PERBANDINGAN KINERJA ALGORITMA RANDOM FOREST DAN XGBOOST TERHADAP ANALISIS SENTIMEN APLIKASI E-COMMERCE

SKRIPSI

DISUSUN OLEH

LAILA SALSABILA
NPM. 2109020051

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025

PERBANDINGAN KINERJA ALGORITMA RANDOM FOREST DAN XGBOOST TERHADAP ANALISIS SENTIMEN APLIKASI E-COMMERCE

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

LAILA SALSABILA
NPM. 2109020051

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2025

LEMBAR PENGESAHAN

Judul Skripsi	:	Perbandingan Kinerja Algoritma Random Forest dan XGBoost Terhadap Analisis Sentimen Aplikasi E-Commerce
Nama Mahasiswa	:	Laila Salsabila
NPM	:	2109020051
Program Studi	:	Teknologi Informasi

Menyetujui
Komisi Pembimbing

(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Ketua Program Studi

(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PERBANDINGAN KINERJA ALGORITMA RANDOM FOREST DAN XGBOOST TERHADAP ANALISIS SENTIMEN APLIKASI E-COMMERCE

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juni 2025

Yang membuat pernyataan

Laila Salsabila

NPM. 2109020051

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Laila Salsabila
NPM	: 2109020051
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

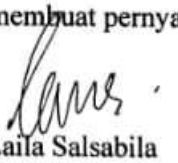
PERBANDINGAN KINERJA ALGORITMA RANDOM FOREST DAN XGBOOST TERHADAP ANALISIS SENTIMEN APLIKASI E-COMMERCE

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juni 2025

Yang membuat pernyataan



Laila Salsabila

NPM. 2109020051

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Laila Salsabila
Tempat dan Tanggal Lahir	: Medan, 10 Juni 2003
Alamat Rumah	: Asrama Yonkav 6, Jl. Bunga Raya, Medan
Telepon/Faks/HP	: 081375321931
E-mail	: shalsabila0610@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SDN 065011 Medan	TAMAT: 2015
SMP	: SMP Negeri 30 Medan	TAMAT: 2018
SMA	: SMA Brigjend Katamso I Medan	TAMAT: 2021

KATA PENGANTAR

Pendahuluan

Dengan rasa syukur yang mendalam, penulis menyampaikan apresiasi setinggi-tingginya kepada berbagai pihak yang telah berkontribusi memberikan dukungan, inspirasi, serta restu selama proses penyusunan karya ilmiah ini. Secara khusus, ucapan terima kasih penulis sampaikan kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Halim Maulana, ST., M.Kom., Selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatra Utara.
4. Bapak Lutfi Basit, S.Sos., M.I.Kom., Selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatra Utara
5. Ibu Fatma Sari Hutagalung S.Kom., M.Kom. Ketua Program Studi Teknologi Informasi sekaligus Pembimbing Skripsi
6. Bapak Mhd. Basri S.Si, M.Kom. Sekretaris Program Studi Teknologi Informasi
7. Rasa syukur yang terdalam penulis haturkan kepada Bapak Ngadiyo dan Ibunda Rini Sundari selaku orang tua tercinta yang senantiasa melimpahkan restu, kasih, serta dukungan tanpa reserve. Atas segala pengorbanan mereka dalam memenuhi kebutuhan material selama ini, penulis akhirnya dapat menyelesaikan jenjang pendidikan hingga mencapai gelar sarjana.
8. Penulis juga berterima kasih kepada teman-teman dan semua pihak yang telah berkontribusi, baik secara langsung maupun tidak langsung. Maaf apabila penulis tidak dapat menyampaikan rasa terima kasih secara individual kepada setiap pihak.

ABSTRAK

Perkembangan teknologi digital yang begitu pesat telah membuat aplikasi e-commerce, termasuk Shopee sebagai salah satu platform terkemuka dengan jutaan pengguna aktif, semakin banyak digunakan di Indonesia. Review pengguna pada Google Play Store adalah sumber data yang sangat berharga untuk mengevaluasi kualitas pelayanan sekaligus menganalisis sentimen konsumen. Namun, tingginya volume data serta kerumitan bahasa yang digunakan dalam ulasan mengharuskan adanya pendekatan analisis yang tepat dan efisien. Studi ini dirancang untuk mengevaluasi dan membandingkan efektivitas dua algoritma klasifikasi yang berbeda, yaitu Random Forest dan XGBoost dalam mengidentifikasi sentimen dari ulasan pengguna Shopee. Akuisisi data tekstual ulasan dilakukan secara terprogram melalui proses scraping lewat Play Store, proses lanjutan menerapkan workflow NLP standar yang mencakup: annotasi label, case normalization, data cleansing, tokenization, stopword elimination, dan morphological stemming. Tahap berikutnya melibatkan ekstraksi fitur dengan menerapkan skema TF-IDF, dilanjutkan dengan partisi korpus data dipartisi ke dalam dua kelompok terpisah, yakni subset pelatihan (training) dan subset validasi (testing). Kerangka klasifikasi kemudian dikonstruksi menggunakan algoritma Random Forest dan XGBoost, yang selanjutnya dinilai kinerjanya melalui serangkaian indikator evaluasi meliputi akurasi, presisi, recall, dan F1-score. Temuan dari studi ini diharapkan mampu menawarkan rekomendasi tentang algoritma yang memiliki efektivitas tertinggi dalam penilaian sentimen untuk platform e-commerce, serta memberikan kontribusi bagi peningkatan kualitas layanan dan penelitian lanjutan di bidang data mining dan pemrosesan bahasa alami.

Kata Kunci: *Random Forest*, *XGBoost*, TF-IDF, metrik akurasi, *precision*, *recall*, *F1-score*.

ABSTRACT

The rapid advancement of digital technology has led to the widespread adoption of e-commerce applications in Indonesia, with Shopee standing out as a leading platform boasting millions of active users. User reviews on the Google Play Store serve as a highly valuable data source for evaluating service quality and analyzing consumer sentiment. However, the substantial volume of data and the linguistic complexity within these reviews necessitate a precise and efficient analytical approach. This study is designed to evaluate and compare the effectiveness of two distinct classification algorithms Random Forest and XGBoost in identifying sentiment from Shopee user reviews. The textual review data was acquired programmatically via scraping from the Play Store. Subsequent processing implemented a standard NLP workflow, encompassing: label annotation, case normalization, data cleansing, tokenization, stopword elimination, and morphological stemming. The next stage involved feature extraction using the TF-IDF scheme, followed by partitioning the dataset into two separate subsets: training data and testing data. A classification framework was then constructed using the Random Forest and XGBoost algorithms, the performance of which was assessed through a series of evaluation metrics including accuracy, precision, recall, and F1-score. The findings of this study are expected to provide a recommendation on which algorithm demonstrates the highest effectiveness for sentiment assessment on e-commerce platforms, while also contributing to the enhancement of service quality and to further research within the disciplines of data mining and natural language processing.

Keywords: Random Forest, XGBoost, TF-IDF, accuracy metric, precision, recall, F1-score.

DAFTAR ISI

DAFTAR TABEL

HALAMAN

DAFTAR GAMBAR

HALAMAN

BAB I
PENDAHULUAN

1.1. Latar Belakang Masalah

Melambungnya teknologi kontemporer telah mentransformasi berbagai aspek dalam kehidupan, salah satunya adalah cara masyarakat berinteraksi dengan layanan digital. Dalam konteks ini, platform e-commerce hadir menjadi terobosan penting yang menyederhanakan aktivitas beli-beli online. Shopee, yang merupakan salah satu penyedia layanan e-commerce terdepan di Indonesia, mencatatkan jumlah pengguna aktif yang masif. Pengguna ini secara rutin memberikan penilaian dan umpan balik berdasarkan pengalaman mereka selama menggunakan aplikasi tersebut. Penilaian pengguna yang dituangkan melalui ulasan di Google Play Store berperan penting dalam menilai kualitas aplikasi, baik dari aspek fitur, kenyamanan penggunaan, hingga kepuasan terhadap layanan yang ditawarkan (Saputra et al., 2025).

Umpan balik yang dituliskan pengguna di Google Play Store memiliki nilai ganda: selain merefleksikan pengalaman langsung mereka, komentar-komentar tersebut juga berfungsi sebagai sumber data berharga bagi developer aplikasi dalam mengevaluasi dan meningkatkan kualitas platform. Dengan adanya berbagai masukan tersebut, pengembang dapat mengetahui kebutuhan pengguna dan meningkatkan kualitas layanan yang ditawarkan. Di sisi lain, calon pengguna baru juga dapat mempertimbangkan ulasan sebagai referensi sebelum mengunduh dan menggunakan aplikasi Shopee, sehingga pelaksanaan umpan balik ini memberikan dampak luas baik bagi pengembang maupun komunitas pengguna aplikasi.

Ulasan ini tidak hanya berisi penilaian positif terhadap fitur yang memudahkan transaksi, tetapi juga berisi penilaian negatif seperti keluhan mengenai kendala teknis, seperti bug, error, atau masalah pada sistem pembayaran dan pengiriman. Jumlah ulasan yang sangat banyak sering kali membuat sulit bagi pengembang aplikasi untuk menganalisis sentimen pengguna secara manual. Selain itu, kompleksitas bahasa seperti penggunaan slang (“mantap”, “gak nyampe”) dan ambiguitas, konteks yang berpotensi menyebabkan kesalahan klasifikasi. Dengan adanya volume data yang besar dan ketidakseimbangan kelas (lebih banyak ulasan positif atau negatif) sehingga memerlukan algoritma yang robust untuk meminimalkan bias dalam proses analisis. Dengan meningkatnya volume dan ragam data yang harus diolah, diperlukan metode komputasi yang mampu menangani informasi tersebut secara efektif dan efisien. Pendekatan ini sangat penting untuk memastikan proses pengolahan data dapat berjalan cepat dan akurat, sekaligus mendukung pengambilan keputusan yang tepat berdasarkan data yang tersedia (Nurian et al., 2024).

Berdasarkan permasalahan diatas, analisis sentimen diperlukan untuk mengubah data tekstual ulasan menjadi insight strategis, seperti identifikasi masalah logistik atau kualitas produk. Data ulasan diambil langsung dari aplikasi Google Playstore pada ulasan Aplikasi Shopee, dengan memperhatikan kebijakan privasi Shopee dan menghindari penggunaan informasi pribadi pengguna. Menurut penelitian, 70% konsumen menganggap ulasan sebagai faktor krusial dalam keputusan pembelian, sehingga klasifikasi sentimen yang akurat menjadi dasar peningkatan layanan. Analisis sentimen merupakan proses komputasi yang memanfaatkan kemampuan Natural Language Processing (NLP) serta machine

learning untuk mengkategorikan teks ke bagian kelompok positif maupun negatif. Salah satu model yang sering digunakan adalah Random Forest, yang membangun banyak pohon keputusan secara bersamaan (metode bagging) dan menggabungkan hasilnya guna meningkatkan ketepatan prediksi. Algoritma ini dikenal karena ketahanannya terhadap overfitting serta kemampuannya dalam menangani data dalam jumlah besar. Sementara itu, XGBoost mengoptimalkan hasil melalui teknik *gradient boosting* yang iteratif. Studi sebelumnya menunjukkan XGBoost 532% lebih cepat dalam pemrosesan data, sedangkan Random Forest lebih stabil pada dataset tidak seimbang. Namun, keduanya memiliki akurasi serupa (~95%) dalam klasifikasi (Budaya et al., 2024).

Keluaran dari penelitian ini adalah model klasifikasi sentimen yang mampu membedakan ulasan ke dalam dua bagian, yakni ulasan positif dan negatif. Evaluasi kinerja model tersebut dilakukan melalui serangkaian indikator kinerja yang lazim digunakan dalam machine learning, seperti accuracy, precision, recall, dan F1-score. Akurasi tersebut mengukur keahlian model dalam mengkategorikan kumpulan data dengan tepat secara global. Precision dan recall digunakan untuk mengevaluasi kinerja model pada masing-masing kelas, terutama penting saat data tidak seimbang antar kelas. Di sisi lain, F1-score berperan sebagai sebuah indikator komposit yang menggabungkan nilai precision dan recall. Fungsi utamanya adalah untuk menghasilkan evaluasi kinerja model yang lebih stabil dan tidak bias dengan mencari titik tengah dari kedua metrik tersebut. (Atmajaya et al., 2023).

Penelitian ini bertujuan untuk meningkatkan kinerja analisis sentimen dengan membandingkan hasilnya terhadap metode yang telah digunakan pada studi sebelumnya. Pendekatan utama yang diterapkan adalah penggunaan algoritma

XGBoost, yang sudah dikenal efektif dalam berbagai bidang, terutama dalam tugas klasifikasi yang menggunakan pembelajaran mesin. Keunggulan XGBoost terletak pada kemampuannya menangani data dalam jumlah besar, mengurangi risiko overfitting melalui teknik regularisasi, serta memanfaatkan metode boosting untuk meningkatkan ketepatan model.

Selain fokus pada algoritma, penelitian ini juga menitikberatkan pada proses preprocessing dan ekstraksi fitur yang lebih teliti. Langkah-langkah preprocessing meliputi pembersihan data dari karakter yang tidak relevan, normalisasi teks, dan penghapusan stemming yang dianggap kurang berkontribusi pada hasil analisis sentimen. Dengan penerapan strategi ini, diharapkan model yang dibangun dapat mencapai akurasi dan efektivitas yang lebih unggul dibanding dengan penelitian terdahulu. Keseluruhan proses ini memberikan pondasi yang kokoh bagi pengembangan metode analisis sentimen yang lebih akurat dan handal dalam menghadapi ragam tantangan data yang kompleks.

1.2. Rumusan Masalah

Berdasarkan uraian latar belakang yang telah disampaikan, penelitian ini didasari oleh beberapa permasalahan utama, yaitu:

1. Bagaimana cara mengimplementasikan Algoritma Random Forest dan XGBoost untuk menganalisis sentimen aplikasi e-commerce shopee?
2. Bagaimana perbandingan hasil kinerja dari Algoritma Random Forest dan XGBoost dalam mengklasifikasi sentimen aplikasi e-commerce shopee?

1.3. Batasan Masalah

Untuk memastikan penelitian ini tetap terfokus dan tersusun dengan rapi, beberapa batasan ditetapkan sebagai pedoman pelaksanaan:

1. Data yang digunakan peneliti dalam penelitian ini berasal dari kumpulan review pengguna Shopee yang didapatkan dengan Google Colab, dengan total sebanyak 500 dataset melalui Google Play Store.
2. Penelitian ini hanya menggunakan ulasan yang ditulis dalam bahasa Indonesia.
3. Pengembangan dan penerapan algoritma dilakukan menggunakan bahasa pemrograman Python dengan Jupyter Notebook.
4. Sentimen dalam ulasan dibagi ke dalam dua kelompok, yakni positif dan negatif.
5. Algoritma yang akan dibandingkan dalam studi penelitian ini merupakan Algoritma Random Forest dan XGBoost.
6. Evaluasi model dilakukan dengan memanfaatkan confusion matrix guna mengukur performa berdasarkan metrik seperti akurasi, presisi, recall, dan F1-score.

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dibahas sebelumnya, penelitian ini difokuskan pada hal-hal berikut:

1. Mengimplementasikan algoritma Random Forest dan XGBoost untuk menganalisis sentimen aplikasi e-commerce Shopee.

2. Mengetahui perbandingan hasil kinerja dari Algoritma Random Forest dan XGBoost dalam mengklasifikasi sentimen aplikasi e-commerce Shopee.

