

**EVALUASI KINERJA ALGORITMA REGRESI LOGISTIK
MENGUNAKAN *OVERSAMPLING* SMOTE PADA ANALISIS
SENTIMEN ULASAN PENGGUNA APLIKASI *MOBILE* JKN**

SKRIPSI

**OLEH
DIAH MARIATUL ULYA
NIM. 220601110002**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**EVALUASI KINERJA ALGORITMA REGRESI LOGISTIK
MENGUNAKAN *OVERSAMPLING* SMOTE PADA ANALISIS
SENTIMEN ULASAN PENGGUNA APLIKASI *MOBILE* JKN**

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Diah Mariatul Ulya
NIM. 220601110002**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

EVALUASI KINERJA ALGORITMA REGRESI LOGISTIK MENGUNAKAN *OVERSAMPLING SMOTE* PADA ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI *MOBILE JKN*

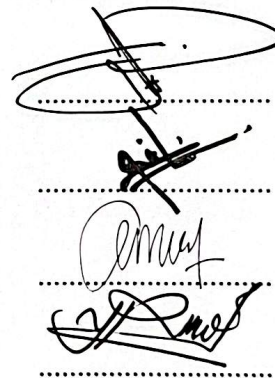
SKRIPSI

Oleh
Diah Mariatul Ulya
NIM. 220601110002

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Malang, 19 Februari 2026

Ketua Penguji : Hisyam Fahmi, M.Kom
Anggota Penguji 1 : Mohammad Nafie Jauhari, M.Si
Anggota Penguji 2 : Dr. Mohammad Jamhuri, M.Si
Anggota Penguji 3 : Erna Herawati, M.Pd



.....
.....
.....
.....

Mengetahui,
Ketua Program Studi Matematika

Achror Rozi, M.Si
NIP. 19800527 200801 1 012

**EVALUASI KINERJA ALGORITMA REGRESI LOGISTIK
MENGUNAKAN *OVERSAMPLING* SMOTE PADA ANALISIS
SENTIMEN ULASAN PENGGUNA APLIKASI *MOBILE* JKN**

SKRIPSI

Oleh
Diah Mariatul Ulya
NIM. 220601110002

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

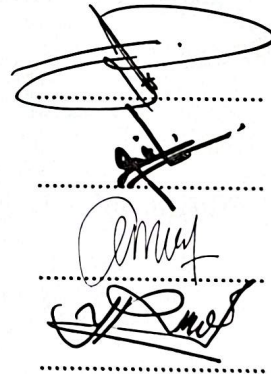
Malang, 19 Februari 2026

Ketua Penguji : Hisyam Fahmi, M.Kom

Anggota Penguji 1 : Mohammad Nafie Jauhari, M.Si

Anggota Penguji 2 : Dr. Mohammad Jamhuri, M.Si

Anggota Penguji 3 : Erna Herawati, M.Pd



Mengetahui,
Ketua Program Studi Matematika

Dochrur Rozi, M.Si
NIP. 19800527 200801 1 012

PERNYATAN KEASLIAN TULISAN

Saya bertanda tangan di bawah ini

Nama : Diah Mariatul Ulya
NIM : 220601110002
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Evaluasi Kinerja Algoritma Regresi Logistik
Menggunakan *Oversampling* SMOTE pada Analisis
Sentimen Ulasan Pengguna Aplikasi *Mobile* JKN

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan di halaman terakhir. Apabila dikemudian hari terbukti skripsi ini adalah hasil jiplakan atau tiruan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 19 Februari 2026



Diah Mariatul Ulya
NIM. 220601110002

MOTO

"Allah memang tidak menjanjikan hidupmu akan selalu mudah. tapi dua kali Allah berjanji bahwa: *"fa inna ma'al-'usri Yusra, Inna ma'al-'usri yusra".*"

(QS. Al-Insyirah 94:5-6)

“Apapun yang sudah terjadi dalam hidupmu, jangan katakan “seandainya”, tapi katakan “*Qodarullah*” karena semua yang terjadi adalah takdir Allah dan takdir Allah itu selalu baik, karena Allah maha baik.”

(Ustadz Hanan Attaki)

“Semua jatuh bangunmu hal yang biasa. Angan dan pertanyaan waktu yang menjawabnya, berikan tenggat waktu bersedihlah secukupnya. Rayakan perasaanmu sebagai manusia.”

(Baskara Putra–Hindia)

“Orang lain ga akan paham *struggle* dan masa sulitnya kita yang mereka ingin tahu hanya bagian *success stories*. Berjuanglah untuk diri sendiri walaupun tidak ada tepuk tangan. Kelak diri kita dimasa depan akan sangat bangga dengan apa yang kita perjuangkan hari ini.”

(Fardi Yandi)

HALAMAN PERSEMBAHAN

Tiada kata yang lebih indah dalam perjalanan penulisan skripsi ini selain rasa syukur atas setiap proses yang telah dilalui. Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang, karya sederhana ini penulis persembahkan untuk:

1. Sembah sujud dan syukur kepada Allah SWT, yang selalu menguatkan saat lelah, menenangkan saat *overthinking*, dan memudahkan hal-hal yang terasa mustahil.
2. Cinta pertama dan pintu surgaku, Ayah Imam Muhsin dan Ibu Umi Sa'diyah. Terima kasih untuk doa yang tidak pernah putus, untuk kerja keras yang mungkin tidak selalu terlihat, dan untuk keyakinan bahwa aku pasti bisa. Semua ini tidak akan sampai di titik ini tanpa kalian.
3. Vivi, Indi, dan Caca, saudara kandung yang selalu jadi tempat pulang, tempat cerita, dan tempat bercanda di tengah stresnya revisian.
4. Feri, yang sudah sabar membersamai proses ini. Terima kasih sudah tetap ada, mendengar keluh kesah, dan menguatkan saat rasanya ingin menyerah.
5. Afifah, Faizah, Maulany, dan Nada, partner berlima yang tahu banget naik turunnya perjalanan ini. Terima kasih sudah jadi bagian dari cerita yang akan selalu dikenang.
6. Dan untuk Syafi, Thoriq, Bintang, Farah, Oci, Firman, dan Fisal, teman-teman Komputasi yang berjuang bareng dari awal sampai akhir. Terima kasih untuk diskusi, support, dan kebersamaan yang nggak tergantikan.

Semoga setiap doa, dukungan, dan kebaikan yang kalian beri, Allah balas dengan hal-hal terbaik.

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas rahmat, taufik, dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Matematika (S.Mat) pada Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, serta motivasi dari berbagai pihak. Oleh karena itu, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu selama proses penyusunan skripsi ini. Ucapan terima kasih secara khusus penulis sampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nurdiana, M.Si., CAHRM., CRMP, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang, atas kepemimpinan dan arahnya.
2. Dr. H. Agus Mulyono, S.Pd., M.Kes, selaku Dekan Fakultas Sains dan Teknologi, atas dukungan dan fasilitas yang diberikan selama masa studi.
3. Dr. Fachrur Rozi, M.Si., selaku Ketua Program Studi Matematika, atas bimbingan dan motivasi yang diberikan selama penulis menempuh pendidikan.
4. Dr. Mohammad Jamhuri, M.Si., selaku Dosen Pembimbing I, yang telah memberikan bimbingan, dukungan, dan arahan yang sangat berharga dalam penyusunan skripsi ini. Semoga ilmu dan waktu yang telah beliau berikan menjadi amal jariyah yang bernilai di sisi Allah SWT.
5. Erna Herawati, M.Pd., selaku Dosen Pembimbing II, atas kesabaran, masukan, serta saran yang sangat membantu dalam proses penelitian dan penulisan skripsi ini.
6. Dr. Elly Susanti, M.Sc., selaku Dosen Wali, yang senantiasa memberikan arahan, nasihat, dan motivasi selama masa perkuliahan.
7. Hisyam Fahmi, M.Kom., selaku Dosen sekaligus Ketua Konsorsium Komputasi, atas ilmu, bimbingan, serta inspirasi yang diberikan dalam bidang komputasi.
8. Seluruh Dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang telah

membekali penulis dengan ilmu, motivasi, dan semangat belajar selama masa studi.

9. Bapak Imam Muhsin dan Ibu Umi Sa'diyah, selaku orang tua penulis, atas doa, kasih sayang, serta dukungan yang tiada henti. Segala jerih payah dan doa yang diberikan menjadi sumber kekuatan utama bagi penulis untuk menyelesaikan skripsi ini.
10. Dia Zulfi Arifah selaku kakak penulis, adik Nindi, dan adik Caca, sekaligus sanak saudara yang senantiasa memberikan doa, dukungan, dan cinta kasih tanpa henti.
11. Seorang yang selalu memberi dukungan, bernama Feri, atas perhatian, semangat, dan doa yang tulus selama proses penyusunan naskah skripsi ini. Terima kasih atas dorongan yang menjadi penyemangat bagi penulis hingga penelitian ini dapat terselesaikan dengan baik.
12. Teman-teman berlima, Afifah, Faizah, Maulany, dan Nada, selaku teman dekat yang telah bersama penulis sejak semester satu hingga sekarang.
13. Teman-teman Konsorsium Komputasi (Bintang, Farah, Firman, Fisal, Ochi, Syafi, dan Thoriq) atas kerja sama, dan dukungan selama perkuliahan. Kebersamaan ini menjadi pengalaman berharga bagi penulis dalam mengembangkan kemampuan akademik maupun profesional.
14. Seluruh mahasiswa Matematika angkatan 2022 (Mathdeux), atas kebersamaan, dukungan, dan semangat yang diberikan selama proses perkuliahan hingga penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi kesempurnaan karya ini. Semoga skripsi ini dapat memberikan manfaat, baik bagi penulis maupun bagi pembaca.

Malang, 19 Februari 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAN KEASLIAN TULISAN	v
MOTO.....	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR SIMBOL	xiii
DAFTAR LAMPIRAN	xiv
ABSTRAK	xv
ABSTRACT	xvi
مستخلص البحث.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	7
1.3 Tujuan Penelitian.....	8
1.4 Manfaat Penelitian.....	8
1.5 Batasan Masalah	9
1.6 Definisi Istilah	11
BAB II KAJIAN TEORI	12
2.1 Kajian Teori Pendukung.....	12
2.1.1 <i>Natural Language Processing</i> (NLP).....	12
2.1.2 Analisis Sentimen	12
2.1.3 <i>Text Preprocessing</i>	14
2.1.4 <i>Machine learning</i> (ML).....	16
2.1.5 Algoritma Klasifikasi.....	18
2.1.6 Regresi Logistik.....	20
2.1.7 <i>Term Frequency–Inverse Document Frequency</i> (TF-IDF)	26
2.1.8 <i>Synthetic Minority Oversampling Technique</i> (SMOTE)	28
2.1.9 Metrik Evaluasi.....	30
2.1.10 <i>Mobile JKN</i>	33
2.2 Kajian Integrasi Topik dengan Alquran/Hadis	35
2.3 Kajian Topik dengan Teori Pendukung.....	37
BAB III METODE PENELITIAN	40
3.1 Jenis Penelitian	40
3.2 Data dan Sumber Data.....	40
3.3 Teknik Pengumpulan Data	41
3.4 Teknik Analisis Data	43
3.4.1 <i>Exploratory Data Analysis</i> (EDA).....	45
3.4.2 <i>Text Preprocessing</i>	45

3.4.3	<i>Split Data</i>	47
3.4.4	Pelabelan.....	48
3.4.5	<i>Wordcloud Visualization</i>	48
3.4.6	Ekstraksi Fitur.....	48
3.4.7	Penanganan <i>Imbalanced Data</i>	49
3.4.8	Pemodelan dan Pelatihan Data	50
3.4.9	Evaluasi Model	51
3.5	Skenario Pengujian dan Analisis Perbandingan	52
BAB IV HASIL DAN PEMBAHASAN.....		54
4.1	Hasil.....	54
4.1.1	Pengumpulan Data.....	54
4.1.2	<i>Exploratory Data Analysis (EDA)</i>	55
4.1.3	<i>Text Preprocessing</i>	56
4.1.4	Pelabelan.....	61
4.1.5	<i>Wordcloud Visualization</i>	63
4.1.6	<i>Split Data</i>	65
4.1.7	Ekstraksi Fitur.....	65
4.1.8	Penanganan <i>Imbalanced Data</i>	71
4.1.9	Pemodelan dan Pelatihan Data	75
4.1.10	Evaluasi Model	86
4.2	Pembahasan	92
4.2.1	Skenario Tanpa SMOTE dan Dengan SMOTE.....	92
4.2.2	Uji Pengaruh Ukuran Data terhadap Performa Model.....	95
4.2.3	Pembahasan Hasil Evaluasi Variasi Parameter Temperatur (α) pada Fungsi Sigmoid.....	98
4.2.4	Pembahasan Perbandingan dengan Metode Perbandingan	101
4.2.5	Perbandingan dengan Penelitian Terdahulu	104
4.3	Kajian Integrasi Islam pada Proses Penyeimbangan Data Menggunakan SMOTE	106
BAB V PENUTUP.....		108
5.1	Kesimpulan.....	108
5.2	Saran	109
DAFTAR PUSTAKA		111
LAMPIRAN.....		118
RIWAYAT HIDUP		122

DAFTAR TABEL

Tabel 4.1	Sampel Data Ulasan <i>Mobile JKN</i>	54
Tabel 4.2	Sampel Hasil <i>Character Cleaning</i>	57
Tabel 4.3	Sampel Hasil <i>Case Folding</i>	58
Tabel 4.4	Sampel Hasil <i>Text Normalization</i>	58
Tabel 4.5	Sampel Hasil <i>Stopword Removal</i>	59
Tabel 4.6	Sampel Hasil <i>Stemming</i>	60
Tabel 4.7	Hasil Pelabelan.....	61
Tabel 4.8	Sampel Data TF-IDF.....	66
Tabel 4.9	Frekuensi Kemunculan Kata pada Dokumen.....	67
Tabel 4.10	Perhitungan TF pada D1	67
Tabel 4.11	Perhitungan TF.....	68
Tabel 4.12	Perhitungan IDF	69
Tabel 4.13	Perhitungan TF-IDF pada D1	70
Tabel 4.14	Perhitungan TF-IDF	70
Tabel 4.15	Perhitungan Fungsi Sigmoid.....	80
Tabel 4.16	Gradien pada D1	83
Tabel 4.17	<i>Confusion Matrix</i> Klasifikasi Data <i>Testing</i> tanpa SMOTE	89
Tabel 4.18	<i>Confusion Matrix</i> Klasifikasi Data <i>Testing</i> dengan SMOTE	90
Tabel 4.19	Hasil Perbandingan Evaluasi dengan dan tanpa SMOTE.....	91
Tabel 4.20	Hasil Uji Proporsi Data.....	95

DAFTAR GAMBAR

Gambar 3.1	<i>Flowchart</i> Penelitian	44
Gambar 4.1	Persebaran Rating pada Dataset	55
Gambar 4.2	Wordcloud Dataset sebelum Preprocessing	56
Gambar 4.3	Hasil Clean Data.....	60
Gambar 4.4	Perbandingan Dataset Mentah & Clean Data.....	61
Gambar 4.5	Grafik Distribusi Label Sentimen	62
Gambar 4.6	Wordcloud Sentimen Positif	63
Gambar 4.7	Wordcloud Sentimen Negatif.....	64
Gambar 4.8	Grafik Distribusi Label Sentimen Setelah SMOTE	74
Gambar 4.9	<i>Learning curve</i> tanpa SMOTE	87
Gambar 4.10	<i>Learning curve</i> dengan SMOTE	88
Gambar 4.11	<i>Learning curve</i> Hasil Evaluasi Variasi Parameter Sigmoid.....	99

DAFTAR SIMBOL

- $\mathbf{x}^{(i)}$: Vektor fitur ulasan ke- i .
 $y^{(i)}$: Label kelas sentimen sebenarnya untuk ulasan ke- i , dengan $y^{(i)} \in \{0,1\}$.
 \mathbf{w} : Vektor bobot/koeffisien regresi logistik.
 b : Bias (*intercept*) pada model regresi logistik.
 $z^{(i)}$: Skor linier hasil kombinasi fitur, $z^{(i)} = \mathbf{w}^T \mathbf{x}^{(i)} + b$.
 $p^{(i)}$: Probabilitas ulasan ke- i termasuk kelas positif.
 $\hat{y}^{(i)}$: Prediksi kelas sentimen ulasan ke- i .
 α : *Learning rate*, laju pembaruan parameter algoritma *gradient descent*.

DAFTAR LAMPIRAN

Lampiran 1 Data Ulasan Aplikasi <i>Mobile JKN</i>	117
Lampiran 2 <i>Source Code Scraping Data</i>	117
Lampiran 3 Hasil <i>Text Preprocessing</i>	118
Lampiran 4 <i>Source Code</i> Evaluasi Kinerja Regresi Logistik dengan <i>Oversampling SMOTE</i>	118

ABSTRAK

Ulya, Diah Mariatul. 2025. **Evaluasi Kinerja Algoritma Regresi Logistik Menggunakan *Oversampling* SMOTE pada Analisis Sentimen Ulasan Pengguna Aplikasi *Mobile JKN***. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (1) Dr. Mohammad Jamhuri, M.Si. (2) Erna Herawati, M.Pd.

Kata kunci: Analisis Sentimen, Regresi Logistik, SMOTE, TF-IDF, *Mobile JKN*

Penelitian ini membahas penerapan algoritma Regresi Logistik dalam klasifikasi sentimen berbasis teks serta pengaruh penggunaan *Synthetic Minority Over-sampling Technique* (SMOTE) pada data dengan distribusi kelas tidak seimbang. Data yang digunakan berupa kumpulan ulasan pengguna aplikasi *Mobile JKN* yang diperoleh dari *Google Play Store* melalui proses pengumpulan data *web scraping*, kemudian dilakukan pelabelan sentimen untuk keperluan analisis. Pendekatan penelitian dilakukan melalui pengolahan data teks dan pembentukan fitur numerik, kemudian model klasifikasi dilatih pada kondisi tanpa SMOTE dan dengan SMOTE untuk dibandingkan kinerjanya. Evaluasi performa model dilakukan menggunakan metrik klasifikasi umum serta dianalisis melalui pola *learning curve* untuk melihat dinamika pembelajaran model. Hasil penelitian menunjukkan bahwa perbedaan performa akhir antara model tanpa SMOTE dan dengan SMOTE secara numerik relatif kecil, sehingga peningkatan yang diperoleh bersifat moderat. Meskipun demikian, analisis *learning curve* memperlihatkan bahwa model dengan SMOTE menghasilkan pola pembelajaran yang lebih stabil, memiliki fluktuasi performa yang lebih rendah, serta menunjukkan konvergensi yang lebih konsisten seiring bertambahnya ukuran data latih dibandingkan model tanpa SMOTE. Temuan ini menunjukkan bahwa kontribusi utama SMOTE tidak terletak pada peningkatan nilai metrik secara signifikan, melainkan pada perbaikan stabilitas dan proses pembelajaran model dalam menangani data teks yang tidak seimbang.

ABSTRACT

Ulya, Diah Mariatul. 2025. **Performance Evaluation of Regresi Logistik Using SMOTE Oversampling in Sentiment Analysis of Mobile JKN Application User Reviews.** Undergraduate Thesis. Department of Mathematics, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisors: (1) Dr. Mohammad Jamhuri, M.Si. (2) Erna Herawati, M.Pd.

Keywords: Sentiment Analysis, Logistic Regression, SMOTE, TF-IDF, Mobile JKN

This study examines the application of the Regresi Logistik algorithm in text-based sentiment classification and analyzes the effect of the Synthetic Minority Over-sampling Technique (SMOTE) on datasets with imbalanced class distributions. The dataset consists of user reviews of the Mobile JKN application collected from the Google Play Store through a web scraping process, followed by sentiment labeling for analytical purposes. The research methodology involves text preprocessing and numerical feature representation, after which classification models are trained under two conditions: without SMOTE and with SMOTE, to enable performance comparison. Model performance is evaluated using standard classification metrics and further analyzed through *learning curve* patterns to observe the dynamics of the learning process. The experimental results indicate that the numerical difference in final performance between models without SMOTE and with SMOTE is relatively small, resulting in only moderate performance improvement. Nevertheless, the *learning curve* analysis demonstrates that the model trained with SMOTE exhibits a more stable learning pattern, lower performance fluctuation, and more consistent convergence as the training data size increases compared to the model without SMOTE. These findings suggest that the primary contribution of SMOTE lies not in significantly boosting evaluation metrics, but in improving the stability and learning process of the model when handling imbalanced text data.

مستخلص البحث

العلي، دياه مارية. ٢٠٢٥. تقييم كفاءة خوارزمية الانحدار اللوجستي باستخدام تقنية *SMOTE* في تحليل مشاعر تعليقات مستخدمي تطبيق *Mobile JKN*. البحث الجامعي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (1) د. محمد جمهوري، الماجستير في العلوم. (2) إيرنا هيراواتي، الماجستير في التربية.

الكلمات الأساسية: تحليل المشاعر، الانحدار اللوجستي، *Mobile JKN*، *TF-IDF*، *SMOTE*

يهدف هذا البحث إلى دراسة تطبيق خوارزمية الانحدار اللوجستي في تحليل المشاعر النصية، وكذلك تحليل تأثير استخدام تقنية *SMOTE* على البيانات ذات التوزيع غير المتوازن. استخدمت الدراسة بيانات تعليقات مستخدمي تطبيق *Mobile JKN* التي تم جمعها من متجر *Google Play* من خلال أسلوب *WebScraping*، ثم جرى تصنيفها إلى فئات مشاعر لغرض التحليل. تم تنفيذ منهجية البحث عبر مراحل المعالجة المسبقة للنصوص، وبناء تمثيل عددي للبيانات باستخدام *TF-IDF*، ثم تدريب نموذج الانحدار اللوجستي في حالتين، الأولى بدون استخدام *SMOTE*، والثانية باستخدامها، وذلك لمقارنة أداء النموذجين. أظهرت نتائج البحث أن الفارق في الأداء النهائي بين النموذجين كان محدوداً نسبياً، حيث لم يحقق استخدام *SMOTE* زيادة كبيرة في قيم مقاييس الأداء. ومع ذلك، أوضح تحليل منحني التعلم أن النموذج المدعوم بتقنية *SMOTE* يتمتع بدرجة أعلى من الاستقرار وتقارب أكثر اتساقاً مع زيادة حجم بيانات التدريب مقارنة بالنموذج بدون *SMOTE*. وتشير هذه النتائج إلى أن الدور الأساسي لتقنية *SMOTE* يتمثل في تحسين استقرار عملية التعلم والتعامل مع عدم توازن البيانات، وليس في تحقيق ارتفاع كبير في قيم مقاييس الأداء النهائية.

BAB I PENDAHULUAN

1.1 Latar Belakang

Analisis sentimen merupakan salah satu bidang yang berkembang pesat dalam kajian data teks *modern* karena ulasan, komentar, dan opini pengguna di platform digital mampu memberikan gambaran nyata tentang pengalaman mereka terhadap suatu layanan. Analisis sentimen yang dilakukan secara tepat waktu membantu pengembang dalam merespons keluhan, memperbaiki kualitas fitur, serta merancang strategi pengembangan yang lebih efektif untuk mencegah ketidakpuasan pengguna (Crocamo dkk., 2021). Selain itu, pemahaman terhadap penilaian pengguna—baik melalui ulasan teks maupun *rating*—dapat menggambarkan tingkat kepuasan mereka terhadap kualitas layanan dan menjadi dasar evaluasi bagi pengembang dalam meningkatkan kualitas sistem (Rahmawati dkk., 2023). Dengan demikian, analisis sentimen tidak hanya berfungsi sebagai alat pemantauan reaktif, tetapi juga sebagai landasan dalam pengambilan keputusan strategis untuk menciptakan pengalaman pengguna yang lebih baik. Respons yang cepat dan tepat terhadap penilaian negatif juga dapat memperkuat citra positif aplikasi dan meningkatkan loyalitas pengguna dalam jangka panjang (Fauzan dkk., 2024).

Secara umum, analisis sentimen merupakan permasalahan klasifikasi, yaitu proses pemetaan suatu data masukan ke dalam kategori tertentu berdasarkan pola dan karakteristiknya (Tao & Fang, 2020). Dalam konteks penelitian ini, polaritas sentimen ditentukan berdasarkan *rating* yang diberikan pengguna, sehingga sentimen positif dan negatif diperoleh langsung dari nilai *rating* tersebut.

Pendekatan ini dipilih karena *rating* dianggap mewakili tingkat kepuasan pengguna dan memberikan dasar pelabelan yang jelas untuk proses klasifikasi. Meskipun penentuan label sentimen bertumpu pada *rating*, data ulasan tetap perlu diubah ke dalam bentuk numerik agar dapat diproses oleh model pembelajaran mesin. Representasi fitur dilakukan melalui teknik seperti *bag-of-words*, *term frequency-inverse document frequency* (TF-IDF), atau *word embedding* yang berfungsi menangkap pola kemunculan kata dalam suatu dokumen (Naeem dkk., 2022). Algoritma klasifikasi yang umum digunakan dalam analisis sentimen di antaranya *Naive Bayes*, *Support Vector Machine (SVM)*, *K-Nearest Neighbor (KNN)*, dan *Regresi Logistik* (Gupta, 2022). Setiap algoritma memiliki keunggulan dan keterbatasan, baik dari segi akurasi, efisiensi pelatihan, maupun kemampuannya dalam menangani karakteristik data teks yang berdimensi tinggi, sehingga pemilihan metode perlu disesuaikan dengan kebutuhan penelitian.

Analisis sentimen seperti ini secara umum merupakan bagian dari teknik data mining, yaitu proses mengekstraksi pola atau informasi tersembunyi dari kumpulan data besar. Namun, dalam konteks ini pendekatan *machine learning* berperan penting dalam mendukung klasifikasi sentimen karena mampu mempelajari pola dari data historis. Hal ini membuat sistem dapat mengenali karakteristik teks dan memprediksi kategori sentimen pada data baru secara otomatis.

Salah satu algoritma klasifikasi yang dapat digunakan dalam analisis sentimen adalah Regresi Logistik. Algoritma ini termasuk dalam metode *machine learning* yang dirancang untuk memodelkan hubungan antara satu atau lebih variabel independen dengan variabel dependen biner (dua kelas), seperti sentimen positif dan negatif. Algoritma ini bekerja dengan menghitung probabilitas suatu

data masuk ke dalam salah satu kelas, sehingga ideal untuk membedakan opini positif dan negatif. Regresi Logistik bekerja dengan menghitung probabilitas suatu data termasuk ke dalam salah satu kelas, menjadikannya sangat cocok untuk klasifikasi biner (Lee & Seo, 2022).

Berdasarkan BPJS RI tahun 2025, *Mobile JKN* (Jaminan Kesehatan Negara) merupakan inovasi digital berbasis *smartphone* yang bertujuan untuk memudahkan masyarakat dalam mengakses layanan jaminan kesehatan. Aplikasi ini menyediakan berbagai fitur seperti pendaftaran dan perubahan data kepesertaan, pengecekan informasi anggota keluarga, serta riwayat pembayaran iuran. Selain itu, *Mobile JKN* juga memfasilitasi akses ke pelayanan di FKTP dan FKTL, serta menyediakan sarana bagi pengguna untuk menyampaikan saran atau keluhan secara langsung (Lumi dkk., 2023). Seiring meningkatnya jumlah pengguna, muncul berbagai opini dan ulasan di *platform* digital seperti *Google Play Store*. Beberapa pengguna merasa terbantu oleh kemudahan akses yang ditawarkan aplikasi ini, sementara yang lain mengeluhkan kendala seperti *error* sistem, lambatnya waktu respon, maupun desain antarmuka yang kurang intuitif. Opini-opini pengguna tersebut menjadi sumber informasi berharga yang dapat dianalisis untuk memahami persepsi masyarakat terhadap kualitas aplikasi. Analisis sentimen berperan dalam mengungkap kecenderungan opini pengguna, baik positif, negatif, maupun netral, sehingga perusahaan dapat memahami persepsi mereka dan melakukan peningkatan layanan yang tepat (Fauzan dkk., 2024).

Beberapa penelitian sebelumnya yang pernah dilakukan terkait klasifikasi yaitu penerapan metode Regresi Logistik untuk analisis sentimen pada media sosial. Penelitian oleh Aliman dkk. (2022) menerapkan Regresi Logistik untuk

menganalisis sentimen pada data *Twitter* dengan tujuan mendeteksi potensi krisis kesehatan mental, dan hasil penelitian tersebut menunjukkan bahwa metode ini efektif digunakan dalam klasifikasi sentimen. Namun, penelitian oleh (Husen dkk., 2023) pada opini publik *Twitter* mengenai Bank Syariah Indonesia (BSI) menunjukkan bahwa Regresi Logistik tidak memberikan hasil terbaik, karena terdapat algoritma lain seperti SVM yang menunjukkan performa lebih tinggi. Hal serupa juga ditemukan pada penelitian (Burrhanuddin & Fauzan Rozi, 2025) terhadap ulasan pengguna aplikasi *RedBus* di *Google Play Store*, di mana Regresi Logistik digunakan sebagai pembanding namun hasilnya lebih rendah dibandingkan SVM. Oleh karena itu, penelitian ini difokuskan pada penerapan Regresi Logistik dalam analisis sentimen ulasan *Mobile JKN* untuk melihat sejauh mana efektivitas algoritma ini pada konteks layanan kesehatan *digital*.

Selain itu, penelitian yang secara langsung membahas aplikasi *Mobile JKN* telah dilakukan salah satunya yaitu penelitian oleh (Al dkk., 2025) menggunakan algoritma BERT dengan representasi fitur berupa *BERT Embedding* untuk analisis sentimen ulasan *Mobile JKN* dari *Google Play Store* dan menunjukkan bahwa pendekatan *deep learning* mampu memberikan hasil klasifikasi yang baik. Sementara itu, (Tamami dkk., 2025) menerapkan metode LSTM dengan fitur dari *Embedding Layer* yang dipadukan dengan teknik SMOTE untuk mengatasi ketidakseimbangan kelas pada dataset ulasan *Mobile JKN*. Penelitian lain oleh (Sugihartono & Putra, 2024) menggunakan metode *Support Vector Machine* (SVM) dalam analisis sentimen ulasan *Mobile JKN* dan berhasil menunjukkan bahwa algoritma berbasis *machine learning* klasik tetap kompetitif dalam menghasilkan klasifikasi yang akurat. Selain itu, studi oleh (Hematang dkk., 2025)

membandingkan KNN dan *Regresi Logistik* pada data Mobile JKN, dan menunjukkan bahwa *Regresi Logistik* memiliki performa yang lebih baik dibandingkan KNN dalam klasifikasi sentimen.

