ANALISIS PERBANDINGAN METODE RANDOM FOREST DENGAN LOGISTIC REGRESSION TERHADAP SENTIMEN PUBLIK PADA ULASAN FILM AGAK LAEN

SKRIPSI

DISUSUN OLEH

NANDA RAFIKHI AZHARI LUBIS
NPM. 2109020061

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

ANALISIS PERBANDINGAN METODE RANDOM FOREST DENGAN LOGISTIC REGRESSION TERHADAP SENTIMEN PUBLIK PADA ULASAN FILM AGAK LAEN

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

NANDA RAFIKHI AZHARI LUBIS
NPM. 2109020061

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

LEMBAR PENGESAHAN

Judul Skripsi	:	Analisis Perbandingan Metode Random Forest Dengan Logistic Regression Terhadap Sentimen Publik Pada Ulasan Film Agak Laen
Nama Mahasiswa	:	Nanda Rafikhi Azhari Lubis
NPM	:	2109020061
Program Studi	:	Teknologi Informasi

Menyetujui
Komisi Pembimbing

(Amrullah, S.Kom., M.Kom)
NIDN. 0125118604

Ketua Program Studi

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

Dekan

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

PERNYATAAN ORISINALITAS

ANALISIS PERBANDINGAN METODE RANDOM FOREST DENGAN LOGISTIC REGRESSION TERHADAP SENTIMEN PUBLIK PADA ULASAN FILM AGAK LAEN

SKRIPSI

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

Medan, Juli 2025

Yang membuat pernyataan

Nanda Rafikhi Azhari Lubis

NPM. 2109020061

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

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

Nama	: Nanda Rafikhi Azhari Lubis
NPM	: 2109020061
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:

ANALISIS PERBANDINGAN METODE RANDOM FOREST DENGAN LOGISTIC REGRESSION TERHADAP SENTIMEN PUBLIK PADA ULASAN FILM AGAK LAEN

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, Juli 2025

Yang membuat pernyataan

Nanda Rafikhi Azhari Lubis

NPM. 2109020061

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Nanda Rafikhi Azhari Lubis
Tempat dan Tanggal Lahir	: Rantauprapat, 15 Juni 2003
Alamat Rumah	: Jl. Wira Asri, Rantauprapat
Telepon/Faks/HP	: 081269851435
E-mail	: nandalubis535@gmail.com
Instansi Tempat Kerja	:
Alamat Kantor	:

DATA PENDIDIKAN

SD	: Panglima Polem Rantauprapat	TAMAT: 2015
SMP	: Panglima Polem Rantauprapat	TAMAT: 2018
SMA	: SMA N 3 Rantau Utara	TAMAT: 2021

KATA PENGANTAR

Assalamuallaikum Warahmatullahi Wabarakatuh.

Segala puji dan syukur penulis ucapkan kehadirat Allah Subhanahu Wata`alayang telah memberikan anugerahNya dan segala kenikmatan yang luar biasa banyaknya. Sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Analisis Perbandingan Metode Random Forest dan Logistic Regression Terhadap Sentimen Publik pada Ulasan Film Agak Laen"** yang ditujukan untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) Komputer, pada program studi Teknologi Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi – Universitas Muhammadiyah Sumatera Utara. Shalawat dan salam semoga selalu tercurahkan kepada Nabi Muhammad Shalallahu `Alaihi Wasallam, yang telah membawa kita kezaman yang penuh dengan ilmu pengetahuan.

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih 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. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom., Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom, Sekretaris Program Studi Teknologi Informasi
5. Pembimbing Amrullah, S.Kom., M.Kom, yang telah menyediakan waktunya untuk memberikan ilmu dan tentunya dukungan kepada penulis.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu selama perkuliahan.

7. **Kedua orang tua tercinta, Babah Indrawansyah Putra Lubis dan Umi Nelli Juliana, atas doa yang tulus, kasih sayang tanpa batas, serta dukungan moril dan materil yang senantiasa mengiringi setiap langkah penulis. Semangat dan cinta dari Babah dan Umi menjadi kekuatan utama dalam menyelesaikan skripsi ini dan menjalani proses pendidikan hingga tahap akhir.**
8. **Risasti Dwi Ardini, sosok yang selalu mendampingi dengan semangat, kesabaran dan ketulusan, serta rekan seperjuangan dalam ruang kelas yang sama. Terima kasih atas kebersamaan, motivasi, dan dukungan tanpa lelah selama proses perkuliahan dan penyusunan skripsi ini. Kehadiranmu memberikan arti lebih dalam setiap langkah perjalanan ini.**
9. Teman-teman seperjuangan, Raja Iman dan Krisna Aditya, yang telah menjadi tempat berbagi suka dan duka, saling mendukung, serta memberikan semangat dalam menyelesaikan masa perkuliahan ini.
10. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis tidak menutup diri atas segala bentuk saran dan kritik yang bersifat membangun kedepannya. Akhir kata, semoga skripsi ini dapat bermanfaat bagi pembaca dan penulis sendiri.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Medan, 14 Juli 2025

Nanda Rafikhi Azhari Lubis
NPM. 2109020061

ANALISIS PERBANDINGAN METODE RANDOM FOREST DAN LOGISTIC REGRESSION TERHADAP SENTIMEN PUBLIK PADA ULASAN FILM AGAK LAEN

ABSTRAK

Penelitian ini bertujuan untuk membandingkan performa dua metode klasifikasi, yaitu Logistic Regression dan Random Forest, dalam menganalisis sentimen publik terhadap film Agak Laen berdasarkan komentar di media sosial Instagram. Data yang digunakan berjumlah 1.000 komentar yang telah melalui proses praproses teks, seperti pembersihan data, normalisasi, stopword removal, dan stemming. Setiap komentar diberi label sentimen secara manual ke dalam tiga kategori: negatif, netral, dan positif. Setelah dilakukan proses ekstraksi fitur menggunakan metode TF-IDF, data dibagi menjadi data latih dan data uji dengan rasio 90:10. Model kemudian dilatih dan dievaluasi menggunakan metrik evaluasi seperti akurasi dan f1-score. Hasil evaluasi menunjukkan bahwa metode Random Forest memiliki akurasi lebih tinggi sebesar 98% dan f1-score sebesar 0,975, dibandingkan dengan Logistic Regression yang menghasilkan akurasi 96% dan f1-score sebesar 0,94. Berdasarkan hasil tersebut, Random Forest dinilai lebih unggul dalam mengklasifikasikan komentar sentimen terhadap film Agak Laen. Penelitian ini menunjukkan bahwa pemilihan metode klasifikasi yang tepat sangat berpengaruh dalam analisis sentimen publik, terutama pada data yang memiliki distribusi kelas tidak seimbang.

Kata Kunci: Analisis Sentimen; Logistic Regression; Random Forest; TF-IDF; Film Agak Laen.

COMPARATIVE ANALYSIS OF RANDOM FOREST AND LOGISTIC REGRESSION METHODS ON PUBLIC SENTIMENT IN REVIEWS OF THE FILM AGAK LAEN

ABSTRACT

This study aims to compare the performance of two classification methods, Logistic Regression and Random Forest, in analyzing public sentiment toward the film *Agak Laen* based on comments from the social media platform Instagram. The dataset consists of 1,000 comments that underwent text preprocessing, including data cleaning, normalization, stopword removal, and stemming. Each comment was manually labeled into three sentiment categories: negative, neutral, and positive. Feature extraction was conducted using the TF-IDF method, and the data was split into training and testing sets with a 90:10 ratio. The models were then trained and evaluated using performance metrics such as accuracy and F1-score. The evaluation results show that the Random Forest method achieved higher accuracy (98%) and F1-score (0.975) compared to Logistic Regression, which obtained 96% accuracy and an F1-score of 0.94. Based on these results, Random Forest is considered superior in classifying sentiment comments about the film *Agak Laen*. This study highlights that choosing an appropriate classification method significantly affects sentiment analysis outcomes, especially when dealing with imbalanced class distributions.

Keyword: Sentiment Analysis; Logistic Regression; Random Forest; TF-IDF; Agak Laen Film.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Setelah sukses merilis film Ngeri-ngeri Sedap, rumah produksi Imajinari telah kembali menyajikan karya terbarunya yang bertajuk Agak Laen. Agak Laen merupakan film garapan sutradara Muhadkly Acho. Film ber-genre komedi yang diberi sentuhan horor ini telah resmi dirilis sejak tanggal 1 Februari 2024 kemarin. Melalui film Indonesia terbaru Agak Laen, pemirsa akan diajak untuk menyaksikan aksi konyol empat orang penjaga di sebuah rumah hantu (Milagista, 2024).

Film Agak Laen sukses meraup lebih dari 9 juta penonton sejak pertama kali tayang pada 1 Februari 2024 lalu. Bahkan, film ini juga merambah layar lebar di beberapa negara lain, seperti Singapura, Malaysia, Brunei Darussalam, hingga Amerika Serikat (Dwi, 2024).

Film Agak Laen dibuka dengan menampilkan sebuah pasar malam. Salah satu wahana yang ditawarkan oleh pasar malam tersebut adalah sebuah rumah hantu, yang mana di dalamnya terdapat berbagai macam jenis hantu dari Indonesia. Namun, alih-alih menyeramkan dan membuat para pengunjung terkesan, rumah hantu tersebut justru sama sekali tidak menunjukkan kesan yang seram. Bahkan hantu-hantu di dalamnya pun tidak berhasil membuat pengunjung terkejut maupun ketakutan. Situasi tersebut membuat pihak pengelola wahana mencoba sebisa mungkin untuk menghadirkan wahana rumah hantu yang menyeramkan. Hingga akhirnya ada sebuah kejadian yang berhasil membuat panik para petugas wahana rumah hantu. Diketahui bahwa ada salah satu pengunjung yang mengalami gagal

jantung hingga membuatnya kehilangan nyawa karena terlalu terkejut. Para hantu yang bertugas pada saat itu lantas panik dan mencoba untuk menguburkan mayatnya. Setelah kejadian tersebut, wahana rumah hantu menjadi viral karena banyak pengunjung yang mengaku mengalami kejadian mistis saat berkunjung di sana dan mereka pun mendapatkan banyak keuntungan karena banyaknya pengunjung yang datang (Milagista, 2024).

Film Agak Laen ini sangat viral dan menjadi perbincangan masyarakat Indonesia, terutama mengenai komedi yang diberi sentuhan horor serta pengemasan permasalahan sosial yang dikemas dengan apik. Dengan persepsi masyarakat terhadap film Agak Laen ini, terbentuk sentimen masyarakat diantaranya ada masyarakat yang memberi pendapat positif dan negative terhadap penayangan film Agak Laen (Saragih, 2024)

Dalam era digital yang semakin berkembang, media sosial dan platform *review* film telah menjadi sarana utama bagi masyarakat untuk berbagi pendapat dan pengalaman mereka terhadap berbagai film. Ulasan-ulasan ini tidak hanya berfungsi sebagai sumber informasi bagi calon penonton, tetapi juga mencerminkan sentimen publik yang dapat memengaruhi popularitas dan kesuksesan sebuah film. Oleh karena itu, analisis sentimen menjadi sangat penting untuk memahami persepsi masyarakat terhadap film tertentu (Triyantono et al., 2021)

Beberapa penelitian terdahulu telah mengkaji penggunaan metode machine learning untuk analisis sentimen yang membandingkan *Random Forest* dengan *Logistic Regression* dalam klasifikasi data berupa sentiment masyarakat terkait perpindahan ibukota. Dimana hasil penelitian ini menemukan bahwa mayoritas masyarakat Indonesia cenderung memiliki sentimen netral terhadap rencana

pemindahan ibu kota. Dari 1393 data tweet yang dianalisis, sebanyak 1353 termasuk dalam kategori netral, 24 menunjukkan sentimen positif, dan 16 menunjukkan sentimen negatif. Model Random Forest menunjukkan performa lebih baik dibandingkan Logistic Regression, dengan hasil akurasi yang lebih tinggi, menunjukkan bahwa waktu tidak berpengaruh signifikan terhadap klasifikasi sentimen. Hal ini mengindikasikan bahwa masyarakat lebih memilih untuk menyerahkan keputusan kepada pemerintah, dengan asumsi bahwa pemerintah telah mempertimbangkan dampak jangka panjang dari program ini secara matang (Agustina & Hendry, 2021).

Penelitian mengenai perbandingan metode logistic regression, random forest, dan gradient boosting untuk prediksi diabetes menunjukkan hasil yang menarik. Perbandingan antara logistic regression dan random forest menemukan bahwa Random Forest menghasilkan performa terbaik dengan akurasi 77%, precision 74%, recall 83%, dan F1 score 79%. Disusul oleh Logistic Regression dengan akurasi 76%, precision 76%, recall 77%, dan F1 score 77%. Sementara Gradient Boosting berada di posisi terakhir dengan akurasi 75%, precision 74%, recall 77%, dan F1 score 76%. Random Forest unggul karena kemampuannya menangani kompleksitas dan variabilitas data yang tinggi, sedangkan Logistic Regression tetap relevan dalam kondisi dengan hubungan linear dan kebutuhan interpretasi yang jelas. Gradient Boosting meskipun kuat, memerlukan tuning parameter yang cermat untuk menghindari overfitting. Kesimpulannya, Random Forest direkomendasikan untuk prediksi diabetes dalam konteks dataset ini karena keseimbangan antara akurasi tinggi dan ketahanan terhadap overfitting (Setyawan & Wakhidah, 2025).

Pada penelitian yang membahas tentang analisis sentimen ulasan pengguna aplikasi pelayanan masyarakat dengan menggunakan algoritma random forest. Peneliti menyatakan bahwa peneliti berhasil mencapai akurasi tertinggi sebesar 84% setelah dilakukan hyperparameter tuning, meningkat dari akurasi awal sebesar 80%. Keunggulan utama dari model Random Forest dalam penelitian ini terletak pada kemampuannya membangun banyak decision tree secara ensemble, yang memperkuat kemampuan klasifikasi dengan mengurangi risiko overfitting dan meningkatkan generalisasi model. Selain itu, Random Forest juga efektif dalam menangani data teks yang telah dikonversi menjadi vektor fitur melalui teknik TF-IDF, serta menunjukkan performa yang konsisten pada data latih, validasi, dan uji. Keunggulan tersebut menjadikan Random Forest sebagai pilihan yang kuat untuk analisis sentimen, khususnya pada data ulasan yang memiliki variabilitas tinggi (Ardika & Wibawa, 2022).