1.5. Manfaat Penelitian

Temuan dari penelitian ini diharapkan bisa memberi guna sebagai berikut:

1. Bagi akademisi : Memperbanyak pemahaman mengenai keefektifan algoritma Random Forest dan XGBoost dalam analisis sentimen sekaligus menyediakan referensi untuk studi-studi yang akan datang.
2. Bagi pelaku industri e-commerce : Membantu dalam memahami pola sentimen pelanggan untuk meningkatkan kualitas layanan dan pengalaman pengguna.
3. Bagi pengguna umum : Mempermudah dalam mendapatkan insight terkait kepuasan pelanggan terhadap Shopee berdasarkan analisis ulasan.

BAB II
LANDASAN TEORI

2.1. Penelitian Terdahulu

Pada sebuah penelitian, penting untuk meninjau studi-studi sebelumnya guna memahami evolusi topik yang sedang diteliti serta mengidentifikasi celah dalam penelitian yang bisa menjadi dasar bagi penelitian ini. Dengan melakukan kajian terhadap penelitian sebelumnya, penelitian ini dapat lebih terfokus dan memiliki fondasi teoritis yang kokoh. Di bawah ini disajikan beberapa penelitian sebelumnya yang relevan dengan skripsi penulis, yang ditampilkan dalam **Tabel 2.1**

Tabel 2.1 Penelitian Terdahulu

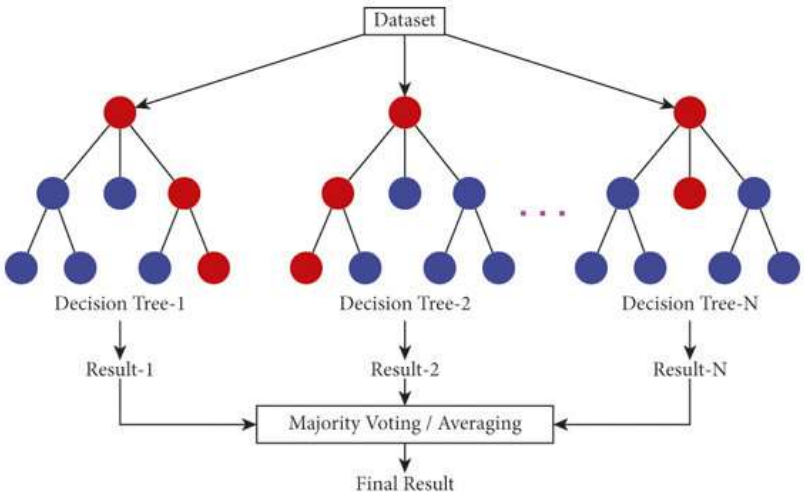
No	Nama Penulis	Judul	Tahun	Kesimpulan
1	Wardana, K. et al.	Analisis Perbandingan Algoritma XGBoost Dan Algoritma Random Forest Untuk Klasifikasi Data Kesehatan Mental	2024	Dari kedua model algoritma yang telah dikembangkan, terdapat hasil perbandingan klasifikasi kedua algoritma yaitu XGBoost dan Random Forest. XGBoost memiliki akurasi terbesar yaitu 99,82% dengan tingkat

				kesalahan yang begitu sedikit sedangkan Random Forest memiliki akurasi 99,04% dengan tingkat kesalahan yang lebih banyak.
2	Saepudin, A. et al.	Perbandingan Algoritma Klasifikasi SVM, Random Forest, dan Logistic Regression Pada Ulasan Shopee	2024	Dari ketiga model algoritma yang telah dikembangkan, terdapat hasil perbandingan klasifikasi yaitu SVM dengan akurasi 91%, lalu Random Forest dengan akurasi 94%, dan Logistic Regression dengan akurasi 86%.
3	Yulianti, R. et al.	Perbandingan Algoritma Random Forest dan XGBoost dalam Klasifikasi Penerima Bantuan Pangan Non-Tunai	2025	Dari perbandingan kedua model, diperoleh hasil klasifikasi untuk model Random Forest sebesar 80,01% dan XGBoost sebesar 74,04%. Berdasarkan

		(BPNT) di Provinsi Jawa Barat		hasil diatas, menunjukkan bahwa model Random Forest memiliki akurasi lebih besar dibandingkan XGBoost.

2.2. Random Forest

Random Forest merupakan salah satu model machine learning yang menggunakan kumpulan pohon keputusan guna membuat prediksi. Cara kerjanya adalah dengan membangun setiap pohon keputusan secara independen berdasarkan data acak dan subset fitur acak, lalu menggabungkan hasil voting dari semua pohon tersebut untuk menentukan hasil akhir prediksi. Metode ini efektif untuk meningkatkan akurasi dan mengurangi risiko model terlalu menyesuaikan diri dengan data pelatihan (overfitting) (Kurniawan et al., 2024).



Gambar 2.1 Konsep Kerja Random Forest

Setiap pohon memberikan prediksi sendiri-sendiri dan hasilnya ditentukan berdasarkan suara mayoritas (untuk klasifikasi) dan rata-rata (untuk regresi) dari semua pohon yang ada dalam "hutan" tersebut. Pendekatan ini dikenal dengan istilah ensemble learning, yang meningkatkan akurasi dan kestabilan prediksi dibanding menggunakan satu pohon keputusan tunggal.

Proses pembuatan setiap pohon di Random Forest melibatkan pemilihan sampel data dengan metode bootstrap (pengambilan sampel dengan penggantian) dan pemilihan fitur secara acak untuk membangun pohon. Cara ini membantu mengurangi overfitting dan menghasilkan model yang lebih generalisasi terhadap data baru.

Kelebihan Random Forest adalah kemampuannya untuk bekerja dengan baik pada data yang besar dan kompleks, serta tahan terhadap noise dan data yang hilang. Algoritma ini banyak digunakan di berbagai bidang seperti perbankan, kesehatan, dan e-commerce untuk prediksi yang akurat dan dapat diandalkan. Jadi secara singkat, Random Forest merupakan kumpulan sejumlah decision tree yang digabungkan dalam satu kerangka kerja untuk meningkatkan ketepatan dan stabilitas prediksi dengan cara melibatkan variasi acak di dalam proses pembuatannya.

2.3. XGBoost

XGBoost (Extreme Gradient Boosting) merupakan algoritma boosting berbasis gradient yang dikembangkan untuk meningkatkan performa model pembelajaran mesin, terutama dalam kompetisi data science. Algoritma ini mengoptimalkan pohon keputusan secara bertahap dengan memperbaiki kesalahan prediksi pada iterasi sebelumnya (Srinivas et al., 2021).

Secara sederhana, XGBoost memanfaatkan metode boosting, yaitu proses menambah model-model kecil yang lemah sehingga menjadi satu model yang kuat dan mampu mengenali pola kompleks dalam data, baik untuk tugas klasifikasi maupun regresi. Algoritma ini sangat cepat dan mampu menangani dataset besar berkat optimasi komputasi dan paralelisme yang dimilikinya.

Selain kecepatan, XGBoost juga memiliki fitur penting seperti regularisasi yang membantu mencegah model overfitting, otomatisasi dalam pemilihan fitur, serta kemampuannya dalam menangani data yang mengandung nilai hilang secara otomatis. Pendekatan ini membuat XGBoost sangat populer digunakan dalam kompetisi data science dan berbagai aplikasi industri, termasuk prediksi harga, deteksi anomali, dan analisis data besar lainnya. Jadi secara singkat, XGBoost merupakan algoritma boosting yang sangat efisien dan powerful, mampu memberikan prediksi berkualitas tinggi dan digunakan secara luas karena kestabilan, kecepatan, dan fleksibilitasnya.

2.4. Analisis Sentimen

Analisis sentimen berfungsi dalam mengenali dan mengelompokkan opini, perasaan, maupun ekspresi emosional yang tersirat dalam suatu data teks, baik berupa ulasan produk, komentar di media sosial, maupun konten berita. Esensi utama dari penerapan teknik ini adalah untuk mengetahui pandangan pengguna terhadap suatu topik tertentu, seperti merek dagang, layanan, atau kebijakan, sehingga dapat membantu dalam pengambilan keputusan atau peningkatan kualitas layanan. Analisis ini banyak digunakan dalam bidang pemasaran, politik, dan layanan pelanggan untuk mengukur respons publik terhadap suatu produk atau kebijakan (Mahawardana et al., 2022).

Dalam praktiknya, terdapat berbagai jenis analisis sentimen, yaitu pandangan yang mendukung (positif), menolak (negatif), dan tidak memihak (netral). Selain itu, ada juga analisis sentimen berbasis aspek, yang berfokus pada elemen spesifik dalam sebuah ulasan, misalnya menilai aspek pelayanan pelanggan atau kualitas produk. Sementara itu, analisis sentimen berbasis emosi lebih lanjut mengkategorikan opini berdasarkan spektrum emosi, seperti marah, bahagia, atau kecewa.

2.5. Shopee

Shopee adalah sebuah platform e-commerce yang menghubungkan penjual dengan pembeli secara online. Diluncurkan pada tahun 2015, Shopee berkembang pesat di kawasan Asia Tenggara, termasuk Indonesia, dengan menawarkan spektrum komoditas yang luas, mulai dari gawai elektronik dan pakaian, perabot rumah tangga, sampai pada aneka fasilitas jasa penunjang.. Shopee memudahkan pengguna untuk berbelanja melalui aplikasi mobile dengan berbagai fitur seperti diskon, promo, dan layanan pengiriman yang cepat dan aman. Platform ini juga mendukung interaksi langsung antara penjual dan pembeli sehingga pengalaman belanja menjadi lebih menyenangkan dan terpercaya (Fradesa et al., 2022).

Gambar 2.2 Logo Shopee

2.6. Jupyter Notebook

Jupyter Notebook adalah platform komputasi interaktif berbasis web yang memfasilitasi penulisan dan eksekusi kode program secara real-time. Tool ini banyak diadopsi oleh praktisi data science dan programmer karena kemampuannya mengintegrasikan segmen kode, teks penjelasan, visualisasi data, dan persamaan matematika dalam satu dokumen yang mudah dibaca dan dibagikan. Dengan fitur interaktifnya, pengguna dapat langsung melihat hasil dari kode yang dijalankan, sehingga proses pengembangan dan analisis data menjadi lebih efisien dan transparan.

Gambar 2.3 Logo Jupyter Notebook

Salah satu aspek penting dari Jupyter Notebook adalah kemampuannya dalam mendukung berbagai bahasa pemrograman seperti Python, R, dan Julia melalui fitur kernel yang fleksibel. Hal ini membuat Jupyter Notebook sangat berguna untuk berbagai keperluan mulai dari eksplorasi data, pembuatan model machine learning, hingga penyusunan laporan dan presentasi hasil penelitian. Teks dapat ditulis menggunakan format Markdown, sehingga penjelasan dan dokumentasi menjadi lebih terstruktur dan mudah dipahami.

Selain itu, Jupyter Notebook menyediakan banyak fitur yang membantu proses pengkodean seperti kemampuan untuk memecah kode dalam bentuk sel-sel yang bisa dijalankan satu per satu, memudahkan pengorganisasian dan debugging kode. Pengguna juga dapat mengekspor dokumen ini ke berbagai format lain seperti HTML, PDF, dan slide presentasi. Dengan kemudahan ini, Jupyter Notebook telah menjadi alat standar yang banyak digunakan dalam bidang data science, pembelajaran mesin, dan riset ilmiah.

2.7. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) adalah sebuah teknik pembobotan statistik yang bertujuan untuk mengukur tingkat kepentingan relatif sebuah kata dalam suatu dokumen dibandingkan dengan seluruh koleksi dokumen. Pendekatan ini banyak digunakan dalam pengolahan teks, khususnya pada bidang information retrieval dan natural language processing (Jalilifard et al., 2021).

Komponen Term Frequency (TF) bertugas menghitung seberapa sering suatu istilah muncul dalam sebuah dokumen tunggal, dengan asumsi bahwa kata dengan frekuensi tinggi mencerminkan konten dokumen tersebut. Sementara itu, Inverse Document Frequency (IDF) mengukur tingkat kelangkaan kata tersebut di seluruh korpus, di mana kata yang terlalu umum dan muncul di banyak dokumen akan diberi nilai penting yang lebih rendah.

Dengan menggabungkan kedua komponen ini, TF-IDF memberikan nilai bobot yang besar pada kata-kata yang sering muncul dalam satu dokumen tetapi jarang ditemukan di dokumen lain. Pendekatan ini efektif untuk menyoroti kata-kata penting atau istilah khusus dalam teks, dan secara luas digunakan dalam

pengolahan bahasa alami serta penambangan teks, contohnya untuk pencarian informasi, klasifikasi dokumen, dan analisis topik.

Secara sederhana, TF-IDF membantu menyaring kata-kata umum yang tidak membawa banyak informasi, sehingga fokus tertuju pada kata-kata yang memberikan arti khusus dalam suatu dokumen. Rumus utama TF-IDF adalah sebagai berikut:

$$w = TF \times IDF \tag{2. 3}$$

$$w = f_{t,d} \times log \frac{D}{df(t)} \tag{2. 4}$$

Keterangan :

W = weight/bobot

$TF = f_{t,d}$ = nilai term frequency atau jumlah kata t yang muncul pada dokumen d.

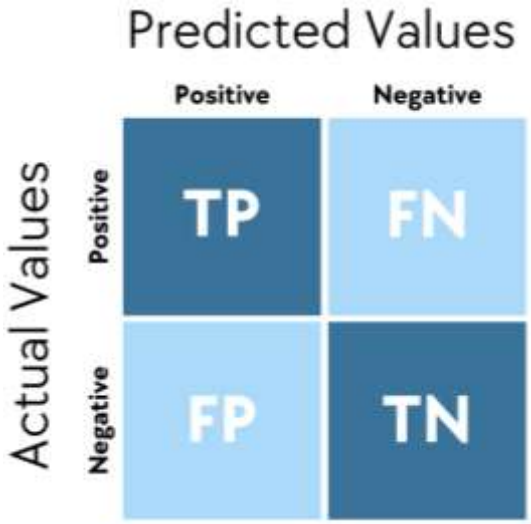
IDF = nilai inverse document frequency

D = total dokumen atau jumlah kalimat

$df(t)$ = jumlah dokumen yang di dalamnya terdapat kata t

2.8. Confusion Matrix

Confusion Matrix berfungsi sebagai alat evaluasi performa model klasifikasi yang disajikan dalam format tabel. Matriks ini mengelompokkan hasil prediksi ke dalam empat kuadran utama: True Positive (TP) yang merepresentasikan prediksi benar untuk kelas positif, True Negative (TN) yang menunjukkan prediksi benar untuk kelas negatif, False Positive (FP) yang mencerminkan kesalahan prediksi positif, dan False Negative (FN) yang mengindikasikan kesalahan prediksi negatif (Siregar et al., 2023).



Gambar 2.4 Confusion Matrix

Confusion Matrix sangat berguna dalam menganalisis performa model dalam mengklasifikasikan data, terutama ketika terdapat ketidakseimbangan kelas. Dengan menggunakan Confusion Matrix, berbagai metrik evaluasi, yakni akurasi, presisi, recall, dan F1-score bisa dihitung untuk mendapatkan pemahaman yang lebih mendalam terkait kinerja model.

Berikut merupakan pemaparan dari masing-masing bagian dalam Confusion Matrix:

1. True Positive (TP): dimana jumlah data positif yang diklasifikasikan dengan benar sebagai positif.
2. True Negative (TN): dimana jumlah data negatif yang diklasifikasikan dengan benar sebagai negatif.
3. False Positive (FP): dimana jumlah data negatif yang salah diklasifikasikan sebagai positif (*Type I Error*).
4. False Negative (FN): dimana jumlah data positif yang salah diklasifikasikan sebagai negatif (*Type II Error*).

