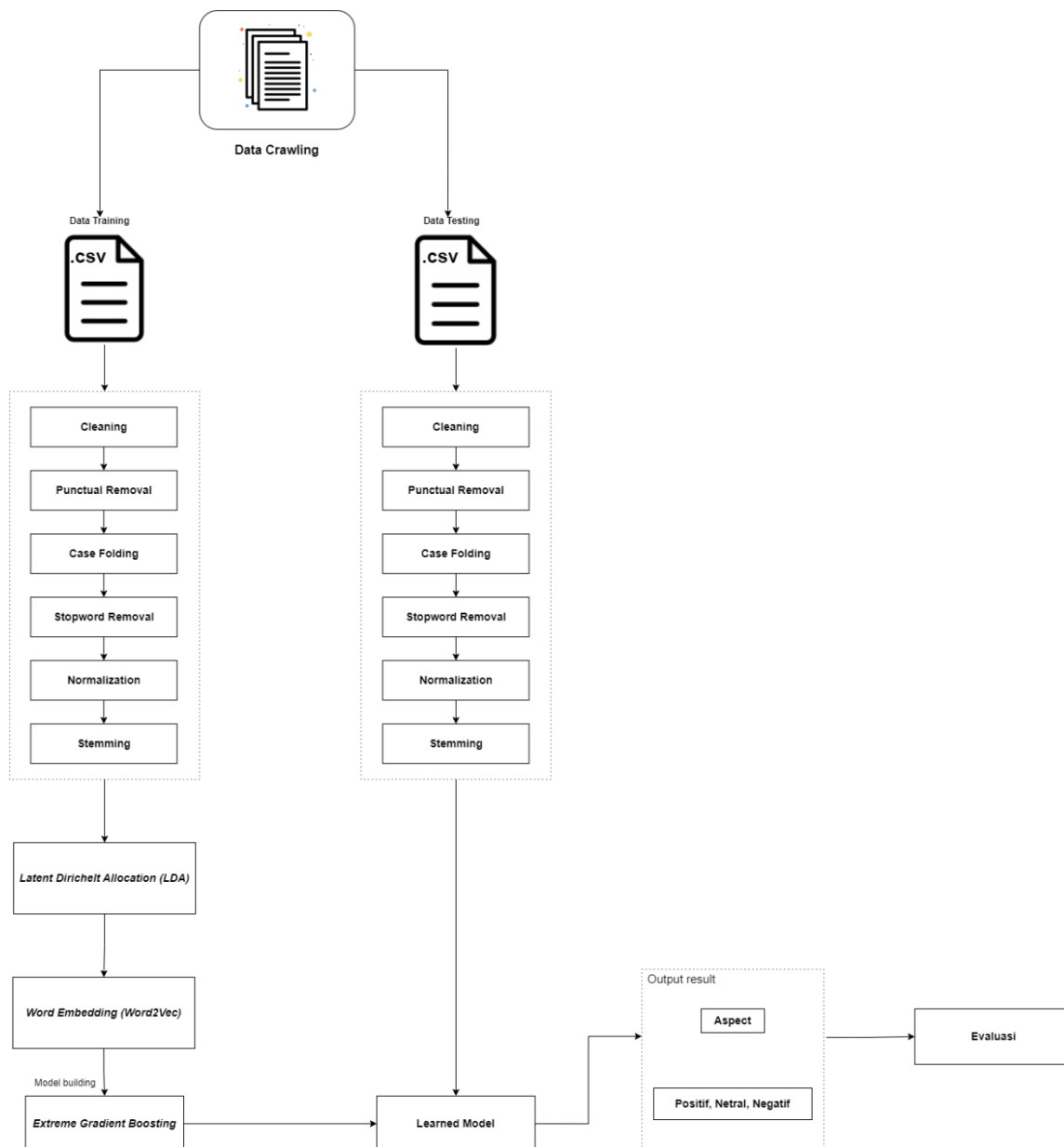
Dataset yang digunakan penelitian ini adalah kalimat ulasan pengguna dalam bahasa Indonesia aplikasi LinkAja melalui *platform Google Play Store*. Proses pengumpulan data dilakukan melalui teknik *scraping* dan data tersebut disimpan dalam *format .csv*. Total data yang berhasil terkumpul mencapai 2000 data. Peneliti memilih 2000 data karena dengan 2000 data sudah bisa mendapatkan akurasi yang bisa dikatakan tinggi yaitu sebesar 90%. Perlu dicatat bahwa urutan waktu tidak menjadi dasar dalam pengumpulan ulasan melalui proses *scraping*. Sebagaimana terlihat pada Tabel 3.1 disajikan beberapa contoh data awal yang berhasil dikumpulkan melalui metode *scraping*.

Tabel 3.1 *Dataset* hasil *scraping* ulasan pengguna LinkAja

No.	Content
1	Sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus Padahal koneksi wifi saya saja sudah 5mbps lebih, pake kuota juga sama saja Tolong diurus secepatnya, karna aplikasi ada buat digunakan dan mempermudah user, bukan ajang buat” aja tp bermasalah
2	Akun udah ber hari2 bahkan dari bulan 9 di coba upgrade sampe skrg ga bisa , syarat dan data diri udah sesuai semua . Mana udah top up lagi , tolong lah respon cepatnyaaaa . Bukan gua aja ini , udah ratusan orang . Bukan nya memudahkan malah mempersulit
3	Aplikasi error coy, dibilang internet terputus padahal saya wifi dan data seluler, ada pakatnya. Padahal cari yang lain biasa biasa aja internet nya, terkecuali ini nih, dari pagi disuruh cek koneksi intenet, tolong perbaiki dong . Terimakasih.

3.2. Arsitektur Umum

Dalam penelitian ini, berbagai tahapan akan dijalankan. Pertama, yang dilakukan adalah pengumpulan data atau *data crawling* untuk menghasilkan *dataset*. Data yang terkumpul berjumlah 2000 data dan akan tersimpan dalam bentuk dokumen dengan *format .csv (comma separated values)*. Selanjutnya akan dilakukan *split data* dengan rasio 70:30 yaitu 70% data *training* dan 30% data *testing*. Kemudian, masing-masing set data tersebut akan melalui tahap *preprocessing*. Tahap ini melibatkan serangkaian proses seperti pembersihan data, penghapusan tanda baca, penyesuaian huruf, penghapusan *stopword*, normalisasi, dan *stemming*. Setelah tahap *preprocessing*, akan dilakukan pemodelan topik dengan data *training* menggunakan LDA untuk mengekstrak aspek-aspek yang dibutuhkan. LDA melibatkan pemodelan dan ekstraksi topik dari dataset. Jika melakukan LDA sebelum membagi dataset, informasi dari data pengujian dapat bocor ke dalam data pelatihan dan sebaliknya, menyebabkan data *leakage* atau kebocoran data. Hal ini dapat memberikan perkiraan yang tidak realistis tentang seberapa baik model dapat menangani data baru. Setelah aspek telah didapat data yang telah di *preprocessing* akan diteruskan ke tahap *word embedding*, di mana kata-kata dalam data akan diubah menjadi representasi vektor menggunakan metode *Word2Vec*. Tahap berikutnya adalah data akan dilatih (*training*) menggunakan metode *Extreme Gradient Boosting (XGBoost)* untuk menghasilkan model yang nantinya digunakan dalam tahap pengujian (*testing*).



Gambar 3.1 Arsitektur Umum

3.2.1. Text Preprocessing

Pada bagian ini, data akan dibersihkan terlebih dahulu dengan beberapa tahap *preprocessing* untuk menghasilkan data yang bagus agar bisa dibaca dan dipelajari oleh mesin dan mendapatkan hasil yang akurat. Berikut adalah penjelasan tahapan *preprocessing* yang dilakukan dalam penelitian ini.

2.2.1.1. *Cleaning*

Data *cleaning* adalah tahap awal yang dilakukan pada *preprocessing*. Pada tahap ini akan dilakukan pembersihan seperti menghilangkan emoji, angka, spasi berlebihan, dan karakter spesial lainnya yang tidak diperlukan.

2.2.1.2. *Punctual Removal*

Punctual removal merupakan proses penghapusan semua tanda baca dari teks, termasuk tanda titik, koma, dan tanda baca lainnya. Tujuan dari langkah ini adalah untuk membersihkan teks dari tanda baca yang sering kali tidak berkontribusi pada analisis teks dan dapat mengganggu proses pemrosesan lebih lanjut. Tahap *Punctual removal* akan ditunjukkan pada tabel 3.2.

Tabel 3.2 Tahapan *Punctual Removal*

Ulasan sebelum proses <i>Punctual Removal</i>	Ulasan setelah proses <i>Punctual Removal</i>
Sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus Padahal koneksi wifi saya saja sudah 5mbps lebih, pake kuota juga sama saja Tolong diurus secepatnya, karna aplikasi ada buat digunakan dan mempermudah user, bukan ajang buat" aja tp bermasalah	Sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus Padahal koneksi wifi saya saja sudah mbps lebih pake kuota juga sama saja Tolong diurus secepatnya karna aplikasi ada buat digunakan dan mempermudah user bukan ajang buat aja tp bermasalah

2.2.1.3. *Case Folding*

Case folding adalah suatu teknik membuat semua karakter huruf pada teks menjadi huruf kecil atau huruf besar, sehingga perbedaan dalam huruf besar dan kecil diabaikan. Tujuan nya adalah untuk menjadikan teks yang memiliki variasi huruf besar dan kecil menjadi lebih konsisten dan seragam. Tahap *Case folding* akan ditunjukkan pada tabel 3.3.

Tabel 3.3 Tahapan *Case Folding*

Ulasan sebelum proses <i>Case Folding</i>	Ulasan setelah proses <i>Case Folding</i>
Sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus Padahal koneksi wifi saya saja sudah mbps lebih pake kuota juga sama saja Tolong diurus secepatnya karna aplikasi ada buat digunakan dan mempermudah user bukan ajang buat aja tp bermasalah	sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus padahal koneksi wifi saya saja sudah mbps lebih pake kuota juga sama saja tolong diurus secepatnya karna aplikasi ada buat digunakan dan mempermudah user bukan ajang buat aja tp bermasalah

2.2.1.4. Stopword Removal

Dalam tahap *Preprocessing*, salah satu proses yang dilakukan adalah *stopword removal* atau penghapusan *stopwords*. *Stopwords* adalah kata-kata yang umumnya tidak punya arti atau kontribusi yang signifikan pada analisis teks. Contoh *stopwords* dalam bahasa Indonesia adalah "yang", "dari", "di", "dan", "untuk", dan sejenisnya. Tahap *Stopword removal* akan ditunjukkan pada tabel 3.4

Tabel 3.4 Tahapan *Stopword Removal*

Ulasan sebelum proses <i>Stopword Removal</i>	Ulasan setelah proses <i>Stopword Removal</i>
sudah daftar sebelumnya tapi saat mau login selalu bilang koneksi bermasalah terus padahal koneksi wifi saya saja sudah mbps lebih pake kuota juga sama saja tolong diurus secepatnya karna aplikasi ada buat digunakan dan mempermudah user bukan ajang buat aja tp bermasalah	daftar login bilang koneksi bermasalah koneksi wifi mbps pake kuota tolong diurus secepatnya aplikasi digunakan mempermudah ajang aja bermasalah

2.2.1.5. Normalization

Dalam tahap *Preprocessing*, *normalization* atau normalisasi adalah proses pengubahan kata-kata dalam teks menjadi bentuk yang lebih standar atau normal. Tujuan dari normalisasi adalah untuk mengurangi variasi bentuk kata yang sebenarnya memiliki makna yang sama. Tahap normalisasi akan ditunjukkan pada tabel 3.5.

Tabel 3.5 Tahapan *Normalization*

Ulasan sebelum proses <i>Normalization</i>	Ulasan setelah proses <i>Normalization</i>
daftar login bilang koneksi bermasalah koneksi wifi mbps pake kuota tolong diurus secepatnya aplikasi digunakan mempermudah ajang aja bermasalah	daftar masuk bilang koneksi bermasalah koneksi internet mbps pakai kuota tolong diurus secepatnya aplikasi digunakan mempermudah ajang aja bermasalah

2.2.1.6. Stemming

Dalam tahap *Preprocessing*, salah satu proses yang sering digunakan adalah *stemming* atau *stemming* kata. *Stemming* adalah proses menghilangkan afiks atau akhiran kata dalam teks dengan tujuan untuk mendapatkan kata dasar atau akar kata. Dengan melakukan *stemming*, variasi bentuk kata yang sebenarnya memiliki akar kata yang sama dapat dipersempit menjadi satu bentuk. Tahap *Stemming* akan ditunjukkan pada tabel 3.6.

Tabel 3.6 Tahapan *Stemming*

Ulasan sebelum proses <i>Stemming</i>	Ulasan setelah proses <i>Stemming</i>
daftar masuk bilang koneksi bermasalah koneksi internet mbps pakai kuota tolong diurus secepatnya aplikasi digunakan mempermudah ajang aja bermasalah	daftar masuk bilang koneksi masalah koneksi internet mbps pakai kuota tolong urus cepat aplikasi mudah guna ajang aja masalah

3.2.2. Latent Dirichlet Allocation (LDA)

Dalam penelitian ini aspek-aspek yang dibutuhkan untuk ABSA akan di ekstrak sendiri menggunakan metode *Latent Dirichlet Allocation*. Metode LDA dipilih karena memiliki keunggulan, yakni LDA mampu merangkum informasi, mengelompokkan topik, mengidentifikasi jumlah optimal topik, dan memberikan label pada setiap topik dalam *dataset*. Hasil dari penerapan LDA digunakan untuk menentukan jumlah aspek terbaik melalui evaluasi nilai koherensi tertinggi, yang selanjutnya dijadikan referensi untuk memberi label otomatis pada data ulasan. Proses ekstraksi aspek menggunakan LDA ada beberapa langkah, sebagai berikut:

1. Data dokumen diubah menjadi token atau kata-kata, kemudian diubah ke dalam bentuk list.
2. Pembentukan *bigram* dilakukan, yaitu mengenali dua kata yang sering muncul bersama dalam dokumen.
3. Kamus kata unik dibuat dan corpus dibangun.
4. Perhitungan nilai koherensi dilakukan pada berbagai jumlah topik, lalu dipilih jumlah topik dengan nilai koherensi tertinggi.
5. Aspek-aspek dari hasil pengelompokan model topik diinterpretasikan melalui pemberian nama yang mencerminkan aspek utama dalam setiap kalimat berdasarkan kata kunci yang muncul.
6. Setiap baris ulasan dianalisis untuk menghitung persentase kontribusi dari masing-masing topik berdasarkan kata kunci. Hasilnya, setiap baris ulasan dapat diberi label aspek berdasarkan persentase kontribusi topik.