Penelitian yang menggunakan model Regresi Logistik untuk melakukan analisis sentimen berbasis teks ulasan film, dengan dua skenario klasifikasi, yaitu klasifikasi 10 kelas rating dan 2 kelas sentimen (positif dan negatif). Dataset yang digunakan terdiri dari ulasan 10 film yang diperoleh dari Mendeley Data. Berdasarkan hasil evaluasi, model Regresi Logistik mampu mencapai akurasi tertinggi sebesar 83% pada klasifikasi 2 kelas sentimen dengan metode ekstraksi fitur CountVectorizer dan penerapan dimensionality reduction. Namun, untuk klasifikasi 10 kelas rating, akurasi tertinggi yang diperoleh hanya sebesar 36%, menunjukkan bahwa model ini lebih efektif digunakan dalam kasus klasifikasi biner daripada klasifikasi multi-kelas. Keunggulan utama model Regresi Logistik dalam penelitian ini terletak pada efisiensi komputasi, kemudahan implementasi,

serta kinerjanya yang stabil untuk klasifikasi dua kelas. Model ini mampu memberikan prediksi sentimen yang akurat meskipun menggunakan metode ekstraksi fitur sederhana, dan tetap menunjukkan performa yang baik walaupun data ulasan memiliki karakteristik teks yang beragam. Oleh karena itu, Logistic Regression dinilai cocok untuk diterapkan pada analisis sentimen yang membutuhkan interpretasi cepat dan sederhana, khususnya pada data teks dengan distribusi yang tidak terlalu kompleks (Averina et al., 2022)

Penelitian ini menggunakan 3 metode yang bertujuan untuk menganalisis sentimen masyarakat terhadap film *Sri Asih* dengan menggunakan tiga metode, yaitu Convolutional Neural Network (CNN), K-Nearest Neighbour (KNN), dan Logistic Regression. Data penelitian diambil dari media sosial Twitter sebanyak 8.362 tweet. Hasil penelitian menunjukkan bahwa Logistic Regression berhasil mencapai akurasi tertinggi sebesar 81,5%, mengungguli CNN yang memperoleh akurasi 78,0% dan KNN dengan akurasi 53,5%. Analisis distribusi sentimen menunjukkan bahwa mayoritas opini masyarakat bersifat netral (42,63%), disusul oleh opini positif (36,20%) dan negatif (21,17%). Selain itu, dari hasil analisis emosi diketahui bahwa emosi kebahagiaan (*joy*) merupakan emosi yang paling dominan dalam opini masyarakat terhadap film tersebut. Logistic Regression memiliki beberapa keunggulan dibandingkan metode lainnya. Selain menghasilkan akurasi yang paling tinggi, Logistic Regression juga merupakan model yang sederhana dan tidak membutuhkan banyak sumber daya komputasi. Hal ini membuatnya lebih cepat dan efisien dibandingkan CNN yang kompleks. Logistic Regression juga terbukti lebih stabil dalam menangani data teks dari media sosial yang seringkali mengandung noise. Temuan ini menunjukkan bahwa meskipun

banyak algoritma baru yang lebih kompleks bermunculan, model sederhana seperti Logistic Regression masih sangat relevan dan efektif untuk tugas-tugas analisis sentimen, khususnya dalam konteks opini masyarakat terhadap film *Sri Asih* (Wijaya et al., 2023)

Penelitian yang menggunakan metode SMOTE & Logistic Regression bertujuan untuk menganalisis sentimen komentar terhadap konten tenun ikat NTT di YouTube dengan menggunakan metode SMOTE untuk penyeimbangan data dan Logistic Regression sebagai algoritma klasifikasi. Dari 934 komentar yang dikumpulkan, hasil pelatihan model menunjukkan bahwa Logistic Regression mampu mencapai akurasi sebesar 91% setelah dilakukan proses penyeimbangan data. Model ini menunjukkan performa sangat baik, terutama dalam mengklasifikasikan komentar dengan sentimen negatif, dengan nilai precision mencapai 1.00 dan recall 0.99, serta f1-score sempurna sebesar 1,00. Sementara itu, untuk klasifikasi sentimen netral dan positif, model masih menunjukkan kinerja tinggi dengan f1-score masing-masing 0.88 dan 0.86, walaupun terdapat sedikit kesalahan akibat tumpang tindih kata-kata kunci antar kategori. Keunggulan Logistic Regression dalam penelitian ini terletak pada kemampuannya menghasilkan akurasi yang tinggi serta stabilitas kinerja di semua kelas sentimen, baik positif, netral, maupun negatif. Logistic Regression juga terbukti efisien secara komputasi dan mampu beradaptasi dengan baik terhadap data yang tidak seimbang setelah penerapan metode SMOTE. Kesederhanaan model ini, dikombinasikan dengan teknik ekstraksi fitur TF-IDF, menjadikannya solusi yang efektif untuk analisis sentimen berbasis teks pendek seperti komentar YouTube, tanpa mengorbankan akurasi prediksi. Dengan performa yang konsisten dan ketepatan

klasifikasi yang tinggi, Logistic Regression menjadi pilihan yang kuat dalam konteks penelitian ini (Radjah & Talakua, 2024).

Penelitian bertujuan untuk mengeksplorasi dan membandingkan efektivitas dua metode *machine learning* yang umum digunakan dalam analisis data yaitu *Random Forest* dan *Logistic Regression*. Kedua metode ini memiliki karakteristik yang berbeda dalam hal akurasi, interpretabilitas, dan kompleksitas, sehingga pemilihan metode yang tepat sangat berpengaruh terhadap hasil analisis sentimen.

Melalui penelitian ini, diharapkan dapat ditemukan metode mana yang lebih unggul dalam mengklasifikasikan sentimen publik terhadap ulasan film Agak Laen. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi pada pengembangan ilmu pengetahuan di bidang analisis data, tetapi juga memberikan wawasan praktis bagi para peneliti dan praktisi yang ingin menerapkan teknik analisis sentimen dalam konteks lain.

1.2. Rumusan Masalah

Menurut penulis, rumusan masalah berfungsi sebagai panduan utama dalam pelaksanaan penelitian. Adapun rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana performa model *Random Forest* dan *Logistic Regression* dalam menganalisis sentimen publik pada ulasan film Agak Laen?
2. Bagaimana sentimen pengguna Instagram terhadap film Agak Laen?
3. Seberapa efektif metode *Random Forest* dibandingkan dengan *Logistic Regression* dalam menganalisis sentimen?

1.3. Batasan Masalah

Dalam penelitian ini, untuk menjaga fokus dan kejelasan, terdapat beberapa batasan masalah telah ditetapkan sebagai berikut:

1. Dataset yang digunakan dalam penelitian ini akan dibatasi hanya pada Instagram. Pembatasan ini bertujuan untuk memastikan bahwa analisis dapat dilakukan secara efisien dan akurat.
2. Penelitian ini akan fokus pada klasifikasi sentimen menjadi tiga kategori yaitu positif, negatif, dan netral. Komentar yang tidak jelas atau ambigu tidak akan dimasukkan dalam analisis.
3. Penelitian ini hanya akan membandingkan dua metode analisis sentimen, yaitu *Random Forest* dan *Logistic Regression*.
4. Penelitian ini dibatasi pada 1.000 komentar dari akun resmi Instagram film Agak Laen dan tidak mencakup data dari platform lain.

1.4. Tujuan Penelitian

Penelitian ini memiliki tujuan utama yang ingin dicapai. Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengidentifikasi dan mengklasifikasikan sentimen pengguna Instagram terhadap film Agak Laen berdasarkan komentar yang diberikan. Dengan demikian, penelitian ini bertujuan untuk memberikan gambaran umum mengenai persepsi masyarakat terhadap film tersebut.
2. Membandingkan efektivitas metode *Random Forest* dan *Logistic Regression* dalam menganalisis sentimen pada ulasan film di media sosial. Penelitian ini juga akan mengevaluasi kinerja kedua algoritma.

3. Menyediakan referensi bagi peneliti selanjutnya dalam studi analisis sentimen, khususnya terkait perbandingan model machine learning, serta menjadi acuan dalam pemilihan metode dan pengembangan proses analisis.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah sebagai kontribusi dalam pengembangan ilmu pengetahuan dan penerapannya, dengan rincian sebagai berikut:

1. Dengan melakukan analisis sentimen, penelitian ini dapat mendorong diskusi yang lebih konstruktif mengenai film dan isu-isu sosial yang terkait, serta meningkatkan kesadaran akan pentingnya tanggapan yang positif dan negatif di media sosial.
2. Dengan membandingkan *Random Forest* dan *Logistic Regression*, penelitian ini dapat memperdalam pemahaman mengenai kelebihan dan kekurangan masing-masing metode dalam klasifikasi teks, serta faktor-faktor yang mempengaruhi kinerja algoritma.
3. Dengan menyediakan referensi dan landasan awal, penelitian ini diharapkan dapat membantu peneliti selanjutnya dalam melakukan studi serupa di bidang analisis sentimen dan pengembangan metode klasifikasi.

BAB II

LANDASAN TEORI

2.1. Analisis Sentimen

Analisis sentiment merupakan proses yang digunakan untuk memahami dan mengelompokkan emosi, opini, dan sikap pengguna terhadap suatu produk, layanan, atau merek melalui analisis teks. Proses ini melibatkan pengolahan data dari berbagai sumber, seperti media sosial, ulasan produk, dan survei, untuk menentukan apakah sentimen yang diekspresikan bersifat positif, negatif, atau netral (Kurniawati, 2023).

Gambar 2.1 Jenis Ekspresi Analisis Sentimen

(Sumber: Kurniawati, 2023)

Menurut Penelitian (Septian, 2023), analisis sentimen adalah teknik analisis data yang digunakan untuk memahami perasaan, opini, dan sikap pengguna terhadap suatu merek, produk, atau layanan. Analisis sentimen dilakukan dengan mengumpulkan, mengelompokkan, dan menganalisis data berupa teks dari berbagai sumber, termasuk media sosial, survei, dan ulasan pelanggan. Analisis sentimen dapat digunakan untuk menentukan apakah sentimen yang diekspresikan dalam teks bernada positif, negatif, atau netral. Umumnya, analisis ini melibatkan

penggunaan algoritma dan model pembelajaran mesin untuk memproses teks dan mengidentifikasi kata dan frasa yang menunjukkan sentimen tertentu.

Menurut pendapat yang dikemukakan oleh (Putri, 2024), analisis sentimen memiliki beberapa tujuan utama yang berperan penting dalam memahami opini atau persepsi publik terhadap suatu objek, baik dalam bentuk produk, layanan, maupun isu sosial. Tujuan-tujuan tersebut membantu peneliti maupun pelaku industri dalam mengambil keputusan yang lebih tepat berdasarkan data berbasis teks yang bersifat subjektif. Adapun beberapa tujuan dari analisis sentimen dijelaskan sebagai berikut:

1. Brand Monitoring, Analisis sentimen digunakan untuk memantau reaksi dan opini masyarakat terhadap merek di berbagai platform media sosial. Dengan mengekstrak data dari berbagai sumber, perusahaan dapat memperoleh informasi yang berguna sebagai dasar evaluasi produk dan layanan di masa depan.
2. Riset Pasar, Metode ini mendukung proses riset pasar dengan membantu perusahaan tetap terinformasi mengenai tren dan kebutuhan konsumen saat ini. Melalui analisis data teks yang berkaitan dengan produk atau layanan yang sering dibahas masyarakat di media sosial, perusahaan dapat mengetahui preferensi dan minat pelanggan.
3. Mengukur Kinerja Kampanye, Analisis sentimen juga berfungsi sebagai alat untuk mengukur efektivitas kampanye atau promosi yang sedang berjalan. Tim pemasaran dapat memantau percakapan pengguna di berbagai platform media sosial terkait kampanye tersebut dan mencatat umpan balik dari pengguna sebagai wawasan untuk pengembangan selanjutnya.

Analisis sentimen memberikan berbagai manfaat penting dalam mengolah data berbasis opini, terutama dalam mengidentifikasi kecenderungan emosi pengguna terhadap suatu entitas, baik itu produk, layanan, maupun isu sosial. Manfaat ini mencakup dukungan dalam pengambilan keputusan, evaluasi kepuasan pelanggan, hingga pemantauan reputasi suatu merek atau institusi. Beberapa manfaat utama dari analisis sentimen dapat diuraikan sebagai berikut, sebagaimana dijelaskan oleh (Putri, 2024).

1. Meningkatkan Kepuasan Pelanggan

   Tools sentiment analysis bisa memberikan data secara *real-time*, sehingga jika terdapat isu bisa langsung ditangani. Penanganan yang cepat ini pastinya bisa meningkatkan kepuasan dan loyalitas *customer*.

2. Keunggulan Kompetitif

   Dengan menerapkan *sentiment analysis*, sebuah bisnis bisa selalu *terupdate* perihal *market trend* terbaru. Hal ini menguntungkan secara kompetitif karena dapat selangkah lebih maju dibandingkan kompetitor yang ada di *market*.

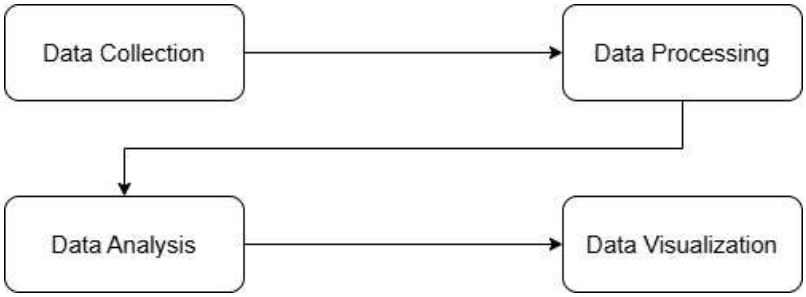
3. Meningkatkan Reputasi *Brand*

   Selain unggul secara kompetitif, implementasi analisis sentimen juga *bernilai* positif bagi reputasi sebuah bisnis. Karena analisis ini dapat membantu memonitor opini masyarakat terkait suatu *brand* di berbagai *platform* sosial media.

4. Product Development Insight

   Manfaat lainnya, melalui analisis teks ini, pemilik usaha bisa tahu apa kebutuhan *customer*. Dengan demikian, perusahaan bisa punya *lead*

information produk apa yang harus dikembangkan selanjutnya berdasarkan kebutuhan pelanggan.



Gambar 2.2 Tahapan Proses Analisis Sentimen

(Sumber: Penulis)

Proses analisis sentimen umumnya dilakukan melalui beberapa tahapan sistematis yang saling berkaitan, dimulai dari pengumpulan data hingga evaluasi hasil klasifikasi. Setiap tahapan memiliki peran penting dalam memastikan kualitas dan akurasi dari hasil analisis yang diperoleh. Adapun proses tersebut telah digambarkan secara visual pada Gambar 2.2, sebagaimana dijelaskan oleh (Putri, 2024) sebagai berikut:

1. Tahap awal yaitu *data collection* atau tahap pengumpulan data yang akan dianalisis. Tahap ini penting karena kualitas sumber data yang dikumpulkan dapat mempengaruhi proses selanjutnya. Pada umumnya terdapat dua cara untuk mendapatkan sumber data yang akan dianalisis. Pertama melalui API *platform* sosial media atau kedua bisa menggunakan *data internal*.
2. Tahap selanjutnya setelah pengumpulan data yaitu pemrosesan sumber data. Tahap *processing* ini menyesuaikan format media dari sumber data yang sudah dikumpulkan. Sehingga, *dataset* yang diproses bisa berupa video, *caption* video, gambar, logo atau yang paling umum yaitu teks. Dari sini, setiap media

akan melalui sejumlah pemrosesan matematik sehingga didapatkan informasi untuk tahapan selanjutnya.