Komponen tersebut dapat digunakan unruk menghitung berbagai metrik evaluasi untuk mengukur performa model klasifikasi yaitu *accuracy, precision, recall,* dan F1 *score*.

Accuracy merupakan nilai hasil perbandingan dari jumlah data yang diprediksi benar dengan total seluruh data. *Accuracy* digunakan untuk mengukur tingkat keakuratan model algoritma dalam ketepatan hasil klasifikasi. Secara matematis, *accuracy* dapat dijabarkan sebagai berikut.

$$Accuracy = \frac{Jumlah\ prediksi\ benar}{Total\ Seluruh\ Data} \quad (2.\ 5)$$

Precision merupakan nilai hasil perbandingan dari proporsi sampel yang terprediksi sebagai positif benar dibandingkan dengan keseluruhan instansi yang diklasifikasikan sebagai positif. *Precision* digunakan untuk mengukur seberapa sering prediksi itu benar ketika model algoritma memprediksi positif. Secara matematis, *precision* dapat dijabarkan sebagai berikut.

$$Precision = \frac{TP}{TP + FP} \quad (2.\ 6)$$

Recall merupakan nilai hasil perbandingan dari jumlah data yang diprediksi benar positif dengan total seluruh data yang diprediksi benar. *Recall* digunakan untuk mengukur seberapa sering prediksi itu positif ketika model algoritma memprediksi benar. Secara matematis, *recall* dapat dijabarkan sebagai berikut.

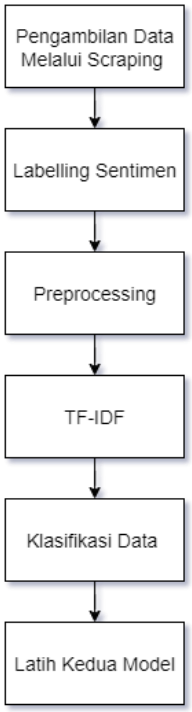
$$Recall = \frac{TP}{TP + FN} \quad (2.\ 7)$$

F1 *score* digunakan untuk mengukur nilai rata-rata harmonik dari nilai *precision* dan *recall.* Secara matematis, F1 *score* dapat dijabarkan sebagai berikut.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.\ 8)$$

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang sistematis untuk membangun model klasifikasi sentimen berbasis machine learning. Setiap tahap memiliki peran penting dalam memastikan kualitas data serta meningkatkan akurasi model yang digunakan.

Tahap pertama adalah pengumpulan data, di mana ulasan dari platform e-commerce dikumpulkan melalui web scraping dan diberi label sesuai dengan sentimen yang dikandungnya, yaitu positif, netral, dan negatif. Proses pelabelan dilakukan dengan script oleh peneliti.

Setelah data terkumpul, dilakukan preprocessing guna menghilangkan komponen non-esensial dari data teks. Tahapan ini meliputi normalisasi huruf (*case-folding*) dengan mengonversi keseluruhan karakter menjadi format leksikal non-kapital, dilanjutkan dengan pembersihan data (*cleaning*) yang bertujuan menghilangkan simbol khusus dan elemen tanda baca, *tokenizing* (melakukan segmentasi teks menjadi unit-unit leksikal), filtrasi stopword (menyaring kata-kata fungsional yang tidak bermakna substantif), serta reduksi morfologis (*stemming*) untuk mengembalikan setiap istilah ke bentuk akarnya.

Data yang telah diproses kemudian diekstraksi fiturnya menggunakan Term Frequency-Inverse Document Frequency (TF-IDF). Metode ini mengubah teks menjadi representasi numerik dengan memberi bobot lebih tinggi pada kata-kata yang lebih penting dalam analisis sentimen.

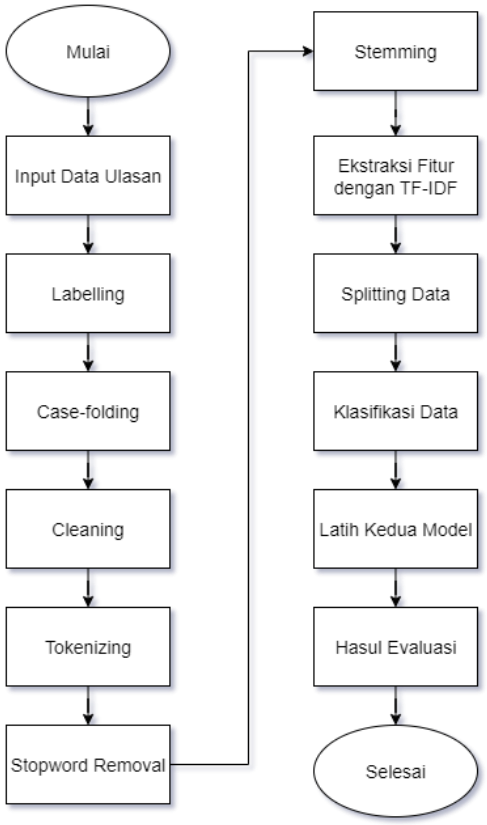
Pasca ekstraksi fitur, dataset dipartisi menjadi dua subset: data latih (training set) untuk pembentukan model dan data uji (testing set) untuk validasi kinerja model. Partisi ini diimplementasikan secara acak guna menjamin distribusi sampel yang representatif.

Dua arsitektur machine learning yang diadopsi dalam studi ini adalah Random Forest dan XGBoost, yang secara empiris terbukti unggul dalam menyelesaikan tugas klasifikasi. Kedua model tersebut melalui fase pelatihan menggunakan subset latih sebelum dievaluasi pada subset uji untuk menghasilkan klasifikasi sentimen.

Fase final penelitian melibatkan penilaian kinerja model melalui implementasi Confusion Matrix dan serangkaian indikator kuantitatif mencakup akurasi, presisi, recall, serta F1-score. Proses penilaian ini bertujuan mengukur

tingkat akurasi model dalam mengelompokkan opini dari feedback pengguna. Visualisasi pada Gambar 3.1 mengilustrasikan alur metodologis yang diterapkan dalam studi ini, sementara uraian detail setiap tahapan akan dipaparkan secara komprehensif dalam subbab-subbab berikutnya.

3.2 Flowchart Penerapan Kedua Algoritma



Gambar 3.2 Flowchart Penerapan Kedua Algoritma

1. Mulai : Penerapan dimulai dengan klik command prompt, lalu ketik “pip install jupyter”, setelah itu install “jupyter notebook” dan jalankan dengan jupyter notebook.
2. Input Data Ulasan : Tambahkan data ulasan ke direktori.
3. Labelling : Setelah data ulasan muncul di direktori, jalankan script perintah untuk memberi label sentimen pada data ulasan.

4. Case-folding : Setelah data labelling tersimpan, tambah kode baru untuk menjalankan script case-folding.
5. Cleaning : Setelah data case-folding tersimpan, jalankan script perintah untuk proses cleaning.
6. Tokenizing : Setelah data cleaning tersimpan, jalankan script perintah untuk proses tokenizing.
7. Stopword Removal : Setelah data tokenizing tersimpan, jalankan script perintah untuk proses stopword removal.
8. Stemming : Setelah data stopword removal tersimpan, jalankan script perintah untuk proses stemming.
9. Ekstraksi Fitur TF-IDF : Setelah data stemming tersimpan, tambah kode baru untuk menjalankan script TF-IDF.
10. Splitting Data : Setelah data TF-IDF tersimpan, tambah kode baru untuk menjalankan script splitting data.
11. Klasifikasi Data : Setelah data splitting tersimpan, tambah kode baru untuk menjalankan script klasifikasi data.
12. Latih Kedua Model : Tambah kode baru untuk menjalankan script melatih model Random Forest dan XGBoost
13. Hasil Evaluasi : Tambah kode baru untuk menjalankan script hasil evaluasi, dimana semua evaluasi akan muncul seperti confusion matrix, classification report, pie chart, dan word cloud. Pastikan hasil evaluasi disimpan.
14. Selesai : Penerapan algoritma berakhir yang menunjukkan bahwa proses telah selesai.

3.3 Pengumpulan Data

Akuisisi data menjadi fondasi utama yang menentukan dalam studi ini. Integritas data yang diperoleh akan berdampak signifikan terhadap presisi dan keandalan model analisis sentimen yang dibangun. Pada penelitian ini, sumber data yang diambil terdiri dari kumpulan tanggapan pengguna terhadap aplikasi Shopee yang tersedia di platform Google Play Store. Data opini ini dimanfaatkan untuk mengevaluasi persepsi pengguna terhadap aplikasi, mencakup kategorisasi respons positif, netral, maupun negatif.

Untuk memperoleh data review dari Google Play Store, diterapkan teknik web scraping sebagai mekanisme otomatis dalam mengekstrak konten langsung dari situs web. Proses ekstraksi data ini diimplementasikan menggunakan Python dengan memanfaatkan library Google Play Scraper yang memfasilitasi pengambilan data ulasan berdasarkan berbagai kriteria seperti volume review, bahasa, peringkat, dan parameter pendukung lainnya.Setelah data dikumpulkan, tahap selanjutnya adalah memberikan label sentimen pada masing-masing ulasan. Pelabelan dilakukan secara manual dan otomatis berdasarkan rating bintang yang diberikan pengguna:

- Sentimen Positif : Jika rating $\geq$ 4 bintang, ulasan dianggap memiliki sentimen positif.
- Sentimen Netral : Jika rating = 3 bintang, ulasan dikategorikan netral.
- Sentimen Negatif : Jika rating $\leq$ 2 bintang, ulasan dianggap memiliki sentimen negatif.

Selain berdasarkan rating, peneliti juga melakukan pengecekan manual terhadap beberapa ulasan untuk memastikan bahwa sentimen sesuai dengan isi

teksnya. Hal ini dilakukan untuk menghindari kesalahan label, terutama pada ulasan yang memiliki rating tinggi tetapi mengandung keluhan, atau ulasan dengan rating rendah tetapi berisi komentar positif.

3.4 Preprocessing Data

Tahap selanjutnya setelah tahap pengumpulan data kemudian memasuki tahap pra-pemrosesan. Intinya, tahapan ini berfungsi mempersiapkan dan mengondisikan data sehingga memenuhi kriteria kelayakan untuk proses-proses analitis berikutnya. Salah satu langkah yang dilakukan adalah mengubah semua teks menjadi huruf non-kapital agar tidak ada perbedaan antara kata yang sama tetapi berbeda dalam kapitalisasi. Kemudian melakukan proses cleaning, yaitu pembersihan teks dari karakter atau simbol yang tidak esensial, meliputi elemen-elemen seperti simbol punctuasi, digit numerik, serta karakter non-standar yang tidak memberikan kontribusi signifikan terhadap proses penambangan opini. Tahap selanjutnya adalah tokenizing, yaitu memisahkan kalimat menjadi satuan kata..

Tahap berikutnya adalah penghapusan stopword, yang berfungsi menyaring kata-kata fungsional yang tidak memiliki makna substantif. Proses diakhiri dengan stemming, yaitu mereduksi kata berimbuhan menjadi bentuk dasar leksikalnya. Sebagai contoh, variasi kata seperti "memakan" dan "dimakan" akan dikembalikan ke bentuk akar "makan". Dengan dilakukan tahap pra pemrosesan ini, ulasan yang digunakan dalam penelitian menjadi lebih tertata, bersih, dan siap untuk diolah lebih mendalam pada tahap ekstraksi fitur dan klasifikasi sentimen (Yuyun et al., 2023).

3.4.1. Case-folding

Tabel 3.1 Contoh Case-folding

Sebelum proses *case-folding*	Setelah proses *case-folding*
Pelayanan ... Memberi panduan disaat salah input nomor pembayaran ðŸ‘□ðŸ□¼ðŸ‘□ðŸ□¼ðŸ‘□ðŸ□¼	pelayanan ... memberi panduan disaat salah input nomor pembayaran ðŸ‘□ðŸ□¼ðŸ‘□ðŸ□¼ðŸ‘□ðŸ□¼

Dalam proses normalisasi huruf (case-folding), seluruh karakter teks dikonversi menjadi format leksikal non-kapital secara konsisten. Kata "Pelayanan" dan "Memberi" pada contoh di atas yang huruf awalnya kapital, setelah proses case-folding ini akan berubah menjadi huruf kecil, yaitu "pelayanan" dan "memberi".

3.4.2. Cleaning

Tabel 3.1 Contoh Cleaning

Sebelum proses *cleaning*	Setelah proses *cleaning*
pelayanan ... memberi panduan disaat salah input nomor pembayaran ðŸ‘□ðŸ□¼ðŸ‘□ðŸ	pelayanan ramah memberi panduan disaat salah input nomor pembayaran

Pada tahap cleaning ini akan dilakukan penghapusan kata dari elemen-elemen non-esensial seperti digit numerik, simbol punctuasi, karakter khusus, serta spasi yang redundan. Karakter ðŸ‘□¼ pada contoh di atas akan hilang setelah proses cleaning dilakukan.

3.4.3. Tokenizing

Tabel 3.2 Contoh Tokenizing

Sebelum proses *tokenizing*	Setelah proses *tokenizing*
pelayanan ramah memberi panduan disaat salah input nomor pembayaran	pelayanan
	memberi
	panduan
	disaat
	salah
	input
	nomor
	pembayaran

Pada tahap tokenizing ini akan dilakukan pemotongan kalimat agar terpisah-pisah menjadi satuan kata. Contoh kalimat pada tabel di atas "pelayanan ramah memberi panduan disaat salah input nomor pembayaran " akan dipisah-pisah menjadi satuan kata, yaitu pelayanan, ramah, , memberi, panduan, disaat, salah, input, nomor, dan pembayaran .

3.4.4. Stopword Removal

Tabel 3.4 Contoh Stopword Removal

Sebelum proses *stopword removal*	Setelah proses *stopword removal*
pelayanan	pelayanan
ramah	ramah
memberi	memberi
panduan	panduan
disaat	
salah	salah
input	input
nomor	nomor
pembayaran	pembayaran

Pada langkah *stopword removal* ini akan dilakukan penghilangan kata-kata pada kalimat yang tidak perlu dalam tahap pemrosesan olah data. Seperti kata hubung, kata depan, akan dibuang pada tahap ini. Contoh pada tabel di atas ialah kata “disaat” yang akan dihilangkan pada proses *stopword removal* ini. Mengapa demikian? Hal ini dikarenakan kata disaat sama maknanya dengan kata “ketika” yang merupakan kata hubung dan harus dihapuskan.

3.4.5. Stemming

Tabel 3. 5 Contoh Stemming

Sebelum proses *stemming*	**Setelah proses *stemming***
pelayanan	layan
ramah	ramah
memberi	beri
panduan	pandu
salah	salah
input	input
nomor	nomor
pembayaran	bayar

Dalam tahap stemming ini akan dilakukan mengubah bentuk pada kata-kata yang mempunyai imbuhan menjadi bentuk dasar. Contoh pada tabel di atas ialah kata “pelayanan” yang mempunyai awalan pe- dan akhiran -an yang setelah proses stemming ini akan berubah bentuknya menjadi kata dasar yaitu “layan”. Begitu pula dengan kata “memberi”, “panduan”, Kata-kata tersebut memiliki imbuhan sehingga akan diubah bentuknya menjadi kata dasar yaitu “beri” dan “pandu”.

3.5 Fitur Ekstraksi dengan TF-IDF

Sesudah melewati langkah pengumpulan dan pra-pemrosesan data, proses analisis dilanjutkan dengan ekstraksi fitur menerapkan pendekatan TF-IDF yang berfungsi sebagai pendekatan statistik dalam pengolahan teks guna mentransformasikan data tekstual menjadi representasi numerik. Metode ini menghitung nilai penting setiap kata dalam sebuah dokumen dengan mempertimbangkan dua aspek: seringnya muncul kata pada dokumen tertentu dan tingkat kelangkaan leksikal tersebut dalam keseluruhan korpus. Prinsip utamanya adalah leksikal yang sering muncul pada dokumen spesifik tapi jarang ditemui pada dokumen lainnya akan memperoleh bobor yang lebih tinggi, menandai signifikansinya dalam dokumen tersebut.