Berdasarkan penerapan LDA yang telah dijelaskan maka di dapatlah interpretasi aspek yan ditunjukkan pada tabel 3.7.

Tabel 3.7 Interpretasi Aspek

<i>Topic</i>	<i>Kata Kunci</i>	Interpretasi Aspek
1	0.050*"aplikasi" + 0.043*"tingkat" + 0.033*"gak" + 0.027*"penuh" + 0.026*"servis" + 0.024*"nya" + 0.018*"proses" + 0.015*"bantu" + 0.015*"layanan" + 0.015*"aja"	<i>Customer Service</i>

2	0.049*"aplikasi" + 0.042*"nya" + 0.036*"banget" + 0.032*"mudah" + 0.031*"fitur" + 0.028*"linkaja" + 0.027*"bayar" + 0.021*"beli" + 0.019*"listrik" + 0.017*"aja"	<i>Fitur Aplikasi</i>
3	0.069*"aplikasi" + 0.056*"mudah" + 0.034*"linkaja" + 0.034*"transaksi" + 0.032*"bayar" + 0.028*"banget" + 0.023*"pakai" + 0.022*"aja" + 0.022*"bantu" + 0.021*"ribet"	<i>User Experience</i>
4	0.036*"aplikasi" + 0.034*"gak" + 0.030*"saldo" + 0.024*"masuk" + 0.021*"udah" + 0.020*"akun" + 0.020*"ga" + 0.019*"linkaja" + 0.018*"email" + 0.017*"uang"	<i>Verifikasi</i>

pemodelan topik yang telah dilakukan maka di dapatlah 4 topik yang masing-masing mengandung kata-kata yang memiliki koherensi, Topic 1 memiliki keyword yang menunjukkan keluhan tentang layanan dan bantuan tentang akun yang ingin ditingkatkan (*upgrade*). Kata-kata seperti bantu dan layanan dan servis juga dibahas dalam penelitian (Putra & Yudhoatmojo, 2021) yang dimana kata-kata itu masuk ke dalam kategori aspek *Customer Service*. Keyword ribet, mudah, bantu, banget termasuk ke dalam aspek *User Experience* seperti yang dibalas dalam penelitian Alqaryouti *et al.*, (2018). Berikut adalah *pseudocode* algoritma LDA yang digunakan untuk mengekstraksi kata kunci yang telah dijelaskan sebelumnya.

```

import pandas as pd
import re
import nltk

from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from gensim import corpora, models

df = pd.read_csv("data_train_full_1.csv")
df = df.filter(['content'])

nltk.download('punkt')
nltk.download('stopwords')

stop_words = set(stopwords.words('indonesian'))

stemmer = PorterStemmer()

def preprocess_text(text):

    tokens = word_tokenize(text.lower())

    tokens = [stemmer.stem(token) for token in tokens if
token.isalpha() and token not in stop_words]

    return tokens

df['tokens'] = df['content'].apply(preprocess_text)

dictionary = corpora.Dictionary(df['tokens'])

corpus = [dictionary.doc2bow(tokens) for tokens in df['tokens']]

num_topics = 4

lda_model = models.LdaModel(corpus, num_topics=num_topics,
id2word=dictionary, passes=10)

for topic_id, topic in lda_model.print_topics():

    print(f"Topic {topic_id}: {topic}\n")

```

3.2.3. *Word Embedding*

Dalam penelitian ini, dilakukan identifikasi terhadap pernyataan yang dibuat menggunakan frasa dalam bahasa Indonesia. Oleh karena itu, model *Word2Vec* yang pernah dibuat oleh Mikolov *et al* (2013) tidak bisa dipakai dalam konteks penelitian ini, karena model tersebut dibangun berdasarkan bahasa Inggris (dilatih menggunakan

sekitar 100 miliar kata dari *dataset Google News* dengan dimensi vektor sebanyak 300 fitur). Peneliti juga tidak menggunakan model *Word2Vec* yang menggunakan korpus Wikipedia Bahasa Indonesia yang berisi 459.239 artikel Bahasa Indonesia. Peneliti akan membangun sebuah model *Word2Vec* Bahasa Indonesia dengan menggunakan *dataset* 2000 ulasan aplikasi LinkAja. Peneliti memilih menggunakan *dataset* ulasan karena penelitian ini tentang Analisis Sentimen Berbasis Aspek 2000 kalimat ulasan aplikasi LinkAja di *Google Play Store*. Jadi akan lebih baik jika model *Word2Vec* yang dibangun lebih spesifik. Pada penelitian ini model melatih dengan *default* yaitu CBOW dengan nilai parameter *vector_size* = 100, *window* = 5, *min_count* = 3, *workers* = 4, *epochs* atau *iterasi* = 1000, *sg* = 0, *hs* = 0. Parameter *vector_size* adalah dimensi dari *vector* kata dimana semakin kecil dimensi yang digunakan maka semakin banyak informasi yang dibuang (Juwiantho et al., 2020). Penjelasan proses membangun model *word2vec* akan dijelaskan dalam bentuk *Pseudocode*:

```

#import library yang digunakan untuk word2vec
import os
import pandas as pd
import numpy as np
from tqdm.auto import tqdm
from nltk.tokenize import word_tokenize
from gensim.models import Word2Vec
import nltk
nltk.download('punkt')

#memanggil dataset yang digunakan untuk membangun model
df = pd.read_csv("after_prepro.csv", delimiter=";")
#memodelkan teks dalam dataset menggunakan tools gensim
sentences = [word_tokenize(konten) for konten in tqdm
(df.konten)]

sentences[:5]

#membuat parameter untuk membangun model word2vec
model = Word2Vec(sentences, vector_size=100, window=5,
min_count=3, workers=4, epochs=1000, sg=0, hs=0)

#menyimpan model word2vec
model.save("linkaja_word2vec")

#menload model word2vec
model = Word2Vec.load("linkaja_word2vec")

#lanjut training model word2vec agar domain spesifik
model.train(sentences, total_examples=len(sentences),
epochs=1000)

#simpan lagi model word2vec
model.save("linkaja_word2vec")

#memberikan variabel agar model lebih mudah dipanggil
w2v = model.wv

#contoh convert kata ke dalam vector menggunakan word2vec
w2v["aplikasi"]

```


Akan ditampilkan nilai *vector* dari kata ‘aplikasi’ pada gambar 3.2.

```
array([ 0.06963123, -1.0128715, -0.05725843, 0.9880763, 0.27406493,
       -0.7535094, -1.5920572, -0.3120199, 1.0676502, 1.2582871,
        1.4552094, 1.380905, -1.2919332, 1.7452681, 0.8463683,
       -1.5371537, 0.13249953, 1.5427938, -1.844609, -0.12308154,
        1.6268371, -1.8486102, -0.9970391, 1.8952967, 0.9411799,
        1.2676388, -0.5336448, 1.1506414, -0.4270301, 0.362411,
       -1.1926585, 0.33535495, 0.10978813, -2.224516, 0.25711256,
       -2.7267148, 0.35026845, 0.5454953, -1.6977862, 1.4067333,
        0.3076887, 1.1609141, -1.3795674, 1.4486182, 0.6878506,
       -0.8688687, 0.65074074, -1.0573096, 1.3579785, 0.36147276,
       -2.5275247, 0.8917867, -2.1473346, 0.90943396, -0.5141301,
       -1.590803, -0.12381034, -0.0904642, 0.1557771, 0.15814562,
       -1.2835094, 0.47151056, -0.5008601, -1.3213483, -0.26052776,
       -0.33954027, 0.03921036, -0.02035406, 0.0983533, -2.9999979,
       -0.2913604, -0.4546682, -1.2181493, -1.2301683, -0.5058741,
        0.6336569, -0.29575405, 0.07473517, 0.6860664, -1.4802592,
       -0.7028281, 0.19198997, 1.5541604, 0.26105127, -1.4506646,
       -0.41518086, -0.10935909, -0.43736005, -1.3038696, 0.9145455,
        0.22481303, 0.10609993, -0.47850597, 0.23078243, -1.6192131,
       -1.433882, -0.40061197, 0.5148982, -0.0258891, 0.02200621],
      dtype=float32)
```

Gambar 3.2 Representasi *Vector* kata ‘aplikasi’

3.2.4. *Extreme Gradient Boosting (XGBoost)*

Pada penelitian ini *hyperparameter* yang digunakan adalah *learning_rate* = 0,75, *n_estimators* = 100. Parameter *learning_rate* (atau sering disebut juga sebagai *eta*) adalah salah satu parameter penting pada model XGBoost yang mengontrol kontribusi dari setiap pohon (*tree*) terhadap model. Ini adalah faktor pembelajaran yang menentukan seberapa besar model ingin "membelajar" dari kesalahan model pada iterasi sebelumnya. Parameter *n_estimators* adalah parameter pada model XGBoost yang menentukan jumlah total pohon (*tree*) yang akan dibangun dalam *ensemble*. Setiap pohon menambahkan kompleksitas ke model dan berkontribusi pada akurasi prediksi. Oleh karena itu, *n_estimators* mengontrol seberapa "kuat" atau "besar" *ensemble* model XGBoost. Sebagai parameter utama, *n_estimators* sering di-tune selama proses *fine-tuning* model. Jika nilai *n_estimators* terlalu kecil, model mungkin belum sempurna "belajar" dari data, sementara nilai yang terlalu besar dapat menyebabkan *overfitting*. Setelah ulasan diubah menjadi *vector* maka data akan siap untuk diklasifikasi menggunakan model, dengan menggunakan dataset [X,Y] terlebih dahulu akan dicari *mean* dari nilai target (Y) untuk mendapatkan nilai prediksi awal $h_0(x)$ dan nilai *residual* \hat{Y} sesuai titik awal. Kemudian dilakukan pelatihan data dengan model pertama yaitu *decision tree* yang dilatih dengan variabel independen dan

residual $[X, \hat{Y}]$ untuk mendapatkan nilai prediksi model pertama. Dalam algoritma *Extreme Gradient Boosting*, menetapkan jumlah pohon dan *depth* merupakan faktor krusial. Upaya mengatasi kesulitan dalam menemukan konfigurasi optimal algoritma dapat diubah dengan mencari solusi klasifikasi yang baru, dengan tujuan mengurangi *loss function*. Pendekatan ini melibatkan penyesuaian fungsi kerugian, seperti yang dijelaskan dalam persamaan berikut ini:

$$obj^{(t)} = \sum_{i=1}^t l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^t \Omega(f_i) \quad (3,1)$$

Dimana:

- $\hat{y}_i^{(t)}$ = Nilai prediksi
- y_i = Nilai *actual*
- $l(\hat{y}_i^{(t)}, y_i)$ = *loss function*
- $\Omega(f_i)$ = istilah regularisasi

Karena model pohon *ensemble* berperan sebagai parameter dan tidak dapat dioptimalkan melalui metode pengoptimalan konvensional dalam ruang *Euclidean*, maka dilakukan substitusi dengan model yang dilatih melalui pendekatan aditif. Pendekatan ini memanfaatkan $\hat{y}_i^{(t)}$ pada prediksi ke-*i* dan iterasi ke-*t* (Chen & Guestrin, 2016).

Berikut adalah contoh proses penyusunan algoritma *Extreme Gradient Boosting* (Dayananda, 2020):

Pada tabel 3.8 ditunjukkan tabel dengan menggunakan dataset $[X, Y]$.

Tabel 3.8 Dataset untuk membangun pohon XGBoost

X	Y
2	0
8	1
12	1
18	0

Parameter yang dipakai adalah $n_estimators = 2$, $learning_rate = 1$, $min_child_weight = 0$, $reg_lambda = 0$, $base_score = 0,5$.

1. Prediksi Awal

Prediksi awal atau $base_score$ diberi nilai sebesar 0,5 untuk seluruh $node$ dalam $dataset$:

$$f_0(x) = h_0(x) = 0,5 \quad (3,2)$$

2. Kalkulasi *error* atau *residuals*

Hitung *residuals* \hat{Y} untuk semua $node$ dari prediksi sebelumnya:

$$\hat{Y} = y - f_0(x) \quad (3,3)$$

Tabel 3.9 Kalkulasi *error* atau *residuals*

X	Y	$f_0(x)$	$\hat{Y} = y - f_0(x)$
2	0	0.5	-0.5
8	1	0.5	0.5
12	1	0.5	0.5
18	0	0.5	-0.5

3. Model latih

train model pertama (proses *train* model mengacu ke pembangunan pohon) yang dinotasikan sebagai M1, dengan menerapkan data $[X, \hat{Y}]$ dan menerapkan algoritma *tree* khusus XGBoost yang berbeda dalam strukturnya jika dibandingkan dengan *decision tree*. Sebelum membangun pohon XGBoost, ada beberapa istilah yang perlu didefinisikan untuk menurunkan rumus yang mengatasi permasalahan optimasi dalam konteks XGBoost, yang diuraikan sebagai berikut.