3. Tahap ketiga merupakan tahapan terpenting karena disinilah sumber data akan diolah agar mendapatkan hasil analisis sentimennya. Pada tahap ini terdapat beberapa proses, pertama ada *data training* dimana data akan di-*training* menggunakan sebuah *dataset* khusus yang sudah diberi label sebelumnya. Selanjutnya jika terdapat *dataset* yang menggunakan berbagai bahasa, maka akan melewati proses penerjemahan bahasa. Proses selanjutnya yaitu pembuatan *tag* khusus untuk mengkategorikan data, misal *tag* merek, nama produk dan *tag* lainnya. Setelah itu *dataset* tersebut akan diklasifikasikan per kategori sebelum dilakukan analisis. *Nah*, *part* selanjutnya dan yang terpenting yaitu analisis sentimen. Pada proses ini data akan dianalisis menggunakan teknik-teknik khusus *data science* kemudian akan diberi nilai 0, (-1) atau (-1). Angka 0 melambangkan sentimen netral, (-1) untuk negatif dan (+1) melambangkan sentimen positif.
4. Setelah melalui tahap pemrosesan yang sangat panjang maka *dataset* tadi akan berubah menjadi sebuah informasi atau *insight*. Agar mudah untuk dibaca dan dipahaminya, hasilnya tersebut akan diubah ke bentuk grafik statistik sesuai dengan kebutuhan.

Analisis sentimen dapat diklasifikasikan ke dalam beberapa tipe berdasarkan pendekatan dan cakupan analisis yang digunakan, baik dari segi kedalaman pemahaman terhadap opini maupun konteks linguistik yang dianalisis. Tiap tipe memiliki karakteristik dan penerapan yang berbeda tergantung pada kebutuhan penelitian atau tujuan bisnis tertentu. Adapun beberapa tipe dari analisis

sentimen dapat dijelaskan sebagai berikut, sebagaimana diuraikan oleh (Kurniawati, 2023).

1. *Fine-grained sentiment analysis* adalah tipe ini biasanya digunakan untuk mengukur opini, contohnya penilaian produk di *e-commerce*. Penilaian mulai dari bintang satu yang menyatakan nilai sangat negatif sampai bintang lima yang menyatakan nilai sangat positif.
2. *Emotion detection analysis* adalah analisis untuk mendeteksi perasaan misalnya kesedihan, kemarahan, frustasi, dan kebahagiaan dengan cara mencocokkan teks dengan daftar kata yang sebelumnya sudah ditandai dengan salah satu emosi tersebut. Misalnya, kata "parah" dapat bermakna negatif, tetapi jika seseorang mengatakan "Parah, minuman ini enak banget" maka maknanya menjadi positif.
3. *Fine-grained analysis* adalah tipe analisis dengan memecah kalimat menjadi beberapa bagian penyusunnya kemudian dari bagian tersebut akan dianalisis lebih rinci. Hasil analisis dapat digunakan untuk perbandingan atau komparatif. Misalnya "Produk A lebih baik daripada Produk B" dan dapat menilai sentiment pada subjek tertentu mulai dari "sangat negatif" sampai "sangat positif".
4. *Aspect-based sentiment analysis* mirip dengan *fined-graded analysis* karena menggunakan masukan untuk mencari sentiment negatif atau positif. Misalnya, ketika pelanggan chat dengan *chatbot* "Saya gagal mengirimkan pesan". Chatbot akan mengenali bahwa pelanggan tersebut membutuhkan bantuan dan meneruskan percakapan tersebut untuk mendapatkan bantuan dari operator manusia.

5. *Intent analysis* dapat menentukan apakah suatu pernyataan adalah pertanyaan, keluhan, saran, pendapat, berita, dll. Contohnya Gmail akan dapat mengelompokkan pesan sebagai “Social”, “Promotions”, “Updates”, dan “Forums”.
6. *Multilingual sentiment analysis* dapat dimanfaatkan untuk menganalisis kata-kata dalam berbagai bahasa. Hanya saja, tipe analisis ini cukup sulit karena harus memiliki daftar kata-kata dari bermacam-macam bahasa.

2.2. Random Forest

Random Forest (RF) adalah algoritma *supervised learning* yang digunakan untuk menyelesaikan masalah klasifikasi dan regresi dengan menggabungkan beberapa pohon keputusan (Permana & Bunyamin, 2024).

Random Forest (RF) pertama kali diperkenalkan oleh Leo Breiman (2001). RF merupakan salah satu metode yang dapat meningkatkan hasil akurasi dalam membangkitkan atribut untuk setiap node yang dilakukan secara acak. RF terdiri dari sekumpulan decision tree, dimana kumpulan pohon keputusan ini digunakan untuk mengklasifikasi data ke suatu kelas. Pohon keputusan dibuat dengan menentukan node akar dan berakhir dengan beberapa node daun untuk mendapatkan hasil akhir (Amaliah et al., 2022).

Membentuk pohon keputusan pada metode RF sama dengan proses pada *Classsification and Regression Tree* (CART), hanya saja pada RF tidak dilakukan *prunning* (pemangkasan). Indeks Gini digunakan untuk memilih fitur di setiap simpul internal dari pohon keputusan. Nilai Indeks Gini dapat dihitung sebagai berikut:

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \quad (2.1)$$

dengan p_i merupakan frekuensi relative kelas C_i di dalam set. C_i merupakan kelas untuk i = 1, …, c-1, dan c adalah jumlah kelas yang telah ditentukan. Kualitas split pada fitur k ke dalam subset S_i merupakan jumlah sampel milik kelas C_i, kemudian dihitung sebagai jumlah pertimbangan indikasi Gini dari subset yang dihasilkan. Data dapat dihitung dengan rumus sebagai berikut:

$$Gini_{split} = \sum_{i=0}^{k-1} Gini(S_i) \quad (2.2)$$

dimana n_i merupakan jumlah sampel dalam subset S_i setelah di split dan n merupakan jumlah sampel di node yang dberikan.

Misalkan $\{h(x, \Theta_k), k= 1, \ldots\}$ dimana $\{\Theta_k\}$ merupakan vector random yang independent identically distributed (iid) dan tiap pohon memilih kelas yang paling banyak dari rata rata (majority vote). Untuk RF, batas atas dapat diturunkan untuk kesalahan generalisasi dalam hal dua parameter yang mengukur seberapa kuat pengklasifikasian individu dan ketergantungan diantara keduanya (Amaliah et al., 2022).

Fungsi margin untuk RF adalah

$$mr(X,Y) = P_\Theta(h(X,\Theta) = Y) - \max_{j \neq Y} P_\Theta(h(X,\Theta) = j) \quad (2.3)$$

dan kekuatan himpunan pengklasifikasi $\{h(X,\Theta)\}$ adalah

$$s = E_{x,y} mr(X,Y) \quad (2.4)$$

Dengan asumsi $s \geq 0$, ketidaksamaan Chebychev serta penurunan variansi mr dari fungsi margin untuk metode RF, akan didapatkan persamaan batas atas kesalahan generalisasi sebagai berikut:

$$PE \leq \frac{\bar{\rho}(1-s^2)}{s^2} \tag{2.5}$$

Dimana ρ adalah nilai rata-rata korelasi, yaitu:

$$\bar{\rho} = \frac{E_{\Theta,\Theta'}(\rho(\Theta,\Theta')sd(\Theta)sd(\Theta'))}{E_{\Theta,\Theta'}(sd(\Theta)sd(\Theta'))} \tag{2.6}$$

Sebagaimana dijelaskan oleh (Stihec, 2024), penerapan metode Random Forest terdiri atas serangkaian prosedur sistematis yang dijalankan untuk menghasilkan model yang optimal, sebagai berikut:

1. Siapkan dataset yang sudah dibersihkan dan diproses, terdiri dari fitur (X) dan label (y) untuk supervised learning.
2. Dari dataset pelatihan, secara acak ambil beberapa subset data (dengan pengembalian/with replacement). Setiap subset ini disebut *bootstrap sample*, dan bisa jadi ada data yang terpilih lebih dari sekali atau tidak terpilih sama sekali.
3. Untuk setiap subset data, bangun sebuah pohon keputusan. Pada setiap node (simpul) pohon, hanya sebagian acak dari fitur yang dipertimbangkan untuk pemilihan split terbaik. Hal ini menambah keragaman antar pohon. Dan juga proses ini diulang hingga mencapai syarat berhenti tertentu, misal kedalaman maksimum pohon atau jumlah minimal data di daun pohon.
4. Ulangi proses bootstrapping dan pembuatan pohon sebanyak N kali (sesuai jumlah pohon yang diinginkan). Kumpulan pohon-pohon inilah yang membentuk “forest”.
5. Untuk data baru, setiap pohon menghasilkan prediksi klasifikasi & regresi. Pada klasifikasi, Setiap pohon melakukan voting pada kelas, dan hasil akhir

diambil dari mayoritas suara (*majority voting*). Dan untuk regresi, Hasil akhir diambil dari rata-rata prediksi seluruh pohon.

6. Ukur performa model menggunakan metrik yang sesuai (misal: akurasi, precision, recall untuk klasifikasi dan MAE, MSE untuk regresi).
7. Sesuaikan parameter penting seperti jumlah pohon, kedalaman maksimum pohon, jumlah fitur yang dipilih di setiap split, dan ukuran minimal daun pohon untuk mengoptimalkan performa model.

Berikut rangkuman tentang Langkah-langkah penerapan Random Forest, akan dijelaskan pada tabel 2.1.

Tabel 2.1 Langkah Penerapan Random Forest

Langkah	Penjelasan Singkat	Tujuan
Persiapan Data	Siapkan dataset terstruktur dan bersih	Menyiapkan data yang bersih dan siap dipakai.
Bootstrapping	Ambil subset data secara acak dengan pengembalian	Membuat variasi data untuk tiap pohon.
Bangun Pohon Keputusan	Bangun pohon dengan subset data dan subset fitur acak	Membuat pohon berbeda dengan data dan fitur acak.
Bentuk Forest	Ulangi proses hingga terbentuk kumpulan pohon (forest)	Menggabungkan banyak pohon jadi model kuat.
Prediksi	Voting mayoritas (klasifikasi) atau rata-rata (regresi)	Menghasilkan prediksi akhir dari semua pohon.
Evaluasi Model	Gunakan metrik evaluasi yang sesuai	Mengukur seberapa baik model bekerja.
Tuning Parameter	Sesuaikan parameter model untuk hasil optimal	Mengatur parameter agar performa optimal.

2.3. Logistic Regression

Regresi logistik adalah algoritma pembelajaran mesin terawasi yang banyak digunakan untuk tugas klasifikasi biner, seperti mengidentifikasi apakah email merupakan spam atau bukan dan mendiagnosis penyakit dengan menilai ada tidaknya kondisi tertentu berdasarkan hasil tes pasien. Pendekatan ini menggunakan fungsi logistik (atau sigmoid) untuk mengubah kombinasi linier fitur masukan menjadi nilai probabilitas yang berkisar antara 0 dan 1 (Basit, 2024).

Algoritma Logistic Regression merupakan tipe analisis regresi yang bertujuan untuk menggambarkan hubungan antara variabel dependen dan variabel independen, mengaitkan satu atau lebih variabel bebas dengan variabel terikat dalam bentuk kategori seperti 0 dan 1, benar atau salah. Variabel bebasnya bersifat kategori. Inilah yang membedakan regresi logistik dari regresi berganda atau regresi linear lainnya (Fadhillah et al., 2025)

Dalam persamaan:

$$Ln\left(\frac{p}{1-p}\right) = B_0 + B_1X \tag{2.7}$$

Merupakan persamaan *Logistic Regression*, dan persamaan untuk mencari peluang atau nilai p (Y=1) dapat digunakan rumus sebagai berikut.

$$p = \frac{e(B_0 + B_1X)}{1 - e(B_0 + B_1X)} \tag{2.8}$$

2.4. Perbandingan Random Forest dan Logistic Regression

Random Forest menunjukkan kinerja yang lebih baik dibandingkan Logistic Regression dalam klasifikasi sentimen terkait pemindahan ibukota. Model ini unggul dalam berbagai metrik seperti AUC, akurasi, precision, recall, dan F1-score,

dengan AUC sempurna sebesar 1.000, sementara Logistic Regression hanya 0.382. Hasil ini menunjukkan keunggulan Random Forest yang lebih stabil dan tidak terpengaruh waktu (Agustina & Hendry, 2021).

Random Forest lebih unggul dibandingkan Logistic Regression dalam memprediksi diabetes. Logistic Regression cocok untuk data linear dan mudah diinterpretasikan, namun kurang efektif untuk data kompleks. Sementara itu, Random Forest mampu menangani data non-linear dan menghasilkan akurasi yang lebih tinggi, yaitu 79% dibandingkan 76% pada Logistic Regression. Selain itu, Random Forest juga mencatat precision dan F1 score yang lebih baik, sehingga lebih direkomendasikan untuk prediksi diabetes dalam konteks ini (Setyawan & Wakhidah, 2025).

Logistic Regression merupakan metode klasifikasi linier yang memodelkan hubungan antara variabel independen dengan probabilitas kelas target menggunakan fungsi logistik. Metode ini bekerja dengan baik pada data yang memiliki hubungan linier dan cukup efisien dalam komputasi. Sementara itu, Random Forest adalah metode ansambel berbasis pohon keputusan yang menggabungkan banyak pohon untuk meningkatkan akurasi dan mengurangi overfitting. Hasil penelitian menunjukkan bahwa Random Forest memiliki akurasi yang lebih tinggi dibandingkan Logistic Regression dalam menganalisis sentimen pada data TikTokShop. Hal ini disebabkan karena Random Forest mampu menangani kompleksitas dan variasi data dengan lebih baik, sementara Logistic Regression cenderung kurang efektif pada data yang tidak linier atau memiliki fitur yang saling berinteraksi secara kompleks (Fadhillah et al., 2025)

Tabel 2.2 Perbandingan Random Forest dan Logistic Regression

Aspek	Random Forest	Logistic Regression
Jenis Model	Klasifikasi linear	Ensemble learning (gabungan decision tree)
Kemampuan Data Non-Linear	Kurang baik	Sangat baik
Overfitting	Lebih rentan jika tidak ditangani	Cenderung lebih tahan terhadap overfitting
Interpretasi Model	Mudah diinterpretasi (koefisien jelas)	Sulit diinterpretasi karena kompleksitas pohon ganda
Kecepatan Komputasi	Lebih cepat dan ringan	Lebih lambat karena membangun banyak pohon keputusan
Akurasi (hasil penelitian)	Lebih rendah dibanding Random Forest	Lebih tinggi – menghasilkan kinerja terbaik dalam penelitian
Kelebihan Utama	Sederhana dan cocok untuk data linear	Akurat, tangguh, dan cocok untuk data kompleks

2.5. Python

Menurut (M. Azhar N.H, 2024), Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum pada tahun 1991. Python dikenal dengan sintaksnya yang mudah dipahami dan mendukung berbagai paradigma pemrograman, termasuk pemrograman berorientasi objek, fungsional, dan prosedural. Python biasa dipakai dalam pengembangan situs dan perangkat lunak, membuat analisis data, visualisasi data dan otomatisasi tugas. Karena sifatnya yang relatif mudah dipelajari, bahasa pemrograman ini digunakan secara luas oleh non-programmer seperti ilmuwan dan akuntan yang digunakan untuk mengatur keuangan. Python telah menjadi andalan dalam ilmu data. Bahasa pemrograman ini

memungkinkan analisis data untuk melakukan perhitungan statistik yang rumit, membuat visualisasi data serta algoritma *machine learning*. Python juga bisa digunakan untuk memanipulasi, menganalisis data, dan menyelesaikan berbagai tugas lain terkait data. Selain itu, Python bisa membantu membangun berbagai visualisasi data yang berbeda. Misalnya, grafik garis dan batang, diagram lingkaran, histogram, dan lain sebagainya.