Proses ekstraksi fitur dengan TF-IDF diawali dengan tokenisasi, yaitu memecah teks menjadi kata-kata individu. Setelah itu, setiap kata diberikan bobot berdasarkan frekuensi kemunculannya di dalam dokumen tertentu serta relevansinya dibandingkan dengan dokumen lain dalam dataset. Proses ini dilakukan secara otomatis menggunakan pustaka pemrosesan teks, seperti scikit-learn, yang mampu mengonversi teks menjadi vektor numerik.

Penerapan TF-IDF dalam penelitian ini bertujuan untuk menangkap makna dari ulasan pengguna dengan lebih efektif dibandingkan dengan metode sederhana seperti bag-of-words. Dengan representasi berbasis bobot ini, model machine learning bisa lebih mudah mengetahui pola yang muncul pada data. Setelah fitur berhasil diekstrak, data yang telah dikonversi menjadi bentuk numerik siap digunakan dalam tahap splitting data untuk pelatihan model klasifikasi.

3.6 Splitting Data

Usai ekstraksi ciri teks menggunakan TF-IDF, tahap berikutnya adalah mempartisi dataset menjadi dua kelompok terpisah: data pelatihan dan data pengujian. Data pelatihan berperan dalam pembentukan model klasifikasi, sedangkan data pengujian berfungsi sebagai alat validasi kinerja model kemampuan algoritma dalam memprediksi sampel data baru yang belum dikenali.

Pada studi ini, partisi data diterapkan dengan teknik stratified splitting. Metode ini menjamin proporsi distribusi setiap kategori sentimen tetap proporsional antara subset latih dan uji, sehingga mencegah bias dalam evaluasi model. Rasio umum yang digunakan dalam pembagian data adalah 70:30 atau 80:20, dengan mayoritas data digunakan untuk pelatihan model. Pemilihan rasio ini bertujuan untuk memberikan model cukup banyak contoh untuk belajar, tetapi tetap menyisakan sejumlah data yang cukup untuk mengukur akurasi model secara objektif.

Proses pembagian data ini sangat penting karena berpengaruh terhadap performa model. Jika data yang digunakan untuk pelatihan terlalu sedikit, model bisa kurang mampu mengenali pola yang ada, sedangkan jika terlalu banyak apabila proporsi data yang dialokasikan untuk pelatihan jauh lebih dominan dibandingkan untuk pengujian, maka akurasi penilaian kinerja model bisa menjadi kurang representatif. Oleh karena itu, pemilihan rasio pembagian yang optimal menjadi salah satu pertimbangan utama dalam eksperimen ini.

3.7 Klasifikasi Data

Tahap berikutnya dalam penelitian ini adalah proses klasifikasi data, di mana representasi numerik dari ulasan akan diolah menggunakan algoritma machine learning guna mengidentifikasi sentimen setiap ulasan. Penelitian ini

mengelompokkan data ke dalam tiga kategori sentimen, yakni positif (dikodekan sebagai 2), netral (1), dan negatif (0).

Untuk keperluan klasifikasi, dua algoritma utama yang digunakan adalah Random Forest dan XGBoost. Random Forest merupakan algoritma ensemble yang mengandalkan kumpulan pohon keputusan. Cara kerjanya adalah dengan membangun banyak decision tree dan mengagregasikan hasilnya guna meningkatkan ketepatan prediksi. Kelebihan utama metode ini terletak pada kemampuannya mengolah data kompleks sekaligus meminimalkan overfitting. Di sisi lain, XGBoost adalah algoritma gradient boosting yang dikenal lebih efisien dan mampu memproses data berskala besar dengan akurasi yang unggul.

3.8 Melatih Kedua Algoritma

Setelah proses klasifikasi selesai, tahap berikutnya adalah mengevaluasi akurasi model dalam mengkategorikan data dengan sempurna. Penilaian ini berguna sebagai tolak ukur sejauh mana model ini menandai pola pada suatu data dan melakukan kategorisasi sentimen dengan tingkat ketepatan yang optimal. Studi ini menerapkan kerangka evaluasi yang memanfaatkan empat indikator kinerja utama, meliputi accuracy, precision, recall, dan F1-score.

3.9 Jadwal Penelitian

Tabel 3.6 Jadwal Penelitian

No	Kegiatan Penelitian	Tahun 2025						
		Januari	Februari	Maret	April	Mei	Juni	Juli
1	Pengajuan Judul							
2	BAB I							
3	BAB II							
4	BAB III							
5	Seminar Proposal							
6	BAB IV							
7	BAB V							
8	Sidang							

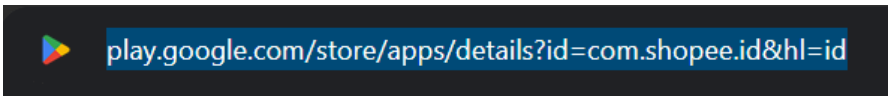
BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Pengambilan Data

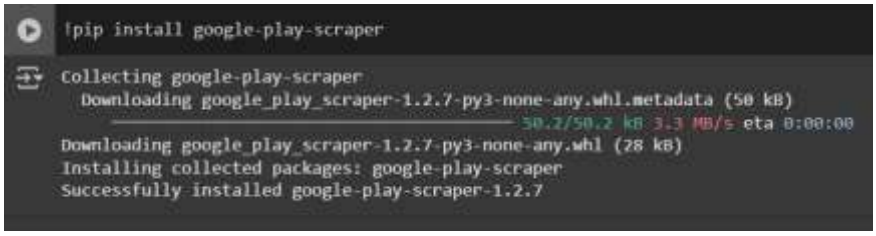
Pengambilan data ulasan menggunakan teknik *scraping* dengan Google Colab untuk mendapatkan komentar terkait dengan aplikasi Shopee di Google Playstore sebanyak 500 dataset. Berikut tahapannya:

a. Ambil URL langsung di Playstore



Gambar 4.1 URL shopee

b. Gunakan Library google_play_scraper



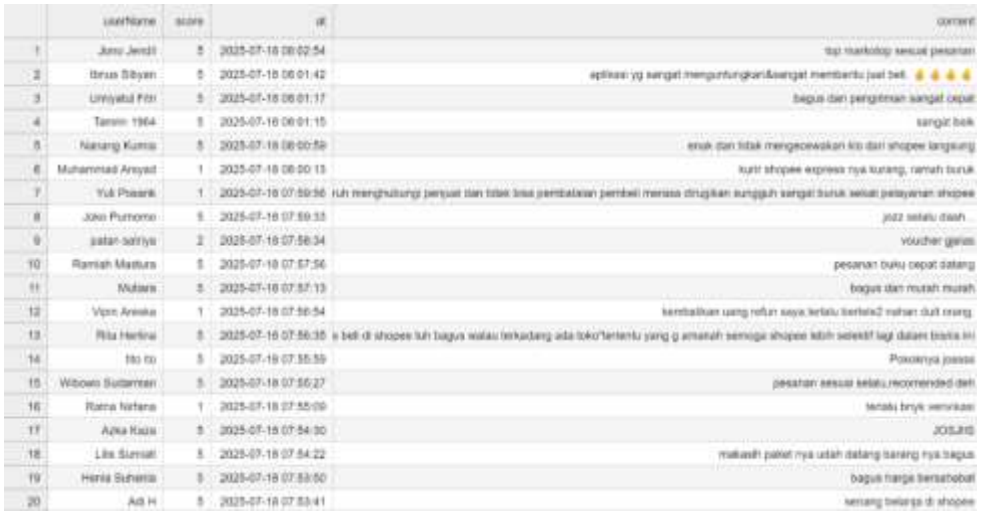
Gambar 4.2 Library google_play_scraper

c. Script ambil komentar

```
from google_play_scraper import app
import pandas as pd
import numpy as np

from google_play_scraper import Sort, reviews

result, continuation_token = reviews(
    'com.shopee.id',
    lang='id',  #bahasa
    country='id',  #negara
    sort=Sort.NEWEST,
    count=500,
    filter_score_with=None
)

df_busu = pd.DataFrame(np.array(result),columns=['review'])
df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
df_busu.head()

len(df_busu.index)
df_busu[['userName', 'score', 'at', 'content']].head()
my_df = df_busu[['userName', 'score', 'at', 'content']]
my_df.to_csv("ulasan_500.csv", index = False)
```

Gambar 4.3 Script Ambil Komentar

d. Data ulasan yang telah tersimpan



	userName	score	at	content
1	Jono Jendil	5	2025-07-18 08:02:54	top markotop sesuai pesanan
2	Ibnus Sibyan	5	2025-07-18 08:01:42	aplikasi yg sangat menguntungkan&sangat membantu jual beli. 🔥🔥🔥🔥
3	Umiyatul Fitri	5	2025-07-18 08:01:17	bagus dan pengiriman sangat cepat
4	Tamrin 1964	5	2025-07-18 08:01:15	sangat baik
5	Nanang Kurnia	5	2025-07-18 08:00:59	enak dan tidak mengecewakan klo dari shopee langsung
6	Muhammad Arsyad	1	2025-07-18 08:00:13	kurir shopee express nya kurang, ramah buruk
7	Yuli Posarik	1	2025-07-18 07:59:56	ruh menghutung penjual dan tidak bisa pembatalan pembeli merasa dirugikan sungguh sangat buruk sekali pelayanan shopee
8	Joko Purnomo	5	2025-07-18 07:59:33	jozz selalu dash ..
9	patan sahriya	2	2025-07-18 07:58:34	voucher gelas
10	Ramlah Mastura	5	2025-07-18 07:57:56	pesanan buku cepat datang
11	Mutiara	5	2025-07-18 07:57:13	bagus dan murah murah
12	Vipin Areska	1	2025-07-18 07:56:54	kembalikan uang refun saya,terlalu bertele2 nahan duit orang
13	Rita Hertina	5	2025-07-18 07:56:35	a beli di shopee tuh bagus walau terkadang ada toko"tertentu yang g amanah semoga shopee lebih selektif lagi dalam bisnis ini
14	Ito Ito	5	2025-07-18 07:55:59	Pokoknya josss
15	Wibowo Sudarman	5	2025-07-18 07:55:27	pesanan sesuai selalu,recomended deh
16	Ratna Nirfana	1	2025-07-18 07:55:09	terlalu bnyk vervikasi
17	Azka Kaza	5	2025-07-18 07:54:30	JOSJIIS
18	Lilis Sumiati	5	2025-07-18 07:54:22	makasih paket nya udah datang barang nya bagus
19	Henia Suhenia	5	2025-07-18 07:53:50	bagus harga bersahabat
20	Adi H	5	2025-07-18 07:53:41	senang belanja di shopee

Gambar 4.4 Data Ulasan

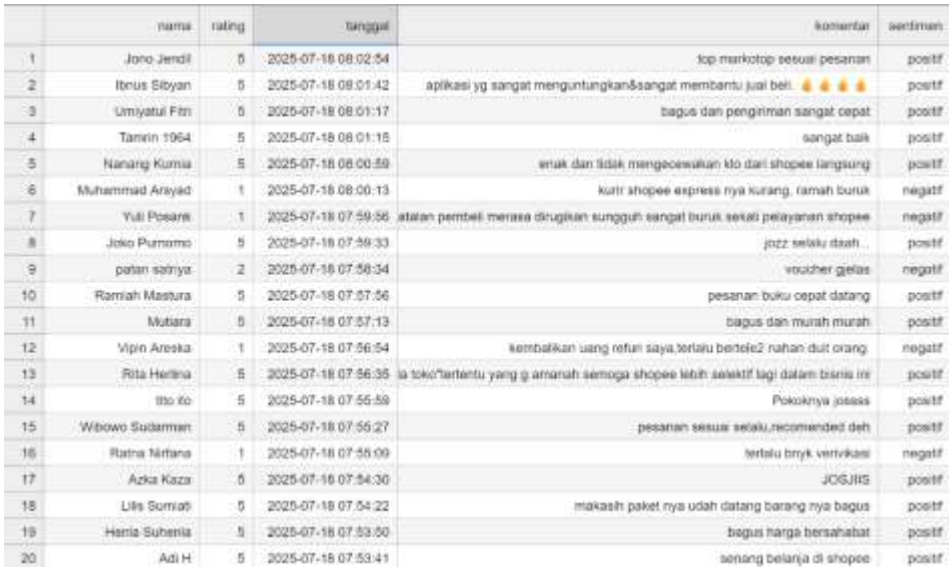
4.1.2 Hasil Labelling Sentimen

Labelling sentimen dilakukan berdasarkan rating bintang yang berikan pengguna dengan ketentuan, dimana:

a. Sentimen Positif: Jika rating ≥ 4 bintang, ulasan dikategorikan positif.

b. Sentimen Netral: Jika rating = 3 bintang, ulasan dikategorikan netral.

c. Sentimen Negatif: Jika rating ≤ 2 bintang, ulasan dikategorikan negatif.



	nama	rating	tanggal	komentar	sentimen
1	Jono Jendil	5	2025-07-18 08:02:54	top markotop sesuai pesanan	positif
2	Ibnus Sibyan	5	2025-07-18 08:01:42	aplikasi yg sangat menguntungkan&sangat membantu jual beli. 🔥🔥🔥🔥	positif
3	Umiyatul Fitri	5	2025-07-18 08:01:17	bagus dan pengiriman sangat cepat	positif
4	Tamrin 1964	5	2025-07-18 08:01:15	sangat baik	positif
5	Nanang Kurnia	5	2025-07-18 08:00:59	enak dan tidak mengecewakan klo dari shopee langsung	positif
6	Muhammad Arsyad	1	2025-07-18 08:00:13	kurir shopee express nya kurang, ramah buruk	negatif
7	Yuli Posarik	1	2025-07-18 07:59:56	atatan pembeli merasa dirugikan sungguh sangat buruk sekali pelayanan shopee	negatif
8	Joko Purnomo	5	2025-07-18 07:59:33	jozz selalu dash ..	positif
9	patan sahriya	2	2025-07-18 07:58:34	voucher gelas	negatif
10	Ramlah Mastura	5	2025-07-18 07:57:56	pesanan buku cepat datang	positif
11	Mutiara	5	2025-07-18 07:57:13	bagus dan murah murah	positif
12	Vipin Areska	1	2025-07-18 07:56:54	kembalikan uang refun saya,terlalu bertele2 nahan duit orang	negatif
13	Rita Hertina	5	2025-07-18 07:56:35	a toko"tertentu yang g amanah semoga shopee lebih selektif lagi dalam bisnis ini	positif
14	Ito Ito	5	2025-07-18 07:55:59	Pokoknya josss	positif
15	Wibowo Sudarman	5	2025-07-18 07:55:27	pesanan sesuai selalu,recomended deh	positif
16	Ratna Nirfana	1	2025-07-18 07:55:09	terlalu bnyk vervikasi	negatif
17	Azka Kaza	5	2025-07-18 07:54:30	JOSJIIS	positif
18	Lilis Sumiati	5	2025-07-18 07:54:22	makasih paket nya udah datang barang nya bagus	positif
19	Henia Suhenia	5	2025-07-18 07:53:50	bagus harga bersahabat	positif
20	Adi H	5	2025-07-18 07:53:41	senang belanja di shopee	positif

Gambar 4.5 Hasil Labelling Sentimen

4.1.3 Case-folding

Setelah mengklasifikasikan sentimen, selanjutnya akan dilakukan proses *case-folding* pada beberapa data ulasan yang telah dipilih, dimana semua huruf diubah ke bentuk non-kapital (lowercase) semua.