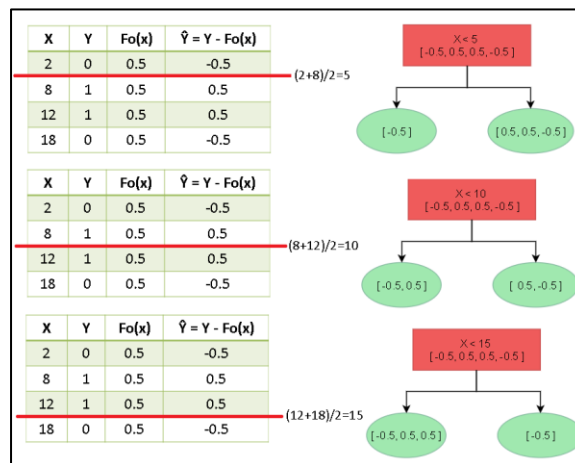
$$Gain = (Left_{similarity} + Right_{similarity}) - Root_{similarity} \quad (3,4)$$

$$Similarity\ Score = \frac{(\sum \hat{y}_i)^2}{\sum [Previous\ f_i(x) \cdot (1 - Previous\ f_i(x)) + \lambda]} \quad (3,5)$$

$$Output\ Value = \frac{(\sum \hat{y}_i)}{\sum [Previous\ f_i(x) \cdot (1 - Previous\ f_i(x)) + \lambda]} \quad (3,6)$$

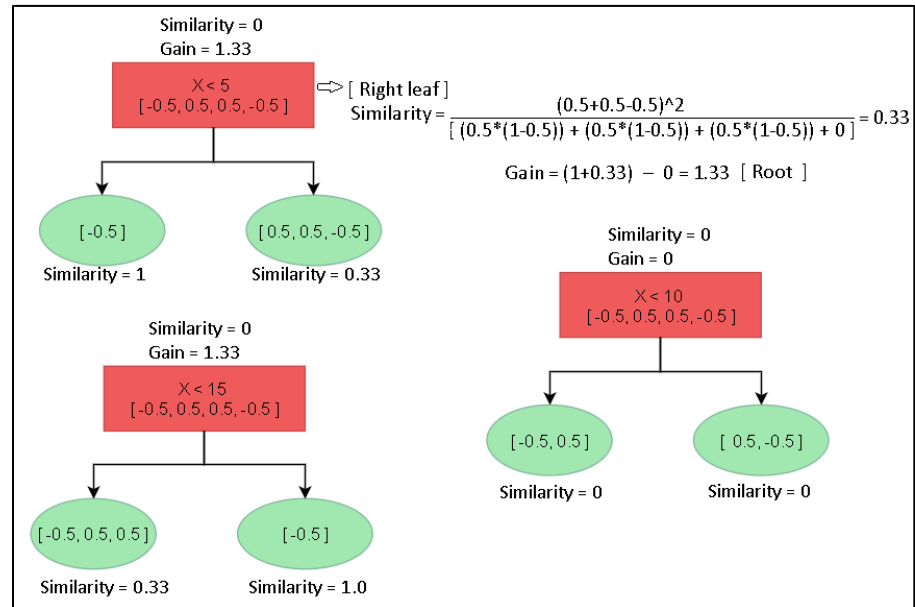
Dimana:

- \hat{Y}_i = Residual ke-i
 - λ = *reg_lambda*
 - $Previousf_i(x)$ =
 - Kalkulasi *gain* hanya untuk menghitung akar *tree*
 - Kalkulasi *similarity* berlaku untuk semua *node*
 - Kalkulasi *output value* hanya untuk *leaf node*
 - *Lambda* adalah indikator, jika angka *lambda* meningkat maka akan membuat *pruning* lebih banyak *node* pada *tree* yang dibangun.
- i. Pohon dibangun lewat cara membagi data menjadi dua bagian dari berbagai kemungkinan pemisahan atau *split*.



Gambar 3.3 Membangun pohon XGBoost

- Penetapan batasan pada *root* dihitung dengan menarik nilai tengah antara dua titik percabangan atau *split*, sementara bagian lain mengarah pada setiap *leaf node*. Apabila *dataset* terdiri dari n data, maka jumlah pohon yang dapat dibangun adalah sebanyak $n-1$.
- ii. Hitung nilai *similarity* dan *gain* pada seluruh pohon untuk menemukan pohon dengan *split* yang optimal.

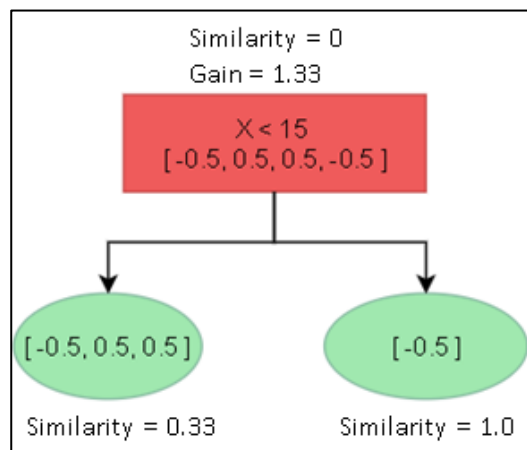


Gambar 3.4 Perhitungan *similarity* dan *gain*

Pilih nilai dari suatu pohon yang memiliki nilai *gain* optimal,

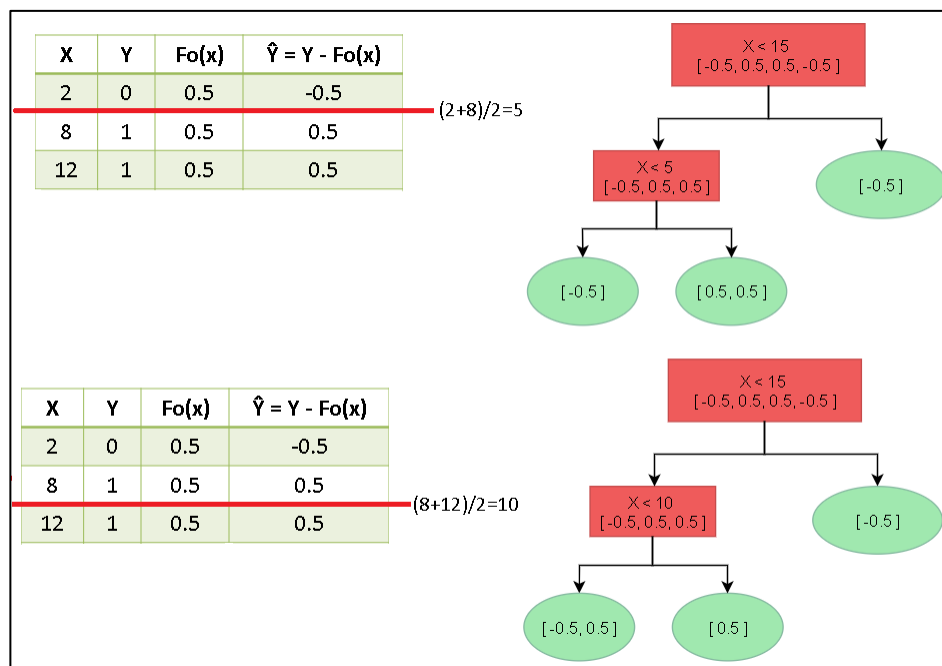
Pada 3.4 pohon dengan $x < 15$ memiliki *gain max* sebesar 1.33.

- iii. Buatlah *split* lagi untuk pohon yang mempunyai nilai *gain* maksimum hingga *max_depth* untuk pembangunan pohon penuh.



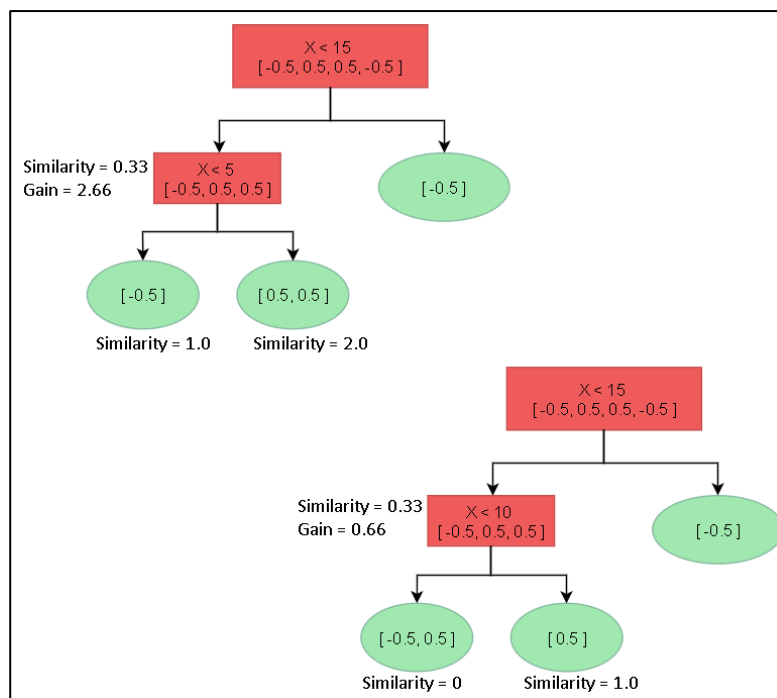
Gambar 3.5 Split pada Pohon XGBoost

sebab nilai *max_depth* yang dipakai berjumlah dua, buat pohon lagi dengan memisahkan data yang terdapat pada *leaf* sebelah kiri.



Gambar 3.6 Split pada Turunan Percabangan

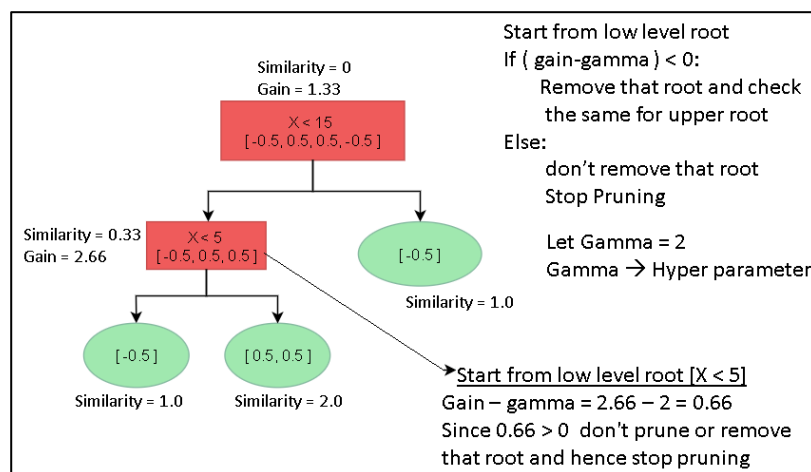
Hitunglah lagi nilai *similarity* dan *gain* untuk *split* baru pada cabang lanjutan untuk memilih *internal root* yang mempunyai nilai *gain* yang *max*.



Gambar 3.7 Cara Kalkulasi *Similarity* dan *Gain* pada *Split* Lanjutan

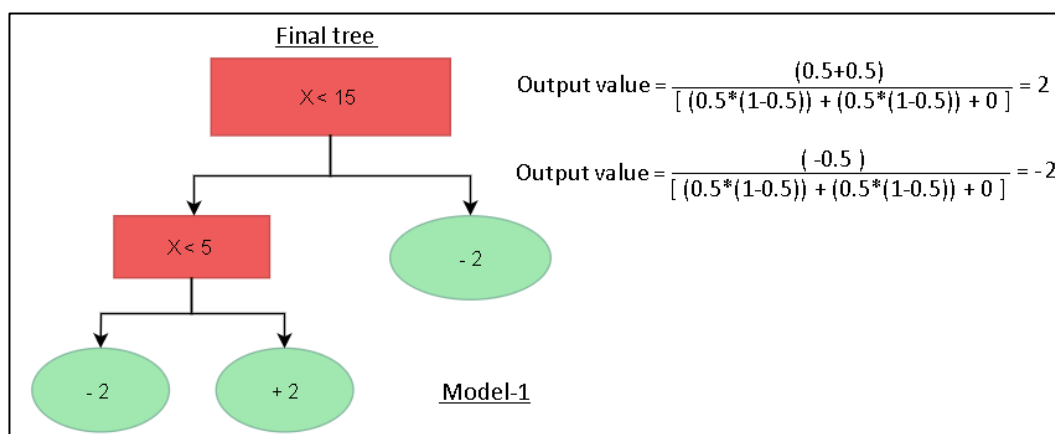
akan dipilih *internal root* $X < 5$ untuk pembangunan pohon pertama seperti gambar 3.7 karena *internal root* itu memiliki nilai *gain* maksimum sebesar 2.66.

- iv. Ketika pohon sudah terbentuk, maka akan dilakukan *tree pruning* yang berfungsi untuk memperkecil ukuran *decision trees* dengan menghapuskan bagian pohon yang berkekuatan kecil untuk mengklasifikasikan kejadian.



Gambar 3.8 Tahap *Tree Pruning*

- v. Menghitung *output value* untuk semua *leaf* untuk mendapat *tree* terakhir pada akhir model 1 karena beberapa *leaf* memiliki *residuals* lebih dari satu.



Gambar 3.9 Perhitungan *Output Value*

- vi. Mendapatkan prediksi dari model 1 dengan cara lewatkan semua titik data melewati pohon terakhir (model 1) untuk mendapatkan nilai $h_1(x)$ dan hitung prediksi $f_1(x)$ dan nilai *residuals*.