Dari studi literatur di atas, dapat disimpulkan bahwa meskipun algoritma *deep learning* seperti BERT sering memberikan hasil dengan evaluasi tertinggi, metode klasik seperti SVM dan *Naïve Bayes* tetap relevan digunakan karena lebih ringan, efisien, dan mudah diimplementasikan. Pemilihan metode dalam analisis sentimen perlu mempertimbangkan *trade-off* antara akurasi dan kompleksitas, serta kebutuhan praktis dalam pengembangan aplikasi *Mobile JKN*. Di sisi lain, penelitian dengan algoritma *machine learning* klasik, khususnya *Regresi Logistik*, masih relatif terbatas dalam konteks aplikasi layanan publik digital. *Regresi Logistik* dikenal memiliki sejumlah keunggulan, antara lain kemudahan implementasi, efisiensi komputasi, serta interpretabilitas model yang menjadikannya populer untuk klasifikasi biner. Selain itu, algoritma ini juga terbukti mampu menangani dataset yang besar dan beragam secara efektif sehingga sering dipilih dalam penelitian klasifikasi teks. Dengan demikian, *Regresi Logistik* menjadi pilihan yang tepat untuk analisis polaritas penilaian pengguna aplikasi *Mobile JKN*, yang jumlah datanya terus meningkat menggunakan pendekatan SMOTE.

Dalam mengembangkan layanan publik berbasis teknologi, *machine learning* berperan penting dalam menganalisis data secara cepat dan tepat untuk meningkatkan kualitas layanan. Salah satu penerapannya adalah analisis sentimen, yang mampu mengolah ribuan ulasan pengguna secara otomatis guna mengidentifikasi kecenderungan penilaian positif dan negatif terhadap suatu

aplikasi. Proses ini memberikan masukan konstruktif bagi pengembang untuk memperbaiki sistem dan memastikan layanan publik berjalan lebih optimal. Peran *machine learning* ini selaras dengan prinsip Islam tentang kebermanfaatannya ilmu dan amal, sebagaimana firman Allah SWT dalam Surah Al-Qashash ayat 77:

وَابْتِغِ فِيمَا آتَاكَ اللَّهُ الدَّارَ الْآخِرَةَ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا وَأَحْسِنْ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ وَلَا تَبْغِ
الْفُسَادَ فِي الْأَرْضِ إِنَّ اللَّهَ لَا يُحِبُّ الْمُفْسِدِينَ ﴿٧٧﴾

“Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan bagianmu dari (kenikmatan) duniawi, serta berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik kepadamu, dan janganlah kamu berbuat kerusakan di muka bumi. Sungguh, Allah tidak menyukai orang-orang yang membuat kerusakan.” (NU Online, 2025)

Tafsir QS. Al-Qashash ayat 77 menjelaskan bahwa manusia diperintahkan untuk menyeimbangkan orientasi akhirat dan dunia, yaitu memanfaatkan segala karunia Allah SWT—baik harta, ilmu, maupun teknologi—untuk meraih kebahagiaan akhirat tanpa melupakan bagian kenikmatan dunia yang halal. Ayat ini juga menekankan pentingnya berbuat baik (*ihsan*) kepada sesama sebagaimana Allah SWT telah berbuat baik kepada manusia, sekaligus melarang keras perbuatan *fasād* atau kerusakan di muka bumi karena Allah SWT tidak menyukai orang-orang yang merusak. Dalam konteks penelitian ini, ayat tersebut dapat dimaknai bahwa ilmu pengetahuan dan teknologi seperti *machine learning* adalah karunia Allah SWT yang harus digunakan untuk menghadirkan kebaikan. Penerapan *machine learning* pada analisis sentimen tidak hanya bermanfaat dalam meningkatkan efisiensi dan ketepatan layanan publik, tetapi juga mencerminkan pengamalan nilai Islami berupa kepedulian sosial, kebermanfaatannya, serta upaya menghindari kerusakan, sehingga selaras dengan pesan QS. Al-Qashash ayat 77.

Berdasarkan uraian latar belakang tersebut, analisis polaritas penilaian pengguna memiliki peran penting dalam mengevaluasi tingkat kepuasan terhadap aplikasi *Mobile JKN* serta memberikan masukan bagi pengembang dalam meningkatkan kualitas layanan. Penelitian terdahulu menunjukkan bahwa algoritma seperti SVM, LSTM, dan BERT sering menghasilkan performa lebih tinggi dibandingkan Regresi Logistik, namun Regresi Logistik tetap relevan digunakan karena unggul dalam efisiensi komputasi, interpretabilitas, dan kemudahan penerapan pada data berdimensi besar. Oleh karena itu, penelitian ini berfokus pada evaluasi dan peningkatan kinerja Regresi Logistik melalui penerapan *Synthetic Minority Oversampling Technique* (SMOTE) sebagai strategi penyeimbangan data, dengan harapan mampu memperbaiki kemampuan model dalam mengenali kelas minoritas serta menghasilkan klasifikasi polaritas yang lebih akurat dan seimbang pada analisis sentimen berbasis *rating* pengguna aplikasi *Mobile JKN*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, rumusan masalah pada penelitian ini adalah:

1. Bagaimana performa algoritma Regresi Logistik dalam mengklasifikasikan polaritas ulasan pengguna aplikasi *Mobile JKN*?
2. Bagaimana pengaruh penerapan teknik *oversampling* dengan metode *Synthetic Minority Oversampling Technique* (SMOTE) terhadap peningkatan performa klasifikasi Regresi Logistik pada data ulasan pengguna yang tidak seimbang?

1.3 Tujuan Penelitian

Berdasarkan uraian rumusan masalah di atas, tujuan pada penelitian ini adalah:

1. Untuk menganalisis performa algoritma Regresi Logistik dalam mengklasifikasikan polaritas ulasan pengguna aplikasi *Mobile* JKN.
2. Untuk mengetahui pengaruh penerapan teknik *oversampling* dengan metode *Synthetic Minority Oversampling Technique* (SMOTE) terhadap peningkatan performa klasifikasi Regresi Logistik pada data ulasan pengguna yang tidak seimbang.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian, maka penelitian ini diharapkan dapat memberikan manfaat bagi seluruh pihak sebagai berikut:

1. Manfaat Teoritis

Secara teoritis, hasil penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang *machine learning*, khususnya dalam penerapan algoritma Regresi Logistik untuk klasifikasi polaritas penilaian pengguna pada data yang tidak seimbang. Penelitian ini juga memperkaya kajian mengenai pengaruh teknik *oversampling Synthetic Minority Oversampling Technique* (SMOTE) terhadap peningkatan performa model klasifikasi. Dengan demikian, penelitian ini dapat menjadi referensi tambahan bagi peneliti lain dalam mengembangkan model prediktif pada kasus serupa yang melibatkan data ulasan pengguna atau dataset dengan distribusi kelas yang tidak seimbang.

2. Manfaat Praktis

Manfaat praktis dari penelitian ini dapat dirasakan oleh berbagai pihak. Bagi pengembang aplikasi *Mobile JKN*, hasil analisis polaritas *rating* pengguna dapat memberikan gambaran umum mengenai kecenderungan penilaian positif dan negatif terhadap layanan, sehingga dapat dijadikan dasar dalam meningkatkan kualitas fitur dan kinerja aplikasi. Bagi BPJS Kesehatan maupun instansi terkait, penelitian ini dapat menjadi acuan dalam mengevaluasi kepuasan pengguna secara kuantitatif berdasarkan distribusi *rating* serta membantu dalam merumuskan strategi pengembangan layanan yang lebih adaptif terhadap kebutuhan masyarakat. Selain itu, penelitian ini juga bermanfaat bagi peneliti maupun praktisi, karena dapat dijadikan rujukan dalam mengimplementasikan metode analisis polaritas berbasis *rating* menggunakan algoritma Regresi Logistik dan teknik SMOTE pada berbagai domain lain, seperti pendidikan, kesehatan, maupun layanan publik digital.

1.5 Batasan Masalah

Penelitian ini memiliki beberapa batasan yang perlu diperhatikan untuk menjaga relevansi dan keakuratan hasil yang diperoleh, yaitu sebagai berikut:

1. Sumber data penelitian dibatasi pada ulasan pengguna aplikasi *Mobile JKN* yang diambil dari *Google Play Store* pada perangkat berbasis *Android*.
2. Dataset terdiri atas 10.000 ulasan berbahasa Indonesia, yang dianggap representatif untuk menggambarkan persepsi pengguna terhadap aplikasi *Mobile JKN*.
3. Analisis sentimen dalam penelitian ini menggunakan asumsi bahwa *rating* pengguna merepresentasikan isi ulasan yang diberikan. Dengan demikian,

proses pelabelan sentimen dilakukan berdasarkan nilai *rating* tanpa melakukan pembacaan atau penilaian manual terhadap konten teks secara mendalam. Analisis yang dilakukan bersifat polarisasi sentimen (positif vs negatif) dan tidak mencakup analisis aspek (*aspect-based sentiment analysis/ABSA*), sehingga model tidak mengidentifikasi aspek spesifik seperti fitur aplikasi, kinerja layanan, maupun pengalaman teknis pengguna.

4. Kelas sentimen ditentukan berdasarkan nilai *rating*: *rating* 1–2 dikategorikan sebagai negatif dan *rating* 4–5 sebagai positif. *Rating* 3 diabaikan karena bersifat netral dan tidak menunjukkan kecenderungan yang jelas. Penghapusan *rating* 3 dilakukan sejak tahap *web scraping* untuk memastikan fokus analisis pada dua kelas utama (positif dan negatif), sehingga model Regresi Logistik dapat beroperasi secara optimal.
5. Penelitian ini menggunakan algoritma Regresi Logistik sebagai model utama klasifikasi. Untuk mengatasi ketidakseimbangan data, diterapkan teknik *Synthetic Minority Oversampling Technique (SMOTE)*. Algoritma lain hanya digunakan sebagai pembanding dan tidak menjadi fokus utama penelitian.
6. Variasi parameter kecuraman sigmoid (α) dibatasi pada empat nilai, yaitu 0.5, 1.0, 2.0, dan 3.0. Nilai tersebut dipilih untuk merepresentasikan tingkat kecuraman rendah, standar, menengah, dan tinggi dalam konteks regresi logistik. Variasi ini digunakan untuk mengevaluasi pengaruh sensitivitas fungsi aktivasi terhadap dinamika pelatihan dan performa model. Nilai α di luar rentang tersebut tidak diuji dalam penelitian ini.
7. Perhitungan matematis yang ditampilkan secara manual pada bagian pemodelan menggunakan $\alpha = 1.0$ sebagai representasi regresi logistik standar.

Nilai tersebut dipilih agar ilustrasi perhitungan lebih sederhana dan konsisten dengan formulasi dasar regresi logistik. Variasi nilai α lainnya hanya dianalisis melalui eksperimen komputasional dan evaluasi performa, tanpa ditampilkan perhitungan manual secara rinci.

8. Data ulasan yang dianalisis diambil dalam rentang waktu 15 September 2025 hingga 18 November 2025. Dengan demikian, hasil penelitian hanya mencerminkan persepsi pengguna pada periode tersebut dan tidak dapat digeneralisasikan untuk seluruh waktu penggunaan aplikasi *Mobile JKN*.

1.6 Definisi Istilah

1. *Feature Extraction* : Proses mengubah teks mentah menjadi format numerik yang dapat diproses oleh algoritma *machine learning*.
2. *Imbalanced Data* : Kondisi ketika jumlah data dalam satu kelas jauh lebih banyak dibandingkan kelas lainnya yang dapat mempengaruhi performa model klasifikasi.
3. *Threshold* : Nilai batas probabilitas yang digunakan untuk mengubah *output* probabilistik menjadi klasifikasi biner.
4. *Confusion matrix* : Tabel yang menunjukkan performa model dengan membandingkan hasil prediksi dan hasil sebenarnya.
5. *Split Data* : Proses membagi data menjadi dua bagian, yaitu *training* data untuk melatih model, dan *testing* data untuk menguji performa model.

6. *Training* : Proses melatih model untuk mengenali pola pada data latih sehingga mampu melakukan klasifikasi.
7. *Testing* : Proses menguji kinerja model pada data baru yang belum pernah digunakan saat *training* untuk mengetahui performa prediksi model.
8. SMOTE : Teknik untuk mengatasi *imbalanced data* dengan cara membuat data sintetis dari kelas minoritas agar distribusi kelas menjadi lebih seimbang.
9. *Overfitting* : Kondisi di mana model terlalu menyesuaikan diri dengan data latih sehingga performanya tinggi pada data tersebut tetapi buruk pada data baru.
10. *Tunning* : Proses menyetel atau mengatur nilai parameter pada suatu model agar performanya optimal.
11. *Trade-off* : Suatu kondisi ketika peningkatan pada satu aspek harus dibayar dengan penurunan pada aspek lain.

BAB II

KAJIAN TEORI

2.1 Kajian Teori Pendukung

2.1.1 *Natural Language Processing* (NLP)

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada interaksi antara komputer dan bahasa alami manusia (Tao & Fang, 2020). NLP memungkinkan mesin untuk memahami, menginterpretasikan, dan menghasilkan bahasa dalam bentuk teks maupun suara sehingga dapat digunakan dalam berbagai aplikasi berbasis bahasa. Tujuan utama NLP adalah menjembatani komunikasi antara manusia dan komputer agar mesin dapat mengolah informasi linguistik secara otomatis, akurat, dan bermakna (Widiantoro dkk., 2024).

Secara umum, NLP mencakup berbagai tugas penting, seperti *text classification*, *machine translation*, *information retrieval*, *text summarization*, hingga *sentiment analysis* (Widiantoro dkk., 2024). Salah satu tantangan utama dalam NLP adalah menangani keragaman bahasa alami yang penuh dengan ambiguitas, variasi penulisan, serta konteks yang berbeda. Oleh karena itu, NLP memerlukan kombinasi antara linguistik, ilmu komputer, dan statistika untuk menghasilkan model yang mampu memproses bahasa manusia secara efektif.

2.1.2 Analisis Sentimen

Analisis sentimen adalah cabang dari NLP yang berfokus pada pengolahan bahasa alami untuk mengidentifikasi dan mengkategorikan opini, sikap, atau emosi dalam sebuah teks (Tao & Fang, 2020). Hasil analisis ini umumnya diklasifikasikan ke dalam kategori positif, negatif, atau netral (Fauzan dkk., 2024).

Teknik ini banyak digunakan pada berbagai platform digital seperti media sosial, ulasan produk, dan forum daring, karena dapat memberikan gambaran mengenai persepsi dan pengalaman pengguna (Rahmawati dkk., 2023). Dengan demikian, analisis sentimen menjadi salah satu metode penting dalam memahami respons masyarakat terhadap suatu produk atau layanan.

Dalam perkembangannya, analisis sentimen tidak selalu bergantung pada pemrosesan isi teks secara mendalam. Salah satu pendekatan yang semakin banyak digunakan adalah analisis polaritas berbasis *rating*, yaitu teknik yang memanfaatkan nilai skor atau *rating* yang diberikan pengguna sebagai representasi kecenderungan sentimen mereka. *Rating* numerik dianggap mampu mencerminkan tingkat kepuasan atau ketidakpuasan secara langsung, di mana *rating* tinggi menunjukkan opini positif dan *rating* rendah menandakan kecenderungan negatif. Pendekatan ini dikenal sebagai *sentiment polarity based on rating*, yang menilai arah sentimen secara umum tanpa melakukan analisis semantik terhadap isi ulasan (Hu dkk., 2014).

Penelitian ini secara khusus menggunakan pendekatan tersebut, yaitu memfokuskan analisis sentimen pada polaritas *rating* sebagai indikator kepuasan pengguna. Pendekatan ini dipilih karena lebih efisien, objektif, serta sesuai untuk dataset besar yang memiliki keragaman bahasa tinggi, sehingga proses pelabelan dapat dilakukan secara konsisten tanpa menghadapi *bias labeler* ataupun kompleksitas analisis semantik.

Seiring berkembangnya teknologi digital, analisis sentimen tidak lagi terbatas pada pengelompokan opini, melainkan juga dimanfaatkan sebagai alat strategis dalam mendukung pengambilan keputusan (Rahmawati dkk., 2023).

Perusahaan dapat menggunakannya untuk mengevaluasi kualitas produk dan layanan berdasarkan opini konsumen, sedangkan lembaga pemerintah dapat memanfaatkannya untuk mengukur keberhasilan program publik. Lebih lanjut, analisis sentimen tidak hanya berfungsi sebagai alat monitoring yang bersifat reaktif, melainkan juga sebagai sistem prediktif yang mampu mengantisipasi permasalahan dan mengidentifikasi peluang pengembangan (Nurian dkk., 2024). Melalui pola sentimen yang diperoleh dari ulasan pengguna, pengembang aplikasi dapat mengetahui kebutuhan yang belum terpenuhi, kelemahan sistem, serta fitur yang paling dihargai. Dengan demikian, analisis sentimen dapat menjadi dasar dalam penyusunan strategi peningkatan kualitas layanan agar lebih responsif terhadap kebutuhan masyarakat.

2.1.3 *Text Preprocessing*

Text preprocessing merupakan tahapan awal yang sangat penting dalam analisis teks, khususnya pada penelitian analisis sentimen. Tahap ini berfungsi untuk menyiapkan data mentah agar lebih bersih, konsisten, dan terstruktur sehingga dapat diolah secara matematis oleh algoritma machine learning. Teks mentah umumnya masih mengandung variasi penulisan, tanda baca, kata tidak baku, serta karakter khusus yang dapat mengganggu proses pembelajaran model (Bhakti dkk., 2024). Melalui tahapan *preprocessing*, informasi yang relevan dapat dipertahankan secara representatif, sementara elemen-elemen yang tidak memiliki nilai analisis dapat dihilangkan. Dengan demikian, *preprocessing* menjadi fondasi utama dalam menghasilkan data teks yang berkualitas tinggi sebelum dilakukan tahap representasi fitur dan klasifikasi.

Dalam praktik modern, seluruh tahapan *text preprocessing* biasanya dikelompokkan ke dalam suatu *pipeline*. *Pipeline* ini memungkinkan serangkaian proses seperti *case folding*, *text normalization*, *stopword removal*, *stemming*, dan *cleaning dataset* dijalankan secara berurutan dan konsisten. Keunggulan penggunaan *pipeline* adalah kemampuannya dalam menjaga integritas proses transformasi data sekaligus mencegah terjadinya *data leakage* atau kebocoran informasi antara data latih (*training set*) dan data uji (*test set*) (Ulya dkk., 2025). Misalnya, proses vektorisasi teks seperti *Term Frequency–Inverse Document Frequency* (TF-IDF) atau reduksi dimensi hanya boleh dipelajari dari data latih dan diterapkan pada data uji. Dengan demikian, *pipeline* memastikan setiap transformasi dilakukan dengan cara yang terkontrol dalam *loop* sehingga hasil pelatihan dan evaluasi tetap objektif (Ulya dkk., 2025). Berikut tahapan dari *text preprocessing*:

1. Pembersihan Karakter (*Character Cleaning*), yaitu proses menghilangkan elemen teks yang tidak relevan atau tidak memiliki makna penting dalam analisis. Tahap ini mencakup penghapusan karakter non-alfanumerik seperti tanda baca, simbol, dan emotikon yang biasanya tidak memberikan kontribusi signifikan dalam proses representasi fitur.
2. *Case Folding*, yaitu proses mengubah seluruh huruf dalam teks menjadi huruf kecil (*lowercase*) (Bourequat & Mourad, 2021).
3. *Text Normalization*, yaitu menyamakan bentuk kata agar konsisten, terutama pada teks yang mengandung kata tidak baku, singkatan, atau istilah slang (Burrhanuddin & Fauzan Rozi, 2025).

4. *Stopwords Removal*, yaitu menghapus kata-kata umum yang tidak memiliki makna penting dalam analisis sentimen (Ulya dkk., 2025).
5. *Stemming*, yaitu proses mengubah kata berimbuhan menjadi bentuk dasarnya agar kata yang memiliki makna sama tidak dianggap berbeda oleh model (Burrhanuddin & Fauzan Rozi, 2025).
6. *Cleaning Data*, yaitu proses memastikan tidak terdapat entri kosong atau teks yang hilang akibat proses preprocessing sebelumnya (Bhakti dkk., 2024).

Secara keseluruhan, tahapan *preprocessing* yang terdiri dari *case folding*, *text normalization*, *stopwords removal*, Pembersihan Karakter (*Character Cleaning*), *stemming*, dan *cleaning data* merupakan fondasi yang penting untuk memastikan data teks berkualitas baik. Tahap ini tidak hanya membantu meningkatkan performa algoritma klasifikasi, tetapi juga memastikan bahwa pola sentimen dapat ditangkap dengan lebih akurat dalam proses analisis.

2.1.4 *Machine learning* (ML)

Machine learning (ML) merupakan salah satu cabang dari kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan membuat keputusan tanpa harus diprogram secara *eksplisit* setiap kali (Wijoyo A dkk., 2024). Teknologi ini bekerja dengan cara mengenali pola dan hubungan dalam kumpulan data yang besar, kemudian menggunakan pengetahuan tersebut untuk melakukan prediksi atau klasifikasi terhadap data baru. Seiring dengan bertambahnya jumlah data yang digunakan, sistem ML mampu meningkatkan akurasi dan efisiensinya secara otomatis. Dengan demikian, komputer tidak hanya menjalankan instruksi statis, tetapi juga mampu menyesuaikan perilakunya

berdasarkan pengalaman yang diperoleh, menjadikan ML sebagai fondasi penting dalam pengembangan sistem cerdas modern yang adaptif dan dinamis (Zaki & Meira, 2020).

Kelebihan utama ML terletak pada kemampuannya untuk belajar secara otomatis dan memperbarui pengetahuan yang dimiliki tanpa perlu pemrograman aturan secara manual. Model dapat menemukan pola tersembunyi dan keterkaitan antar variabel dengan mandiri, sehingga sangat bermanfaat untuk menyelesaikan berbagai permasalahan kompleks di kehidupan nyata. Namun, di balik keunggulannya, terdapat pula kelemahan, terutama terkait ketergantungan model terhadap kualitas data yang digunakan. Data yang mengandung *noise*, *outlier*, atau inkonsistensi dapat menurunkan kinerja model secara signifikan dan menghasilkan prediksi yang tidak akurat.

Secara umum, ML dikelompokkan menjadi tiga bagian besar, yaitu (Roihan dkk., 2020):

1. *Supervised Learning* yaitu metode pembelajaran dengan data berlabel, di mana model dilatih untuk mempelajari hubungan antara input dan output berdasarkan contoh yang telah diketahui. Model ini umum digunakan dalam klasifikasi, seperti prediksi penyakit, atau regresi, seperti prediksi harga rumah.
2. *Unsupervised Learning*, yaitu metode pembelajaran tanpa data berlabel yang bertujuan menemukan struktur atau pola tersembunyi dalam data. Salah satu penerapsan populernya adalah *clustering*, misalnya dalam segmentasi pelanggan atau pengelompokan dokumen.

3. *Reinforcement Learning*, yaitu metode pembelajaran berbasis interaksi antara agen dan lingkungan, di mana agen belajar melalui sistem penghargaan (*reward*) dan hukuman (*penalty*). Model ini banyak diaplikasikan pada bidang robotika, pengembangan game, hingga sistem navigasi otomatis.

Perkembangan ini membuat ML semakin relevan dalam berbagai bidang, mulai dari layanan kesehatan, pendidikan, hingga aplikasi publik digital seperti *Mobile JKN*. Dengan memanfaatkan ML, analisis data dapat dilakukan secara lebih cepat, akurat, dan adaptif, sehingga menghasilkan informasi yang bermanfaat bagi pengambil keputusan.

2.1.5 Algoritma Klasifikasi

Algoritma klasifikasi merupakan salah satu pendekatan utama dalam ML yang digunakan untuk memprediksi label atau kategori dari suatu data berdasarkan nilai fitur-fiturnya (Pramudita dkk., 2025). Tujuan dari klasifikasi adalah membangun model yang mampu membedakan antara kelas-kelas yang berbeda dengan akurasi tinggi, berdasarkan pola dari data latih (*training data*) yang telah diberi label (Hanum dkk., 2024). Proses klasifikasi ini terdiri dari dua tahap utama, yaitu proses pelatihan (*training*) di mana model mempelajari pola dari data yang telah diklasifikasi sebelumnya, dan proses prediksi (*testing*) di mana model digunakan untuk mengklasifikasikan data baru yang belum diketahui labelnya. Algoritma klasifikasi sangat berguna dalam berbagai bidang seperti diagnosis medis, deteksi penipuan, analisis sentimen, pengenalan wajah, dan sistem rekomendasi.

Secara umum, algoritma klasifikasi dibagi menjadi dua jenis berdasarkan jumlah kelas yang ingin diprediksi, yaitu klasifikasi biner dan klasifikasi multikelas (Husin, 2023). Klasifikasi biner hanya memiliki dua kelas, seperti “ya” atau “tidak”, sedangkan klasifikasi multikelas memiliki lebih dari dua kategori yang mungkin, misalnya klasifikasi jenis bunga iris menjadi *setosa*, *versicolor*, dan *virginica*. Beberapa algoritma yang umum digunakan untuk tugas klasifikasi meliputi *K-Nearest Neighbor* (KNN), *Decision Tree*, *Naive Bayes*, *SVM*, *Logistic Regression*, *Random Forest*, dan algoritma *ensemble* seperti *AdaBoost* dan *XGBoost*. Setiap algoritma memiliki kelebihan dan kelemahan masing-masing, tergantung pada karakteristik data, jumlah fitur, kecepatan komputasi, serta kemampuan menangani data tidak seimbang dan noise.

Evaluasi performa dari algoritma klasifikasi dilakukan dengan menggunakan berbagai metrik seperti akurasi, *precision*, *recall*, dan *f1-score*, yang memberikan gambaran tentang seberapa baik model dalam memprediksi label dengan benar (Hanum dkk., 2024). Dalam kasus klasifikasi biner, *confusion matrix* sering digunakan untuk menilai performa dengan menunjukkan jumlah prediksi benar dan salah untuk masing-masing kelas. Untuk meningkatkan performa dan mencegah *overfitting*, berbagai teknik seperti validasi silang CV, regularisasi, dan penanganan data tidak seimbang (misalnya dengan SMOTE atau *undersampling*) dapat diterapkan. Pemilihan algoritma klasifikasi yang sesuai serta proses *tuning* parameter yang tepat sangat berpengaruh terhadap keberhasilan model dalam pengklasifikasian data.

2.1.6 Regresi Logistik

Regresi logistik merupakan salah satu metode statistik yang digunakan untuk menganalisis hubungan antara variabel independen dengan variabel dependen yang bersifat dikotomis (biner), seperti ya/tidak, sukses/gagal, atau positif/negatif (Dichotomous dkk., 2023). Berbeda dengan regresi linier yang digunakan untuk memprediksi variabel dependen berskala numerik, regresi logistik memodelkan probabilitas terjadinya suatu peristiwa dengan menggunakan fungsi logistik (sigmoid) (Djamaris dkk., 2021). Metode ini banyak digunakan dalam berbagai bidang, termasuk kesehatan, pemasaran, dan analisis teks, karena kemampuannya dalam menangani data kategorik secara efektif.

Dalam konteks machine learning, regresi logistik termasuk dalam supervised learning, sebab memerlukan data berlabel untuk membangun model. Model ini menerima sekumpulan fitur sebagai masukan dan menghasilkan nilai probabilitas yang menunjukkan tingkat keyakinan terhadap suatu kelas. Dalam analisis sentimen, khususnya pada penelitian ini, regresi logistik digunakan untuk mengklasifikasikan ulasan pengguna *Mobile* JKN ke dalam kategori positif atau negatif. Setiap ulasan terlebih dahulu direpresentasikan sebagai vektor numerik menggunakan metode TF-IDF, sehingga model dapat mempelajari hubungan antara bobot kata dengan sentimen pengguna.

Misalkan terdapat n fitur dan m sampel data. Setiap sampel ke- i direpresentasikan dalam bentuk vektor kolom berdimensi n :

$$\mathbf{x}^{(i)} = \begin{bmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^n \quad (2.1)$$

Parameter model dilambangkan dengan vektor bobot $\mathbf{w} \in \mathbb{R}^n$ dan sebuah bias $b \in \mathbb{R}$:

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \quad (2.2)$$

serta sebuah bias $b \in \mathbb{R}$. Kombinasi linier antara fitur dan parameter dituliskan sebagai:

$$z = \mathbf{w}^T \mathbf{x}^{(i)} + b \quad (2.3)$$

Dalam regresi logistik standar, probabilitas kelas positif dihitung menggunakan fungsi sigmoid:

$$\sigma(z^{(i)}) = \frac{1}{1 + e^{-z^{(i)}}} \quad (2.4)$$

Namun dalam penelitian ini, digunakan modifikasi fungsi sigmoid dengan parameter temperatur $\alpha > 0$, sehingga probabilitas prediksi menjadi:

$$p^{(i)} = \sigma(\alpha z^{(i)}) = \frac{1}{1 + e^{-\alpha z^{(i)}}} \quad (2.5)$$

dengan $z = \mathbf{w}^T \mathbf{x} + b$ (Bishop, 2006). Penskalaan $z \rightarrow \alpha z$ tidak mengubah struktur model, melainkan hanya memodifikasi kemiringan kurva sigmoid melalui transformasi pada sumbu horizontal. Dari sudut pandang statistik, regresi logistik termasuk dalam kerangka *generalized linear model* (GLM), di mana fungsi logit bertindak sebagai *link function* yang memetakan kombinasi linear prediktor ke probabilitas (McCullagh & Nelder, 1989). Dalam kerangka tersebut, penskalaan pada argumen fungsi logistik tetap konsisten secara teoritis karena tidak mengubah bentuk distribusi dasarnya.

Parameter α berfungsi sebagai faktor pengatur sensitivitas probabilitas terhadap perubahan logit. Ketika $\alpha > 1$, kurva sigmoid menjadi lebih curam

sehingga perubahan kecil pada nilai logit menghasilkan perubahan probabilitas yang lebih signifikan. Sebaliknya, jika $0 < \alpha < 1$, kurva menjadi lebih landai dan probabilitas berubah secara lebih gradual. Ketika $\alpha = 1$, formulasi kembali pada bentuk regresi logistik standar. Konsep penskalaan logit ini juga sejalan dengan pendekatan *temperature scaling* dalam literatur kalibrasi probabilitas, di mana logit dikalikan atau dibagi oleh suatu konstanta sebelum diterapkan fungsi sigmoid atau *softmax* (Guo dkk., 2017).