	komentar
1	top markotop sesuai pesanan
2	aplikasi yg sangat menguntungkan&sangat membantu jual beli. 👍👍👍👍
3	bagus dan pengiriman sangat cepat
4	sangat bark
5	enak dan tidak mengecewakan klo dari shopee langsung
6	kurir shopee express nya kurang ramah buruk
7	si gk ad respon, lapor shopee berkali-kali jawaban x itu itu saja suruh menghubungi penjual dan tidak bisa pembatalan pembeli merasa dirugikan sungguh sangat buruk sekali pelayanan shopee
8	jozz selalu dah..
9	voucher gaksa
10	pesanan buku cepat datang
11	bagus dan murah murah
12	kembalikan uang refun saya terlalu berbelit2 rahan duit orang
13	mayoritas beli di shopee tuh bagus walau terkadang ada toko"tertentu yang g amanah semoga shopee lebih selektif lagi dalam biarin ini
14	pokoknya jossss
15	pesanan sesuai selalu recomended deh
16	terlalu bnyk verivikasi
17	josjis
18	makasih paket nya udah datang barang nya bagus
19	bagus harga bersahabat
20	senang belanja di shopee

Gambar 4.6 Hasil Case-folding

4.1.4 Cleaning

Setelah dilakukan *case-folding*, tahap selanjutnya adalah proses *cleaning*, dimana akan dilakukan pembersihan terhadap angka, simbol, ataupun tanda baca.

	komentar
1	top markotop sesuai pesanan
2	aplikasi yg sangat menguntungkansangat membantu jual beli
3	bagus dan pengiriman sangat cepat
4	sangat bark
5	enak dan tidak mengecewakan klo dari shopee langsung
6	kurir shopee express nya kurang ramah buruk
7	si gk ad respon lapor shopee berkalikali jawaban x itu itu saja suruh menghubungi penjual dan tidak bisa pembatalan pembeli merasa dirugikan sungguh sangat buruk sekali pelayanan shopee
8	jozz selalu dah
9	voucher gaksa
10	pesanan buku cepat datang
11	bagus dan murah murah
12	kembalikan uang refun sayaterlalu berbelit rahan duit orang
13	mayoritas beli di shopee tuh bagus walau terkadang ada tokotertentu yang g amanah semoga shopee lebih selektif lagi dalam biarin ini
14	pokoknya jossss
15	pesanan sesuai selalurecomended deh
16	terlalu bnyk verivikasi
17	josjis
18	makasih paket nya udah datang barang nya bagus
19	bagus harga bersahabat
20	senang belanja di shopee

Gambar 4.7 Hasil Cleaning

4.1.5 Tokenizing

Setelah dilakukan *cleaning*, tahap selanjutnya adalah proses *tokenizing*, dimana akan dilakukan pemecahan teks menjadi frasa.

	komentar
1	top markotop sesuai pesanan
2	aplikasi yg sangat menguntungkansangat membantu jual beli
3	bagus dan pengiriman sangat cepat
4	sangat baik
5	enak dan tidak mengecewakan klo dari shopee langsung
6	kurir shopee ekpresa nya kurang ramah buruk
7	ibu gk ad respon lapor shopee berkalikali jawaban x itu itu saja suruh menghubung penjual dan tidak bisa pembatalan pembeli merasa dirugikan sungguh sangat buruk sekali pelayanan shopee
8	jozz sekeliu dah
9	voucher gratis
10	pesanan buku cepat datang
11	bagus dan murah murah
12	kembalikan uang refun sayatertalu berlele rohan duit orang
13	mayoritas beli di shopee tuh bagus walau terkadang ada tokotertentu yang g amanah semoga shopee lebih seleksi lagi dalam biarsa ini
14	pokoknya joosss
15	pesanan sesuai selalurecomended deh
16	lmyk variasi
17	jogja
18	makasih paket nya udah datang barang nya bagus
19	bagus harga bersahabat
20	senang belanja di shopee

Gambar 4.8 Hasil Tokenizing

4.1.6 Stopword Removal

Tahap berikutnya setelah tokenisasi adalah stopword removal, yaitu proses penyaringan kata-kata umum yang tidak memberikan kontribusi signifikan dalam analisis teks.

	komentar
1	top markotop sesuai pesanan
2	aplikasi yg menguntungkansangat membantu jual beli
3	bagus pengiriman cepat
4	
5	enak mengecewakan klo shopee langsung
6	kurir shopee ekpresa nya ramah buruk
7	ia chat penjual klo shopee mengembalikan dana chat ulespedisi gk ad respon lapor shopee berkalikali x suruh menghubungi penjual pembatalan pembeli dirugikan sungguh buruk pelayanan shopee
8	jozz dah
9	voucher gratis
10	pesanan buku cepat
11	bagus murah murah
12	kembalikan uang refun sayatertalu berlele rohan duit orang
13	mayoritas beli shopee tuh bagus terkadang tokotertentu g amanah semoga shopee seleksi biarsa
14	pokoknya joosss
15	pesanan sesuai selalurecomended deh
16	lmyk variasi
17	jogja
18	makasih paket nya udah barang nya bagus
19	bagus harga bersahabat
20	senang belanja shopee

Gambar 4.9 Hasil Stopword Removal

4.1.7 Stemming

Tahap berikutnya setelah stopword removal adalah proses stemming, yang berfungsi untuk mereduksi kata berimbuhan menjadi bentuk dasarnya.

	komentar
1	top markotop sesuai pesanan
2	aplikasi yg menguntungkansangat membantu jual beli
3	bagu pengiriman cepat
4	
5	enak mengecewakan klo shope langsung
6	kurir shope express nya ramah bunuh
7	soalx sya chat penjual klo shope mengembalikan bana chat ekspedisi gk ad respon lapor shope berkalikali x suruh menghubungi penjual pembatalan pemb dirugikan sungguh buruk pelayanan shope
8	jusz deah
9	voucher gaiks
10	pesanan buru cepat
11	bagu murah murah
12	kembalikan uang refun sayatertipu berlat rohan duit orang
13	mayanta beli shope tuh bagu terkadang kekcelamtu g amanah semoga shope sekeloll bisni
14	pokoknya jossss
15	pesanan sesuai sekalirecomend doh
16	bnyk vervisasi
17	joss
18	makasih paket nya udah barang nya bagu
19	bagu harga bersahabat
20	senang belanja shopu

Gambar 4.10 Hasil Stemming

4.1.8 TF-IDF

Setelah melalui tahap pra-pemrosesan, proses dilanjutkan dengan ekstraksi fitur menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency) untuk mengonversi data teks menjadi representasi numerik. Dalam metode ini, kata-kata yang frekuensi kemunculannya tinggi dalam keseluruhan dokumen justru akan diberi bobot yang lebih rendah. Sebaliknya, istilah-istilah yang unik dan jarang muncul akan memperoleh nilai bobot yang lebih signifikan. Berikut merupakan daftar kata beserta bobot TF-IDF yang mencakup kata umum bernilai rendah dan kata khusus bernilai tinggi:

Top 10 Kata dengan IDF Tertinggi (Paling Unik/Langka):

	Feature	IDF
0	abi	6.523459
1	nungguin	6.523459
2	ngelag	6.523459
3	ngerepotin	6.523459
4	ngga	6.523459
5	ngirim	6.523459
6	ngomong	6.523459
7	ngsih	6.523459
8	ni	6.523459
9	nice	6.523459

Top 10 Kata dengan IDF Terendah (Paling Umum):

	Feature	IDF
1121	sesuai	3.815409
1122	mantap	3.720099
1123	aplikasi	3.661258
1124	barang	3.579020
1125	yg	3.579020
1126	membantu	3.553044
1127	nya	3.553044
1128	belanja	3.105732
1129	bagu	2.899118
1130	shope	2.673311

Gambar 4.11 Hasil TF-IDF

4.1.9 Splitting Data

Setelah tahap TF-IDF selesai, tahap selanjutnya adalah splitting data, dimana dataset dibagi dengan proporsi 80% untuk data latih (training data) guna membangun model, dan 20% untuk data uji (testing data) yang berfungsi mengevaluasi kinerja model. Berikut adalah hasil pembagian datanya:

```
Distribusi Sentimen di Data Latih (y_train):
sentimen
positif    331
negatif     61
netral       8
Name: count, dtype: int64

Distribusi Sentimen di Data Uji (y_test):
sentimen
positif    83
negatif    15
netral      2
Name: count, dtype: int64
```

Gambar 4.12 Hasil Splitting Data

Berdasarkan gambar diatas, terdapat hasil dari splitting data yaitu pada data latih menunjukkan sebanyak 331 data positif, 61 data negatif, dan 8 data netral. Total keseluruhan data latih sebanyak 400 data, sesuai dengan pembagian 80% data latih dari 500 data yang ada. Pada data uji menunjukkan sebanyak 83 data positif, 15 data negatif, dan 2 data netral. Total keseluruhan data uji sebanyak 100 data, sesuai dengan pembagian 20% data uji dari 500 data yang ada.

4.1.10 Klasifikasi Data

Tahap selanjutnya adalah klasifikasi data, dimana data akan dikategorikan menjadi 3 kelas, yakni sentimen positif (2), sentimen netral (1), dan sentimen negatif (0). Berikut hasil dari klasifikasi data:

```
Distribusi kelas di y_train:
2    331
0     61
1      8
Name: count, dtype: int64

Distribusi kelas di y_test:
2    83
0    15
1     2
Name: count, dtype: int64
```

Gambar 4.13 Hasil Klasifikasi Data

Setelah klasifikasi data selesai, kedua model dilatih dengan script dan hasil sebagai berikut:

```
# --- 8. Melatih Model ---

# Melatih model Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_resampled, y_train_resampled)
print("\nModel Random Forest telah dilatih.")

# Melatih model XGBoost
xgb_model = XGBClassifier(eval_metric='mlogloss', random_state=42)
xgb_model.fit(X_train_resampled, y_train_resampled)
print("Model XGBoost telah dilatih.")
```

```
Model Random Forest telah dilatih.
Model XGBoost telah dilatih.
```

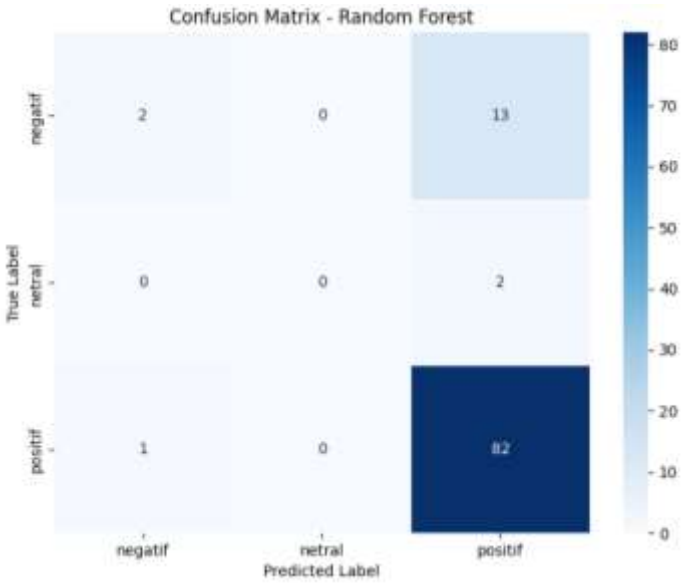
Gambar 4.14 Melatih Model

4.2 Pembahasan

Pembahasan meliputi hasil evaluasi dari perbandingan algoritma Random Forest dan XGBoost pada studi penelitian ini.

4.2.1 Hasil Confusion Matrix

Setelah model Random Forest dan XGBoost dilatih, akan muncul beberapa hasil evaluasi seperti Confusion Matrix. Berikut tampilan Confusion Matrix dari Random Forest dan XGBoost:



Gambar 4.15 Confusion Matrix Random Forest

Berdasarkan gambar diatas, terdapat klasifikasi dan perhitungan setiap kelas, yaitu:

1. Kelas Negatif
    - TP (Negatif): 2 (Prediksi: Negatif, Sebenarnya: Negatif)
    - FP (Negatif): 0 (Netral→Negatif) + 1 (Positif→Negatif) = 1 (Diprediksi Negatif, padahal Netral/Positif)
    - FN (Negatif): 0 (Negatif→Netral) + 13 (Negatif→Positif) = 13 (Sebenarnya Negatif, diprediksi Netral/Positif)
    - TN (Negatif): 0 + 2 + 0 + 82 = 84 (Bukan Negatif, diprediksi bukan Negatif)

2. Kelas Netral

   - TP (Netral): 0 (Prediksi: Netral, Sebenarnya: Netral)
   - FP (Netral): 0 (Negatif→Netral) + 0 (Positif→Netral) = 0
   - FN (Netral): 2 (Netral→Negatif) + 0 (Netral→Positif) = 2
   - TN (Netral): 2 + 13 + 1 + 82 = 98

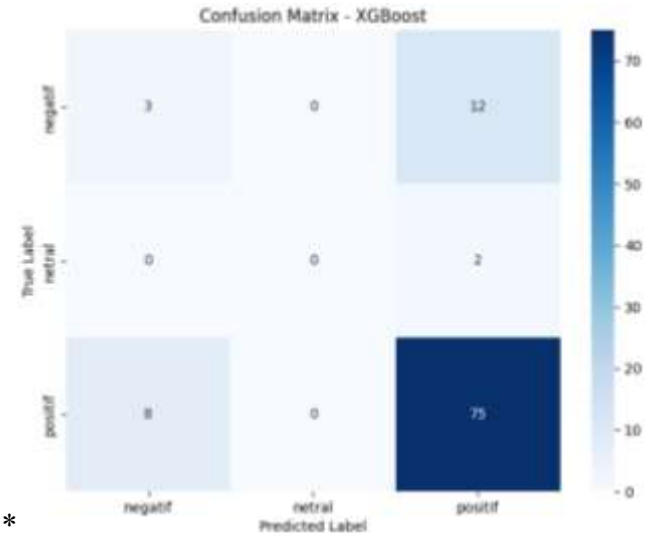
3. Kelas Positif

   - TP (Positif): 82 (Prediksi: Positif, Sebenarnya: Positif)
   - FP (Positif): 13 (Negatif→Positif) + 2 (Netral→Positif) = 15
   - FN (Positif): 1 (Positif→Negatif) + 0 (Positif→Netral) = 1
   - TN (Positif): 2 + 0 + 0 + 0 = 2

Tabel 4.1 Rangkuman Tiap Kelas pada RF

Kelas	TP	FP	FN	TN
Negatif	2	1	13	84
Netral	0	0	2	98
Positif	82	15	1	2

Selain itu juga terdapat hasil dari confusion matrix XGBoost yang akan dilampirkan sebagai berikut:



Gambar 4.16 Confusion Matrix XGBoost

Berdasarkan gambar diatas, terdapat klasifikasi dan perhitungan setiap kelas, yaitu:

1. Kelas Negatif
    - TP (Negatif): 3 (positif diprediksi negatif dan asli negatif)
    - FP (Negatif): Netral yang diprediksi negatif: 0, Positif yang diprediksi negatif: 8. Total FP = 0 + 8 = 8
    - FN (Negatif): Negatif yang diprediksi netral: 0, Negatif yang diprediksi positif: 12. Total FN = 0 + 12 = 12
    - TN (Negatif): Semua selain baris/kolom negatif: TN = Netral yang diprediksi netral + Netral yang diprediksi positif + Positif yang diprediksi netral + Positif yang diprediksi positif. TN = 0 + 2 + 0 + 75 = 77
2. Kelas Netral
    - TP (Netral): 0 (Benar diprediksi netral dan memang netral)
    - FP (Netral): Negatif yang diprediksi netral: 0, Positif yang diprediksi netral: 0. Total FP = 0 + 0 = 0
    - FN (Netral): Netral yang diprediksi negatif: 0, Netral yang diprediksi positif: 2. Total FN = 0 + 2 = 2
    - TN (Netral): Semua selain baris/kolom netral: TN = Negatif-negatif + Negatif-positif + Positif-negatif + Positif-positif. TN = 3 + 12 + 8 + 75 = 98
3. Kelas Positif
    - TP (Positif): 75 (Benar diprediksi positif dan memang positif)