Berikan nilai $learning_rate = 1.0$

Maka:

$$f_1(x) = \sigma \left[\left(\frac{h_0(x)}{1-h_0(x)} \right) + (\eta \times h_1(x)) \right] \quad (3,7)$$

Pecahkan $f_1(x)$ pada klasifikasi untuk mendapatkan:

$$f_1(x) = \sigma(0 + 1x(h_1(x))) \quad (3,8)$$

Fungsi sigmoid:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \quad (3,9)$$

Tabel 3.10 Kalkulasi Nilai Prediksi pada Model 1

X	Y	$h_1(x)$	$f_1(x) = \sigma(0 + 1x(h_1(x)))$	$\hat{Y} = y - f_1(x)$
2	0	-2	-0.5	-0.11
8	1	2	0.5	0.12
12	1	2	0.5	0.12
18	0	-2	-0.5	-0.11

4. Ulangi langkah iii untuk membangun model *XGBoost* lainnya.

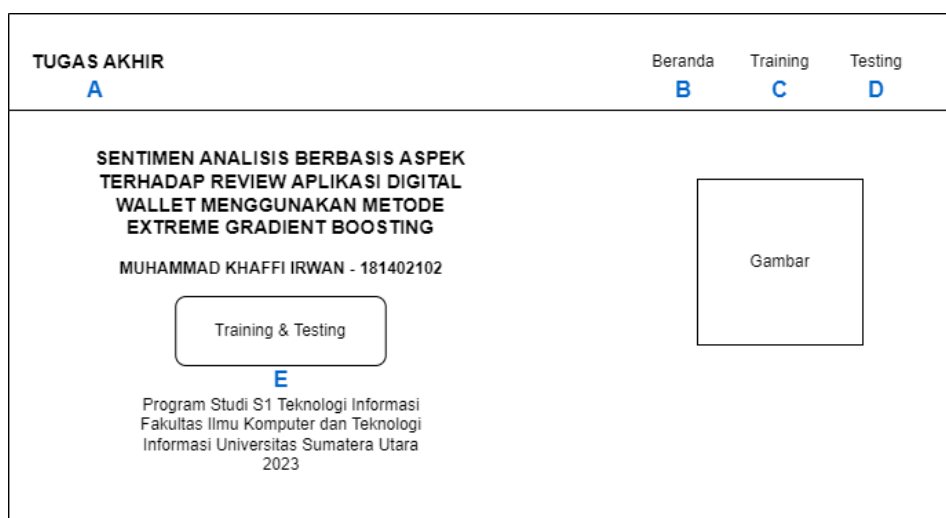
Salah satu hal yang memiliki peran utama dibalik keberhasilan *XGBoost* adalah kemampuan yang lebih efisien dan dapat diaplikasikan secara luas dalam berbagai skenario. Hal ini dikarenakan kemampuannya dalam menangani beragam tugas seperti regresi, klasifikasi, dan juga *ranking*. "scalability" dari algoritma ini merupakan hasil dari peningkatan dalam proses optimasi pada algoritma sebelumnya. Proses optimasi dalam Algoritma *XGBoost* diselesaikan dalam waktu yang 10 kali lebih cepat dibandingkan dengan implementasi metode lain. Keberhasilan ini terbukti melalui popularitas metode *XGBoost* yang mendominasi berbagai kompetisi dalam bidang *machine learning*. (Chen & Guestrin, 2016).

3.3. Perancangan Sistem

Bagian ini akan dijelaskan desain antarmuka untuk sistem “Sentimen Analisis Berbasis Aspek Terhadap *Review Aplikasi Digital Wallet Menggunakan Metode Extreme Gradient Boosting*”. Antarmuka sistem akan didesain secara *single page website*. Desain antarmuka sistem ini memiliki tujuan untuk memberikan pengguna gambaran tentang fungsionalitas sistem yang akan dikembangkan serta untuk menyederhanakan penggunaan sistem bagi pengguna.

3.3.1. Desain Halaman Beranda

Beranda merupakan tampilan pertama atau tampilan utama yang muncul saat sistem pertama kali dibuka oleh pengguna. Desain tampilan beranda akan ditampilkan pada Gambar 3.10.



Gambar 3.10 Desain Halaman Beranda

Keterangan:

1. Tombol (A) berguna untuk menavigasi ke halaman utama.
2. Tombol (B) berguna untuk menavigasi ke halaman utama.
3. Tombol (C) berguna untuk menavigasi ke halaman *training*.
4. Tombol (D) berguna untuk menavigasi ke halaman *testing*.

5. Tombol (E) berguna untuk menavigasi pengguna untuk memulai proses *training* dan *testing*.

3.3.2. Desain Halaman *Training*

Halaman *training* data adalah halaman yang di desain untuk melakukan pelatihan data. Pengguna dapat mengunggah *file dataset* dalam format .csv yang telah disiapkan sebelumnya. Setelah file diunggah, halaman ini juga memaparkan tombol yang berfungsi untuk menjalankan proses *training*. Pelaksanaan *training* akan diinisiasi dan data akan diolah. Hasil akhir dari proses *training* akan muncul dalam format file .sav sebagai model. Pada halaman ini, hasil data sebelum dan sesudah melalui tahap *preprocessing* akan terlihat, sehingga perbaikan dan struktur yang lebih baik dalam kalimat-kalimat data dapat terlihat dengan jelas. Rancangan tampilan dari halaman data *Training* akan ditampilkan pada Gambar 3.11.

The wireframe shows a web page layout for the 'Training' section. At the top, there is a header bar with four navigation links: 'TUGAS AKHIR A', 'Beranda B', 'Training C', and 'Testing D'. Below the header, the main content area is titled 'Training' and has a subtitle 'Upload Dataset untuk proses preprocessing dan training'. In the center, there are two buttons: 'Masukkan Dataset' (labeled E) and 'Mulai Training' (labeled F). Below these buttons, there is a table labeled 'Hasil Preprocessing' with three columns and three rows.

TUGAS AKHIR A				
		Beranda B	Training C	Testing D
Training				
Upload Dataset untuk proses preprocessing dan training				
Masukkan Dataset E		Mulai Training F		
Hasil Preprocessing				

Gambar 3.11 Desain Halaman *Training*

Keterangan:

1. Tombol (A) berfungsi untuk menavigasi ke halaman utama.
2. Tombol (B) berfungsi untuk menavigasi ke halaman utama.
3. Tombol (C) berfungsi untuk menavigasi ke halaman *training*.
4. Tombol (D) berfungsi untuk menavigasi ke halaman *testing*.

5. Tombol (E) berfungsi untuk memasukkan dataset data *training*.
6. Tombol (F) berfungsi untuk memulai proses *training*.

3.3.3. Desain Halaman Data *Testing*

Halaman Data *Testing* adalah halaman yang didesain untuk melakukan proses *testing* data. Pengguna dapat mengunggah file dataset dalam *format* .csv yang telah dipersiapkan sebelumnya. Setelah file diunggah, halaman ini juga memaparkan tombol yang berfungsi untuk memulai proses data *testing*. Proses *testing* akan diproses dan akan menghasilkan tabel yang berisi ulasan sebelum dan sesudah melewati tahap *preprocessing*, hasil prediksi, dan hasil kinerja algoritma *XGBoost* yang ditampilkan dalam bentuk *confusion matrix* yang menampilkan hasil *Precision*, *Recall*, *F1-Score*, dan *Accuracy*. Desain tampilan dari halaman data *Testing* akan ditampilkan pada Gambar 3.12.

TUGAS AKHIR		Beranda	Training	Testing									
A		B	C	D									
Testing Upload Dataset untuk proses Testing <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 10px; text-align: center;">Masukkan Dataset</div> <div style="border: 1px solid black; padding: 10px; text-align: center;">Mulai Testing</div> </div> <div style="display: flex; justify-content: center; align-items: center; margin-top: 10px;"> <div style="margin-right: 20px;">Hasil Prediksi</div> <div style="margin-right: 20px;">E</div> <div>F</div> </div> <table border="1" style="margin: 20px auto; width: 60%;"> <thead> <tr> <th>Content</th> <th>Preprocess</th> <th>Prediksi</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>					Content	Preprocess	Prediksi						
Content	Preprocess	Prediksi											

Gambar 3.12 Desain Halaman Data *Testing*

Keterangan:

1. Tombol (A) berfungsi untuk menavigasi ke halaman utama.
2. Tombol (B) berfungsi untuk menavigasi ke halaman utama.
3. Tombol (C) berfungsi untuk menavigasi ke halaman *training*.
4. Tombol (D) berfungsi untuk menavigasi ke halaman *testing*.

5. Tombol (E) berfungsi untuk memasukkan dataset data *testing*.
6. Tombol (F) berfungsi untuk memulai proses *testing*.

3.3.4. Desain Halaman Uji Coba *Real Time*

Halaman uji coba *real time* adalah halaman yang didesain sebagai tempat dimana pengguna dapat melakukan analisis terhadap data tunggal. Pada halaman ini, pengguna akan memberikan *input* dalam bentuk ulasan berbahasa Indonesia mengenai aplikasi LinkAja. *Output* yang dihasilkan adalah prediksi sentimen dan identifikasi aspek yang terkandung dalam ulasan yang disampaikan oleh pengguna. Desain tampilan dari halaman uji coba real time akan ditampilkan pada Gambar 3.13.

TUGAS AKHIR

Beranda Training Testing

Data Testing - Uji Coba Realtime

Masukkan Text

Masukkan Kalimat Mulai Testing

Hasil Prediksi Real Time

Input	Aspek
	Sentiment

Gambar 3.13 Desain Halaman Uji Coba Real Time

Keterangan:

1. Tombol (A) berfungsi untuk menavigasi ke halaman utama.
2. Tombol (B) berfungsi untuk menavigasi ke halaman utama.
3. Tombol (C) berfungsi untuk menavigasi ke halaman *training*.
4. Tombol (D) berfungsi untuk menavigasi ke halaman *testing*.
5. Tombol (E) berfungsi untuk menginput kalimat yang ingin diidentifikasi.
6. Tombol (F) berfungsi untuk memulai proses identifikasi.

3.4. Tahap Evaluasi

Proses evaluasi perlu dilakukan guna memperoleh pemahaman tentang akurasi dan keberhasilan sistem yang telah dikembangkan dalam mengidentifikasi pernyataan. Metode evaluasi yang dipakai di penelitian ini adalah metode *Confusion Matrix*. Untuk melakukan evaluasi terhadap sistem yang telah dibangun, akan dibuat perhitungan nilai *Precision*, *Recall*, *F1-Score*, dan *Accuracy*. *Precision* merupakan parameter yang mengukur jumlah prediksi positif yang akurat, dihitung sebagai perbandingan antara jumlah sampel positif yang diprediksi secara benar dengan total jumlah sampel positif yang diprediksi. *Recall* mengukur jumlah prediksi positif yang tepat yang diperoleh dari seluruh prediksi positif yang dilakukan, dihitung sebagai perbandingan antara observasi positif yang diprediksi dengan benar terhadap semua observasi dalam kelas yang sesungguhnya. *F1-Score* merupakan rata-rata terbobot dari *Precision* dan *Recall*. Oleh karena itu, skor ini mempertimbangkan *False Positive* (FP) dan *False Negative* (FN). *Accuracy* adalah metrik kinerja yang paling intuitif dan dihitung sebagai perbandingan antara observasi yang diprediksi secara benar dengan total observasi. Implementasi metode evaluasi *Confusion Matrix* dapat dilihat pada Tabel 3.11.

Tabel 3.11 *Confusion Matrix*

	<i>Actual Positive</i> (1)	<i>Actual Negative</i> (0)
Prediksi Positif (1)	TP (<i>True Positive</i>)	FP (<i>False Positive</i>)
Prediksi Negatif (0)	TN (<i>True Negative</i>)	FN (<i>False Negative</i>)

Untuk menghitung nilai *Precision*, *Recall*, *F1-Score*, dan *Accuracy* akan digunakan persamaan seperti di bawah ini:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3,10)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3,11)$$

$$F1 - Score = 2 \times \frac{precision \times recall}{precision+recall} \times 100\% \quad (3,12)$$

$$Accuracy = \frac{TP+TN}{Total\ Data} \times 100\% \quad (3,13)$$

Keterangan:

True Positive (TP) = Jumlah label *actual* positif yang terprediksi positif

False Positive (FP) = Jumlah label *actual* negatif yang terprediksi positif

True Negative (TN) = Jumlah label *actual* negatif yang terprediksi negatif

False Negative (FN) = Jumlah label *actual* positif yang terprediksi negatif

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Implementasi Sistem

Pada tahap pengembangan sistem sentimen analisis berbasis aspek terhadap *review* aplikasi *digital wallet* menggunakan metode *Extreme Gradient Boosting*, diperlukan sejumlah perangkat keras dan perangkat lunak untuk mendukung penelitian, yaitu:

4.1.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Perangkat keras yang digunakan dalam pengembangan sistem ini memiliki spesifikasi sebagai berikut:

1. Laptop Asus A456U
2. *Processor Intel Core i7-7500U CPU @ 2.70GHz - 2.90 GHz*
3. RAM sebesar 8GB

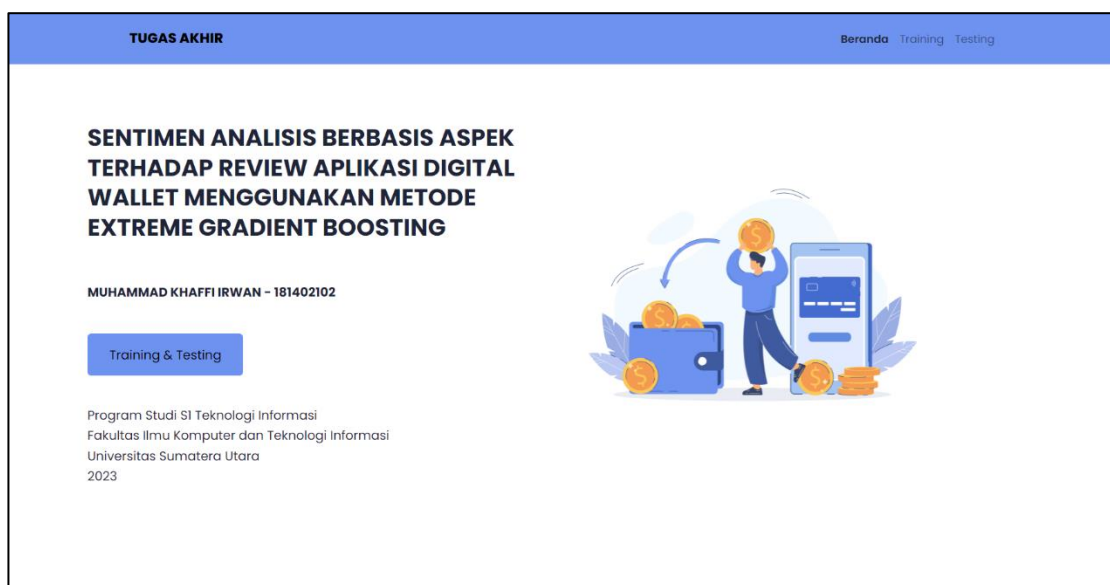
Perangkat lunak yang digunakan memiliki spesifikasi sebagai berikut:

1. *Windows 10 PRO 64 bit*
2. Python 3.8.6 dengan library *flask* versi 2.1.2, *matplotlib* versi 3.5.2, *nlTK* versi 3.7, *numpy* versi 1.22.4, *pandas* versi 1.4.2, *sastrawi* versi 1.0.1, *sckit-learn* versi 1.2.2, *seaborn* versi 0.11.2, dan *xgboost* versi 0.90.
3. Microsoft Visual Studio Code