Dalam konteks klasifikasi biner, probabilitas hasil prediksi dibandingkan dengan ambang batas (threshold) 0.5 untuk menentukan kelas. Jika $p^{(i)} \geq 0.5$, maka data diklasifikasikan sebagai kelas positif, sedangkan jika $p^{(i)} < 0.5$, maka diklasifikasikan sebagai kelas negatif.

$$\hat{y}^{(i)} = \begin{cases} 1, & p^{(i)} \geq 0.5 \\ 0, & p^{(i)} < 0.5 \end{cases}$$

Selain pendekatan melalui fungsi *sigmoid*, regresi logistik juga dapat ditinjau melalui fungsi logit (Bishop, 2006). Fungsi logit menyatakan hubungan linier antara kombinasi variabel bebas dengan logaritma peluang (*odds*) terjadinya suatu kejadian. *Odds* merupakan perbandingan antara probabilitas sukses dengan probabilitas gagal. Dengan demikian, regresi logistik dapat dilihat sebagai model linier terhadap log odds, bukan terhadap probabilitas langsung. Persamaan fungsi logit dituliskan sebagai:

$$\text{logit}(p^{(i)}) = \ln\left(\frac{p^{(i)}}{1-p^{(i)}}\right) = \alpha z^{(i)} = \alpha(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \quad (2.6)$$

Dengan merepresentasikan hubungan antara probabilitas dan fungsi logit, dari Persamaan (2.5) diperoleh:

$$\frac{p^{(i)}}{1 - p^{(i)}} = e^{\alpha(\mathbf{w}^\top \mathbf{x}^{(i)} + b)} \quad (2.7)$$

Selanjutnya, dengan mengalikan kedua ruas dengan $(1 - p^{(i)})$ dan menyusun kembali, didapat:

$$p^{(i)} = \frac{e^{e^{\alpha(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}}}{1 + e^{e^{\alpha(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}}} \quad (2.8)$$

Untuk mempelajari parameter \mathbf{w} dan b , regresi logistik menggunakan pendekatan *Maximum Likelihood Estimation* (MLE). Ide dasarnya adalah mencari parameter yang memaksimalkan peluang (*likelihood*) bahwa model yang dibangun sesuai dengan data yang diamati. Fungsi *likelihood* didefinisikan sebagai:

$$L(\mathbf{w}, b) = \prod_{i=1}^m (p^{(i)})^{y^{(i)}} \cdot (1 - p^{(i)})^{(1-y^{(i)})} \quad (2.9)$$

Untuk mempermudah perhitungan, dilakukan transformasi ke bentuk *log-likelihood*:

$$\ell(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^m [y^{(i)} \ln(p^{(i)}) + (1 - y^{(i)}) \ln(1 - p^{(i)})] \quad (2.10)$$

Tujuan pelatihan model adalah menemukan parameter yang memaksimalkan *log-likelihood* ini.

Dalam implementasi *machine learning modern*, fungsi objektif biasanya diformulasikan sebagai *loss function* yang hendak diminimalkan. Fungsi *loss* yang digunakan dalam regresi logistik adalah *Binary Cross-Entropy* atau *Log Loss*, yang merupakan negatif dari *log-likelihood* rata-rata:

$$J(\mathbf{w}, b) = -\ell(\mathbf{w}, b) \quad (2.11)$$

$$= -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \ln(p^{(i)}) + (1 - y^{(i)}) \ln(1 - p^{(i)})]$$

dengan $y^{(i)} \in \{0,1\}$ merupakan label sebenarnya dan $p^{(i)}$ adalah probabilitas hasil prediksi (Zaki & Meira, 2020).

Turunan fungsi *loss* terhadap setiap bobot w_j dilakukan menggunakan aturan rantai (*chain rule*). Untuk setiap sampel ke- i , komponen fungsi *loss* dapat dituliskan sebagai:

$$J_i = -[y^{(i)} \ln(p^{(i)}) + (1 - y^{(i)}) \ln(1 - p^{(i)})] \quad (2.12)$$

Selanjutnya, turunan J_i terhadap bobot w_j diberikan oleh:

$$\frac{\partial J_i}{\partial w_j} = - \left[y^{(i)} \frac{\partial \ln(p^{(i)})}{\partial w_j} + (1 - y^{(i)}) \frac{\partial \ln(1 - p^{(i)})}{\partial w_j} \right] \quad (2.13)$$

Kedua turunan di dalam tanda kurung dihitung dengan menerapkan aturan rantai bertingkat, yaitu menurunkan fungsi logaritma terhadap probabilitas prediksi, kemudian terhadap skor linier $z^{(i)}$, dan terakhir terhadap bobot w_j . Untuk bagian pertama, diperoleh:

$$\begin{aligned} \frac{\partial \ln(p^{(i)})}{\partial w_j} &= \frac{\partial \ln(p^{(i)})}{\partial p^{(i)}} \cdot \frac{\partial p^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial w_j} \\ &= \frac{1}{p^{(i)}} \cdot \alpha p^{(i)} (1 - p^{(i)}) \cdot x_j^{(i)} \\ &= \alpha (1 - p^{(i)}) \cdot x_j^{(i)} \end{aligned}$$

Dengan cara yang sama, turunan untuk komponen kedua diperoleh dari:

$$\begin{aligned} \frac{\partial \ln(1 - p^{(i)})}{\partial w_j} &= \frac{\partial \ln(1 - p^{(i)})}{\partial p^{(i)}} \cdot \frac{\partial p^{(i)}}{\partial z^{(i)}} \cdot \frac{\partial z^{(i)}}{\partial w_j} \\ &= -\frac{1}{1 - p^{(i)}} \cdot \alpha p^{(i)} (1 - p^{(i)}) \cdot x_j^{(i)} \end{aligned}$$

$$= -\alpha p^{(i)} x_j^{(i)}$$

Kedua hasil tersebut kemudian disubstitusikan kembali ke turunan awal, sehingga:

$$\frac{\partial J_i}{\partial w_j} = -[y^{(i)} \alpha (1 - p^{(i)}) - (1 - y^{(i)}) \alpha p^{(i)}] x_j^{(i)} = \alpha (p^{(i)} - y^{(i)}) x_j^{(i)}$$

Persamaan di atas dapat disederhanakan menjadi bentuk umum:

$$\frac{\partial J_i}{\partial w_j} = \alpha (p^{(i)} - y^{(i)}) x_j^{(i)}$$

Turunan ini menyatakan bahwa kontribusi error dari satu sampel terhadap bobot ke- j bergantung pada selisih probabilitas prediksi dengan label sebenarnya, dikalikan nilai fitur pada dimensi tersebut. Karena bobot w berbentuk vektor berdimensi n , maka gradien *loss* untuk satu sampel dapat dituliskan sebagai vektor:

$$\nabla J_i = \begin{bmatrix} (p^{(i)} - y^{(i)}) x_1^{(i)} \\ (p^{(i)} - y^{(i)}) x_2^{(i)} \\ \vdots \\ (p^{(i)} - y^{(i)}) x_n^{(i)} \end{bmatrix}$$

Dalam pembelajaran menggunakan *batch* berukuran m , gradien total merupakan rata-rata kontribusi gradien dari seluruh sampel. Dengan demikian, gradien keseluruhan terhadap bobot ke- j diperoleh sebagai:

$$\frac{\partial J}{\partial w_j} = \frac{\alpha}{m} \sum_{i=1}^m (p^{(i)} - y^{(i)}) x_j^{(i)} \quad (2.15)$$

Turunan terhadap bias b dapat dihitung dengan cara serupa, karena $\frac{\partial z^{(i)}}{\partial b} = 1$, sehingga diperoleh:

$$\frac{\partial J}{\partial b} = \frac{\alpha}{m} \sum_{i=1}^m (p^{(i)} - y^{(i)}) \quad (2.16)$$

Dengan gradien pada Persamaan (2.15) dan (2.16), pembaruan parameter dilakukan menggunakan learning rate η :

$$w_j = w_j - \eta \frac{\partial J}{\partial w_j}, \quad b = b - \eta \frac{\partial J}{\partial b} \quad (2.17)$$

Aturan ini memastikan bahwa parameter bergerak menuju arah yang menurunkan *loss*, sehingga model secara bertahap belajar menghasilkan prediksi yang lebih akurat.

Dalam konteks aplikasi modern, Regresi Logistik juga banyak digunakan dalam bidang *machine learning*, terutama sebagai metode klasifikasi dasar dalam *supervised learning*. Model ini menghasilkan *output* probabilitas yang dapat dikonversi menjadi label sentimen dengan menggunakan *threshold* tertentu, umumnya 0.5, tetapi dapat disesuaikan sesuai kebutuhan untuk menyeimbangkan presisi dan *recall*. Selain itu, regresi logistik juga memiliki pengembangan lanjutan, seperti *regularized Regresi Logistik* (L1, L2) untuk mencegah *overfitting* pada data berdimensi tinggi (Witten dkk., 2017).

2.1.7 *Term Frequency–Inverse Document Frequency* (TF-IDF)

Term Frequency–Inverse Document Frequency (TF-IDF) merupakan salah satu metode representasi teks yang paling banyak digunakan dalam analisis teks dan *text mining*. Teknik ini digunakan untuk mengubah teks tidak terstruktur menjadi vektor numerik sehingga dapat diproses oleh algoritma ML. Inti dari TF-IDF adalah memberikan bobot pada setiap kata dalam dokumen berdasarkan frekuensinya di dalam dokumen tertentu serta seberapa jarang kata tersebut muncul di seluruh koleksi dokumen (Bourequat & Mourad, 2021). Dengan

pendekatan ini, kata-kata yang umum muncul di hampir semua dokumen (seperti “dan”, “atau”, “yang”) akan diberi bobot lebih rendah, sedangkan kata yang jarang muncul tetapi penting bagi dokumen tertentu akan diberi bobot lebih tinggi.

Secara matematis, *Term Frequency–Inverse Document Frequency* (TF-IDF) merupakan hasil perkalian antara dua komponen utama, yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Nilai TF menunjukkan seberapa sering sebuah kata muncul dalam suatu dokumen dibandingkan dengan jumlah total kata dalam dokumen tersebut, sedangkan IDF digunakan untuk mengukur tingkat kepentingan kata terhadap seluruh kumpulan dokumen dengan memberikan bobot rendah pada kata-kata yang sering muncul di banyak dokumen. Konsep ini pertama kali diperkenalkan oleh Salton & Buckley (1988) sebagai salah satu pendekatan *term weighting* dalam bidang *information retrieval*.. Rumus umum TF-IDF dituliskan sebagai berikut:

$$TF(t, d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \quad (2.18)$$

$$IDF(t, D) = \log \frac{N}{1 + |\{d \in D: t \in d\}|} \quad (2.19)$$

$$TF\ IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (2.20)$$

Pada rumus (2.6), (2.7), dan (2.8), simbol t menyatakan sebuah term atau kata tertentu, sedangkan d menunjukkan sebuah dokumen tunggal, misalnya satu ulasan pengguna. Himpunan seluruh dokumen dituliskan dengan D yang merepresentasikan koleksi lengkap dokumen yang dianalisis, misalnya semua ulasan yang dikumpulkan. Nilai $f_{t,d}$ menggambarkan jumlah kemunculan term t dalam dokumen d , sedangkan $\sum_k f_{k,d}$ menunjukkan total frekuensi seluruh term yang terdapat pada dokumen d . Selain itu, N merepresentasikan jumlah total

dokumen dalam koleksi D , sementara $|\{d \in D: t \in d\}|$ menyatakan banyaknya dokumen dalam koleksi D yang mengandung term t .

Sehingga, notasi (t, d) pada $TF(t, d)$ menunjukkan hubungan antara term dengan sebuah dokumen tunggal, notasi (t, D) pada $IDF(t, D)$ menunjukkan hubungan antara term dengan keseluruhan koleksi dokumen, sedangkan notasi (t, d, D) pada $TFIDF(t, d, D)$ menunjukkan bobot akhir suatu term t dalam dokumen d dengan mempertimbangkan seluruh koleksi dokumen D .

Penggunaan TF-IDF memiliki beberapa kelebihan, antara lain kesederhanaan perhitungan, interpretabilitas bobot, dan efektivitas dalam mengidentifikasi kata-kata penting (Wilson & Anwar, 2024). Pada analisis sentimen, TF-IDF dapat digunakan untuk merepresentasikan teks ulasan pengguna dalam bentuk matriks fitur yang dapat langsung diproses oleh algoritma klasifikasi seperti Regresi Logistik atau *Support Vector Machine*. Misalnya, kata “bagus” pada ulasan aplikasi kesehatan akan mendapat bobot tinggi jika sering muncul di ulasan positif tetapi jarang muncul di ulasan negatif. Dengan demikian, bobot TF-IDF membantu model klasifikasi dalam membedakan ulasan positif dan negatif berdasarkan kata-kata dominan.

2.1.8 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu metode yang digunakan untuk mengatasi permasalahan ketidakseimbangan kelas (*class imbalance*) pada data (Sajiwo dkk., 2024). Ketidakseimbangan kelas sering terjadi pada masalah klasifikasi, termasuk analisis sentimen, di mana jumlah data pada kelas positif dan negatif tidak seimbang. Kondisi ini dapat menyebabkan model cenderung bias terhadap kelas mayoritas, sehingga akurasi

tinggi yang ditunjukkan model seringkali menyesatkan karena performa pada kelas minoritas sangat rendah. Metode ini melakukan *oversampling* pada kelas minoritas bukan melalui duplikasi data, melainkan dengan membentuk *sampel sintetis* baru yang dihasilkan dari hubungan kedekatan antar sampel minoritas dalam ruang fitur. Sampel sintetis tersebut muncul melalui proses interpolasi linier, sehingga pola baru yang tercipta tetap berada dalam distribusi alami kelas minoritas.

Metode SMOTE dikembangkan oleh Chawla dkk., (2002) sebagai solusi untuk mengatasi permasalahan *imbalanced dataset* pada klasifikasi. Algoritma ini bekerja dengan cara menghasilkan sampel sintetis baru dari kelas minoritas melalui interpolasi antara satu sampel minoritas dan salah satu tetangga terdekatnya (*k-nearest neighbors*). Dalam konteks penelitian ini, istilah *sampel* merujuk pada representasi numerik ulasan yang telah diubah menjadi vektor fitur TF-IDF berdimensi n . Dengan demikian, setiap x_i dan x_j bukan berupa teks mentah, melainkan vektor TF-IDF yang memungkinkan pengukuran jarak secara matematis dalam ruang fitur.

Untuk mencari salah satu tetangga terdekat, digunakan perhitungan jarak *Euclidean* antara sampel minoritas x_i dan setiap sampel minoritas lain x_j menggunakan rumus:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^n (x_{im} - x_{jm})^2} \quad (2.21)$$

Dengan:

$d(x_i, x_j)$: Jarak Euclidean yang dihitung dalam ruang fitur TF-IDF

x_i : Sampel minoritas yang sedang diproses

- x_j : Sampel minoritas lain yang dibandingkan dengan x_i
 x_{im} : Nilai fitur ke- m dari sampel x_i
 x_{jm} : Nilai fitur ke- m dari sampel x_j
 n : Banyaknya fitur (misalnya fitur hasil TF-IDF).

Nilai x_{im} dan x_{jm} tersebut merupakan bobot TF-IDF dari kata tertentu pada masing-masing dokumen, sehingga jarak *Euclidean* mengukur tingkat kemiripan antar ulasan berdasarkan pola kata yang terkandung di dalamnya.

Setelah tetangga terdekat ditentukan, sampel sintetis baru (x_{new}) dihitung menggunakan rumus interpolasi berikut:

$$x_{new} = x_i + \delta \times (x_{zi} - x_i) \quad (2.22)$$

- x_i : Sampel asli dari kelas minoritas,
 x_{zi} : Tetangga terdekat yang akhirnya dipilih untuk interpolasi
 δ : Bilangan acak yang nilainya berada pada interval $[0,1]$.

Pada penelitian ini, bentuk x_{new} berupa vektor fitur TF-IDF sintetis yang posisinya berada di antara dua vektor asli (x_i dan x_{zi}). Vektor ini tidak menghasilkan teks baru, tetapi menciptakan titik baru dalam ruang fitur yang mengikuti pola distribusi kelas minoritas. Proses interpolasi ini diulang hingga jumlah sampel pada kelas minoritas mencapai proporsi yang seimbang dengan kelas mayoritas.

2.1.9 Metrik Evaluasi

Metrik evaluasi merupakan alat ukur yang digunakan untuk menilai performa suatu model klasifikasi, termasuk pada penelitian analisis sentimen. Dalam konteks Regresi Logistik, model menghasilkan prediksi dalam bentuk probabilitas yang kemudian dikonversi menjadi label menggunakan threshold

tertentu. Untuk mengetahui sejauh mana model mampu membedakan kelas positif dan negatif, diperlukan pengukuran kuantitatif melalui metrik evaluasi. Pemilihan metrik yang tepat sangat penting, karena hasil evaluasi akan memengaruhi interpretasi kualitas model dan pengambilan keputusan yang berbasis pada hasil klasifikasi.

Beberapa metrik evaluasi yang umum digunakan dalam klasifikasi biner adalah akurasi, presisi, *recall*, dan *F1-score*. Akurasi mengukur proporsi prediksi yang benar dibandingkan dengan seluruh prediksi, namun metrik ini seringkali menyesatkan jika data tidak seimbang (Bhakti dkk., 2024). Presisi digunakan untuk mengukur seberapa banyak prediksi positif yang benar-benar positif, sedangkan *recall* mengukur seberapa banyak data positif yang berhasil diidentifikasi oleh model. *F1-score* kemudian hadir sebagai metrik yang menggabungkan presisi dan *recall* dalam satu nilai harmonis, sehingga lebih representatif ketika data tidak seimbang (Sausan dkk., 2024).

Secara matematis, metrik evaluasi tersebut dapat dituliskan sebagai berikut (Powers, 2011):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.23)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.24)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.25)$$

$$F1 \text{ score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.26)$$

Dengan *TP* (*True Positive*), *TN* (*True Negative*), *FP* (*False Positive*), dan *FN* (*False Negative*). Rumus di atas membantu peneliti untuk memahami kelebihan dan kekurangan model dari berbagai sudut pandang performa.

Seiring berkembangnya metode evaluasi, penelitian klasifikasi modern termasuk analisis sentimen juga menggunakan metrik rata-rata seperti *macro average* dan *weighted average* untuk mendapatkan gambaran performa yang lebih komprehensif (Widagdo dkk., 2024). *Macro average* menghitung nilai rata-rata dari presisi, *recall*, atau *F1-score* pada setiap kelas tanpa mempertimbangkan jumlah sampel tiap kelas, sehingga setiap kelas diperlakukan memiliki bobot yang sama. Metrik ini sangat berguna untuk melihat keseimbangan performa ketika data tidak seimbang. Secara matematis, *macro average* dirumuskan sebagai berikut:

$$\text{Macro Avg} = \frac{\sum_{i=1}^k M_i}{k} \quad (2.27)$$

dengan M_i merupakan nilai metrik (presisi, *recall*, atau *F1-score*) pada kelas ke- i .

Di sisi lain, *weighted average* memberikan nilai rata-rata dari presisi, *recall*, atau *F1-score* dengan mempertimbangkan jumlah sampel pada masing-masing kelas sehingga kelas yang memiliki jumlah data lebih banyak akan memberikan kontribusi lebih besar terhadap nilai akhir. Metrik ini memberikan gambaran yang lebih merepresentasikan performa keseluruhan dataset, terutama ketika terdapat perbedaan signifikan dalam distribusi kelas. Rumus *weighted average* dituliskan sebagai berikut:

$$\text{Weighted Avg} = \frac{\sum_{i=1}^k (M_i \times n_i)}{\sum_{i=1}^k n_i} \quad (2.28)$$

dengan n_i adalah jumlah sampel pada kelas ke- i . Kedua metrik tambahan ini membantu memastikan bahwa evaluasi model tidak hanya mempertimbangkan performa setiap kelas secara individual, tetapi juga relevansinya terhadap struktur dataset secara keseluruhan.

Dalam penelitian analisis sentimen, pemilihan metrik evaluasi harus disesuaikan dengan tujuan. Misalnya, jika tujuan penelitian adalah untuk mengidentifikasi keluhan pengguna aplikasi *Mobile JKN* secara tepat, maka *recall* pada kelas negatif menjadi lebih penting agar tidak ada keluhan yang terlewat. Sebaliknya, jika fokus penelitian adalah memastikan opini positif benar-benar akurat, maka presisi pada kelas positif lebih diperhatikan. Dengan demikian, metrik evaluasi tidak hanya digunakan sebagai indikator teknis, tetapi juga sebagai dasar strategi untuk peningkatan kualitas layanan yang berbasis pada opini pengguna.

2.1.10 *Mobile JKN*

Aplikasi *Mobile JKN* merupakan inovasi digital dari BPJS Kesehatan yang dirancang untuk mendukung implementasi Program Jaminan Kesehatan Nasional (JKN) melalui pemanfaatan teknologi informasi. Aplikasi ini dapat diakses melalui perangkat *smartphone* dan menyediakan berbagai fitur yang memudahkan peserta dalam mengakses layanan kesehatan secara mandiri. Beberapa fitur utama meliputi pendaftaran peserta, perubahan data kepesertaan, pengecekan status iuran, informasi fasilitas kesehatan (FKTP dan FKTL), serta pengaduan atau saran layanan (BPJS Kesehatan, 2025). Kehadiran aplikasi ini menjadi salah satu upaya transformasi layanan publik menuju sistem kesehatan yang lebih efisien, transparan, dan responsif terhadap kebutuhan peserta.

Mobile JKN juga merupakan bentuk transformasi dari layanan manual menjadi layanan berbasis digital yang sederhana, cepat, dan efektif. Transformasi ini tidak hanya memudahkan pengguna dalam mengakses layanan BPJS Kesehatan, tetapi juga mempercepat proses administrasi serta meningkatkan efisiensi pelayanan. Salah satu inovasi penting adalah ketersediaan identitas peserta digital, sehingga peserta tidak lagi wajib membawa kartu fisik untuk mendapatkan layanan kesehatan. Aplikasi ini juga dilengkapi dengan fitur *Mobile Screening* dan *Care Center*, yang memungkinkan peserta melakukan deteksi dini kondisi kesehatan serta berkonsultasi secara *daring* dengan dokter. Dengan demikian, *Mobile JKN* mendukung percepatan tercapainya UHC melalui pendekatan layanan berbasis teknologi (Suhadi dkk., 2022).

Secara keseluruhan, *Mobile JKN* mencerminkan komitmen BPJS Kesehatan dalam menghadirkan layanan kesehatan yang lebih responsif, efisien, dan mudah diakses oleh masyarakat luas. Namun, efektivitas penggunaannya masih dipengaruhi oleh faktor-faktor seperti literasi digital peserta, ketersediaan jaringan internet, serta persepsi dan kepuasan mereka terhadap pengalaman penggunaan aplikasi. Ulasan yang diberikan pengguna di platform digital, misalnya *Google Play Store*, menjadi sumber data penting untuk memahami pengalaman nyata masyarakat dalam memanfaatkan aplikasi ini. Oleh karena itu, analisis sentimen terhadap ulasan pengguna menjadi langkah penting untuk mengevaluasi kualitas layanan, mengidentifikasi kelemahan, serta memberikan rekomendasi pengembangan *Mobile JKN* di masa mendatang.

2.2 Kajian Integrasi Topik dengan Alquran/Hadis

Penelitian mengenai analisis sentimen dengan *machine learning* tidak hanya relevan dalam konteks teknologi, pelayanan publik, dan peningkatan kualitas sistem informasi, tetapi juga dapat dikaji dari perspektif Islam. Dalam ajaran Islam, ilmu pengetahuan dan teknologi merupakan karunia Allah SWT yang wajib dimanfaatkan untuk menghadirkan kemaslahatan bagi umat manusia. *Machine learning*, sebagai cabang kecerdasan buatan, mampu mengolah ribuan data secara cepat dan akurat sehingga dapat membantu pengambil kebijakan dalam memahami kebutuhan dan aspirasi masyarakat. Dengan kemampuan tersebut, penerapan *machine learning* dapat dipandang sebagai wujud nyata dari prinsip Islam tentang tolong-menolong dalam kebaikan serta kebermanfaatan ilmu yang membawa masalah luas.

Al-Qur'an memberikan landasan yang kuat mengenai pentingnya nilai kolaborasi dan kebermanfaatan. Allah SWT berfirman dalam Surah Al-Māidah ayat 2:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَحْلُوا شَعَائِرَ اللَّهِ وَلَا الشَّهْرَ الْحَرَامَ وَلَا الْهَدْيَ وَلَا الْقَلَائِدَ وَلَا آمِينَ الْبَيْتِ الْحَرَامِ يَبْتَغُونَ فَضْلًا مِّن رَّبِّهِمْ وَرِضْوَانًا وَإِذَا حَلَلْتُمْ فَاصْطَادُوا وَلَا يَجْرِمَنَّكُمْ شَنَا نُ فَؤُومٍ أَنْ صَدُّوكُمْ عَنِ الْمَسْجِدِ الْحَرَامِ أَنْ تَعْتَدُوا وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ ﴿٢﴾

“Wahai orang-orang yang beriman, janganlah kamu melanggar syiar-syiar (kesucian) Allah, jangan (melanggar kehormatan) bulan-bulan haram, jangan (mengganggu) hadyu (hewan-hewan kurban) dan qalā'id (hewan-hewan kurban yang diberi tanda), dan jangan (pula mengganggu) para pengunjung Baitulharam sedangkan mereka mencari karunia dan rida Tuhannya! Apabila kamu telah bertahalul (menyelesaikan ihram), berburulah (jika mau). Janganlah sekali-kali kebencian(-mu) kepada suatu kaum, karena mereka menghalang-halangi dari Masjidilharam, mendorongmu berbuat melampaui batas (kepada mereka). Tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan permusuhan. Bertakwalah kepada Allah, sesungguhnya Allah sangat berat siksaan-Nya.” (NU Online, 2025)

Ayat tersebut menegaskan bahwa kerja sama yang berorientasi pada kemaslahatan umat merupakan bagian dari amal kebajikan yang bernilai ibadah. Implementasi *machine learning* dalam bidang sosial dan pelayanan publik, misalnya analisis sentimen, dapat dilihat sebagai bentuk kolaborasi modern yang sejalan dengan nilai tolong-menolong dalam Islam

Adapun *asbābun nuzūl* ayat tersebut diriwayatkan dari Ibn Abbas, bahwa ayat tersebut turun pada peristiwa Hudaibiyah ketika kaum Quraisy menghalangi Nabi Muhammad SAW dan para sahabat untuk masuk ke Mekah. Dalam situasi sulit tersebut, Allah SWT memerintahkan kaum Muslimin untuk tetap saling menolong dalam kebaikan, meski dihadapkan pada permusuhan. Perintah ini mengandung pelajaran bahwa kolaborasi dalam kebajikan tidak boleh berhenti meskipun menghadapi rintangan dari pihak lain. Pesan ini relevan dengan era modern, bahwa teknologi seperti *machine learning* seharusnya dipakai sebagai alat tolong-menolong dalam kebaikan, bukan sebaliknya digunakan untuk kerusakan atau manipulasi..

Dari sisi bahasa, kata *al-birr* berasal dari akar kata بَرَّ yang berarti luas, kebaikan, atau kelapangan. Dalam konteks tafsir Islam, *al-birr* tidak hanya bermakna amal baik yang sederhana, tetapi mencakup seluruh bentuk kebajikan, mulai dari iman, amal saleh, ketaatan, hingga kepedulian sosial. Hal ini berbeda dengan kata *ḥasanah*, yang biasanya digunakan untuk merujuk pada kebaikan tertentu atau spesifik, misalnya doa untuk memperoleh kebaikan dunia dan akhirat. Dengan demikian, *al-birr* memiliki cakupan makna yang lebih komprehensif, sedangkan *ḥasanah* bersifat parsial. Relevansinya dalam *machine learning* adalah bahwa teknologi ini seharusnya diposisikan sebagai instrumen *al-birr*, yaitu

membawa kemaslahatan yang luas, bukan hanya sebatas kebaikan kecil yang sesaat.

Selain Al-Qur'an, hadis Nabi Muhammad SAW juga menggarisbawahi pentingnya kebermanfaatn bagi sesama. Rasulullah SAW bersabda (Al-Ṭabarānī, 2012):

عَنْ جَابِرِ بْنِ عَبْدِ اللَّهِ رَضِيَ اللَّهُ عَنْهُ، قَالَ: قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ: الْمُؤْمِنُ يَأْلَفُ وَيُؤْلَفُ، وَلَا خَيْرَ فِيمَنْ لَا يَأْلَفُ وَلَا يُؤْلَفُ، وَخَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ (رواه الطبراني في المعجم الأوسط)

“Dari Jabir bin Abdullah RA, ia berkata: Rasulullah SAW bersabda: Seorang mukmin itu mudah bergaul dan disenangi. Tidak ada kebaikan pada orang yang tidak bisa bergaul dan tidak disenangi. Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia lainnya.” (HR. al-Ṭabarānī dalam al-Mu‘jam al-Awsaṭ, no. 5787; al-Haitsami, Majma‘ al-Zawāid, 8/16).

Hadis tersebut diriwayatkan oleh al-Ṭabarānī melalui jalur sanad Jābir bin ‘Abdillāh RA, dan dinilai hasan karena perawinya tsiqah (terpercaya), sebagaimana dikomentari oleh al-Haitsami dalam *Majma‘ al-Zawāid*. Pesan utama hadis tersebut menekankan bahwa seorang mukmin idealnya memiliki sifat *sociable* (mau berinteraksi dengan baik), serta membawa manfaat nyata bagi orang lain.

Relevansinya dalam konteks *machine learning* adalah bahwa teknologi ini seharusnya menjadi sarana *ta‘āwun* (tolong-menolong) dan menghadirkan manfaat bagi masyarakat luas. Sama seperti mukmin yang sebaik-baiknya adalah yang bermanfaat bagi sesama, maka algoritma, model, dan inovasi *machine learning* pun harus diarahkan agar bisa memberi solusi nyata, bukan sekadar produk teknis. Misalnya, analisis sentimen membantu pemerintah memahami keluhan masyarakat, atau mendukung peningkatan kualitas layanan publik agar lebih adil dan merata.

2.3 Kajian Topik dengan Teori Pendukung

Analisis sentimen merupakan cabang dari *Natural Language Processing* (NLP) yang berfokus pada pengolahan bahasa alami untuk mengidentifikasi,

mengekstraksi, dan mengklasifikasikan *opmach*, sikap, maupun emosi yang terkandung dalam suatu teks. Tujuan dari analisis sentimen adalah untuk memberikan gambaran mengenai persepsi masyarakat terhadap suatu objek, baik berupa produk, layanan, maupun kebijakan. Dalam konteks penelitian ini, analisis sentimen diterapkan pada ulasan pengguna aplikasi *Mobile JKN* yang diperoleh dari *Google Play Store*. Dengan pendekatan ini, dapat diketahui persepsi masyarakat secara lebih objektif berbasis data, sehingga hasilnya dapat digunakan sebagai masukan dalam pengembangan aplikasi.

Dalam melakukan analisis sentimen, representasi teks menjadi salah satu tahap penting karena data teks bersifat tidak terstruktur. Salah satu metode yang banyak digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF bekerja dengan memberikan bobot pada setiap kata berdasarkan frekuensinya dalam dokumen tertentu dibandingkan dengan frekuensinya di seluruh kumpulan dokumen. Kata-kata yang umum muncul, seperti kata sambung, akan mendapat bobot rendah, sedangkan kata yang jarang muncul namun relevan akan mendapat bobot lebih tinggi. Dengan metode ini, data teks yang semula berupa kumpulan kata dapat diubah menjadi vektor numerik yang dapat diproses oleh algoritma *machine learning*.