- FP (Positif): Negatif yang diprediksi positif: 12, Netral yang diprediksi positif: 2. Total FP = 12 + 2 = 14
- FN (Positif): Positif yang diprediksi negatif: 8, Positif yang diprediksi netral: 0. Total FN = 8 + 0 = 8
- TN (Positif): Semua selain baris/kolom positif: TN = Negatif-negatif + Negatif-netral + Netral-negatif + Netral-netral

  TN = 3 + 0 + 0 + 0 = 3

Tabel 4.2 Rangkuman Tiap Kelas pada XGBoost

Kelas	TP	FP	FN	TN
Negatif	3	8	12	77
Netral	0	0	2	98
Positif	75	14	8	3

4.2.2 Hasil Classification Report

Setelah muncul evaluasi Confusion Matrix, selanjutnya akan muncul hasil Classification Report dari kedua model. Berikut tampilan Classification Report dari Random Forest dan XGBoost:

```
Classification Report - Random Forest:
              precision    recall  f1-score   support

     negatif       0.67      0.13      0.22        15
      netral       0.00      0.00      0.00         2
     positif       0.85      0.99      0.91        83

    accuracy                           0.84       100
   macro avg       0.50      0.37      0.38       100
weighted avg       0.80      0.84      0.79       100

Akurasi: 0.8400
```

Gambar 4.17 Classification Report Random Forest

Berdasarkan gambar diatas, diperoleh hasil akurasi dari model Random Forest sebesar 0,8400 atau sebesar 84%. Adapun perhitungan manual dari tiap-tiap Classification Report Random Forest dapat menggunakan Confusion Matrix Random Forest. Berikut perhitungan manualnya:

- Accuracy : $\frac{Jumlah\ prediksi\ benar}{Total\ Seluruh\ Data} = \frac{2+0+82}{2+0+13+0+0+2+1+0+82} = \frac{84}{100} = 84\%$ (4.1)

Untuk menghitung Precision, Recall, dan F1-Score, dapat menggunakan TP, TN, FP, dan FN yang sudah dihitung sebelumnya.

- Precision
  - Negatif : $\frac{TP}{TP+FP} = \frac{2}{2+1} = \frac{2}{3} = 0{,}67$ (4.2)
  - Netral : $\frac{TP}{TP+FP} = \frac{0}{0+0} = \frac{0}{0} = 0{,}00$ (4.3)
  - Positif : $\frac{TP}{TP+FP} = \frac{82}{82+15} = \frac{82}{97} = 0{,}85$ (4.4)
  - Macro avg : $\frac{M1+M2+M3}{n} = \frac{0{,}67+0{,}00+0{,}85}{3} = \frac{1{,}52}{3} = 0{,}50$ (4.4)
  - Weighted avg : $\frac{(M1\times s1)+(M2\times s2)+(M3\times s3)}{jumlah\ support} = \frac{(0{,}67\times 15)+(0{,}00\times 2)+(0{,}85\times 83)}{100}$

    $= \frac{10{,}05+0+70{,}55}{100} = 0{,}80$ (4.5)
- Recall
  - Negatif : $\frac{TP}{TP+FN} = \frac{2}{2+13} = \frac{2}{15} = 0{,}13$ (4.6)
  - Netral : $\frac{TP}{TP+FN} = \frac{0}{0+2} = \frac{0}{2} = 0{,}00$ (4.7)
  - Positif : $\frac{TP}{TP+FN} = \frac{82}{82+1} = \frac{82}{97} = 0{,}99$ (4.8)
  - Macro avg : $\frac{M1+M2+M3}{n} = \frac{0{,}13+0{,}00+0{,}99}{3} = \frac{1{,}12}{3} = 0{,}37$ (4.9)
  - Weighted avg : $\frac{(M1\times s1)+(M2\times s2)+(M3\times s3)}{jumlah\ support} = \frac{(0{,}13\times 15)+(0{,}00\times 2)+(0{,}99\times 83)}{100}$

    $= \frac{1{,}95+0+82{,}17}{100} = 0{,}84$ (4.10)

- F1-Score

  - Negatif : $2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0{,}67 \times 0{,}13}{0{,}67 + 0{,}13} = \frac{0{,}1742}{0{,}8} = 0{,}22$ (4.11)

  - Netral : $2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0{,}00 \times 0{,}00}{0{,}00 + 0{,}00} = 0{,}00$ (4.12)

  - Positif : $2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0{,}85 \times 0{,}99}{0{,}85 + 0{,}99} = \frac{1{,}683}{1{,}84} = 0{,}91$ (4.13)

  - Macro avg : $\frac{M1 + M2 + M3}{n} = \frac{0{,}13 + 0{,}00 + 0{,}99}{3} = \frac{1{,}12}{3} = 0{,}37$ (4.14)

  - Weighted avg : $\frac{(M1 \times s1) + (M2 \times s2) + (M3 \times s3)}{jumlah\ support} = \frac{(0{,}22 \times 15) + (0{,}00 \times 2) + (0{,}91 \times 83)}{100}$

    $= \frac{3{,}3 + 0 + 75{,}53}{100} = 0{,}79$ (4.15)

Classification Report - XGBoost:

	precision	recall	f1-score	support
negatif	0.27	0.20	0.23	15
netral	0.00	0.00	0.00	2
positif	0.84	0.90	0.87	83
accuracy			0.78	100
macro avg	0.37	0.37	0.37	100
weighted avg	0.74	0.78	0.76	100

Akurasi: 0.7800

Gambar 4.18 Classification Report XGBoost

Berdasarkan gambar diatas, diperoleh hasil akurasi dari model XGBoost sebesar 0,7800 atau sebesar 78%. Adapun perhitungan manual dari tiap-tiap Classification Report XGBoost dapat menggunakan Confusion Matrix XGBoost. Berikut perhitungan manualnya:

- Accuracy : $\frac{Jumlah\ prediksi\ benar}{Total\ Seluruh\ Data} = \frac{3+0+75}{3+0+12+0+0+2+8+0+75} = \frac{78}{100} = 78\%$ (4.16)

Untuk menghitung Precision, Recall, dan F1-Score, dapat menggunakan TP, TN, FP, dan FN yang sudah dihitung sebelumnya.

- Precision

  - Negatif : $\frac{TP}{TP+FP} = \frac{3}{3+8} = \frac{3}{11} = 0{,}27$ (4.17)

  - Netral : $\frac{TP}{TP+FP} = \frac{0}{0+0} = \frac{0}{0} = 0{,}00$ (4.18)

  - Positif : $\frac{TP}{TP+FP} = \frac{75}{75+14} = \frac{75}{89} = 0{,}84$ (4.19)

  - Macro avg : $\frac{M1+M2+M3}{n} = \frac{0{,}27+0{,}00+0{,}84}{3} = \frac{1{,}11}{3} = 0{,}37$ (4.20)

  - Weighted avg : $\frac{(M1\times s1)+(M2\times s2)+(M3\times s3)}{jumlah\ support} = \frac{(0{,}27\times 15)+(0{,}00\times 2)+(0{,}84\times 83)}{100}$

    $= \frac{4{,}05+0+69.72}{100} = 0{,}74$ (4.21)

- Recall

  - Negatif : $\frac{TP}{TP+FN} = \frac{3}{3+12} = \frac{3}{15} = 0{,}20$ (4.22)

  - Netral : $\frac{TP}{TP+FN} = \frac{0}{0+2} = \frac{0}{2} = 0{,}00$ (4.23)

  - Positif : $\frac{TP}{TP+FN} = \frac{75}{75+8} = \frac{75}{83} = 0{,}90$ (4.24)

  - Macro avg : $\frac{M1+M2+M3}{n} = \frac{0{,}20+0{,}00+0{,}90}{3} = \frac{1{,}12}{3} = 0{,}37$ (4.25)

  - Weighted avg : $\frac{(M1\times s1)+(M2\times s2)+(M3\times s3)}{jumlah\ support} = \frac{(0{,}20\times 15)+(0{,}00\times 2)+(0{,}90\times 83)}{100}$

    $= \frac{3+0+74{,}7}{100} = 0{,}78$ (4.26)

- F1-Score

  - Negatif : $2 \times \frac{Precision\times Recall}{Precision+Recall} = 2 \times \frac{0{,}27\times 0{,}20}{0{,}27+0{,}20} = \frac{0{,}108}{0{,}47} = 0{,}23$ (4.27)

  - Netral : $2 \times \frac{Precision\times Recall}{Precision+Recall} = 2 \times \frac{0{,}00\times 0{,}00}{0{,}00+0{,}00} = 0{,}00$ (4.28)

  - Positif : $2 \times \frac{Precision\times Recall}{Precision+Recall} = 2 \times \frac{0{,}84\times 0{,}90}{0{,}84+0{,}90} = \frac{1{,}512}{1{,}74} = 0{,}87$ (4.29)

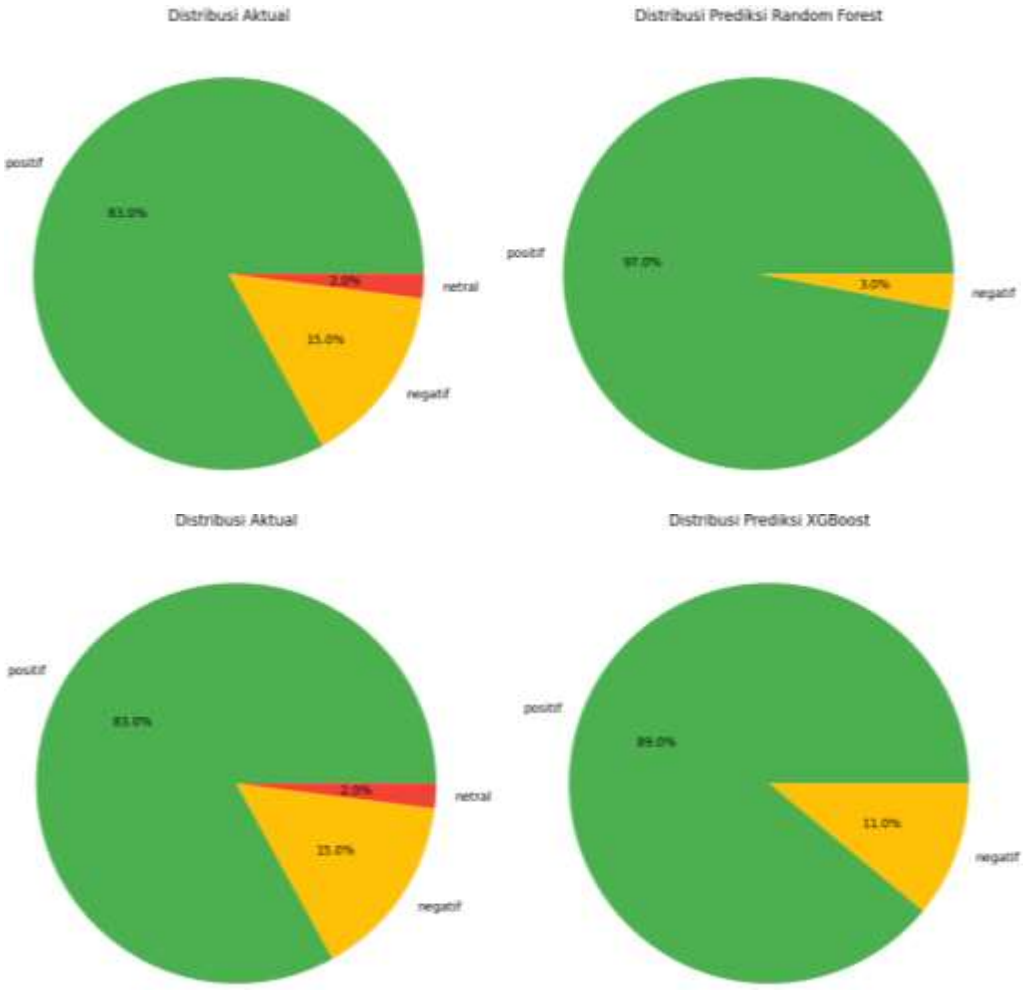
- Macro avg : $\frac{M1+M2+M3}{n} = \frac{0,23+0,00+0,87}{3} = \frac{1,1}{3} = 0,37$ (4.30)

- Weighted avg : $\frac{(M1 \times s1)+(M2 \times s2)+(M3 \times s3)}{jumlah\ support} = \frac{(0,23 \times 15)+(0,00 \times 2)+(0,87 \times 83)}{100}$

$= \frac{3,45+0+72,21}{100} = 0,76$ (4.31)

4.2.3 Hasil Pie Chart

Setelah muncul Classification Report, akan muncul Pie Chart distribusi aktual dan prediksi dari kedua model. Berikut tampilan Pie Chart distribusi aktual dan prediksi dari Random Forest dan XGBoost:



Gambar 4.19 Hasil Pie Chart

Berdasarkan Kedua Pie Chart Distribusi Random Forest dan XGBoost, dapat disimpulkan bahwa kedua prediksi model ini menunjukkan peningkatan proporsi sentimen positif dibandingkan dengan distribusi aktual, sedangkan proporsi negatif menurun dan hilangnya kategori netral. Hal ini menunjukkan bahwa model berhasil mengenali polaritas positif dengan baik, tetapi mungkin kurang sensitif dalam menangkap sentimen netral.

4.2.4 Hasil WordCloud

Gambar 4.20 Hasil WordCloud

Berdasarkan WordCloud diatas, dapat disimpulkan bahwa kedua model menunjukkan persepsi positif terhadap Shopee, dengan penekanan pada kemudahan dalam berbelanja dan efisiensi layanan. Meskipun ada keluhan, kebanyakan komentar cenderung positif, yang menunjukkan bahwa Shopee berhasil memenuhi harapan banyak pengguna. Analisis ini dapat menjadi acuan untuk perbaikan layanan, terutama dalam hal pengiriman dan respons layanan pelanggan untuk meningkatkan kepuasan pengguna.

BAB V
PENUTUP

5.1. Kesimpulan

Temuan utama dari penelitian yang membandingkan efektivitas Algoritma Random Forest dan XGBoost Terhadap Analisis Sentimen E-Commerce disajikan sebagai berikut:

1. Berdasarkan studi yang telah dilaksanakan, implementasi Algoritma Random Forest dan XGBoost untuk menganalisis sentimen ulasan aplikasi Shopee dilakukan melalui serangkaian tahapan metodologis. Proses berawal dari pengumpulan data, setelah itu tahap pra-pemrosesan yang meliputi case-folding untuk mengonversi seluruh teks ke dalam bentuk huruf kecil. Kemudian melakukan tahapan cleaning, yakni pembersihan teks dari karakter atau simbol yang tidak diperlukan. Tahap selanjutnya, yakni tokenisasi, dengan mengubah teks ulasan menjadi unit kata individual. Proses ini dilanjutkan dengan menghapus stopword guna menyaring kata-kata yang tidak memiliki makna substantif. Selanjutnya, dilakukan proses stemming guna mereduksi setiap kata ke dalam bentuk dasarnya dengan menghilangkan seluruh afiks. Pada tahap akhir pra-pemrosesan, ekstraksi fitur menggunakan TF-IDF diterapkan untuk mengonversi data teks menjadi representasi numerik berdasarkan bobot statistik setiap kata. Kemudian dilakukan splitting data dan klasifikasi, serta melatih model algoritma dengan menggunakan beberapa metrik, seperti akurasi, precision, recall, dan f1-score. Dengan langkah-langkah sistematis ini, Random Forest

dan XGBoost dapat diterapkan secara efektif untuk analisis sentimen ulasan Shopee.