Gambar 2.3 Logo Bahasa Pemrograman Python

(Sumber: (M. Azhar N.H, 2024))

Menurut (Alfarizi et al., 2023), Python merupakan sebuah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yanng sangat populer belakangan ini. Selain itu python juga merupakan bahasa pemrograman yang multi fungsi salah satunya pada bidang Machine Learning dan Deep Learning.

2.6. Machine Learning

Machine Learning (Pembelajaran Mesin) adalah cabang dari kecerdasan buatan (*AI*) yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya seiring waktu tanpa pemrograman eksplisit. Sistem ini dilatih untuk mengenali pola dalam data dan menggunakan pola

tersebut untuk membuat prediksi atau keputusan berdasarkan data baru. Dengan kata lain, mesin dapat belajar dari pengalaman dan menyesuaikan diri untuk menghasilkan hasil yang lebih akurat seiring berjalannya waktu. Dalam *machine learning*, algoritma memproses data untuk menemukan hubungan atau pola tertentu, yang kemudian digunakan untuk memprediksi hasil di masa depan (Meilana, 2025).

Jika Machine Learning di ibaratkan kendaraan bermotor maka data adalah bahan bakar utama dari machine learning, hal ini dikarenakan *Machine Learning* membutuhkan data untuk dapat membuat sebuah metode penyelesaian masalah. Untuk bisa mengaplikasikan teknik-teknik machine learning maka harus ada data. Tanpa data maka algoritma machine learning tidak dapat bekerja. Data yang ada biasanya dibagi menjadi dua, yaitu data training dan data testing. Data training digunakan untuk melatih algoritma, sedangkan data testing digunakan untuk mengetahui performa algoritma yang telah dilatih sebelumnya ketika menemukan data baru yang belum pernah dilihat (Alfarizi et al., 2023).

2.7. Google Colab

Google Colab atau Google Colaboratory, adalah sebuah *executable document* yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui Google Drive. Google Colab memungkinkan penggunanya untuk menjalankan kode Python tanpa perlu melakukan proses instalasi dan *setup* lainnya. Justru, semua keperluan *setting* dan *adjustment* akan diserahkan ke *cloud*. Maka dari itulah, aplikasi ini merupakan tempat yang baik bagi *programmer* yang ingin mengasah pengetahuan mengenai Python. Selain itu,

Google Colaboratory juga terkenal karena dapat mendorong kebutuhan kolaborasi tim. Di mana *notebook* yang akan dibuat nantinya juga dapat diedit secara bersamaan oleh anggota tim lain, seperti halnya menyunting dokumen di Google Documents. Keuntungan terbesar dari Google Colaboratory adalah bahwa Google Colab memiliki kumpulan *built-in-library machine learning* paling populer yang dapat dimuat dengan mudah (Oliver, 2022).

Gambar 2.4 Logo Google Colab

(Sumber: DigitalSkola, 2023)

Menurut (Digital Skola, 2023), terdapat sejumlah manfaat yang dapat diperoleh dari penggunaan Google Colab, di antaranya sebagai berikut:

1. Built In Machine Learning

   Tools ini memiliki fitur *build in machine learning* yang lengkap sehingga pengguna bisa melakukan berbagai aktivitas seperti mengimpor set data atau melakukan evaluasi model melalui beberapa baris kode saja.

2. Berbasis Cloud

   Tools ini juga berbasis *cloud* sehingga pengguna bisa lebih menghemat memori lokal di perangkat laptop atau *smartphone* bahkan pengguna bisa menggunakannya tanpa perlu proses instalasi yang rumit.

3. Fleksibel

tools ini memiliki fungsi yang sangat fleksibel meski hanya diakses melalui smartphone pengguna tetap bisa menggunakannya dengan maksimal dan bisa menjalankan berbagai *source code* yang ada.

4. Tersedia Fitur TPU dan GPU Gratis

   Pengguna tidak perlu melakukan pembayaran apapun bahkan pengguna bisa menggunakan fitur TPU (Tensor Processing Unit) dan GPU (Graphical Processing Unit) untuk berbagai proyek *machine learning* secara gratis juga.

5. Mempermudah Kolaborasi Tim

   Fitur kolaborasi yang disediakan bisa diakses oleh siapapun yang memiliki akses pada *file* tersebut sehingga pekerjaan juga bisa selesai dengan lebih cepat.

Menurut penjelasan dari (Yonatan, 2022), penggunaan Google Colaboratory atau yang biasa dikenal dengan Google Colab memerlukan beberapa tahapan yang harus dipahami terlebih dahulu, mulai dari proses akses awal hingga pelaksanaan kode program di dalamnya. Adapun langkah-langkah penggunaan Google Colab dijabarkan sebagai berikut:

1. Pengguna harus memiliki akun Google untuk bisa menggunakan Google Colab. Apabila belum ada, pengguna bisa membuat akun Google terlebih dahulu.
2. Kunjungi laman colab.research.google.com. Pengguna akan langsung masuk ke halaman utama dari Google Colab.
3. Buat notebook baru dengan mengklik tombol New Notebook di bagian bawah kanan. Pilih antara New Python 3 Notebook atau Python 2 tergantung bahasa pemrograman apa yang hendak pengguna gunakan.

4. Pengguna akan dialihkan ke halaman yang mirip dengan Jupyter Notebook. Setiap notebook yang pengguna buat akan otomatis tersimpan di Google Drive dari akun Google milikmu, jadi jangan khawatir.
5. Untuk menjalankan Python dengan menggunakan GPU atau TPU, pengguna cukup klik Edit, kemudian pilih Notebook Settings. Pada bagian Hardware Accelerator, pilih GPU. Terakhir, klik Save.
6. Pengguna dapat mengunggah data yang akan diolah pada Google Colab dalam format csv. Caranya adalah, cukup klik Upload, pilih file yang akan diunggah, kemudian klik Open.
7. Ketika pengguna membuat file baru pada Google Colab, file tersebut biasanya tidak langsung terhubung dengan computing di Google. Untuk itu, klik panah ke bawah pada opsi Connect, kemudian pilih Connect to a hosted runtime.
8. Pengguna dapat mengubah tampilan notebook sesuai keinginan. Pilih opsi Tools, masuk ke Settings, lalu pilih Site.

Sebagai pendukung penjelasan mengenai langkah-langkah penggunaan Google Colab menurut Penulis, berikut ditampilkan Gambar 2.4 yang memperlihatkan antarmuka utama dari platform tersebut.

Gambar 2.5 Tampilan Interface Google Colab

(Sumber: Penulis)

2.8. IGExporter

Menurut situs (*Export Instagram Followers with IGExport*, n.d.), IGExport merupakan ekstensi Chrome canggih yang dirancang untuk meningkatkan pengalaman Instagram pengguna dengan memberi pengguna kemampuan untuk mengekspor pengikut dan pengikut Instagram pengguna. Alat ini memungkinkan pengguna mengekstrak data berharga yang dapat digunakan untuk berbagai tujuan, seperti perolehan prospek dan pemasaran influencer. Dengan IGExport, Pengguna dapat memperoleh wawasan yang lebih dalam tentang demografi audiens pengguna, pola keterlibatan, dan preferensi. Dengan mengekspor pengikut dan pengikut pengguna, pengguna akan memiliki akses ke banyak informasi yang dapat membantu pengguna membuat keputusan yang tepat tentang strategi konten, audiens target, dan upaya pemasaran Instagram secara keseluruhan.

Penjelasan menurut situs (*Export Instagram Followers with IGExport*, n.d.), salah satu manfaat utama menggunakan IGExport adalah kemampuannya untuk membantu dalam perolehan prospek Instagram. Dengan mengekspor data pengikut

pengguna, pengguna dapat menganalisis profil, minat, dan tingkat keterlibatan mereka. Informasi ini bisa sangat berharga dalam hal mengidentifikasi prospek potensial dan menyesuaikan kampanye pemasaran pengguna. Dengan analisis pengikut IGExport yang komprehensif, pengguna akan dapat mengidentifikasi prospek bernilai tinggi dan terlibat dengan mereka secara lebih efektif.

Gambar 2.6 Logo IGExporter

(Sumber: IGExporter)

Gambar 2.7 Tampilan IGExporter

(Sumber: IGExporter)

2.9. Penelitian Terdahulu

Penelitian terdahulu yang dilakukan oleh (Agustina & Hendry, 2021) membahas tentang analisis sentimen masyarakat terhadap wacana pemindahan ibu kota Indonesia. Tujuannya adalah untuk menganalisis dan mengklasifikasikan sentimen masyarakat berdasarkan tweet terkait isu pemindahan ibu kota. Hasil dari penelitian ini menunjukkan bahwa metode Random Forest menghasilkan akurasi yang lebih tinggi dibandingkan Logistic Regression, serta ditemukan bahwa waktu tidak memengaruhi klasifikasi. Berdasarkan pohon keputusan yang dihasilkan, mayoritas masyarakat menunjukkan sentimen netral, diikuti oleh sentimen positif dan negatif, yang menunjukkan bahwa masyarakat cenderung mempercayakan Keputusan pemindahan ibu kota kepada pemerintah.

Penelitian yang dilakukan oleh (Agustia & Suryono, 2025) membahas tentang nalisis sentimen terhadap judi online dengan tujuan untuk membandingkan kinerja algoritma Random Forest dan Logistic Regression dalam mengklasifikasikan opini publik dari media sosial ke dalam sentimen positif, negatif, dan netral. Untuk mengatasi ketidakseimbangan data, digunakan teknik SMOTE agar distribusi sentimen menjadi seimbang. Hasil evaluasi menunjukkan bahwa Random Forest memberikan performa terbaik dengan akurasi 78%, unggul dalam mengklasifikasikan sentimen positif dan negatif secara konsisten. Setelah penerapan SMOTE, Random Forest tetap stabil dan menunjukkan peningkatan performa tanpa banyak mengorbankan akurasi di kelas lainnya. Sementara itu, Logistic Regression juga mencatat akurasi 77%, tetapi mengalami penurunan presisi dan recall, khususnya pada sentimen netral, bahkan setelah SMOTE diterapkan. Logistic Regression juga menunjukkan kecenderungan salah klasifikasi

yang lebih tinggi dibandingkan Random Forest. Berdasarkan temuan ini, Random Forest direkomendasikan sebagai algoritma yang lebih andal untuk analisis sentimen pada isu sosial seperti judi online, terutama karena kemampuannya mempertahankan performa stabil meskipun dataset mengalami penyeimbangan.

Pada penelitian yang dilakukan oleh (Permana & Bunyamin, 2024) membahas tentang analisis sentimen terhadap ulasan film pada platform IMDb dengan tujuan untuk membandingkan kinerja dua algoritma machine learning, yaitu Logistic Regression dan Random Forest, dalam memprediksi sentimen pengguna. Dataset yang digunakan terdiri dari ulasan film berbahasa Inggris yang diproses melalui tahapan seperti tokenisasi, stemming, penghapusan stop words, serta transformasi teks menjadi fitur numerik menggunakan TF-IDF. Penelitian ini juga mengembangkan antarmuka web agar pengguna dapat memasukkan ulasan dan memperoleh hasil klasifikasi sentimen secara langsung. Berdasarkan hasil evaluasi, Logistic Regression menunjukkan performa yang lebih unggul dibandingkan Random Forest, dengan akurasi prediksi yang lebih tinggi dalam mengklasifikasikan ulasan menjadi sentimen positif atau negatif. Meskipun Random Forest juga memberikan hasil yang memuaskan, Logistic Regression dinilai lebih akurat dan efisien dalam konteks dataset IMDb yang digunakan. Temuan ini memberikan masukan penting bagi pengembang sistem analisis sentimen dalam memilih model klasifikasi yang tepat berdasarkan karakteristik data.

Penelitian yang dilakukan oleh (Britanthia et al., 2020) membahas tentang perbandingan kinerja metode Regresi Logistik dan Random Forest dalam mengklasifikasikan fitur mode (mayor atau minor) pada data audio dari Spotify.

Tujuan utama penelitian ini adalah untuk menentukan metode klasifikasi yang paling efektif berdasarkan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score, serta untuk mengidentifikasi fitur-fitur penting yang memengaruhi klasifikasi mode audio. Penelitian ini menggunakan dataset yang terdiri dari 21.789 lagu yang telah diproses dan dibagi menjadi data latih dan data uji. Hasil penelitian menunjukkan bahwa metode Random Forest memiliki kinerja yang lebih baik dibandingkan Regresi Logistik, dengan akurasi sebesar 72%, sedangkan Regresi Logistik hanya mencapai 65%. Selain itu, Random Forest juga lebih tepat dalam mengidentifikasi fitur-fitur yang relevan secara musikal, seperti speechiness, acousticness, danceability, tempo, dan valence, yang berkaitan erat dengan mode musik. Penelitian ini menyimpulkan bahwa meskipun kedua metode dapat digunakan untuk klasifikasi, Random Forest lebih unggul baik dari segi performa maupun relevansi fitur yang dihasilkan.

Pada penelitian terdahulu yang dilakukan oleh (Junianto et al., 2024) membahas tentang analisis sentimen terhadap ulasan pengguna aplikasi Disdukcapil yang diunduh dari Play Store, dengan tujuan untuk membandingkan performa algoritma Logistic Regression dan Random Forest dalam mengklasifikasikan sentimen menjadi positif dan negatif. Penelitian ini bertujuan untuk membantu pengembang aplikasi dalam memahami persepsi pengguna serta meningkatkan kualitas pelayanan publik. Dataset yang digunakan terdiri dari 18.810 data ulasan, yang diproses melalui tahapan scraping, preprocessing (case folding, tokenisasi, stopwords, dan stemming), serta vektorisasi dengan TF-IDF. Hasil penelitian menunjukkan bahwa Logistic Regression memiliki kinerja sedikit lebih unggul, dengan akurasi 91%, precision 91%, recall 89%, dan F1-score 90%,

serta mampu mengidentifikasi sekitar 75% ulasan sebagai sentimen positif. Di sisi lain, Random Forest mencatat akurasi 90%, precision 92%, recall 86%, dan F1-score 89%, dengan 68,75% ulasan terklasifikasi sebagai sentimen positif. Berdasarkan evaluasi menggunakan K-Fold Cross Validation, Logistic Regression menunjukkan performa yang lebih stabil dan seimbang, meskipun Random Forest unggul pada beberapa aspek recall. Kesimpulannya, Logistic Regression dinilai lebih optimal untuk analisis sentimen dalam konteks ini, namun pemilihan algoritma tetap disesuaikan dengan kebutuhan dan karakteristik data.