4.1.2. Implementasi Perancangan Tampilan Antarmuka

1. Halaman Beranda

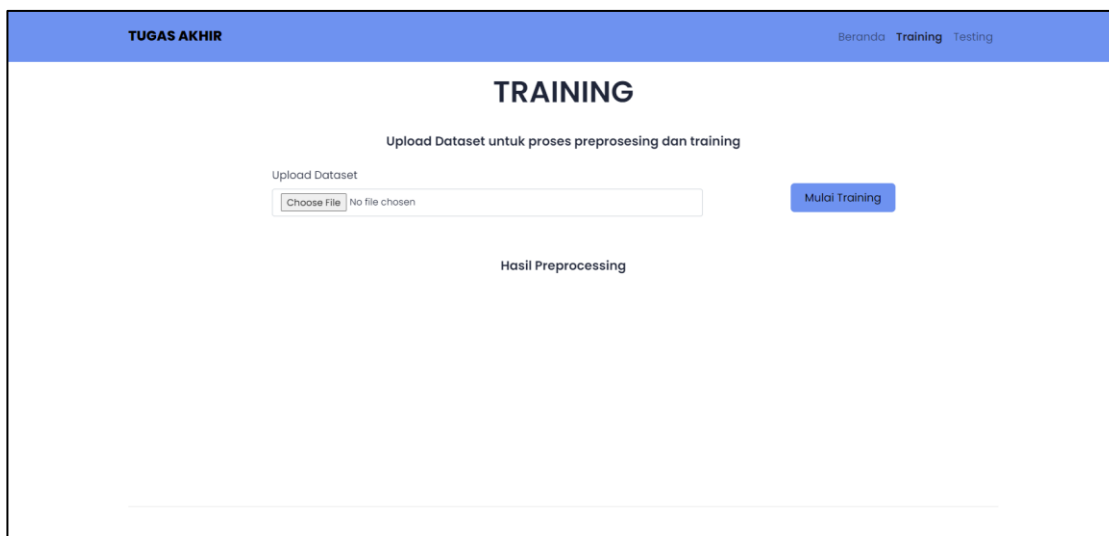
Beranda adalah menu awal yang terlihat ketika sistem dijalankan yang berisi informasi mengenai judul penelitian dan informasi mengenai penulis. Halaman ini juga berisi tombol-tombol *header* yang berguna untuk menavigasi dari satu halaman ke halaman lain. Tampilan halaman Beranda akan ditunjukkan pada gambar 4.1.



Gambar 4.1 Halaman Beranda

2. Tampilan Halaman *Training*

Halaman *Training* adalah halaman yang dipakai untuk mengerjakan proses *training* data. Pada halaman ini akan disediakan tombol dan kolom untuk meng *upload* dataset data *training* yang berformat csv. Jika sudah mengupload dataset nya pengguna bisa mengklik tombol mulai *training* guna memulai proses *training* data. Tampilan halaman *training* akan ditunjukkan pada gambar 4.2.



Gambar 4.2 Halaman *Training*

Jika proses *training* telah selesai, pada halaman ini akan muncul hasil *training* data dalam bentuk tabel. Tabel tersebut akan berisi ulasan sebelum dan sesudah tahap *preprocessing*, aspek dan label manual. Tampilan halaman hasil *training* data akan ditunjukkan pada gambar 4.3.

TUGAS AKHIR

BerandaTrainingTesting

Hasil Preprocessing

Show 10 entries

Search:

content	Preprocess	CustomerService	FiturAplikasi	UserExperience	Verifikasi
Dari semua aplikasi, ini yang paling lama cek verifikasi ktp. Padahal udah sangat butuh. Tolong di percepat verifnya	aplikasi lama cek verifikasi ktp udah butuh tolong cepet verifnya	0	0	-1	-1
Uang yg udah masuk aplikasi ini ga bisa di ambil lagi. Harus verifikasi dulu dan di setuju ny lama banget kecewa.	uang udah masuk aplikasi ga ambil verifikasi tuju lama kecewa	0	-1	-1	-1
Minta tolong ini mau transfer uang tapi disuruh upgrade pake foto ktp dan foto wajah foto wajahnya udah pas malah prosesnya lama bgt nggak jadi jadi lagi. Mohon bantuannya	tolong transfer uang suruh tingkat pakai foto ktp foto wajah foto wajah udah pas proses lama mohon bantu	0	0	-1	-1
Sudah cape rasanya, leher sampai pegal, mau upgrade ke premium sangat susah sekali.	cape leher pegal tingkat premium susah	0	0	-1	-1
gimana ini, mau di upgrate ke premium tp proses trus suruh tunggu 24 jam. padahal udah tunggu lebih dri waktunya tetep di proses trus. tolong dong di prbalikin lagi. mau buka bantuan juga gabisa padahal kuota full. trpkas ksh bintang 2. Tar kalo udh bisa di upgrate ke	gimana tingkat premium proses suruh tunggu jam udah tunggu proses trus tolong perbaikiin buka bantu gabisa kuota full trpkas kasih bintang entar udah tingkat premium ganti bintang	-1	0	-1	-1

Gambar 4.3 Halaman hasil *Training* Data

3. Tampilan Halaman *Testing*

Halaman *Testing* adalah halaman dimana proses *testing* data berlangsung. Pada bagian ini akan disediakan tombol dan kolom untuk mengupload dataset data *testing* yang

berformat csv. Jika sudah mengupload *dataset* nya pengguna bisa mengklik tombol mulai testing untuk memulai proses *testing* data. Tampilan halaman *testing* akan ditunjukkan pada gambar 4.4.

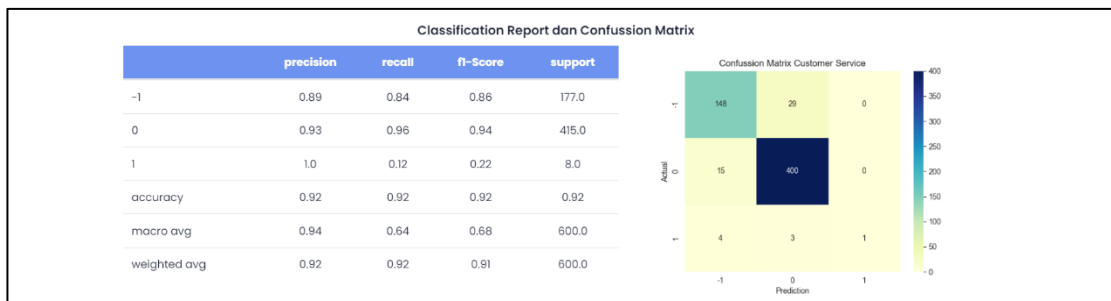
Gambar 4.4 Halaman *Testing*

Jika proses *testing* telah selesai, pada halaman ini akan muncul hasil *training* data dalam bentuk tabel. Tabel tersebut akan berisi ulasan sebelum dan sesudah tahap *preprocessing*, aspek dan label prediksi. Tampilan halaman hasil *testing* data akan ditunjukkan pada gambar 4.5.

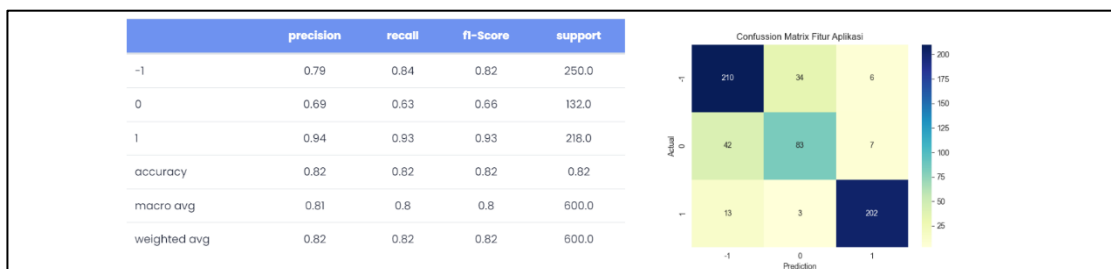
content		Preprocess	Prediksi			
			CustomerService	FiturAplikasi	UserExperience	Verifikasi
Saya upgrade full service kok sudah 5 hari belum di verifikasi ya? Padahal katanya 1x24 jam sudah disetujui. Jadinya saya gabisa transaksi. Udah ngubungin call center, live chat, email bantuan linkaja juga gak ada jawaban tuh. Tolong dong solusinya, di akun saya sudah terisi sejumlah nominal. Masa gabisa di apa2in. Di tarik gabisa, di buat transaksi gabisa. Tolong dong perbaiki	tingkat penuh servis verifikasi jam tuju gabisa transaksi udah hubung panggil pusat langsung obrol email bantu linkaja gak tolong solusi akun isi nominal gabisa in tarik gabisa transaksi gabisa tolong baik		-1	-1	-1	-1
Sampah pelayanan nya, masa ini udh hampir 1 bulan upgrade akun belum jadi jadi. Nanya di live chat juga cuma template doang, ngrim email kaga ada balesan sama sekali. Pada buta apa ya staff nya, orang butuh cepet. Sekalun aja satu tahun di lama lamain, maaf saya bicara kasar, SAMPAH STAF PELAYANANNYA.	sampah layan udah tingkat akun nanya langsung obrol templat ngrim email kaga balesan buta staf orang butuh cepet aja lama lamain maaf bicara kasar sampah staf layan		-1	0	-1	-1
Aplikasinya kurang bagus mau	aplikasi kurang bagus tingkat		0	0	-1	-1

Gambar 4.5 Halaman Hasil *Testing* Data

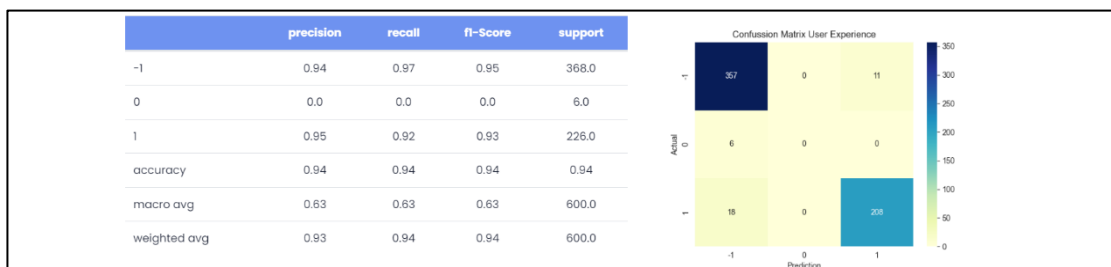
Halaman hasil *testing* data juga menampilkan hasil evaluasi setiap aspek dalam bentuk *confusion matrix* dan *classification report*. Hasil evaluasi setiap aspek masing-masing akan ditunjukkan pada gambar 4.6, gambar 4.7, gambar 4.8, gambar 4.9.



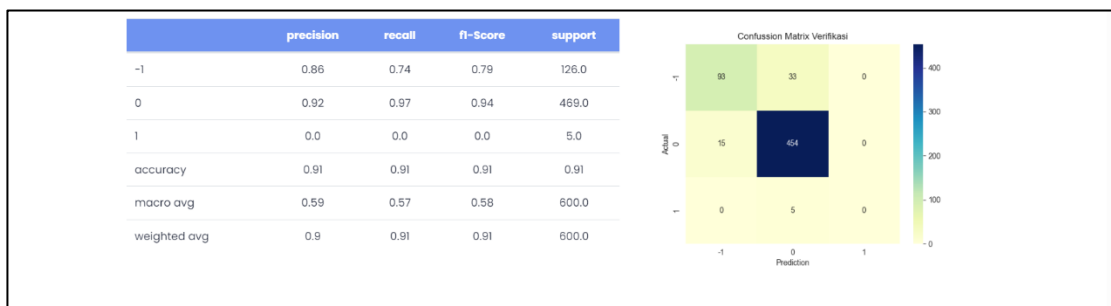
Gambar 4.6 Evaluasi Aspek Customer Service



Gambar 4.7 Evaluasi Aspek Fitur Aplikasi



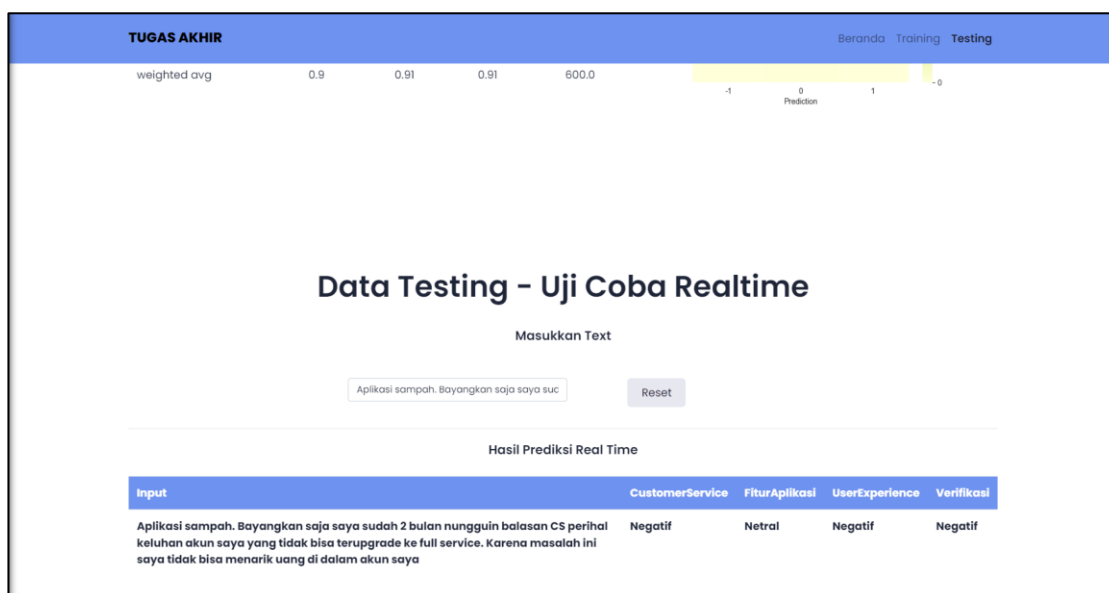
Gambar 4.8 Evaluasi Aspek User Experience



