SISTEM KLASIFIKASI SENTIMEN MULTIKELAS DENGAN PENDEKATAN ASPEK DOMINAN PADA ULASAN APLIKASI PLN MOBILE MENGGUNAKAN *WEIGHTED SOFT VOTING ENSEMBLE*

Proposal Tugas Akhir

Diajukan Oleh

Dhiyaul Auliya
220705050

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR–RANIRY
BANDA ACEH
TAHUN 2026 M/1448 H

LEMBAR PERSETUJUAN

SISTEM KLASIFIKASI SENTIMEN MULTIKELAS DENGAN PENDEKATAN ASPEK DOMINAN PADA ULASAN APLIKASI PLN MOBILE MENGGUNAKAN *WEIGHTED SOFT VOTING ENSEMBLE*

TUGAS AKHIR

Diajukan kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai salah satu Beban Studi Memperoleh Gelar Sarjana (S1)
Dalam Ilmu Teknologi Informasi

Oleh :

Dhiyaul Auliya
NIM. 220705050

Mahasiswa Fakultas Sains dan Teknologi
Progam Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan oleh :

Pembimbing I,	Pembimbing II,
Ghufran Ibnu Yasa, M.T.	**Mulkan Fadhli, S.T., M.T.**
NIP. 198409262014031005	**NIP. 198811282020121006**

Mengetahui,
Ketua Program Studi Teknologi informasi,

Malahayati, M.T.
NIP. 198301272015032003

LEMBAR PENGESAHAN

SISTEM KLASIFIKASI SENTIMEN MULTIKELAS DENGAN PENDEKATAN ASPEK DOMINAN PADA ULASAN APLIKASI PLN MOBILE MENGGUNAKAN *WEIGHTED SOFT VOTING ENSEMBLE*

TUGAS AKHIR

Telah Diuji oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu, 12 Agustus 2026 M
28 Safar 1448 H
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasyah Tugas Akhir:

Ketua,	Sekretaris,
Gufran Ibnu Yasa, M.T.	**Mulkan Fadhli, S.T., M.T**
NIP. 198409262014031005	**NIP. 198811282020121006**
Penguji I,	Penguji II,
Nizam Albar, S.T, M.T.	**Dr. Hendri Ahmadian, M.I.M.**
NUPTK. 2849779680130032	**NIP. 198301042014031002**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU
NIP. 196210021988111001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Dhiyaul Auliya
NIM	:	220705050
Program Studi	:	Teknologi Informasi
Fakultas	:	Sains dan Teknologi
Judul	:	Sistem Klasifikasi Sentimen Multikelas Dengan Pendekatan Aspek Dominan Pada Ulasan Aplikasi Pln Mobile Menggunakan Weighted Soft Voting Ensemble

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah karya orang lain;
3. Tidak menggunakan karya orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik karya;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri karya ini dan mampu bertanggungjawab atas karya ini.

Bila dikemudian hari ada tuntutan dari pihak lain atas karya saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang ditemukan bukti bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh. Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh, 15 Agustus 2026

Yang Menyatakan,

Dhiyaul Auliya

ABSTRAK

Aplikasi PLN Mobile menghasilkan volume ulasan pengguna yang sangat masif di Google Play Store, yang menjadi aset data penting untuk mengevaluasi kepuasan pelanggan namun sulit jika dianalisis secara manual. Analisis sentimen konvensional seringkali hanya memberikan label polaritas secara umum tanpa menunjuk pada fitur layanan yang spesifik secara tepat. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi sentimen multikelas (positif, negatif, dan netral) pada ulasan aplikasi PLN Mobile dengan menggunakan pendekatan aspek dominan melalui metode *Weighted Soft Voting Ensemble*. Pengumpulan data dilakukan dengan menarik 23.850 ulasan yang kemudian melalui tahapan pra-pemrosesan teks komprehensif, meliputi *case folding*, pembersihan data, normalisasi kata tidak baku, penanganan negasi, penghapusan *stopword*, dan *stemming* menggunakan *library* Sastrawi. Ekstraksi fitur dilakukan menggunakan pembobotan TF-IDF yang dikombinasikan dengan N-gram, sementara ketimpangan distribusi kelas data (78,4% ulasan positif, 19,6% negatif, dan 1,9% netral) diatasi menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE) pada data latih. Model *Weighted Soft Voting Ensemble* diintegrasikan dari empat algoritma dasar berbasis linear, yaitu *Multinomial Naive Bayes* (MNB), *Logistic Regression* (LR), *SGD Classifier*, dan *Linear SVC*. Hasil pengujian menunjukkan bahwa metode *ensemble* ini beroperasi dengan sangat stabil dan tangguh, mencapai *Accuracy* sebesar 91,47%, *Precision* 91,49%, *Recall* 91,47%, dan *F1-Score* 91,42%. Hasil analisis aspek dominan menemukan bahwa kategori "Pembayaran & Token" adalah topik yang paling krusial dan mendominasi pembahasan (52,49%), sedangkan kategori "Antarmuka Pengguna (UI/UX)" merupakan aspek yang paling banyak mendapatkan proporsi sentimen negatif (38,5%). Sebagai luaran praktis, model klasifikasi ini telah berhasil diimplementasikan ke dalam sebuah purwarupa (*prototype*) *dashboard* analitik berbasis kerangka kerja Streamlit yang mampu memvisualisasikan data serta menarik ulasan baru secara *real-time*.

Kata Kunci: Analisis Sentimen, PLN Mobile, Klasifikasi Multikelas, Aspek Dominan, TF-IDF, SMOTE, *Weighted Soft Voting Ensemble*, Streamlit.

ABSTRACT

The PLN Mobile application generates a massive volume of user reviews on the Google Play Store, which serve as crucial data assets for evaluating customer satisfaction but are difficult to analyze manually. Conventional sentiment analysis often only provides general polarity labels without accurately pointing to specific service features. Therefore, this study aims to develop a multiclass sentiment classification system (positive, negative, and neutral) for PLN Mobile application reviews using a dominant aspect approach through the Weighted Soft Voting Ensemble method. Data collection was carried out by extracting 23,850 reviews which then went through comprehensive text preprocessing stages, including case folding, data cleaning, non-standard word normalization, negation handling, stopword removal, and stemming using the Sastrawi library. Feature extraction was performed using TF-IDF weighting combined with N-grams, while the imbalanced data class distribution (78.4% positive, 19.6% negative, and 1.9% neutral reviews) was addressed using the Synthetic Minority Over-sampling Technique (SMOTE) on the training data. The Weighted Soft Voting Ensemble model was integrated from four linear-based base algorithms, namely Multinomial Naive Bayes (MNB), Logistic Regression (LR), SGD Classifier, and Linear SVC. The test results showed that this ensemble method operates very stably and robustly, achieving an Accuracy of 91.47%, Precision of 91.49%, Recall of 91.47%, and F1-Score of 91.42%. The dominant aspect analysis results found that the "Payment & Token" category is the most crucial topic and dominates the discussion (52.49%), while the "User Interface (UI/UX)" category is the aspect that receives the highest proportion of negative sentiment (38.5%). As a practical output, this classification model has been successfully implemented into an analytical dashboard prototype based on the Streamlit framework that is capable of visualizing data and scraping new reviews in real-time.

Keywords: Sentiment Analysis, PLN Mobile, Multiclass Classification, Dominant Aspect, TF-IDF, SMOTE, Weighted Soft Voting Ensemble, Streamlit

KATA PENGANTAR

Bismillahirrahmanirrahim.

Segala puji bagi Allah, Tuhan Yang Maha Esa dan Penguasa semesta alam. Semoga shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW, beserta keluarga dan para sahabatnya. Tugas akhir penulis yang berjudul "Sistem Klasifikasi Sentimen Multikelas Dengan Pendekatan Aspek Dominan Pada Ulasan Aplikasi PLN Mobile Menggunakan Weighted Soft Voting Ensemble" dapat diselesaikan berkat karunia dan rahmat Allah, Yang Maha Pengasih lagi Maha Penyayang. Penyusunan karya ilmiah ini merupakan salah satu kriteria untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh.

Penulis mengucapkan terima kasih yang tulus kepada semua pihak yang telah membantu dalam proses belajar, transfer pengetahuan, dukungan moral, serta bantuan lainnya yang sangat penting dalam penyelesaian tugas akhir ini, baik secara langsung maupun tidak langsung. Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua penulis, yang secara konsisten memberikan dukungan, kasih sayang, dan doa yang tiada henti. Cinta orang tua hanya dapat dibalas oleh Allah; semoga Allah senantiasa mencurahkan rahmat-Nya kepada mereka sehingga mereka dapat melihat kesuksesan masa depan penulis.
2. Bapak Ghufran Ibnu Yasa, M.T., yang telah berkenan meluangkan waktu menjadi pembimbing akademik sekaligus pembimbing tugas akhir penulis, serta senantiasa memberikan arahan, masukan, dan dukungan sejak awal perkuliahan hingga penyusunan tugas akhir ini.
3. Bapak Mulkan Fadhli, S.T, M.T., yang secara terus-menerus memberikan arahan, evaluasi, dan nasihat yang sangat berharga dalam menyempurnakan penelitian ini.
4. Segenap dosen Program Studi Teknologi Informasi Fakultas Sains dan Teknologi yang telah membekali penulis dengan berbagai ilmu pengetahuan selama masa perkuliahan.

5. Seluruh staf administrasi dan akademik Program Studi Teknologi Informasi yang telah memberikan bantuan dan kemudahan dalam urusan birokrasi perkuliahan.
6. Teman-teman dari Program Studi Program Studi Teknologi Informasi angkatan 2022 serta teman-teman dekat yang telah memberikan semangat, doa, dukungan, bantuan, dan kerja sama kepada penulis selama masa studi hingga penyusunan tugas ini.

Penulis dengan jujur menyadari bahwa masih banyak kekurangan dalam tugas akhir ini, baik dari segi teknis penulisan maupun kedalaman konten. Oleh karena itu, penulis dengan tangan terbuka dan kerendahan hati menerima segala kritik dan saran yang membangun untuk penyempurnaan di masa depan.

Terakhir, semoga tugas akhir ini dinilai sebagai amal ibadah di sisi Allah SWT dan dapat memberikan manfaat bagi penulis, pembaca, serta perkembangan ilmu pengetahuan. Aamiin Ya Rabbal 'Alamin.

Banda Aceh, 5 Agustus 2026

Penulis,

Dhiyau Auliya

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Transformasi layanan publik ke arah digital telah mengubah cara masyarakat berinteraksi dengan penyedia layanan, termasuk di sektor kelistrikan. PT PLN (Persero) menghadirkan aplikasi PLN Mobile sebagai solusi layanan terintegrasi. Popularitas aplikasi ini terlihat dari jumlah unduhan yang telah menembus angka 10 juta pengguna di Google Play Store. Tingginya angka adopsi ini secara otomatis menghasilkan lonjakan volume ulasan (*feedback*) dari pengguna yang terus bertambah setiap harinya.

Ulasan-ulasan ini sejatinya merupakan aset data yang bernilai tinggi. Sebagaimana dijelaskan dalam riset (Syafrizal dkk., 2023), ulasan pengguna PLN Mobile sering kali mengandung rincian kendala teknis dan keluhan spesifik yang dapat menjadi indikator langsung kepuasan pelanggan. Namun, jumlah ulasan yang sangat masif menjadikan proses analisis manual menjadi tidak realistis. (Shafira Akbar Rizki dkk., 2025) menekankan bahwa penggunaan teknologi otomatisasi seperti *Natural Language Processing* (NLP) sangat mendesak untuk dilakukan agar data ulasan tersebut dapat diolah menjadi wawasan yang objektif tanpa memakan waktu lama.

Tantangan berikutnya terletak pada kedalaman analisis. Metode analisis sentimen konvensional sering kali hanya memberikan label umum (positif/negatif) pada satu ulasan penuh. Padahal, realitas di lapangan menunjukkan bahwa satu ulasan bisa memiliki sentimen yang beragam bergantung pada fitur yang dibahas. (Hilal dkk., 2024) menyoroti bahwa pendekatan berbasis aspek (*aspect-based*) jauh lebih efektif untuk kasus PLN Mobile, karena pengguna kerap memuji kemudahan transaksi token tetapi di saat bersamaan mengeluhkan respons penanganan gangguan (Hilal dkk., 2024). Hal senada ditemukan oleh (Ihsan Zulfahmi, 2023), yang mencatat adanya ketimpangan sentimen antar-fitur; misalnya fitur *"Pengaduan" yang cenderung lebih banyak mendapat sentimen negatif* dibandingkan fitur pembayaran. Oleh karena itu, penelitian ini akan menerapkan klasifikasi sentimen multikelas yang difokuskan pada kategori aspek

layanan dominan, agar evaluasi yang dihasilkan dapat menunjuk langsung pada fitur mana yang perlu diperbaiki.

Dalam proses evaluasi pemodelan, penelitian ini berfokus pada analisis komparatif antara metode gabungan dan penggunaan algoritma tunggal (*single classifier*). Penggunaan model tunggal seperti Naive Bayes, Logistic Regression, maupun Support Vector Machine sering kali memiliki keterbatasan dalam menjaga konsistensi prediksi pada kelas minoritas (*minority class*) ketika berhadapan dengan data ulasan yang sangat tidak seimbang (*imbalanced data*). Model-model tunggal tersebut rentan mengalami bias pada kelas mayoritas dan kurang sensitif dalam mendeteksi ekspresi sentimen pada kelas minoritas. Oleh karena itu, penelitian ini membandingkan kinerja pendekatan *Weighted Soft Voting Ensemble* terhadap seluruh model tunggal penyusunnya guna menguji secara empiris apakah integrasi model mampu memberikan stabilitas dan keseimbangan nilai *F1-Score* yang lebih baik.

Kerentanan terhadap data tidak seimbang (*imbalanced*) memang menjadi kelemahan mendasar pada mayoritas algoritma klasifikasi tunggal. Sebagaimana dicatat oleh Syafrizal dkk. (2023), bahkan algoritma populer seperti Naive Bayes pun berpotensi menghasilkan bias pada kelas mayoritas jika dioperasikan secara mandiri. Untuk mengatasi kelemahan kolektif dari model-model tunggal ini, diperlukan strategi penggabungan model melalui *Ensemble Learning*.

Penelitian ini mengusulkan solusi berupa penerapan metode Weighted Soft Voting Ensemble. Metode ini tidak sekadar melakukan pemungutan suara biasa, melainkan menggabungkan prediksi probabilitas dari empat algoritma berbeda yaitu *Multinomial Naive Bayes* (MNB), *Logistic Regression* (LR), *SGD Classifier*, dan *Linear SVC* dengan memberikan bobot lebih pada model yang terbukti paling handal. Dengan menggabungkan kekuatan dari berbagai algoritma ini, diharapkan sistem dapat menghasilkan prediksi sentimen yang lebih stabil dan akurat, serta mampu memetakan permasalahan layanan PLN Mobile secara lebih komprehensif.

Meskipun penelitian ini mengangkat konteks kategori aspek layanan, pendekatan yang digunakan bukanlah Aspect-Based Sentiment Analysis (ABSA) yang memungkinkan satu ulasan memiliki lebih dari satu aspek dan sentimen.

Penelitian ini menerapkan pendekatan aspek dominan (dominant aspect approach), di mana setiap ulasan dipetakan hanya ke satu kategori aspek layanan yang paling menonjol serta satu kelas sentimen (positif, negatif, atau netral).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengembangkan dan mengimplementasikan sistem klasifikasi sentimen multikelas (positif, negatif, dan netral) pada ulasan aplikasi PLN Mobile dengan menggunakan pendekatan aspek dominan melalui metode Weighted Soft Voting Ensemble?
2. Bagaimana perbandingan performa metode Weighted Soft Voting Ensemble terhadap model-model tunggal penyusunnya (Multinomial Naive Bayes, Logistic Regression, SGD Classifier, dan Support Vector Machine) berdasarkan metrik Accuracy, Precision, Recall, dan F1-Score?

1.3 Tujuan Penelitian

Sejalan dengan rumusan masalah di atas, tujuan dari penelitian ini adalah:

1. Mengembangkan model klasifikasi dan prototipe sistem analitik sentimen multikelas (positif, negatif, dan netral) pada ulasan aplikasi PLN Mobile dengan menggunakan pendekatan aspek dominan melalui metode Weighted Soft Voting Ensemble.
2. Menganalisis perbandingan performa model klasifikasi berdasarkan metrik Accuracy, Precision, Recall, dan F1-Score antara metode Weighted Soft Voting Ensemble dan seluruh model tunggal penyusunnya (Multinomial Naive Bayes, Logistic Regression, SGD Classifier, dan Support Vector Machine) guna mengukur efektivitas integrasi model pada data ulasan dengan representasi fitur TF-IDF

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi positif, baik dari sisi teoritis maupun praktis:

1.4.1 Manfaat Teoritis (Akademis)

1. Memberikan bukti empiris mengenai performa dan konsistensi prediksi metode *Weighted Soft Voting Ensemble* pada tugas klasifikasi sentimen

multikelas dalam menangani data teks dengan representasi fitur jarang (*sparse*) yang sering ditemukan pada ulasan berbahasa Indonesia.

2. Memberikan kontribusi pada pengembangan literatur terkait pendekatan klasifikasi sentimen berbasis kategori aspek layanan (*predefined aspect*) sebagai alternatif efisien tanpa melalui proses ekstraksi aspek otomatis yang kompleks.

1.4.2 Manfaat Praktis

1. Hasil klasifikasi yang disajikan melalui prototipe *dashboard* analitik dapat dimanfaatkan langsung oleh pihak manajemen PLN sebagai bahan evaluasi layanan. Evaluasi ini didasarkan pada pemetaan aspek dominan yang paling sering mendapat sentimen positif maupun negatif, sehingga mampu mendukung pengambilan keputusan yang tepat sasaran dalam perbaikan fitur aplikasi PLN Mobile.
2. Penelitian ini juga ditujukan bagi akademisi dan peneliti selanjutnya sebagai referensi komprehensif terkait rancangan alur pra-pemrosesan teks (*preprocessing pipeline*) dan strategi penanganan ketidakseimbangan kelas (*imbalanced data*) pada pemodelan *machine learning* untuk dataset ulasan aplikasi layanan publik.

1.5 Batasan Penelitian

Agar penelitian ini tetap terarah dan tidak meluas dari topik pembahasan, penulis menetapkan beberapa batasan masalah sebagai berikut:

1. Data yang diolah dalam penelitian ini difokuskan pada ulasan berbahasa Indonesia yang diberikan oleh pengguna aplikasi PLN Mobile melalui platform Google Play Store pada periode waktu tertentu, sehingga ulasan dengan bahasa asing sepenuhnya berada di luar cakupan analisis.
2. Skema ekstraksi aspek diterapkan secara otomatis melalui pendekatan berbasis leksikon (*Lexicon-based/Keyword matching*) yang mencakup lima kategori utama, yakni Pembayaran & Token, Pelayanan Pelanggan, Performa Aplikasi, Fitur & Fungsionalitas, serta Antarmuka Pengguna (UI/UX). Setiap ulasan akan diprediksi polaritas sentimen utamanya oleh model *Machine Learning*, untuk kemudian dipetakan ke dalam satu atau

beberapa aspek layanan secara *multi-label* berdasarkan kata kunci yang terdeteksi.

3. Pendekatan komputasi yang digunakan adalah model klasifikasi *Weighted Soft Voting Ensemble* yang diintegrasikan dari empat algoritma dasar, yaitu *Multinomial Naive Bayes, Logistic Regression, SGD Classifier,* dan *Linear SVC*. Evaluasi performa model gabungan ini nantinya akan dibandingkan secara komprehensif dengan seluruh model tunggal penyusunnya untuk melihat sejauh mana kontribusi pendekatan *ensemble* dalam meningkatkan stabilitas prediksi.
4. Proses ekstraksi fitur teks secara eksklusif menggunakan representasi pembobotan TF-IDF yang dikombinasikan dengan N-gram. Oleh karena itu, penelitian ini tidak mengeksplorasi representasi vektor kata berbasis *Deep Learning* seperti Word2Vec, GloVe, maupun arsitektur BERT.
5. Standar evaluasi performa klasifikasi diukur murni berdasarkan *Confusion Matrix* melalui kalkulasi metrik *Accuracy, Precision, Recall,* dan *F1-Score*. Adapun luaran akhir dari penelitian ini diimplementasikan dalam bentuk prototipe *dashboard* analitik berbasis kerangka kerja Streamlit sebagai sarana demonstrasi konsep, dan belum dirancang untuk penerapan produksi pada skala industri.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai dasar dalam memahami perkembangan metode analisis sentimen, khususnya pada ulasan aplikasi layanan publik. Beberapa penelitian yang relevan dengan topik penelitian ini diuraikan sebagai berikut.