Tabel 2.3 Penelitian Terdahulu

No.	Judul	Metode	Hasil
1.	Sentimen Masyarakat Terkait Perpindahan Ibukota Via Model Random Forest dan Logistic Regression (Marta liana Putri Agustina, Hendry, 2021)	Random Forest dan Logistic Regression	Metode Random Forest memiliki akurasi lebih tinggi dibandingkan Logistic Regression, dengan mayoritas sentimen masyarakat terhadap pemindahan ibu kota bersifat netral, menunjukkan kepercayaan kepada pemerintah.
2.	Komparasi Algoritma Naïve Bayes, Random Forest, Dan Logistic	Naïve Bayes, Random Forest,	Random Forest menunjukkan performa terbaik dengan akurasi

	Regresion Untuk Analisis Sentimen Judi Online (Dwi Nanda Agustia, Ryan Randy Suryono, 2025)	dan Logistic Regresion	78% dan klasifikasi sentimen yang konsisten, sementara Logistic Regression mencatat akurasi 77% namun kurang optimal, terutama pada sentimen netral; sehingga Random Forest direkomendasikan untuk analisis sentimen isu judi online.
3.	Perbandingan Logistic Regression dengan Random Forest dalam Memprediksi Sentimen Pada IMDb Moview Review (Nandi Agung Permana, Hendra Bunyamin, 2024)	Logistic Regression dan Random Forest	Logistic Regression unggul dalam memprediksi sentimen ulasan film IMDb dibandingkan Random Forest, dengan akurasi lebih tinggi dan efisiensi lebih baik, sehingga lebih direkomendasikan untuk analisis sentimen pada dataset tersebut.

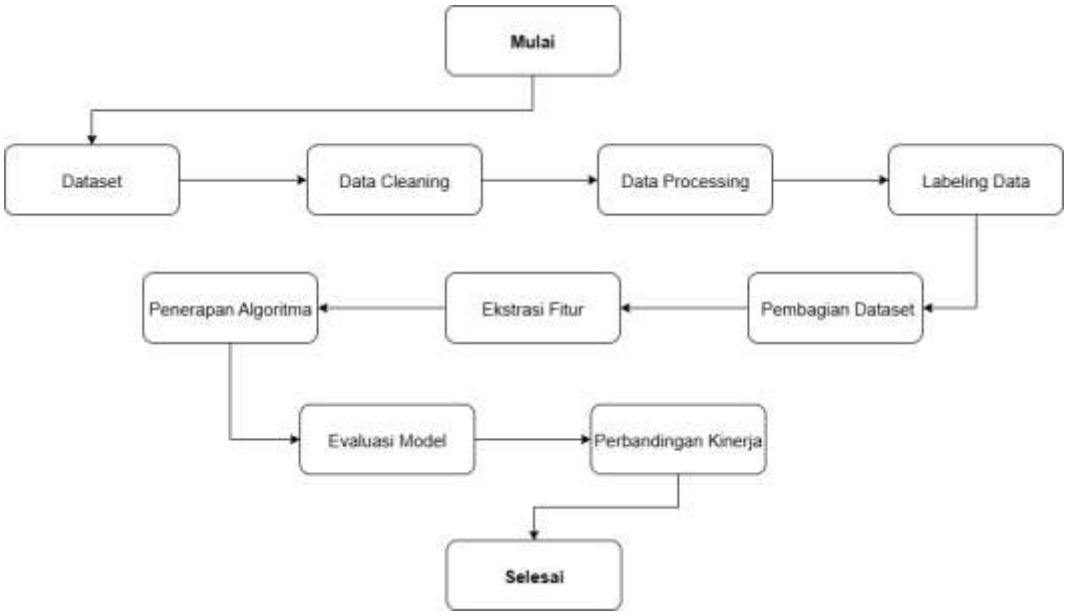
4.	Perbandingan Metode Regresi Logistik dan Random Forest untuk Klasifikasi Fitur Mode Audio Spotify (Lukhia Britanthia Christina Tanujaya, Bambang Susanto, Asido Saragih. 2020)	Logistic Regression dan Random Forest	Random Forest menunjukkan kinerja lebih baik daripada Regresi Logistik dalam klasifikasi mode audio Spotify, dengan akurasi 72% dan kemampuan lebih tepat dalam mengidentifikasi fitur-fitur musik yang relevan.
5.	Komparasi Logistic Regression Dan Random Forestdalam Analisis Sentimen Ulasan Aplikasi Disdukcapil (Haris Junianto, Rujianto Eko Saputro, Bagus Adhi Kusuma, Dhanar Intan Surya Saputra, 2024)	Logistic Regression dan Random Forest	Logistic Regression menunjukkan performa lebih stabil dan akurat (akurasi 91%) dibandingkan Random Forest (akurasi 90%) dalam analisis sentimen ulasan aplikasi Disdukcapil, menjadikannya algoritma yang lebih optimal dalam konteks ini.

BAB III

METODOLOGI PENELITIAN

3.1. Tahap Penelitian

Penelitian ini melalui berbagai tahapan proses. Langkah pertama adalah mengumpulkan data dari komentar Instagram, yang dimana komentar Instagram akan di ekspor menjadi file CSV yang akan menjadi dataset. Langkah yang dilakukan adalah Setelah data di kumpulkan terjadi tahap prapemrosesan, yaitu tahap dimana data diproses kembali. Setelah data diolah terlebih dahulu, maka data tersebut diklasifikasi menggunakan Teknik Random Forest & Logistic Regression. Berikut Langkah-langkah yang dilakukan dalam penelitian ini, yaitu:



Gambar 3.1 Alur Penelitian

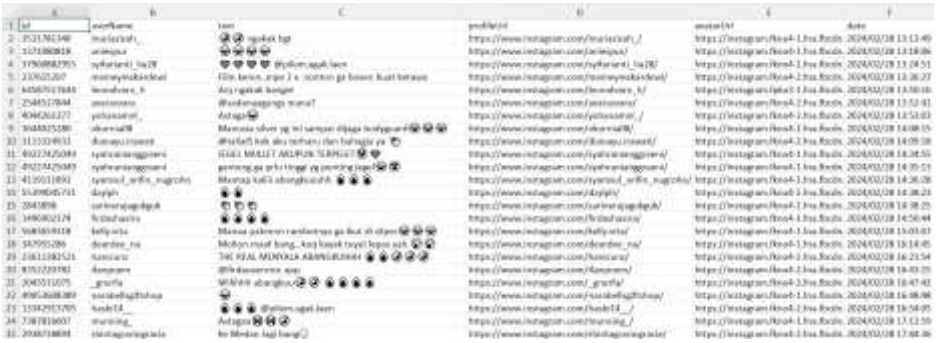
3.1.1. Dataset

Pada tahap awal penelitian ini, proses dimulai dengan pengumpulan dataset berupa ulasan publik terhadap film Agak Laen yang diambil dari postingan pada akun film Agak Laen seperti pada gambar 3.2. Untuk mendapatkan kumpulan data berupa komentar Instagram tentang review film Agak Laen, digunakan tools dari website yang bernama IGExporter.

Gambar 3.2 Postingan Instagram Film Agak Laen

(Sumber: Instagram pilem.agak.laen)

Data yang diperoleh kemudian disimpan dalam format CSV seperti pada gambar 3.3 dan gambar 3.4 agar mudah diproses dalam tahap selanjutnya. Tahap ini bertujuan untuk menyediakan data mentah yang merepresentasikan sentimen publik secara aktual, sehingga dapat digunakan untuk analisis sentimen menggunakan algoritma Random Forest dan Logistic Regression. Kualitas dan keberagaman data ulasan menjadi aspek penting agar model yang dibangun dapat belajar dengan baik dan menghasilkan evaluasi yang akurat.



Gambar 3.3 Tampilan dataset dalam file CSV

Gambar 3.4 Tampilan dataset dalam Google Colab

3.1.2. Data Cleaning

Setelah data dikumpulkan, tahap selanjutnya adalah data cleaning atau pembersihan data. Tahapan ini penting dilakukan untuk menghilangkan elemen-elemen yang tidak relevan atau dapat mengganggu proses analisis, seperti tanda baca, angka, simbol, tautan (URL), serta karakter khusus lainnya seperti pada gambar 3.5. Selain itu, proses lowercasing diterapkan untuk menyamakan format huruf agar model tidak membedakan kata yang sama hanya karena perbedaan huruf kapital. Pembersihan ini juga mencakup penghapusan spasi ganda dan karakter kosong yang tidak diperlukan. Dengan melakukan data cleaning, data menjadi lebih terstruktur dan siap untuk melalui tahap pra-pemrosesan lanjutan seperti tokenisasi, stopword removal, dan ekstraksi fitur. Tahap ini sangat penting agar model klasifikasi dapat bekerja dengan akurat dan tidak terganggu oleh noise dalam data teks.

text	clean_text
"Filmnya keren banget!!!"	"film keren"
"Gak nyangka selucu itu!"	"nyangka lucu"
"⭐⭐⭐"	–

Gambar 3.5 Tampilan data cleaning

3.1.3. Data Processing

Tahap data processing merupakan proses lanjutan setelah data dibersihkan. Pada tahap ini, data teks diolah agar dapat digunakan sebagai input untuk algoritma machine learning. Proses yang dilakukan meliputi tokenisasi, yaitu memecah kalimat menjadi kata-kata individual, kemudian dilanjutkan dengan stopword removal untuk menghilangkan kata-kata umum yang tidak memiliki makna signifikan dalam analisis, seperti “dan”, “yang”, atau “dengan”. Selanjutnya dilakukan proses stemming atau lemmatization untuk mengubah kata ke bentuk dasarnya, seperti “menonton” menjadi “tonton”. Setelah teks selesai diproses, tahap terakhir adalah mengubah data teks menjadi bentuk numerik menggunakan metode ekstraksi fitur yaitu TF-IDF (Term Frequency-Inverse Document Frequency), agar dapat dibaca oleh model klasifikasi. Data yang telah diproses ini kemudian siap untuk digunakan dalam pelatihan model Random Forest dan Logistic Regression.

3.1.4. Labeling Data

Tahap data labeling atau pelabelan data dilakukan untuk memberikan kategori sentimen pada setiap data ulasan yang telah melewati proses pra-pemrosesan. Dalam penelitian ini, pelabelan dilakukan dengan membagi ulasan ke dalam tiga kelas, yaitu positif, negatif, dan netral. Kategori positif mencakup ulasan yang menunjukkan kesan puas, dukungan, atau pujian terhadap film Agak Laen, sedangkan negatif mencakup ulasan yang mengandung kekecewaan, kritik, atau ketidakpuasan. Sementara itu, netral digunakan untuk ulasan yang bersifat informatif, tidak berpihak, atau tidak menunjukkan ekspresi emosional tertentu. Proses pelabelan ini dapat dilakukan secara manual untuk memastikan akurasi, atau semi-otomatis dengan bantuan kamus sentimen dan alat bantu lainnya. Ketepatan pelabelan sangat krusial karena menjadi dasar pembelajaran bagi model klasifikasi. Data yang telah dilabeli kemudian digunakan dalam proses pelatihan dan pengujian algoritma Random Forest dan Logistic Regression.

3.1.5. Pembagian Dataset

Tahap selanjutnya yaitu pembagian dataset, pada tahap ini, dataset yang telah selesai diproses dan dilabeli dibagi menjadi dua bagian, yaitu data latih dan data uji. Pembagian dilakukan dengan proporsi 90:10, di mana 90% digunakan untuk melatih model dan 10% digunakan untuk menguji performanya. Setelah data terbagi, masing-masing subset disimpan dalam variabel terpisah dan siap digunakan pada tahap pelatihan dan evaluasi model. Pembagian ini dilakukan sebelum model dijalankan agar evaluasi yang dihasilkan mencerminkan kinerja model pada data yang belum pernah dikenali sebelumnya.

3.1.6. Ekstrasi Fitur

Setelah proses pembagian dataset selesai, tahap berikutnya adalah ekstraksi fitur, yaitu mengubah data teks yang telah diproses menjadi representasi numerik agar dapat dikenali dan diproses oleh algoritma machine learning. Dalam penelitian ini, digunakan metode dalam ekstraksi fitur yaitu TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF mempertimbangkan tidak hanya frekuensi kata dalam satu dokumen, tetapi juga seberapa umum atau jarang kata tersebut muncul di seluruh dokumen. Hasil dari proses ini adalah matriks fitur yang berisi nilai-nilai numerik yang merepresentasikan karakteristik teks. Matriks ini kemudian digunakan sebagai input dalam proses pelatihan model klasifikasi menggunakan algoritma Random Forest dan Logistic Regression. Pemilihan metode ekstraksi fitur yang tepat sangat mempengaruhi performa model dalam mengenali dan mengklasifikasikan sentimen secara akurat.

3.1.7. Penerapan Algoritma

Setelah data dibersihkan, diproses, dan dibagi, tahap selanjutnya adalah menerapkan algoritma klasifikasi untuk melakukan analisis sentimen. Dalam penelitian ini digunakan dua algoritma, yaitu Random Forest dan Logistic Regression. Masing-masing algoritma diterapkan pada data latih yang telah diekstraksi fiturnya menggunakan metode TF-IDF. Setelah model dilatih, masing-masing diuji menggunakan data uji untuk memprediksi sentimen. Hasil prediksi dari kedua model kemudian disimpan dan akan dievaluasi pada tahap selanjutnya

guna melihat performa dan keakuratan masing-masing algoritma dalam mengklasifikasikan sentimen publik terhadap ulasan film Agak Laen.

3.1.8. Evaluasi Model

Tahap evaluasi model dilakukan untuk mengukur kinerja algoritma Random Forest dan Logistic Regression dalam mengklasifikasikan sentimen ulasan film. Evaluasi dilakukan dengan membandingkan hasil prediksi model terhadap label asli pada data uji. Beberapa metrik evaluasi yang digunakan dalam penelitian ini meliputi akurasi, presisi, recall, dan f1-score. Evaluasi ini bertujuan untuk mengetahui sejauh mana model mampu mengenali pola sentimen secara tepat dan konsisten, serta untuk membandingkan performa kedua algoritma secara objektif. Hasil evaluasi akan menjadi dasar dalam menarik kesimpulan terkait algoritma mana yang lebih efektif dalam menganalisis sentimen publik terhadap film Agak Laen.