2. Perbandingan hasil kinerja antara Algoritma Random Forest dan XGBoost mengilustrasikan bahwa kedua algoritma mempunyai tingkat akurasi yang beragam. Implementasi algoritma Random Forest menghasilkan nilai presisi sebesar 0,8400 atau sebesar 84%, sedangkan pada algoritma XGBoost diperoleh hasil akurasi sebesar 0,7800 atau sebesar 78%.

5.2. Saran

Saran dari Perbandingan Algoritma Random Forest dan XGBoost untuk Analisis Sentimen Pada Aplikasi E-Commerce Shopee dapat dilihat sebagai berikut:

1. Sebaiknya dataset yang telah dikembangkan dengan algoritma Random Forest dan XGBoost dapat dikembangkan dengan algoritma dan bahasa pemrograman yang lain.
2. Disarankan untuk penelitian selanjutnya mengadopsi pendekatan atau fitur tambahan yang mampu meningkatkan kemampuan model dalam mendeteksi sentimen netral. Hal ini penting agar model tidak bias hanya pada polaritas positif dan negatif, tetapi juga mampu menangkap nuansa yang lebih halus dari data sentimen.

DAFTAR PUSTAKA

Saputra, D. C., Fauzan, M., & Aldosion, G. C. (2025). Dampak Rating Dan Ulasan Pengguna Di Google Playstore Terhadap Keputusan Pengunduhan Aplikasi. *Spectrum: Multidisciplinary Journal*, *2*(1), 22–29.

Nurian, A., Amalia, I. N., & Rozikin, C. (2024). Implementasi Algoritma Naïve Bayes Classifier dalam Analisis Sentimen terhadap Ulasan Pengguna Aplikasi Shopee di Platform Google Play. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(1).

Budaya I. G. B. A. & I. K. P. Suniantara. Perbandingan Algoritma Analisis Sentimen dengan Penerapan SMOTE Oversampling dan TF-IDF pada Ulasan Google untuk Puskesmas. *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 3, pp. 1077–1086, 2024, doi: 10.57152/malcom.v4i3.1459.

Atmajaya, D., Febrianti, A., & Darwis, H. (2023). Analisis Sentimen untuk ChatGPT di Twitter dengan Metode SVM dan Naive Bayes. *The Indonesian Journal of Computer Science*, *12*(4).

Kurniawan, D., Wahyudi, M., Pujiastuti, L., & Sumanto, S. (2024). Sistem Deteksi dan Prediksi Cerdas untuk Penyakit Paru-Paru Menggunakan Algoritma Random Forest. *Indonesian Journal Computer Science*, *3*(1), 51–56.

Srinivas, C. G. P., Balachander, S., Singh Samant, Y. C., Hariharan, B. V., & Devi, M. N. (2021). *Deteksi Hardware Trojan pada Perangkat IoT Menggunakan Algoritma XGBoost dengan Augmentasi Data CTGAN dan SMOTE* (pp. 116–127). https://doi.org/10.1007/978-3-030-79276-3_10

Mahawardana, P. P. O., Sasmita, G. A., & Pratama, I. P. A. E. (2022). Eksplorasi Sentimen Opini Publik di Twitter mengenai "Figur Pemimpin" Menggunakan Pemrograman Python. *Jurnal Ilmiah Teknologi Dan Komputer*, *3*(1), 810–820. https://www.neliti.com/publications/432979/analisis-sentimen-berdasarkan-opini-dari-media-sosial-twitter-terhadap-figure-pe#cite

Fradesa, F., Abadi, S. P., Maani, B., Hardi, & Sucipto, S. (2022). Inovasi Fitur Halal Shopee Barokah dan Tokopedia Salam sebagai Strategi Pengembangan Ekonomi Digital Syariah. *Jurnal Ilmiah Ekonomi Islam*, *8*(3), 2893.

Jalilifard, A., Caridá, V. F., Mansano, A. F., Cristo, R. S., & da Fonseca, F. P. C. (2021). *Pengembangan TF-IDF Sensitif Semantik untuk Menentukan Relevansi Kata dalam Dokumen.* 327–337.

Siregar, A. P., Purba, D. P., Pasaribu, J. P., Bakara, K. R. (2023). Pemanfaatan Algoritma Random Forest untuk Klasifikasi Diagnosis Penyakit Stroke. *Jurnal Penelitian Rumpun Ilmu Teknik*, *2*(4), 155–164. https://doi.org/10.55606/juprit.v2i4.3039

Yuyun, Latief, A. D., Sampurno, T., Hazriani, Arisha, A. O., & Mushaf. (2023). Pengaruh Teknik Pra-pemrosesan Data terhadap Kinerja Model Deep Learning dalam Next Sentence Prediction. *2023 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 274–278.

Lampiran

Lampiran 1 Surat Penetapan Dosen Pembimbing

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 39/IL3-AU/UMSU-09/F/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi	**: Teknologi Informasi**
Pada tanggal	**: 03 Januari 2025**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama	**: Laila Salsabila**
NPM	**: 2109020051**
Semester	**: VII (Tujuh)**
Program studi	**: Teknologi Informasi**
Judul Proposal / Skripsi	**: Perbandingan Algoritma Random Forest dan XGBoost untuk Analisis Sentimen pada Aplikasi E-Commerce Shopee**
Dosen Pembimbing	**: Fatma Sari Hutagalung, S.Kom.,M.Kom.**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **03 Januari 2026**
4. Revisi judul.....

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

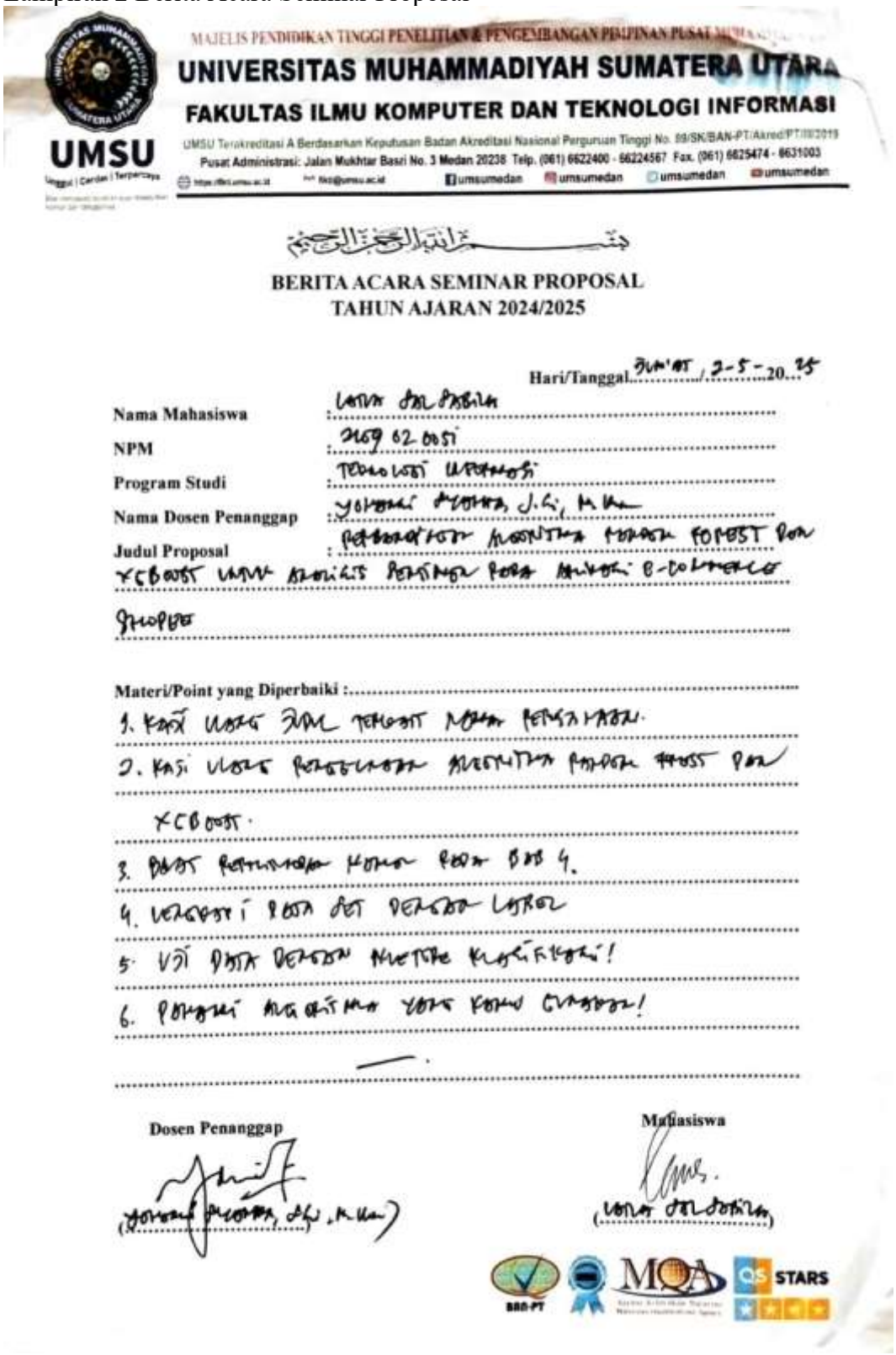
Ditetapkan di : Medan
Pada Tanggal : 03 Rajab 1446 H
03 Januari 2025 M

a.n.Dekan
Wakil Dekan I

Dedar Maulana, S.T.,M.Kom.
NIDN : 0121119102

Cc. File

Lampiran 2 Berita Acara Seminar Proposal



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

UMSU Unggul | Cerdas | Terpercaya

بسم الله الرحمن الرحيم

BERITA ACARA SEMINAR PROPOSAL
TAHUN AJARAN 2024/2025

Hari/Tanggal: Jum'at / 2-5-2025

Nama Mahasiswa : [illegible]

NPM : 2109 02 0051

Program Studi : Teknologi Informasi

Nama Dosen Penanggap : [illegible]

Judul Proposal : [illegible]

Materi/Point yang Diperbaiki :

1. [illegible]
2. [illegible]
3. [illegible]
4. [illegible]
5. [illegible]
6. [illegible]

Dosen Penanggap

([illegible])

Mahasiswa

([illegible])

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
Nomor : 744/II.3-AU/UMSU-09/G/2025
HAL UJIAN MEJA HIJAU SARJANA (S1)

UNDANGAN
PANGGILAN

Kepada Yang Terhormat
Bapak/Ibu Dosen Penguji Meja HIjau
di
Medan

UMSU
Unggul | Cerdas | Terpercaya

Fakultas : Ilmu Komputer dan Teknologi Informasi
Program Studi : Teknologi Informasi
Hari/Tanggal : Selasa, 22 Juli 2025
Waktu /Tempat : 08:00-14.00WIB/G

Catatan :
*Harap datang tepat waktu karena ujian dalam bentuk tim (2 Orang) penguji I & II
*Dosen Penguji yang terlambat 30 menit akan digamti

28	Aidil Azhar	2109020137	Monitoring Alat Hitung Tetesan Infus Dengan ESP32	Dr. Al-Khowarizmi, M.Kom.	Yoshida Sary, S.Kom., M.Kom	Amrullah, S.Kom, M.Kom	
29	Laila Salsabila	2109020051	Perbandingan Kinerja Algoritma Random Forest dan XGBoost Terhadap Analisis Sentimen Aplikasi E-Commerce	Yohanni Syahra, S.Si.,M.Kom.	Halim Maulana, S.T, M.Kom	Fatma Sari Hutagalung, S.Kom, M.Kom m	

Medan, 22 Muharram 1447 H
17 Juli 2025 M

Asisten Pengambilan Berita Acara :
1. Suvia Agustin S.I.Kom
2. Andika Suras Saputra, S.M