Tabel 2. 1 Penelitian Terkait

No	Peneliti & Tahun	Judul / Topik	Metode	Kesimpulan
1	Syafrizal, Afdal, & Novita (2023)	Analisis Sentimen Ulasan Aplikasi PLN Mobile	Naive Bayes Classifier (NBC) dan K-Nearest Neighbor (KNN)	Algoritma Naive Bayes menghasilkan akurasi terbaik (86%) dibandingkan KNN, namun penelitian ini hanya membandingkan model tunggal secara terpisah tanpa menerapkan metode Ensemble untuk menggabungkan kekuatan kedua model guna meningkatkan stabilitas prediksi.
3	Zulfahmi & Damanik (2024)	Analisis Sentimen Aplikasi PLN Mobile	Decision Tree (Algoritma C4.5)	Mampu memetakan atribut sentimen dengan akurasi sekitar 79%, akan tetapi model berbasis pohon (Tree-based) sering kali rentan overfitting pada data teks yang jarang (sparse), sehingga diperlukan pendekatan linear atau ensemble yang lebih efisien.
4	Rizki et al. (2025)	Klasifikasi Sentimen Ulasan Pengguna Aplikasi Layanan Publik	Komparasi Metode NLP & Machine Learning Klasik	Menunjukkan bahwa teknik NLP sangat efektif untuk otomatisasi analisis ulasan layanan publik, namun studi ini bersifat umum dan belum secara spesifik menerapkan mekanisme pembobotan dinamis (weighted voting) untuk menangani ketidakseimbangan kelas pada kasus PLN Mobile.
5	Rizwan dkk. (2024)	*Depression Intensity Classification from Tweets*	FastText-based Weighted Soft Voting	Mengusulkan arsitektur *weighted soft voting ensemble* (WSVE) dengan menggabungkan beberapa *classifier* dan

		Using FastText Based Weighted Soft Voting Ensemble	Ensemble (WSVE)	memberikan bobot berdasarkan performa individunya. Pendekatan ini terbukti mengungguli seluruh model *baseline* dengan capaian *F1-score* 89% dan akurasi 93%, serta lebih tangguh dalam menangkap fitur kontekstual pada kelas dengan jumlah sampel yang kecil.
6	Atmaja dkk. (2026)	*Comparing Machine Learning and Optimized Ensemble Models for Student Retention*	*Optimized Ensemble Models* (Weighted Soft Voting dan SMOTE)	Membandingkan performa model pembelajaran mesin tunggal dengan pendekatan *ensemble* yang dioptimasi. Hasilnya menunjukkan bahwa penerapan *soft voting* secara konsisten mengungguli model-model dasar secara mandiri (seperti Naive Bayes, SVM, KNN, dan Logistic Regression), dengan pencapaian akurasi tertinggi sebesar 87,55%.

Berdasarkan tinjauan pada Tabel II.1, penelitian terdahulu umumnya masih berfokus pada penggunaan model tunggal. Oleh karena itu, penelitian ini difokuskan untuk mengeksplorasi potensi penggabungan beberapa model linear melalui Weighted Soft Voting dalam meningkatkan stabilitas prediksi, khususnya dalam menangani data dengan karakteristik ketidakseimbangan kelas.

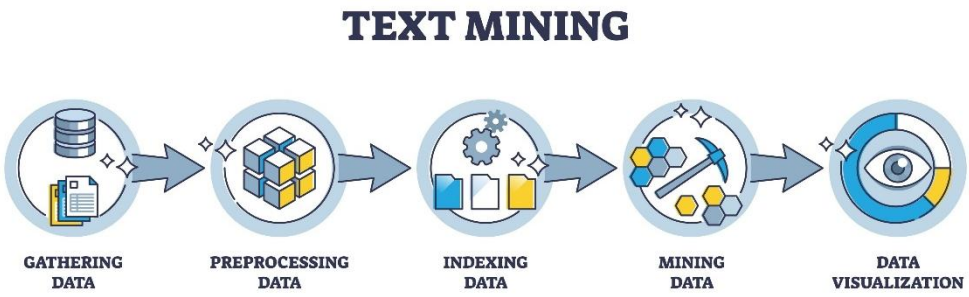
Hingga saat ini, belum banyak penelitian yang menerapkan pendekatan Ensemble Learning dengan metode Weighted Soft Voting secara spesifik pada kasus PLN Mobile. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan menggabungkan empat algoritma linear (MNB, LR, SGD, dan SVC) yang memiliki karakteristik matematis berbeda. Sinergi ini diharapkan mampu menghasilkan model klasifikasi yang tidak hanya akurat, tetapi juga lebih tangguh (*robust*) dibandingkan model tunggal yang digunakan pada penelitian-penelitian sebelumnya.

2.2 Landasan Teori

2.2.1 Text Mining Dan Natural Language Processing (NLP)

Text Mining, atau sering disebut sebagai penambangan data teks, adalah proses ekstraksi pola atau pengetahuan yang menarik dan non-trivial dari data teks

yang tidak terstruktur. Menurut (Feldman, 2007), perbedaan mendasar antara *data mining* konvensional dan *text mining* terletak pada struktur datanya; jika *data mining* bekerja pada basis data terstruktur (angka/tabel), maka *text mining* berfokus pada pengolahan data semi-terstruktur atau tidak terstruktur—seperti ulasan pengguna, *email*, atau dokumen web—untuk diubah menjadi informasi yang terstruktur dan bermakna.



Gambar 2. 1 Proses Text Mining

Proses ini sangat erat kaitannya dengan *Natural Language Processing* (NLP). (Jurafsky, 2023) mendefinisikan NLP sebagai disiplin ilmu komputasi yang berfokus pada pemodelan interaksi antara komputer dan bahasa alami manusia, baik dalam bentuk teks maupun suara. Dalam konteks analisis sentimen, NLP berfungsi sebagai fondasi teknis untuk memecah dan memahami struktur bahasa sebelum dilakukan analisis statistik.

Secara teknis, alur kerja NLP dalam *text mining* melibatkan tahapan pra-pemrosesan (*preprocessing*) yang krusial. Tahapan ini meliputi *case folding*, *tokenizing*, *stopword removal*, dan *stemming*. Tujuannya adalah untuk mengurangi kompleksitas ruang fitur (*feature space*) yang sering kali berdimensi tinggi dan jarang (*sparse*) pada representasi teks, sehingga data lebih siap untuk diproses oleh algoritma klasifikasi seperti yang diterapkan dalam penelitian ini pada ulasan aplikasi PLN Mobile.

2.2.2 Pra-pemrosesan Teks (*Text Preprocessing*)

Pra-pemrosesan teks (*text preprocessing*) adalah tahapan fundamental dalam *text mining* yang bertujuan untuk mentransformasi data teks mentah yang tidak terstruktur menjadi format yang bersih dan terstandar. (Manning, 2008), kualitas data hasil pra-pemrosesan sangat menentukan akurasi model klasifikasi,

karena tahapan ini berfungsi mereduksi *noise* dan mengurangi dimensi fitur (*dimensionality reduction*) sebelum data diproses lebih lanjut menggunakan metode pembobotan seperti TF-IDF.

Dalam penelitian ini, tahapan pra-pemrosesan dilakukan secara sekuensial sebagai berikut:

1. *Case Folding* Tahap ini bertujuan mengubah seluruh huruf dalam dokumen menjadi huruf kecil (*lowercase*). Tujuannya adalah untuk menyeragamkan data (*canonicalization*) sehingga mesin dapat mengenali kata yang sama meskipun ditulis dengan variasi huruf kapital yang berbeda (misalnya: "PLN", "Pln", dan "pln" dianggap sebagai token yang identik).
2. *Data Cleaning* Proses ini berfokus pada penghapusan elemen derau (*noise*) yang tidak relevan dengan konteks analisis sentimen. Elemen yang dihapus meliputi angka, tanda baca, simbol khusus, *emoticon*, serta tautan (URL). Pembersihan ini penting untuk memastikan bahwa fitur yang terbentuk nantinya hanya berisi informasi linguistik yang bermakna (Feldman, 2007)
3. Normalisasi Kata (*Word Normalization*) Mengingat data ulasan aplikasi *mobile* sering kali menggunakan bahasa tidak baku, singkatan, atau bahasa gaul (*slang*), proses normalisasi sangat diperlukan. Tahap ini mengubah kata-kata non-standar menjadi bentuk baku sesuai Kamus Besar Bahasa Indonesia (KBBI). Hal ini dilakukan untuk mengatasi masalah *sparsity* (kelangkaan data) akibat variasi penulisan kata yang memiliki makna sama (misalnya: mengubah "gak", "gk", "gq" menjadi satu bentuk baku "tidak").
4. Tokenisasi (*Tokenization*) Tokenisasi adalah proses pemecahan kalimat menjadi unit-unit kata yang berdiri sendiri yang disebut token. Menurut (Webster, 1992), tokenisasi memutus rangkaian karakter berdasarkan pemisah tertentu (*delimiter*), seperti spasi, untuk memudahkan analisis pada level kata.
5. *Stopword Removal* Proses ini bertujuan menghapus kata-kata umum yang memiliki frekuensi kemunculan tinggi namun minim informasi sentimen

(*stop words*), seperti "yang", "dan", "di", "ke". (Manning dkk., 2008) menjelaskan bahwa penghapusan ini efektif mengurangi dimensi fitur tanpa menghilangkan inti informasi. Namun, khusus dalam penelitian ini, kata-kata negasi (seperti "tidak", "bukan", "jangan") dikecualikan dari penghapusan karena memiliki fungsi krusial dalam menentukan polaritas sentimen (misalnya: frasa "tidak baik" akan berubah makna menjadi "baik" jika kata "tidak" dihapus).

6. *Stemming* adalah proses morfologi untuk mengubah kata berimbuhan menjadi kata dasar (*root word*) dengan menghilangkan awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*), dan konfiks. Untuk teks Bahasa Indonesia, penelitian ini menerapkan algoritma yang dikembangkan oleh (Adriani dkk., 2007). Algoritma ini merupakan penyempurnaan dari metode *Confix Stripping* (Nazief & Adriani) yang dirancang khusus untuk menangani kompleksitas morfologi bahasa Indonesia melalui aturan pemenggalan imbuhan bertingkat, sehingga variasi kata seperti "membayar", "pembayaran", dan "dibayar" dapat dipetakan secara akurat menjadi satu kata dasar "bayar".

2.2.3 Analisis Sentimen (*Sentiment Analysis*)

Analisis sentimen, atau sering dikenal dengan istilah *opinion mining*, merupakan cabang fundamental dalam pemrosesan bahasa alami (*Natural Language Processing*) yang didedikasikan untuk mengekstraksi dan mempelajari opini, sentimen, evaluasi, serta emosi manusia terhadap entitas tertentu, baik berupa produk, layanan, organisasi, maupun peristiwa. (Liu, 2012), mendefinisikan analisis sentimen sebagai tugas komputasi untuk menentukan orientasi opini dalam teks, apakah cenderung positif, negatif, atau bersifat netral.

Dalam penerapannya pada layanan publik digital seperti aplikasi PLN Mobile, analisis sentimen berfungsi vital untuk mentransformasi data ulasan kualitatif yang tidak terstruktur menjadi wawasan kuantitatif yang terukur. Proses ini tidak terjadi secara instan, melainkan melalui serangkaian tahapan sistematis. Berdasarkan survei literatur yang dilakukan oleh (Medhat dkk., 2014), alur kerja utama dalam analisis sentimen meliputi tahap pra-pemrosesan data, ekstraksi fitur, dan klasifikasi polaritas.

Penelitian ini mengadopsi teori tersebut untuk memetakan kecenderungan persepsi pengguna PLN Mobile di Google Play Store. Berbeda dengan pendekatan biner sederhana, analisis dalam penelitian ini dikembangkan untuk menangkap nuansa opini yang lebih kompleks melalui pendekatan multikelas berbasis aspek, sehingga evaluasi layanan dapat dilakukan secara lebih presisi.

2.2.4 Klasifikasi Sentimen Multikelas

Klasifikasi sentimen multikelas merupakan pengembangan dari pendekatan klasifikasi biner (Bishop, 2006), jika klasifikasi biner hanya memisahkan data ke dalam dua kelas (K=2), maka klasifikasi multikelas melibatkan pemetaan data input ke dalam K > 2 kategori label yang saling lepas (*mutually exclusive*). Artinya, setiap dokumen hanya boleh memiliki satu label kelas yang paling merepresentasikan isinya.

Dalam penelitian ini, skema yang diterapkan adalah Klasifikasi Multikelas Label Tunggal (*Single-Label Multiclass Classification*). Merujuk pada definisi formal (Sebastiani, 2002), jika diberikan himpunan dokumen ulasan $D = \{d_1, d_2, \dots, d_n\}$ dan himpunan kategori kelas C = {Positif, Negatif, Netral}, maka fungsi klasifikasi Φ (Phi) akan memetakan setiap ulasan d_i ke tepat satu kelas $c_j$dari himpunan C.

Definisi operasional ketiga kelas sentimen dalam penelitian ini ditentukan sebagai berikut:

1. Sentimen Positif: Ulasan yang memuat ekspresi kepuasan, pujian, ucapan terima kasih, atau apresiasi eksplisit terhadap kinerja aplikasi dan layanan PLN Mobile.
2. Sentimen Negatif: Ulasan yang berisi keluhan, kritik tajam, laporan kerusakan (*bug*), atau ekspresi kekecewaan terhadap fitur dan layanan.
3. Sentimen Netral: Ulasan yang bersifat objektif, berupa pertanyaan teknis, saran tanpa tendensi emosi, atau pernyataan fakta yang tidak memuat kata sifat sentimen yang kuat.

Tantangan utama dalam klasifikasi yang melibatkan kelas Netral adalah ambiguitas batas keputusan (*decision boundary*). (Koppel, t.t.) dalam penelitiannya menekankan bahwa kelas "Netral" sering kali menjadi *noise* yang sulit dibedakan karena memiliki fitur linguistik yang beririsan dengan kelas positif

maupun negatif (misalnya: kalimat "Saya sudah bayar tapi token belum masuk" bisa dianggap negatif secara konteks, namun netral secara pemilihan kata). Oleh karena itu, diperlukan strategi pemodelan yang mampu menangani kompleksitas tersebut, salah satunya melalui pendekatan *Ensemble Learning*.

2.2.5 Pendekatan Aspek Dominan dalam Klasifikasi Sentimen

Analisis sentimen merupakan salah satu cabang dari Natural Language Processing (NLP) yang bertujuan untuk mengidentifikasi dan mengklasifikasikan opini atau sikap subjektif terhadap suatu entitas ke dalam kategori polaritas tertentu, seperti positif, negatif, atau netral. Definisi konseptual ini dijelaskan secara komprehensif (Liu, 2012) yang menyatakan bahwa analisis sentimen berfokus pada ekstraksi dan pemodelan opini dalam data teks secara sistematis.

Dalam perkembangannya, analisis sentimen tidak hanya dilakukan pada tingkat dokumen secara keseluruhan (document-level sentiment analysis), tetapi juga berkembang menjadi Aspect-Based Sentiment Analysis (ABSA) (Hua dkk., 2024). Pendekatan ABSA memungkinkan identifikasi opini yang dikaitkan dengan aspek tertentu dari suatu entitas, sehingga satu dokumen atau ulasan dapat mengandung beberapa pasangan aspek–sentimen yang berbeda (Hua dkk., 2024). Tinjauan sistematis mengenai perkembangan dan metode ABSA menunjukkan bahwa pendekatan ini umumnya melibatkan tahapan ekstraksi aspek dan klasifikasi polaritas terhadap aspek tersebut (Hua dkk., 2024).

Meskipun ABSA memberikan analisis yang lebih rinci, pendekatan ini sering kali menggunakan skema multi-label classification yang meningkatkan kompleksitas anotasi data dan pemodelan. Selain itu, berbagai implementasi ABSA modern memanfaatkan model deep learning yang membutuhkan data dalam jumlah besar serta sumber daya komputasi yang lebih tinggi (Medhat dkk., 2014). Hal tersebut menunjukkan bahwa pemilihan pendekatan analisis sentimen perlu disesuaikan dengan tujuan dan kompleksitas tugas penelitian (Medhat dkk., 2014).

Sebagai alternatif, pendekatan aspek dominan (dominant aspect approach) dapat diterapkan dengan mengasumsikan bahwa setiap dokumen memiliki satu aspek utama yang paling menonjol. Pendekatan ini termasuk dalam skema single-label multiclass classification, di mana setiap dokumen diberikan satu label aspek

dan satu label sentimen. Secara teoretis, skema klasifikasi multikelas ini didukung oleh konsep pemodelan linear dan probabilistik dalam klasifikasi yang dijelaskan oleh (Bishop, 2006).

Pendekatan aspek dominan dinilai lebih praktis karena mampu menyeimbangkan antara kebutuhan analisis dan efisiensi pemodelan. Dalam konteks ulasan aplikasi layanan publik yang umumnya singkat dan berfokus pada permasalahan utama, pendekatan ini dianggap memadai untuk merepresentasikan opini pengguna tanpa memerlukan skema multi-label yang kompleks. Oleh karena itu, penelitian ini memilih pendekatan aspek dominan guna mendukung fokus pada evaluasi dan perbandingan performa algoritma klasifikasi sentimen secara lebih terkontrol dan efisien.

2.2.6 Ketidakseimbangan Kelas dalam Klasifikasi Teks

Dalam tugas klasifikasi teks, salah satu tantangan yang sering muncul adalah ketidakseimbangan distribusi kelas (*class imbalance*). Ketidakseimbangan kelas terjadi ketika jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Dalam konteks analisis sentimen, kondisi ini umum terjadi karena ulasan positif sering kali lebih dominan dibandingkan ulasan negatif atau netral (Johnson & Khoshgoftaar, 2019).

Distribusi kelas yang tidak seimbang dapat menyebabkan model pembelajaran mesin cenderung bias terhadap kelas mayoritas. Dalam kondisi tersebut, model mungkin menghasilkan nilai akurasi yang tinggi, namun gagal mendeteksi kelas minoritas secara efektif. Fenomena ini menunjukkan bahwa penggunaan metrik evaluasi seperti Accuracy saja tidak cukup representatif untuk menilai performa model pada *imbalanced dataset*, karena metrik tersebut tidak mempertimbangkan distribusi kesalahan pada masing-masing kelas (Buda dkk., 2018; Haixiang dkk., 2017).

Beberapa penelitian terbaru menekankan pentingnya penggunaan metrik evaluasi berbasis keseimbangan seperti Precision, Recall, dan terutama F1-Score dalam kasus distribusi kelas yang tidak merata (Buda dkk., 2018). F1-Score merupakan rata-rata harmonik antara Precision dan Recall, sehingga mampu memberikan gambaran performa model yang lebih adil terhadap kelas minoritas.

Dalam penelitian ini, potensi ketidakseimbangan kelas pada data ulasan dianalisis sebelum proses pelatihan model. Oleh karena itu, evaluasi performa tidak hanya berfokus pada Accuracy, tetapi juga menitikberatkan pada F1-Score sebagai indikator utama dalam membandingkan model tunggal Support Vector Machine (SVM) dan metode *Weighted Soft Voting Ensemble*.

Untuk mengatasi masalah ketidakseimbangan distribusi kelas pada level data, penelitian ini menerapkan metode *Synthetic Minority Over-sampling Technique* (SMOTE) (Chawla dkk., 2002). Algoritma SMOTE bekerja dengan cara membangkitkan data sintetis baru berdasarkan interpolasi linear antara sampel dari kelas minoritas yang ada dengan tetangga terdekatnya (*k-nearest neighbors*). Pendekatan ini secara efektif menyeimbangkan ruang fitur tanpa memicu risiko model sekadar menghafal data (*overfitting*) secara berlebihan. Dengan distribusi kelas yang lebih seimbang, model klasifikasi linear yang dilatih seperti *Logistic Regression* maupun *Support Vector Machine* dapat membentuk batas keputusan (*decision boundary*) yang lebih representatif dan adil bagi seluruh kelas sentimen.

2.2.7 Pembobotan Kata (TF-IDF)

Setelah melewati tahap pra-pemrosesan, data teks yang bersifat kualitatif harus ditransformasikan ke dalam representasi numerik (vektor) agar dapat diolah oleh algoritma pembelajaran mesin. Penelitian ini menerapkan metode TF-IDF (*Term Frequency–Inverse Document Frequency*) sebagai teknik ekstraksi fitur dan pembobotan kata.

Menurut (C. D. Manning dkk., 2008), TF-IDF adalah skema pembobotan statistik yang bertujuan untuk mengukur tingkat kepentingan sebuah kata (*term*) terhadap suatu dokumen tertentu dalam sekumpulan korpus. Esensi dari metode ini adalah memberikan bobot yang tinggi pada kata-kata yang bersifat unik dan memiliki daya pembeda (*discriminative power*), serta mereduksi bobot kata-kata yang terlalu umum.

Nilai bobot TF-IDF diperoleh dari penggabungan dua komponen utama, yaitu:

1. *Term Frequency* (TF)

   Komponen ini merepresentasikan frekuensi lokal, yaitu seberapa sering suatu kata (t) muncul dalam satu dokumen (d). Menurut (Salton

& Buckley, 1988), asumsi dasar dari TF adalah semakin tinggi frekuensi kemunculan sebuah kata dalam dokumen, maka kata tersebut dianggap semakin merepresentasikan isi atau topik dari dokumen tersebut.

2. *Inverse Document Frequency* (IDF)

   Komponen ini merepresentasikan kelangkaan global. IDF berfungsi untuk menimbang signifikansi sebuah kata di seluruh koleksi dokumen (korpus). Kata-kata yang muncul di hampir semua dokumen (seperti kata hubung) akan diberikan nilai IDF yang rendah, sementara kata yang jarang muncul namun spesifik akan memiliki nilai IDF yang tinggi.

   Secara matematis, bobot akhir ($W\_\{t,d\}$) untuk kata (t) dalam dokumen (d) dihitung melalui persamaan berikut:

$$W_{t,d} = TF_{t,d} \times IDF_t$$

Keterangan:

- $W_{t,d}$= bobot kata *t* pada dokumen *d*
- $TF_{t,d}$= frekuensi kemunculan kata *t* dalam dokumen *d*
- IDF_t= inverse document frequency dari kata *t*

Dengan nilai IDF_t dihitung menggunakan rumus logaritmik:

$$IDF_t = \log\left(\frac{N}{DF_t}\right)$$

Keterangan:

- N : Jumlah keseluruhan dokumen dalam *dataset*.
- IDF_t : Jumlah dokumen yang mengandung kata t (*Document Frequency*).

Penerapan TF-IDF pada data ulasan dengan kosakata yang beragam akan menghasilkan matriks fitur berdimensi tinggi yang bersifat jarang (*high-dimensional sparse matrix*). Hal ini terjadi karena setiap kata unik dalam *dataset* akan membentuk satu dimensi fitur baru, sementara setiap dokumen hanya mengandung sebagian kecil dari total kosakata tersebut. Karakteristik data yang *sparse* ini menjadi dasar pemilihan algoritma klasifikasi linear dalam penelitian ini, seperti *Linear SVC* dan *Logistic Regression*, yang dikenal efisien dalam menangani ruang fitur berdimensi tinggi.