Setelah teks direpresentasikan ke dalam bentuk numerik, langkah berikutnya adalah melakukan klasifikasi. Salah satu algoritma yang sederhana namun efektif dalam klasifikasi biner adalah Regresi Logistik. Berbeda dengan regresi linier, Regresi Logistik digunakan untuk memodelkan hubungan antara variabel independen dengan variabel dependen yang bersifat dikotomis, misalnya positif dan negatif. Algoritma ini memanfaatkan fungsi logistik atau *sigmoid* untuk memetakan

probabilitas suatu sampel masuk ke dalam kelas tertentu. Kelebihan Regresi Logistik adalah interpretabilitas yang mudah, kompleksitas rendah, serta performa yang baik untuk data dengan jumlah fitur besar seperti teks.

Dalam praktiknya, permasalahan ketidakseimbangan data sering muncul pada analisis sentimen, terutama ketika jumlah ulasan positif jauh lebih banyak dibandingkan ulasan negatif atau sebaliknya. Ketidakseimbangan data dapat menyebabkan model cenderung bias terhadap kelas mayoritas sehingga menurunkan performa klasifikasi. Salah satu solusi yang umum digunakan adalah *Synthetic Minority Over-sampling Technique* (SMOTE). SMOTE bekerja dengan menghasilkan sampel baru pada kelas minoritas melalui interpolasi antar data yang ada, sehingga distribusi kelas menjadi lebih seimbang. Dengan data yang lebih seimbang, model diharapkan mampu mempelajari pola dari kedua kelas secara lebih adil dan meningkatkan performa metrik evaluasi.

Evaluasi model merupakan tahap penting untuk menilai seberapa baik algoritma bekerja dalam klasifikasi sentimen. Beberapa metrik yang sering digunakan antara lain akurasi, presisi, *recall*, dan *F1-score*. Akurasi mengukur proporsi prediksi benar dari seluruh data, presisi mengukur seberapa tepat model dalam memprediksi kelas positif, *recall* menilai seberapa baik model dalam menangkap seluruh data positif, sedangkan *F1-score* merupakan harmonisasi antara presisi dan *recall*. Dengan menggunakan kombinasi metrik ini, performa model dapat dievaluasi secara lebih menyeluruh, tidak hanya berdasarkan prediksi keseluruhan, tetapi juga sensitivitas terhadap data minoritas.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis pendekatan yang diterapkan dalam penelitian ini adalah kuantitatif eksperimental. Pendekatan ini dipilih karena penelitian tidak hanya menggambarkan data secara deskriptif, tetapi juga melibatkan proses pengujian dan perbandingan performa model melalui serangkaian percobaan terkontrol. Data ulasan pengguna aplikasi *Mobile JKN* diolah secara numerik menggunakan representasi fitur TF-IDF dan penyeimbangan data dengan SMOTE, kemudian diuji menggunakan algoritma Regresi Logistik. Melalui pendekatan eksperimental, dilakukan evaluasi terhadap performa model berdasarkan metrik kuantitatif seperti akurasi, presisi, *recall*, dan *F1-score*. Hasil dari eksperimen ini digunakan untuk menilai efektivitas metode yang diterapkan dalam mengklasifikasikan sentimen pengguna secara objektif dan terukur, sehingga dapat memberikan kesimpulan yang valid mengenai kinerja model pada data yang digunakan.

3.2 Data dan Sumber Data

Penelitian ini menggunakan data sekunder, yaitu data yang diperoleh dari sumber yang telah tersedia tanpa melakukan pengumpulan data secara langsung. Data yang digunakan berupa 10.000 ulasan pengguna aplikasi *Mobile JKN* yang bersumber dari *Google Play Store*. Ulasan tersebut dipilih karena bersifat terbuka serta dianggap representatif dalam menggambarkan pengalaman dan persepsi pengguna terhadap layanan yang disediakan oleh BPJS Kesehatan melalui aplikasi *digital*.

Data ulasan terdiri atas teks komentar dan penilaian (*rating*) yang diberikan oleh pengguna. Informasi tersebut digunakan sebagai dasar untuk mengidentifikasi sentimen, baik yang bersifat positif dan negatif. Dalam konteks penelitian ini, teks ulasan diperlakukan sebagai fitur (x) yang direpresentasikan dalam bentuk numerik menggunakan metode TF-IDF, sedangkan label sentimen diperlakukan sebagai kelas (y) yang dikategorikan ke dalam dua kelas, yaitu positif dan negatif.

3.3 Teknik Pengumpulan Data

Data penelitian diperoleh melalui teknik *web scraping*, yaitu pengambilan data secara otomatis dari halaman *web* menggunakan program komputer. Dalam penelitian ini, proses *web scraping* dilakukan dengan bahasa pemrograman *Python* di lingkungan *Google Colab* menggunakan pustaka *google-play-scraper* untuk mengunduh ulasan pengguna aplikasi *Mobile JKN* pada *platform Google Play Store*. Secara sistematis, tahapan pengumpulan data dilakukan sebagai berikut:

1. Identifikasi sumber data

Halaman resmi aplikasi *Mobile JKN* pada *Google Play Store* ditetapkan sebagai sumber data utama. Ulasan yang diambil merupakan ulasan publik yang ditulis oleh pengguna aplikasi tersebut.

2. Pengambilan data otomatis dengan *google-play-scraper*

Pengumpulan ulasan dilakukan menggunakan fungsi `reviews()` dari pustaka *google-play-scraper* dengan beberapa parameter, antara lain:

- a. `lang='id'` untuk membatasi ulasan berbahasa Indonesia,
- b. `country='id'` untuk membatasi ulasan dari regional Indonesia,
- c. `sort=Sort.NEWEST` agar ulasan diambil berdasarkan urutan terbaru.

Pengambilan dilakukan secara bertahap (*batch*) dengan ukuran tertentu (misalnya 2.000 ulasan per *batch*) dan memanfaatkan *continuation_token* untuk melanjutkan pengambilan ke halaman berikutnya hingga jumlah ulasan yang diinginkan tercapai. Mekanisme *retry* dengan batas percobaan dan jeda waktu tertentu juga digunakan untuk mengantisipasi kegagalan koneksi atau respons *server*, sehingga proses pengumpulan data menjadi lebih stabil dan andal.

3. Penyaringan dan penentuan jumlah data

Ulasan yang berhasil diunduh kemudian disaring sesuai dengan kebutuhan penelitian. Pada tahap ini dilakukan:

- a. Penghapusan ulasan dengan *rating* 3 agar fokus analisis berada pada sentimen yang cenderung positif (*rating* 4–5) dan negatif (*rating* 1–2),
- b. Penyaringan berdasarkan tahun ulasan sehingga hanya ulasan pada tahun tertentu (misalnya tahun 2025) yang digunakan.

Setelah proses penyaringan, ditetapkan jumlah target 10.000 ulasan tanpa *rating* 3. Jika hasil *scraping* melebihi jumlah tersebut, data dipotong sehingga hanya 10.000 ulasan pertama yang memenuhi kriteria yang digunakan sebagai dataset penelitian.

4. Data ulasan yang telah terkumpul disusun ke dalam bentuk *DataFrame* menggunakan pustaka *Pandas*. Dari struktur data lengkap hasil *google-play-scraping*, dipilih atribut yang relevan untuk analisis sentimen, antara lain:

- a. *score* (*rating* pengguna),
- b. *at* (waktu ulasan dibuat),
- c. *content* (teks ulasan).

Pemilihan atribut ini dimaksudkan agar dataset lebih terfokus pada informasi yang berkaitan langsung dengan proses pelabelan sentimen, analisis temporal, serta pemrosesan teks pada tahap selanjutnya.

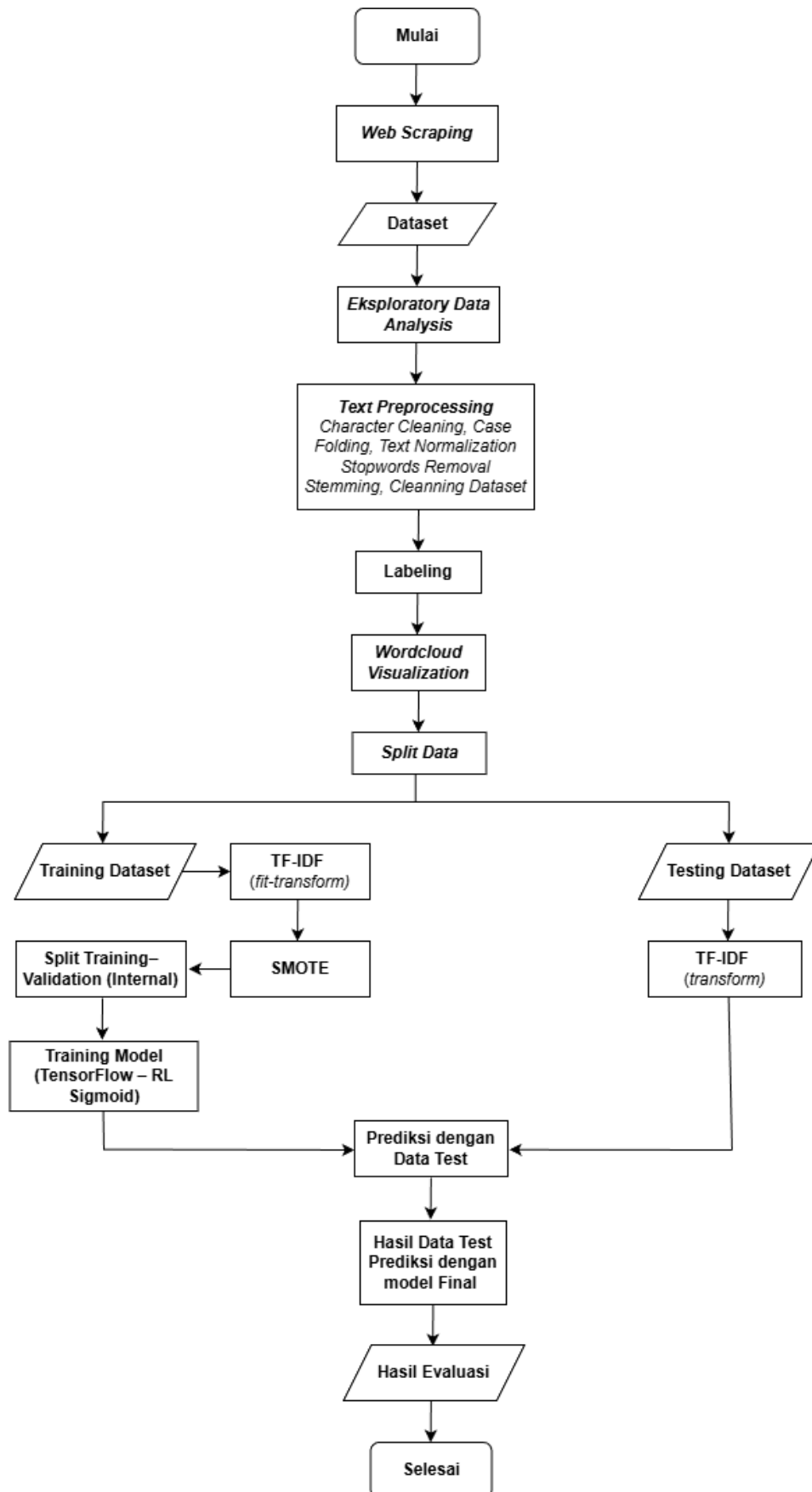
5. Penyimpanan data

Dataset yang telah difilter dan dipilih atributnya kemudian disimpan dalam format CSV. Format ini dipilih karena kompatibel dengan berbagai pustaka analisis data di *Python* dan memudahkan proses pemanggilan data pada tahap *preprocessing*, pembentukan fitur TF-IDF, penerapan SMOTE, serta pelatihan model Regresi Logistik.

Penggunaan teknik *web scraping* dengan *google-play-scraper* ini dipilih karena data ulasan di *Google Play Store* bersifat dinamis dan terus diperbarui, sehingga memungkinkan peneliti memperoleh data yang aktual dalam jumlah besar. Selain itu, pengaturan parameter, penyaringan *rating* (menghapus *rating* 3), serta pembatasan periode waktu menjadikan proses pengumpulan data lebih terstruktur dan sesuai dengan kebutuhan desain eksperimen penelitian.

3.4 Teknik Analisis Data

Teknik analisis data dalam penelitian ini dilakukan melalui beberapa tahapan yang sistematis untuk memastikan kualitas data serta ketepatan hasil analisis. Dalam penelitian analisis polaritas berbasis *rating* ini, seluruh tahapan penelitian disusun secara terstruktur dan divisualisasikan melalui sebuah *flowchart*. *Flowchart* tersebut memberikan gambaran umum mengenai proses yang dilakukan mulai dari pengumpulan data hingga evaluasi model. Alur lengkap penelitian dapat dilihat pada Gambar 3.1.

Gambar 3.1 *Flowchart* Penelitian

3.4.1 *Exploratory Data Analysis (EDA)*

Tahap awal bertujuan untuk memahami karakteristik umum dataset ulasan pengguna aplikasi *Mobile JKN* sebelum dilakukan pemodelan. Analisis ini membantu dalam mengenali struktur data dan menemukan potensi permasalahan seperti ketidakseimbangan kelas atau *missing value*. Langkah-langkahnya meliputi:

1. Mengidentifikasi jumlah total ulasan, distribusi *rating* (1-2, dan 4-5), serta panjang rata-rata teks ulasan.
2. Membuat *visualisasi* sederhana seperti diagram batang atau histogram untuk melihat persebaran data.
3. Menampilkan *wordcloud* awal untuk melihat kata yang paling sering muncul di seluruh dataset.
4. Mengevaluasi keseimbangan jumlah data antara kelas positif dan negatif agar dapat ditangani lebih lanjut pada tahap penyeimbangan data.

3.4.2 *Text Preprocessing*

Proses *preprocessing* dilakukan untuk membersihkan dan menyiapkan data teks agar dapat diolah menjadi representasi numerik yang konsisten. Tahapan ini bertujuan menghapus elemen-elemen yang tidak relevan, menstandarkan format penulisan, serta menyamakan struktur teks sehingga dapat diolah oleh algoritma pembelajaran mesin. Berdasarkan implementasi dalam kode program, tahap *preprocessing* dilakukan melalui beberapa langkah berikut:

1. Pembersihan Karakter (*Character Cleaning*)

Proses ini dilakukan menggunakan pola *regular expression* `[^a-zA-Z0-9\s]` yang berfungsi menghapus seluruh karakter non-alfanumerik,

termasuk tanda baca, simbol, dan emoji. Setelah penghapusan karakter, dilakukan pula normalisasi spasi dengan mengganti spasi berlebih menjadi satu spasi dan menghapus spasi di awal maupun akhir teks.

2. *Case Folding*

Pada tahap ini seluruh teks ulasan diubah menjadi huruf kecil agar konsisten dalam penulisan. Proses *case folding* ini dilakukan dengan memanfaatkan fungsi bawaan *Python string method* (.lower()).

3. *Text Normalization*

Dilakukan menggunakan kamus normalisasi kata yang diunduh dari repositori *GitHub* melalui pustaka *Pandas*. Kamus tersebut berisi pasangan kata singkatan dan bentuk bakunya. Setiap kata dalam teks dipisahkan menggunakan NLTK (*word_tokenize*) lalu dibandingkan dengan kamus; jika ditemukan padanannya, kata diganti dengan bentuk baku, sehingga teks menjadi lebih seragam dan siap untuk tahap pengolahan berikutnya.

4. *Stopword Removal*

Pada tahap ini, kata-kata umum yang tidak memiliki makna signifikan dalam analisis sentimen. Proses ini dilakukan menggunakan pustaka Sastrawi (*StopWordRemoverFactory*) dengan daftar *stopword* Bahasa Indonesia, atau menggunakan daftar *stopword* dari NLTK.

5. *Stemming*

Proses *stemming* mengubah kata berimbuhan menjadi bentuk dasar. Dengan demikian, variasi kata dengan makna sama tidak dianggap sebagai

fitur yang berbeda. Tahap *stemming* dilakukan dengan menggunakan *Sastrawi Stemmer* untuk Bahasa Indonesia.

6. *Cleaning Dataset*

Tahap pembersihan akhir dilakukan untuk memastikan tidak ada data kosong atau berisi spasi saja. Baris data yang kosong atau hanya mengandung whitespace dihapus menggunakan fungsi `str.strip()` dan filter `df[df['content'] != ""]`. Jumlah data akhir kemudian dicetak untuk memastikan bahwa dataset siap digunakan.

7. *Wordcloud Visualization*

Visualisasi ini memberikan gambaran awal mengenai pola dan kecenderungan kata dalam setiap kategori sentimen setelah melalui tahap pembersihan teks. Pembuatan dilakukan menggunakan pustaka *WordCloud* dan *Matplotlib* pada *Python*.

3.4.3 *Split Data*

Dataset yang telah direpresentasikan kemudian dipisah menjadi data latih dan data uji menggunakan pendekatan per kelas. Data positif dan negatif terlebih dahulu dipisahkan, kemudian masing-masing kelas dibagi dengan rasio 80:20 menggunakan fungsi *train_test_split*. Dengan cara ini, 80% data positif dan 80% data negatif menjadi data latih, sedangkan masing-masing 20% sisanya menjadi data uji. Setelah proses pemisahan, bagian latih dari kedua kelas digabungkan kembali menjadi satu himpunan data latih, demikian pula bagian ujinya digabungkan menjadi satu himpunan data uji. Skema ini memastikan bahwa baik data latih maupun data uji mempertahankan porsi asli masing-masing kelas.

3.4.4 Pelabelan

Proses pelabelan dilakukan untuk mengubah *rating* pengguna menjadi label sentimen yang akan digunakan dalam proses klasifikasi.

1. *Rating* 1–2 dikategorikan sebagai sentimen negatif, sedangkan *rating* 4–5 dikategorikan sebagai sentimen positif.
2. *Rating* 3 otomatis tidak ikut terambil pada proses *web scraping*, sehingga seluruh dataset (10.000 ulasan) tanpa ada *rating* 3.
3. Hasil pelabelan menghasilkan dua variabel utama yaitu x teks ulasan sebagai fitur (variabel independen), dan y label sentimen sebagai kelas (variabel dependen). Hubungan antara x dan y sesuai dengan formulasi matematis pada Persamaan (2.1)–(2.5), di mana regresi logistik digunakan untuk memodelkan probabilitas sentimen melalui fungsi sigmoid.

3.4.5 Wordcloud Visualization

Setelah proses pelabelan, dilakukan visualisasi *wordcloud* secara terpisah untuk setiap label sentimen (positif dan negatif). Visualisasi ini memberikan gambaran awal mengenai pola dan kecenderungan kata dalam setiap kategori sentimen setelah melalui tahap pembersihan teks. Pembuatan dilakukan menggunakan pustaka *WordCloud* dan *Matplotlib* pada *Python*.

3.4.6 Ekstraksi Fitur

Tahap ini bertujuan untuk mengubah data teks menjadi bentuk numerik menggunakan metode TF-IDF (*Term Frequency–Inverse Document Frequency*).

1. *Term Frequency* (TF) menghitung seberapa sering suatu kata muncul dalam dokumen tertentu, sesuai Persamaan (2.18).

2. *Inverse Document Frequency* (IDF) menghitung seberapa jarang kata tersebut muncul di seluruh dokumen, sesuai Persamaan (2.19).
3. Nilai akhir TF-IDF diperoleh dari hasil perkalian kedua komponen di atas, sebagaimana Persamaan (2.20).

Proses *fit* dan *transform* dilakukan hanya pada data latih, sedangkan data uji hanya melalui *transform* menggunakan vektor hasil pelatihan. Langkah ini dilakukan untuk menghindari data *leakage*, yaitu kebocoran informasi dari data uji ke data latih.

3.4.7 Penanganan *Imbalanced Data*

Jika dataset menunjukkan ketidakseimbangan antara jumlah kelas positif dan negatif, maka digunakan teknik *Synthetic Minority Oversampling Technique* (SMOTE) untuk menyeimbangkan distribusi kelas. SMOTE bekerja dengan cara menghasilkan sampel sintetis baru pada kelas minoritas berdasarkan kedekatan antar data dalam ruang fitur. Proses pembangkitan sampel sintetis dalam SMOTE dilakukan melalui:

1. Menentukan tetangga terdekat (*k-nearest neighbors*)

Jarak antara sampel minoritas x_i dan kandidat tetangganya x_j dihitung menggunakan jarak Euclidean sebagaimana ditunjukkan pada Persamaan (2.21).

2. Membentuk sampel sintetis melalui interpolasi

Setelah tetangga terdekat diperoleh, sampel baru x_{new} dibentuk menggunakan Persamaan interpolasi pada Persamaan (2.22)

Penerapan SMOTE hanya dilakukan pada data latih, bukan pada data uji, agar distribusi data uji tetap mencerminkan kondisi sebenarnya dan tidak terjadi data

leakage. Dengan data latih yang telah seimbang, model diharapkan mampu mengenali pola sentimen secara lebih akurat dan tidak bias terhadap salah satu kelas.

3.4.8 Pemodelan dan Pelatihan Data

Tahap ini menggunakan algoritma Regresi Logistik sebagai metode utama klasifikasi sentimen.

1. Representasi fitur dan parameter model mengikuti Persamaan (2.1)–(2.3), di mana setiap data direpresentasikan dalam bentuk vektor fitur hasil pembobotan TF-IDF. Model dinyatakan sebagai kombinasi linear antara fitur dan parameter, sehingga diperoleh nilai logit $z^{(i)}$.
2. Probabilitas prediksi dihitung menggunakan fungsi sigmoid. Secara umum, penelitian ini juga mengeksplorasi bentuk sigmoid bertemperatur sebagaimana Persamaan (2.5). Namun, model utama dan hasil evaluasi akhir menggunakan nilai $\alpha = 1$, sehingga formulasi kembali pada regresi logistik standar. Variasi nilai α (0.5, 1.0, 2.0, dan 3.0) digunakan hanya untuk analisis pengaruh kecuraman fungsi sigmoid terhadap dinamika pelatihan dan stabilitas performa model.
3. Hasil probabilitas dibandingkan dengan ambang batas 0.5 untuk menentukan kelas. Jika $p^{(i)} \geq 0.5$, maka data diklasifikasikan sebagai kelas positif, sedangkan jika $p^{(i)} < 0.5$, maka diklasifikasikan sebagai kelas negatif.
4. Parameter bobot w dan bias b diestimasi menggunakan prinsip *Maximum Likelihood Estimation* sebagaimana Persamaan (2.10) dan *log-likelihood* pada Persamaan (2.11).

5. Fungsi objektif dinyatakan dalam bentuk *Binary Cross-Entropy* sesuai Persamaan (2.12), yang ekuivalen dengan meminimalkan negatif *log-likelihood*.
6. Gradien dihitung sesuai Persamaan (2.13)–(2.16), kemudian parameter diperbarui menggunakan algoritma *Adam Optimizer*.

Model dibangun menggunakan *Tensorflow* dengan arsitektur Regresi Logistik yang direpresentasikan sebagai satu neuron dengan fungsi aktivasi sigmoid. Mekanisme *early stopping* diterapkan untuk menghentikan proses pelatihan ketika *validation loss* tidak menunjukkan perbaikan, sehingga risiko *overfitting* dapat diminimalkan.

3.4.9 Evaluasi Model

Evaluasi dilakukan untuk menilai kinerja model regresi logistik dalam mengklasifikasikan sentimen ulasan. Metrik evaluasi yang digunakan didasarkan pada nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN), meliputi akurasi (Persamaan 2.23) untuk mengukur proporsi prediksi benar, presisi (Persamaan 2.24) untuk menilai ketepatan model dalam memprediksi sentimen positif, *recall* (Persamaan 2.25) untuk mengukur kemampuan model dalam mengenali seluruh data positif, serta *f1-score* (Persamaan 2.26) sebagai rata-rata harmonis antara presisi dan *recall*.

Selain metrik numerik, evaluasi juga mencakup *Confusion Matrix*, yang memberikan informasi komprehensif mengenai distribusi prediksi model pada setiap kelas. *Confusion matrix* digunakan untuk mengidentifikasi pola kesalahan model, seperti kecenderungan model melakukan *false positive* atau *false negative*, sehingga hasil evaluasi menjadi lebih interpretatif. Evaluasi dilakukan

sepenuhnya menggunakan pendekatan *hold-out test*, sesuai dengan alur penelitian, tanpa menerapkan metode *k-fold cross validation*. Dengan demikian, hasil evaluasi menggambarkan performa sebenarnya dari model final pada data uji yang tidak pernah digunakan dalam proses pelatihan maupun validasi.

3.5 Skenario Pengujian dan Analisis Perbandingan

Penelitian ini disusun dengan beberapa skenario pengujian yang dirancang untuk mengevaluasi efektivitas algoritma Regresi Logistik dalam mengklasifikasikan sentimen ulasan pengguna aplikasi *Mobile JKN*. Skenario tersebut difokuskan pada bagaimana perubahan distribusi data maupun perubahan karakteristik fungsi aktivasi dapat memengaruhi stabilitas, akurasi, serta kinerja model secara keseluruhan. Secara umum, terdapat dua kondisi utama yang diuji, yaitu tanpa penerapan *oversampling* dan dengan penerapan SMOTE sebagai teknik penyeimbang data latih. Melalui pengujian ini, penelitian mengamati sejauh mana SMOTE mampu meningkatkan kemampuan model dalam mengenali pola pada kelas minoritas dan mengurangi bias klasifikasi.

Selain melakukan evaluasi berdasarkan distribusi data, penelitian ini juga menyertakan skenario tambahan berupa pengujian variasi parameter kemiringan fungsi sigmoid melalui nilai α . Skenario ini dilakukan dengan tujuan untuk melihat bagaimana perubahan sensitivitas sigmoid terhadap nilai skor linier (logit) memengaruhi kualitas pembelajaran model. Nilai α yang diuji mencakup $\{0.5, 1.0, 2.0, 3.0\}$, dan pemilihan nilai terbaik dilakukan berdasarkan performa model pada data validasi maupun data uji. Pengujian ini memberikan gambaran tambahan mengenai perilaku model ketika kurva aktivasi dibuat lebih landai maupun lebih

curam, sehingga penelitian dapat menemukan konfigurasi model yang paling stabil dan optimal.

Analisis hasil dari setiap skenario dilakukan dengan meninjau metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score*. Selain itu, *confusion matrix* digunakan untuk mengidentifikasi pola kesalahan prediksi yang terjadi pada masing-masing kondisi. Melalui pendekatan ini, penelitian memperoleh pemahaman komprehensif mengenai bagaimana perubahan distribusi data akibat SMOTE dan perubahan kemiringan sigmoid akibat variasi α memengaruhi performa model regresi logistik. Hasil-hasil tersebut kemudian dibandingkan dengan temuan penelitian sebelumnya yang mengadopsi teknik serupa, sehingga diperoleh justifikasi yang lebih kuat mengenai efektivitas Regresi Logistik berbasis SMOTE dan modifikasi parameter sigmoid dalam konteks analisis sentimen pada aplikasi layanan publik digital.

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Pengumpulan Data

Pada tahap pengumpulan data, penelitian ini berhasil memperoleh 10.000 ulasan pengguna aplikasi *Mobile JKN* yang diambil dari *Google Play Store* melalui proses *web scraping*. Seluruh ulasan yang dikumpulkan merupakan ulasan berbahasa Indonesia dan berasal dari pengguna di Indonesia dalam rentang waktu 15 September 2025 hingga 18 November 2025. Dari setiap ulasan, dua elemen yang digunakan sebagai dasar penelitian adalah teks ulasan dan *rating* yang diberikan oleh pengguna. Kedua elemen tersebut kemudian disimpan dalam format CSV dan menjadi bahan utama pada proses pelabelan serta analisis pada tahap selanjutnya.

Tabel 4.1 menyajikan sampel data ulasan pengguna yang digunakan dalam penelitian ini.

Tabel 4.1 Sampel Data Ulasan *Mobile JKN*

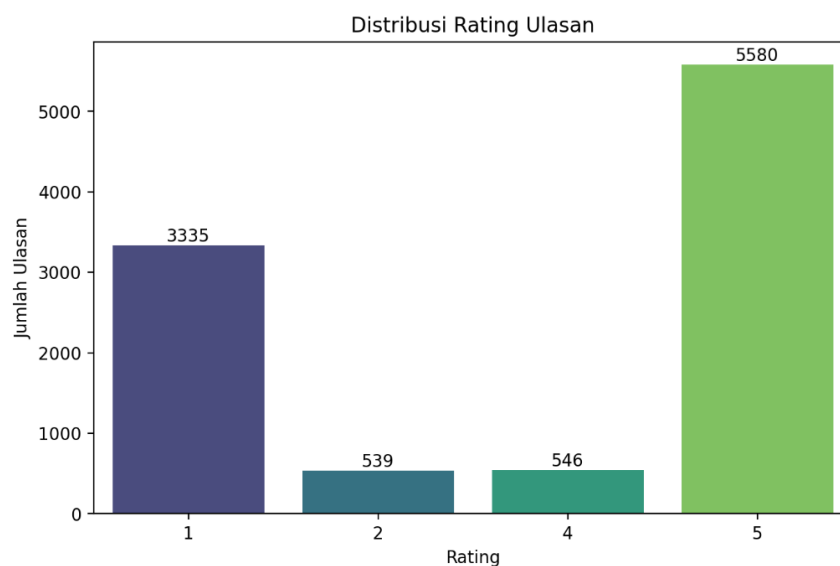
Rating	Ulasan
1	Verifikasi wajah sulitnya minta ampun 🙄
5	sangat membantu top bgt
1	"ga bisa verifikasi semua, lewat no hp dsruh coba lagi terus sudh dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance. Jd gmn mau verifikasi?????"
4	Terima kasih <i>Mobile JKN</i> sudah bisa aktif kembali...
2	banyak bug nya. setelah kemarin mau daftar online tp keterangannya yg intinya dokter belum tersedia.skrng giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia.app aneh

Dari Tabel 4.1 terlihat bahwa setiap data ulasan tersusun atas dua atribut utama, yaitu teks ulasan sebagai sumber informasi tekstual dan *rating* sebagai dasar penentuan sentimen pada penelitian ini.

4.1.2 *Exploratory Data Analysis (EDA)*

Tahap *Exploratory Data Analysis (EDA)* dilakukan untuk memahami karakteristik awal dataset sebelum memasuki proses *preprocessing* dan pemodelan. Pemeriksaan awal terhadap dua atribut yang digunakan dalam penelitian, yaitu teks ulasan dan *rating*, menunjukkan bahwa seluruh data berjumlah 10.000 baris dapat dibaca dengan baik dan tidak mengandung nilai hilang (*missing value*). Kondisi ini memastikan bahwa dataset siap digunakan secara penuh pada tahap analisis selanjutnya.