3.1.9. Perbandingan Kinerja

Setelah masing-masing model dievaluasi, tahap selanjutnya adalah melakukan perbandingan kinerja antara algoritma Random Forest dan Logistic Regression. Perbandingan dilakukan dengan membandingkan nilai metrik evaluasi seperti akurasi, presisi, recall, dan f1-score yang diperoleh dari kedua model. Hasil dari evaluasi tersebut disajikan dalam bentuk tabel dan grafik untuk mempermudah analisis. Tujuan dari tahap ini adalah untuk mengetahui algoritma mana yang memberikan performa terbaik dalam mengklasifikasikan sentimen ulasan terhadap film Agak Laen. Selain itu, perbandingan ini juga membantu dalam memberikan

rekomendasi model yang paling sesuai untuk diterapkan dalam analisis sentimen dengan karakteristik data serupa.

3.2. Perangkat Penelitian

Perangkat penelitian memiliki peran krusial dalam proses pelaksanaan sebuah penelitian, karena digunakan sebagai sarana utama dalam pengumpulan, analisis, serta interpretasi data. Pada penelitian ini, perangkat yang digunakan mencakup perangkat keras (hardware) dan perangkat lunak (software) yang disesuaikan dengan kebutuhan studi. Deskripsi masing-masing perangkat disajikan sebagai berikut:

Tabel 3.1 Kebutuhan Perangkat Keras

No.	Nama Perangkat	Deskripsi Perangkat
1.	Laptop	ASUS TUF Gaming A15
2.	Storage	512 GB SSD
3.	Processor	AMD Ryzen 7 4000 Series
4.	Graphical Processing Unit (GPU)	NVIDIA GeForce RTX 3050
5.	Random Access Memory (RAM)	16 GB

Tabel 3.2 Kebutuhan Perangkat Lunak

No.	Nama Perangkat	Deskripsi Perangkat
1.	Windows 11 64-bit	Sistem operasi
2.	Python	Bahasa pemrograman yang akan digunakan untuk membangun sistem
3.	Google Chrome	Membantu menyediakan tools untuk melakukan scraping data dan riset terhadap penelitian.
4.	Google Colab	Website yang berguna sebagai tempat pengerjaan penelitian.
5.	IG Comment Exporter	Tools yang digunakan untuk scraping data dari komentar Instagram.

3.3. Jadwal Penelitian

Adapun penelitian ini dilaksanakan dari bulan Januari 2025 – Juni 2025 dan dapat dilihat pada tabel 3.3:

Tabel 3.3 Jadwal Penelitian

NO.	Keterangan	Bulan/Tahun						
		Jan 2025	Feb 2025	Mar 2025	Apr 2025	Mei 2025	Jun 2025	Jul 2025
1.	Pengajuan Judul							
2.	Penyusunan Proposal							
3.	Seminar Proposal							
4.	Pengumpulan Data							
5.	Penyusunan Tugas Akhir							
6.	Bimbingan Tugas Akhir							
7.	Sidang Meja Hijau							

BAB IV

HASIL DAN PEMBAHASAN

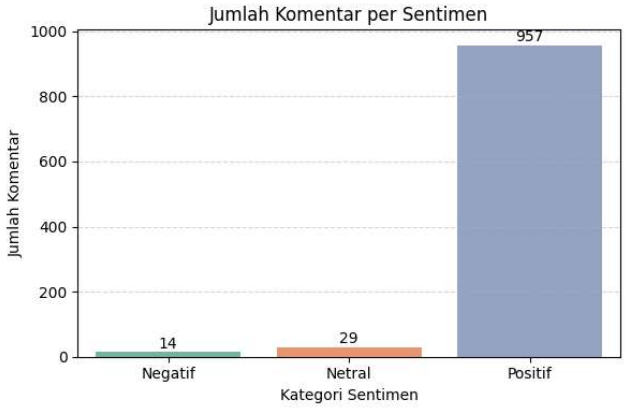
4.1. Deskripsi Data

Data yang digunakan diambil dari Instagram film Agak Laen dengan total jumlah 1.000 komentar dan dikumpulkan untuk dianalisis sentimennya, berfungsi untuk mengetahui film Agak Laen memiliki respon positif, negatif, atau netral guna memahami tanggapan masyarakat terhadap film tersebut. Adapun tampilan sebagian data mentah yang digunakan dalam penelitian ini dapat dilihat pada Gambar 4.1.

	text	username	Sentimen
0	🤣🤣 ngakak bgt	inuriazizah_	Positif
1	👏😂 emang agak laen lah yaa😂😂	ofixokefix	Positif
2	WKWKWK SUMPAH AKU NGAKAK BANGET LIATNYA😭😭😭	ewrikaaaaan	Positif
3	Film keren..mpe 2 x nonton ga bosen buat ketawa	memeymekardewi	Positif
4	Anj ngakak banget	leonalvaro_h	Positif
5	Astaga😂	yohanamnl_	Positif
6	Manusia silver yg ini sampai dijaga bodyguard😂😂😂	okurnia08	Positif
7	@talial5 kok aku terharu dan bahagia ya 👏	dianayu.irawati	Positif
8	JEGEL MULLET AKUPUN TERPELET😍❤️	syahranianggraeni	Positif
9	ganteng ga prlu tinggi yg penting jegel😂😍	syahranianggraeni	Positif

Gambar 4.1 Tampilan data mentah

Data tersebut terbagi ke dalam tiga kategori sentimen, yaitu positif, netral, dan negatif. Dari keseluruhan data. sebanyak 957 komentar tergolong dalam sentimen positif, 29 komentar dalam kategori netral, dan 14 komentar termasuk sentimen negatif. Adapun tampilan visual dari pembagian kategori sentimen tersebut dapat dilihat pada Gambar 4.2.

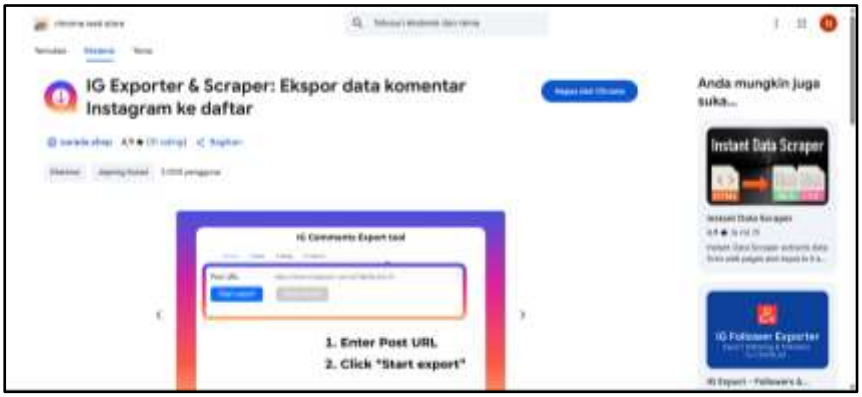


Gambar 4.2 Jumlah Kategori Sentimen

4.2. Penambangan Data

Penambangan data komentar dalam penelitian ini dilakukan dengan mengambil data dari platform media sosial Instagram, khususnya dari postingan akun resmi film Agak Laen. Proses pengambilan komentar dilakukan menggunakan ekstensi browser Google Chrome bernama IG Comments Export Tool atau IG Exporter. Adapun tahapan penambangan data meliputi:

1. Akses Chrome Web Store, kemudian cari ekstensi bernama IG Exporter seperti yang ditampilkan pada Gambar 4.3, lalu tambahkan ekstensi tersebut ke Chrome.



Gambar 4.3 IG Comment Export Tool

2. Kunjungi postingan Instagram dari film Agak Laen, lalu salin (copy) link dari postingan tersebut.

3. Jalankan ekstensi IG Exporter yang telah terpasang, tempelkan (paste) link postingan ke kolom yang tersedia, kemudian klik “Start Export” dan tunggu beberapa saat hingga proses ekspor selesai.
4. Setelah selesai, klik “Save to CSV” untuk menyimpan hasil ekspor. Terakhir, ganti nama file menjadi “KomenIG.csv” agar mudah digunakan dalam proses analisis.

Berikut ini adalah contoh hasil penambangan data yang ditampilkan pada Tabel 4.1. Contoh-contoh yang ditampilkan dipilih berdasarkan beberapa pertimbangan. Pertama, penulis memastikan bahwa ulasan yang dipilih relevan dengan permasalahan penelitian, seperti adanya komentar yang mengandung sentimen bias serta komentar yang hanya terdiri dari emoji. Kedua, penulis menyajikan contoh ulasan dari pengguna yang mewakili beragam kategori sentimen, yaitu sentimen positif, negatif, dan netral.

Tabel 4.1 Contoh Hasil Proses Penambangan Data

No.	Username	Komentar	Sentimen
1.	inuriazizah_	🤣🤣 ngakak bgt	Positif
2.	Ofixokefix	👏😂 emang agak laen lah yaa😂😂	Positif
3.	Ewrikaaaaan	WKWKWK SUMPAH AKU NGAKAK BANGET LIATNYA😭😭😭	Positif
4.	Nbilhmrhm	Weeeh keren banget kalian bang, TERIMA KASIH abang-abangkuh	Positif

		@borisbokir_ @indrajegel @bene_dion @okirengga33 😍😍😍😍😍🤣🤣	
5.	rendy.p.97	Kenapa sadana menghilang, padahal bilangnya mau ikutan	Negatif

Penambangan data dilakukan pada tanggal 23 Mei 2025. Dari hasil proses tersebut, diperoleh sebanyak 2.429 komentar yang berasal dari postingan Instagram film Agak Laen. Selanjutnya, dilakukan proses penyaringan atau filter terhadap data tersebut, hingga diperoleh 1.000 komentar yang digunakan sebagai data akhir dalam penelitian ini.

4.3. Pemanggilan Dataset

Setelah proses pengumpulan data selesai dilakukan, seluruh komentar yang diperoleh diekspor dan disimpan dalam bentuk file berformat CSV. Berikut ini adalah langkah-langkah pemanggilan dataset ke dalam Google Colab agar data dapat digunakan dalam proses analisis selanjutnya:

1. File dataset tersebut kemudian diunggah ke Google Drive agar dapat dengan mudah diakses dan digunakan dalam lingkungan kerja Google Colab.
2. Untuk memanggil dataset ke dalam Google Colab, langkah yang dilakukan adalah menyalin (copy) jalur atau path file dari Google Drive, lalu memasukkannya ke dalam script Python.
3. Pemanggilan ini dilakukan agar data dapat dibaca dan diolah menggunakan berbagai library pemrosesan data. Adapun script pemanggilan dataset ditampilkan pada Gambar 4.4.

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Machine Learning - IT SMT 7/Skripsi/komenIG.csv') # Sumber File
```

Gambar 4.4 Script Pemanggilan Dataset

4.4. Import Library dan Tools

Setelah dataset berhasil disimpan dan siap digunakan, langkah selanjutnya adalah melakukan import library dan tools yang dibutuhkan untuk proses analisis. Tahap ini penting karena library dan tools tersebut menyediakan berbagai fungsi yang diperlukan, seperti manipulasi data, pembersihan teks, prapemrosesan, hingga implementasi dan evaluasi model machine learning.

Penulis menggunakan beberapa library dan tools yang umum digunakan dalam analisis teks berbasis Python. Tampilan script instalasi library dan tools eksternal dapat dilihat pada Gambar 4.5, sedangkan script import library yang digunakan dalam kode Python ditampilkan pada Gambar 4.6.

```
!apt-get install graphviz -y # Menginstal Graphviz, alat bantu untuk visualisasi struktur pohon keputusan (.dot format)
!pip install pydot # Menginstal pydot, library Python untuk membaca dan menggambar file .dot dari Graphviz
!pip install Sastrawi # Menginstal library Sastrawi untuk pemrosesan teks Bahasa Indonesia (steeming & stopword removal)
```

Gambar 4.5 Script Instalasi Library dan Tools Eksternal

```
# Manipulasi dan Pengolahan Data
import numpy as np # Untuk operasi numerik dan array
import pandas as pd # Untuk manipulasi data dan pembuatan DataFrame
import re # Untuk regex (membersihkan teks)

# Visualisasi data
import matplotlib.pyplot as plt # Untuk membuat grafik visualisasi
import seaborn as sns # Untuk visualisasi yang lebih menarik berbasis matplotlib
from IPython.display import Image, display # Untuk menampilkan gambar (misalnya tree.png) di Colab

# Natural Language Processing (NLP)
#NLTK (Bahasa Inggris/umum)
from nltk.corpus import stopwords # Untuk mengakses daftar stopwords (kata umum yang dibuang)
from nltk.stem import PorterStemmer # Untuk stemming kata (mengubah ke bentuk dasar)
#Sastrawi (Bahasa Indonesia)
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Ekstraksi Fitur Teks
from sklearn.feature_extraction.text import TfidfVectorizer # Untuk mengubah teks menjadi angka (TF-IDF)

# Pra-pemrosesan dan Pembagian Dataset
from sklearn.model_selection import train_test_split  # Untuk membagi data latih dan data uji
from sklearn.preprocessing import LabelEncoder # Untuk mengubah label string ke angka
from sklearn.preprocessing import StandardScaler  # Untuk standarisasi data numerik (jarang dipakai di TF-IDF)

# Model Klasifikasi (Machine Learning)
from sklearn.linear_model import LogisticRegression # Model klasifikasi Logistic Regression
from sklearn.ensemble import RandomForestClassifier # Model klasifikasi Random Forest

# Evaluasi Model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report # Untuk evaluasi model

# Visualisasi dan Ekspor Pohon Keputusan
import pydot # Untuk membuat dan menampilkan grafik dari file .dot (pohon keputusan)
from sklearn.tree import export_graphviz # Untuk mengekspor pohon keputusan ke format .dot
```

Gambar 4.6 Script Import Library dalam Kode Python

4.5. Preprocessing Data

Setelah data berhasil dikumpulkan dan dimuat ke dalam Google Colab, langkah selanjutnya adalah melakukan pemrosesan data. Proses ini bertujuan untuk membersihkan dan menyiapkan data agar dapat digunakan dalam tahap analisis sentimen secara optimal.

4.5.1. Data Cleaning

Penulis membersihkan teks dari elemen-elemen yang tidak relevan seperti tanda baca, angka, simbol, URL, emoji, dan spasi berlebih. Teks juga diubah menjadi huruf kecil agar seragam dan siap dianalisis. Berikut ini merupakan tahapan data cleaning:

1. Penulis menjalankan script untuk memilih tiga kolom utama, yaitu text, username, dan sentiment, dengan menggunakan script yang ditampilkan pada Gambar 4.7.
2. Menjalankan script untuk membersihkan teks dari tanda baca, angka, simbol, URL, emoji, dan diubah menjadi huruf kecil. dengan menggunakan script yang ditampilkan pada Gambar 4.8.
3. Penulis membuat kolom baru bernama “cleaned_comment” yang berisi hasil pembersihan dari kolom “text”.

Berikut merupakan script yang digunakan oleh penulis dalam proses data cleaning, yaitu untuk membersihkan teks dari elemen-elemen yang tidak relevan sebelum dilakukan analisis lebih lanjut.