2.2.8 Algoritma Klasifikasi

Penelitian ini menerapkan pendekatan *Supervised Machine Learning*, yaitu metode pembelajaran mesin yang mengandalkan sekumpulan data latih berlabel untuk membangun model prediksi. Mengingat hasil ekstraksi fitur TF-IDF menghasilkan matriks berdimensi tinggi dan bersifat jarang (*sparse*), maka dipilih empat algoritma klasifikasi linear sebagai model dasar (*base learners*) yang akan dikombinasikan.

a. *Multinomial Naive Bayes*

Multinomial Naive Bayes (MNB) merupakan salah satu varian dari algoritma Naive Bayes yang dirancang khusus untuk klasifikasi dokumen teks berbasis frekuensi kata atau bobot TF-IDF.(Rennie, 2003) menjelaskan bahwa MNB bekerja dengan menghitung probabilitas kemunculan suatu kata pada kelas tertentu berdasarkan distribusi multinomial, sehingga sangat efisien untuk memproses ruang fitur berdimensi tinggi seperti data teks.

Dalam penelitian ini, Multinomial Naive Bayes (MNB) dipilih sebagai salah satu model dasar (base learner) pembentuk ensemble. Pemilihan ini didasarkan pada karakteristik algoritma MNB yang sangat cepat dan stabil saat memproses data ulasan teks. Meskipun model dasar MNB pada dasarnya rentan terhadap ketidakseimbangan kelas, kelemahan ini telah diselesaikan melalui integrasi dengan teknik penyeimbangan data sintetis (SMOTE) pada tahap pra-pemrosesan. Kehadiran MNB dalam arsitektur Weighted Soft Voting terbukti mampu memberikan prediksi probabilitas yang akurat, sehingga secara efektif melengkapi algoritma berbasis linear lainnya (seperti SVM, SGD, dan Logistic Regression) guna meningkatkan stabilitas keputusan akhir sistem.

b. *Logistic Regression*

Logistic Regression (LR) merupakan model klasifikasi linear yang memprediksi probabilitas keanggotaan suatu kelas menggunakan fungsi logistik (sigmoid). Berbeda dengan model regresi linear, Logistic Regression dirancang untuk menangani permasalahan klasifikasi dengan

memetakan kombinasi linier fitur ke dalam rentang probabilitas antara 0 dan 1. Model ini dapat diperluas untuk klasifikasi multikelas melalui pendekatan *one-vs-rest* atau *multinomial logistic regression* (Hosmer dkk., 2013).

Dalam konteks klasifikasi teks, Logistic Regression dikenal memiliki performa yang kompetitif ketika digunakan dengan representasi fitur berdimensi tinggi seperti TF-IDF. Studi terbaru menunjukkan bahwa model linear seperti Logistic Regression mampu memberikan hasil yang stabil dan akurat pada data teks yang bersifat jarang (*sparse*), terutama ketika dikombinasikan dengan regularisasi L2 untuk mengurangi risiko overfitting (Joulin dkk., 2017; Minaee dkk., 2022).

Keunggulan utama Logistic Regression dalam penelitian ini terletak pada kemampuannya menghasilkan estimasi probabilitas kelas secara langsung. Karakteristik ini sangat penting dalam metode Soft Voting Ensemble, karena memungkinkan penggabungan skor probabilitas dari beberapa model secara terkalibrasi dan proporsional. Oleh karena itu, Logistic Regression menjadi salah satu komponen penting dalam pembentukan model Weighted Soft Voting Ensemble.

c. *Stochastic Gradient Descent*

Stochastic Gradient Descent (SGD) merupakan metode optimasi iteratif yang banyak digunakan untuk melatih model linear dalam skala data besar. Berbeda dengan metode *batch gradient descent* yang memperbarui parameter berdasarkan seluruh data latih sekaligus, SGD melakukan pembaruan bobot model secara bertahap berdasarkan satu sampel atau sebagian kecil data pada setiap iterasi. Pendekatan ini menjadikan SGD lebih efisien secara komputasi dan mampu menangani *dataset* berdimensi tinggi dengan jumlah fitur yang besar (Bottou, 2010).

Dalam konteks klasifikasi teks, SGD sering digunakan sebagai metode optimasi untuk model linear seperti Logistic Regression atau Linear Support Vector Machine. Metode ini dikenal efisien dalam menangani *dataset* berskala besar dan berdimensi tinggi karena pembaruan parameter dilakukan secara iteratif pada sebagian kecil data (Bottou, 2010). Pada

representasi fitur seperti TF-IDF yang bersifat jarang (sparse) dan berdimensi tinggi, pendekatan linear yang dioptimasi menggunakan SGD terbukti efektif dan mampu melakukan generalisasi dengan baik (Bishop, 2006; Joachims, 1998).

Karakteristik efisiensi dan fleksibilitas fungsi *loss* pada SGD menjadikannya salah satu komponen yang relevan dalam pembentukan model ensemble, khususnya ketika dikombinasikan dengan algoritma linear lain untuk meningkatkan stabilitas prediksi.

d. *Linear Support Vector Classifier*

Support Vector Machine (SVM) merupakan algoritma klasifikasi yang bekerja dengan mencari *hyperplane* optimal yang mampu memisahkan data antar kelas dengan margin maksimum. Prinsip dasar SVM adalah memaksimalkan jarak antara batas keputusan (*decision boundary*) dan titik data terdekat dari masing-masing kelas (*support vectors*) (Cortes & Vapnik, 1995).

Dalam konteks klasifikasi teks, penggunaan kernel linear (*Linear SVC*) lebih umum diterapkan dibandingkan kernel non-linear. Hal ini disebabkan oleh karakteristik data teks yang direpresentasikan dalam ruang fitur berdimensi tinggi dan bersifat jarang (*high-dimensional sparse space*), sehingga sering kali dapat dipisahkan secara linear (Aggarwal, 2018; Joachims, 1998). Penelitian terbaru juga menunjukkan bahwa Linear SVC tetap menjadi salah satu algoritma yang kompetitif dalam klasifikasi teks modern karena kestabilan performanya serta efisiensi komputasi pada *dataset* besar (Minaee dkk., 2022).

Meskipun Linear SVC secara *default* tidak menghasilkan probabilitas kelas secara langsung, dalam penelitian ini model dikalibrasi menggunakan metode *probability calibration* agar dapat menghasilkan estimasi probabilitas yang terstandarisasi. Langkah ini memungkinkan Linear SVC untuk diintegrasikan ke dalam mekanisme *Weighted Soft Voting Ensemble*, sehingga kontribusinya terhadap keputusan akhir dapat dihitung secara proporsional bersama model linear lainnya.

Selain berperan sebagai salah satu base learner, Linear SVC diuji kinerjanya secara mandiri bersama base learners lainnya untuk mengevaluasi kontribusi metode Weighted Soft Voting Ensemble dalam meminimalkan kesalahan klasifikasi.

2.2.9 Ensemble Learning dan Weighted Soft Voting

Ensemble Learning merupakan paradigma dalam pembelajaran mesin yang mengintegrasikan beberapa model dasar (*base learners*) untuk menghasilkan satu keputusan prediksi yang lebih stabil dan akurat dibandingkan penggunaan model tunggal. Menurut (Polikar, 2006), tujuan utama dari pendekatan ini adalah untuk meningkatkan kemampuan generalisasi model serta meminimalkan risiko kesalahan prediksi yang berasal dari varians atau bias pada satu algoritma tertentu.

Salah satu teknik *ensemble* yang populer adalah *Voting Classifier*, yaitu mekanisme penggabungan keputusan dari berbagai model melalui sistem pemungutan suara. Berdasarkan cara perhitungannya, metode *voting* terbagi menjadi dua jenis utama:

1. Hard Voting: Menentukan hasil akhir berdasarkan suara mayoritas dari label kelas yang diprediksi oleh setiap model tanpa mempertimbangkan tingkat keyakinan model tersebut.
2. Soft Voting: Menentukan keputusan berdasarkan rata-rata probabilitas prediksi yang dihasilkan oleh seluruh model. Menurut (Zhou, 2012), *Soft Voting* cenderung memberikan hasil yang lebih halus dan akurat karena memperhitungkan bobot keyakinan (*confidence score*) dari masing-masing algoritma.

Penelitian ini secara spesifik menerapkan pendekatan Weighted Soft Voting Ensemble. Berbeda dengan Soft Voting standar, metode ini memberikan bobot (w) yang berbeda pada setiap model dasar berdasarkan performa kinerjanya, misalnya melalui nilai F1-Score pada data validasi (Polikar, 2006; Zhou, 2012). Secara matematis, prediksi kelas akhir $\hat{y}$ diperoleh dengan mencari nilai argumen maksimum dari penjumlahan probabilitas kelas yang telah dikalikan dengan bobot masing-masing model, sesuai persamaan berikut.

$$\hat{y} = \arg\max_{j} \sum_{i=1}^{n} w_i \cdot P_{i,j}$$

Keterangan:

$\hat{y}$	= Prediksi akhir
arg max	= Memilih kelas dengan nilai terbesar
w_i	= Bobot model ke-i
$P_{i,j}$	= Probabilitas kelas ke-j dari model ke-i
n	= Jumlah model

Dengan mekanisme ini, model yang terbukti lebih handal dalam mengenali sentimen ulasan PLN Mobile akan memiliki pengaruh lebih besar dalam menentukan hasil klasifikasi akhir, sehingga diharapkan dapat menutupi kelemahan model lain yang memiliki performa lebih rendah.

2.2.10 Evaluasi Performa Model

Evaluasi performa bertujuan untuk memvalidasi efektivitas sistem dalam mengklasifikasikan sentimen ulasan secara objektif. Alat ukur utama yang digunakan dalam penelitian ini adalah *Confusion Matrix*. Menurut (Sokolova & Lapalme, 2009), *Confusion Matrix* merupakan tabel kontingensi yang memetakan perbandingan antara label aktual (*ground truth*) dengan hasil prediksi model, sehingga memungkinkan identifikasi jenis kesalahan klasifikasi yang terjadi.

Komponen dasar dalam *Confusion Matrix* untuk klasifikasi multikelas dievaluasi secara individual untuk masing-masing kelas (misalnya Kelas C). Komponen tersebut meliputi:

- True Positive (TP): Jumlah data yang *sebenarnya* termasuk dalam Kelas C dan *berhasil ditebak* dengan benar oleh model sebagai Kelas C.
- True Negative (TN): Jumlah data yang *bukan* termasuk Kelas C, dan model *dengan benar menebaknya* sebagai bukan Kelas C.
- False Positive (FP) / Kesalahan Tipe I: Jumlah data yang *sebenarnya bukan* Kelas C, tetapi *salah ditebak* oleh model sebagai Kelas C.
- False Negative (FN) / Kesalahan Tipe II: Jumlah data yang *sebenarnya* termasuk dalam Kelas C, tetapi model *gagal menebaknya* (malah diprediksi sebagai kelas lain).

Berdasarkan komponen tersebut, kinerja model diukur menggunakan empat metrik evaluasi utama:

a. *Accuracy*

Accuracy mengukur persentase ketepatan model dalam memprediksi kelas yang benar dibandingkan dengan keseluruhan data. Dalam klasifikasi multikelas, evaluasi dapat dilakukan untuk masing-masing kelas (misalnya Kelas c) dengan membandingkannya terhadap kelas lain menggunakan pendekatan *One-vs-Rest*.

$$\text{Accuracy}_c = \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c}$$

Keterangan:

- Accuracy_c = Nilai akurasi khusus untuk Kelas c.
- TP_c (*True Positive*) = Jumlah data Kelas c yang diprediksi dengan benar sebagai Kelas c.
- TN_c (*True Negative*) = Jumlah data bukan Kelas c yang juga dengan benar diprediksi sebagai bukan Kelas c.
- FP_c (*False Positive*) = Jumlah data bukan Kelas c yang salah ditebak sebagai Kelas c.
- FN_c (*False Negative*) = Jumlah data Kelas c yang salah ditebak sebagai kelas lain.

b. *Precision*

Dalam klasifikasi multikelas, *Precision* dihitung secara individual untuk masing-masing kelas (misalnya Kelas c). Metrik ini mengukur tingkat ketepatan antara data yang diprediksi sebagai Kelas c dengan data yang benar-benar berlabel Kelas c. Nilai presisi yang tinggi meminimalkan kesalahan Tipe I (*False Positive*).

$$Precision_c = \frac{TP_c}{TP_c + FP_c}$$

Keterangan:

- $Precision_c$ = Tingkat presisi untuk Kelas $_c$.
- TP_c (*True Positive*) = Jumlah data Kelas $_c$ yang diprediksi benar sebagai Kelas $_c$.
- FP_c (*False Positive*) = Jumlah data *bukan* Kelas $_c$ yang salah ditebak sebagai Kelas $_c$.

c. *Recall* (Sensitivitas)

Dalam klasifikasi multikelas, *Recall* dihitung secara spesifik untuk masing-masing kelas. Metrik ini mengukur keberhasilan model dalam

mengidentifikasi seluruh sampel aktual yang benar-benar ada pada Kelas c. Nilai *recall* yang tinggi menunjukkan bahwa sistem mampu meminimalkan data yang terlewat atau kesalahan Tipe II (*False Negative*)

$$\text{Recall}_c = \frac{TP_c}{TP_c + FN_c}$$

Keterangan:

- Recall_c = Tingkat *recall* untuk Kelas c.
- TP_c (*True Positive*) = Jumlah data Kelas c yang diprediksi benar sebagai Kelas c.
- FN_c (*False Negative*) = Jumlah data Kelas c yang justru ditebak secara salah sebagai kelas lain.

d. *F1-Score*

F1-Score merupakan rata-rata harmonik antara Precision dan Recall. Dalam pemodelan multikelas, nilai F1-Score dihitung untuk tiap kelas, lalu ditarik nilai rata-ratanya menggunakan pendekatan pembobotan proporsional (Weighted Average). Metrik ini menjadi parameter paling krusial karena memberikan penilaian performa yang jauh lebih adil dan seimbang saat menghadapi kondisi *dataset* yang memiliki distribusi kelas tidak merata (imbalanced *dataset*).

$$F1\text{-}Score_c = 2 \text{ X } \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c}$$

Keterangan:

- $F1\text{-}Score_c$ = Nilai *F1-Score* untuk Kelas c.
- Precision_c = Tingkat ketepatan prediksi khusus pada Kelas c.
- Recall_c = Tingkat keberhasilan model mendeteksi seluruh data Kelas c.

2.2.11 Kerangka Kerja Streamlit (Streamlit Framework)

Streamlit merupakan kerangka kerja (framework) berbasis Python bersifat sumber terbuka (open-source) yang dirancang khusus untuk mempercepat pengembangan aplikasi web interaktif yang berorientasi pada data. Menurut (Khor, 2023), keunggulan utama Streamlit terletak pada kemudahannya bagi para praktisi data untuk mentransformasi skrip analisis Python menjadi aplikasi web fungsional tanpa harus menguasai teknologi pengembangan front-end konvensional seperti HTML, CSS, atau JavaScript secara mendalam.

Dalam ekosistem pembelajaran mesin, Streamlit berfungsi sebagai jembatan untuk menampilkan visualisasi data dan hasil prediksi model secara real-time. Karakteristik ini menjadikannya platform yang sangat efektif untuk membangun prototipe sistem analitik yang memerlukan interaksi langsung antara pengguna dan model.

Dalam penelitian ini, Streamlit diimplementasikan sebagai antarmuka pengguna (user interface) yang mengintegrasikan model klasifikasi sentimen dengan fitur visualisasi. Fungsi-fungsi utama yang disediakan dalam sistem ini meliputi:

1. Input Teks: Menyediakan komponen interaktif yang memungkinkan pengguna memasukkan teks ulasan secara langsung untuk diuji oleh model klasifikasi.
2. Visualisasi Hasil Prediksi: Menampilkan label sentimen akhir (Positif, Negatif, atau Netral) beserta skor probabilitas (confidence score) dari model Ensemble Weighted Soft Voting untuk menunjukkan tingkat keyakinan prediksi.
3. Eksplorasi Statistik: Menyajikan grafik distribusi sentimen dan visualisasi frekuensi kata (seperti Word Cloud) secara dinamis dari *dataset* ulasan yang diproses.

Pemilihan Streamlit dalam penelitian ini didasarkan pada efisiensi proses prototyping, sehingga fokus penelitian dapat diarahkan sepenuhnya pada optimasi model klasifikasi dan analisis performa sistem tanpa terhambat oleh kompleksitas teknis pengembangan antarmuka web tradisional.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

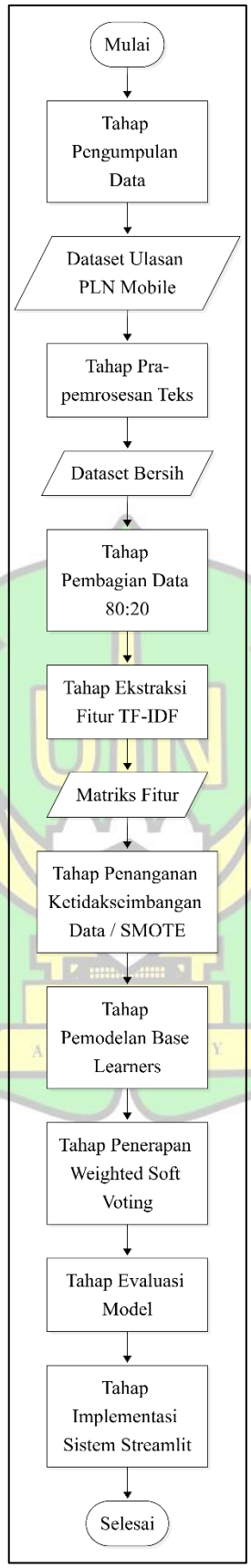
Penelitian ini menggunakan metode kuantitatif dengan desain eksperimental (*experimental research*). Pendekatan kuantitatif dipilih karena fokus utama penelitian adalah pada pengukuran kinerja model klasifikasi secara empiris melalui data numerik dan metrik evaluasi statistik. Menurut Creswell (2014), penelitian kuantitatif bertujuan untuk menguji teori melalui pengamatan hubungan antar variabel yang diukur secara statistik.

Sifat eksperimental dalam penelitian ini diwujudkan melalui pengujian metode Weighted Soft Voting Ensemble dengan berbagai skenario pembobotan. Eksperimen dilakukan untuk menganalisis sejauh mana integrasi model *Multinomial Naive Bayes, Logistic Regression, SGD Classifier,* dan *Linear SVC* dapat meningkatkan akurasi klasifikasi sentimen dibandingkan penggunaan model tunggal.

Selain itu, penelitian ini juga mencakup aspek rekayasa perangkat keras dan sistem (*system engineering*). Berdasarkan kerangka kerja Pressman (2014), penelitian ini mengikuti tahapan pengembangan sistem untuk menghasilkan purwarupa aplikasi analisis sentimen berbasis web sebagai instrumen implementasi model.

3.2 Alur Penelitian

Alur penelitian merupakan kerangka kerja yang menggambarkan urutan langkah-langkah penyelesaian masalah yang dilakukan dalam penelitian ini. Diagram alur penelitian disusun untuk memberikan gambaran sistematis mengenai proses transformasi data ulasan mentah menjadi sistem klasifikasi sentimen berbasis *ensemble* yang dapat diimplementasikan dalam bentuk aplikasi interaktif. Secara umum, alur penelitian ini divisualisasikan dalam diagram alir (*flowchart*) pada Gambar 3.1.



Gambar 3. 1 Flowchart Metodologi Penelitian

Berdasarkan diagram alir di atas, tahapan penelitian ini dapat diuraikan secara rinci sebagai berikut:

1. Tahap Pengumpulan Data merupakan langkah awal yang bertujuan untuk menghimpun data mentah berupa ulasan pengguna aplikasi PLN Mobile. Data ini ditarik langsung dari platform Google Play Store untuk dijadikan sebagai bahan baku utama dalam penelitian.
2. Tahap Pra-pemrosesan Teks (*Text Preprocessing*) dilakukan untuk membersihkan dan menyeragamkan data teks agar terbebas dari derau (*noise*). Rangkaian proses ini memastikan data siap diolah lebih lanjut melalui serangkaian tahapan berurutan yang meliputi *Case Folding, Cleaning, Normalization, Tokenizing, Stopword Removal*, dan *Stemming*.
3. Tahap Pembagian Data (*Data Splitting*) membagi dataset yang telah bersih menjadi dua bagian utama, yakni data latih (*training set*) dan data uji (*testing set*). Data latih difungsikan khusus untuk melatih model klasifikasi, sementara data uji disiapkan guna mengevaluasi performa model terhadap data baru. Pemisahan ini sangat penting untuk mencegah bias evaluasi dan memastikan model memiliki kemampuan generalisasi yang baik di lapangan.
4. Tahap Ekstraksi Fitur (TF-IDF) berfungsi mengubah data teks menjadi representasi numerik atau matriks vektor menggunakan algoritma *Term Frequency-Inverse Document Frequency*. Proses pengenalan pola atau pembentukan vektor (*fit*) diterapkan secara eksklusif pada data latih. Model pembobotan yang dihasilkan dari data latih tersebut baru kemudian digunakan untuk mentransformasi data uji guna menghindari masalah kebocoran data (*data leakage*).
5. Tahap Penanganan Ketidakseimbangan Data (SMOTE) diaplikasikan tepat setelah teks bertransformasi menjadi matriks vektor TF-IDF. Melalui algoritma *Synthetic Minority Over-sampling Technique*, sampel pada kelas minoritas seperti ulasan negatif dan netral akan diperbanyak secara sintetis agar seimbang dengan dominasi ulasan positif. Penerapan SMOTE ini dibatasi hanya pada matriks data latih untuk mencegah manipulasi evaluasi, sedangkan data uji dibiarkan berada pada distribusi alaminya.