Gambaran mengenai persebaran *rating* pada dataset ditunjukkan pada Gambar 4.1, yang memvisualisasikan jumlah ulasan pada masing-masing kategori *rating* berikut.



Gambar 4.1 Persebaran *Rating* pada Dataset

Berdasarkan Gambar 4.1, *rating* 5 merupakan kategori terbanyak dengan 5.580 ulasan, diikuti *rating* 1 sebanyak 3.335 ulasan. Sementara itu, *rating* 2 dan 4 muncul dalam jumlah yang relatif kecil, yaitu masing-masing 539 dan 546 ulasan. Distribusi ini memperlihatkan kecenderungan ulasan yang terkonsentrasi pada dua kutub ekstrem, yakni sangat positif dan sangat negatif. Karena penelitian

ini berfokus pada analisis polaritas berbasis *rating*, distribusi kategori ini menjadi dasar penting dalam menentukan label sentimen, tanpa menganalisis aspek-aspek khusus dalam teks ulasan pengguna.

Selain pemeriksaan numerik pada *rating*, dilakukan pula visualisasi kata melalui *wordcloud* untuk memperoleh gambaran umum mengenai kata-kata yang sering muncul dalam ulasan, seperti pada Gambar 4.2 berikut.



Gambar 4.2 Wordcloud Dataset sebelum *Preprocessing*

Hasil visualisasi pada Gambar 4.2 menunjukkan bahwa kata seperti “*aplikasi*”, “*sangat*”, “*bisa*”, “*login*”, “*membantu*”, dan “*susah*” muncul dengan frekuensi tinggi. Temuan ini menggambarkan keberagaman pengalaman pengguna, meskipun dalam penelitian ini teks ulasan digunakan semata-mata sebagai representasi fitur, bukan sebagai dasar untuk menentukan sentimen secara semantik ataupun berdasarkan aspek tertentu. Dengan demikian, *wordcloud* hanya berfungsi sebagai gambaran umum karakteristik teks, bukan sebagai komponen penentu polaritas.

4.1.3 *Text Preprocessing*

Tahap text preprocessing dilakukan untuk menyiapkan teks ulasan agar memiliki format yang bersih, konsisten, dan siap diubah menjadi representasi

numerik. Seluruh proses ini diterapkan pada atribut teks ulasan. Selain itu, preprocessing berperan penting dalam mengurangi *noise* pada teks, seperti variasi penulisan, kata tidak baku, simbol yang tidak relevan, serta elemen-elemen linguistik yang tidak memberikan informasi berarti. Hasil *preprocessing* menunjukkan bahwa data mengalami perubahan struktur yang signifikan dari segi penulisan, kebakuan kata, hingga penyederhanaan bentuk kata.

1. Pembersihan Karakter (*Character Cleaning*)

Tahap ini menghapus semua karakter non-alfanumerik (selain huruf, angka, dan spasi) menggunakan pola *regex* `[\^a-zA-Z0-9\s]`. Emoji, tanda baca berlebihan, simbol, dan karakter tidak relevan dihilangkan. Seperti pada Tabel 4.2 berikut.

Tabel 4.2 Sampel Hasil *Character Cleaning*

Sebelum <i>character cleaning</i>	Sesudah <i>character cleaning</i>
Verifikasi wajah sulitnya minta ampun 🙄	Verifikasi wajah sulitnya minta ampun
sangat membantu top bgt	sangat membantu top bgt
"ga bisa verifikasi semua, lewat no hp disuruh coba lagi terus sudah dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance. Jd gmn mau verifikasi?????"	ga bisa verifikasi semua lewat no hp dsuruh coba lagi terus sudh dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance Jd gmn mau verifikasi
Terima kasih <i>Mobile</i> JKN sudah bisa aktif kembali...	Terima kasih <i>Mobile</i> JKN sudah bisa aktif kembali
banyak bug nya. setelah kemarin mau daftar online tp keterangannya yg intinya dokter belum tersedia.skrng giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia.app aneh	banyak bug nya setelah kemarin mau daftar online tp keterangannya yg intinya dokter belum tersedia skrg giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia app aneh

2. *Case Folding*

Pada tahap ini, seluruh huruf dalam teks ulasan diubah menjadi huruf kecil.

Hasil *case folding* memastikan tidak ada perbedaan perlakuan antara kata

yang sama tetapi ditulis dengan huruf kapital berbeda seperti pada Tabel 4.3 berikut.

Tabel 4.3 Sampel Hasil *Case Folding Dataset*

Sebelum <i>case folding</i>	Sesudah <i>case folding</i>
Verifikasi wajah sulitnya minta ampun	verifikasi wajah sulitnya minta ampun
sangat membantu top bgt	sangat membantu top bgt
ga bisa verifikasi semua lewat no hp dsruh coba lagi terus sudh dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance Jd gmn mau verifikasi	ga bisa verifikasi semua lewat no hp disuruh coba lagi terus sudah dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance jd gmn mau verifikasi
Terima kasih <i>Mobile</i> JKN sudah bisa aktif kembali	terima kasih mobile jkn sudah bisa aktif kembali
banyak bug nya setelah kemarin mau daftar online tp keterangannya yg intinya dokter belum tersedia skrg giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia app aneh	banyak bug nya setelah kemarin mau daftar online tp keterangannya yg intinya dokter belum tersedia skrg giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia app aneh

3. *Text Normalization*

Normalisasi dilakukan dengan mengganti kata tidak baku, singkatan, atau bentuk informal menjadi bentuk standar berdasarkan kamus normalisasi. Seperti pada Tabel 4.4 berikut.

Tabel 4.4 Sampel Hasil *Text Normalization*

Sebelum <i>text normalization</i>	Sesudah <i>text normalization</i>
verifikasi wajah sulitnya minta ampun	verifikasi wajah sulitnya minta ampun
sangat membantu top bgt	sangat membantu top banget
ga bisa verifikasi semua lewat no hp disuruh coba lagi terus sudah dua minggu dicoba berkala gabisa lewat wajah katanya tp maintenance jd gmn mau verifikasi	tidak bisa verifikasi semua lewat nomor <i>handphone</i> disuruh coba lagi terus sudah dua minggu dicoba berkala tidak bisa lewat wajah katanya tapi <i>maintenance</i> jadi bagaimana mau verifikasi
terima kasih <i>mobile</i> jkn sudah bisa aktif kembali	terima kasih mobile jkn sudah bisa aktif kembali
banyak bug nya setelah kemarin mau daftar online tp	banyak bugnya setelah kemarin mau daftar online tapi

keterangannya yg intinya dokter belum tersedia skrg giliran dokter udah bisa muncul eh malah ada keterangan poli yang anda tuju tidak tersedia app aneh	keterangannya yang intinya dokter belum tersedia sekarang giliran dokter sudah bisa muncul ada keterangan poli yang anda tuju tidak tersedia aplikasi aneh
---------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------

4. *Stopword Removal*

Stopword removal dilakukan untuk menghapus kata-kata umum dan kata fungsional lain yang tidak memberikan kontribusi signifikan pada pembentukan fitur. Hasilnya, teks menjadi lebih ringkas dan fokus pada kata bermakna. Seperti pada Tabel 4.5.

Tabel 4.5 Sampel Hasil *Stopword Removal*

Sebelum <i>stopword removal</i>	Sesudah <i>stopword removal</i>
verifikasi wajah sulitnya minta ampun	verifikasi wajah sulit minta ampun
sangat membantu top banget	membantu top banget
tidak bisa verifikasi semua lewat nomor <i>handphone</i> disuruh coba lagi terus sudah dua minggu dicoba berkala tidak bisa lewat wajah katanya tapi <i>maintenance</i> jadi bagaimana mau verifikasi	tidak bisa verifikasi lewat nomor <i>handphone</i> disuruh coba dua minggu dicoba berkala tidak bisa lewat wajah <i>maintenance</i> bagaimana mau verifikasi
terima kasih <i>mobile</i> jkn sudah bisa aktif kembali	terima kasih <i>mobile</i> jkn aktif
banyak bugnya setelah kemarin mau daftar online tapi keterangannya yang intinya dokter belum tersedia sekarang giliran dokter sudah bisa muncul ada keterangan poli yang anda tuju tidak tersedia aplikasi aneh	banyak bug kemarin daftar online keterangan inti dokter belum tersedia sekarang giliran dokter muncul keterangan poli tidak tersedia aplikasi aneh

5. *Stemming*

Stemming mengubah kata berimbuhan menjadi bentuk dasar menggunakan algoritma *stemming* bahasa Indonesia (*Sastrawi*). Tahap ini mengurangi variasi kata yang memiliki akar makna sama. Seperti pada Tabel 4.6 berikut.

Tabel 4.6 Sampel Hasil *Stemming*

Sebelum <i>stemming</i>	Sesudah <i>stemming</i>
verifikasi wajah sulit minta ampun	verifikasi wajah sulit minta ampun
membantu top banget	bantu top banget
tidak bisa verifikasi lewat nomor handphone disuruh coba dua minggu dicoba berkala tidak bisa lewat wajah maintenance bagaimana mau verifikasi	tidak bisa verifikasi lewat nomor handphone suruh coba dua minggu coba berkala tidak bisa lewat wajah maintenance mana mau verifikasi
terima kasih <i>Mobile</i> jkn aktif	terima kasih mobile jkn aktif
banyak bug kemarin daftar online keterangan inti dokter belum tersedia sekarang giliran dokter muncul keterangan poli tidak tersedia aplikasi aneh	banyak bug kemarin daftar online terang inti dokter belum sedia sekarang giliran dokter muncul terang poli tidak sedia aplikasi aneh

6. *Cleaning Data*

Tahap akhir dilakukan untuk memastikan tidak terdapat entri kosong yang muncul akibat penghapusan *stopword* atau pembersihan karakter.

Dataframe after cleaning data:		
	score	content
0	4	bagus aplikasi
1	1	aplikasi lambat ribet
2	2	aplikasi busuksetiap kali buka butuh banget su...
3	1	aplikasi banget logout kerja rumah banget must...
4	5	bantu

Number of rows remaining: 9736

Gambar 4.3 Hasil *Clean Data*

Gambar 4.3 menunjukkan cuplikan hasil akhir dari data setelah melalui seluruh tahapan preprocessing, di mana setiap ulasan telah berubah menjadi bentuk teks yang bersih dan ringkas. Berdasarkan hasil tersebut, jumlah baris yang tersisa setelah pembersihan adalah 9.736 baris dari total 10.000 data awal. Untuk memberikan gambaran yang lebih komprehensif, Gambar 4.4 menyajikan perbandingan antara teks ulasan asli (*original_content*) dan hasil pembersihannya (*clean_data*).

Comparison table of original content, score, and cleaned data:

	original_content	score	clean_data
0	bagus sudah ada aplikasinya	4.0	bagus aplikasi
1	aplikasi lambat, ribet	1.0	aplikasi lambat ribet
2	Aplikasi busuk,setiap kali mau buka saat butuh...	2.0	aplikasi busuksetiap kali buka butuh banget su...
3	apk sering bgt ke logout ,pr bgt klo musti log...	1.0	aplikasi banget logout kerja rumah banget must...
4	sangat membantu	5.0	bantu
...
9995	ribett	1.0	ribett
9996	ok	5.0	ok
9997	tolong terus update	4.0	tolong update
9998	kode otp sms gk masuk masuk ini gimana	1.0	kode otp sms masuk masuk
9999	Sistem ngebug lelet & tidak pesan dipakai	1.0	sistem ngebug lelet pesan pakai

10000 rows x 3 columns

Gambar 4.4 Perbandingan Dataset Mentah & *Clean Data*

Melalui tabel perbandingan tersebut terlihat bahwa pembersihan karakter, normalisasi teks, *stopword removal*, dan *stemming* mampu menghilangkan elemen teks yang tidak relevan tanpa mengubah makna utama dari ulasan. Proses ini memastikan bahwa data yang digunakan pada tahap TF-IDF memiliki kualitas yang optimal, konsisten, dan bebas *noise*.

4.1.4 Pelabelan

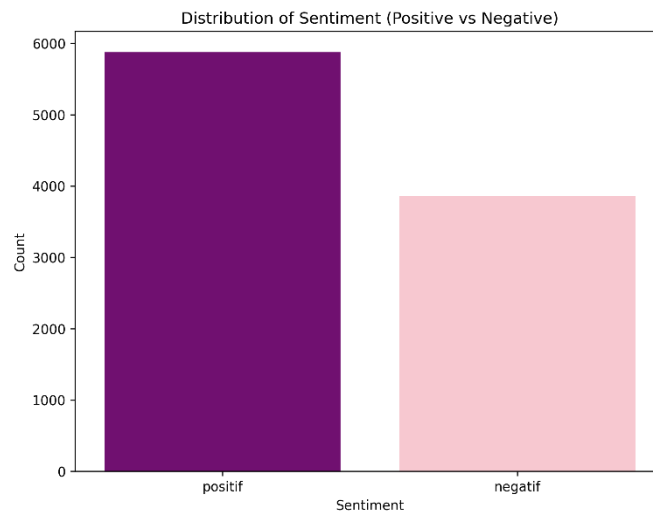
Proses pelabelan dilakukan dengan memetakan nilai *rating* pengguna ke dalam dua kategori sentimen utama. *Rating* 1–2 diklasifikasikan sebagai sentimen negatif, sementara *rating* 4–5 dikategorikan sebagai sentimen positif, sesuai dengan batasan masalah yang ditetapkan dalam penelitian ini. *Rating* 3 tidak disertakan sejak tahap pengumpulan data karena bersifat netral dan tidak menunjukkan kecenderungan sentimental tertentu. Pelabelan dilakukan menggunakan fungsi kategorisasi pada *Python*, dan diterapkan pada seluruh data ulasan. Beberapa sampel hasil pelabelan ditunjukkan pada Tabel 4.7 berikut.

Tabel 4.7 Hasil Pelabelan

<i>Rating</i>	Ulasan	Label Sentimen
1	verifikasi wajah sulit ampun	Negatif
5	bantu top banget	Positif

1	tidak bisa verifikasi lewat nomor handphone suruh coba dua minggu coba berkala tidak bisa lewat wajah maintenance mana mau verifikasi	Negatif
4	terima kasih mobile jkn aktif	Positif
2	banyak bug kemarin daftar online terang inti dokter belum sedia sekarang giliran dokter muncul terang poli tidak sedia aplikasi aneh	Negatif

Proses pelabelan dilakukan dengan mengonversi nilai *rating* menjadi dua kategori sentimen, yaitu negatif untuk *rating* 1–2 dan positif untuk *rating* 4–5. Setelah pelabelan diterapkan pada seluruh data, diperoleh 5.876 ulasan positif dan 3.860 ulasan negatif, dengan total 9.736 ulasan yang telah melalui proses pembersihan tanpa menyertakan *rating* 3. Distribusi ini divisualisasikan pada Gambar 4.5 di mana grafik batang yang menunjukkan perbedaan jumlah yang cukup mencolok antara kedua kelas.



Gambar 4.5 Grafik Distribusi Label Sentimen

Pada Gambar 4.5 memperlihatkan bahwa ulasan positif mendominasi dataset, tergambar dari batang kelas positif yang jauh lebih tinggi dibandingkan batang kelas negatif.

berdasarkan *rating*, tanpa melakukan pemetaan isu atau evaluasi kualitas fitur tertentu.

4.1.6 *Split Data*

Proses pemisahan data dilakukan dengan membagi dataset ke dalam dua subset, yaitu data latih sebesar 80% dan data uji sebesar 20%, dengan tetap mempertahankan proporsi masing-masing kelas sentimen. Hasil pemisahan menunjukkan bahwa data latih terdiri atas 4.700 ulasan positif dan 3.088 ulasan negatif, sedangkan data uji berisi 1.176 ulasan positif dan 772 ulasan negatif. Distribusi ini memberikan gambaran bahwa komposisi kedua subset masih selaras dengan pola awal dataset, di mana ulasan positif memang lebih dominan dibandingkan ulasan negatif. Dengan tetap menjaga proporsi tersebut, model dapat mempelajari pola sentimen secara lebih stabil, sementara evaluasi pada data uji tetap mencerminkan kondisi nyata dataset. Pemisahan seperti ini juga penting agar proses evaluasi performa model berlangsung objektif dan konsisten dengan fokus penelitian yang hanya menganalisis polaritas *rating*, sehingga hasil pengujian benar-benar menggambarkan kemampuan model dalam membedakan ulasan positif dan negatif tanpa bias terhadap salah satu kelas.

4.1.7 Ekstraksi Fitur

Tahap ekstraksi fitur dilakukan dengan mengubah teks ulasan yang telah melalui proses *preprocessing* menjadi representasi numerik menggunakan metode TF-IDF (*Term Frequency–Inverse Document Frequency*). Pada tahap ini, setiap kata diberi bobot berdasarkan frekuensi kemunculannya dalam satu dokumen serta tingkat kelangkaannya pada seluruh kumpulan dokumen. Perhitungan bobot tersebut dilakukan melalui dua komponen utama, yaitu *Term Frequency* (TF) dan

Inverse Document Frequency (IDF). Nilai TF dihitung menggunakan Persamaan (2.18), sedangkan nilai IDF ditentukan menggunakan Persamaan (2.19). Bobot akhir TF-IDF kemudian diperoleh dari hasil perkalian kedua komponen tersebut, sebagaimana ditetapkan dalam Persamaan (2.20). Berikut sampel data yang akan digunakan untuk proses TF-IDF dapat dilihat pada Tabel 4.8.

Tabel 4.8 Sampel Data TF-IDF

D	Kalimat	Label
D1	update susah login nik salah lupa password email daftar salah ujung kantor bpjs langsung aplikasi kasih bintang	0
D2	daftar online periksa anak faskes bisa niat mudah orang rumah nyusahin	0
D3	aplikasi gampang aman kirim sms isi pulsa masyarakat faham mohon sosialisasi orang bantu terimakasih	1
D4	jelek banget susah login lelet ribet jelek banget sumpah kecewa sistem	0
D5	bagus bantu daftar periksa faskes informasi riwayat layanan muncul cuma riwayat istri 2 anak muncul masuk benah aplikasi terima kasih	1
D6	ubah data langsung verifikasi mudah gampang banget terimakasih jkn	1
D7	aplikasi bantu pindah faskes tuju terimakasih bantu banget mudah terima kasih	1
D8	aplikasi bantu mudah moga manfaat guna jkn <i>mobile</i> terimakasih	1
D9	bagus mudah deteksi wajah kadang ulang lebih praktis pindah faskes repot kantor cukup aplikasi <i>mobile</i> jkn	1
D10	aplikasi susah verifikasi muka gagal adopsi teknologi baru buru susah orang obat ribet wajib daftar <i>mobile</i> jkn gilir daftar verifikasi muka hasil	0

Setelah dokumen sampel diperoleh, setiap dokumen akan diuraikan menjadi term atau kata, kemudian disaring sehingga hanya mempertahankan kata-kata yang termasuk dalam daftar fitur yang telah ditentukan. Tabel 4.9 menampilkan frekuensi kemunculan masing-masing term pada setiap dokumen sampel (D1–D10). Setiap kolom merepresentasikan term yang telah ditentukan sebelumnya, yaitu *aplikasi*, *daftar*, *mudah*, *bantu*, *susah*, *faskes*, *terimakasih*, dan *jkn*. Sementara itu, setiap baris menunjukkan keberadaan term tersebut pada

dokumen tertentu. Nilai 1 menunjukkan bahwa term muncul dalam dokumen, sedangkan nilai 0 menunjukkan bahwa *term* tidak ditemukan.

Tabel 4.9 Frekuensi Kemunculan Kata pada Dokumen

D	Frekuensi Kemunculan <i>Term</i>							
	aplikasi	daftar	mudah	bantu	susah	faskes	terimakasih	jkn
D1	1	1	0	0	1	0	0	0
D2	0	1	1	0	0	1	0	0
D3	1	0	0	1	0	0	1	0
D4	0	0	0	0	1	0	0	0
D5	1	1	0	1	0	1	0	0
D6	0	0	1	0	0	0	1	1
D7	1	0	1	2	0	1	1	0
D8	1	0	1	1	0	0	1	1
D9	1	0	1	0	0	1	0	1
D10	1	2	0	0	2	0	0	1

Berikut contoh perhitungan pada penelitian ini menggunakan Persamaan (2.18). Salah satu dokumen sampel yang digunakan adalah dokumen D1, di mana dokumen tersebut memiliki 17 kata, sehingga nilai TF setiap *term* diperoleh dengan membagi jumlah kemunculan *term* dengan total kata.

Tabel 4.10 Perhitungan TF pada D1

Term	Frekuensi	Total Kata	Rumus	TF
aplikasi	1	17	$TF = \frac{1}{17}$	0.0588
daftar	1	17	$TF = \frac{1}{17}$	0.0588
mudah	0	17	$TF = \frac{0}{17}$	0
bantu	0	17	$TF = \frac{0}{17}$	0
susah	1	17	$TF = \frac{1}{17}$	0.0588
faskes	0	17	$TF = \frac{0}{17}$	0
terimakasih	0	17	$TF = \frac{0}{17}$	0
jkn	0	17	$TF = \frac{0}{17}$	0

Setelah memperoleh nilai TF untuk dokumen D1, proses yang sama diterapkan pada seluruh dokumen sampel (D2–D10). Perhitungan TF dilakukan secara konsisten menggunakan delapan term fitur yang telah ditentukan. Hasil perhitungan TF untuk keseluruhan dokumen dapat dilihat pada Tabel 4.11 berikut.

Tabel 4.11 Perhitungan TF

D	TF							
	aplikasi	daftar	mudah	bantu	susah	faskes	terimakasih	jkn
D1	0.0588	0.0588	0	0	0.0588	0	0	0
D2	0	0.0909	0.0909	0	0	0.0909	0	0
D3	0.0714	0	0	0.0714	0	0	0.0714	0
D4	0	0	0	0	0.0833	0	0	0
D5	0.0500	0.0500	0	0.0500	0	0.0500	0	0
D6	0	0	0.1111	0	0	0	0.1111	0.1111
D7	0.0769	0	0.0769	0.1538	0	0.0769	0.0769	0
D8	0.1111	0	0.1111	0.1111	0	0	0.1111	0.1111
D9	0.0625	0	0.0625	0	0	0.0625	0	0.0625
D10	0.0435	0.0870	0	0	0.0870	0	0	0.0435

Setelah nilai TF diperoleh untuk setiap dokumen, tahap berikutnya adalah menghitung nilai *Inverse Document Frequency* (IDF). IDF digunakan untuk mengetahui tingkat kelangkaan suatu term dalam seluruh dokumen, sehingga term yang muncul di lebih sedikit dokumen akan mendapatkan bobot yang lebih tinggi. Perhitungan IDF mengacu pada Persamaan (2.19), sebagai ilustrasi perhitungan IDF untuk term “aplikasi”. Di mana diketahui term “aplikasi” muncul pada 7 dokumen ($df = 7$), sedangkan total dokumen berjumlah 10 ($N = 10$). Maka,

$$IDF (\text{aplikasi}) = \log\left(\frac{10}{7}\right)$$

$$IDF (\text{aplikasi}) = \log(1.4286) = 0.1549$$

Dengan demikian, nilai IDF term *aplikasi* adalah 0.0969.

Setelah memperoleh nilai *Document Frequency* (DF) untuk setiap term, tahap berikutnya adalah menghitung nilai *Inverse Document Frequency* (IDF).

Perhitungan IDF dilakukan menggunakan Persamaan (2.19), yang bertujuan untuk mengukur tingkat kelangkaan sebuah term dalam keseluruhan dokumen. Term yang muncul pada lebih sedikit dokumen akan memiliki nilai IDF yang lebih tinggi, dan sebaliknya. Proses perhitungan dilakukan secara konsisten untuk seluruh delapan term fitur yang telah ditentukan. Hasil perhitungan IDF secara lengkap ditampilkan pada Tabel 4.12 berikut.

Tabel 4.12 Perhitungan IDF

Term	DF	Rumus	IDF
aplikasi	7	$\log\left(\frac{10}{7}\right)$	0.1549
daftar	4	$\log\left(\frac{10}{4}\right)$	0.3979
mudah	5	$\log\left(\frac{10}{5}\right)$	0.3010
bantu	4	$\log\left(\frac{10}{4}\right)$	0.3979
susah	3	$\log\left(\frac{10}{3}\right)$	0.5229
faskes	4	$\log\left(\frac{10}{4}\right)$	0.3979
terimakasih	4	$\log\left(\frac{10}{4}\right)$	0.3979
jkn	4	$\log\left(\frac{10}{4}\right)$	0.3979

Setelah memperoleh nilai TF dan IDF untuk setiap *term*, tahap selanjutnya adalah menghitung bobot TF-IDF. Bobot ini diperoleh dengan mengalikan nilai *Term Frequency* (TF) dengan nilai *Inverse Document Frequency* (IDF) sesuai Persamaan (2.20). TF-IDF digunakan untuk mengukur tingkat kepentingan sebuah *term* dalam dokumen, di mana nilai yang lebih besar menunjukkan bahwa term tersebut lebih representatif terhadap dokumen tertentu. Digunakan D1 untuk contoh perhitungan yang telah diketahui nilai TF dan IDF pada Tabel 4.13 berikut.

Tabel 4.13 Perhitungan TF-IDF pada D1

Term	TF	IDF	TF-IDF
aplikasi	0.0588	0.1549	0.0091
daftar	0.0588	0.3979	0.0234
mudah	0	0.3010	0
bantu	0	0.3979	0
susah	0.0588	0.5229	0.0308
faskes	0	0.3979	0
terimakasih	0	0.3979	0
jkn	0	0.3979	0

Berdasarkan hasil perhitungan pada dokumen D1, bobot TF-IDF menunjukkan tingkat kepentingan masing-masing *term* dalam dokumen tersebut. Proses perhitungan yang sama kemudian diterapkan pada seluruh dokumen sampel (D1–D10) menggunakan delapan *term* fitur yang telah ditetapkan. Nilai TF-IDF masing-masing *term* diperoleh dari hasil perkalian antara nilai TF pada tabel sebelumnya dengan nilai IDF pada Tabel 4.13. Hasil lengkap perhitungan TF-IDF disajikan pada Tabel 4.14 berikut.

Tabel 4.14 Perhitungan TF-IDF

D	TF-IDF							
	aplikasi	daftar	mudah	bantu	susah	faskes	terimakasih	jkn
D1	0.0091	0.0234	0	0	0.0308	0	0	0
D2	0	0.0362	0.0274	0	0	0.0362	0	0
D3	0.0110	0	0	0.0284	0	0	0.0284	0
D4	0	0	0	0	0.0436	0	0	0
D5	0.0077	0.0199	0	0.0199	0	0.0199	0	0
D6	0	0	0.0334	0	0	0	0.0442	0.0442
D7	0.0119	0	0.0334	0.0611	0	0.0307	0.0307	0
D8	0.0172	0	0.0334	0.0442	0	0	0.0442	0.0442
D9	0.0097	0	0.0188	0	0	0.0249	0	0.0249
D10	0.0067	0.0341	0	0	0.0455	0	0	0.0173

Tabel 4.14 menunjukkan hasil perhitungan bobot TF-IDF untuk seluruh dokumen sampel. Nilai TF-IDF memberikan gambaran mengenai tingkat kepentingan suatu term dalam dokumen tertentu. *Term susah* memiliki nilai TF-

IDF tertinggi pada beberapa dokumen, terutama ketika *term* tersebut muncul pada teks yang relatif panjang dengan nilai DF yang rendah. Sebaliknya, *term aplikasi* cenderung memiliki bobot TF-IDF lebih kecil karena meskipun sering muncul, *term* tersebut memiliki nilai IDF rendah akibat kemunculannya di banyak dokumen. Hasil perhitungan TF-IDF ini selanjutnya digunakan sebagai representasi fitur pada proses pemodelan menggunakan algoritma regresi logistik.

4.1.8 Penanganan *Imbalanced Data*

Distribusi awal kelas pada dataset menunjukkan adanya ketidakseimbangan antara jumlah ulasan positif dan negatif. Ketidakseimbangan ini dapat menyebabkan model regresi logistik cenderung bias terhadap kelas mayoritas karena model lebih sering melihat pola dari kelas tersebut. Oleh karena itu, diperlukan teknik *oversampling* untuk menyeimbangkan jumlah data antar kelas. Berdasarkan Gambar 4.5, jumlah ulasan positif sebanyak 5.876 data, sedangkan ulasan negatif sebanyak 3.860 data. Perbandingan ini menunjukkan ketidakseimbangan kelas dengan proporsi sekitar 60:40, di mana kelas positif berperan sebagai kelas mayoritas. Kondisi ini dapat memengaruhi performa algoritma dalam mengenali pola sentimen negatif karena ukuran kelas minoritas lebih kecil, sehingga dibutuhkan metode penyeimbangan seperti SMOTE untuk menghasilkan representasi data latih yang lebih seimbang.

SMOTE diterapkan hanya pada data latih untuk menghasilkan sampel sintetis pada kelas minoritas. Penerapan SMOTE dilakukan setelah proses split agar tidak terjadi *data leakage*, sehingga data uji tetap mencerminkan kondisi distribusi sebenarnya. Proses ini dilakukan melalui pendekatan berbasis kedekatan antar data. Teknik ini bekerja dengan menambahkan sampel sintetis hasil

interpolasi antara satu sampel minoritas dengan tetangga terdekatnya. Secara umum, tahapan SMOTE dapat dijelaskan sebagai berikut.

1. Pemilihan Sampel dari Kelas Minoritas

Langkah pertama yaitu memilih data dari kelas minoritas sebagai titik awal pembangkitan sampel sintetis. Misalnya pada kelas negatif terdapat dua data dengan representasi vektor fitur $x_i = (0.10, 0.30)$ dan $x_j = (0.14, 0.25)$. Kedua titik ini berada dalam ruang fitur dua dimensi dan akan digunakan sebagai dasar proses interpolasi.

2. Menentukan Tetangga Terdekat (*Nearest Neighbors*)

Setelah memilih satu sampel minoritas, SMOTE mencari tetangga terdekat dari sampel tersebut menggunakan metode *k-Nearest Neighbors* (k-NN). Jarak antara dua titik dihitung menggunakan jarak Euclidean sesuai Persamaan (2.21):

$$d(x_i, x_j) = \sqrt{(0.14 - 0.10)^2 + (0.25 - 0.30)^2}$$

$$d = \sqrt{(0.04)^2 + (-0.05)^2}$$

$$d = \sqrt{0.0016 + 0.0025}$$

$$d = \sqrt{0.0041} \approx 0.064$$

Hasil perhitungan menunjukkan bahwa kedua titik tersebut memiliki jarak *Euclidean* sekitar 0.064, sehingga dapat dianggap dekat dan memenuhi kriteria untuk proses pembangkitan sampel sintetis.

3. Membangkitkan Sampel Sintetis Melalui Interpolasi Linier

Setelah tetangga terdekat ditemukan, SMOTE menggunakan parameter acak $\lambda \in [0,1]$, maka dipilih $\lambda = 0.5$. Kemudian, membentuk sampel baru dengan

melakukan interpolasi linier antara titik x_i dan x_j menggunakan Persamaan (2.22).