```
df = df[['text', 'username', 'Sentimen']]
```

Gambar 4.7 Script Pemilihan Kolom

```
def clean_text(text):
    text = text.lower()  # huruf kecil semua
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)  # hapus mention
    text = re.sub(r'#\S+', '', text)  # hapus hashtag
    text = re.sub(r'http\S+', '', text)  # hapus link
    text = re.sub(r'[^\w\s]', '', text)  # hapus tanda baca
    text = re.sub(r'\d+', '', text)  # hapus angka
    text = re.sub(r'(.)\1{2,}', r'\1', text)      # Ubah lebih dari 2 huruf sama → 1
    text = re.sub(r'(.)\1{1}$', r'\1', text)      # Ubah sisa akhir yang dobel jadi 1
    text = re.sub(r'(.)\1+', r'\1', text)         # 🔥 semua huruf berulang → satu huruf
    text = re.sub(r'\s+', ' ', text).strip()  # hapus spasi berlebih
    return text


df['cleaned_comment'] = df['text'].astype(str).apply(clean_text)
```

Gambar 4.8 Script Data Cleaning

Setelah melakukan pembersihan data, penulis membuat kolom baru bernama “cleaned_comment” dimana berfungsi sebagai pembeda antara kolom data mentah dan data setelah melakukan pembersihan. Berikut contoh hasil

komentar sebelum dan sesudah melalui proses pembersihan data yang terdapat pada tabel 4.2.

Tabel 4.2 Contoh Hasil Proses Data Cleaning

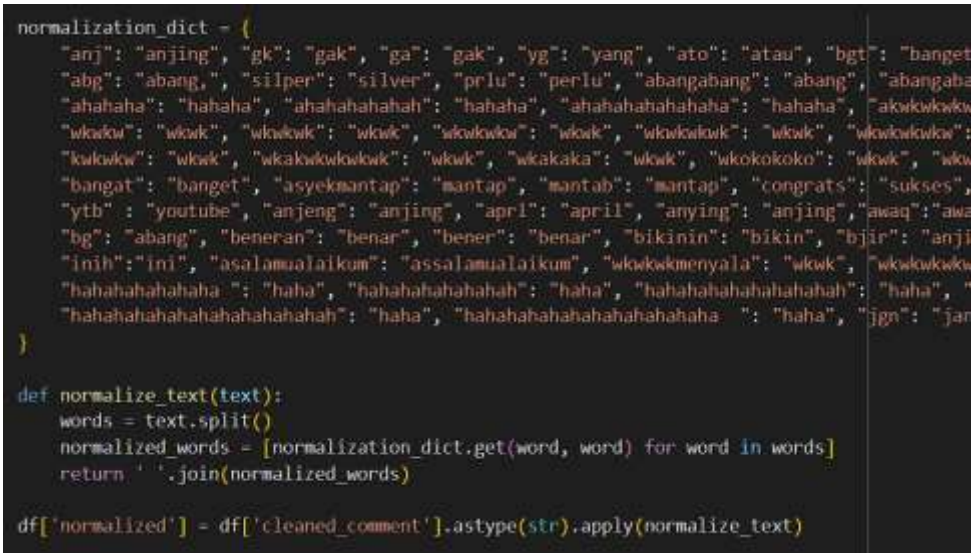
No.	Username	Sebelum (Mentah)	Sesudah (Cleaning)
1.	inuriazizah_	🤣🤣 ngakak bgt	ngakak bgt
2.	Ofixokefix	👏😂 emang agak laen lah yaa😂😂	emang agak laen lah ya
3.	Ewrikaaaaan	WKWKWK SUMPAH AKU NGAKAK BANGET LIATNYA😭😭😭	wkwkwk sumpah aku ngakak banget liatnya
4.	Nbilhmrhm	Weeeh keren banget kalian bang, TERIMA KASIH abang-abangkuh @borisbokir_ @indrajegel @bene_dion @okirengga33 😍😍😍😍😍🤣🤣	weh keren banget kalian bang terima kasih abangabangkuh
5.	rendy.p.97	Kenapa sadana menghilang, padahal bilangnya mau ikutan	kenapa sadana menghilang padahal bilangnya mau ikutan

4.5.2. Normalization

Penulis melakukan normalisasi data dengan mengganti kata-kata tidak baku, singkatan, atau bahasa gaul menjadi bentuk kata yang lebih formal dan konsisten. Langkah ini bertujuan untuk menyamakan berbagai variasi penulisan agar lebih mudah dipahami oleh model. Berikut ini merupakan tahapan normalisasi data:

1. Penulis menyiapkan kamus normalisasi yang berisi pasangan kata tidak baku atau singkatan dengan padanan kata formalnya, seperti “gk” menjadi “gak” dan “bgt” menjadi “banget”.
2. Menjalankan script untuk normalisasi data dengan menggunakan script yang ditampilkan pada Gambar 4.9.
3. Penulis membuat kolom baru bernama “normalized” yang berisi hasil pembersihan dari kolom “cleaned_comment”.

Berikut merupakan script yang digunakan oleh penulis dalam proses normalisasi data, untuk mengganti kata-kata tidak baku menjadi bentuk kata yang lebih formal agar memudahkan model dalam memahami konteks kalimat secara lebih akurat.



Gambar 4.9 Script Normalisasi Data

Setelah melakukan normalisasi data, penulis membuat kolom baru bernama “normalized” dimana berfungsi sebagai pembeda antara kolom data cleaning dan data setelah melakukan normalisasi. Berikut contoh hasil komentar sebelum dan sesudah melalui proses normalisasi data yang terdapat pada tabel 4.3.

Tabel 4.3 Contoh Hasil Proses Normalisasi Data

No.	Username	Sebelum (Cleaning)	Sesudah (Normalized)
1.	inuriazizah_	ngakak bgt	ngakak banget
2.	Ofixokefix	emang agak laen lah ya	emang agak laen lah ya
3.	Ewrikaaaaan	wkwkwk sumpah aku ngakak banget liatnya	wkwk sumpah aku ngakak banget liatnya
4.	Nbilhmrhm	weh keren banget kalian bang terima kasih abangabangkuh	weh keren banget kalian abang terima kasih abang
5.	rendy.p.97	kenapa sadana menghilang padahal bilangnya mau ikutan	kenapa sadana menghilang padahal bilangnya mau ikutan

4.5.3. Tokenizing

Penulis melakukan tokenisasi data dengan memecah setiap kalimat atau teks ulasan menjadi potongan-potongan kata atau token. Tokenisasi membantu mengubah teks menjadi format yang dapat dikenali dan diproses oleh algoritma machine learning. Berikut ini merupakan tahapan tokenisasi data.

1. Penulis menjalankan proses tokenisasi untuk memecah setiap kalimat atau teks dalam kolom “normalized” menjadi daftar kata-kata (token) secara terpisah, dengan menggunakan script yang ditampilkan pada Gambar 4.10.
2. Hasil dari proses tokenisasi disimpan ke dalam kolom baru bernama “tokens” yang berisi kumpulan token dari setiap komentar.

Berikut merupakan script yang digunakan oleh penulis dalam proses tokenisasi data, yaitu untuk memecah teks ulasan menjadi kata-kata (token) agar lebih mudah dianalisis oleh model.

```
# Fungsi tokenisasi sederhana (split berdasarkan spasi)
def tokenize(text):
    return text.split()

# Terapkan ke kolom 'stemmed'
df['tokens'] = df['normalized'].astype(str).apply(tokenize)
```

Gambar 4.10 Script Tokenisasi Data

Setelah melakukan tokenisasi data, penulis membuat kolom baru bernama "tokens" dimana berfungsi sebagai pembeda antara kolom normalisasi data dan data setelah melakukan tokenisasi. Berikut contoh hasil komentar sebelum dan sesudah melalui proses tokenisasi data yang terdapat pada tabel 4.4.

Tabel 4. 4 Contoh Hasil Proses Tokenisasi Data

No.	Username	Sebelum (Normalized)	Sesudah (Tokens)
1.	inuriazizah_	ngakak banget	[ngakak, banget]
2.	Ofixokefix	emang agak laen lah ya	[emang, agak, laen, lah, ya]
3.	Ewrikaaaaan	wkwk sumpah aku ngakak banget liatnya	[wkwk, sumpah, aku, ngakak, banget, liatnya]
4.	Nbilhmrhm	weh keren banget kalian abang terima kasih abang	[weh, keren, banget, kalian, abang, terima, kasih, abang]
5.	rendy.p.97	kenapa sadana menghilang padahal bilangnya mau ikutan	[kenapa, sadana, menghilang, padahal, bilangnya, mau, ikutan]

4.5.4. Stopword Removal

Penulis menghapus kata-kata umum yang tidak memiliki pengaruh besar dalam analisis, seperti yang, dan, di, dan sejenisnya. Tujuannya agar model fokus pada kata-kata penting. Berikut ini merupakan tahapan dari stopword removal:

1. Penulis menjalankan proses penghapusan kata-kata umum (stopword) seperti “yang”, “dan”, “di” yang tidak memiliki pengaruh besar terhadap analisis dengan menggunakan script yang ditampilkan pada Gambar 4.11.
2. Pada tahap ini, penulis memilih kata “agak” tidak dihapus karena merupakan bagian penting dari judul film Agak Laen.
3. Penulis membuat kolom baru bernama “no_stopword” yang berisi hasil pembersihan dari kolom “tokens”.

Berikut merupakan script yang digunakan oleh penulis dalam proses stopword removal, yaitu untuk menghapus kata-kata umum yang tidak memiliki makna penting agar model lebih fokus pada kata-kata yang berpengaruh terhadap sentimen.

```
import nltk
nltk.download('stopwords')

from nltk.corpus import stopwords
stopwords_ind = set(stopwords.words('indonesian'))
stopwords_ind.discard('agak')

#nama_orang = ['boris', 'bokir', 'ernest', 'jegel', 'indra', 'sadana', 'agung', 'bene']
kata_tak_bermakna = {'ach', 'aci', 'ah', 'al', 'amago', 'aw', 'be', 'an', 'apas', 'akh', 'weh'}

def remove_stopwords(tokens):
    tokens = [word.lower() for word in tokens]  # pastikan lowercase
    tokens = [word for word in tokens if word not in stopwords_ind]
    #tokens = [word for word in tokens if word not in nama_orang]
    tokens = [word for word in tokens if word not in kata_tak_bermakna]
    return ' '.join(tokens)

df['no_stopwords'] = df['tokens'].apply(remove_stopwords)
```

Gambar 4.11 Script Stopword Removal

Setelah melakukan stopword removal, penulis membuat kolom baru bernama “no_stopword” dimana berfungsi sebagai pembeda antara kolom tokenisasi data dan data setelah melakukan stopword removal. Berikut contoh hasil komentar sebelum dan sesudah melalui proses stopword removal yang terdapat pada tabel 4.5.

Tabel 4. 5 Contoh Hasil Proses Stopword Removal

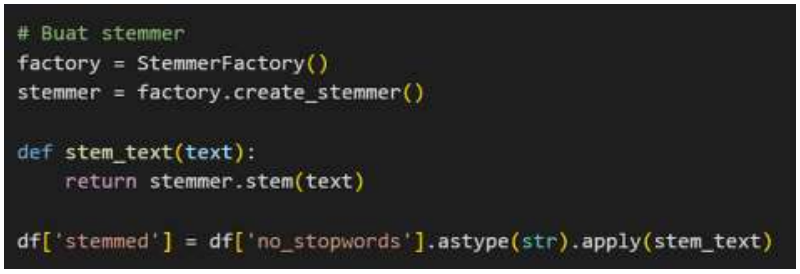
No.	Username	Sebelum (Tokens)	Sesudah (Stopword)
1.	inuriazizah_	[ngakak, banget]	ngakak banget
2.	Ofixokefix	[emang, agak, laen, lah, ya]	emang agak laen ya
3.	Ewrikaaaaan	[wkwk, sumpah, aku, ngakak, banget, liatnya]	wkwk sumpah ngakak banget liatnya
4.	Nbilhmrhm	[weh, keren, banget, kalian, abang, terima, kasih, abang]	keren banget abang terima kasih abang
5.	rendy.p.97	[kenapa, sadana, menghilang, padahal, bilangnya, mau, ikutan]	sadana menghilang bilangnya ikutan

4.5.5. Stemming

Penulis melakukan stemming untuk mengubah kata menjadi bentuk dasarnya agar kata-kata dengan makna serupa dikenali sebagai satu kata. Proses ini membantu meningkatkan konsistensi data dan dilakukan menggunakan library Sastrawi. Berikut ini merupakan tahapan dari proses stemming:

1. Penulis menjalankan proses stemming untuk mengubah setiap kata menjadi bentuk dasar atau akar katanya, seperti “menonton” menjadi “tonton”, menggunakan script yang ditampilkan pada Gambar 4.12.
2. Proses ini dilakukan dengan memanfaatkan library *Sastrawi* yang sesuai untuk Bahasa Indonesia.
3. Penulis membuat kolom baru bernama “stemmed” yang berisi hasil stemming dari kolom “no_stopword”.

Berikut merupakan script yang digunakan oleh penulis dalam proses stemming, yaitu untuk mengubah kata menjadi bentuk dasar agar model lebih mudah memahami makna secara konsisten.



```
# Buat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return stemmer.stem(text)

df['stemmed'] = df['no_stopwords'].astype(str).apply(stem_text)
```

Gambar 4.12 Script Stemming

Setelah melakukan stemming, penulis membuat kolom baru bernama “stemmed” dimana berfungsi sebagai pembeda antara kolom stopword removal dan data setelah melakukan stemming. Berikut contoh hasil komentar sebelum dan sesudah melalui proses stemming yang terdapat pada tabel 4.6.

Tabel 4. 6 Contoh Hasil Proses Stemming

No.	Username	Sebelum (Stopword)	Sesudah (Stemming)
1.	inuriazizah_	ngakak banget	ngakak banget
2.	Ofixokefix	emang agak laen ya	emang agak laen ya

3.	Ewrikaaaaan	wkwk sumpah ngakak banget liatnya	wkwk sumpah ngakak banget liat
4.	Nbilhmrhm	keren banget abang terima kasih abang	keren banget abang terima kasih abang
5.	rendy.p.97	sadana menghilang bilangnya ikutan	sadana hilang bilang ikut

4.6. Labeling Data

Setelah seluruh proses data processing selesai, penulis melakukan labeling data dengan mengubah nilai sentimen dari bentuk teks menjadi format numerik agar dapat diproses oleh algoritma machine learning. Misalnya, “positif” menjadi 2, “netral” menjadi 1, dan “negatif” menjadi 0. Berikut ini merupakan tahapan dari labeling data:

1. Penulis melakukan encoding label sentimen menggunakan label encoder dari library sklearn, yang mengubah kategori sentimen menjadi nilai numerik. Script yang digunakan ditampilkan pada Gambar 4.13.
2. Hasil dari proses ini disimpan dengan diperbarui pada kolom Sentimen yang sebelumnya berisi label dalam bentuk teks.

Berikut merupakan script yang digunakan oleh penulis dalam proses labeling data, yaitu untuk mengubah kategori sentimen dalam bentuk teks menjadi format numerik. Selain itu, penulis juga menyertakan script untuk menganalisis distribusi jumlah komentar berdasarkan masing-masing label sentimen.