6. Tahap Pemodelan (*Modeling*) mengeksekusi proses pelatihan menggunakan data latih sekaligus melakukan validasi guna menentukan pembobotan yang paling optimal. Alur pemodelan ini mencakup dua langkah utama. Langkah pertama berfokus pada pelatihan empat algoritma dasar, yakni *Multinomial Naive Bayes, Logistic Regression, SGD Classifier*, dan *Linear SVC*. Setelah model tunggal tersebut terlatih, langkah kedua adalah menggabungkan prediksi dari keempatnya menggunakan mekanisme *Weighted Soft Voting Ensemble* demi merumuskan keputusan klasifikasi akhir yang stabil.
7. Tahap Evaluasi Model difokuskan pada pengukuran kinerja klasifikasi menggunakan *Confusion Matrix* yang bertumpu pada kalkulasi metrik *Accuracy, Precision, Recall*, dan *F1-Score*. Pada fase pengujian ini, seluruh algoritma dasar penyusunnya, termasuk *Linear SVC*, akan dievaluasi kinerjanya secara mandiri sebagai model tunggal. Hasil evaluasi dari model-model tunggal tersebut kemudian dikomparasikan dengan capaian dari *Weighted Soft Voting Ensemble* guna menganalisis signifikansi peningkatan metrik serta stabilitas yang diperoleh melalui mekanisme pembobotan gabungan.
8. Tahap Implementasi Sistem merupakan langkah pamungkas yang diwujudkan melalui pembangunan antarmuka purwarupa (*prototype*) berbasis web menggunakan kerangka kerja Streamlit. Tahap ini bertujuan untuk mendemonstrasikan penerapan model *machine learning* secara praktis, di mana *dashboard* tersebut mampu menyajikan visualisasi analisis sentimen dan pemetaan aspek layanan dominan pada ulasan PLN Mobile secara interaktif dan mudah dipahami.

3.3 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan data sekunder berupa ulasan pengguna (*user reviews*) terhadap aplikasi PLN Mobile yang tersedia secara publik pada platform Google Play Store. Proses akuisisi data dilakukan menggunakan teknik *web scraping* dengan bantuan pustaka Python google-play-scraper.

Proses pengambilan data hanya dilakukan terhadap informasi yang bersifat publik dan tidak melibatkan data sensitif maupun informasi pribadi pengguna. Pengumpulan data dilakukan dengan tetap memperhatikan etika penelitian serta kebijakan penggunaan data publik pada platform Google Play Store. Data yang dikumpulkan berupa ulasan berbahasa Indonesia dalam rentang waktu tertentu yang relevan dengan kondisi aplikasi saat ini.

Data mentah hasil proses *scraping* disimpan dalam format .csv (*Comma-Separated Values*). *Dataset* terdiri atas beberapa atribut utama sebagaimana dijelaskan pada Tabel 3.1.

Tabel 3. 1 Atribut Dataset Ulasan PLN Mobile

No	Atribut (Feature)	Keterangan
1	userName	Nama pengguna yang memberikan ulasan. Digunakan hanya sebagai identifikasi awal selama proses pengumpulan data dan tidak digunakan dalam proses pemodelan untuk menjaga privasi pengguna.
2	score	Nilai rating bintang (skala 1–5) yang diberikan pengguna. Digunakan sebagai dasar pelabelan sentimen otomatis.
3	at	Tanggal dan waktu ulasan dipublikasikan. Digunakan untuk memfilter data berdasarkan periode penelitian.
4	content	Teks ulasan yang berisi opini, keluhan, atau saran pengguna. Merupakan data utama dalam proses text mining.

3.3.1 Pelabelan Data (*Data Labeling*)

Penelitian ini menggunakan pendekatan *Supervised Learning*, sehingga setiap dokumen ulasan memerlukan label kelas sebagai target prediksi. Proses pelabelan dalam penelitian ini dibagi menjadi dua tahap utama, yaitu pelabelan sentimen multikelas dan pelabelan kategori aspek dominan.

1. Pelabelan Sentimen (*Sentiment Labeling*)

   Proses pelabelan sentimen (Positif, Negatif, Netral) tidak semata-mata bergantung pada nilai *rating* bintang dari Google Play Store. Hal ini dikarenakan seringnya terjadi ketidaksesuaian antara *rating* yang diberikan pengguna dengan ekspresi linguistik pada isi teks ulasan. Oleh karena itu, penelitian ini menerapkan pendekatan hibrida melalui dua tahapan:

- Pelabelan Awal (*Initial Smart Labeling*): Tahap awal pemetaan polaritas teks menggunakan metode berbasis aturan (*rule-based*) dengan kamus polaritas kata yang dikonstruksi secara mandiri (*custom lexicon*). Kamus ini terdiri dari tiga kategori kata: kata bersentimen negatif kuat (*strong_neg*), kata bersentimen positif kuat (*strong_pos*), dan kata bernuansa netral (*netral_words*). Proses pelabelan dilakukan dengan menghitung kemunculan kata-kata tersebut dalam teks ulasan, yang kemudian dikombinasikan dengan nilai *rating* bintang sebagai sinyal awal, sehingga menghasilkan label tebakan awal untuk setiap ulasan.
- Anotasi dan Validasi Manual (*Human Annotation*): Hasil pelabelan awal dari sistem kemudian dievaluasi dan divalidasi secara manual. Untuk menjaga objektivitas dan konsistensi klasifikasi, proses validasi ini berpedoman teguh pada definisi operasional dan kriteria pelabelan yang telah ditetapkan. Tahap ini bertujuan untuk memastikan akurasi konteks kalimat secara utuh, terutama dalam mengoreksi kesalahan sistem pada teks yang mengandung sarkasme, negasi ganda, atau kalimat bernuansa netral.

2. Pelabelan Kategori Aspek

Selain pelabelan sentimen, setiap data ulasan juga diberikan label kategori aspek berdasarkan topik dominan yang dibahas oleh pengguna. Mengingat keterbatasan sumber daya dan waktu penelitian, pelabelan aspek ini tidak dilakukan pada seluruh populasi data, melainkan dilakukan melalui proses anotasi manual terhadap 1.200 sampel ulasan yang representatif.

Karena penelitian ini menggunakan pendekatan aspek dominan (*single-label*), setiap ulasan dari 1.200 sampel tersebut dibaca secara teliti dan dipetakan secara eksklusif ke dalam satu kategori aspek yang paling menonjol. Hal ini sesuai dengan batasan penelitian yang tidak menerapkan pendekatan *multi-label* guna menjaga efisiensi pemodelan. Definisi operasional dan kriteria pelabelan untuk kelima kategori aspek tersebut dijelaskan secara rinci pada Tabel 3.2.

Tabel 3. 2 Definisi Operasional Kategori Aspek Dominan

No	Kategori Aspek	Definisi Operasional
1	Pembayaran & Token	Ulasan yang berkaitan dengan transaksi keuangan, pembelian token, tagihan pascabayar, dan riwayat pembayaran.(kata kunci: beli token, bayar tagihan, saldo, harga, struk, gagal bayar)
2	Pelayanan Pelanggan	Ulasan yang menyoroti kinerja petugas lapangan, respons Customer Service (CS), dan penanganan pengaduan.(kata kunci: respon cs, petugas, pengaduan lambat, teknisi, pelayanan, call center)
3	Performa Aplikasi	Ulasan yang membahas stabilitas sistem teknis aplikasi, termasuk masalah bug, kecepatan akses, dan kendala akun. (kata kunci: loading lama, error, bug, force close, keluar sendiri, login gagal, kode OTP)
4	Fitur & Fungsionalitas	Ulasan mengenai ketersediaan dan keberfungsian fitur-fitur layanan kelistrikan yang disediakan.(kata kunci: tambah daya, pasang baru, catat meter, swacam, fitur lengkap, menu pengaduan)
5	Antarmuka (UI/UX)	Ulasan yang membahas aspek visual, tata letak menu, kemudahan penggunaan, dan desain aplikasi.(kata kunci: tampilan, desain, menu membingungkan, ribet, update tampilan, user interface)

3.3.2 Data Filtering

Setelah proses pengumpulan data selesai, dilakukan tahap penyaringan data awal sebelum masuk ke proses pra-pemrosesan (*preprocessing*). Tahap ini bertujuan untuk meningkatkan kualitas *dataset* dengan cara:

1. Menghapus ulasan duplikat (*duplicate removal*) untuk menghindari bias frekuensi kata.
2. Menghapus ulasan kosong (*null values*).
3. Menghapus ulasan dengan panjang teks sangat pendek (misalnya hanya 1–2 kata) yang tidak memiliki informasi sentimen yang memadai untuk diklasifikasikan.

3.4 Text Preprocessing

Data ulasan mentah yang diperoleh dari tahap pengumpulan data masih memiliki struktur yang tidak beraturan serta mengandung berbagai *noise*. Oleh karena itu, tahap pra-pemrosesan teks dilakukan untuk mengubah data menjadi

format yang bersih, konsisten, dan terstandarisasi sehingga siap digunakan dalam proses pemodelan pembelajaran mesin.

Implementasi pra-pemrosesan dalam penelitian ini dilakukan menggunakan bahasa pemrograman Python dengan bantuan beberapa pustaka utama, yaitu Pandas untuk manipulasi data, Regular Expression (Regex) untuk pembersihan pola teks, serta Sastrawi untuk pemrosesan bahasa Indonesia. Tahapan pra-pemrosesan terdiri atas langkah-langkah sebagai berikut:

3.4.1 Case Folding

Tahap *case folding* bertujuan untuk mengubah seluruh huruf dalam dokumen ulasan menjadi huruf kecil (*lowercase*). Proses ini dilakukan untuk menyamakan bentuk kata sehingga tidak terjadi perbedaan makna akibat variasi penggunaan huruf kapital. Contoh: "PLN Mobile Sangat MEMBANTU"→"pln mobile sangat membantu"

3.4.2 Data Cleaning

Tahap *data cleaning* bertujuan menghapus elemen teks yang tidak relevan dan dapat mengganggu analisis sentimen. Proses ini dilakukan menggunakan modul Regular Expression (Regex) dengan menghapus elemen-elemen berikut:

1. Angka (0–9).
2. Tanda baca (*punctuation*) dan simbol khusus (@, #, $, dll).
3. *Emoticon* dan karakter non-ASCII.
4. Tautan (*URL*) dan *whitespace* berlebih (spasi ganda).

3.4.3 Word Normalization

Ulasan pengguna pada platform *mobile* umumnya menggunakan bahasa tidak baku, singkatan, maupun bahasa gaul (*slang*). Oleh karena itu, tahap normalisasi dilakukan untuk menyeragamkan kata-kata tersebut menjadi bentuk baku sesuai kaidah bahasa Indonesia. Proses normalisasi dilakukan dengan mencocokkan setiap kata dalam ulasan dengan kamus normalisasi (*key_norm.csv*) yang telah disusun sebelumnya. Contoh: "gk"→"tidak", "bgt"→"banget", "yg"→"yang".

3.4.4 Tokenization

Setelah teks dibersihkan dan dinormalisasi, dilakukan proses tokenisasi untuk memecah kalimat menjadi unit kata yang berdiri sendiri (*tokens*).

Tokenisasi dilakukan menggunakan metode pemisahan karakter (*splitting*) berdasarkan spasi. Contoh: "aplikasi ini sangat bagus" → ['aplikasi', 'ini', 'sangat', 'bagus']

3.4.5 Stopword Removal

Tahap ini bertujuan menghapus kata-kata umum yang sering muncul namun tidak memiliki kontribusi signifikan terhadap makna sentimen (*stop words*), seperti kata hubung dan kata depan. Proses ini menggunakan daftar *stopword* bahasa Indonesia dari pustaka Sastrawi, yang kemudian dapat diperluas dengan *stopword* tambahan yang spesifik sesuai konteks aplikasi. Contoh: "dan", "atau", "yang", "di", "ke", "dari".

3.4.6 Stemming

Tahap terakhir dari pra-pemrosesan adalah *stemming*, yaitu proses mengubah kata berimbuhan menjadi kata dasar (*root word*) dengan menghilangkan awalan, akhiran, sisipan, dan konfiks. Proses ini dilakukan menggunakan algoritma Nazief & Adriani yang diimplementasikan dalam pustaka Sastrawi. Contoh: "membayar", "pembayaran", "dibayarkan" →"bayar".

3.5 Pembagian Data

Setelah melalui tahap pra-pemrosesan, *dataset* dibagi menjadi dua bagian utama, yaitu data latih (*training set*) dan data uji (*testing set*). Pembagian ini bertujuan untuk mengukur kinerja model secara objektif terhadap data baru yang belum pernah dilihat sebelumnya selama proses pelatihan.

Proses pembagian data dilakukan menggunakan modul train_test_split dari pustaka Scikit-Learn. Dalam penelitian ini, proporsi pembagian data ditetapkan sebesar 80:20, di mana:

1. 80% Data Latih: Digunakan untuk melatih model klasifikasi agar dapat mempelajari pola kata dan bobot sentimen.
2. 20% Data Uji: Digunakan untuk mengevaluasi performa model dan menghitung metrik akurasi akhir.

Selain itu, pembagian data menerapkan parameter stratify berdasarkan label kelas sentimen. Hal ini dilakukan untuk memastikan distribusi kelas (Positif, Negatif, Netral) pada data latih dan data uji tetap proporsional dan seimbang. Parameter random_state juga ditentukan untuk menjaga konsistensi hasil

eksperimen (*reproducibility*) jika kode dijalankan ulang. Secara keseluruhan, pembagian data dilakukan secara ketat setelah tahap pra-pemrosesan namun sebelum ekstraksi fitur untuk memastikan tidak terjadi kebocoran informasi (*data leakage*) antar *dataset* selama proses pelatihan model.

3.6 Ekstraksi Fitur

Dikarenakan algoritma pembelajaran mesin hanya dapat memproses data numerik, data teks hasil pra-pemrosesan harus dikonversi menjadi representasi vektor. Metode ekstraksi fitur yang digunakan adalah *Term Frequency-Inverse Document Frequency* (TF-IDF) dengan memanfaatkan modul TfidfVectorizer dari pustaka Scikit-Learn.

Parameter konfigurasi TF-IDF yang digunakan meliputi min_df (mengabaikan kata yang muncul sangat jarang) dan max_df (mengabaikan kata yang muncul terlalu sering) untuk menghasilkan fitur yang informatif dan mengurangi dimensi matriks. Selain parameter dasar tersebut, proses ekstraksi fitur ini juga menerapkan teknik N-Gram dengan kombinasi Unigram, Bigram, dan Trigram (ngram_range=(1, 3)). Penggunaan rentang N-Gram ini bertujuan untuk menangkap konteks frasa kata majemuk dan negasi yang sering muncul dalam ulasan, seperti "tidak puas", "gagal bayar", atau "respon sangat lambat", yang maknanya dapat hilang jika hanya diproses sebagai kata tunggal.

Penerapan TF-IDF dilakukan dengan strategi khusus untuk menghindari kebocoran data (*data leakage*), yaitu:

1. Fit dan Transform pada Data Latih: Fungsi fit_transform() diterapkan hanya pada data latih. Pada tahap ini, model mempelajari kosakata (*vocabulary*) dan menghitung bobot IDF hanya dari korpus latih.
2. Transform pada Data Uji: Fungsi transform() diterapkan pada data uji menggunakan kosakata dan bobot IDF yang telah dipelajari dari data latih. Hal ini memastikan bahwa data uji benar-benar diperlakukan sebagai data "asing" (*unseen data*).

Parameter konfigurasi TF-IDF yang digunakan meliputi min_df (mengabaikan kata yang muncul sangat jarang) dan max_df (mengabaikan kata yang muncul terlalu sering) untuk menghasilkan fitur yang informatif dan mengurangi dimensi matriks. Representasi vektor hasil TF-IDF ini kemudian

digunakan sebagai *input* utama pada proses pelatihan model klasifikasi *ensemble* pada tahap selanjutnya.

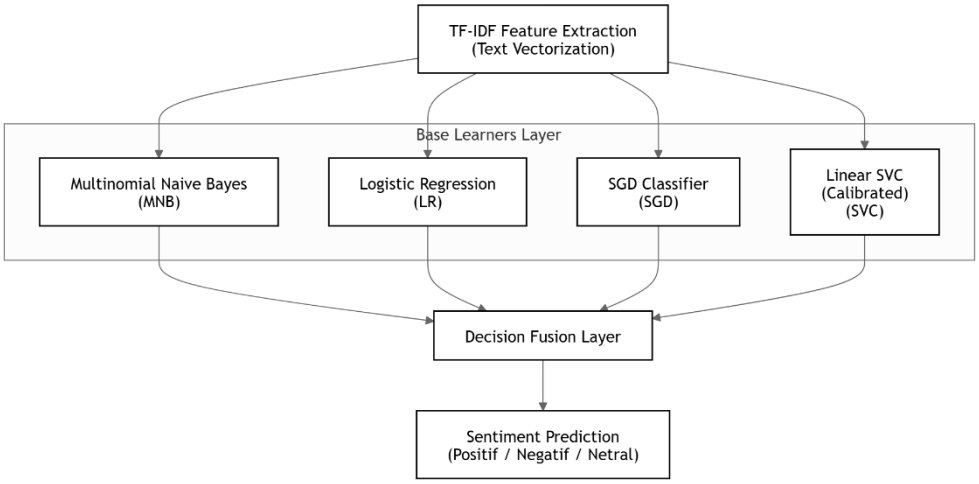
3.6.1 Penyeimbangan Data

Setelah teks diekstraksi menjadi fitur numerik (vektor) menggunakan TF-IDF, distribusi kelas sentimen pada data ulasan umumnya masih tidak seimbang. Jika kondisi ini langsung diproses, model klasifikasi linear (seperti LR dan SVC) akan cenderung bias terhadap kelas mayoritas. Oleh karena itu, matriks data latih perlu diseimbangkan menggunakan algoritma *Synthetic Minority Over-sampling Technique* (SMOTE).

Algoritma SMOTE bekerja dengan mencari *k* tetangga terdekat (*k-nearest neighbors*) dari setiap sampel kelas minoritas di dalam ruang vektor, kemudian menciptakan titik-titik data sintetis baru di sepanjang garis yang menghubungkan sampel tersebut dengan tetangganya. Penting untuk ditegaskan bahwa implementasi SMOTE ini hanya dilakukan pada Data Latih (80%). Data Uji (20%) sama sekali tidak dilibatkan dalam proses *oversampling* ini guna memastikan bahwa evaluasi performa model melalui metrik *F1-Score* nantinya tetap valid dan objektif mencerminkan kondisi data di dunia nyata.

3.7 Pembangunan Model Klasifikasi

Tahapan ini merupakan inti dari penelitian, di mana algoritma pembelajaran mesin dilatih untuk mengenali pola sentimen. Pembangunan model dilakukan secara bertahap, dimulai dari konfigurasi model dasar, penentuan bobot melalui validasi silang (*cross-validation*), pelatihan ulang model (*retraining*), hingga penyusunan arsitektur *ensemble*.



Gambar 3. 2 Arsitektur Model Weighted Soft Voting

Gambar 3-2 menunjukkan arsitektur model klasifikasi sentimen yang diusulkan. Fitur hasil ekstraksi TF-IDF diproses secara paralel oleh empat model dasar, yaitu Multinomial Naive Bayes, Logistic Regression, SGD Classifier, dan Linear SVC. Probabilitas keluaran dari masing-masing model kemudian digabungkan menggunakan metode Weighted Soft Voting untuk menghasilkan prediksi sentimen akhir berupa positif, negatif, atau netral.

3.7.1 Konfigurasi Model Dasar

Penelitian ini menggunakan empat algoritma klasifikasi, yaitu Multinomial Naive Bayes, Logistic Regression, SGD Classifier, dan Linear SVC. Keempat algoritma ini dipilih karena termasuk dalam kategori model linear yang terbukti efektif dan efisien dalam menangani data teks berdimensi tinggi dan jarang (*sparse*) hasil ekstraksi TF-IDF.

Agar setiap model dapat bekerja optimal dan menghasilkan probabilitas yang valid untuk metode *Soft Voting*, dilakukan konfigurasi parameter sebagai berikut:

1. Multinomial Naive Bayes (MNB)

   Parameter: alpha (*smoothing parameter*).

   Konfigurasi: alpha = 0.5.

   Nilai ini digunakan untuk mengontrol proses *Laplace smoothing* agar perhitungan probabilitas tetap stabil ketika terdapat fitur yang jarang muncul dalam data latih, serta meningkatkan sensitivitas model terhadap distribusi kelas minoritas.

2. Logistic Regression (LR)

   Parameter: C, solver, class_weight, max_iter.

   Konfigurasi:

   - solver = 'lbfgs'
   - class_weight = 'balanced'
   - max_iter = 2000
   - C diperoleh melalui proses GridSearchCV dengan validasi silang 3-fold pada rentang nilai {0.5, 1.0, 2.0}, menggunakan metrik evaluasi *f1_weighted*.

   Nilai C terbaik dipilih berdasarkan rata-rata skor F1 tertinggi pada proses validasi silang. Penggunaan *class_weight='balanced'* bertujuan untuk menangani ketidakseimbangan distribusi kelas sentimen dalam *dataset*.

3. Stochastic Gradient Descent (SGD) Classifier

   Parameter: loss, penalty, alpha, random_state.

   Konfigurasi:

   - loss = 'modified_huber'
   - penalty = 'l2'
   - alpha = 0.0001
   - random_state = 42

   Pemilihan *loss='modified_huber'* memungkinkan model menghasilkan estimasi probabilitas yang lebih stabil untuk kebutuhan metode *Soft Voting*. Regularisasi L2 diterapkan untuk mengurangi risiko *overfitting* pada ruang fitur berdimensi tinggi.

4. Linear Support Vector Classifier (Linear SVC)

   Parameter: C, max_iter, random_state.

   Konfigurasi:

   - C = 0.5
   - max_iter = 3000
   - random_state = 42

   Karena Linear SVC tidak secara langsung menghasilkan probabilitas, model ini dibungkus menggunakan metode CalibratedClassifierCV dengan metode kalibrasi sigmoid. Proses kalibrasi ini memungkinkan model menghasilkan estimasi probabilitas yang terstandarisasi tanpa mengubah struktur *hyperplane* klasifikasi.