Untuk dimensi pertama $x_i = 0.10$ dan $x_j = 0.14$

$$x_{new,1} = 0.10 + 0.5(0.14 - 0.10)$$

$$x_{new,1} = 0.10 + 0.5(0.04)$$

$$x_{new,1} = 0.10 + 0.02 = 0.12$$

Untuk dimensi kedua $x_i = 0.30$ dan $x_j = 0.25$

$$x_{new,2} = 0.30 + 0.5(0.25 - 0.30)$$

$$x_{new,2} = 0.30 + 0.5(-0.05)$$

$$x_{new,2} = 0.30 + (-0.025) = 0.275$$

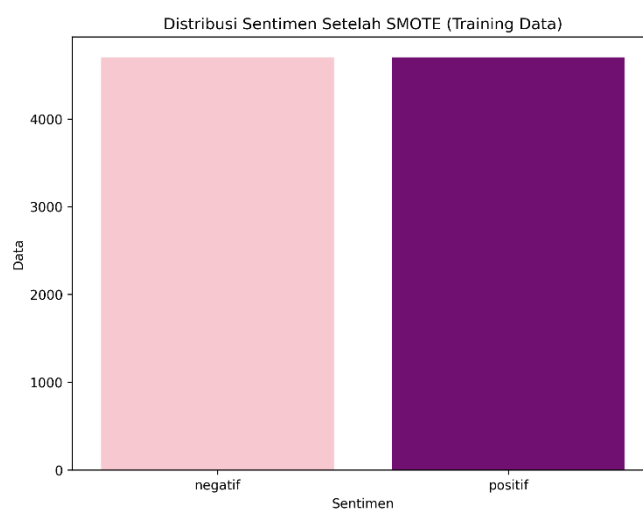
Jadi, titik sintetis baru yang dihasilkan adalah $x_{new} = (0.12, 0.275)$

4. Menambahkan Titik Sintetis ke Dataset

Titik sintetis (0.12, 0.275) tersebut kemudian ditambahkan sebagai data baru pada kelas minoritas. Proses ini dilakukan berulang kali hingga jumlah sampel minoritas mendekati jumlah kelas mayoritas, sehingga distribusi data latih menjadi seimbang. Dengan distribusi yang lebih merata, algoritma pembelajaran mesin dapat mempelajari pola kelas minoritas secara lebih efektif, tanpa bias dominan terhadap kelas mayoritas.

SMOTE tidak hanya berfungsi untuk menyeimbangkan jumlah sampel pada kelas minoritas, tetapi juga membantu mengurangi bias model terhadap kelas mayoritas. Dengan menambahkan sampel sintetis yang dibentuk berdasarkan kedekatan antar data minoritas, SMOTE memungkinkan model untuk memperoleh representasi pola yang lebih beragam dari kelas yang jarang muncul. Hal ini membuat proses pembelajaran menjadi lebih seimbang, sehingga model

tidak lagi cenderung memprioritaskan kelas mayoritas dan mampu mengenali karakteristik kelas minoritas secara lebih efektif. Pada Gambar 4.8 menunjukkan distribusi hasil *oversampling* menggunakan SMOTE pada keseluruhan data latih sebagai berikut.



Gambar 4.8 Grafik Distribusi Label Sentimen Setelah SMOTE

Gambar 4.8 menunjukkan distribusi sentimen pada data latih setelah dilakukan proses *oversampling* menggunakan SMOTE. Terlihat bahwa jumlah data pada kelas negatif dan positif telah berhasil disetarakan, di mana masing-masing kelas memiliki jumlah sampel yang hampir sama. Kondisi ini menunjukkan bahwa proses SMOTE bekerja efektif dalam menyeimbangkan proporsi data, sehingga masalah ketidakseimbangan kelas yang terdapat pada data asli berhasil diatasi. Dengan distribusi yang lebih seimbang, model pembelajaran mesin tidak lagi didominasi oleh pola yang berasal dari kelas mayoritas. Hal ini memungkinkan model untuk mempelajari karakteristik kelas minoritas secara lebih optimal dan mengurangi kecenderungan bias dalam proses prediksi. Dampaknya, performa model dalam membedakan sentimen positif dan negatif menjadi lebih stabil dan representatif terhadap keseluruhan data.

4.1.9 Pemodelan dan Pelatihan Data

Pada tahap pemodelan Regresi Logistik, setiap dokumen ulasan yang telah direpresentasikan dalam bentuk vektor TF-IDF diproyeksikan ke dalam skor linier (*logit*) menggunakan operasi perkalian antara vektor parameter model dan vektor fitur. Secara umum, skor linier dihitung menggunakan Persamaan (2.5). Secara matematis, operasi $\mathbf{w}^T \mathbf{x}^{(i)}$ merupakan *dot product* atau perkalian titik antara dua vektor berdimensi sama. Perkalian tersebut dilakukan dengan cara mengalikan setiap elemen vektor bobot dengan elemen vektor fitur yang bersesuaian, kemudian menjumlahkan seluruh hasil perkaliannya. Bentuk ekspansinya adalah:

$$\mathbf{w}^T \mathbf{x}^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \dots + w_n x_n^{(i)}$$

Untuk menunjukkan proses komputasi secara eksplisit, digunakan contoh bobot ilustratif berikut:

$$\mathbf{w} = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.05 \\ 0.02 \\ 0.15 \\ 0 \\ 0 \\ 0 \end{bmatrix}, b = 0.01$$

Bobot ilustratif tersebut digunakan hanya untuk menunjukkan mekanisme perhitungan manual, karena bobot asli model Regresi Logistik yang dilatih menggunakan *TensorFlow* berjumlah sangat banyak (ribuan fitur TF-IDF) dan tidak dapat ditampilkan seluruhnya. Selain itu, *TensorFlow* menginisialisasi bobot secara otomatis menggunakan metode *Glorot Uniform Initializer*, sehingga nilai bobot awal bersifat acak dalam rentang kecil untuk menjaga stabilitas proses pelatihan.

Perlu ditegaskan bahwa contoh perhitungan manual berikut menggunakan nilai $\alpha = 1$, sehingga fungsi aktivasi yang digunakan adalah sigmoid standar. Berikut perhitungan Regresi Logistik:

1. Perhitungan Skor Linier (Persamaan 2.5)

- Untuk D1,

$$x_1^{(i)} = \begin{bmatrix} 0.0091 \\ 0.0234 \\ 0 \\ 0 \\ 0.0308 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_1^{(i)} &= (0.2)(0.0091) + (-0.1)(0.0234) + (0.05)(0) + (0.02)(0) \\ &\quad + (0.15)(0.0308) + (0)(0) + (0)(0) + (0)(0) + 0.01 \\ &= 0.00182 - 0.00234 + 0.00462 + 0.01 = 0.01410 \end{aligned}$$

- Untuk D2,

$$x_2^{(i)} = \begin{bmatrix} 0 \\ 0.0362 \\ 0.0274 \\ 0 \\ 0 \\ 0.0362 \\ 0 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_2^{(i)} &= (0.2)(0) + (-0.1)(0.0362) + (0.05)(0.0274) + (0.02)(0) \\ &\quad + (0.15)(0) + (0)(0.0362) + (0)(0) + (0)(0) + 0.01 \\ &= -0.00362 + 0.00137 + 0.01 = 0.00775 \end{aligned}$$

- Untuk D3,

$$x_3^{(i)} = \begin{bmatrix} 0.0110 \\ 0 \\ 0 \\ 0.0284 \\ 0 \\ 0.0284 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_3^{(i)} &= (0.2)(0.0110) + (-0.1)(0) + (0.05)(0) + (0.02)(0.0284) \\ &\quad + (0.15)(0) + (0)(0) + (0)(0.0284) + (0)(0) + 0.01 \\ &= 0.00220 + 0.000568 + 0.01 = 0.01277 \end{aligned}$$

- Untuk D4,

$$x_4^{(i)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.0436 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_4^{(i)} &= (0.2)(0) + (-0.1)(0) + (0.05)(0) + (0.02)(0) \\ &\quad + (0.15)(0.0436) + (0)(0) + (0)(0) + (0)(0) + 0.01 \\ &= 0.00654 + 0.01 = 0.01654 \end{aligned}$$

- Untuk D5,

$$x_5^{(i)} = \begin{bmatrix} 0.0077 \\ 0.0199 \\ 0 \\ 0.0199 \\ 0 \\ 0.0199 \\ 0 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned}
 z_5^{(i)} &= (0.2)(0.0077) + (-0.1)(0.0199) + (0.05)(0) + (0.02) \\
 &\quad (0.0199) + (0.15)(0) + (0)(0.0199) + (0)(0) + (0)(0) \\
 &\quad + 0.01 \\
 &= 0.00154 + 0.00199 + 0.000398 + 0.01 = 0.00995
 \end{aligned}$$

- Untuk D6,

$$x_6^{(i)} = \begin{bmatrix} 0 \\ 0 \\ 0.0334 \\ 0 \\ 0 \\ 0 \\ 0.0442 \\ 0.0442 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned}
 z_6^{(i)} &= (0.2)(0) + (-0.1)(0) + (0.05)(0.0334) + (0.02)(0) \\
 &\quad + (0.15)(0) + (0)(0) + (0)(0.0442) + (0)(0.0442) + 0.01 \\
 &= 0.00167 + 0.01 = 0.01167
 \end{aligned}$$

- Untuk D7,

$$x_7^{(i)} = \begin{bmatrix} 0.0119 \\ 0 \\ 0.0334 \\ 0.0611 \\ 0 \\ 0.0307 \\ 0.0307 \\ 0 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned}
 z_7^{(i)} &= (0.2)(0.0119) + (-0.1)(0) + (0.05)(0.0334) + (0.02) \\
 &\quad (0.0611) + (0.15)(0) + (0)(0.0307) + (0)(0.0307) + (0)(0) \\
 &\quad + 0.01 \\
 &= 0.00238 + 0.00167 + 0.001222 + 0.01 = 0.01527
 \end{aligned}$$

- Untuk D8,

$$x_8^{(i)} = \begin{bmatrix} 0.0172 \\ 0 \\ 0.0334 \\ 0.0442 \\ 0 \\ 0 \\ 0.0442 \\ 0.0442 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_8^{(i)} &= (0.2)(0.0172) + (-0.1)(0) + (0.05)(0.0334) + (0.02) \\ &\quad (0.0442) + (0.15)(0) + (0)(0) + (0)(0.0442) + (0)(0.0442) \\ &\quad + 0.01 \\ &= 0.00344 + 0.00167 + 0.000884 + 0.01 = 0.01599 \end{aligned}$$

- Untuk D9,

$$x_9^{(i)} = \begin{bmatrix} 0.0097 \\ 0 \\ 0.0188 \\ 0 \\ 0 \\ 0.0249 \\ 0 \\ 0.0249 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_9^{(i)} &= (0.2)(0.0097) + (-0.1)(0) + (0.05)(0.0188) + (0.02)(0) \\ &\quad + (0.15)(0) + (0)(0.0249) + (0)(0) + (0)(0.0249) + 0.01 \\ &= 0.00194 + 0.00094 + 0.01 = 0.01288 \end{aligned}$$

- Untuk D10,

$$x_{10}^{(i)} = \begin{bmatrix} 0.0067 \\ 0.0341 \\ 0 \\ 0 \\ 0.0455 \\ 0 \\ 0 \\ 0.0173 \end{bmatrix}$$

Maka skor linier dihitung sebagai:

$$\begin{aligned} z_{10}^{(i)} &= (0.2)(0.0067) + (-0.1)(0.0341) + (0.05)(0) + (0.02)(0) \\ &\quad + (0.15)(0.0455) + (0)(0) + (0)(0) + (0)(0.0173) + 0.01 \\ &= 0.00134 - 0.00341 + 0.006825 + 0.01 = 0.01475 \end{aligned}$$

2. Perhitungan Probabilitas dengan Fungsi Sigmoid *Tensorflow* (Persamaan 2.3)

Tabel 4.15 Perhitungan Fungsi Sigmoid

Dokumen	$z^{(i)}$	Perhitungan Sigmoid	$p^{(i)}$
D1	0.01410	$\frac{1}{1+e^{-0.01410}} = \frac{1}{1+0.9860}$	0.5035
D2	0.0075	$\frac{1}{1+e^{-0.0075}} = \frac{1}{1+0.9923}$	0.5019
D3	0.01277	$\frac{1}{1+e^{-0.01277}} = \frac{1}{1+0.9873}$	0.5032
D4	0.01654	$\frac{1}{1+e^{-0.01654}} = \frac{1}{1+0.9836}$	0.5041
D5	0.00995	$\frac{1}{1+e^{-0.00995}} = \frac{1}{1+0.9901}$	0.5025
D6	0.01167	$\frac{1}{1+e^{-0.01167}} = \frac{1}{1+0.9884}$	0.5029
D7	0.01527	$\frac{1}{1+e^{-0.01527}} = \frac{1}{1+0.9848}$	0.5038
D8	0.01599	$\frac{1}{1+e^{-0.01599}} = \frac{1}{1+0.9841}$	0.5039
D9	0.01288	$\frac{1}{1+e^{-0.01288}} = \frac{1}{1+0.9872}$	0.5032
D10	0.01475	$\frac{1}{1+e^{-0.01475}} = \frac{1}{1+0.9854}$	0.5027

3. Perhitungan *Loss* (*Binary Cross Entropy*) (Persamaan 2.13)

- Untuk D1, label sebenarnya adalah $y_1^{(i)} = 0$ (negatif)

$$\begin{aligned} J_1^{(i)} &= -[0 \cdot \ln(0.5035) + (1 - 0) \ln(1 - 0.5035)] \\ &= -[\ln(1 - 0.5035)] \end{aligned}$$

$$= -\ln(0.4965)$$

$$= -(-0.70004) = 0.70004$$

- Untuk D2, label sebenarnya adalah $y_2^{(i)} = 0$ (negatif)

$$J_2^{(i)} = -[0 \cdot \ln(0.5019) + (1 - 0) \ln(1 - 0.5019)]$$

$$= -[\ln(1 - 0.5019)]$$

$$= -\ln(0.4981)$$

$$= -(-0.69697) = 0.69697$$

- Untuk D3, label sebenarnya adalah $y_3^{(i)} = 1$ (positif)

$$J_3^{(i)} = -[1 \cdot \ln(0.5032) + (1 - 1) \ln(1 - 0.5032)]$$

$$= -\ln(0.5032)$$

$$= -(-0.68616) = 0.68616$$

- Untuk D4, label sebenarnya adalah $y_4^{(i)} = 0$ (negatif)

$$J_4^{(i)} = -[0 \cdot \ln(0.5041) + (1 - 0) \ln(1 - 0.5041)]$$

$$= -[\ln(1 - 0.5041)]$$

$$= -\ln(0.4959)$$

$$= -(-0.70138) = 0.70138$$

- Untuk D5, label sebenarnya adalah $y_5^{(i)} = 1$ (positif)

$$J_5^{(i)} = -[1 \cdot \ln(0.5025) + (1 - 1) \ln(1 - 0.5025)]$$

$$= -\ln(0.5025)$$

$$= -(-0.68764) = 0.68764$$

- Untuk D6, label sebenarnya adalah $y_6^{(i)} = 1$ (positif)

$$J_6^{(i)} = -[1 \cdot \ln(0.5029) + (1 - 1) \ln(1 - 0.5029)]$$

$$= -\ln(0.5029)$$

$$= -(-0.68689) = 0.68689$$

- Untuk D7, label sebenarnya adalah $y_7^{(i)} = 1$ (positif)

$$\begin{aligned} J_7^{(i)} &= -[1 \cdot \ln(0.5038) + (1 - 1) \ln(1 - 0.5038)] \\ &= -\ln(0.5038) \\ &= -(-0.68536) = 0.68536 \end{aligned}$$

- Untuk D8, label sebenarnya adalah $y_8^{(i)} = 1$ (positif)

$$\begin{aligned} J_8^{(i)} &= -[1 \cdot \ln(0.5038) + (1 - 1) \ln(1 - 0.5038)] \\ &= -\ln(0.5038) \\ &= -(-0.68516) = 0.68516 \end{aligned}$$

- Untuk D9, label sebenarnya adalah $y_9^{(i)} = 1$ (positif)

$$\begin{aligned} J_9^{(i)} &= -[1 \cdot \ln(0.5032) + (1 - 1) \ln(1 - 0.5032)] \\ &= -\ln(0.5032) \\ &= -(-0.68676) = 0.68676 \end{aligned}$$

- Untuk D10, label sebenarnya adalah $y_{10}^{(i)} = 0$ (negatif)

$$\begin{aligned} J_{10}^{(i)} &= -[0 \cdot \ln(0.5037) + (1 - 0) \ln(1 - 0.5037)] \\ &= -[\ln(1 - 0.5037)] \\ &= -\ln(0.4963) \\ &= -(-0.70041) = 0.70041 \end{aligned}$$

4. Perhitungan Gradien (Persamaan 2.15 dan Persamaan 2.16)

Untuk satu dokumen D_1 , gradien keseluruhan didefinisikan sebagai:

$$\nabla J_1 = \left(\frac{\partial J_1}{\partial w_1}, \frac{\partial J_1}{\partial w_2}, \dots, \frac{\partial J_1}{\partial w_8}, \frac{\partial J_1}{\partial b} \right)$$

Gradien ini terdiri dari turunan parsial terhadap setiap bobot w_j sebagaimana dinyatakan pada Persamaan (2.15), serta turunan parsial terhadap bias b sebagaimana pada Persamaan (2.16).

Dengan vektor TF-IDF untuk dokumen D_1 :

$$x_1^{(i)} = \begin{bmatrix} 0.0091 \\ 0.0234 \\ 0 \\ 0 \\ 0.0308 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

dan probabilitas prediksi $p^{(1)} = 0.5035$ serta label $y^{(1)} = 0$, maka komponen gradien dihitung dengan Persamaan 2.15 dan Persamaan 2.16.

Tabel 4.16 Gradien pada D1

Fitur	$\frac{\partial J_1}{\partial w_j} = (p^{(1)} - y^{(1)})x_j^{(1)}$	Hasil
aplikasi	$\frac{\partial J_1}{\partial w_1} = (0.5035 - 0)0.0091$	0.00458
daftar	$\frac{\partial J_1}{\partial w_2} = (0.5035 - 0)0.0234$	0.01178
mudah	$\frac{\partial J_1}{\partial w_3} = (0.5035 - 0)0$	0
bantu	$\frac{\partial J_1}{\partial w_4} = (0.5035 - 0)0$	0
susah	$\frac{\partial J_1}{\partial w_5} = (0.5035 - 0)0.0308$	0.01551
faskes	$\frac{\partial J_1}{\partial w_6} = (0.5035 - 0)0$	0
terimakasih	$\frac{\partial J_1}{\partial w_7} = (0.5035 - 0)0$	0
jkn	$\frac{\partial J_1}{\partial w_8} = (0.5035 - 0)0$	0
Bias b	$\frac{\partial J_1}{\partial b} = 0.5035 - 0$	0.5035

Berdasarkan nilai turunan parsial pada Tabel 4.16, seluruh komponen tersebut dapat direpresentasikan dalam bentuk *vektor gradien* menggunakan simbol nabla (∇). Gradien merupakan vektor yang memuat seluruh turunan parsial fungsi *loss* terhadap setiap parameter w_j , sehingga untuk dokumen D_1 , vektor gradien dirumuskan sebagai:

$$\nabla J_1 = \begin{bmatrix} 0.00458 \\ 0.01178 \\ 0 \\ 0 \\ 0.01551 \\ 0 \\ 0 \\ 0 \\ 0.5035 \end{bmatrix}$$

Gradien ini menunjukkan arah perubahan parameter yang akan menurunkan nilai *loss*. Seluruh komponen gradien digunakan pada langkah pembaruan parameter menggunakan algoritma *gradient descent*.

Keterangan: Indeks fitur dan bobot w_j pada tabel ini dimulai dari $j = 1$ hingga $j = 8$, sesuai dengan definisi vektor fitur dan vektor parameter pada Persamaan (2.1) dan (2.2). Bias tidak dinyatakan sebagai w_0 , tetapi sebagai parameter terpisah b .

Perhitungan turunan parsial *loss* terhadap setiap bobot pada Dokumen D1 ditunjukkan secara rinci pada Tabel 4.16. Untuk dokumen lainnya (D2–D10), proses perhitungannya dilakukan dengan cara yang sama, yaitu menggunakan Persamaan 2.15 dan Persamaan 2.16. Namun, mengingat jumlah fitur dan dokumen yang jauh lebih besar pada dataset sebenarnya, perhitungan turunan dan pembaruan bobot untuk seluruh data dilakukan secara otomatis menggunakan bantuan perangkat lunak *Python* dengan library *TensorFlow*, sehingga perhitungan manual hanya ditampilkan pada satu contoh dokumen sebagai ilustrasi.

5. Pembaruan Bobot dan Bias (Persamaan 2.17)

Bobot dan bias diperbarui menggunakan aturan *gradient descent*. Dengan \mathbf{w} adalah bobot awal, bias $b = 0.01$, dan $\eta = 0.1$.

- Untuk D1, sebagai ilustrasi perhitungan pembaruan bobot w_j , menggunakan Persamaan (2.17) diperoleh:

$$w^{new} = w^{lama} - 0.1\nabla J_1$$

$$w^{new} = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.05 \\ 0.02 \\ 0.15 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 0.1 \begin{bmatrix} 0.00458 \\ 0.01178 \\ 0 \\ 0 \\ 0.01551 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$w^{new} = \begin{bmatrix} 0.2 \\ -0.1 \\ 0.05 \\ 0.02 \\ 0.15 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.000458 \\ 0.001178 \\ 0 \\ 0 \\ 0.001551 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.199542 \\ -0.101178 \\ 0.050000 \\ 0.020000 \\ 0.148449 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$b^{new} = 0.01 - 0.1(0.5035) = -0.04035$$

Untuk dokumen D2 hingga D10, proses pembaruan bobot dan bias dilakukan dengan langkah yang sama seperti pada D1. Hasil hasil turunan tersebut digunakan untuk memperbarui parameter model melalui aturan *gradient descent* pada Persamaan (2.17). Seluruh prosedur ini menghasilkan pembaruan bobot dan bias pada setiap iterasi pelatihan. Namun, untuk menghindari repetisi yang panjang, perhitungan manual secara rinci ditampilkan hanya pada D1 sebagai ilustrasi.

Proses di atas menggambarkan mekanisme dasar pelatihan Regresi Logistik menggunakan *gradient descent* sebagaimana dijelaskan pada kajian teori. Pada implementasi sebenarnya, seluruh operasi ini dilakukan secara otomatis oleh TensorFlow untuk seluruh dokumen dalam dataset pada setiap iterasi pelatihan,

sehingga model dapat memperbarui parameter secara efisien dan mencapai konvergensi secara optimal.

4.1.10 Evaluasi Model

1. Proses Pelatihan Model

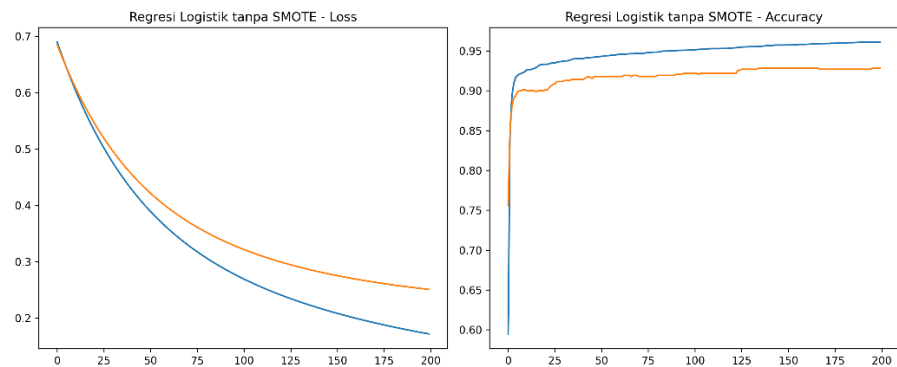
Pelatihan model Regresi Logistik dilakukan untuk mengoptimalkan bobot parameter sehingga model mampu mempelajari pola teks pada ulasan positif dan negatif. Pada tahap ini, evaluasi belum difokuskan pada performa prediksi, melainkan pada konsistensi proses pelatihan. Model dilatih menggunakan representasi TF-IDF dengan 4689 fitur sebagai input pada satu lapisan *dense* beraktivasi sigmoid. Optimasi dilakukan menggunakan *optimizer* Adam dan *loss function* *Binary Cross-Entropy*. Pelatihan dijalankan hingga 200 epoch dengan mekanisme *Early Stopping* (*patience* = 10) agar model berhenti ketika performa validasi tidak mengalami peningkatan lebih lanjut.

Pada skenario tanpa SMOTE, jumlah data latih adalah 7778 sampel dengan distribusi kelas tidak seimbang (positif lebih banyak). Pada skenario dengan SMOTE, jumlah data latih meningkat menjadi 9506 sampel setelah proses oversampling pada kelas negatif. Pada kedua skenario tersebut, *Early Stopping* menghentikan pelatihan secara otomatis ketika nilai *validation loss* berhenti membaik. Tahap ini memastikan bahwa pelatihan berlangsung stabil sebelum model dievaluasi menggunakan *learning curve*, confusion matrix, dan metrik evaluasi.

2. *Learning curve* (*Loss* dan *Accuracy*)

Learning curve digunakan untuk menggambarkan dinamika pelatihan berdasarkan perubahan *training loss*, *validation loss*, *training accuracy*, dan *validation accuracy* sepanjang epoch. Hasil *learning curve* menunjukkan bagaimana model beradaptasi terhadap data dan apakah terdapat indikasi *overfitting* atau *underfitting*.

a. *Learning curve* tanpa SMOTE

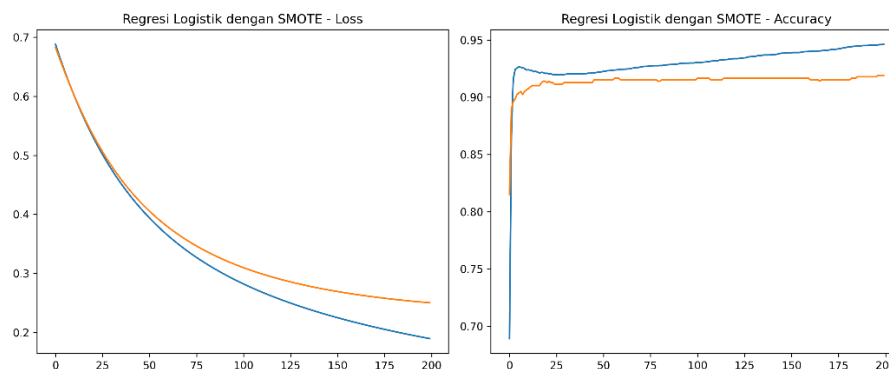


Gambar 4.9 *Learning curve* tanpa SMOTE

Gambar 4.9 menunjukkan bahwa *training loss* turun lebih cepat dan mencapai nilai yang lebih rendah dibandingkan *validation loss*, sehingga terbentuk gap *loss* yang konsisten hingga akhir *epoch*. Kondisi ini menandakan bahwa model menyesuaikan diri lebih baik pada data latih dibandingkan data validasi. Pada grafik akurasi, terlihat gap yang jelas dan bertahan antara *training accuracy* dan *validation accuracy*. *Training accuracy* terus meningkat hingga mencapai sekitar 0.96, sedangkan *validation accuracy* berhenti pada kisaran sekitar 0.92–0.93 dan tidak mengikuti peningkatan selanjutnya. Gap akurasi yang tidak mengecil ini menunjukkan keterbatasan kemampuan generalisasi model, yang secara tegas

dipengaruhi oleh ketidakseimbangan distribusi kelas pada data tanpa SMOTE.

b. *Learning curve* dengan SMOTE



Gambar 4.10 *Learning curve* dengan SMOTE

Gambar 4.10 menunjukkan bahwa *training loss* dan *validation loss* menurun dengan pola yang hampir sejajar, sehingga tidak terbentuk gap *loss* yang berarti sepanjang proses pelatihan. Hal ini menandakan bahwa model belajar dengan tingkat kesesuaian yang relatif seimbang antara data latih dan data validasi. Pada grafik akurasi, *training accuracy* dan *validation accuracy* meningkat dan kemudian stabil dengan jarak yang lebih kecil dibandingkan model tanpa SMOTE. *Training accuracy* berada pada kisaran 0.94–0.95, sementara *validation accuracy* stabil di kisaran 0.91–0.92. Gap akurasi tetap ada, tetapi lebih terkendali dan tidak melebar, yang menunjukkan bahwa penerapan SMOTE mengurangi ketimpangan performa antara data latih dan data validasi serta memperbaiki konsistensi generalisasi model.

3. Evaluasi dengan *Confusion Matrix*

Confusion matrix digunakan untuk menggambarkan kualitas prediksi model terhadap data uji berdasarkan empat komponen utama: *True*

Positive (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Matriks ini memberikan informasi rinci mengenai jenis kesalahan klasifikasi yang dilakukan model, sehingga memudahkan proses interpretasi dan identifikasi kelemahan model. Evaluasi dilakukan menggunakan 1948 data uji yang terdiri dari 1176 ulasan positif dan 772 ulasan negatif.

Berikut merupakan hasil *confusion matrix* yang diperoleh dari model klasifikasi menggunakan algoritma Regresi Logistik menggunakan data *testing*. Hasil tersebut ditampilkan pada Tabel 4.17 *confusion matrix* tanpa SMOTE.

Tabel 4.17 *Confusion Matrix* Klasifikasi Data Testing tanpa SMOTE

Label		Hasil Prediksi	
		Negatif	Positif
Data Aktual	Negatif	729	43
	Positif	90	1086

a. Akurasi (Persamaan 2.23)

$$Akurasi = \frac{1086 + 729}{1086 + 729 + 43 + 90}$$

$$Akurasi = \frac{1815}{1948} = 0.9319$$

b. Presisi (Persamaan 2.24)

$$\text{Presisi untuk kelas positif} = \frac{1086}{1086+43} = \frac{1086}{1129} = 0.9620$$

$$\text{Presisi untuk kelas negatif} = \frac{729}{729+90} = \frac{729}{819} = 0.8901$$

$$\text{Presisi } macro \text{ avg} = \frac{0.9620+0.8901}{2} = 0.9261$$

$$\text{Presisi } weighted \text{ avg} = \frac{(0.9620 \times 1176) + (0.8901 \times 772)}{1948} = 0.9338$$

c. *Recall* (Persamaan 2.25)

$$\text{Recall untuk kelas positif} = \frac{1086}{1086+90} = \frac{1086}{1175} = 0.9235$$

$$\text{Recall untuk kelas negatif} = \frac{729}{729+43} = \frac{729}{772} = 0.9443$$

$$\text{Recall macro avg} = \frac{0.9235+0.9443}{2} = 0.9339$$

$$\text{Recall weighted avg} = 0.9319$$

d. *F1-score* (Persamaan 2.26)

$$\text{F1-score untuk kelas positif} = 2 \cdot \frac{0.9620 \times 0.9235}{0.9620 + 0.9235} = 0.9423$$

$$\text{F1-score untuk kelas negatif} = 2 \cdot \frac{0.8901 \times 0.9443}{0.8901 + 0.9443} = 0.9165$$

$$\text{F1-score macro avg} = \frac{0.9423 + 0.9165}{2} = 0.9294$$

$$\text{F1-score weighted avg} = \frac{(0.9423 \times 1176) + (0.9165 \times 772)}{1948} = 0.9321$$

Confusion matrix pada skenario SMOTE digunakan untuk memberikan gambaran lebih detail mengenai distribusi prediksi model setelah dilakukan penyeimbangan kelas pada data latih. Tabel 4.18 *confusion matrix* dengan SMOTE berikut.