Ditetapkan Oleh

a.n Rektor
Wakil Rektor I

Prof. Dr. MUHAMMAD ARIFIN, SH.M, Hum

Panitia Ujian

Ketua
Dekan

Dr. Al-Khowarizmi, M.Kom

Sekretaris
Wakil Dekan I

Halim Maulana, S.T., M.Kom

Lampiran 3 Surat Undangan Sidang Meja Hijau

Lampiran 4 Source Code*-

```python
# --- 1. Import Library yang Dibutuhkan ---
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import os # Import modul os untuk debugging FileNotFoundError

# NLTK untuk Text Preprocessing
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer # Atau Sastrawi untuk Bahasa Indonesia yang lebih baik
from nltk.tokenize import word_tokenize

# Scikit-learn untuk Feature Extraction, Model, dan Evaluasi
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder
from xgboost import XGBClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
from xgboost import XGBClassifier, plot_tree

# Imbalanced-learn untuk Oversampling (SMOTE)
from imblearn.over_sampling import SMOTE

# --- Tambahan untuk Visualisasi Pohon Keputusan ---
from sklearn.tree import plot_tree
# Untuk XGBoost, Anda mungkin perlu menginstal graphviz:
# pip install graphviz
# import graphviz
# from xgboost import plot_tree as xgb_plot_tree

# --- PERBAIKAN: Mengunduh NLTK data secara langsung tanpa try-except yang kompleks ---
# Ini akan memastikan resource terunduh. Jika sudah ada, NLTK akan melewatkannya.
print("Memeriksa dan mengunduh NLTK data...")
nltk.download('stopwords', quiet=True) # quiet=True agar tidak terlalu banyak output
nltk.download('punkt', quiet=True)
nltk.download('punkt_tab', quiet=True) # Ini adalah resource yang menyebabkan LookupError
print("NLTK data siap.")

# Jika ingin menggunakan Sastrawi (lebih baik untuk stemming Bahasa Indonesia)
# from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
# factory = StemmerFactory()
# stemmer = factory.create_stemmer()
print("Library berhasil diimpor.")
```

```python
# --- 2. Memuat Dataset ---
# Pastikan file 'ulasan_500.csv' berada di direktori yang sama dengan notebook ini
file_path = 'ulasan_500.csv' # Sesuaikan path jika perlu
df = pd.read_csv(file_path)

# Menampilkan 5 baris pertama dataset
print("\n--- 2. Dataset Awal ---")
print(df.head())

# Menampilkan informasi dasar dataset
print("\nInformasi Dataset:")
df.info()

# Mengganti nama kolom agar lebih mudah diakses
df.rename(columns={'userName': 'nama', 'score': 'rating', 'at': 'tanggal', 'content': 'komentar'}, inplace=True)
print("\nKolom setelah diganti nama:")
print(df.head())
```

```python
# --- 3. Pelabelan Sentimen ---
# Fungsi untuk melabeli sentimen berdasarkan rating
def label_sentiment(rating):
    if rating >= 4:
        return 'positif'
    elif rating == 3:
        return 'netral'
    else: # rating <= 2
        return 'negatif'

# Menerapkan fungsi pelabelan ke kolom 'rating' untuk membuat kolom 'sentimen'
df['sentimen'] = df['rating'].apply(label_sentiment)

print("\n--- 3. Hasil Pelabelan Sentimen ---")
print(df[['rating', 'sentimen', 'komentar']].head())

# Menampilkan distribusi sentimen
sentiment_counts = df['sentimen'].value_counts()
print("\nDistribusi Sentimen:")
print(sentiment_counts)

# --- TAMBAHAN: Menyimpan DataFrame yang Sudah Dilabeli ke CSV Baru ---
output_file_path = 'ulasan_shopee_dilabeli.csv'
df.to_csv(output_file_path, index=False) # index=False agar tidak menyimpan indeks DataFrame sebagai kolom
print(f"\nDataFrame yang sudah dilabeli sentimen berhasil disimpan ke '{output_file_path}'")
```

```python
# --- 4. Text Preprocessing (Tahapan Terpisah dengan Penyimpanan CSV) ---
# Inisialisasi stop words Bahasa Indonesia dan stemmer
indonesian_stopwords = set(stopwords.words('indonesian'))
stemmer = PorterStemmer()
# --- DEFINISI FUNGSI PREPROCESSING TERPISAH ---
def do_case_folding(text):
    return text.lower()
def do_cleaning(text):
    # Menghapus angka, tanda baca, dan karakter non-alfabetik lainnya
    text = re.sub(r'[^a-z\s]', '', text)
    # Menghapus spasi berlebih
    text = re.sub(r'\s+', ' ', text).strip()
    return text

def do_tokenizing(text):
    return word_tokenize(text)
def do_stopword_removal(tokens):
    return [word for word in tokens if word not in indonesian_stopwords]
def do_stemming(tokens):
    return [stemmer.stem(word) for word in tokens]

# --- AKHIR DEFINISI FUNGSI PREPROCESSING TERPISAH ---
print("\n--- 4. Memulai Proses Text Preprocessing ---")

# --- Tahap 1: Case Folding ---
df['komentar_casefolded'] = df['komentar'].apply(do_case_folding)
output_casefolded_file = 'komentar_casefolded.csv'
df[['komentar_casefolded']].to_csv(output_casefolded_file, index=False, header=['komentar'])
print(f"Hasil Case Folding disimpan ke '{output_casefolded_file}'")
print(f"Contoh Case Folding: {df['komentar_casefolded'].iloc[0]}\n")
```

```
# --- Tahap 2: Cleaning ---
df['komentar_cleaned'] = df['komentar_casefolded'].apply(do_cleaning)
output_cleaned_file = 'komentar_cleaned.csv'
df[['komentar_cleaned']].to_csv(output_cleaned_file, index=False, header=['komentar'])
print(f"Hasil Cleaning disimpan ke '{output_cleaned_file}'")
print(f"Contoh Cleaning: {df['komentar_cleaned'].iloc[0]}\n")

# --- Tahap 3: Tokenizing ---
# Hasil tokenizing adalah list of strings, jadi kita simpan sebagai string yang dipisahkan spasi
df['komentar_tokenized_list'] = df['komentar_cleaned'].apply(do_tokenizing)
# Untuk penyimpanan CSV, kita gabungkan kembali menjadi string
df['komentar_tokenized'] = df['komentar_tokenized_list'].apply(lambda x: ' '.join(x))
output_tokenized_file = 'komentar_tokenized.csv'
df[['komentar_tokenized']].to_csv(output_tokenized_file, index=False, header=['komentar'])
print(f"Hasil Tokenizing disimpan ke '{output_tokenized_file}'")
print(f"Contoh Tokenizing: {df['komentar_tokenized'].iloc[0]}\n")

# --- Tahap 4: Stopword Removal ---
df['komentar_stopwords_removed_list'] = df['komentar_tokenized_list'].apply(do_stopword_removal)
# Untuk penyimpanan CSV, kita gabungkan kembali menjadi string
df['komentar_stopwords_removed'] = df['komentar_stopwords_removed_list'].apply(lambda x: ' '.join(x))
output_stopwords_removed_file = 'komentar_stopwords_removed.csv'
df[['komentar_stopwords_removed']].to_csv(output_stopwords_removed_file, index=False, header=['komentar'])
print(f"Hasil Stopword Removal disimpan ke '{output_stopwords_removed_file}'")
print(f"Contoh Stopword Removal: {df['komentar_stopwords_removed'].iloc[0]}\n")
```

```
# --- Tahap 5: Stemming ---
df['komentar_stemmed_list'] = df['komentar_stopwords_removed_list'].apply(do_stemming)
# Untuk penyimpanan CSV, kita gabungkan kembali menjadi string
df['komentar_stemmed'] = df['komentar_stemmed_list'].apply(lambda x: ' '.join(x))
output_stemmed_file = 'komentar_stemmed.csv'
df[['komentar_stemmed']].to_csv(output_stemmed_file, index=False, header=['komentar'])
print(f"Hasil Stemming disimpan ke '{output_stemmed_file}'")
print(f"Contoh Stemming: {df['komentar_stemmed'].iloc[0]}\n")

# Kolom 'komentar_bersih' yang akan digunakan untuk TF-IDF adalah hasil akhir stemming
df['komentar_bersih'] = df['komentar_stemmed']
print("\n--- 4. Hasil Preprocessing Akhir (5 Komentar Pertama) ---")
for i in range(5):
    print(f"Original: {df['komentar'].iloc[i]}")
    print(f"Cleaned (Final):  {df['komentar_bersih'].iloc[i]}\n")
```

```
# --- 5. Ekstraksi Fitur dengan TF-IDF ---
# Inisialisasi TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(max_features=5000)

# Menerapkan TF-IDF pada kolom 'komentar_bersih'
X = tfidf_vectorizer.fit_transform(df['komentar_bersih'])
# Target variabel (sentimen)
y = df['sentimen']
print("\n--- 5. Hasil Ekstraksi Fitur TF-IDF ---")
print(f"Dimensi Matriks TF-IDF (Jumlah Dokumen x Jumlah Fitur): {X.shape}")
print(f"Contoh Fitur (Kata-kata) yang Diekstraksi: {tfidf_vectorizer.get_feature_names_out()[:20]}...\n")
# --- TAMBAHAN: Menampilkan dan Menyimpan Perhitungan TF-IDF ---
# 1. Dapatkan daftar fitur (kata-kata)
feature_names = tfidf_vectorizer.get_feature_names_out()

# 2. Dapatkan nilai IDF untuk setiap fitur
# Scikit-learn's IDF: log((1 + n_samples) / (1 + df)) + 1
idf_values = tfidf_vectorizer.idf_

# Buat DataFrame untuk menampilkan IDF
idf_df = pd.DataFrame({'Feature': feature_names, 'IDF': idf_values})
idf_df = idf_df.sort_values(by='IDF', ascending=False).reset_index(drop=True)
print("\n--- Perhitungan IDF (Inverse Document Frequency) ---")
print("IDF mengukur seberapa penting sebuah kata di seluruh koleksi dokumen.")
print("Semakin tinggi nilai IDF, semakin unik/langka kata tersebut.")
print("Rumus Scikit-learn: IDF(t, D) = log((1 + N) / (1 + df(t))) + 1")
print("N = Jumlah total dokumen, df(t) = Jumlah dokumen yang mengandung kata t.")
print("\nTop 10 Kata dengan IDF Tertinggi (Paling Unik/Langka):")
print(idf_df.head(10))
```

```
print("\nTop 10 Kata dengan IDF Terendah (Paling Umum):")
print(idf_df.tail(10))

# Simpan IDF ke CSV
idf_df.to_csv('tfidf_idf_values.csv', index=False)
print(f"\nNilai IDF berhasil disimpan ke 'tfidf_idf_values.csv'")

# 3. Dapatkan Matriks TF-IDF (nilai TF-IDF untuk setiap kata di setiap dokumen)
# Ubah matriks sparse X menjadi dense array untuk inspeksi
tfidf_matrix_dense = X.toarray()
# Buat DataFrame TF-IDF
# Baris adalah dokumen, kolom adalah fitur (kata-kata)
tfidf_df = pd.DataFrame(tfidf_matrix_dense, columns=feature_names)
print("\n--- Perhitungan TF-IDF (Term Frequency-Inverse Document Frequency) ---")
print("TF-IDF adalah hasil perkalian TF (Term Frequency) dan IDF (Inverse Document Frequency).")
print("TF mengukur seberapa sering kata muncul dalam satu dokumen.")
print("TF-IDF mengukur seberapa penting sebuah kata dalam sebuah dokumen relatif terhadap seluruh korpus.")
print("\nContoh Matriks TF-IDF (5 Dokumen Pertama, 10 Fitur Pertama):")
print(tfidf_df.iloc[:5, :10]) # Menampilkan 5 baris pertama dan 10 kolom pertama

# Simpan Matriks TF-IDF ke CSV
tfidf_df.to_csv('tfidf_matrix.csv', index=False)
print(f"\nMatriks TF-IDF berhasil disimpan ke 'tfidf_matrix.csv'")
```

```
# --- 6. Splitting Data (80% Latih, 20% Uji) ---
# stratify=y memastikan distribusi kelas sentimen yang sama di train dan test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
print("\n--- 6. Hasil Splitting Data ---")
print(f"Ukuran Data Latih (X_train): {X_train.shape}")
print(f"Ukuran Label Latih (y_train): {y_train.shape}")
print(f"Ukuran Data Uji (X_test): {X_test.shape}")
print(f"Ukuran Label Uji (y_test): {y_test.shape}")
print("\nDistribusi Sentimen di Data Latih (y_train):")
print(y_train.value_counts())
print("\nDistribusi Sentimen di Data Uji (y_test):")
print(y_test.value_counts())

# Gabungkan kembali X_train/X_test dengan y_train/y_test dan kolom komentar_bersih
# Karena X_train/X_test adalah matriks sparse, kita perlu mengambil indeksnya
# untuk mencocokkan dengan DataFrame asli.
# Data Latih
train_indices = y_train.index
train_df_for_csv = df.loc[train_indices, ['komentar_bersih', 'sentimen']].copy()
train_df_for_csv['TFIDF_Vector'] = [X_train[i].toarray().tolist() for i in range(X_train.shape[0])] # Simpan vektor TFIDF
train_df_for_csv.to_csv('train_data.csv', index=False)
print(f"\nData latih (komentar bersih, sentimen, dan vektor TFIDF) berhasil disimpan ke 'train_data.csv'")

# Data Uji
test_indices = y_test.index
test_df_for_csv = df.loc[test_indices, ['komentar_bersih', 'sentimen']].copy()
test_df_for_csv['TFIDF_Vector'] = [X_test[i].toarray().tolist() for i in range(X_test.shape[0])] # Simpan vektor TFIDF
test_df_for_csv.to_csv('test_data.csv', index=False)
print(f"Data uji (komentar bersih, sentimen, dan vektor TFIDF) berhasil disimpan ke 'test_data.csv'")
```

```python
# --- 7. Oversampling dengan SMOTE ---
print("\n--- 7. Oversampling dengan SMOTE ---")

# Mengonversi kelas target ke format numerik
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)  # Mengonversi 'negatif', 'netral', 'positif' menjadi 0, 1, 2

# Membagi data dengan kelas yang sudah dienkode
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42, stratify=y)

# Cek distribusi kelas
print("\nDistribusi kelas di y_train:")
print(pd.Series(y_train).value_counts())
print("\nDistribusi kelas di y_test:")
print(pd.Series(y_test).value_counts())

# Terapkan SMOTE pada data latih
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

print("\nDistribusi kelas di y_train setelah SMOTE:")
print(pd.Series(y_train_resampled).value_counts())
```

```python
# --- 8. Melatih Model ---

# Melatih model Random Forest
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_resampled, y_train_resampled)
print("\nModel Random Forest telah dilatih.")

# Melatih model XGBoost
xgb_model = XGBClassifier(eval_metric='mlogloss', random_state=42)
xgb_model.fit(X_train_resampled, y_train_resampled)
print("Model XGBoost telah dilatih.")

# --- PREDIKSI (Jika diperlukan) ---
# Jika Anda ingin melakukan prediksi setelah pelatihan, Anda bisa menambahkan:
rf_predictions = rf_model.predict(X_test)
xgb_predictions = xgb_model.predict(X_test)

print("\nPrediksi untuk data uji telah dilakukan.")
```

```python
# --- LANGKAH 9: EVALUASI MODEL ---
def evaluate_model(y_true, y_pred, model_name):
    """Fungsi untuk evaluasi dan visualisasi hasil model"""
    # Dekode label angka kembali ke string
    y_true_str = label_encoder.inverse_transform(y_true)
    y_pred_str = label_encoder.inverse_transform(y_pred)

    # Confusion Matrix
    cm = confusion_matrix(y_true_str, y_pred_str)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=label_encoder.classes_,
                yticklabels=label_encoder.classes_)
    plt.title(f'Confusion Matrix - {model_name}')
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    plt.savefig(f'confusion_matrix_{model_name}.png')  # Simpan confusion matrix sebagai gambar
    plt.show()

    # Classification Report
    report = classification_report(y_true_str, y_pred_str, target_names=label_encoder.classes_, zero_division=0)
    print(f"\nClassification Report - {model_name}:")
    print(report)

    # Akurasi
    accuracy = accuracy_score(y_true_str, y_pred_str)
    print(f"Akurasi: {accuracy:.4f}")
```

```
    # Pie Chart Distribusi
    plt.figure(figsize=(12, 6))

    plt.subplot(1, 2, 1)
    y_true_counts = pd.Series(y_true_str).value_counts()
    plt.pie(y_true_counts, labels=y_true_counts.index, autopct='%1.1f%%',
            colors=['#4CAF50', '#FFC107', '#F44336'])
    plt.title('Distribusi Aktual')

    plt.subplot(1, 2, 2)
    y_pred_counts = pd.Series(y_pred_str).value_counts()
    plt.pie(y_pred_counts, labels=y_pred_counts.index, autopct='%1.1f%%',
            colors=['#4CAF50', '#FFC107', '#F44336'])
    plt.title(f'Distribusi Prediksi {model_name}')

    plt.tight_layout()
    plt.savefig(f'pie_chart_{model_name}.png')  # Simpan pie chart sebagai gambar
    plt.show()

    return report, accuracy  # Kembalikan report dan accuracy

# --- EVALUASI MODEL RANDOM FOREST ---
print("\n=== HASIL RANDOM FOREST ===")
rf_report, rf_accuracy = evaluate_model(y_test, rf_predictions, "Random Forest")

# --- EVALUASI MODEL XGBOOST ---
print("\n=== HASIL XGBOOST ===")
xgb_report, xgb_accuracy = evaluate_model(y_test, xgb_predictions, "XGBoost")
```

```
# --- WORDCLOUD ---
def generate_wordcloud(text, model_name):
    """Fungsi untuk menghasilkan dan menyimpan WordCloud dari teks"""
    wordcloud = WordCloud(
        width=800,
        height=400,
        background_color='white',
        max_words=200,
        colormap='viridis',
        collocations=False
    ).generate(text)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'WordCloud Komentar - {model_name}')
    plt.savefig(f'wordcloud_{model_name}.png')  # Simpan WordCloud sebagai gambar
    plt.show()

# Menghasilkan WordCloud untuk komentar bersih
text_rf = ' '.join(df['komentar_bersih'])  # Ganti dengan komentar yang relevan untuk Random Forest
generate_wordcloud(text_rf, "Random Forest")
text_xgb = ' '.join(df['komentar_bersih'])  # Ganti dengan komentar yang relevan untuk XGBoost
generate_wordcloud(text_xgb, "XGBoost")

# --- MENYIMPAN HASIL EVALUASI KE FILE ---
# Simpan hasil evaluasi ke file teks
with open('evaluation_results.txt', 'w') as f:
    f.write("=== HASIL RANDOM FOREST ===\n")
    f.write(rf_report + "\n")
    f.write(f"Akurasi: {rf_accuracy:.4f}\n\n")

    f.write("=== HASIL XGBOOST ===\n")
    f.write(xgb_report + "\n")
    f.write(f"Akurasi: {xgb_accuracy:.4f}\n")

print("\nHasil evaluasi telah disimpan ke 'evaluation_results.txt'.")
```