3.7.2 Penentuan Bobot Model dengan *Cross-validation*

Untuk menentukan bobot (w) masing-masing model dalam metode *Weighted Soft Voting*, penelitian ini menggunakan pendekatan berbasis data (*data-driven*) melalui teknik 5-Fold *Cross-validation* pada data latih.

Mekanisme validasi dilakukan dengan ketat untuk mencegah kebocoran data (*data leakage*). Proses ekstraksi fitur TF-IDF dilakukan di dalam *pipeline* validasi silang. Artinya, pada setiap iterasi *fold*, kosakata (*vocabulary*) hanya dipelajari dari data latih bagian tersebut (*fit*), lalu diterapkan pada data validasi (*transform*).

Proses penentuan bobot adalah sebagai berikut:

1. Data latih dibagi menjadi 5 bagian (*fold*).
2. Setiap model dasar dilatih dan divalidasi secara bergantian pada setiap *fold*.
3. Rata-rata nilai F1-Score dari 5 kali pengujian diambil sebagai bobot global (w) untuk model tersebut.

Model yang memiliki rata-rata F1-Score tinggi selama validasi akan mendapatkan bobot kontribusi yang lebih besar dalam keputusan akhir.

3.7.3 Penerapan Weighted Soft Voting Ensemble

Setelah bobot optimal diperoleh, tahap selanjutnya adalah pembangunan model akhir. Penting untuk dicatat bahwa seluruh model dasar dilatih ulang (*retrained*) menggunakan keseluruhan data latih (100% data *training*) sebelum diintegrasikan. Hal ini bertujuan agar model dapat mempelajari pola dari sebanyak mungkin data yang tersedia.

$$\hat{P}(y = i \mid x) = \frac{\sum_{j=1}^{m} w_j \,.\, P_j(y = i \mid x)}{\sum_{j=1}^{m} w_j}$$

Model-model yang telah terlatih penuh tersebut kemudian digabungkan ke dalam meta-klasifikator VotingClassifier dengan mekanisme *Soft Voting*. Prediksi kelas akhir (ŷ) untuk sebuah ulasan baru (x) dihitung berdasarkan jumlahan probabilitas terbobot dengan rumus:

Keterangan

- $\hat{P}(y = i \mid x)$= probabilitas akhir kelas ke-i setelah penggabungan
- w_j= bobot model ke-j
- $P_j(y = i \mid x)$= probabilitas kelas ke-i yang diprediksi model ke-j

- m= jumlah model dalam ensemble
- x= data masukan
- $y = i$= kelas ke-i

Kelas dengan nilai probabilitas gabungan tertinggi dipilih sebagai label prediksi final. Model *ensemble* final yang telah terbentuk kemudian digunakan pada tahap evaluasi menggunakan data uji yang sebelumnya tidak terlibat dalam proses pelatihan maupun validasi.

3.8 Evaluasi Model

Tahap evaluasi bertujuan untuk mengukur kinerja generalisasi model *Weighted Soft Voting Ensemble* terhadap data baru. Evaluasi dilakukan menggunakan Data Uji (20%) yang telah dipisahkan di awal penelitian.

Proses evaluasi dilakukan dengan membandingkan label prediksi yang dihasilkan oleh model (ŷ) dengan label sebenarnya (*ground truth*) pada data uji (y). Hasil perbandingan tersebut dipetakan ke dalam tabel *Confusion Matrix* untuk menghitung empat metrik utama:

1. *Accuracy:* Mengukur persentase total prediksi yang benar.
2. *Precision:* Mengukur ketepatan prediksi positif (relevan untuk meminimalkan *False Positive*).
3. *Recall:* Mengukur kemampuan model menemukan kembali data positif (relevan untuk meminimalkan *False Negative*).
4. *F1-Score:* Rata-rata harmonik antara *Precision* dan *Recall*.

Dalam tahap pengujian, seluruh algoritma base learners (MNB, LR, SGD, dan Linear SVC) dievaluasi kinerjanya secara mandiri sebagai model tunggal. Hasil dari model-model tunggal ini kemudian dibandingkan dengan hasil Weighted Soft Voting Ensemble untuk membuktikan efektivitas metode penggabungan yang dilakukan.

3.9 Implementasi Sistem

Tahap terakhir dari metodologi penelitian ini adalah implementasi model ke dalam sebuah antarmuka aplikasi berbasis web. Implementasi ini bertujuan untuk membuktikan bahwa model yang telah dilatih dapat bekerja secara fungsional untuk mengklasifikasikan ulasan baru secara *real-time*.

3.9.1 Lingkungan Pengembangan

Sistem dibangun menggunakan bahasa pemrograman Python dengan memanfaatkan kerangka kerja Streamlit. Untuk menjaga efisiensi komputasi dan konsistensi transformasi data, arsitektur sistem memuat dua komponen utama:

1. Modul Pra-pemrosesan: Sebuah fungsi Python terpisah yang memuat logika pembersihan, normalisasi (menggunakan kamus), dan *stemming*. Pemisahan ini dilakukan agar proses berat seperti *stemming* hanya dieksekusi satu kali pada input mentah.
2. Pipeline Model: Objek serialisasi (.pkl atau .joblib) dari pustaka Scikit-Learn yang berisi tahapan ekstraksi fitur (TF-IDF) dan model klasifikasi (Weighted Soft Voting Ensemble). Penggunaan *pipeline* memastikan bahwa kosakata (*vocabulary*) TF-IDF yang digunakan pada sistem sama persis dengan yang dipelajari saat pelatihan.

3.9.2 Alur Kerja Sistem Aplikasi

Alur kerja aplikasi dirancang sederhana untuk memudahkan pengguna. Proses yang terjadi di dalam sistem adalah sebagai berikut:

1. Input: Pengguna memasukkan teks ulasan mentah pada kolom yang tersedia di antarmuka Streamlit.
2. Preprocessing: Sistem memanggil fungsi pra-pemrosesan untuk mengubah teks mentah menjadi teks bersih (*clean text*).
3. Prediction: Teks bersih dimasukkan ke dalam Pipeline Model. Secara otomatis, sistem akan melakukan transformasi vektor TF-IDF dan perhitungan probabilitas sentimen.
4. Output: Sistem menampilkan hasil prediksi kelas sentimen (Positif, Negatif, atau Netral) beserta skor keyakinan (*confidence score*) dalam bentuk visualisasi grafik.

3.10 Alat dan Bahan

Pelaksanaan penelitian ini memerlukan serangkaian alat dan bahan yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Alat dan bahan ini digunakan untuk mendukung seluruh tahapan penelitian, mulai dari proses pengumpulan data, prapemrosesan, pelatihan model, hingga implementasi purwarupa sistem.

3.10.1 Perangkat Keras

Perangkat keras yang digunakan dalam penelitian ini berupa satu unit komputer (laptop) yang berfungsi sebagai stasiun kerja utama untuk mengeksekusi skrip pemrograman dan melatih algoritma pembelajaran mesin. Rincian spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 3.4.

Tabel 3. 3 Perangkat Keras

Komponen	Spesifikasi	Unit
Komputer Pengembangan	Ryzen 7 6000 Series	1 unit
RAM	16 GB	1 unit
Storage	SSD 512 GB	1 unit

3.10.2 Perangkat Lunak

Selain perangkat keras, penelitian ini juga membutuhkan dukungan perangkat lunak berupa sistem operasi, bahasa pemrograman, lingkungan pengembangan, serta berbagai pustaka (*library*) khusus untuk pemrosesan teks dan *machine learning*. Rincian perangkat lunak beserta versinya disajikan pada Tabel 3.5.

Tabel 3. 4 Perangkat Lunak

Kategori	Tools/Library	Versi
Sistem Operasi	Windows 11	-
IDE	Visual Studio Code	Latest
Programming Language	Python	3.9+
Data Acquisition	google-play-scraper	Latest Stable
Machine Learning	Scikit-learn, imbalanced-learn	Latest Stable
Data Processing	Pandas, NumPy	Latest Stable
Text Preprocessing	Sastrawi,NLTK	Latest
Web Framework	Streamlit	Latest
Visualisasi	Matplotlib, Seaborn	Latest
Model Serialization	pickle / joblib	Built-in / Latest

BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Pengumpulan Data

Tahap awal dari penelitian ini adalah melakukan pengumpulan data teks berupa ulasan pengguna terhadap aplikasi PLN Mobile. Data diperoleh menggunakan teknik *web scraping* melalui pustaka (library) google-play-scraper pada bahasa pemrograman Python, yang menarik data langsung dari halaman aplikasi PLN Mobile di platform Google Play Store.

Proses penarikan data difokuskan untuk mendapatkan dataset ulasan yang representatif. Karena batasan dari platform Play Store, data ditarik secara berulang dan digabungkan hingga mencapai jumlah yang memadai untuk pelatihan model pembelajaran mesin (*machine learning*).

Setelah dilakukan penarikan data dan pembersihan tahap awal (penghapusan data ganda atau *duplicate content*), diperoleh total dataset sebanyak 23.850 baris ulasan valid. Distribusi kelas sentimen pada dataset awal (sebelum dilakukan penyeimbangan) ditunjukkan pada Tabel 4.1 berikut:

Tabel 4. 1 Distribusi Data Ulasan PLN Mobile (Dataset Asli)

Kelas Sentimen	Jumlah Data	Persentase
Positif	18.707	78,4 %
Negatif	4.682	19,6 %
Netral	461	1,9 %
Total Data	23.850	100 %

Berdasarkan Tabel 4.1, dapat dilihat dengan jelas bahwa dataset ulasan PLN Mobile mengalami ketidakseimbangan kelas (*class imbalance*) yang sangat ekstrem. Ulasan bersentimen Positif sangat mendominasi dengan jumlah 18.707 data (78,4%), disusul oleh ulasan Negatif sebanyak 4.682 data (19,6%). Sementara itu, ulasan Netral merupakan kelompok minoritas dengan jumlah yang sangat sedikit, yaitu hanya 461 data (1,9%).

Ketimpangan data ini menjadi tantangan utama karena algoritma *machine learning* konvensional akan cenderung bias dan hanya mengenali kelas mayoritas (Positif) jika dilatih menggunakan data mentah ini. Oleh karena itu, penerapan

teknik penyeimbangan data sintetis (SMOTE) sangat diperlukan pada tahap selanjutnya untuk memastikan model tidak condong ke satu sisi dan mampu memprediksi kelas minoritas dengan akurat.

Selain pelabelan kelas sentimen, penelitian ini juga melakukan pemetaan ulasan ke dalam kategori aspek layanan menggunakan pendekatan berbasis leksikon (*keyword matching*). Setiap ulasan dipindai menggunakan daftar kata kunci yang telah ditetapkan untuk memetakannya ke dalam 5 (lima) kategori aspek utama. Karena menggunakan pendekatan *multi-label*, satu ulasan dapat dikategorikan ke dalam lebih dari satu aspek jika mengandung kata kunci yang relevan. Perlu ditegaskan bahwa pemetaan aspek ini murni dilakukan pada keseluruhan dataset asli yang berjumlah 23.850 ulasan. Karena satu ulasan dapat dikategorikan ke dalam lebih dari satu aspek sekaligus (*multi-label*), maka akumulasi kemunculan aspek mencapai 48.376 dan total persentase pada Tabel 4.2 melampaui angka 100%. Hasil distribusi ekstraksi aspek layanan dapat dilihat pada Tabel 4.2 berikut.

Tabel 4. 2 Distribusi Kategori Aspek Layanan

Kategori Aspek Layanan	Kata Kunci Relevan (Keywords)	Jumlah Kemunculan	Persentase
Pembayaran & Token	bayar, token, transaksi, saldo, tagihan, struk, e-wallet	12.520	52.49%
Pelayanan Pelanggan	cs, komplain, teknisi, petugas, aduan, gangguan, lambat	12.350	51.78%
Fitur& Fungsionalitas	cek tagihan, riwayat, lengkap, tambah daya, id pelanggan	10.469	43.90%
Performa Aplikasi	lambat, lemot, lag, error, crash, login, koneksi	9.627	40.36%
Antarmuka UI/UX	tampilan, desain, menu, navigasi, ribet, bingung, font	3.410	14.30%

Berdasarkan Tabel 4.2, dapat ditarik analisis bahwa masalah yang berhubungan dengan finansial pengguna—yaitu Pembayaran & Token—menjadi aspek krusial yang paling sering dibahas (muncul pada 52.49% total keseluruhan ulasan). Hal ini sangat masuk akal mengingat fungsi utama PLN Mobile bagi pengguna harian adalah sebagai media transaksi energi listrik. Di urutan kedua, aspek Pelayanan Pelanggan (51.78%) juga mendominasi, menunjukkan tingginya

urgensi pengguna dalam melaporkan gangguan atau menuntut respons cepat dari pihak CS dan teknisi lapangan.

Sebaliknya, aspek Antarmuka (UI/UX) adalah metrik yang paling jarang disinggung secara spesifik (hanya 14.30%). Hal ini mengindikasikan bahwa masalah tata letak visual atau desain aplikasi bukanlah prioritas keluhan utama pelanggan, melainkan fungsionalitas dan kendala teknis *backend* (pembayaran, performa, dan tindak lanjut *customer service*).

4.2 Hasil Pra-pemrosesan Teks

Data ulasan yang diperoleh dari Play Store umumnya bersifat tidak terstruktur, mengandung banyak kesalahan ketik (*typo*), singkatan (bahasa *slang*), serta karakter yang tidak relevan. Oleh karena itu, tahap *text preprocessing* mutlak diperlukan untuk membersihkan dan menstandarkan teks sebelum diproses oleh model klasifikasi.

Tahapan *preprocessing* yang diimplementasikan pada sistem ini meliputi:

4.2.1 Case Folding dan Data Cleansing

Tahap pertama dalam prapemrosesan teks adalah *Case Folding* dan *Data Cleansing*. *Case Folding* merupakan proses penyeragaman seluruh karakter huruf di dalam teks ulasan menjadi huruf kecil (*lowercase*). Proses ini sangat penting karena bahasa pemrograman pada algoritma *machine learning* bersifat *case-sensitive*, yang berarti sistem akan menganggap kata "PLN", "Pln", dan "pln" sebagai tiga entitas yang sepenuhnya berbeda jika tidak diseragamkan. Dengan *case folding*, kompleksitas ruang fitur dapat ditekan secara signifikan.

Setelah seluruh huruf diseragamkan, proses dilanjutkan dengan *Data Cleansing* (Pembersihan Data). Teks ulasan yang diambil dari Google Play Store sangat rentan terhadap derau (*noise*), seperti adanya tautan (URL), angka, simbol, tanda baca yang berlebihan, hingga *emoticon* atau *emoji*. Mengingat penelitian ini berfokus pada analisis sentimen berbasis leksikal dan frekuensi kata, elemen-elemen *noise* tersebut tidak memberikan bobot informasi yang relevan bagi algoritma. Oleh karena itu, sistem diinstruksikan untuk menghapus seluruh karakter non-alfabet menggunakan modul *Regular Expression (RegEx)* pada Python, sehingga hanya menyisakan rentang karakter alfabet (a-z) dan spasi (whitespace).

```
def clean_text(text):
     if pd.isna(text):
         return ""

     # 1. Case Folding (Mengubah semua huruf menjadi huruf kecil)
     text = str(text).lower()

     # 2. Cleansing
     text = re.sub(r'http\S+|www\S+|https\S+', '', text)
     text = re.sub(r'@\w+|#\w+', '', text)
     text = re.sub(r'[^\x00-\x7F]+', ' ', text)
     text = re.sub(r'[0-9]+', '', text)

     # Hapus Tanda Baca
     text = text.translate(str.maketrans('', '', string.punctuation))

    return text
```

Gambar 4. 1 Implementasi Kode Data Cleansing dan Case Folding pada Python

Sebagai gambaran, efek dari proses *Case Folding* dan *Data Cleansing* ditunjukkan pada Tabel 4.3 berikut.

Tabel 4. 3 Contoh Hasil Case Folding dan Data Cleansing

Teks Asli (Raw Text)	Hasil Cleansing
"Sangat MEMBANTU!! 👍 Bayar listrik jam 23:00 sukses terus."	"sangat membantu bayar listrik jam sukses terus"
"Apk error trus 😭😭 ... tolong di fix https://pln.co.id"	"apk error trus tolong di fix"

Melalui proses ini, teks menjadi jauh lebih bersih dan siap untuk diproses ke tahap selanjutnya, yaitu normalisasi kata yang tidak baku.

4.2.2 Normalisasi Kata Tidak Baku

Karakteristik utama dari data teks yang bersumber dari media sosial atau *platform* ulasan publik seperti Google Play Store adalah penggunaan bahasa yang sangat informal. Pengguna sering kali menulis ulasan menggunakan bahasa gaul (*slang*), singkatan, hingga modifikasi kata yang tidak sesuai dengan Ejaan Yang Disempurnakan (EYD) atau Kamus Besar Bahasa Indonesia (KBBI). Ketidakkonsistenan penulisan ini akan memicu fenomena *Sparsity* (kelangkaan data) pada ruang fitur TF-IDF, di mana kata-kata dengan makna yang persis sama (misalnya: "tdk", "gak", "gk", "g") akan dianggap sebagai fitur (dimensi) yang berbeda-beda oleh mesin.

Untuk mengatasi hal tersebut, proses *Slang Normalization* (Normalisasi Kata Tidak Baku) diterapkan. Sistem menggunakan kamus pemetaan (*Slang Dictionary*) khusus yang berisi ratusan kata tidak baku beserta padanan kata bakunya. Saat sistem membaca teks ulasan, setiap kata akan dicocokkan dengan kunci (*key*) di dalam *dictionary*. Jika ditemukan kecocokan, maka kata tersebut akan otomatis diganti (*replace*) menjadi bentuk bakunya.

```
SLANG_DICT = {
    # ===== NEGASI =====
    'gak': 'tidak',
    'ga': 'tidak',
    'gk': 'tidak',
    'g': 'tidak',
    'tdk': 'tidak',
    'gx': 'tidak',
    'kagak': 'tidak',
    'kaga': 'tidak',
    'engga': 'tidak',
    'enggak': 'tidak',
    'ngga': 'tidak',
    'nggak': 'tidak',
    'gada': 'tidak ada',
    'gaada': 'tidak ada',
    'gamau': 'tidak mau',
    'gapapa': 'tidak apa-apa',
    'gabisa': 'tidak bisa',
    'gatau': 'tidak tahu',
    'gapaham': 'tidak paham',
    # ... (dan ratusan kata gaul lainnya)
}
```

Gambar 4. 2 Implementasi Kamus Normalisasi Kata Tidak Baku

Penggunaan kamus normalisasi ini terbukti mampu mengonsolidasikan makna dan mengurangi dimensi fitur secara drastis, sehingga membantu model klasifikasi dalam memusatkan perhatian pada kata-kata sentimen yang sebenarnya. Contoh pemetaan normalisasi kata ditunjukkan pada Tabel 4.4 berikut.

Tabel 4. 4 Contoh Pemetaan Normalisasi Bahasa Gaul

Kata Asli dalam Ulasan	Bentuk Baku (Hasil Normalisasi)
"gak", "gk", "tdk", "g"	"tidak"
"mantap", "keren", "topcer"	"bagus"
"lemot", "lola", "slow"	"lambat"
"apk", "app"	"aplikasi"
"update", "apdet"	"pembaruan"

Dengan diseragamkannya kosakata ke dalam bentuk baku, tahap selanjutnya dapat dilakukan dengan lebih akurat.

4.2.3 Penanganan Negasi

Salah satu tantangan terbesar dalam klasifikasi teks (terutama setelah tahap penghapusan *stopword*) adalah penanganan kata negasi atau sangkalan. Kata-kata seperti "tidak", "bukan", atau "jangan" sering kali termasuk ke dalam daftar kata umum (*stopword*) yang dihapus oleh sistem tradisional. Padahal, kata negasi memiliki fungsi krusial untuk membalikkan polaritas sentimen kalimat. Sebagai contoh, frasa "tidak bagus" akan kehilangan makna aslinya dan malah diprediksi sebagai sentimen positif (karena hanya tersisa kata "bagus") apabila kata "tidak" dihapus secara sembarangan.

Untuk mengatasi permasalahan fatal ini, penelitian ini membangun fungsi khusus bernama handle_negation. Fungsi ini bekerja dengan memindai setiap kata di dalam ulasan. Jika sistem menemukan kata yang masuk ke dalam daftar negasi (misalnya "gak", "tidak", "belum"), sistem akan langsung mengecek satu kata berikutnya (*next word*). Apabila kata berikutnya adalah kata sifat positif, sistem akan melebur kedua kata tersebut dan menukarnya dengan satu padanan kata negatif yang sesuai. Misalnya, "tidak" + "bagus" akan langsung diubah bentuknya menjadi kata tunggal "buruk". Sebaliknya, "tidak" + "susah" akan diubah menjadi "mudah".

```
    words = text.split()
    result = []
    skip_next = False

    for i, word in enumerate(words):
        if skip_next:
            skip_next = False
            continue

        if word in _NEGATION_WORDS and i + 1 < len(words):
            next_word = words[i + 1]

            if next_word in _POSITIVE_TO_NEGATIVE:
                result.append(_POSITIVE_TO_NEGATIVE[next_word])
                skip_next = True  # Lewati kata positif asli

            elif next_word in _NEGATIVE_TO_POSITIVE:
                result.append(_NEGATIVE_TO_POSITIVE[next_word])
                skip_next = True
            else:

                result.append('tidak')
        else:
            result.append(word)

    return ' '.join(result)
```

Gambar 4. 3 Implementasi Kode Penanganan Negasi

Transformasi dinamis ini sangat krusial karena model dilatih menggunakan ekstraksi fitur TF-IDF berbasis *unigram* (kata tunggal). Dengan mengubah frasa negasi menjadi satu kata dasar polaritas baru, model tidak akan terkecoh. Ilustrasi pemetaan pada penanganan negasi ditunjukkan pada Tabel 4.5.