Tabel 4.18 *Confusion Matrix* Klasifikasi Data Testing dengan SMOTE

Label		Hasil Prediksi	
		Negatif	Positif
Data Aktual	Negatif	746	26
	Positif	106	1070

a. Akurasi (Persamaan 2.23)

$$\text{Akurasi} = \frac{1070 + 756}{1070 + 746 + 26 + 106}$$

$$\text{Akurasi} = \frac{1816}{1946} = 0.9322$$

b. Presisi (Persamaan 2.24)

$$\text{Presisi untuk kelas positif} = \frac{1070}{1070+26} = \frac{1070}{1096} = 0.9763$$

$$\text{Presisi untuk kelas negatif} = \frac{746}{746+106} = \frac{746}{852} = 0.8756$$

$$\text{Presisi macro avg} = \frac{0.9763+0.8756}{2} = 0.9260$$

$$\text{Presisi weighted avg} = \frac{(0.9763 \times 1176) + (0.8756 \times 772)}{1948} = 0.9364$$

c. *Recall* (Persamaan 2.25)

$$\text{Recall untuk kelas positif} = \frac{1071}{1070+106} = \frac{1070}{1176} = 0.9099$$

$$\text{Recall untuk kelas negatif} = \frac{746}{746+26} = \frac{746}{772} = 0.9663$$

$$\text{Recall macro avg} = \frac{0.9099+0.9663}{2} = 0.9281$$

$$\text{Recall weighted avg} = 0.9322$$

d. *F1-score* (Persamaan 2.26)

$$\text{F1-score untuk kelas positif} = 2 \cdot \frac{0.9763 \times 0.9099}{0.9763 + 0.9099} = 0.9419$$

$$\text{F1-score untuk kelas negatif} = 2 \cdot \frac{0.8756 \times 0.9663}{0.8756 + 0.9663} = 0.9188$$

$$\text{F1-score macro avg} = \frac{0.9419+0.9188}{2} = 0.9304$$

$$\text{F1-score weighted avg} = \frac{(0.9419 \times 1176) + (0.9188 \times 772)}{1948} = 0.9328$$

Tabel 4.19 Hasil Perbandingan Evaluasi dengan dan tanpa SMOTE

Evaluasi		Tanpa SMOTE	Dengan SMOTE
Akurasi		0.9319	0.9322
Presisi	Positif	0.9620	0.9763
	Negatif	0.8901	0.8756
	Macro Avg	0.9261	0.9260
	Weighted Avg	0.9338	0.9364
Recall	Positif	0.9235	0.9099
	Negatif	0.9443	0.9663
	Macro Avg	0.9339	0.9381
	Weighted Avg	0.9319	0.9322
F1-score	Positif	0.9423	0.9419
	Negatif	0.9165	0.9188
	Macro Avg	0.9294	0.9304
	Weighted Avg	0.9321	0.9328

Pada Tabel 4.19 di atas menunjukkan hasil perbandingan evaluasi dengan dan tanpa *oversampling* SMOTE.

4.2 Pembahasan

4.2.1 Skenario Tanpa SMOTE dan Dengan SMOTE

Pembahasan ini menguraikan secara komprehensif perbedaan performa model Regresi Logistik pada dua skenario utama, yaitu tanpa SMOTE dan dengan SMOTE. Evaluasi dilakukan berdasarkan stabilitas *learning curve*, pola distribusi prediksi pada *confusion matrix*, serta perubahan signifikan pada metrik evaluasi yang mencerminkan kualitas klasifikasi pada masing-masing kelas. Dataset awal memiliki ketidakseimbangan yang cukup besar, dengan jumlah ulasan positif jauh lebih tinggi dibanding ulasan negatif, sehingga model pada skenario tanpa SMOTE berpotensi mempelajari pola sentimen secara tidak seimbang. Hal ini menjadi dasar penting untuk menilai dampak penerapan SMOTE, yang bertujuan untuk menyeimbangkan distribusi kelas dan memperbaiki representasi pola sentimen negatif dalam proses pelatihan.

Pada skenario tanpa SMOTE, *learning curve* menunjukkan bahwa *training loss* secara konsisten berada di bawah *validation loss* hingga akhir *epoch*. Pola ini membentuk *gap loss* yang tidak mengecil, yang menandakan bahwa model lebih optimal menyesuaikan diri pada data latih dibandingkan data validasi. Kondisi serupa terlihat pada kurva akurasi, di mana *training accuracy* meningkat hingga mencapai sekitar 0.96, sedangkan *validation accuracy* berhenti pada kisaran 0.92–0.93. Gap akurasi yang bertahan ini menunjukkan bahwa peningkatan performa pada data latih tidak sepenuhnya diikuti oleh peningkatan performa pada data

validasi, sehingga kemampuan generalisasi model masih terbatas akibat ketidakseimbangan kelas.

Temuan dari *learning curve* tersebut diperkuat oleh hasil *confusion matrix* tanpa SMOTE. Model mampu mengklasifikasikan 729 ulasan negatif dan 1086 ulasan positif secara benar, namun masih menghasilkan 43 *false positive* dan 90 *false negative*. Jumlah *false negative* yang relatif besar menunjukkan bahwa model masih gagal mengenali sebagian ulasan positif secara tepat. Kondisi ini tercermin pada metrik evaluasi, di mana presisi kelas positif (0.9620) lebih tinggi dibandingkan presisi kelas negatif (0.8901), serta *F1-score* kelas positif (0.9423) lebih tinggi dibandingkan *F1-score* kelas negatif (0.9165). Meskipun akurasi keseluruhan mencapai 0.9319, perbedaan nilai metrik antar kelas menunjukkan bahwa performa model belum seimbang.

Setelah SMOTE diterapkan, karakteristik pembelajaran model mengalami perubahan yang jelas. *Learning curve* menunjukkan bahwa *training loss* dan *validation loss* bergerak lebih sejajar tanpa pelebaran jarak yang mencolok hingga akhir *epoch*. Hal ini menandakan bahwa proses pelatihan menjadi lebih seimbang antara data latih dan data validasi. Pada kurva akurasi, jarak antara *training accuracy* dan *validation accuracy* juga menjadi lebih terkendali dibandingkan skenario tanpa SMOTE. Pola ini menunjukkan bahwa penyeimbangan kelas membantu model mempelajari representasi data secara lebih merata.

Perubahan tersebut berdampak langsung pada distribusi prediksi yang ditunjukkan oleh *confusion matrix* dengan SMOTE. Jumlah *true negative* meningkat dari 729 menjadi 746, sementara *false positive* menurun dari 43 menjadi 26. Peningkatan ini menunjukkan bahwa model menjadi lebih baik dalam

mengenali ulasan negatif. Namun, pada saat yang sama, jumlah *false negative* meningkat dari 90 menjadi 106, yang menandakan bahwa sebagian ulasan positif lebih sering salah diklasifikasikan. Kondisi ini menunjukkan adanya mekanisme *trade-off* antara kemampuan model dalam mengenali kelas negatif dan sensitivitas terhadap kelas positif.

Trade-off tersebut terlihat jelas pada nilai metrik evaluasi. *Recall* kelas negatif meningkat dari 0.9443 menjadi 0.9663, sedangkan *recall* kelas positif menurun dari 0.9235 menjadi 0.9099. *F1-score* kelas negatif juga mengalami kenaikan dari 0.9165 menjadi 0.9188, sementara *F1-score* kelas positif sedikit menurun dari 0.9423 menjadi 0.9419. Kenaikan *F1-score macro average* dari 0.9294 menjadi 0.9304 menunjukkan bahwa performa antar kelas menjadi lebih merata, meskipun terjadi penurunan kecil pada sensitivitas kelas positif. Dengan demikian, penerapan SMOTE tidak menghilangkan kesalahan klasifikasi, tetapi menggeser distribusi kesalahan agar tidak terpusat pada satu kelas saja.

Secara keseluruhan, skenario tanpa SMOTE menghasilkan model dengan performa tinggi tetapi masih menunjukkan ketimpangan generalisasi dan bias terhadap distribusi kelas awal. Sebaliknya, penerapan SMOTE menghasilkan model yang lebih seimbang melalui mekanisme *trade-off* antara kelas positif dan negatif. *Trade-off* ini menyebabkan peningkatan kemampuan model dalam mendeteksi ulasan negatif, meskipun sebagian kemampuan mendeteksi ulasan positif berkurang. Dalam konteks analisis sentimen aplikasi Mobile JKN, kemampuan mengenali ulasan negatif yang merepresentasikan keluhan pengguna merupakan aspek penting, sehingga skenario dengan SMOTE dinilai lebih relevan dan representatif untuk evaluasi kualitas layanan.

4.2.2 Uji Pengaruh Ukuran Data terhadap Performa Model

Uji pengaruh ukuran data dilakukan untuk menganalisis bagaimana variasi proporsi data latih memengaruhi performa model Regresi Logistik dalam mengklasifikasikan sentimen ulasan Mobile JKN. Lima proporsi data latih diuji—20%, 40%, 60%, 80%, dan 100%—baik pada model tanpa SMOTE maupun dengan SMOTE. Seluruh pengujian tetap menggunakan konfigurasi pembagian data utama sebesar 80% untuk pelatihan dan 20% untuk pengujian; sehingga nilai 100% *train* merujuk pada penggunaan seluruh data latih yang tersedia, bukan seluruh dataset. Hasil pengujian ditampilkan pada Tabel 4.20 dan Tabel 4.21.

Tabel 4.20 Hasil Uji Proporsi Data

Proporsi <i>Train</i>	Evaluasi	Tanpa SMOTE	Dengan SMOTE
20%	Akurasi	0.9068	0.9148
	Presisi	0.9163	0.9603
	Recall	0.9306	0.8959
	F1-score	0.9234	0.9270
40%	Akurasi	0.9191	0.9237
	Presisi	0.9353	0.9700
	Recall	0.9306	0.9015
	F1-score	0.9328	0.9345
60%	Akurasi	0.9218	0.9290
	Presisi	0.9405	0.9774
	Recall	0.9293	0.9032
	F1-score	0.9348	0.9388
80%	Akurasi	0.9266	0.9293
	Presisi	0.9492	0.9781
	Recall	0.9281	0.9031
	F1-score	0.9385	0.9391
100%	Akurasi	0.9284	0.9317
	Presisi	0.9524	0.9780
	Recall	0.9279	0.9073
	F1-score	0.9400	0.9413

Berdasarkan Tabel 4.20, model tanpa SMOTE menunjukkan kecenderungan peningkatan performa seiring bertambahnya proporsi data latih. Akurasi meningkat dari 0.9068 pada proporsi 20% menjadi 0.9284 pada proporsi

100%. Presisi dan *F1-score* juga mengalami kenaikan bertahap pada setiap penambahan proporsi data. Sementara itu, nilai *recall* berada pada kisaran yang relatif stabil, yaitu antara 0.9279 hingga 0.9306. Pola ini menunjukkan bahwa penambahan data latih terutama berkontribusi pada peningkatan ketepatan prediksi, namun tidak banyak mengubah kemampuan model dalam menangkap seluruh sampel kelas positif.

Hasil pada Tabel 4.20 memperlihatkan bahwa model dengan SMOTE memiliki performa yang lebih stabil terhadap variasi ukuran data latih. Pada proporsi 20% data latih, model sudah mencapai akurasi 0.9148 dan *F1-score* 0.9270, dan nilai tersebut meningkat secara bertahap hingga mencapai akurasi 0.9317 dan *F1-score* 0.9413 pada proporsi 100%. Perubahan nilai metrik pada setiap penambahan proporsi data relatif kecil dibandingkan skenario tanpa SMOTE, yang menunjukkan bahwa penyeimbangan kelas membantu model mempertahankan performa yang konsisten meskipun jumlah data latih berubah.

Perbedaan pola antara kedua skenario juga terlihat pada metrik presisi dan *recall*. Model dengan SMOTE menunjukkan nilai presisi yang tinggi dan konsisten pada seluruh proporsi data latih, yaitu pada rentang 0.9603–0.9781, sementara nilai *recall* berada pada rentang 0.8959–0.9073. Pola ini mengindikasikan bahwa model dengan SMOTE cenderung lebih selektif dalam memberikan prediksi positif. Sebaliknya, model tanpa SMOTE memiliki presisi yang lebih rendah namun *recall* yang relatif stabil, yang mencerminkan pengaruh distribusi kelas yang tidak seimbang terhadap proses pembelajaran.

Jika kedua tabel dibandingkan, terlihat bahwa penambahan ukuran data latih memberikan perubahan performa yang lebih jelas pada model tanpa SMOTE,

sedangkan model dengan SMOTE menunjukkan perubahan yang lebih landai. Hal ini menunjukkan bahwa penyeimbangan kelas melalui SMOTE mengurangi ketergantungan model terhadap jumlah data latih, sehingga performa model tidak berubah secara drastis ketika proporsi data latih ditingkatkan.

Hasil uji proporsi data ini selaras dengan temuan pada analisis *learning curve* sebelumnya. Meskipun performa tertinggi diperoleh ketika seluruh data latih yang tersedia digunakan, kondisi tersebut tidak merepresentasikan skema evaluasi yang wajar, karena dalam praktik pembelajaran mesin tidak dimungkinkan menggunakan 100% data untuk pelatihan dan 0% data untuk pengujian. Oleh karena itu, proporsi 100% pada pengujian ini dipahami sebagai pemanfaatan penuh bagian data latih yang telah ditetapkan sebelumnya, bukan sebagai penggunaan seluruh dataset tanpa data uji.

Konfigurasi pembagian data sebesar 80% untuk pelatihan dan 20% untuk pengujian tetap dipertahankan sebagai skema utama karena memberikan keseimbangan antara kecukupan data latih dan objektivitas evaluasi performa model. Skema ini memungkinkan model mempelajari pola data secara memadai sekaligus menyediakan data uji yang independen untuk mengukur kemampuan generalisasi. Dengan demikian, pengujian variasi proporsi data latih dilakukan untuk menggambarkan perilaku model secara deskriptif terhadap perubahan ukuran data, tanpa dimaksudkan untuk menggantikan atau melampaui skema evaluasi utama yang digunakan dalam penelitian.

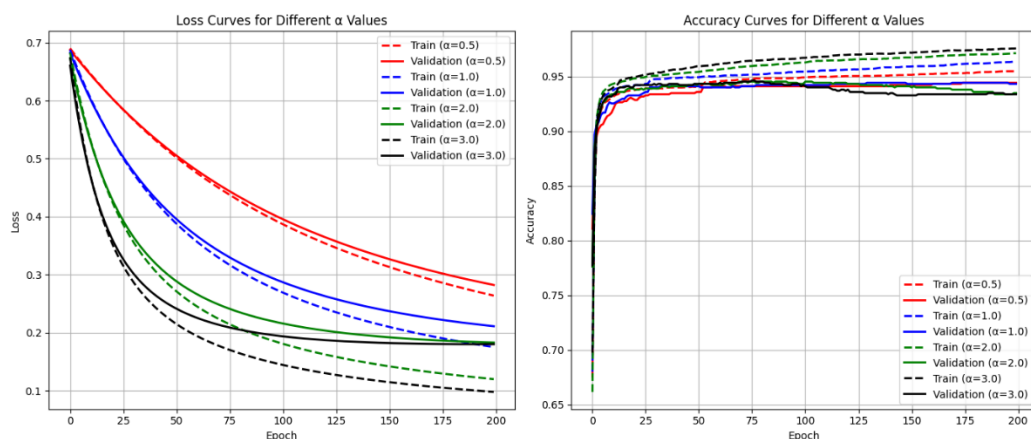
4.2.3 Pembahasan Hasil Evaluasi Variasi Parameter Temperatur (α) pada Fungsi Sigmoid

Parameter kecuraman α pada fungsi sigmoid berperan sebagai faktor penskalaan terhadap nilai logit sebelum dikonversi menjadi probabilitas. Secara matematis, keberadaan α memengaruhi besar gradien karena turunan fungsi *loss* terhadap parameter memuat faktor α sebagai pengali. Dengan demikian, perubahan nilai α tidak hanya mengubah bentuk kurva sigmoid, tetapi juga memengaruhi kecepatan konvergensi selama proses optimisasi. Oleh sebab itu, evaluasi variasi α dilakukan untuk melihat bagaimana sensitivitas fungsi aktivasi berdampak terhadap stabilitas pelatihan dan performa klasifikasi.

Pada penelitian ini diuji empat nilai α , yaitu 0.5, 1.0, 2.0, dan 3.0, dengan $\alpha = 1.0$ sebagai *baseline* regresi logistik standar. Analisis dilakukan melalui dua pendekatan, yaitu evaluasi dinamika pelatihan menggunakan grafik *learning curve* (Gambar 4.11) serta evaluasi kuantitatif menggunakan metrik klasifikasi pada Tabel 4.21. Pendekatan ini memungkinkan interpretasi yang lebih komprehensif karena mempertimbangkan stabilitas konvergensi dan performa akhir secara simultan. Dengan demikian, penentuan konfigurasi optimal tidak hanya didasarkan pada satu indikator saja.

Berdasarkan grafik *loss* pada Gambar 4.11, konfigurasi $\alpha = 0.5$ menunjukkan penurunan *loss* paling lambat dan berhenti pada nilai yang relatif lebih tinggi dibandingkan konfigurasi lainnya. Kondisi ini konsisten dengan teori bahwa nilai α yang kecil menghasilkan gradien yang lebih kecil, sehingga pembaruan bobot berlangsung lebih lambat. Akibatnya, proses konvergensi

menjadi kurang efisien dan model membutuhkan lebih banyak epoch untuk mendekati titik minimum.



Gambar 4.11 *Learning curve* Hasil Evaluasi Variasi Parameter Sigmoid

Pola ini juga tercermin pada Tabel 4.21, di mana $\alpha = 0.5$ menghasilkan akurasi dan F1-score terendah dibandingkan konfigurasi lain.

Tabel 4.21 Evaluasi Hasil Variasi Parameter Sigmoid

α	Akurasi	Presisi	Recall	F1-score
0.5	0.9152	0.9680	0.8907	0.9277
1.0	0.9168	0.9689	0.8923	0.9291
2.0	0.9193	0.9615	0.9041	0.9319
3.0	0.9193	0.9574	0.9083	0.9322

Ketika α ditingkatkan menjadi 1.0 dan 2.0, grafik menunjukkan percepatan konvergensi yang signifikan. Pada $\alpha = 2.0$, jarak antara *training loss* dan *validation loss* terlihat sangat kecil dan stabil hingga akhir epoch. Hal ini menunjukkan bahwa model tidak hanya belajar lebih cepat, tetapi juga mempertahankan kemampuan generalisasi yang baik tanpa indikasi *overfitting* yang berarti. Temuan ini didukung oleh Tabel 4.21, di mana $\alpha = 2.0$ menghasilkan akurasi tertinggi sebesar 0.9193 dan F1-score sebesar 0.9319.

Pada $\alpha = 3.0$, *training loss* mencapai nilai paling rendah dibandingkan seluruh konfigurasi. Secara numerik, kondisi ini menunjukkan bahwa model

mampu meminimalkan *error* pada data pelatihan secara lebih intensif. Namun, *validation loss* pada grafik menunjukkan variasi yang sedikit lebih besar dibandingkan $\alpha = 2.0$, meskipun tetap dalam rentang yang stabil. Hal ini mengindikasikan bahwa peningkatan kecuraman yang lebih tinggi mulai meningkatkan sensitivitas model terhadap variasi kecil pada data validasi.

Analisis grafik akurasi pada Gambar 4.11 memperlihatkan bahwa seluruh konfigurasi mencapai akurasi tinggi secara cepat, tetapi $\alpha = 2.0$ dan $\alpha = 3.0$ konsisten berada pada posisi tertinggi. Kurva validasi $\alpha = 2.0$ tampak lebih halus dan stabil, sedangkan $\alpha = 3.0$ sedikit lebih berfluktuasi meskipun tetap berada pada tingkat akurasi yang sama. Stabilitas ini menjadi faktor penting dalam menilai kualitas generalisasi model. Dengan demikian, grafik memberikan indikasi bahwa $\alpha = 2.0$ memiliki keseimbangan yang lebih baik dalam dinamika pelatihan.

Jika merujuk secara langsung pada Tabel 4.21, terlihat bahwa $\alpha = 3.0$ menghasilkan *F1-score* tertinggi sebesar 0.9322. Namun selisihnya terhadap $\alpha = 2.0$ yang sebesar 0.9319 hanya 0.0003, yang secara praktis hampir tidak signifikan. Selain itu, karakteristik metrik menunjukkan perbedaan distribusi, di mana $\alpha = 2.0$ memiliki presisi lebih tinggi (0.9615) sementara $\alpha = 3.0$ memiliki recall lebih tinggi (0.9083). Perbedaan ini menunjukkan bahwa $\alpha = 3.0$ sedikit lebih agresif dalam mendeteksi kelas positif, tetapi dengan konsekuensi penurunan presisi.

Berdasarkan integrasi analisis grafik dan tabel, dapat ditegaskan bahwa peningkatan α hingga 2.0 memberikan manfaat nyata dalam percepatan konvergensi dan stabilitas validasi. Peningkatan lebih lanjut ke $\alpha = 3.0$ hanya

memberikan keuntungan *marginal* pada *F1-score* tanpa peningkatan akurasi tambahan. Secara numerik murni, $\alpha = 3.0$ memang menghasilkan *F1-score* tertinggi. Namun secara stabilitas pelatihan dan keseimbangan metrik, $\alpha = 2.0$ menunjukkan performa yang lebih konsisten.

Dengan demikian, kesimpulan yang objektif adalah bahwa $\alpha = 3.0$ unggul tipis secara kuantitatif, sedangkan $\alpha = 2.0$ lebih optimal secara praktis karena menawarkan kombinasi stabilitas konvergensi dan performa yang seimbang. Perbedaan yang sangat kecil di antara keduanya menunjukkan bahwa pengaruh α dalam konteks dataset yang telah diseimbangkan menggunakan SMOTE bersifat sebagai parameter *fine-tuning*. Oleh karena itu, pemilihan nilai α perlu mempertimbangkan prioritas tujuan model, apakah menekankan pada peningkatan recall atau pada stabilitas dan keseimbangan performa secara keseluruhan.

4.2.4 Pembahasan Perbandingan dengan Metode Perbandingan

Untuk memperoleh gambaran yang lebih komprehensif mengenai posisi performa model *Regresi Logistik* dalam penelitian ini, dilakukan perbandingan dengan tiga algoritma perbandingan yang umum digunakan dalam analisis sentimen berbasis teks, yaitu *Linear Support Vector Machine* (SVM), *Decision Tree*, dan *K-Nearest Neighbors* (KNN). Seluruh model dilatih menggunakan representasi fitur (TF-IDF) dan diuji pada dataset yang sama, sehingga perbedaan performa dapat diamati secara adil dan berada dalam kondisi evaluasi yang setara. Evaluasi dilakukan menggunakan empat metrik utama, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*, untuk menggambarkan karakteristik performa masing-masing model dari sisi ketepatan prediksi, sensitivitas terhadap data, serta

keseimbangan kinerja secara keseluruhan. Hasil perbandingan kinerja seluruh model ditampilkan pada Tabel 4.22.

Tabel 4.22 Perbandingan Kinerja dengan Metode *Baseline*

Model	Akurasi	Presisi	Recall	F1-score
Linear SVM	0.9235	0.9532	0.9183	0.9354
Decision Tree	0.8850	0.9061	0.9031	0.9046
KNN	0.9035	0.9124	0.9294	0.9208
Regresi Logistik	0.9322	0.9763	0.9099	0.9419

Berdasarkan Tabel 4.22, *Linear SVM* menunjukkan *accuracy* sebesar 0.9235, dengan *precision* 0.9532, *recall* 0.9183, dan *F1-score* 0.9354. Nilai *precision* yang tinggi menunjukkan bahwa *Linear SVM* cukup selektif dalam memberikan prediksi positif, sementara nilai *recall* yang relatif tinggi menunjukkan kemampuan yang baik dalam mengenali data aktual. Namun, dibandingkan dengan *Regresi Logistik*, nilai *accuracy* dan *F1-score* *Linear SVM* masih berada di bawah, yang menunjukkan bahwa performanya belum optimal dalam menjaga keseimbangan antar metrik pada dataset ini.

Model *Decision Tree* menghasilkan performa yang paling rendah di antara seluruh metode pembandingan, dengan *accuracy* 0.8850, *precision* 0.9061, *recall* 0.9031, dan *F1-score* 0.9046. Nilai-nilai tersebut menunjukkan bahwa *Decision Tree* mampu mengenali pola data secara umum, tetapi kurang stabil dan kurang akurat dibandingkan model linear. Hal ini berkaitan dengan karakteristik *Decision Tree* yang cenderung sensitif terhadap variasi data dan kurang sesuai untuk representasi *TF-IDF* berdimensi tinggi, sehingga performanya tertinggal dibandingkan algoritma lain.

Sementara itu, *K-Nearest Neighbors (KNN)* menunjukkan performa menengah dengan *accuracy* 0.9035, *precision* 0.9124, *recall* 0.9294, dan *F1-score* 0.9208. Nilai *recall* yang relatif tinggi menunjukkan bahwa *KNN* cukup baik

dalam mengenali data aktual, namun nilai *precision* dan *accuracy* yang lebih rendah dibandingkan *Linear SVM* dan *Regresi Logistik* menunjukkan bahwa model ini masih menghasilkan kesalahan prediksi yang cukup besar. Selain itu, ketergantungan *KNN* pada perhitungan jarak antar data menyebabkan model ini kurang efisien ketika diterapkan pada data teks dengan dimensi fitur yang besar.

Model *Regresi Logistik* dengan penerapan *SMOTE* menunjukkan performa paling seimbang di antara seluruh model yang dibandingkan. Model ini menghasilkan *accuracy* tertinggi sebesar 0.9322, dengan *precision* 0.9763, *recall* 0.9099, dan *F1-score* 0.9419. Kombinasi nilai *precision* yang sangat tinggi dan *F1-score* tertinggi menunjukkan bahwa *Regresi Logistik* mampu menjaga keseimbangan antara ketepatan prediksi dan sensitivitas terhadap data. Dibandingkan dengan *Linear SVM*, *Regresi Logistik* memiliki *accuracy* dan *F1-score* yang lebih tinggi serta menunjukkan stabilitas yang lebih baik dalam mengelola *trade-off* antar metrik evaluasi.

Secara keseluruhan, hasil perbandingan ini menunjukkan bahwa *Regresi Logistik* merupakan model dengan performa paling unggul dan paling konsisten pada dataset ulasan *Mobile JKN*. *Linear SVM* berada pada posisi kedua dengan performa yang cukup kompetitif, sementara *KNN* menunjukkan performa menengah dan *Decision Tree* memiliki performa paling rendah. Dengan mempertimbangkan keseimbangan antar metrik, stabilitas performa, serta kesesuaian terhadap karakteristik fitur TF-IDF berdimensi tinggi, *Regresi Logistik* menjadi metode yang paling tepat dan paling dapat diandalkan untuk analisis sentimen ulasan *Mobile JKN* dalam penelitian ini.

4.2.5 Perbandingan dengan Penelitian Terdahulu

Untuk memperkuat kontribusi empiris penelitian ini, dilakukan perbandingan dengan sejumlah penelitian terdahulu yang mengkaji analisis sentimen terhadap aplikasi Mobile JKN menggunakan pendekatan algoritmik yang berbeda. Perbandingan ini bertujuan untuk menempatkan hasil penelitian secara proporsional dalam konteks perkembangan riset analisis sentimen pada layanan kesehatan digital. Evaluasi tidak hanya difokuskan pada nilai akurasi, tetapi juga mempertimbangkan karakteristik dataset, metode pelabelan, kompleksitas model, serta kebutuhan komputasi. Dengan pendekatan ini, posisi Regresi Logistik dalam penelitian ini dapat dianalisis secara objektif dan sistematis.

Tabel 4.23 Perbandingan dengan Penelitian Terdahulu

Peneliti	Tahun	Dataset	Metode	Hasil
Tamami, dkk.	2025	<i>Mobile JKN</i>	LSTM	Akurasi 0.88, Presisi 0.90
Maulana, dkk.	2025	<i>Mobile JKN</i>	BERT	Akurasi 0.93, F1 0.89
Hematang, dkk.	2025	<i>Mobile JKN</i>	KNN & RL	LR: 0.8785, KNN: 0.8387
Penelitian ini	2025	<i>Mobile JKN</i>	RL	Akurasi 0.9322

Berdasarkan Tabel 4.23, penelitian oleh Hematang dkk. (2025) menunjukkan bahwa Regresi Logistik memperoleh akurasi sebesar 0.8785, lebih tinggi dibandingkan KNN sebesar 0.8387. Penelitian tersebut menggunakan data media sosial X sebanyak 4648 ulasan yang telah melalui *preprocessing* dan pelabelan otomatis berbasis model transformer. Hasil tersebut menegaskan bahwa Regresi Logistik memiliki konsistensi performa yang lebih baik dibandingkan metode berbasis kedekatan jarak pada klasifikasi teks opini. Temuan ini sejalan

dengan karakteristik Regresi Logistik yang mampu memodelkan probabilitas kelas secara lebih stabil pada data berdimensi tinggi.

Apabila dibandingkan dengan penelitian tersebut, model Regresi Logistik dalam penelitian ini menunjukkan peningkatan performa yang lebih tinggi dengan akurasi sebesar 0,9322. Selisih peningkatan ini mengindikasikan bahwa strategi preprocessing yang lebih sistematis serta penerapan SMOTE dalam menangani ketidakseimbangan kelas memberikan dampak signifikan terhadap generalisasi model. Selain itu, optimasi representasi fitur menggunakan TF-IDF berkontribusi pada peningkatan diskriminasi antar kelas sentimen. Dengan demikian, peningkatan performa bukan semata-mata dipengaruhi oleh algoritma, tetapi juga oleh desain eksperimental yang lebih terkontrol.