Gambar 4.9 Evaluasi Aspek Verifikasi

4. Tampilan Halaman Uji Coba *Real Time*

Halaman uji coba *real time* adalah halaman yang digunakan pengguna untuk menganalisa kalimat ulasan yang di *input* sendiri. Pengguna akan diminta memasukkan teks kalimat yang berkaitan dengan pengalaman nya menggunakan aplikasi LinkAja pada kolom yang disediakan. Jika telah selesai menginput kalimat, pengguna bisa mengklik tombol mulai *testing* untuk memulai proses analisis. Hasil *output* dari proses analisis itu adalah *input* ulasan, aspek, dan hasil prediksi sentimen. Tampilan halaman uji coba *real time* akan ditunjukkan pada gambar 4.10,



Gambar 4.10 Halaman Uji Coba *Real Time*

4.2. Impelementasi Model

4.2.1. Menyematkan *Vector Word2Vec* ke Dalam Aspek

Pertama-tama, data ulasan aplikasi LinkAja yang diberikan oleh pengguna perlu diambil dan diolah. Langkah-langkah *preprocessing* termasuk *case folding*, *punctual removal*, *normalisasi*, *stopword removal*, dan *stemming*. Setiap ulasan kemudian akan menjadi kalimat yang sudah di *preprocessing*. Dalam tahap ini, sebuah model *Word2Vec* akan dibangun menggunakan data ulasan yang sudah melewati tahap *preprocessing*. Model ini akan mempelajari relasi semantik antara kata-kata dalam

ulasan. Setiap kata akan diubah menjadi vektor representasi dalam ruang vektor yang memiliki dimensi tertentu. Model *Word2Vec* ini akan menjadi sarana untuk mengubah kata-kata dalam ulasan menjadi vektor numerik yang mengandung informasi semantiknya. Selanjutnya, setiap kata dalam ulasan akan diubah menjadi vektor menggunakan model *Word2Vec* yang telah dibangun. Namun, dalam analisis sentimen berbasis aspek, kita ingin mengetahui sentimen berdasarkan aspek. Oleh karena itu, kata-kata yang termasuk dalam aspek tertentu akan diambil dan dihitung rata-rata vektor representasinya. Hasil dari tahap ini adalah vektor representasi untuk setiap aspek dalam setiap ulasan. Sebagai contoh pembuatan *vector* untuk kalimat ulasan, disini peneliti menggunakan salah satu ulasan pada data *training* yang sudah melewati tahap *preprocessing*:

[“aplikasi kaya gak niat proses tingkat lama malas isi saldo gak kirim bank gara gara penuh servis tingkat”]

```
[ -1.1179411 -0.25979713 0.41064426 0.26110885 -0.14915258 -0.10759059
 1.2345035 -1.4982299 0.16222633 -0.19957459 -1.09961 -0.0624576
 1.1744814 -0.22923838 -1.4580913 -1.5385079 -0.33154103 -0.6450299
 -0.92036 0.77470666 0.08509225 -0.93687046 -0.27917188 -0.02122769
 0.9083392 0.5430362 0.93631077 -1.1226326 -1.3832319 0.28885198
 0.26033747 0.13311364 -0.40662104 0.06139438 -0.42856658 0.07623879
 -1.1916413 0.8681303 -0.8296256 -1.8651453 -0.42841062 -0.17685802
 -0.22453693 -0.91873956 0.17592007 -2.1726916 -0.59959555 -0.09897772
 -0.8750198 -0.01371063 0.9142535 1.1218116 0.62448317 -0.01631734
 0.07627594 -0.94223297 1.6149904 -0.24860308 1.6506276 -0.78922427
 -0.29092678 -0.50087607 1.7578475 -0.27352926 0.28605804 -1.0952678
 -0.6232148 0.4087165 1.0661445 0.08751667 -0.25947788 1.5708351
 -0.04174739 0.30429766 0.76514214 -0.11878321 0.04507435 1.3083968
 0.8264481 1.2899455 -0.999511 -0.36585823 0.11834821 -0.77981234
 0.10411187 0.583124 0.19477813 -0.9512258 0.6514465 1.0487056
 0.27957094 -1.2538068 -0.6795792 -0.06424116 1.1301312 -0.81229633
 -0.69834554 0.27771354 0.70466816 -0.8745999 ]
```

Gambar 4.11 Vector dari salah satu kalimat ulasan

vector representasi aspek yang sudah dihitung akan menjadi fitur masukan untuk model klasifikasi *XGBoost*. Setiap ulasan akan memiliki beberapa vektor representasi aspek yang sesuai dengan aspek yang ada dalam ulasan tersebut. Model *XGBoost* akan dilatih dengan menggunakan *dataset* yang telah dilabeli sentimen untuk setiap ulasan berdasarkan aspek tertentu. Model ini akan mempelajari pola sentimen berdasarkan representasi vektor aspek.

4.2.2. Pelatihan Model *Extreme Gradient Boosting*

Ketika melakukan klasifikasi menggunakan metode *XGBoost*, lakukan kalkulasi untuk mendapatkan nilai prediksi awal (X) dan nilai *residual* (Y) dengan menggunakan

persamaan 3.2 dan 3.3. Selanjutnya latih model menggunakan nilai prediksi yang sudah didapat dari model sebelumnya lalu lanjutkan ke model selanjutnya hingga M kali sampai menghasilkan model yang baik. Pada tabel 4.1 adalah contoh kalkulasi prediksi awal dan *residual* dengan menggunakan $base_score = 0$.

Tabel 4.1 Implementasi kalkulasi prediksi awal dan *residual*

X (kalimat ulasan)	aplikasi lama cek verifikasi ktp udah butuh tolong cepat verifnya
Y (label <i>actual</i>) <i>Customer Service</i>	0
Y (label <i>actual</i>) Fitur Aplikasi	0
Y (label <i>actual</i>) <i>User Experience</i>	-1
Y (label <i>actual</i>) Verifikasi	-1
$F_0(x)$	0
$\hat{Y} = Y - f_0(x)$ <i>Customer Service</i>	$0 - (0) = 0$
$\hat{Y} = Y - f_0(x)$ Fitur Aplikasi	$0 - (0) = 0$
$\hat{Y} = Y - f_0(x)$ <i>User Experience</i>	$-1 - (0) = -1$
$\hat{Y} = Y - f_0(x)$ Verifikasi	$-1 - (0) = -1$
X (kalimat ulasan)	aplikasi kaya gak niat proses tingkat lama malas isi saldo gak kirim bank gara gara penuh servis tingkat
Y (label <i>actual</i>) <i>Customer Service</i>	0
Y (label <i>actual</i>) Fitur Aplikasi	-1
Y (label <i>actual</i>) <i>User Experience</i>	-1
Y (label <i>actual</i>) Verifikasi	-1
$F_0(x)$	-1
$\hat{Y} = Y - f_0(x)$ <i>Customer Service</i>	$0 - (-1) = 1$
$\hat{Y} = Y - f_0(x)$ Fitur Aplikasi	$-1 - (-1) = 0$
$\hat{Y} = Y - f_0(x)$ <i>User Experience</i>	$-1 - (-1) = 0$
$\hat{Y} = Y - f_0(x)$ Verifikasi	$-1 - (-1) = 0$

Berdasarkan kalkulasi yang dilakukan pada tabel 4.1 akan dihasilkan model yang sudah dilatih (M1) dengan hasil nilai prediksi $f_1(x)$ dan *residual* (\hat{Y}_1) yang akan dipakai untuk melatih model (M2) agar menghasilkan nilai prediksi $f_2(x)$ dan *residual* (\hat{Y}_2). Selanjutnya perulangan akan dilakukan hingga mendapatkan nilai prediksi $f_{100}(x)$ dan *residual* (\hat{Y}_{100}). Pelatihan model adalah mengembangkan model prediksi sentimen berbasis aspek untuk ulasan pada aplikasi LinkAja menggunakan algoritma *XGBoost*. Algoritma *XGBoost* merupakan pilihan yang tepat untuk penelitian ini karena mampu mengatasi permasalahan klasifikasi yang kompleks serta memiliki kemampuan untuk memperoleh tingkat akurasi yang tinggi dalam berbagai skenario. Tahap awal dalam pelatihan model *XGBoost* adalah persiapan data. *Dataset* yang telah melewati tahap *preprocessing* sebelumnya akan dipecah menjadi *Content (features)* dan label (*target*) yang akan digunakan dalam proses *training*. *Content* akan terdiri dari vektor representasi kalimat ulasan berdasarkan model *Word2Vec*, sedangkan label akan berisi kategori sentimen berbasis aspek yang telah ditentukan. Setelah itu data akan dibagi atau di *split* menjadi data *training* dan data *testing*. Penulis membagi data menggunakan rasio perbandingan 70:30 atau 70% untuk data *training* dan 30% data *testing*. Selanjutnya, proses pembuatan model *XGBoost* dimulai. Pemilihan parameter yang optimal seperti *eta (learning_rate)* dan jumlah pohon (*n_estimators*) menjadi kunci dalam membentuk model yang akurat dan efisien. Untuk memilih parameter yang optimal maka dilakukan sebuah metode yang bernama *hyperparameter tuning*. *Hyperparameter tuning* adalah proses mencari kombinasi parameter yang optimal untuk model *machine learning*. Dalam konteks *XGBoost*, ini melibatkan penyesuaian parameter seperti *n_estimators* dan *learning_rate*. Proses ini dapat meningkatkan kinerja model, mencegah *overfitting*, dan menghasilkan model yang lebih baik secara umum. Pada penelitian ini peneliti menggunakan parameter *n_estimators* dengan range 50-150 dengan penambahan sebesar 50 [50, 100, 150] dan parameter *learning_rate* dengan range 0,25-0,75 dengan penambahan sebesar 25 [0,25, 0,50, 0,75]. Untuk mencari nilai *hyperparameter* dengan hasil yang terbaik peneliti menggunakan metode *Grid Search Validation* atau *GridSearchCV*. Proses mencari nilai *hyperparameter* yang terbaik dengan menggunakan metode *GridSearchCV* akan ditampilkan pada tabel 4.2.

Tabel 4.2 *Hyperparameter Tuning Menggunakan Metode Grid Search CV*

<i>Learning_rate</i>	<i>n_estimators</i>	<i>accuracy</i>
0,25	50	85,25%
0,25	100	88,12%
0,25	150	90,01%
0,50	50	87,49%
0,50	100	89,34%
0,50	150	90,18%
0,75	50	88,73%
0,75	100	92,11%
0,75	150	90,18%

Hasil *hyperparameter tuning* menggunakan metode *gridsearchCV* untuk mencari nilai parameter terbaik untuk model *XGBoost* ditunjukkan pada tabel 4.2 dimana telah ditemukan nilai parameter terbaik yaitu *learning_rate* = 0,75 dan *n_estimators* = 100 dengan akurasi sebesar 92,11%. Setelah parameter telah ditentukan maka *dataset* data *training* bisa dimasukkan untuk memulai proses *training data*. Hasil *training data* akan ditunjukkan pada tabel 4.1.

Tabel 4.3 Tabel Hasil *Training Data*

<i>Content</i>	<i>Preprocess</i>	<i>Customer Service</i>	Fitur Aplikasi	<i>User Experience</i>	Verifikasi
Ini kenapa lama banget si upgrade ke full service nya, padahal semua persyaratan dan data juga udah sesuai tapi masih aja lama. Udah 3 hari lewat loh. Mohon untuk segera di tuntaskan ya min.	lama si tingkat penuh servis syarat data udah sesuai aja lama udah mohon tuntas admin	0	0	-1	-1
Kenapa proses verifikasi sangat lama	proses verifikasi lama butuh uang	0	-1	-1	-1

<p>ya? Padahal saya sangat butuh uang yang ada di aplikasi, tapi malah nggak bisa diambil, tolong dong percepat proses verifikasi data, terimakasih, 1 lagi kenapa saldo saya hilang di aplikasi padahal terakhir saldonya 500.500 sekarang kok kosong ya? Link aja kenapa gak jelas banget sih? Siapa yang akan bertanggung jawab atas hilangnya dana nasabah?</p>	<p>aplikasi ambil tolong cepat proses verifikasi data terimakasih saldo hilang aplikasi saldo kosong linkaja aja gak tanggung hilang dana nasabah</p>				
<p>Kaya @*#*\$ Mint upgrade. Pas upgrade mentok di verifikasi wajah... Ga selesai selesai.. ulangggggg teros.. sampe pegel hati. Butuh duit itu untuk di tarik... Ga bisa bisa kepentok disuruh verifikasi wajah yg ga bisa bisa itu. Ngebug teros.... Tau gini mending pake apk dana aja teros. Makasi atas kekecewaan ini.</p>	<p>kaya tingkat pas tingkat mentok verifikasi wajah ga selesai selesai ulang rus gel hati butuh duit tarik ga kepentok suruh verifikasi wajah ga ngebug rus tau gin mending pakai aplikasi dana aja makas kecewa</p>	0	-1	-1	-1

Seperti yang terlihat pada tabel 4.2 kalimat ulasan “lama si tingkat penuh servis syarat data udah sesuai aja lama udah mohon tuntas admin” mendapatkan sentimen netral (0)

untuk aspek *customer service*, aspek fitur aplikasi mendapatkan sentimen netral (0), aspek *user experience* mendapatkan sentimen negatif (-1), dan aspek verifikasi mendapatkan sentimen negatif (-1).

4.2.3. Pengujian Model

Setelah tahap *training* data selesai dilakukan dan menghasilkan model, maka selanjutnya akan dilakukan pengujian model dengan menggunakan dataset data *testing* untuk menghasilkan hasil evaluasi dari algoritma yang digunakan. Hasil dari pengujian model dengan menggunakan dataset data *testing* akan ditunjukkan pada tabel 4.2.