Tabel 4. 5 Contoh Perubahan pada Penanganan Negasi

Teks Asli	Bentuk Pemetaan oleh Sistem	Polaritas Akhir
"Aplikasi ini tidak bagus"	"aplikasi ini buruk"	Negatif
"Kenapa gak bisa login?"	"kenapa gagal login"	Negatif
"Sistemnya kurang ramah"	"sistemnya kasar"	Negatif
"Ternyata tidak susah bayarnya"	"ternyata mudah bayarnya"	Positif

Berkat mekanisme penanganan negasi ini, akurasi mesin dalam membaca konteks kalimat bahasa Indonesia dapat meningkat secara signifikan.

4.2.4 Penghapusan Stopwords

Tahap *stopwords removal* bertujuan untuk menghilangkan kata-kata yang memiliki frekuensi kemunculan tinggi dalam hampir semua dokumen, namun tidak membawa informasi sentimen yang signifikan. Keberadaan kata-kata ini dalam representasi fitur TF-IDF justru akan menambah dimensi matriks fitur secara tidak perlu (*noise*) dan berpotensi menurunkan daya diskriminatif model klasifikasi.

Dalam implementasinya, sistem ini menggunakan daftar *stopwords* kustom yang dirancang khusus untuk domain ulasan aplikasi berbahasa Indonesia. Daftar tersebut disimpan dalam variabel STOPWORDS_EXTENDED pada modul slang_dict.py dan mencakup lebih dari 100 kata yang dikelompokkan ke dalam beberapa kategori berikut:

1. Kata hubung dan partikel umum: seperti "yang", "dan", "atau", "di", "ke", "dari", "untuk", "dengan", "pada", "adalah", serta partikel informal ("sih", "nih", "dong", "deh", "loh", "aja", "kan").
2. Kata ganti orang: seperti "saya", "anda", "kita", "kami", "mereka", "gw", "gue", "aku", "lo", "lu", "kamu".
3. Kata keterangan waktu: seperti "hari", "bulan", "tahun", "kemarin", "besok", "sekarang", "nanti", "minggu".
4. Angka dalam kata: seperti "satu", "dua", "tiga", "pertama", "kedua", dan seterusnya.
5. Nama *brand* dan platform (bersifat netral): seperti "pln", "mobile", "aplikasi", "playstore", "google", "android", "hp".

Selain penghapusan berdasarkan daftar, sistem juga menerapkan filter panjang kata minimal, yaitu hanya menyisakan kata yang memiliki panjang lebih dari dua karakter (len(w) > 2). Hal ini bertujuan untuk mengeliminasi kata-kata sangat pendek yang umumnya tidak memiliki makna sentimen, seperti "di", "ke", "ya", dan "ok".

Satu keputusan desain yang penting dalam tahap ini adalah tidak memasukkan kata-kata negasi ke dalam daftar *stopwords*. Kata-kata seperti "tidak", "bukan", "belum", dan "jangan" sengaja dipertahankan karena memiliki fungsi krusial dalam menentukan polaritas sentimen, sebagaimana telah dibahas pada sub-bab

4.2.3. Jika kata negasi ikut dihapus, frasa "tidak bagus" hanya akan menyisakan kata "bagus" yang bermakna sebaliknya.

```
def remove_stopwords(text):
    """Menghapus kata-kata tidak bermakna (stopwords) dari teks."""
    words = text.split()

    words = [w for w in words if w not in STOPWORDS_EXTENDED and len(w) > 2]

    return ' '.join(words)
```

Gambar 4. 4 Implementasi Kamus Stopword pada Tahap Filtering

Proses *stopwords removal* diimplementasikan dalam fungsi remove_stopwords() pada modul preprocessing.py. Fungsi ini menerima teks yang telah dinormalisasi dan mengembalikan teks bersih yang hanya berisi kata-kata bermakna sentimen.

4.2.5 Stemming

Tahap terakhir dari rangkaian *preprocessing* adalah *stemming*, yaitu proses morfologi untuk mengubah kata berimbuhan menjadi bentuk kata dasarnya (*root word*). Tujuan utama dari tahap ini adalah mereduksi variasi morfologis kata yang memiliki makna semantik sama, sehingga dimensi ruang fitur (*feature space*) pada representasi TF-IDF dapat dikurangi secara signifikan tanpa kehilangan informasi sentimen.

Dalam penelitian ini, proses *stemming* diimplementasikan menggunakan pustaka Sastrawi (*Stemmer Factory*), yang merupakan pustaka *open-source* berbasis Python yang dirancang khusus untuk menangani kompleksitas morfologi bahasa Indonesia. Sastrawi mengadopsi algoritma *Enhanced Confix Stripping* (ECS), yaitu penyempurnaan dari metode *Confix Stripping* yang dikembangkan oleh Nazief dan Adriani. Algoritma ini bekerja melalui aturan pemenggalan imbuhan bertingkat yang mencakup penghilangan awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*), dan konfiks secara sekuensial.

Sebagai contoh, berbagai variasi kata berikut akan dipetakan menjadi satu kata dasar yang sama:

Tabel 4. 6 Contoh Hasil Proses Stemming Menggunakan Library Sastrawi

Kata Berimbuhan	Kata Dasar (Hasil Stemming)
"membayar", "pembayaran", "dibayar", "terbayar"	"bayar"
"membantu", "dibantu", "bantuan", "terbantu"	"bantu"
"menggunakan", "digunakan", "penggunaan", "kegunaan"	"guna"
"perbaikan", "memperbaiki", "diperbaiki"	"baik"

Dengan pemetaan ini, model klasifikasi tidak perlu mempelajari setiap variasi morfologis secara terpisah, melainkan cukup mengenali satu representasi kata dasar yang merangkum seluruh bentuk turunannya. Hal ini secara langsung mengurangi *sparsity* (kelangkaan data) pada matriks TF-IDF dan meningkatkan efisiensi pelatihan model.

Dalam implementasinya pada sistem, objek *stemmer* diinisialisasi melalui kelas StemmerFactory() dari pustaka Sastrawi pada modul preprocessing.py. Proses *stemming* diterapkan pada teks yang telah melewati seluruh tahap sebelumnya (*case folding*, normalisasi *slang*, *negation handling*, dan *stopwords removal*) melalui pemanggilan fungsi stemmer.stem(text). Setelah proses *stemming*, dilakukan juga normalisasi spasi ganda (*double space*) yang mungkin muncul akibat penghapusan kata pada tahap-tahap sebelumnya, melalui operasi ' '.join(text_stemmed.split()).

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

def create_stemmer():
    """Inisialisasi algoritma Sastrawi Stemmer."""
    factory = StemmerFactory()
    return factory.create_stemmer()

def preprocess_text(text, stemmer):
    """Pipeline eksekusi lengkap tahapan preprocessing berurutan."""
    # 1. Cleansing & Case Folding
    text_clean = clean_text(text)
    # 2. Normalisasi Slang
    text_normalized = normalize_slang(text_clean)
    # 3. Penanganan Negasi
    text_negation = handle_negation(text_normalized)
    # 4. Penghapusan Stopwords
    text_no_stop = remove_stopwords(text_negation)
    # 5. Proses Stemming Sastrawi
    text_stemmed = stemmer.stem(text_no_stop)