Jika dibandingkan dengan pendekatan deep learning seperti LSTM dan BERT, posisi penelitian ini tetap kompetitif. Model LSTM pada penelitian Tamami dkk. (2025) memperoleh akurasi 0,88, sedangkan BERT pada penelitian Maulana dkk. (2025) mencapai akurasi 0,93 dengan *F1-score* 0,89. Meskipun model deep learning memiliki kemampuan representasi kontekstual yang lebih kompleks, kebutuhan parameter dan sumber daya komputasi yang tinggi menjadi pertimbangan dalam implementasi praktis. Menariknya, Regresi Logistik dalam penelitian ini mampu mencapai performa yang sebanding bahkan sedikit melampaui capaian BERT pada konteks dataset yang sama.

Secara keseluruhan, penelitian ini menempati posisi yang kuat dalam perbandingan empiris terhadap studi sebelumnya. Hasil penelitian menunjukkan bahwa efektivitas model tidak selalu ditentukan oleh kompleksitas arsitektur, melainkan juga oleh kualitas preprocessing, teknik penyeimbangan data, serta

pemilihan fitur yang tepat. Kontribusi utama penelitian ini terletak pada pembuktian bahwa Regresi Logistik dapat menjadi pendekatan yang efisien, stabil, dan kompetitif dalam analisis sentimen ulasan aplikasi Mobile JKN. Dengan demikian, penelitian ini memberikan implikasi metodologis dan praktis bagi pengembangan sistem evaluasi layanan kesehatan digital berbasis data opini pengguna.

4.3 Kajian Integrasi Islam pada Proses Penyeimbangan Data Menggunakan SMOTE

Hasil penelitian menunjukkan bahwa penggunaan teknik *oversampling* SMOTE pada Regresi Logistik mampu memberikan performa yang lebih seimbang dan stabil dibandingkan ketika model dilatih pada data yang tidak seimbang. Melalui proses penyeimbangan distribusi antara kelas positif dan negatif, model dapat mempelajari pola secara adil tanpa bias terhadap kelas mayoritas. Prinsip “keseimbangan” dan “keadilan” ini memiliki kesesuaian erat dengan nilai-nilai Islam, terutama dalam hal mengelola amanah informasi dan menghindari keputusan yang berat sebelah (*dzulm*) akibat ketimpangan data. Dalam perspektif Islam, mengolah data secara benar, adil, dan proporsional merupakan bagian dari etika intelektual (*adab al-‘ilm*), karena keputusan berbasis informasi yang bias dapat menimbulkan ketidakadilan dalam penerapan teknologi.

Konsep penyeimbangan data (SMOTE) dalam *machine learning* dapat dipandang sebagai bentuk upaya menjaga “*mīzān*” (keseimbangan), yaitu kondisi ideal agar suatu sistem tidak condong pada salah satu sisi secara berlebihan. Al-Quran menjelaskan bahwa Allah SWT menetapkan keseimbangan sebagai prinsip dasar dalam seluruh ciptaan-Nya, termasuk dalam pengambilan keputusan manusia.

Ketika suatu dataset memiliki dominasi kelas tertentu, model akan terdorong untuk lebih “percaya” pada mayoritas dan mengabaikan minoritas. SMOTE hadir sebagai mekanisme untuk mengembalikan proporsi agar keputusan model tidak zalim, sama seperti perintah syariat untuk menjaga keadilan dalam menilai suatu perkara, meskipun kelompok yang dinilai tidak dominan. Hal ini tersirat dalam QS. Ar-Rahmān ayat 7–9 berikut:

وَالسَّمَاءَ رَفَعَهَا وَوَضَعَ الْمِيزَانَ ﴿٧﴾ أَلَّا تَطْغَوْا فِي الْمِيزَانِ ﴿٨﴾ وَأَقِيمُوا الْوَزْنَ بِالْقِسْطِ وَلَا تُخْسِرُوا الْمِيزَانَ ﴿٩﴾

“Langit telah Dia tinggikan dan Dia telah menciptakan timbangan (keadilan dan keseimbangan) {7} agar kamu tidak melampaui batas dalam timbangan itu {8} Tegakkanlah timbangan itu dengan adil dan janganlah kamu mengurangi timbangan itu.” (NU Online, 2025)

Tafsir Tahlili menjelaskan bahwa Allah SWT meninggikan langit dan menetapkan *al-mīzān* sebagai hukum dasar bagi seluruh ciptaan-Nya. Keseimbangan tersebut tidak hanya berkaitan dengan keteraturan fisik alam semesta, tetapi juga mencakup prinsip moral dan spiritual yang wajib ditegakkan manusia—mulai dari keseimbangan akidah, ibadah, perilaku, hingga penggunaan akal dan teknologi. Manusia diperintahkan agar tidak merusak keseimbangan itu, karena ketimpangan akan menimbulkan kerusakan dalam kehidupan sosial maupun dalam proses pengambilan keputusan. Pada ayat berikutnya, Allah SWT memerintahkan manusia menegakkan timbangan dengan adil dan tidak berlaku curang, yang menunjukkan bahwa seluruh aktivitas manusia—baik dalam menimbang barang, menilai data, membuat keputusan, maupun mengelola informasi—harus dilakukan dengan ketelitian dan unsur keadilan.

Selain itu, Allah SWT memberikan contoh melalui keteraturan peredaran matahari dan bulan yang dijadikan dasar penanggalan dan perhitungan waktu, sebagai bukti bahwa alam semesta beroperasi berdasarkan perhitungan yang presisi.

Hal ini mengisyaratkan bahwa manusia didorong untuk memanfaatkan ilmu pengetahuan dan teknologi dengan benar, serta menjaga keseimbangan dalam semua aspek kehidupan. Dengan demikian, QS. Ar-Rahmān ayat 7–9 menjadi landasan etis bahwa dalam proses pengolahan data modern seperti *machine learning*, prinsip keadilan, keseimbangan, dan ketelitian wajib ditegakkan agar tidak terjadi bias atau ketimpangan yang merugikan pihak tertentu. Nilai-nilai ini sejalan dengan tujuan teknik SMOTE dalam penelitian ini, yaitu memastikan bahwa model memberikan penilaian sentimen secara adil tanpa keberpihakan terhadap kelas yang dominan.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan rumusan masalah dan pembahasan pada bab sebelumnya maka diperoleh kesimpulan sebagai berikut:

1. Model Regresi Logistik mampu memberikan performa yang baik dalam mengklasifikasikan polaritas ulasan pengguna *Mobile* JKN. Pada data uji, model Regresi Logistik yang dilatih dengan skema penyeimbangan data menghasilkan akurasi sebesar 0.9322, dengan presisi kelas positif 0.9763, *recall* kelas positif 0.9099, dan *F1-score* kelas positif 0.9419. Selain itu, hasil pengujian variasi parameter kecuraman sigmoid (α) pada nilai 0.5, 1.0, 2.0, dan 3.0 menunjukkan bahwa perubahan α memengaruhi dinamika konvergensi dan stabilitas *learning curve*, namun tidak memberikan peningkatan signifikan terhadap performa akhir model. Nilai $\alpha = 2.0$ menunjukkan keseimbangan terbaik antara penurunan training *loss* dan stabilitas *validation loss*. Kurva pada konfigurasi ini relatif konvergen dengan gap yang terkendali, sehingga mencerminkan kemampuan generalisasi yang baik. Meskipun $\alpha = 3.0$ menghasilkan *training loss* yang lebih rendah, sensitivitas terhadap data validasi sedikit meningkat. Oleh karena itu, $\alpha = 2.0$ dapat dianggap sebagai konfigurasi yang paling proporsional dalam eksperimen ini.
2. Penerapan SMOTE mengubah distribusi performa model dan menghasilkan keseimbangan kinerja antar kelas melalui mekanisme *trade-off*. Setelah SMOTE diterapkan, *recall* kelas negatif meningkat dari 0.9443 menjadi

0.9663, dan *F1-score* kelas negatif meningkat dari 0.9165 menjadi 0.9188, yang menunjukkan bahwa model menjadi lebih baik dalam mengenali ulasan negatif. Di sisi lain, *recall* kelas positif mengalami penurunan dari 0.9235 menjadi 0.9099, dan *F1-score* kelas positif sedikit menurun dari 0.9423 menjadi 0.9419, yang mencerminkan adanya *trade-off* akibat penyeimbangan data. Namun demikian, presisi kelas positif meningkat dari 0.9620 menjadi 0.9763, dan *F1-score macro average* meningkat dari 0.9294 menjadi 0.9304, sehingga performa model menjadi lebih merata antar kelas. Didukung oleh pola *learning curve* yang lebih stabil, penerapan SMOTE membantu model mengurangi bias terhadap kelas mayoritas dan menghasilkan pembelajaran yang lebih proporsional.

5.2 Saran

Adapun beberapa hal yang dapat dijadikan acuan untuk penyempurnaan penelitian selanjutnya, maka penulis memberikan saran berikut:

1. Penelitian selanjutnya disarankan menggunakan metode pelabelan berbasis *Aspect-Based Sentiment Analysis* (ABSA) agar pelabelan sentimen lebih akurat dan tidak bergantung pada polaritas rating yang sering tidak mencerminkan isi ulasan.
2. Untuk meningkatkan *recall*, penelitian mendatang dapat mempertimbangkan penyesuaian *decision threshold*, penggunaan fitur yang lebih kontekstual seperti *word embedding*, atau pemilihan model yang lebih peka terhadap variasi bahasa alami.
3. Jumlah dan kualitas dataset perlu ditingkatkan dengan menambah sumber ulasan, memperluas periode pengambilan data, serta melakukan pembersihan

teks secara lebih menyeluruh agar model memperoleh pola yang lebih beragam dan representatif.

4. Penelitian berikutnya dapat mengeksplorasi varian Regresi Logistik dengan penalti seperti L1, L2, atau *Elastic Net* untuk meningkatkan stabilitas, mencegah *overfitting*, dan melakukan seleksi fitur pada data berdimensi tinggi seperti TF-IDF.

DAFTAR PUSTAKA

- Al-Ṭabarānī, A. al-Q. S. bin A. (2012). *al-Mu‘jam al-Awsaṭ (المعجم الأوسط)*. Dār al-Kutub al-‘Ilmiyyah (DKI).
- Aliman, G., Nivera, T. F. S., Olazo, J. C. A., Ramos, D. J. P., Sanchez, C. D. B., Amado, T. M., Arago, N. M., Jr, R. L. J., Virrey, G. C., & Valenzuela, I. C. (2022). Sentiment Analysis using Logistic Regression. *International Journal of Engineering Research and Applications*, July, 35–40. <https://doi.org/10.9790/9622-1107023640>
- Bhakti, I. N., Sholikhin, A. Z., Lukman, M. A., Daniati, E., & Ristyawan, A. (2024). Klasifikasi Kategori Berita Menggunakan Naive Bayes. *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 8(2), 1155–1164.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer Science+Business Media.
- Bourequat, W., & Mourad, H. (2021). Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine. *International Journal of Advances in Data and Information Systems*, 2(1), 36–44. <https://doi.org/10.25008/ijadis.v2i1.1216>
- Burrhanuddin, I., & Fauzan Rozi, A. (2025). Analisis Dan Klasifikasi Sentimen Ulasan Pengguna Aplikasi Redbus Menggunakan Metode Logistic Regression Dan Svm. *Jurnal Inovtek Polbeng - Seri Informatika*, 10(2), 877–887.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(February), 321–357. <https://doi.org/10.1613/jair.953>
- Crocamo, C., Viviani, M., Famigliani, L., Bartoli, F., Pasi, G., & Carrà, G. (2021).

- Surveilling COVID-19 Emotional Contagion on Twitter by Sentiment Analysis. *European Psychiatry*, 64(1), 1–6.
<https://doi.org/10.1192/j.eurpsy.2021.3>
- Djamaris, A. R. A. (2021). *Pemanfaatan regresi logistik ordinal dan multinomial dengan SPSS*. Universitas Bakrie.
- Fauzan, F. J., M Afdal, Rice Novita, & Mustakim. (2024). Penerapan Machine Learning Pada Analisis Sentimen Aplikasi Mytelkomsel Menggunakan Data Ulasan Google Playstore. *Indonesian Journal of Computer Science*, 13(3), 4747–4761. <https://doi.org/10.33022/ijcs.v13i3.4024>
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On Calibration of Modern Neural Networks. *International Congress on Medical Librarianship*.
<https://doi.org/https://doi.org/10.48550/arXiv.1706.04599>
- Gupta, M. (2022). A Comparative Study on Supervised Machine Learning Algorithm. *International Journal for Research in Applied Science and Engineering Technology*, 10(1), 1023–1028.
<https://doi.org/10.22214/ijraset.2022.39980>
- Hanum, A. R., Zetha, I. A., Putri, S. C., Wulandari, R. A., Andina, S. P., Fajrina, J. N., & Yudistira, N. (2024). Analisis Kinerja Algoritma Klasifikasi Teks Bert dalam Mendeteksi Berita Hoaks. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 11(3), 537–546. <https://doi.org/10.25126/jtiik.938093>
- Hematang, K. V., Utami, N. W., & Purnama, I. N. (2025). Analisis Sentimen Opini Masyarakat Terkait Penggunaan Mobile Jkn Pada Media Sosial X Menggunakan Algoritma K-Nearestneighbor (Knn) Dan Logistic Regression (Lr). *Jurnal Sistem Informasi (JUSIN)*, 6, 131–146.

- Henry, O., Avula, O., & Adekola, S. (2023). Study of Binary logistic and Poisson regression models of Diabetic Patience in Study of Binary Logistic and Poisson Regression Models of Diabetic Patients in Nigeria using Dichotomous and Non- Dichotomous Predictors. *Asian Journal of Probability and Statistics*, February. <https://doi.org/10.9734/AJPAS/2022v17i330425>
- Hu, N., Koh, N. S., & Reddy, S. K. (2014). Ratings lead you to the product, reviews help you clinch it? the mediating role of online review sentiments on product sales. *Decision Support Systems*, 1(57), 42–53. <https://doi.org/https://doi.org/10.1016/j.dss.2013.07.009>
- Husen, R. A., Astuti, R., Marlia, L., Rahmaddeni, R., & Efrizoni, L. (2023). Analisis Sentimen Opini Publik pada Twitter Terhadap Bank BSI Menggunakan Algoritma Machine Learning. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 3(2), 211–218. <https://doi.org/10.57152/malcom.v3i2.901>
- Husin, N. (2023). Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN). *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, 7(1), 75–84. <https://doi.org/10.55886/infokom.v7i1.608>
- Kesehatan, B. (2025). *BPJS Kesehatan*. <https://www.bpjs-kesehatan.go.id/#/> diakses pada 11 Agustus 2025 pukul 20.00 WIB
- Lee, W., & Seo, K. (2022). Downsampling for Binary Classification with a Highly Imbalanced Dataset Using Active Learning. *Big Data Research*, 28, 100314. <https://doi.org/10.1016/j.bdr.2022.100314>
- Lumi, W. M. E., Musak, R. A., Tumiwa, F. F., Waworuntu, M. Y., & Surya, W. S.

- (2023). Edukasi Tentang Penggunaan Aplikasi Mobile JKN Pada Pasien Rawat Jalan RSUD Anugerah Tomohon. *Jurnal Pengabdian Kepada Masyarakat Nusantara (JPkMN)*, 4(3), 1620–1626.
- Maulana, A. A. R. S., Saraswati, N. M., & Millah, A. S. (2025). Analisis Sentimen Review Aplikasi JKN Mobile menggunakan Algoritma Bidirectional Encoder Representations from Transformers. *Jurnal Sistem Informasi Dan Teknologi Peradaban*, 6(1), 42–48.
- McCullagh, P., & Nelder, J. A. (1989). *Generalized Linear Models* (2nd ed.). Routledge.
- Naeem, M. Z., Rustam, F., & Mehmood, A. (2022). Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms. *PeerJ Comput Sci*, 1–28. <https://doi.org/10.7717/peerj-cs.914>
- Nurian, A., Padilah, T. N., & Garno, G. (2024). Analisis Sentimen Terhadap Pelayanan Disdukcapil Karawang Menggunakan Naive Bayes Classifier. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(2). <https://doi.org/10.23960/jitet.v12i2.4178>
- Online, N. (2025). *Al-Quran Online Terjemah dan Tafsir Bahasa*. <https://quran.nu.or.id/>
- Powers, D. M. W. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technology*, 2(1). <https://doi.org/10.9735/2229-3981>
- Pramudita, R., Safitri, N., & Nazah, V. Z. (2025). *Jurnal Studi Komparatif Algoritma Machine Learning dengan Teknik Bagging dan AdaBoost pada*

Klasifikasi Kanker Payudara. 12(1), 102–108.

- Rahmawati, I., Rika Fitriani, T., No'eman, A., & Yusuf, A. Y. P. (2023). Analisis Sentimen Menggunakan Algoritma Logistic Regression Pada Penerbangan Lion Air berdasarkan Ulasan Platform Online. *Jurnal Riset Informatika Dan Teknologi Informasi*, 1(1), 11–16. <https://doi.org/10.58776/jriti.v1i1.60>
- Roihan, A., Sunarya, P. A., & Rafika, A. S. (2020). Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82. <https://doi.org/10.31294/ijcit.v5i1.7951>
- Sajiwo, A. F. B., Rahmat, B., & Junaidi, A. (2024). Klasifikasi Indeks Standar Pencemaran Udara (Ispu) Menggunakan Algoritma Xgboost Dengan Teknik Imbalanced Data (Smote). *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(3). <https://doi.org/10.23960/jitet.v12i3.4699>
- Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
- Sausan, Pratiwi, D. M., & Mufidah, L. (2024). Perbandingan Metode Decision Tree Classifier dan XGBoost Classifier Dalam Memprediksi Penyakit Jantung. *Conference on Electrical Engineering, Informatics Technology and Creative Media 2024*, 4, 991–1000.
- Sugihartono, T., & Putra, R. R. C. (2024). Penerapan Metode Support Vector Machine Dalam Classifikasi Ulasan Pengguna Aplikasi Mobile Jkn. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, 7(2), 144–153. <https://doi.org/10.36080/skanika.v7i2.3193>

- Suhadi, Jumakil, Kamrin, & Irma. (2022). Dampak Penggunaan Aplikasi Mobile Jkn Terhadap Pelayanan Bpjs. *Jurnal Kesehatan*, 15(1), 11–23.
<https://doi.org/10.23917/jk.v15i1.15977>
- Tamami, G., Triyanto, W. A., & Muzid, S. (2025). Sentiment Analysis Mobile JKN Reviews Using SMOTE Based LSTM. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 19(1), 13.
<https://doi.org/10.22146/ijccs.101910>
- Tao, J., & Fang, X. (2020). Toward multi-label sentiment analysis: a transfer learning based approach. *Journal of Big Data*, 7(1), 1–26.
<https://doi.org/10.1186/s40537-019-0278-0>
- Ulya, D. M., Yuliana, R. E., & Jamhuri, M. (2025). Reliable and Efficient Sentiment Analysis on IMDb with Logistic Regression. *Cauchy: Jurnal Matematika Murni Dan Aplikasi*, 10(2), 821–834.
- Widagdo, Saputro, A. S., & Nugroho, F. E. (2024). *Buku Monograf Analisis Sentimen Mobil Listrik Berbasis Ulasan Youtube*. PT. Sonpedia Publishing Indonesia.
- Widiantoro, A. D., Mustafid, & Sanjaya, R. (2024). *Pengantar NLP dan Topik Model LDA*. Asosiasi Doktor Sistem Informasi Indonesia.
- Wijoyo A, Saputra A, Ristanti S, Sya'ban S, Amalia M, & Febriansyah R. (2024). Pembelajaran Machine Learning. *OKTAL (Jurnal Ilmu Komputer Dan Science)*, 3(2), 375–380.
<https://journal.mediapublikasi.id/index.php/oktal/article/view/2305>
- Wilson, A., & Anwar, M. R. (2024). The Future of Adaptive Machine Learning Algorithms in High-Dimensional Data Processing. *International Transactions*

on *Artificial Intelligence (ITALIC)*, 3(1), 97–107.

<https://doi.org/10.33050/italic.v3i1.656>

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2017). FOURTH EDITION
DATA MINING Practical Machine Learning Tools and Techniques. In *Data
mining - Practical machine learning tools and techniques*.

Zaki, M. J., & Meira, W. J. (2020). Data Mining and Machine Learning
Fundamental Concepts and Algorithms. In *Journal of Chemical Information
and Modeling* (Vol. 53, Issue 9).

LAMPIRAN

Lampiran 1 Data Ulasan Aplikasi *Mobile* JKN



Contoh Data:

No	Score	At	Content
0	4	2025-11-18 13:15:18	bagus sudah ada aplikasinya
1	1	2025-11-18 13:12:54	aplikasi lambat, ribet
2	2	2025-11-18 13:04:19	Aplikasi busuk, setiap kali mau buka saat butuh...
3	1	2025-11-18 12:51:04	apk sering bgt ke logout ,pr bgt klo musti log...
4	5	2025-11-18 12:37:45	sangat membantu
...
9995	1	2025-09-15 00:19:54	ribett
9996	5	2025-09-15 00:12:19	ok
9997	4	2025-09-15 00:07:22	tolong terus update
9998	1	2025-09-15 00:06:05	kode otp sms gk masuk masuk ini gimana
9999	1	2025-09-15 00:05:28	Sistem ngebug lelet & tidak pesan dipakai

Lampiran 2 *Source Code Scraping* Data

https://colab.research.google.com/drive/1mHkT_Ro3PflwYV7Y9SBzAf4ZeQcq_dxz?usp=sharing

```
!pip install google-play-scraper
from google_play_scraper import reviews, Sort
import pandas as pd
import time
max_retries = 3
retry_delay = 5
result = None
for i in range(max_retries):
    try:
        result, _ = reviews(
            'app.bpjs.mobile',
            lang='id',
```

```

country='id',
sort=Sort.NEWEST,
count=10000,
filter_score_with=None
)
if result:
    break
else:
    time.sleep(retry_delay)
except Exception as e:
    if i < max_retries - 1:
        time.sleep(retry_delay)
    else:
        raise
df = pd.DataFrame(result)
df['at'] = pd.to_datetime(df['at'])
filtered_df = df[(df['at'].dt.year >= 2025) & (df['at'].dt.year <= 2025)]

filtered_df.to_csv("dataset_skripsi.csv", index=False, encoding='utf-8-sig')
filtered_df[['score','at','content']].to_csv("dataset.csv", index=False,
encoding='utf-8-sig')

```

Lampiran 3 Hasil *Text Preprocessing*

<https://bit.ly/hasiltextpreprocessing>

Lampiran 4 *Source Code* Evaluasi Kinerja Regresi Logistik dengan *Oversampling* SMOTE

<https://colab.research.google.com/drive/1F2AVr02u2qDL0BYF1SJEhd5RyEUyA8z?usp=sharing>

```

import time
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score, precision_score, recall_score,
f1_score, confusion_matrix, classification_report
from imblearn.over_sampling import SMOTE
import tensorflow as tf
from tensorflow.keras.callbacks import EarlyStopping
file_id = "1afypG-AFpflgqqwQZn9mo999teIjLD8T"
url = f'https://drive.google.com/uc?export=download&id={file_id}'
df = pd.read_csv(url)
def categorize_sentiment(score):

```

```

    if score in [1, 2]:
        return "negatif"
    elif score in [4, 5]:
        return "positif"
df["sentiment"] = df["score"].apply(categorize_sentiment)
df_bin = df[df["sentiment"].notna()].copy()
label_encoder = LabelEncoder()
df_bin["sentiment_encoded"] =
label_encoder.fit_transform(df_bin["sentiment"])
df_pos = df_bin[df_bin["sentiment"] == "positif"]
df_neg = df_bin[df_bin["sentiment"] == "negatif"]
X_train_pos, X_test_pos, y_train_pos, y_test_pos = train_test_split(
    df_pos["content"], df_pos["sentiment_encoded"], test_size=0.2,
    random_state=42
)
X_train_neg, X_test_neg, y_train_neg, y_test_neg = train_test_split(
    df_neg["content"], df_neg["sentiment_encoded"], test_size=0.2,
    random_state=42
)
X_train = pd.concat([X_train_pos, X_train_neg])
y_train = pd.concat([y_train_pos, y_train_neg])
X_test = pd.concat([X_test_pos, X_test_neg])
y_test = pd.concat([y_test_pos, y_test_neg])
train_df = pd.concat([X_train, y_train], axis=1).sample(frac=1,
    random_state=42)
test_df = pd.concat([X_test, y_test], axis=1).sample(frac=1, random_state=42)
X_train = train_df["content"]
y_train = train_df["sentiment_encoded"]
X_test = test_df["content"]
y_test = test_df["sentiment_encoded"]
vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
X_train_dense = X_train_tfidf.toarray()
X_tr_ns, X_val_ns, y_tr_ns, y_val_ns = train_test_split(
    X_train_dense, y_train, test_size=0.1, random_state=42, stratify=y_train
)
)
model_no_smote = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(X_train_tfidf.shape[1],)),
    tf.keras.layers.Dense(1, activation="sigmoid")
])
)
model_no_smote.compile(optimizer="adam", loss="binary_crossentropy",
    metrics=["accuracy"])
early_stop_ns = EarlyStopping(monitor="val_loss", patience=10,
    restore_best_weights=True)
history_no_smote = model_no_smote.fit(
    X_tr_ns, y_tr_ns,
    validation_data=(X_val_ns, y_val_ns),

```

```

    epochs=200,
    batch_size=512,
    callbacks=[early_stop_ns],
    verbose=0
)
y_pred_ns = (model_no_smote.predict(X_test_tfidf.toarray()) >=
0.5).astype(int)
sm = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = sm.fit_resample(X_train_dense,
y_train)
X_tr, X_val, y_tr, y_val = train_test_split(
    X_train_resampled, y_train_resampled,
    test_size=0.1, random_state=42, stratify=y_train_resampled
)
final_model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(X_train_resampled.shape[1],)),
    tf.keras.layers.Dense(1, activation="sigmoid")
])
final_model.compile(optimizer="adam", loss="binary_crossentropy",
metrics=["accuracy"])
early_stop = EarlyStopping(monitor="val_loss", patience=10,
restore_best_weights=True)
history_final = final_model.fit(
    X_tr, y_tr,
    validation_data=(X_val, y_val),
    epochs=200,
    batch_size=512,
    callbacks=[early_stop],
    verbose=0
)
y_pred = (final_model.predict(X_test_tfidf.toarray()) >= 0.5).astype(int)
def evaluate(y_true, y_pred):
    acc = accuracy_score(y_true, y_pred)
    prec = precision_score(y_true, y_pred, zero_division=0)
    rec = recall_score(y_true, y_pred, zero_division=0)
    fl = f1_score(y_true, y_pred, zero_division=0)
    return acc, prec, rec, fl
acc_no_smote, prec_no_smote, rec_no_smote, fl_no_smote = evaluate(y_test,
y_pred_ns)
acc_smote, prec_smote, rec_smote, fl_smote = evaluate(y_test, y_pred)

```

RIWAYAT HIDUP



Diah Mariatul Ulya, lahir di Malang pada 23 Agustus 2003. Penulis berasal dari Desa Sukoanyar, Kecamatan Wajak, Kabupaten Malang, dan merupakan putri kedua dari empat bersaudara dari pasangan Bapak Imam Muhsin dan Ibu Umi Sa'diyah. Lingkungan keluarga dan tempat tinggal penulis memberikan nilai kedisiplinan, kemandirian, serta semangat

belajar yang kuat sejak usia dini. Pendidikan formal penulis dimulai di RA Manarul Huda dan diselesaikan pada tahun 2010. Selanjutnya, penulis melanjutkan pendidikan dasar di MI Manarul Huda dan lulus pada tahun 2016. Pendidikan tingkat menengah pertama ditempuh di MTs Alhidayah Wajak dan diselesaikan pada tahun 2019. Setelah itu, penulis melanjutkan pendidikan menengah atas di MA Alhidayah Wajak dan berhasil menyelesaikannya pada tahun 2022. Pada tahun yang sama, penulis melanjutkan pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang dengan memilih Program Studi Matematika pada Fakultas Sains dan Teknologi. Selama menempuh pendidikan di perguruan tinggi, penulis aktif berpartisipasi dalam kegiatan organisasi kemahasiswaan, khususnya sebagai anggota HMPS Integral Matematika Divisi Penerbitan dan Jurnalistik selama satu periode sekitar tahun 2023. Keterlibatan tersebut memberikan pengalaman dalam bidang kepenulisan, pengelolaan media, serta kerja sama tim. Selain aktif berorganisasi, penulis juga terlibat dalam berbagai kegiatan kepanitiaan di lingkungan kampus yang mendukung pengembangan soft skills, seperti kepemimpinan, komunikasi, dan manajemen waktu. Pada semester enam, penulis dipercaya menjadi asisten laboratorium/praktikum, yang berperan dalam membantu proses pembelajaran praktikum serta pendampingan akademik mahasiswa. Berbagai pengalaman tersebut menjadi bekal penting bagi penulis dalam mengembangkan kompetensi akademik, profesional, dan kepribadian selama masa studi.



BUKTI KONSULTASI SKRIPSI

Nama : Diah Mariatul Ulya
NIM : 220601110002
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Evaluasi Kinerja Algoritma Regresi Logistik Menggunakan *Oversampling* SMOTE pada Analisis Sentimen Ulasan Pengguna Aplikasi *Mobile JKN*
Pembimbing I : Dr. Mohammad Jamhuri, M.Si.
Pembimbing II : Erna Herawati, M.Pd.

No	Tanggal	Hal	Tanda Tangan
1.	11 Agustus 2025	Konsultasi Bab I, II, dan III	1.
2.	10 September 2025	Konsultasi Bab I, II, dan III	2.
3.	17 September 2025	Konsultasi Bab I, II, dan III	3.
4.	24 September 2025	Konsultasi Bab I, II, dan III	4.
5.	25 September 2025	Konsultasi Kajian Agama Bab I dan II	5.
6.	30 September 2025	Konsultasi Bab I, II, dan III	6.
7.	2 Oktober 2025	Konsultasi Bab I, II, dan III	7.
8.	7 Oktober 2025	Konsultasi Bab I, II, dan III	8.
9.	17 Oktober 2025	ACC Bab I, II, dan III	9.
10.	21 Oktober 2025	ACC Kajian Agama Bab I dan II	10.
11.	3 November 2025	ACC Seminar Proposal	11.
12.	6 November 2025	Konsultasi Revisi Seminar Proposal	12.
13.	24 November 2025	Konsultasi Bab IV dan V	13.
14.	25 November 2025	Konsultasi Kajian Agama Bab IV	14.
15.	27 November 2025	ACC Bab IV dan V	15.
16.	27 November 2025	ACC Kajian Agama Bab IV	16.



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

17.	3 Desember 2025	ACC Seminar Hasil	17.
18.	11 Desember 2025	Konsultasi Revisi Seminar Hasil	18.
19.	2 Januari 2026	Konsultasi Revisi Seminar Hasil	19.
20.	2 Januari 2026	ACC Sidang Skripsi	20.
21.	19 Februari 2026	ACC Keseluruhan	21.

Malang, 19 Februari 2026

Mengetahui,
Ketua Program Studi Matematika



Dr. Fachrur Rozi, M.Si.

NIP. 19800527 200801 1 012