Tabel 4.4 Hasil Pengujian Model Menggunakan Data *Testing*

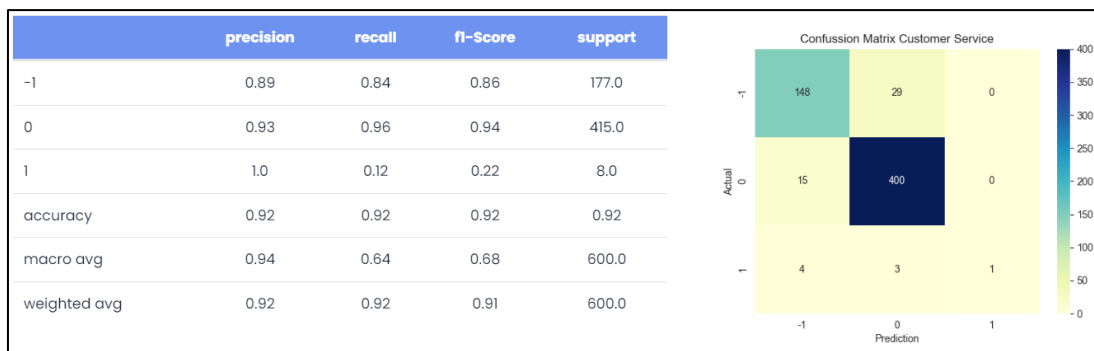
<i>Content</i>	<i>Preprocess</i>	Prediksi			
		<i>Customer Service</i>	Fitur Aplikasi	<i>User Experience</i>	Verifikasi
Gimana ini akun basic mulu mau tarik tunai gabisa. Sengaja banget ya udah update full service proses kok sampai 2minggu!!!! Penipuan ini mah	gimana akun mulu tarik tunai gabisa sengaja udah baharu penuh servis proses minggu tipu	0	-1	-1	-1
Verifikasi TERLAMA jatuh kepada LINKAJA. Beda dengan yg lain paling lama proses verifikasi hanya 1 hari. Ini sudah 5 hari belum juga.	verifikasi lama jatuh linkaja beda lama proses verifikasi	0	0	-1	-1
Sampah pelayanan nya, masa ini udh hampir 1 bulan upgrade akun belum jadi jadi. Nanya di live chat juga cuma template doang, ngirim email kaga ada balesan sama sekali. Pada buta apa ya staff nya, orang butuh cepet. Sekalian	sampah layan udah tingkat akun nanya langsung obrol templat ngirim email kaga balesan buta staf orang butuh	-1	0	-1	-1

aja satu tahun di lama lamain. maaf saya bicara kasar, SAMPAH STAF PELAYANANNYA.	cepat aja lama lamain maaf bicara kasar sampah staf layan				
Aplikasinya kurang bagus mau upgrade full servis ngulang mulu pas verifikasi wajah parah banget saya coba udh 30 menit bolak balik masih aja gk bisa beda ama ewallet yang lain sampah	aplikasi kurang bagus tingkat penuh servis ngulang mulu pas verifikasi wajah parah coba udah menit bolak aja gak beda ama dompet digital sampah	0	0	-1	-1
Proses full servise lama banget, gak guna nyesel gua isi saldo... Uang gak bisa dipake kirim2.... ?????? ðŸ˜ˆ ðŸ˜ˆ ðŸ˜ˆ ðŸ˜ˆ ; Buat tranfer gak bisa mo diambil juga gak bisa.... Maunya apa yaaaaaðŸ˜ˆ ðŸ˜ˆ ðŸ˜ˆ ðŸ˜ˆ ; Dalam proses mulu.... Amatir banget, Yg penting solusinya gmn, percuma hanya jawab pertanyaan aja ,hubungi ini, hubungi itu, gak langsung action.... ðŸ˜ˆ ðŸ˜ˆ ðŸ˜ˆ	proses penuh servis lama gak nyesal gua isi saldo uang gak dipake kirim transfer gak ambil gak yaaaaa proses mulu amatir solusi gimana aja hubung hubung gak langsung action	-1	-1	-1	-1

4.3. Evaluasi Model

Setelah model *XGBoost* diuji menggunakan data *testing*, maka selanjutnya dilakukan evaluasi kinerja menggunakan metrik-metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik-metrik ini memberikan gambaran tentang sejauh mana model dapat

mengklasifikasikan sentimen berdasarkan aspek dengan tepat. Hasil evaluasi juga akan mencakup *confusion matrix* yang menunjukkan perbandingan antara prediksi dan label sebenarnya. *Confusion Matrix* dari aspek *Customer Service* akan ditunjukkan pada gambar 4.12.



Gambar 4.12 *Confusion Matrix Aspek Customer Service*

Berdasarkan *confusion matrix* dari aspek *customer service* yang sudah ditunjukkan bisa dilihat untuk aspek negatif (-1) mendapatkan *precision* sebesar 0,89 atau 89%, *recall* sebesar 0,84, dan *f1-score* sebesar 0,86. Untuk aspek netral (0) mendapatkan *precision* sebesar 0,93, *recall* sebesar 0,96, dan *f1-score* sebesar 0,94. Untuk aspek positif (1) mendapatkan *precision* sebesar 1,0, *recall* sebesar 0,12, dan *f1-score* sebesar 0,22. *accuracy* yang didapat untuk aspek *customer service* adalah sebesar 0,92 atau 92%. Pada tabel 4.3, tabel 4.4, dan tabel 4.5 akan ditunjukkan nilai *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* dari masing-masing sentimen negatif, sentimen netral, dan sentimen positif pada aspek *customer service*.

Tabel 4.5 *Confusion Matrix Sentimen Negatif Aspek Customer Service*

No	Sentimen Negatif	Total
1	TP (<i>True Positif Customer Service</i>)	148
2	TN (<i>True Negatif Customer Service</i>)	404
3	FP (<i>False Positif Customer Service</i>)	19
4	FN (<i>False Negatif Customer Service</i>)	29

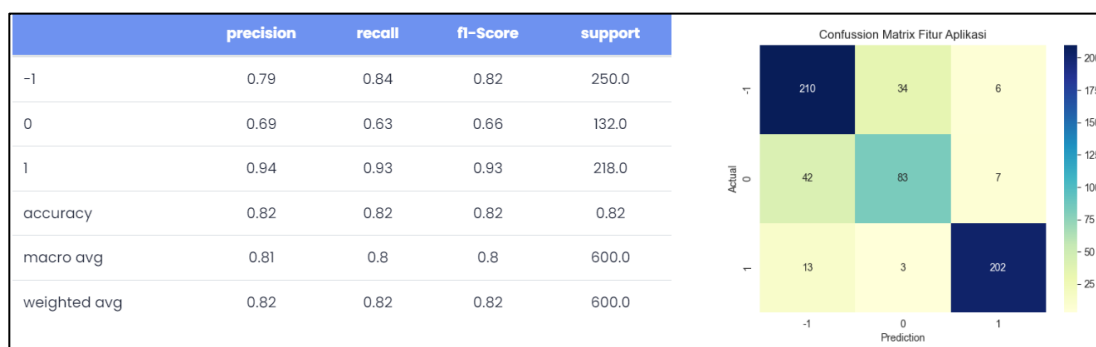
Tabel 4.6 *Confusion Matrix* Sentimen Netral Aspek Customer Service

No	Sentimen Netral	Total
1	TP (<i>True Positif Customer Service</i>)	400
2	TN (<i>True Negatif Customer Service</i>)	153
3	FP (<i>False Positif Customer Service</i>)	32
4	FN (<i>False Negatif Customer Service</i>)	15

Tabel 4.7 *Confusion Matrix* Sentimen Positif aspek Customer Service

No	Sentimen Positif	Total
1	TP (<i>True Positif Customer Service</i>)	1
2	TN (<i>True Negatif Customer Service</i>)	592
3	FP (<i>False Positif Customer Service</i>)	0
4	FN (<i>False Negatif Customer Service</i>)	7

Selanjutnya, *confusion matrix* dari aspek Fitur Aplikasi akan ditampilkan pada gambar 4.13.

**Gambar 4.13** *Confusion Matrix* Aspek Fitur Aplikasi

Berdasarkan *confusion matrix* dari aspek Fitur Aplikasi yang sudah ditunjukkan, bisa dilakukan perhitungan. Pada tabel 4.6, tabel 4.7, dan tabel 4.8 akan ditunjukkan nilai *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* dari

masing-masing sentimen negatif, sentimen netral, dan sentimen positif pada aspek Fitur Aplikasi.

Tabel 4.8 *Confusion Matrix* Sentimen Negatif Aspek Fitur Aplikasi

No	Sentimen Negatif	Total
1	TP (<i>True</i> Positif Fitur Aplikasi)	210
2	TN (<i>True</i> Negatif Fitur Aplikasi)	295
3	FP (<i>False</i> Positif Fitur Aplikasi)	55
4	FN (<i>False</i> Negatif Fitur Aplikasi)	40

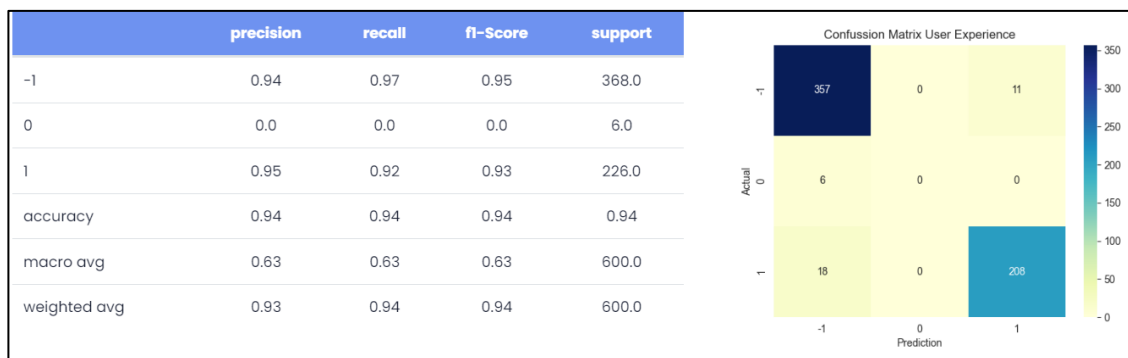
Tabel 4.9 *Confusion Matrix* Sentimen Netral Aspek Fitur Aplikasi

No	Sentimen Netral	Total
1	TP (<i>True</i> Positif Fitur Aplikasi)	83
2	TN (<i>True</i> Negatif Fitur Aplikasi)	431
3	FP (<i>False</i> Positif Fitur Aplikasi)	37
4	FN (<i>False</i> Negatif Fitur Aplikasi)	49

Tabel 4.10 *Confusion Matrix* Sentimen Positif Aspek Fitur Aplikasi

No	Sentimen Positif	Total
1	TP (<i>True</i> Positif Fitur Aplikasi)	202
2	TN (<i>True</i> Negatif Fitur Aplikasi)	369
3	FP (<i>False</i> Positif Fitur Aplikasi)	13
4	FN (<i>False</i> Negatif Fitur Aplikasi)	16

Selanjutnya, *confusion matrix* dari aspek User Experience akan ditampilkan pada gambar 4.14.



Gambar 4.14 *Confusion Matrix* Aspek User Experience

Berdasarkan *confusion matrix* dari aspek User Experience yang sudah ditunjukkan, bisa dilakukan perhitungan. Pada tabel 4.9, tabel 4.10, dan tabel 4.11 akan ditunjukkan nilai *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* dari masing-masing sentimen negatif, sentimen netral, dan sentimen positif pada aspek User Experience.

Tabel 4.11 *Confusion Matrix* Sentimen Negatif Aspek User Experience

No	Sentimen Negatif	Total
1	TP (<i>True Positif User Experience</i>)	357
2	TN (<i>True Negatif User Experience</i>)	208
3	FP (<i>False Positif User Experience</i>)	24
4	FN (<i>False Negatif User Experience</i>)	11

Tabel 4.12 *Confusion Matrix* Sentimen Netral Aspek User Experience

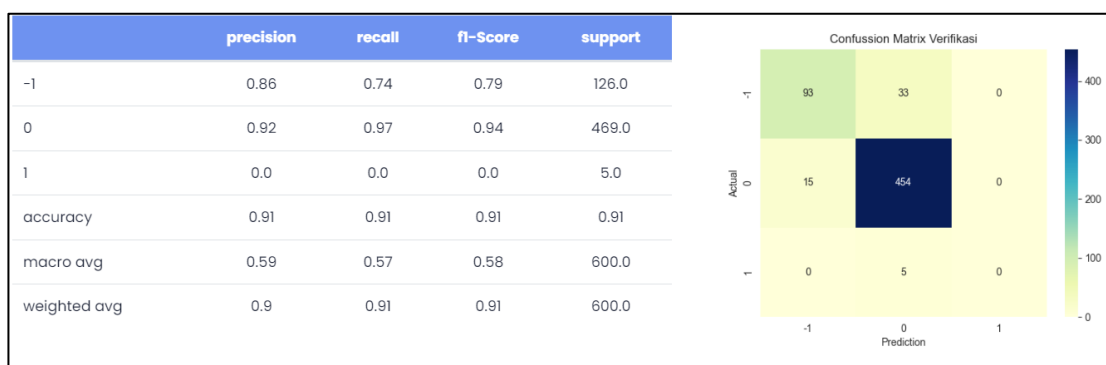
No	Sentimen Netral	Total
1	TP (<i>True Positif User Experience</i>)	0
2	TN (<i>True Negatif User Experience</i>)	594
3	FP (<i>False Positif User Experience</i>)	0
4	FN (<i>False Negatif User Experience</i>)	6

Tabel 4.13 *Confusion Matrix* Sentimen Positif Aspek User Experience

No	Sentimen Positif	Total
----	------------------	-------

1	TP (<i>True Positif User Experience</i>)	208
2	TN (<i>True Negatif User Experience</i>)	363
3	FP (<i>False Positif User Experience</i>)	11
4	FN (<i>False Negatif User Experience</i>)	18

Selanjutnya, *confusion matrix* dari aspek *User Experience* akan ditampilkan pada gambar 4.15.



Gambar 4.15 *Confusion Matrix* Aspek Verifikasi

Berdasarkan *confusion matrix* dari aspek Verifikasi yang sudah ditunjukkan, bisa dilakukan perhitungan. Pada tabel 4.12, tabel 4.13, dan tabel 4.14 akan ditunjukkan nilai *True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)* dari masing-masing sentimen negatif, sentimen netral, dan sentimen positif pada aspek Verifikasi.