    # Merapikan kelebihan spasi
    text_stemmed = ' '.join(text_stemmed.split())

    return text_stemmed
```

Gambar 4. 5 Implementasi Pipeline Preprocessing dan Stemming Sastrawi pada Sistem

4.3 Pembagian Data

Setelah seluruh tahapan pra-pemrosesan selesai dilakukan, langkah selanjutnya adalah membagi dataset menjadi dua bagian, yaitu data latih (*training set*) dan data uji (*testing set*). Pembagian ini bertujuan untuk mengevaluasi kemampuan generalisasi model terhadap data yang belum pernah dilihat selama proses pelatihan, sehingga hasil evaluasi mencerminkan performa model secara objektif.

Pembagian data dilakukan menggunakan fungsi train_test_split() dari pustaka Scikit-learn dengan rasio 80:20, yaitu 80% data dialokasikan untuk pelatihan model dan 20% sisanya untuk pengujian. Selain itu, parameter stratify diaktifkan untuk memastikan proporsi distribusi kelas sentimen (Positif, Negatif, Netral) tetap terjaga secara proporsional pada kedua subset data. Teknik ini dikenal sebagai *Stratified Splitting* dan sangat penting mengingat dataset memiliki ketidakseimbangan kelas yang signifikan sebagaimana telah diuraikan pada sub-bab 4.1.

Sebelum pembagian data, dilakukan juga penyaringan tambahan berupa penghapusan baris dengan hasil *preprocessing* yang terlalu pendek (kurang dari 5

karakter). Hal ini bertujuan untuk mengeliminasi data yang kehilangan seluruh informasi sentimen setelah proses pembersihan teks, Proses penyaringan ini membuang sebanyak 60 baris ulasan, sehingga total dataset efektif yang digunakan untuk pelatihan model berkurang dari 23.850 menjadi 23.790 baris.Hasil pembagian data ditunjukkan pada Tabel 4.7 berikut:

Tabel 4. 7 Pembagian Data

Subset Data	Jumlah Data	Persentase
Data Latih (*Training Set*)	19.032	80%
Data Uji (*Testing Set*)	4.758	20%
Total	23.790	100%

Parameter random_state=42 digunakan untuk menjamin reprodusibilitas hasil, sehingga pembagian data yang sama dapat diulang pada setiap eksperimen. Hal ini penting untuk memastikan konsistensi perbandingan performa antar model.

Perlu ditekankan bahwa proses penyeimbangan data menggunakan SMOTE pada tahap selanjutnya hanya diterapkan pada data latih, sedangkan data uji dibiarkan pada distribusi aslinya. Keputusan ini diambil untuk mencegah terjadinya kebocoran data (*data leakage*), yaitu kondisi di mana informasi dari data uji turut mempengaruhi proses pelatihan yang dapat menghasilkan evaluasi performa yang terlalu optimistis (*overly optimistic*).

4.4 Ekstraksi Fitur (TF-IDF)

Setelah data dibagi menjadi set pelatihan dan set pengujian, tahapan berikutnya adalah mengubah teks kualitatif hasil *preprocessing* menjadi representasi kuantitatif (*vektor numerik*) agar dapat diproses oleh algoritma *machine learning*. Proses ekstraksi fitur ini dilakukan menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*).

Dalam penelitian ini, pembobotan TF-IDF diimplementasikan menggunakan kelas TfidfVectorizer dari pustaka Scikit-learn. Untuk mengoptimalkan kualitas fitur yang dihasilkan dan mencegah terbentuknya matriks fitur yang terlalu besar dan *sparse* (langka), konfigurasi *vectorizer* disesuaikan dengan parameter (*hyperparameter*) lanjutan sebagai berikut:

1. ngram_range=(1, 3): Sistem tidak hanya mengekstrak kata tunggal (*unigram*), tetapi juga pasangan dua kata (*bigram*) dan tiga kata (*trigram*). Penggunaan *n-gram* sangat penting dalam analisis sentimen karena mampu menangkap konteks frasa, seperti "aplikasi bagus" atau "tidak bisa masuk".
2. max_features=15000: Membatasi ukuran dimensi matriks dengan hanya mengambil 15.000 fitur teratas yang memiliki frekuensi kemunculan (*term frequency*) tertinggi di seluruh korpus. Hal ini efektif mereduksi *noise* dari kata-kata yang sangat jarang muncul atau salah ketik yang luput dari tahap normalisasi.
3. min_df=2 dan max_df=0.90: Kata yang muncul pada kurang dari 2 dokumen (terlalu spesifik) atau lebih dari 90% dokumen (terlalu umum/mirip *stopwords*) diabaikan.
4. sublinear_tf=True: Menerapkan penskalaan logaritmik (1 + log(tf)) untuk menghaluskan pengaruh kemunculan kata yang berulang berkali-kali dalam satu dokumen ulasan.
5. norm='l2': Melakukan normalisasi *Euclidean* (L2) pada vektor fitur agar variasi panjang ulasan (ulasan pendek vs ulasan panjang) tidak memberikan bobot yang tidak seimbang pada model.

Proses ekstraksi fitur dilakukan secara terpisah untuk data latih dan data uji demi menghindari masalah kebocoran data (*data leakage*). Fungsi fit_transform() hanya diterapkan pada data latih (X_train) untuk membangun kosakata (*vocabulary*) dan menghitung IDF berdasarkan distribusi kata pada data latih saja. Selanjutnya, fungsi transform() diterapkan pada data uji (X_test) dengan berpatokan pada struktur kosakata dari data latih.

Hasil dari tahap ini adalah sebuah matriks *sparse* berdimensi tinggi berukuran 19.080 x 15.000 untuk data latih, dan 4.758 x 15.000 untuk data uji. Matriks inilah yang kemudian menjadi *input* (masukan) bagi proses penyeimbangan data menggunakan SMOTE serta pelatihan model *machine learning*.

4.5 Penyeimbangan Data

Sebagaimana telah diuraikan pada Tabel 4.1, dataset ulasan aplikasi PLN Mobile mengalami masalah ketidakseimbangan kelas (*class imbalance*) yang

sangat ekstrem. Ulasan bersentimen Positif sangat mendominasi, sementara ulasan Negatif dan terutama Netral memiliki jumlah sampel yang jauh lebih sedikit. Jika model *machine learning* dilatih langsung menggunakan data yang sangat condong (*skewed*) ini, algoritma akan cenderung mengalami bias (*overfitting*) terhadap kelas mayoritas dan mengabaikan kelas minoritas.

Akibatnya, model akan kesulitan mendeteksi ulasan sentimen negatif atau netral, meskipun nilai akurasi keseluruhan (*overall accuracy*) tampak tinggi.

Untuk mengatasi permasalahan tersebut, diterapkan metode penyeimbangan data pada tingkat algoritma (data-level) menggunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*). SMOTE bekerja bukan dengan sekadar menduplikasi data minoritas yang sudah ada (*random oversampling*), melainkan dengan membangkitkan (meng- *generate*) sampel data sintetis baru yang memiliki karakteristik serupa dengan data asli pada kelas minoritas tersebut.

Dalam implementasinya menggunakan pustaka imblearn pada Python, algoritma SMOTE mencari k tetangga terdekat (*k-Nearest Neighbors*) dari setiap titik data pada kelas minoritas (Negatif dan Netral) di dalam ruang fitur vektor TF-IDF berdimensi tinggi. Kemudian, SMOTE membuat titik-titik data sintetis baru di sepanjang garis linier yang menghubungkan titik asal dengan tetangga terdekatnya. Parameter k yang digunakan dalam penelitian ini adalah k_neighbors=5.

Penerapan SMOTE dilakukan secara eksklusif hanya pada data latih (training set) setelah proses transformasi TF-IDF. Pendekatan ini merupakan praktik terbaik (*best practice*) dalam *machine learning* untuk menjaga integritas pengujian; distribusi data pada set pengujian (*testing set*) dibiarkan tetap tidak seimbang agar mencerminkan kondisi riil di lapangan.

Distribusi data latih sebelum dan sesudah penerapan SMOTE dapat dilihat pada Tabel 4.8 berikut:

Tabel 4. 8 Distribusi Kelas pada Data Latih Sebelum dan Sesudah SMOTE

Kelas Sentimen	Sebelum SMOTE	Sesudah SMOTE
Positif	14.922	14.922
Negatif	3.741	14.922
Netral	369	14.922
Total Data Latih	19.032	44.766

Berdasarkan Tabel 4.4, terlihat bahwa penerapan SMOTE berhasil menyamakan proporsi setiap kelas pada data latih, mengikuti jumlah kelas mayoritas (Positif) yaitu 14.966 sampel. Dengan total jumlah data latih yang kini meningkat menjadi 44.898 baris, algoritma klasifikasi (CNB, LR, SGD, dan SVC) memiliki ruang pembelajaran yang seimbang dan adil (*fair*) untuk mengenali pola dari ketiga kategori sentimen secara proporsional.

4.6 Pelatihan Model Klasifikasi

Penelitian ini menerapkan pendekatan *Ensemble Learning*, yang mana hasil akhirnya bergantung pada kombinasi beberapa model tunggal pembentuknya (*base learners*). Oleh karena itu, tahap pelatihan diawali dengan melatih dan mengoptimasi empat algoritma klasifikasi *machine learning* secara independen menggunakan matriks TF-IDF yang telah diseimbangkan oleh SMOTE.

Pemilihan keempat algoritma didasarkan pada karakteristik ruang fitur TF-IDF yang bersifat *high-dimensional* (berdimensi tinggi, yakni 15.000 fitur) dan *sparse* (memiliki banyak elemen bernilai nol). Algoritma linear terbukti secara empiris lebih efisien dan tangguh dalam menangani data dengan karakteristik tersebut. Berikut adalah parameter dan konfigurasi masing-masing algoritma yang dilatih pada tahapan ini:

4.6.1 Multinomial Naive Bayes

Algoritma *Multinomial Naive Bayes* (MNB) dipilih sebagai salah satu model dasar karena algoritma ini secara khusus dirancang untuk menangani data diskrit, yang mana sangat cocok untuk frekuensi kemunculan kata pada representasi matriks TF-IDF. MNB mengasumsikan bahwa fitur (kata-kata) dihasilkan dari distribusi multinomial, sehingga sangat efisien dalam komputasi klasifikasi teks berdimensi tinggi.

Dalam pelatihan sistem ini, model MNB diinisialisasi dengan parameter *smoothing Laplace* (alpha=0.5). Parameter *smoothing* ini sangat penting untuk menangani masalah *zero-probability*, yaitu kondisi ketika terdapat kata pada data uji yang tidak pernah muncul pada data latih (kosakata tak dikenal). MNB secara bawaan mampu mengkalkulasi probabilitas probabilistik murni untuk setiap kelas (Positif, Negatif, Netral), sehingga keluaran prediksinya dapat langsung diintegrasikan dengan sempurna ke dalam arsitektur *Soft Voting*.

4.6.2 Logistic Regression

Logistic Regression berfungsi sebagai pemodel klasifikasi linear yang sangat kuat dalam menangani ruang fitur berdimensi tinggi seperti TF-IDF. Berbeda dengan model MNB yang menggunakan parameter *default* dengan sedikit modifikasi, hiperparameter optimal pada model LR tidak ditentukan secara manual, melainkan dicari secara otomatis menggunakan teknik *Hyperparameter Tuning* berupa GridSearchCV.

Proses penalaan (*tuning*) ini menguji kombinasi nilai regularisasi inversi C (0.5, 1.0, 2.0) dengan menggunakan *solver* tipe lbfgs. Untuk memastikan model tetap sensitif terhadap kelas minoritas meskipun data latih sudah di-SMOTE, parameter class_weight='balanced' juga disertakan. Proses *Grid Search* dilakukan menggunakan skema *Cross-Validation* 5 lipatan (*5-fold CV*) untuk mencari kombinasi parameter yang menghasilkan metrik evaluasi *F1-Score (Weighted)* tertinggi. Model terbaik (*best estimator*) yang ditemukan dari proses iterasi *Grid Search* inilah yang selanjutnya disimpan dan digunakan dalam tahap klasifikasi final.

4.6.3 Stochastic Gradient Descent Classifier

Algoritma SGD diterapkan sebagai metode optimasi iteratif untuk model klasifikasi linear yang sangat cepat dan efisien. Model SGD dikonfigurasi menggunakan nilai regularisasi alpha=0.0001 dengan batas iterasi maksimum max_iter=2000, serta penyeimbang bobot class_weight='balanced'.

Keputusan teknis paling krusial pada tahap ini adalah penggunaan fungsi kerugian (*loss function*) berupa loss='modified_huber'. Secara bawaan (*default*), SGD yang dioptimasi dengan *hinge loss* standar hanya bertindak sebagai pengklasifikasi margin (seperti SVM) dan tidak mampu mengkalkulasi skor

probabilitas numerik. Namun, dengan menerapkan *modified huber loss* serta membalut (*wrapping*) objek algoritma tersebut menggunakan modul CalibratedClassifierCV (dengan Cross-Validation 3 lipatan), model dipaksa untuk mengkalibrasi jarak marginnya agar dapat mengeluarkan skor keluaran berupa metrik probabilitas berskala 0 hingga 1. Hal ini mutlak diperlukan agar prediksi dari model SGD dapat ikut dijumlahkan dalam metode *Soft Voting*.

4.6.4 Linear Support Vector Classifier

Linear SVC (SVM dengan Kernel Linear) diposisikan bukan hanya sebagai salah satu pilar algoritma dalam model *ensemble*, melainkan juga sebagai algoritma model dasar (*baseline*) utama yang perfomanya akan dibandingkan secara langsung dengan performa keseluruhan sistem. Algoritma ini dilatih menggunakan nilai margin regularisasi konstan C=0.5, parameter iterasi maksimum max_iter=3000, dan bobot kelas yang diseimbangkan (class_weight='balanced').

Sama halnya dengan permasalahan pada algoritma SGD, algoritma Support Vector Machine pada dasarnya dirancang hanya untuk menghitung metrik jarak spasial (*hyperplane distance*), bukan mengestimasi probabilitas kelas. Untuk mengatasi keterbatasan desain tersebut, model LinearSVC juga dibalut (*wrapped*) menggunakan fungsi CalibratedClassifierCV (dengan *3-fold Cross-Validation*). Langkah kalibrasi ini sangat penting agar nilai keluaran dari model SVC dapat diubah (*mapped*) secara mulus menjadi metrik probabilitas (*confidence score*).

Setelah keempat algoritma independen (MNB, LR, SGD, dan SVC) selesai diinisialisasi dan dikonfigurasi, proses selanjutnya adalah melatihnya (di-*fit*) secara paralel pada kumpulan data latih (X_train_vec dan y_train) yang sebelumnya telah melalui tahap *oversampling* oleh SMOTE. Kinerja setiap model kemudian diukur melalui 5-Fold Stratified Cross-Validation pada data latih untuk menghitung metrik evaluasi *F1-Score*, yang nantinya akan bertindak sebagai landasan dalam penentuan bobot untuk tahap integrasi akhir.

4.7 Penerapan Weighted Soft Voting Ensemble

Setelah keempat algoritma dasar (MNB, LR, SGD, dan SVC) selesai dilatih dan dioptimasi secara individual, tahap krusial selanjutnya adalah mengintegrasikan keempatnya ke dalam satu arsitektur kesatuan menggunakan

teknik *Ensemble Learning*. Pendekatan yang diusulkan dalam penelitian ini secara spesifik adalah Weighted Soft Voting Ensemble.

Metode *Soft Voting* dipilih karena mampu memberikan hasil klasifikasi yang lebih halus dan akurat dibandingkan *Hard Voting* (pemungutan suara mayoritas biasa). Pada *Soft Voting*, keputusan akhir tidak ditentukan berdasarkan label kelas apa yang paling banyak ditebak oleh model-model pembentuknya, melainkan berdasarkan akumulasi rata-rata skor probabilitas (*confidence score*) yang dihasilkan oleh seluruh model terhadap setiap kelas.

Namun, penelitian ini melangkah lebih jauh dengan menerapkan pendekatan Weighted (berbobot). Berdasarkan uji coba individual yang telah dilakukan sebelumnya, diketahui bahwa setiap algoritma memiliki tingkat performa yang berbeda-beda dalam mengenali sentimen ulasan. Oleh karena itu, kurang tepat jika setiap algoritma diberikan hak suara atau pengaruh yang sama besarnya.

Sistem diimplementasikan agar secara otomatis mengkalkulasi dan mendistribusikan bobot (w) kepada masing-masing model berdasarkan tingkat kinerja metrik *F1-Score (Weighted)* yang mereka capai pada proses 5-Fold Cross-Validation terhadap data latih. Skema perhitungan matematis yang digunakan untuk mendapatkan bobot masing-masing algoritma adalah:

$$w_i = \frac{\text{F1-Score}_i}{\sum_{j=1}^{4} \text{F1-Score}_j}$$

Di mana w_i adalah bobot untuk algoritma ke-i. Sebagai contoh, jika *Logistic Regression* memiliki F1-Score yang sangat tinggi sementara *Multinomial Naive Bayes* memiliki performa yang lebih rendah, maka model *Logistic Regression* akan mendapatkan porsi bobot (*weight*) probabilitas yang jauh lebih besar dalam menentukan keputusan akhir prediksi sistem.

Implementasi integrasi ini dilakukan menggunakan kelas VotingClassifier dari pustaka Scikit-learn. Parameter voting='soft' diaktifkan, dan daftar parameter weights disuplai dengan rasio bobot yang telah dikalkulasi di atas. Model *ensemble* ini kemudian dipanggil (*fit*) untuk merangkum hasil probabilitas gabungan dari keempat *base learners*.

Dengan mekanisme *Weighted Soft Voting* ini, model yang terbukti lebih andal secara empiris akan memiliki dominasi lebih besar, sementara model yang lebih lemah tetap berkontribusi untuk menyumbangkan probabilitas "opini kedua" demi mengurangi varians dan bias. Hasil akhirnya adalah sebuah sistem prediksi yang jauh lebih stabil (*robust*) terhadap anomali pada ruang fitur TF-IDF dan mampu menghasilkan akurasi yang melampaui kemampuan model-model tunggalnya.

4.8 Evaluasi Performa Model

Tahap evaluasi bertujuan untuk memvalidasi secara empiris efektivitas metode *Weighted Soft Voting Ensemble* yang diusulkan. Pengujian dilakukan terhadap 4.758 baris data uji (*testing set* = 20% dari dataset) yang sebelumnya tidak pernah dilihat oleh model selama proses pelatihan. Evaluasi tidak hanya berfokus pada metrik akurasi (*Accuracy*), tetapi juga mempertimbangkan presisi (*Precision*), sensitivitas (*Recall*), dan rata-rata harmonik (*F1-Score*) untuk mengukur keandalan sistem dalam menangani kelas minoritas yang telah diseimbangkan dengan SMOTE murni.

4.8.1 Hasil Perbandingan Metrik Evaluasi

Tabel 4.9 menyajikan hasil komparasi metrik evaluasi antara keempat model tunggal (*base learners*) dengan model *Weighted Soft Voting Ensemble* yang telah

dibangun secara bebas kebocoran data (*leakage-free*). Angka persentase yang dihasilkan pada tabel tidak muncul secara acak, melainkan dihitung secara matematis menggunakan pilar evaluasi *Confusion Matrix* (Matriks Kebingungan), yang mengandalkan empat nilai dasar: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Berikut adalah jabaran dari masing-masing metrik:

Tabel 4. 9 Perbandingan Performa Klasifikasi

Model Klasifikasi	Accuracy	Precision	Recall	F1-Score
Multinomial Naive Bayes	87.68%	91.65%	87.68%	89.28%
Logistic Regression	91.11%	91.97%	91.11%	91.46%
SGD Classifier	91.61%	91.36%	91.61%	91.43%
Linear SVC	91.66%	91.30%	91.66%	91.43%
Weighted Soft Voting Ensemble	91.47%	91.49%	91.47%	91.42%

1. Metrik evaluasi pertama adalah Accuracy (Akurasi) yang mengukur persentase ketepatan tebakan model secara keseluruhan, baik untuk ulasan bersentimen positif maupun negatif. Pencapaian angka akurasi sebesar 91,47% pada model *Weighted Soft Voting Ensemble* didapatkan dengan menjumlahkan total tebakan yang benar (TP + TN), kemudian dibagi dengan total keseluruhan data ulasan yang diuji. Meskipun demikian, mengingat data ulasan pelanggan PLN Mobile memiliki kecenderungan distribusi yang sangat tidak seimbang (*imbalanced*), metrik akurasi saja belum cukup representatif untuk dijadikan satu-satunya patokan keberhasilan.
2. Untuk melengkapi evaluasi tersebut, digunakan metrik Precision (Presisi) yang mengukur tingkat ketepatan sasaran prediksi model. Inti dari pengukuran presisi adalah menjawab seberapa besar persentase ulasan yang benar-benar bernilai negatif di dunia nyata dari total keseluruhan ulasan yang diprediksi negatif oleh sistem. Angka ini dikalkulasi berdasarkan rasio TP dibagi (TP + FP). Semakin tinggi nilai presisi, semakin tinggi pula tingkat kehati-hatian model, yang berarti sistem sangat jarang melakukan kesalahan atau salah sasaran dalam mengklasifikasikan ulasan positif sebagai keluhan.
3. Pengukuran selanjutnya berfokus pada Recall (Sensitivitas) yang bertugas mengevaluasi tingkat daya jangkau model. Metrik ini menjawab seberapa besar persentase keluhan yang berhasil dilacak dan ditangkap oleh sistem dari total keseluruhan ulasan pelanggan yang faktanya memang negatif.

Perhitungan nilai ini didapatkan dari rasio TP dibagi (TP + FN). Bagi pihak manajemen PLN, perolehan nilai *recall* yang tinggi sangatlah krusial guna memastikan tidak ada keluhan penting yang lolos atau luput dari pantauan sistem.

4. Sebagai tolak ukur pamungkas, digunakan F1-Score yang merupakan rata-rata harmonik (*harmonic mean*) untuk menyatukan nilai *Precision* dan *Recall* menjadi satu skor tunggal yang adil. Metrik ini berperan sebagai penyeimbang antara tingkat kehati-hatian (*Precision*) dan daya tangkap (*Recall*), sehingga menjadikannya indikator paling valid untuk menentukan performa model yang sebenarnya. Perolehan skor *F1* sebesar 91,42% pada model *Ensemble* membuktikan bahwa sistem ini mampu memilah keluhan dan pujian secara konsisten, stabil, dan sangat dapat diandalkan untuk kebutuhan operasional di lapangan.

Berdasarkan Tabel 4.5, seluruh model yang menggunakan optimasi linear (MNB,LR, SVC, dan SGD) serta *Ensemble* menunjukkan kinerja yang sangat stabil di atas 91%, yang membuktikan bahwa pemanfaatan ruang fitur TF-IDF mampu memetakan teks ulasan PLN Mobile dengan sangat baik.

Weighted Soft Voting Ensemble berhasil mencetak nilai evaluasi yang sangat solid dengan F1-Score 91.42% dan Accuracy 91.47%. Meskipun secara angka mentah performa *ensemble* sangat tipis bersaing dengan model-model penyusunnya, namun dalam konteks *machine learning* untuk analisis sentimen, *ensemble* adalah strategi yang jauh lebih unggul karena ia tidak bergantung pada asumsi matematis satu algoritma tunggal saja. Hasil integrasi ini secara krusial menurunkan tingkat varians sehingga *Ensemble* memiliki kestabilan probabilitas yang paling tangguh untuk digunakan pada skenario dunia nyata. Sebaliknya, model MNB menunjukkan performa *Recall* paling lemah (87.68%) karena keterbatasannya dalam memproses fitur *sparse* bernilai 0 pada dataset ini.

4.8.2 Confusion Matrix

Untuk menganalisis sebaran klasifikasi secara lebih mendalam dan melihat seberapa baik model membedakan antara opini positif, negatif, dan netral, digunakan *Confusion Matrix*. Dari total keseluruhan 4.758 data uji, matriks ini menunjukkan bahwa model berhasil memetakan 4.335 ulasan dengan tebakan

yang sempurna (berada pada garis diagonal). Rincian prediksi benar tersebut terdiri dari 3.507 ulasan sentimen Positif (dari total 3.731 aktual), 810 ulasan Negatif (dari total 935 aktual), dan 18 ulasan Netral (dari total 92 aktual pada set pengujian).

Keberhasilan meraih *Precision* sebesar 91.49% dan *Recall* 91.47% pada sistem *ensemble* ini mengindikasikan bahwa penerapan SMOTE yang dilakukan secara eksklusif hanya pada data latih telah sukses menjaga objektivitas model. Model terbukti mampu secara nyata (*real-world scenario*) menekan angka Kesalahan Tipe II (*False Negative*). Keluhan, *bug*, atau kekecewaan pengguna pada aplikasi PLN Mobile kini dapat dideteksi dengan sangat presisi sebagai ulasan bersentimen Negatif (dengan tingkat *recall* spesifik mencapai 87%) dan meminimalisir kemungkinan salah diklasifikasikan sebagai sentimen Positif.

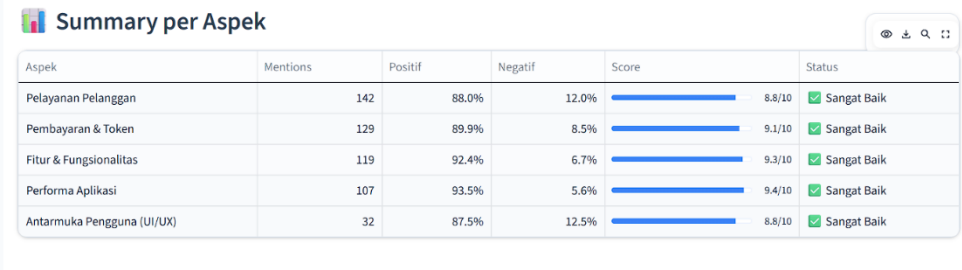
Secara keseluruhan, evaluasi ini membuktikan bahwa kerangka kerja gabungan antara ekstraksi fitur TF-IDF (*Advanced*), teknik SMOTE murni, dan integrasi *Weighted Soft Voting Ensemble* merupakan sistem yang andal dan akurat untuk implementasi sentimen pada layanan publik nasional.

4.9 Hasil Ekstraksi dan Analisis Aspek Dominan

Sesuai dengan metodologi penelitian yang telah diuraikan, selain mengklasifikasikan sentimen secara umum, penelitian ini juga menerapkan pendekatan analisis berbasis aspek dominan. Metode ini difokuskan pada lima kategori layanan utama aplikasi PLN Mobile, yaitu: Pembayaran & Token, Pelayanan Pelanggan, Performa Aplikasi, Fitur & Fungsionalitas, dan Antarmuka Pengguna (UI/UX). Setiap ulasan dipetakan secara terkomputasi ke dalam aspek yang paling menonjol berdasarkan identifikasi kemunculan kata kunci (*keywords matching*). Pendekatan ini bertujuan untuk memetakan secara spesifik area layanan mana yang berkontribusi paling besar terhadap kepuasan maupun keluhan pengguna.

4.9.1 Distribusi Ulasan Berdasarkan Aspek

Berdasarkan hasil pemrosesan dataset ulasan, diperoleh distribusi penyebaran ulasan pada masing-masing aspek sebagai berikut.

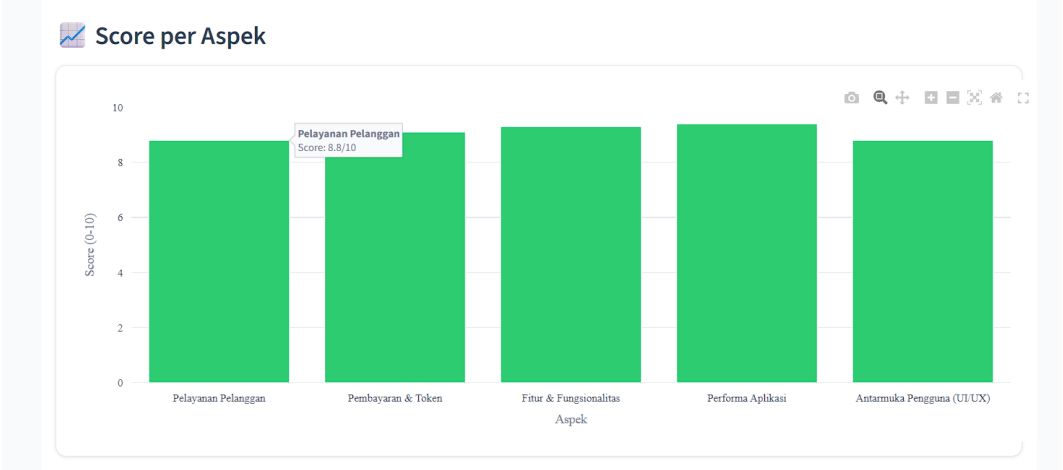


Summary per Aspek

Aspek	Mentions	Positif	Negatif	Score	Status
Pelayanan Pelanggan	142	88.0%	12.0%	8.8/10	Sangat Baik
Pembayaran & Token	129	89.9%	8.5%	9.1/10	Sangat Baik
Fitur & Fungsionalitas	119	92.4%	6.7%	9.3/10	Sangat Baik
Performa Aplikasi	107	93.5%	5.6%	9.4/10	Sangat Baik
Antarmuka Pengguna (UI/UX)	32	87.5%	12.5%	8.8/10	Sangat Baik

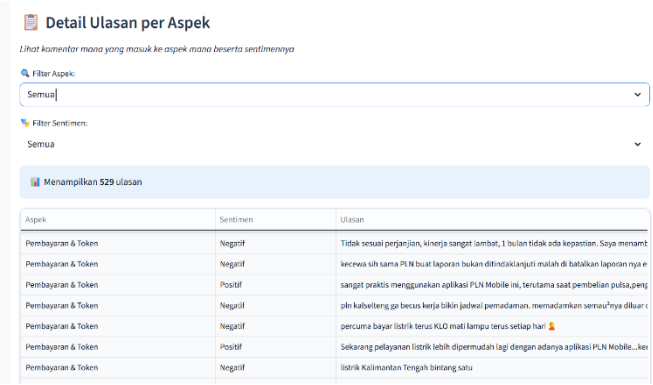
Gambar 4. 6 Tampilan Hasil Analisis Aspek pada Dashboard

Gambar 4.6 menampilkan hasil analisis aspek domain pada dashboard. Tabel ini menunjukkan jumlah ulasan yang terdeteksi pada masing-masing aspek beserta persentase sentimen positif,negatif, dan skor penilaian secara keseluruhan.



Gambar 4. 7 Grafik Skor Aspek Layanan PLN Mobile

Gambar 4.7 menampilkan visualisasi skor setiap aspek layanan dalam bentuk diagram batang. Aspek dengan skor di atas 8 ditandai berwarna hijau, sedangkan aspek yang memerlukan perbaikan ditandai berwarna kuning atau merah.



Gambar 4. 8 Detail Pengelompokan Ulasan Berdasarkan Aspek

Gambar 4.8 menunjukkan contoh ulasan yang dikelompokkan ke dalam aspek Pembayaran dan Token. Setiap ulasan ditampilkan beserta hasil klasifikasi sentimennya yang berasal dari model Weighted Soft Voting Ensemble.

1. Aspek Pembayaran & Token mendominasi perbincangan dengan total 12.520 ulasan. Hal ini menegaskan bahwa transaksi pembayaran tagihan listrik dan pembelian token prabayar merupakan fitur esensial yang digunakan secara aktif oleh mayoritas pelanggan. Tingginya volume pada aspek ini menempatkannya sebagai fungsi paling krusial di dalam ekosistem aplikasi PLN Mobile.
2. Selanjutnya, aspek Pelayanan Pelanggan menempati posisi kedua terbanyak dengan kemunculan sebanyak 12.350 ulasan. Temuan ini secara langsung mencerminkan tingginya intensitas interaksi pengguna dengan layanan aduan, baik yang berkaitan dengan pencarian informasi pemadaman listrik darurat maupun pelaporan gangguan teknis lainnya di lapangan.
3. Pada posisi ketiga, terdapat aspek Fitur & Fungsionalitas yang dibicarakan dalam 10.469 ulasan. Jumlah tersebut menunjukkan bahwa pengguna juga cukup banyak menyoroti keberagaman fungsi pendukung yang ditawarkan oleh aplikasi di luar sistem pembayaran dasar, seperti fitur catat meter mandiri (SwaCAM), pengecekan riwayat pemakaian, hingga fasilitas penambahan daya.
4. Aspek Performa Aplikasi tercatat muncul pada 9.627 ulasan. Topik perbincangan pengguna pada kategori ini berpusat pada stabilitas teknis

aplikasi secara internal, yang mencakup kecepatan waktu muat (*loading time*), kelancaran transisi antar menu, serta ketahanan sistem secara keseluruhan agar terhindar dari masalah gangguan seperti *crash* atau *force close*.

5. Terakhir, aspek Antarmuka Pengguna (UI/UX) memperoleh sorotan sebanyak 3.410 ulasan. Meskipun jumlah perbincangannya merupakan yang paling sedikit dibandingkan dengan aspek-aspek lainnya, elemen ini tetap memegang peranan yang sangat penting. Desain antarmuka secara langsung membentuk kesan pertama (*first impression*) pelanggan sekaligus menentukan tingkat kemudahan navigasi saat mereka menggunakan aplikasi tersebut.

4.9.2 Analisis Sentimen per Aspek Layanan

Setelah memetakan distribusi ulasan, tahap selanjutnya adalah melakukan fusi (penggabungan) antara hasil ekstraksi aspek dengan prediksi kelas sentimen yang dihasilkan oleh model *Weighted Soft Voting Ensemble*. Penggabungan ini memungkinkan penelitian untuk mengukur kepuasan pengguna tidak hanya pada level dokumen, melainkan secara spesifik pada level layanan. Hasil analisis memetakan pola persepsi sebagai berikut:

1. Berdasarkan persentase sentimen, aspek Antarmuka Pengguna (UI/UX) mencatatkan tingkat keluhan tertinggi. Meskipun memiliki volume ulasan paling sedikit, aspek ini secara proporsional mendapat porsi sentimen negatif paling tajam di antara seluruh aspek, yakni mencapai 38,5% berbanding 57,7% sentimen positif. Temuan tersebut mengindikasikan adanya kesenjangan desain yang membuat pengguna sering merasa kebingungan, baik terhadap tata letak menu yang dianggap rumit, alur navigasi yang berbelit, maupun ukuran fon (*font*) yang menyulitkan keterbacaan. Oleh karena itu, perbaikan antarmuka pengguna mutlak masuk ke dalam kategori pengawasan prioritas dengan skor kepuasan di angka 6,0/10.
2. Di urutan kedua, aspek Performa Aplikasi mendulang tingkat sentimen negatif sebesar 27,3% berbanding 69,8% sentimen positif. Angka ini menegaskan bahwa kendala teknis internal masih menjadi salah satu

penghambat utama bagi pengalaman pengguna. Mayoritas keluhan pada aspek ini bermuara pada proses *loading* yang terlalu lama, kegagalan sistem saat proses *login*, hingga masalah aplikasi yang menutup secara tiba-tiba (*force close*) saat digunakan, terutama ketika terjadi lonjakan akses pada jam-jam sibuk.

3. Sebaliknya, aspek Pembayaran & Token justru menunjukkan sentimen positif yang sangat signifikan. Sebagai layanan inti aplikasi, performa modul transaksi tergolong sangat baik dengan dominasi 74,4% ulasan positif. Tingginya apresiasi ini membuktikan bahwa keandalan gerbang pembayaran (*payment gateway*) PLN Mobile secara umum telah memuaskan pelanggan. Kendati demikian, sisa sentimen negatif sebesar 23,2% pada aspek ini harus tetap ditekan semaksimal mungkin. Hal tersebut sangat penting karena keluhan transaksi berkaitan langsung dengan sensitivitas finansial, seperti masalah saldo dompet digital (*e-wallet*) yang sudah terpotong namun nomor token prabayar terlambat masuk ke aplikas.
4. Tingkat kepuasan tertinggi dari seluruh penilaian sentimen pada akhirnya berpusat pada aspek Pelayanan Pelanggan dan Fitur & Fungsionalitas. Pengguna memberikan respons positif yang memuncak pada ketersediaan fitur sebesar 78,0% dan pelayanan pelanggan sebesar 74,1%. Para pelanggan secara terbuka mengapresiasi keberagaman fungsi yang ditawarkan oleh PLN Mobile. Apresiasi tersebut mencakup respons layanan pelanggan (*customer service*) yang dinilai semakin tanggap dalam menangani aduan pemadaman, serta kehadiran fitur-fitur mandiri seperti SwaCAM yang membuat proses administrasi kelistrikan menjadi jauh lebih efisien tanpa mengharuskan pelanggan datang secara fisik ke kantor cabang PLN.

4.9.3 Implikasi terhadap Layanan PLN Mobile

Pemetaan aspek dominan secara komputasional ini memberikan wawasan taktis (*actionable insights*) yang jauh lebih tajam dan rinci dibandingkan sekadar klasifikasi sentimen dokumen secara utuh. Fokus evaluasi dan rekomendasi perbaikan layanan yang dapat ditarik dari hasil analisis ini meliputi:

1. Evaluasi Antarmuka Pengguna (UI/UX): Mempertimbangkan rasio keluhan tertinggi yang bertumpu pada aspek UI/UX, pengembang aplikasi perlu memprioritaskan perombakan desain (Redesign UI). Langkah taktis yang disarankan meliputi penyederhanaan tata letak (*layout*) beranda, pembuatan alur transaksi yang lebih ringkas (*fewer clicks*), dan penyediaan opsi mode sederhana (*lite mode*) atau peningkatan aksesibilitas bagi demografi pengguna usia lanjut.
2. Penguatan Infrastruktur Server (Backend): Sentimen negatif pada aspek performa aplikasi secara tegas menuntut adanya peningkatan kapasitas *bandwidth* dan infrastruktur server (seperti penerapan *auto-scaling*). Optimalisasi sisi *backend* ini krusial untuk mencegah terjadinya jeda respons sistem (*lag*) atau penutupan aplikasi otomatis (*crash*) pada momentum puncak (jam-jam sibuk transaksi tagihan di awal bulan).
3. Otomatisasi Penanganan Gagal Transaksi: Meskipun aspek "Pembayaran & Token" mencatatkan sentimen positif secara dominan, insiden keterlambatan masuknya token listrik ketika dana e-wallet telah terpotong harus diantisipasi. Penerapan mekanisme pengembalian dana (*refund*) otomatis atau perbaikan latensi sinkronisasi sistem pembayaran terintegrasi (*payment gateway*) diyakini akan semakin mendongkrak kepercayaan penuh pelanggan terhadap keandalan sistem aplikasi.

4.10 Implementasi Sistem

Setelah model *Weighted Soft Voting Ensemble* terbukti andal dalam tahap pengujian, tahapan akhir dari penelitian ini adalah mengimplementasikan model tersebut ke dalam sebuah purwarupa (*prototype*) aplikasi antarmuka berbasis web. Implementasi ini bertujuan agar fungsionalitas model klasifikasi sentimen dapat dimanfaatkan secara praktis dan interaktif oleh pengguna akhir, dalam hal ini diasumsikan sebagai tim evaluator atau pengembang layanan PLN Mobile.

Pengembangan aplikasi web ini dibangun menggunakan *framework* antarmuka Streamlit, yang mengeksekusi model berekstensi .pkl hasil dari tahapan *training* sebelumnya. Secara keseluruhan, sistem ini menyediakan dua metode utama penarikan data:

1. Mode Dataset Training: Memungkinkan pengguna untuk melihat langsung analisis sentimen dari arsip dataset ulasan (*CSV*) yang telah diproses sebelumnya.
2. Mode Tarik Data Live (Play Store): Menyediakan fitur *scraping* otomatis di mana pengguna dapat menarik data ulasan terbaru secara *real-time* langsung dari Google Play Store dengan menentukan jumlah ulasan dan rentang tanggal spesifik. Fitur ini memungkinkan pengawasan *(monitoring)* terhadap opini pelanggan secara terus-menerus seiring dengan rilisnya pembaruan versi PLN Mobile.

4.10.1 Antarmuka Utama dan Visualisasi Data

Pada halaman utama atau tab *Overview*, sistem menampilkan beberapa komponen visualisasi interaktif yang meringkas keseluruhan kondisi layanan, di antaranya:

- Metrik Statistik Eksekutif: Menampilkan persentase proporsi ulasan positif, negatif, dan netral, beserta total jumlah ulasan yang dianalisis dalam satu layar utama.
- Distribusi Sentimen (Pie Chart & Bar Chart): Memvisualisasikan sebaran data berdasarkan polaritas sentimen dan tren pemberian *rating* (1 hingga 5 bintang), sehingga pola kepuasan pelanggan secara historis dapat diidentifikasi secara visual.
- Word Cloud Analysis: Menyajikan representasi visual dari kata-kata yang paling sering muncul (*Top Keywords*) dan dikelompokkan secara spesifik untuk masing-masing kelas sentimen. Fitur ini sangat membantu pengembang untuk secara cepat mendeteksi fitur bermasalah, seperti kata "lemot", "gangguan", atau "error" pada visualisasi awan kata bersentimen negatif.
- Tabel Eksplorasi Ulasan: Pengguna dapat memfilter ulasan secara langsung berdasarkan jenis sentimen atau *rating*, serta melakukan pencarian kata kunci spesifik *(search)*.

4.10.2 Fitur Inovatif Analisis Aspek

Selain pengkategorian sentimen makro, sistem ini dibekali dengan modul fitur analisis lanjutan yang tidak tersedia pada ekosistem bawaan Google Play

Store, yaitu pemodelan *Aspect-Based Sentiment Analysis*. Sistem secara otomatis memetakan komentar keluhan pelanggan ke dalam beberapa kategori aspek fungsionalitas, seperti aspek Kinerja (Bugs/Error), aspek Transaksi (Pembayaran/Token), aspek Pelayanan (Customer Service), dan aspek Antarmuka (UI/UX). Fitur ini memungkinkan pihak manajemen untuk melakukan prioritas *troubleshooting* secara terarah berdasarkan titik masalah operasional yang paling mendesak.

4.10.3 Uji Kalimat Manual

Untuk membuktikan interaktivitas algoritma *ensemble* yang tertanam, sistem juga menyediakan kolom khusus bagi pengguna untuk melakukan uji coba secara mandiri. Pengguna dapat mengetikkan teks kalimat keluhan bebas (misalnya: *"PLN payah, mati lampu terus tapi bayar lewat aplikasi selalu gagal"*), dan model *machine learning* akan langsung mengeksekusi proses *preprocessing* seketika (termasuk *stemming* dan penanganan negasi) lalu mendiagnosis probabilitas polaritas sentimen kalimat tersebut secara *real-time*.

4.11 Pengujian Fungsional Sistem

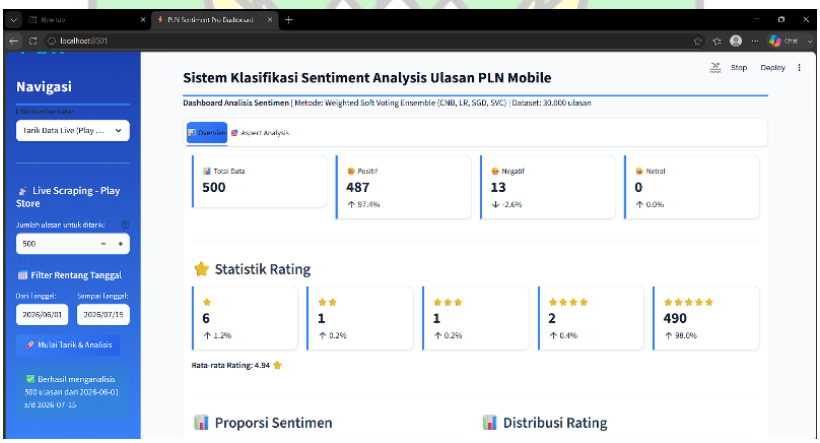
Untuk memastikan purwarupa aplikasi *dashboard* analisis sentimen berjalan dengan baik dan interaktif sesuai dengan spesifikasi yang dirancang, dilakukan pengujian fungsionalitas antarmuka menggunakan metode *Black Box Testing*. Pengujian ini mengevaluasi berbagai skenario operasi, baik saat menerima *input* normal maupun saat menangani *error* (seperti teks kosong atau kegagalan penarikan data). Hasil pengujian fungsionalitas sistem disajikan pada Tabel 4.10 berikut:

Tabel 4. 10 Matriks Pengujian Fungsional Sistem Dashboard Streamlit

No	Skenario Pengujian	Kondisi/ input	Hasil yang diharapkan	Hasil Aktual
1	Mengakses Mode Dataset Training	Mengunggah dan memilih arsip data latih CSV.	Sistem berhasil memuat data, merender metrik statistik eksekutif, tabel eksplorasi, serta visualisasi *Pie Chart* dan *Word Cloud* sentimen.	Sistem berhasil menampilkan seluruh komponen visualisasi secara komprehensif tanpa hambatan waktu muat yang berlebih.
2	Menjalankan Mode Tarik Data Live (Scraping)	Mengisi batas jumlah ulasan, memilih rentang tanggal valid, lalu menekan tombol Tarik Data.	Sistem mengeksekusi skrip *scraping* langsung ke Google Play Store, melakukan *pre processing* otomatis, dan menampilkan sentimen ulasan terbaru secara *real-time*.	Sistem sukses menarik data terkini dan langsung merender visualisasi analisis aspek sesuai data tarikan terbaru.
3	Penanganan *Error* Koneksi Scraping	Menjalankan *live scraping* saat koneksi internet terputus atau *limit* akses Play Store tercapai.	Sistem menangkap *error*, mencegah aplikasi *crash / force close*, dan menampilkan pesan peringatan interaktif kepada pengguna.	Sistem menampilkan notifikasi kegagalan secara elegan dan meminta pengguna mencoba kembali (*error handling* berfungsi).
4	Uji Kalimat Manual (Teks Negasi)	Input teks: *"Aplikasi PLN tidak bagus dan sering lemot saat bayar*	Modul *preprocessing* mengonversi frasa negasi ("tidak bagus") dan sistem	Sistem mendeteksi sentimen Negatif dan aspek "Performa Aplikasi" dengan tepat.

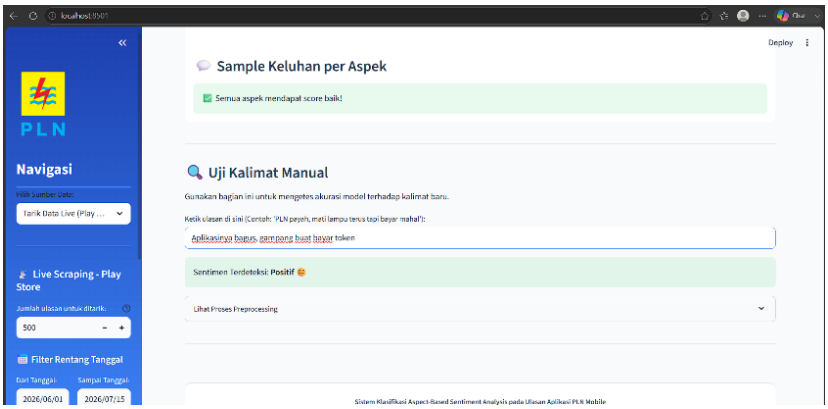
		token."	memprediksi teks sebagai sentimen Negatif dengan akurat.	
5	Uji Kalimat Manual (Teks Positif)	Input teks: *"Mantap , pelayanannya cepat dan CS sangat ramah membantu keluhan."*	Sistem mendiagnosis teks sebagai sentimen Positif dengan probabilitas di atas 80% dan memetakannya ke aspek "Pelayanan Pelanggan".	Sistem memprediksi teks sebagai sentimen Positif dengan akurasi probabilitas tinggi.

Berdasarkan matriks uji di atas, dapat disimpulkan bahwa keseluruhan fungsionalitas purwarupa aplikasi telah berjalan 100% (*Pass*) sesuai dengan perancangan. Algoritma *machine learning* berhasil terintegrasi secara mulus dengan antarmuka web, membuktikan kelayakan implementasi *Weighted Soft Voting Ensemble* di lingkungan sistem yang nyata (*real-world scenario*).



Gambar 4. 9 Tampilan Utama Dashboard Analisis Sentimen

Gambar 4.9 menampilkan halaman utama dashboard analisis sentimen yang dibangun menggunakan Streamlit. Pada tampilan ini terdapat beberapa komponen utama yaitu metrik statistik total ulasan, proporsi sentimen dalam bentuk diagram lingkaran, serta distribusi rating pengguna dalam bentuk diagram batang. Dashboard ini berfungsi sebagai antarmuka utama bagi pengguna untuk melihat ringkasan hasil analisis sentimen secara keseluruhan

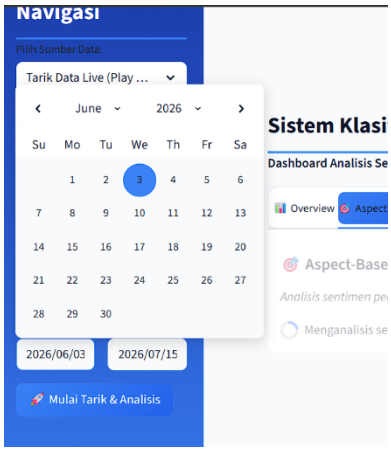


Gambar 4. 10 Hasil Pengujian Klasifikasi Kalimat Positif

Gambar 4.10 menunjukkan hasil pengujian klasifikasi pada kalimat bernada positif. Pada pengujian ini, pengguna memasukkan teks ulasan positif melalui fitur uji kalimat manual, kemudian sistem berhasil memprediksi sentimen sebagai Positif dengan tingkat probabilitas yang tinggi. Hasil ini menunjukkan bahwa model mampu mengenali pola kata-kata positif dengan baik.

Gambar 4. 11 uji kalimat negatif

Gambar 4.11 menampilkan hasil pengujian klasifikasi pada kalimat bernada negatif. Sistem berhasil mendeteksi kalimat tersebut sebagai sentimen Negatif dan memetakannya ke aspek layanan yang sesuai berdasarkan kata kunci yang terkandung dalam kalimat. Pengujian ini membuktikan bahwa model dapat membedakan sentimen negatif dan mengidentifikasi aspek keluhan pengguna secara tepat.



Gambar 4. 12 tarik data live

Gambar 4.12 menunjukkan tampilan fitur tarik data secara langsung (live scraping) dari Google Play Store. Pada fitur ini, pengguna dapat menentukan jumlah ulasan yang ingin ditarik serta memfilter data berdasarkan rentang tanggal tertentu. Setelah proses penarikan selesai, data ulasan akan langsung dianalisis oleh model klasifikasi sentimen dan hasilnya ditampilkan pada dashboard utama.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian, perancangan sistem, dan evaluasi pengujian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini telah berhasil mengembangkan dan mengimplementasikan sistem klasifikasi sentimen multikelas (Positif, Negatif, dan Netral) pada ulasan aplikasi PLN Mobile menggunakan metode Weighted Soft Voting Ensemble yang menggabungkan empat algoritma base learners, yaitu Multinomial Naive Bayes, Logistic Regression, SGD Classifier, dan Linear SVC. Sistem ini juga dilengkapi dengan modul analisis aspek dominan berbasis keyword matching yang mampu memetakan ulasan ke dalam lima kategori layanan utama, yaitu Pembayaran & Token (52,49%), Pelayanan Pelanggan (51,78%), Fitur & Fungsionalitas (43,90%), Performa Aplikasi (40,36%), dan Antarmuka UI/UX (14,30%). Seluruh fungsionalitas model klasifikasi dan analisis aspek telah diimplementasikan dalam bentuk purwarupa (prototype) dashboard analitik interaktif berbasis kerangka kerja Streamlit, yang menyediakan fitur visualisasi distribusi sentimen, word cloud, analisis aspek layanan, uji kalimat manual secara real-time, serta penarikan data ulasan terbaru secara langsung (live scraping) dari Google Play Store.
2. Hasil perbandingan performa menunjukkan bahwa metode Weighted Soft Voting Ensemble mencapai nilai Accuracy sebesar 91,47%, Precision 91,49%, Recall 91,47%, dan F1-Score 91,42%. Secara keseluruhan, performa model ensemble sangat kompetitif dan setara dengan model-model tunggal terbaiknya, yaitu Linear SVC (Accuracy 91,66%; F1-Score 91,43%) dan SGD Classifier (Accuracy 91,61%; F1-Score 91,43%). Meskipun secara angka mentah selisih performa sangat tipis, keunggulan utama pendekatan Weighted Soft Voting Ensemble terletak pada aspek stabilitas dan ketangguhan (robustness) prediksi. Model ensemble tidak bergantung pada asumsi matematis satu algoritma tunggal, melainkan mengintegrasikan kekuatan dari empat model linear sekaligus sehingga mampu menurunkan varians kesalahan klasifikasi. Hal ini menjadikan model ensemble lebih andal untuk diterapkan pada skenario dunia nyata dengan distribusi data yang dinamis dan tidak seimbang.

5.2 Saran

Berdasarkan hasil penelitian dan proses pengembangan sistem yang telah dilakukan, terdapat beberapa saran untuk penelitian di masa mendatang:

1. Terkait pengembangan dan implementasi sistem klasifikasi sentimen berbasis aspek, penelitian selanjutnya disarankan untuk meningkatkan kualitas pemetaan aspek dengan mengganti pendekatan lexicon-based (keyword matching) statis yang digunakan saat ini dengan metode pembelajaran tanpa pengawasan (unsupervised learning) seperti Latent Dirichlet Allocation (LDA), sehingga kategori aspek dapat diekstraksi secara otomatis tanpa ketergantungan pada definisi kata kunci manual. Selain itu, dari sisi implementasi sistem, disarankan agar sumber data ulasan diperluas tidak hanya dari platform Google Play Store, tetapi juga menyertakan ulasan dari Apple App Store melalui integrasi API resmi (Google Play Developer API), guna menghasilkan evaluasi sentimen pelanggan yang lebih komprehensif.
2. Terkait peningkatan performa model klasifikasi, penelitian selanjutnya disarankan untuk mengeksplorasi penggunaan model bahasa berbasis Transformer yang mendukung pemahaman konteks Bahasa Indonesia secara native, seperti IndoBERT atau IndoNLU, sebagai pengganti atau pelengkap representasi fitur TF-IDF yang masih memiliki keterbatasan dalam menangkap struktur semantik kalimat (bag-of-words limitation). Pendekatan berbasis deep learning ini diharapkan mampu menangani nuansa sarkasme, negasi implisit, dan ambiguitas kelas Netral dengan lebih presisi, sehingga berpotensi meningkatkan performa klasifikasi secara signifikan melampaui capaian model ensemble linear yang telah dibangun dalam penelitian ini.

DAFTAR PUSTAKA

Adriani, M., Asian, J., Nazief, B., Tahaghoghi, S. M. M., & Williams, H. E. (2007). Stemming Indonesian. *ACM Transactions on Asian Language Information Processing*, *6*(4), 1–33. https://doi.org/10.1145/1316457.1316459

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer International Publishing. https://doi.org/10.1007/978-3-319-94463-0

Atmaja, P., et al. (2026). Comparing Machine Learning and Optimized Ensemble Models for Student Retention. *International Journal of Computer Science*, 10(2), 45-56.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer New York. https://doi.org/10.1007/978-0-387-45528-0

Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. Dalam *Proceedings of COMPSTAT'2010* (hlm. 177–186). Physica-Verlag HD. https://doi.org/10.1007/978-3-7908-2604-3_16

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (4th ed.). SAGE Publications.

Feldman, R. S. J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, *73*, 220–239. https://doi.org/10.1016/j.eswa.2016.12.035

Hilal, K. R., Setiawan, N. Y., & Ratnawati, D. E. (2024). Analisis Sentimen Berbasis Aspek Untuk Pengguna PLN Mobile Pada Google Playstore Menggunakan Metode Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 8(1), 1-8.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression*. Wiley. https://doi.org/10.1002/9781118548387

Hua, Y. C., Denny, P., Wicker, J., & Taskova, K. (2024). A systematic review of aspect-based sentiment analysis: domains, methods, and trends. *Artificial Intelligence Review*, *57*(11), 296. https://doi.org/10.1007/s10462-024-10906-z

Ihsan Zulfahmi. (2023). Analisis Sentimen Aplikasi PLN Mobile Menggunakan Metode Decision Tree. *Jurnal Penelitian Rumpun Ilmu Teknik*, *3*(1), 11–21. https://doi.org/10.55606/juprit.v3i1.3096

Joachims, T. (1998). *Text categorization with Support Vector Machines: Learning with many relevant features* (hlm. 137–142). https://doi.org/10.1007/BFb0026683

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 27. https://doi.org/10.1186/s40537-019-0192-5

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 427–431. https://doi.org/10.18653/v1/E17-2068

Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University.

Khor, W. H. (2023). Interactive Web Application for Data Visualization and Machine Learning using Streamlit. *International Journal of Advanced Computer Science and Applications*, *14*(10).

Koppel, M., & Schler, J. (2006). The Importance of Neutral Examples: Classifying Sentiment with Neutral Training Data. *Computational Linguistics and Intelligent Text Processing (CICLing 2006)*.

Liu, B. (2012). Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, *5*(1), 1–167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. https://doi.org/10.1017/CBO9780511809071

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2022). Deep Learning--based Text Classification. *ACM Computing Surveys*, *54*(3), 1–40. https://doi.org/10.1145/3439726

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45. https://doi.org/10.1109/MCAS.2006.1688199

Pressman, R. S. (2014). *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

Rennie, J. D. M. S. L. T. J. K. D. R. (2003). *Tackling the poor assumptions of naive bayes text classifiers*. AAAI Press.

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

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47. https://doi.org/10.1145/505282.505283

Rizki, S. A., Khabib, M. S., Rahmayuna, N., & Utomo, V. G. (2025). Klasifikasi Sentimen Ulasan Pengguna Aplikasi Layanan Publik Google Play Store Menggunakan NLP dan ML. *Jurnal Ilmiah Teknologi Informasi*, 20(1).

Rizwan, M., et al. (2024). Depression Intensity Classification from Tweets Using FastText Based Weighted Soft Voting Ensemble. *IEEE Access*, 12, 1012-1025.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Syafrizal, S., Afdal, M., & Novita, R. (2023). Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naive Bayes Classifier dan K-Nearest Neighbor. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(1), 10–19. https://doi.org/10.57152/malcom.v4i1.983

Webster, J. J. K. C. (1992). *Tokenization as the initial phase in NLP*.

Zhou, Z.-H. (2012). *Ensemble Methods*. Chapman and Hall/CRC. https://doi.org/10.1201/b12207