Tabel 4.14 *Confusion Matrix* Sentimen Negatif Aspek Verifikasi

No	Sentimen Negatif	Total
1	TP (<i>True Positif Verifikasi</i>)	93
2	TN (<i>True Negatif Verifikasi</i>)	459
3	FP (<i>False Positif Verifikasi</i>)	15
4	FN (<i>False Negatif Verifikasi</i>)	33

Tabel 4.15 *Confusion Matrix* Sentimen Netral Aspek Verifikasi

No	Sentimen Netral	Total
----	-----------------	-------

1	TP (<i>True</i> Positif Verifikasi)	454
2	TN (<i>True</i> Negatif Verifikasi)	93
3	FP (<i>False</i> Positif Verifikasi)	38
4	FN (<i>False</i> Negatif Verifikasi)	15

Tabel 4.16 *Confusion Matrix* Sentimen Positif Aspek Verifikasi

No	Sentimen Positif	Total
1	TP (<i>True</i> Positif Verifikasi)	0
2	TN (<i>True</i> Negatif Verifikasi)	595
3	FP (<i>False</i> Positif Verifikasi)	0
4	FN (<i>False</i> Negatif Verifikasi)	5

Penyebab terjadinya keakuratan model dalam memprediksi data dapat disebabkan oleh beberapa factor. Peneliti melakukan analisis penyebab terjadinya ketidakakuratan prediksi yang dilakukan oleh model. Yaitu:

1. Bias Data pada data *testing* yang tidak terdeteksi yang dapat mempengaruhi hasil prediksi model. Misal jika dataset *training* cenderung mewakili sentimen negatif atau sentimen tertentu maka model dapat mengikut bias ini.
2. Adanya lebih dari satu kata yang memiliki preferensi negatif yang terkandung dalam ulasan sehingga model salah memprediksi ulasan yang harus nya negatif menjadi positif.

Berdasarkan hasil yang ada tabel 4.3 sampai dengan tabel 4.14 yaitu hasil sentimen setiap aspek bisa dilakukan evaluasi untuk menghitung nilai *accuracy*, *precision*, *recall*, dan *f1-score* berdasarkan persamaan yang sudah dibahas pada bab 3.

Perhitungan nilai *Precision* aspek *Customer Service*:

$$Precision \text{ sentimen negatif aspek } Customer \text{ Service } (-1) = \frac{TP}{TP+FP} = \frac{148}{148+19} = 0,89$$

$$Precision \text{ sentimen netral aspek } Customer \text{ Service } (0) = \frac{TP}{TP+FP} = \frac{400}{400+32} = 0,93$$

$$Precision \text{ sentimen positif aspek } Customer \text{ Service } (1) = \frac{TP}{TP+FP} = \frac{1}{1+0} = 1$$

Perhitungan nilai *Recall* aspek *Customer Service*:

$$\text{Recall sentimen negatif aspek Customer Service (-1)} = \frac{TP}{TP+FN} = \frac{148}{148+29} = 0,84$$

$$\text{Recall sentimen netral aspek Customer Service (0)} = \frac{TP}{TP+FN} = \frac{400}{400+15} = 0,96$$

$$\text{Recall sentimen positif aspek Customer Service (1)} = \frac{TP}{TP+FN} = \frac{1}{1+7} = 0,12$$

Perhitungan nilai *f1-score* aspek *Customer Service*:

F1-score sentimen negatif aspek *Customer Service* (-1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,84 \times 0,89}{0,84 + 0,89} = 0,86$$

F1-score sentimen netral aspek *Customer Service* (0) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,96 \times 0,93}{0,96 + 0,93} = 0,94$$

F1-score sentimen positif aspek *Customer Service* (1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,12 \times 1}{0,12 + 1} = 0,22$$

Perhitungan nilai *Precision* aspek Fitur Aplikasi:

$$\text{Precision sentimen negatif aspek Fitur Aplikasi (-1)} = \frac{TP}{TP+FP} = \frac{210}{210+55} = 0,79$$

$$\text{Precision sentimen netral aspek Fitur Aplikasi (0)} = \frac{TP}{TP+FP} = \frac{83}{83+37} = 0,69$$

$$\text{Precision sentimen positif aspek Fitur Aplikasi (1)} = \frac{TP}{TP+FP} = \frac{202}{202+13} = 0,94$$

Perhitungan nilai *Recall* aspek Fitur Aplikasi:

$$Recall \text{ sentimen negatif aspek Fitur Aplikasi } (-1) = \frac{TP}{TP+FN} = \frac{210}{210+40} = 0,84$$

$$Recall \text{ sentimen netral aspek Fitur Aplikasi } (0) = \frac{TP}{TP+FN} = \frac{83}{83+49} = 0,63$$

$$Recall \text{ sentimen positif aspek Fitur Aplikasi } (1) = \frac{TP}{TP+FN} = \frac{202}{202+16} = 0,93$$

Perhitungan nilai *f1-score* aspek Fitur Aplikasi:

F1-score sentimen negatif aspek Fitur Aplikasi (-1) =

$$2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0,84 \times 0,79}{0,84 + 0,79} = 0,82$$

F1-score sentimen netral aspek Fitur Aplikasi (0) =

$$2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0,63 \times 0,69}{0,63 + 0,69} = 0,66$$

F1-score sentimen positif aspek Fitur Aplikasi (1) =

$$2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0,93 \times 0,94}{0,93 + 0,94} = 0,93$$

Perhitungan nilai *Precision* aspek *User Experience*:

$$Precision \text{ sentimen negatif aspek } User Experience (-1) = \frac{TP}{TP+FP} = \frac{357}{357+24} = 0,94$$

$$Precision \text{ sentimen netral aspek } User Experience (0) = \frac{TP}{TP+FP} = \frac{0}{0+0} = 0$$

$$Precision \text{ sentimen positif aspek } User Experience (1) = \frac{TP}{TP+FP} = \frac{208}{208+11} = 0,95$$

Perhitungan nilai *Recall* aspek *User Experience*:

$$\text{Recall sentimen negatif aspek User Experience (-1)} = \frac{TP}{TP+FN} = \frac{357}{357+11} = 0,97$$

$$\text{Recall sentimen netral aspek User Experience (0)} = \frac{TP}{TP+FN} = \frac{0}{0+6} = 0$$

$$\text{Recall sentimen positif aspek User Experience (1)} = \frac{TP}{TP+FN} = \frac{208}{208+18} = 0,92$$

Perhitungan nilai *f1-score* aspek *User Experience*:

F1-score sentimen negatif aspek *User Experience* (-1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,97 \times 0,94}{0,97 + 0,94} = 0,95$$

F1-score sentimen netral aspek *User Experience* (0) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0 \times 0}{0 + 0} = 0$$

F1-score sentimen positif aspek *User Experience* (1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,92 \times 0,95}{0,92 + 0,95} = 0,93$$

Perhitungan nilai *Precision* aspek Verifikasi:

$$\text{Precision sentimen negatif aspek Verifikasi (-1)} = \frac{TP}{TP+FP} = \frac{93}{93+15} = 0,86$$

$$\text{Precision sentimen netral aspek Verifikasi (0)} = \frac{TP}{TP+FP} = \frac{454}{454+38} = 0,92$$

$$\text{Precision sentimen positif aspek Verifikasi (1)} = \frac{TP}{TP+FP} = \frac{0}{0+0} = 0$$

Perhitungan nilai *Recall* aspek Verifikasi:

$$\text{Recall sentimen negatif aspek Verifikasi (-1)} = \frac{TP}{TP+FN} = \frac{93}{93+33} = 0,74$$

$$\text{Recall sentimen netral aspek Verifikasi (0)} = \frac{TP}{TP+FN} = \frac{454}{454+15} = 0,97$$

$$\text{Recall sentimen positif aspek Verifikasi (1)} = \frac{TP}{TP+FN} = \frac{0}{0+5} = 0$$

Perhitungan nilai *f1-score* aspek Verifikasi:

F1-score sentimen negatif aspek Verifikasi (-1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,74 \times 0,86}{0,74 + 0,86} = 0,79$$

F1-score sentimen netral aspek Verifikasi(0) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0,97 \times 0,92}{0,97 + 0,92} = 0,94$$

F1-score sentimen positif aspek Verifikasi (1) =

$$2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2 \times \frac{0 \times 0}{0 + 0} = 0$$

Perhitungan *accuracy* dari setiap aspek:

$$\text{Accuracy aspek Customer Service} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{549}{600} = 0,92$$

$$\text{Accuracy aspek Fitur Aplikasi} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{495}{600} = 0,82$$

$$\text{Accuracy aspek User Experience} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{565}{600} = 0,94$$

$$\text{Accuracy aspek Verifikasi} = \frac{TP+TN}{TP+FP+TN+FN} = \frac{547}{600} = 0,91$$

Dari perhitungan *accuracy* setiap aspek, maka total *accuracy* dari keempat aspek adalah:

$$\text{Accuracy} = \frac{0,92+0,82+0,94+0,91}{4} \times 100\% = 0,90 \times 100\% = 90\%$$

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang didapat dari penelitian Sentimen Analisis Berbasis Aspek Terhadap *Review Aplikasi Digital Wallet Menggunakan Metode Extreme Gradient Boosting* adalah:

1. Kinerja performa algoritma *Extreme Gradient Boosting* dan *word embedding Word2Vec* menunjukkan hasil yang bagus dengan memperoleh akurasi sebesar 90% yang didapat dari evaluasi *confusion matrix*.
2. Berdasarkan hasil yang didapat, dapat disimpulkan bahwa algoritma *Extreme Gradient Boosting* dan *word embedding Word2Vec* sangat cocok dalam mengatasi masalah identifikasi ataupun klasifikasi dan bekerja dengan baik untuk penelitian Sentimen Analisis Berbasis Aspek Terhadap *Review Aplikasi Digital Wallet Menggunakan Metode Extreme Gradient Boosting*.

5.2. Saran

Walaupun penelitian bisa dikatakan telah selesai dilakukan dengan hasil yang baik, tetapi tetap saja pasti ada kekurangan pada penelitian ini. Saran dari penulis untuk pengembangan penelitian ini adalah menambahkan lebih banyak data sehingga kata-kata di dalam dataset menjadi lebih lengkap dan lebih beragam. Dengan demikian model diharapkan dapat bekerja menjadi lebih baik dan lebih fleksibel dalam mengatasi sentimen analisis berbasis aspek.

DAFTAR PUSTAKA

- Abdurrahman, G., & Sintawati, M. (2020). Implementation of xgboost for classification of parkinson's disease. *Journal of Physics: Conference Series*, 1538(1).
<https://doi.org/10.1088/1742-6596/1538/1/012024>
- Ailiyya, S. (2020). *Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Tokopedia Menggunakan Support Machine*. Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah
- Astuti, S. (2020). *Analisis Sentimen Berbasis Aspek Pada Aplikasi Tokopedia Menggunakan Lda Dan Naïve Bayes*. BS Thesis. Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc."
- Blei, D. M., Ng, A. Y., & Edu, J. B. (2003). Latent Dirichlet Allocation Michael I. Jordan. In *Journal of Machine Learning Research* (Vol. 3).
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.
- Griffiths, T., Steyvers, M., Blei, D., & Tenenbaum, J. (2004). Integrating topics and syntax. *Advances in Neural Information Processing Systems*, 17.
- Haumahu, J. P., Permana, S. D. H., & Yaddarabullah, Y. (2021). Fake news classification for Indonesian news using Extreme Gradient Boosting (XGBoost). *IOP Conference Series: Materials Science and Engineering*, 1098(5), 052081. <https://doi.org/10.1088/1757-899x/1098/5/052081>
- Ikegami, A., Dewa, I., Bayu, M., & Darmawan, A. (2022). Analisis Sentimen dan Pemodelan Topik Ulasan Aplikasi Noice Menggunakan XGBoost dan LDA. In *JNATIA* (Vol. 1, Issue 1).
- Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining text data* (pp. 415–463). Springer.

- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.
- Manyika, J., Lund, S., Singer, M., White, O., & Berry, C. (2016). Digital finance for all: Powering inclusive growth in emerging economies. *McKinsey Global Institute*, 1–15.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. <http://arxiv.org/abs/1301.3781>
- Mubarok, M. S., Adiwijaya, A., & Aldhi, M. D. (2017). Aspect-based sentiment analysis to review products using Naïve Bayes. *AIP Conference Proceedings*, 1867. <https://doi.org/10.1063/1.4994463>
- Rachmi, A. (2020). *Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn*. Universitas Islam Negeri Syarif Hidayatullah
- Rayhan Rahmanda, & Erwin Budi Setiawan. (2022). Word2Vec on Sentiment Analysis with Synthetic Minority Oversampling Technique and Boosting Algorithm. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 6(4), 599–605. <https://doi.org/10.29207/resti.v6i4.4186>
- Rifqy, O. :, Turjaman, M., & Budi, I. (2022). *ANALISIS SENTIMEN BERBASIS ASPEK MARKETING MIX TERHADAP ULASAN APLIKASI DOMPET DIGITAL (STUDI KASUS: APLIKASI LINKAJA PADA TWITTER)* (Vol. 30). Agustus.
- Rizaldi, M. A., Fathoni, M., & Yetty, F. (2021). Faktor Determinasi Minat Penggunaan Layanan LinkAja Syariah pada Masyarakat Jabodetabek. *Journal of Sharia Economics*, 2(2), 120–140.
- Rizki Ariel giffari, M. (2022). *Analisis Sentimen Berbasis Aspek Pada Ulasan Aplikasi Tangerang Live Menggunakan Latent Dirichlet Allocation dan Naive Bayes*. Universitas Islam Negeri Syarif Hidayatullah
- Saldaña, Z. W. (2018). Sentiment Analysis for Exploratory Data Analysis. *Programming Historian*.
- Schouten, K., Frasincar, F., & Dekker, R. (2016). An information gain-driven feature study for aspect-based sentiment analysis. *Natural Language Processing and Information Systems: 21st International Conference on Applications of Natural Language to*

Information Systems, NLDB 2016, Salford, UK, June 22-24, 2016, Proceedings 21, 48–59.

Uddin, M. S., & Akhi, A. Y. (2014). E-wallet system for Bangladesh an electronic payment system. *International Journal of Modeling and Optimization*, 4(3), 216.