```
#LABEL ENCODING✓

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sentimen'] = le.fit_transform(df['Sentimen'])  # pastikan kolomnya benar
#0 = negatif, 1 = netral, 2 = positif
```

Gambar 4.13 Script Labeling Data

```
# Hitung jumlah per sentimen
sentiment_counts = df['Sentimen'].value_counts().sort_index()

# Label kategori
label_sentimen = ['Negatif', 'Netral', 'Positif']

# Tampilkan jumlah komentar per kategori
for i, count in sentiment_counts.items():
    print(f"Jumlah komentar {label_sentimen[i]} (Label {i}): {count}")

# Visualisasi
plt.figure(figsize=(6, 4))
sns.barplot(x=label_sentimen, y=sentiment_counts.values, palette='Set2')

# Tambahkan angka di atas batang
for i, count in enumerate(sentiment_counts.values):
    plt.text(i, count + 2, str(count), ha='center', va='bottom')

plt.title('Jumlah Komentar per Sentimen')
plt.xlabel('Kategori Sentimen')
plt.ylabel('Jumlah Komentar')
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

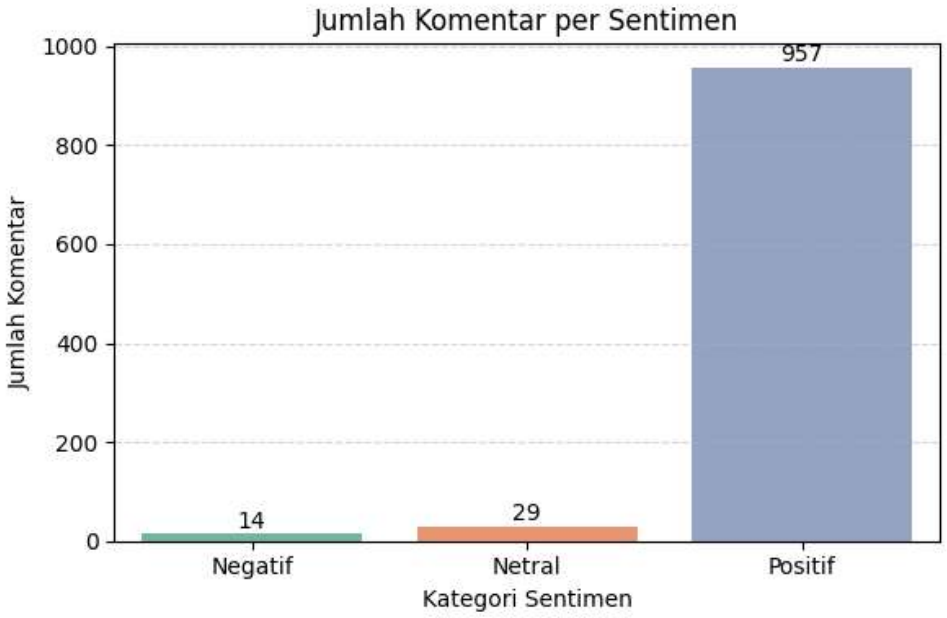
Gambar 4.14 Script Distribusi Jumlah Komentar

```
# Pie chart
plt.figure(figsize=(6,6))
plt.pie(sentiment_counts.values, labels=label_sentimen, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('Set2'))
plt.title('Distribusi Persentase Sentimen')
plt.axis('equal')
plt.show()
```

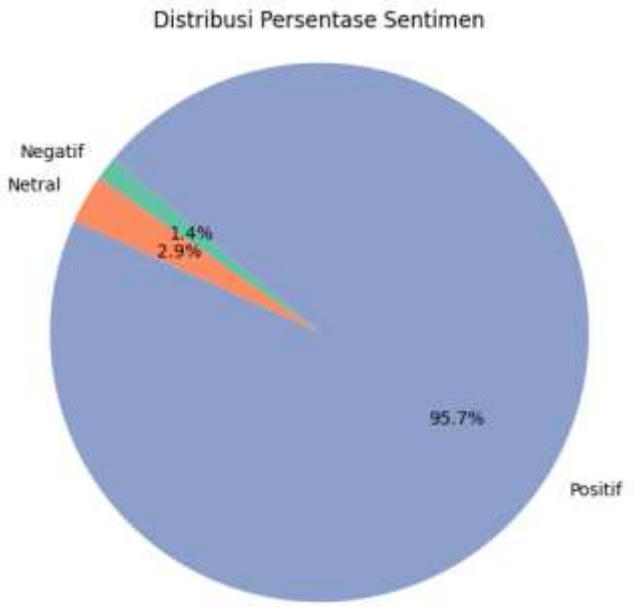
Gambar 4.15 Script Distribusi Presentase Sentimen

Berdasarkan hasil distribusi jumlah komentar setelah dilakukan proses labeling, diperoleh bahwa komentar dengan sentimen positif (Label 2) mendominasi dengan total 957 komentar, disusul oleh sentimen netral (Label 1) sebanyak 29 komentar, dan negatif (Label 0) sebanyak 14 komentar. Hal ini menunjukkan bahwa mayoritas tanggapan pengguna terhadap film *Agak Laen*

bersifat positif. Untuk memperoleh hasil ini, penulis terlebih dahulu menjalankan script analisis distribusi jumlah komentar seperti yang ditampilkan pada Gambar 4.14. Hasil dari script tersebut divisualisasikan dalam bentuk grafik batang pada Gambar 4.16. Selain itu, penulis juga menjalankan script untuk menghitung dan memvisualisasikan distribusi sentimen dalam bentuk persentase yang ditunjukkan pada Gambar 4.15, dengan hasil visualisasi dapat dilihat pada Gambar 4.17. Visualisasi ini memberikan gambaran yang lebih jelas mengenai proporsi masing-masing kategori sentimen secara keseluruhan.



Gambar 4.16 Jumlah Komentar per Sentimen



Gambar 4.17 Distribusi Presentase Sentimen

Berikut contoh hasil kolom sentimen sebelum dan sesudah melalui proses labeling data yang terdapat pada tabel 4.7.

Tabel 4.7 Contoh Hasil Proses Labeling Data

No.	Username	Sentimen (Sebelum)	Sentimen (Sesudah)
1.	inuriazizah_	Positif	2
2.	Ofixokefix	Positif	2
3.	Ewrikaaaaan	Positif	2
4.	Nbilhmrhm	Positif	2
5.	rendy.p.97	Negatif	0

4.7. Pembagian Data Latih dan Data Uji

Penulis membagi dataset menjadi dua bagian, yaitu data latih dan data uji. Tujuan pembagian ini adalah untuk melatih model pada sebagian data dan menguji

performanya pada data lain yang belum dikenali. Berikut ini merupakan tahapan dari proses pembagian dataset:

1. Penulis membagi data menjadi dua bagian, yaitu data latih (training) dan data uji (testing) menggunakan fungsi train_test_split dari library sklearn. Rasio pembagian yang digunakan adalah 90% untuk data latih dan 10% untuk data uji. Script pembagian data ditampilkan pada Gambar 4.18.
2. Hasil dari pembagian ini digunakan untuk proses pelatihan dan pengujian model, di mana data latih digunakan untuk membangun model dan data uji digunakan untuk mengevaluasi performa model.

Berikut merupakan script yang digunakan oleh penulis untuk membagi data menjadi data latih dan data uji, serta script untuk distribusi sentimen pada data latih dan data uji.

```
#SPLIT DATA (DATA UJI & DATA LATIH)✓

from sklearn.model_selection import train_test_split

X = df['stemmed'].apply(lambda x: ' '.join(x) if isinstance(x, list) else (x if isinstance(x, str) else ''))
y = df['Sentimen']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42, stratify=y)
#Untuk mengatur jumlah data, dimana yg direkomendasikan 80%:20% tetapi jika mengikuti itu, hasilnya sama.

# Cek jumlah data
print("Jumlah data latih:", len(X_train))
print("Jumlah data uji:", len(X_test))
```

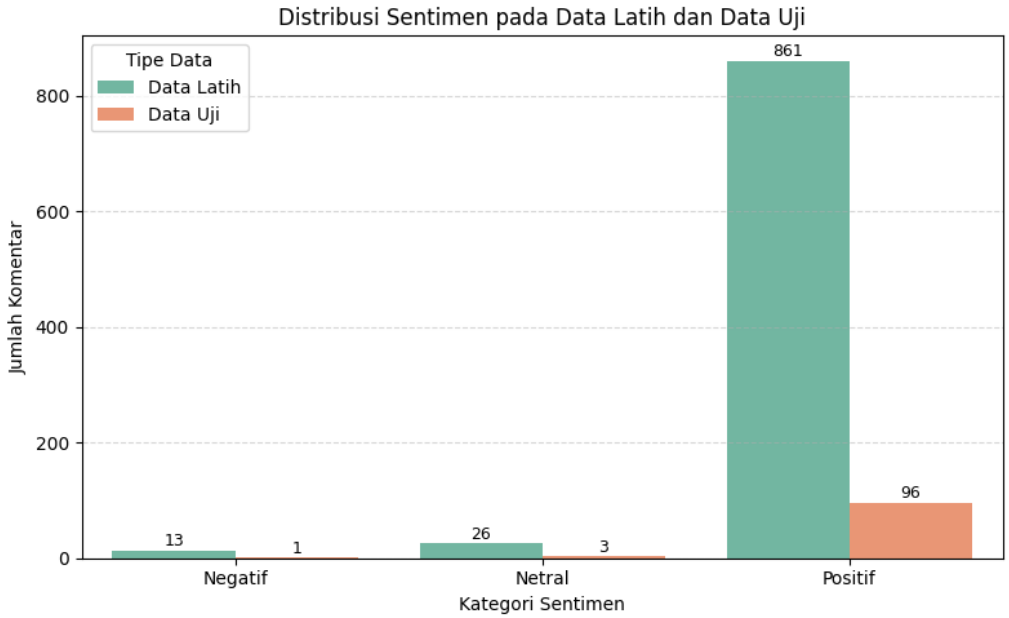
Gambar 4.18 Script Pembagian Dataset

```
# Hitung jumlah sentimen di data latih dan uji
train_counts = y_train.value_counts().sort_index()
test_counts = y_test.value_counts().sort_index()

# Label
label_sentimen = ['Negatif', 'Netral', 'Positif']

# Gabungkan ke DataFrame
df_split = pd.DataFrame({
    'Sentimen': label_sentimen,
    'Data Latih': train_counts.values,
    'Data Uji': test_counts.values
})

# Tampilkan tabel
print(df_split)

# Buat diagram batang
df_split_melted = df_split.melt(id_vars='Sentimen', var_name='Tipe Data', value_name='Jumlah')

plt.figure(figsize=(8, 5))
sns.barplot(x='Sentimen', y='Jumlah', hue='Tipe Data', data=df_split_melted, palette='Set2')

# Tambahkan label angka di atas batang
for index, row in df_split_melted.iterrows():
    plt.text(x=row.name % 3 - 0.2 + 0.4 * (row['Tipe Data'] == 'Data Uji'),
             y=row['Jumlah'] + 1,
             s=row['Jumlah'],
             ha='center', va='bottom', fontsize=9)

plt.title('Distribusi Sentimen pada Data Latih dan Data Uji')
plt.xlabel('Kategori Sentimen')
plt.ylabel('Jumlah Komentar')
plt.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout()
plt.show()
```

Gambar 4.19 Script Distribusi Sentimen Data Latih dan Data Uji

Hasil dari distribusi sentimen pada data latih dan data uji menunjukkan bahwa data terbagi secara proporsional berdasarkan masing-masing kategori sentimen. Pada data latih, terdapat 13 komentar negatif, 26 komentar netral, dan 861 komentar positif. Sementara itu, pada data uji terdapat 1 komentar negatif, 3 komentar netral, dan 96 komentar positif. Hal ini menunjukkan bahwa data dengan sentimen positif mendominasi baik pada data latih maupun data uji, yang mencerminkan adanya ketidakseimbangan distribusi kelas dalam dataset. Untuk memperoleh distribusi ini, penulis menjalankan script analisis distribusi sentimen pada data latih dan uji seperti yang ditampilkan pada Gambar 4.19. Hasil dari script

tersebut divisualisasikan dalam bentuk grafik batang pada Gambar 4.20, yang memberikan gambaran jelas mengenai proporsi masing-masing kategori sentimen setelah pembagian dataset.



Gambar 4.20 Distribusi Sentimen pada Data Latih dan Data Uji

4.8. Proses Ekstraksi Fitur

Penulis mengubah data teks menjadi representasi numerik yang dapat dipahami oleh algoritma machine learning. Proses ini menggunakan metode TF-IDF (Term Frequency–Inverse Document Frequency), yang bertujuan untuk menilai seberapa penting suatu kata dalam sebuah dokumen relatif terhadap seluruh teks ulasan. Dengan teknik ini, setiap komentar teks dikonversi menjadi vektor angka. Berikut ini merupakan tahapan dari proses ekstraksi fitur:

1. Penulis menggunakan metode TF-IDF (Term Frequency–Inverse Document Frequency) untuk mengubah data teks menjadi representasi numerik yang dapat diproses oleh algoritma machine learning.

2. Penulis mengatur parameter max_features=300 pada TfidfVectorizer untuk membatasi jumlah kata unik yang digunakan sebagai fitur agar lebih efisien dan relevan.
3. Data latih dikonversi menggunakan fungsi fit_transform(), sedangkan data uji menggunakan transform() agar proses pembobotan konsisten dengan data latih.
4. Hasil konversi berupa vektor numerik ini digunakan sebagai input dalam proses pelatihan dan pengujian model klasifikasi. Script ekstraksi fitur ditampilkan pada Gambar 4.21.

Berikut merupakan script yang digunakan oleh penulis dalam proses ekstraksi fitur untuk mengubah teks ulasan menjadi bentuk numerik dengan metode TF-IDF agar dapat digunakan dalam pemodelan.

```
#Ekstraksi Fitur: TF-IDF✓

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=300)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

#Melihat data uji setelah TF-IDF✓

print("Shape X_train_tfidf:", X_train_tfidf.shape)
print("Shape X_test_tfidf:", X_test_tfidf.shape)
```

Gambar 4.21 Script Ekstraksi Fitur

Setelah penulis menjalankan proses ekstraksi fitur menggunakan script yang ditampilkan pada Gambar 4.21, langkah selanjutnya adalah mengecek bentuk (shape) dari data hasil vektorisasi TF-IDF. Tujuannya adalah untuk memastikan bahwa proses transformasi data berjalan dengan baik pada data latih dan data uji. Hasil shape dari data uji menunjukkan bahwa terdapat 100 data uji dan 300 fitur

TF-IDF, sedangkan data latih memiliki 900 data dengan jumlah fitur yang sama. Hasil ini ditampilkan pada Gambar 4.22.

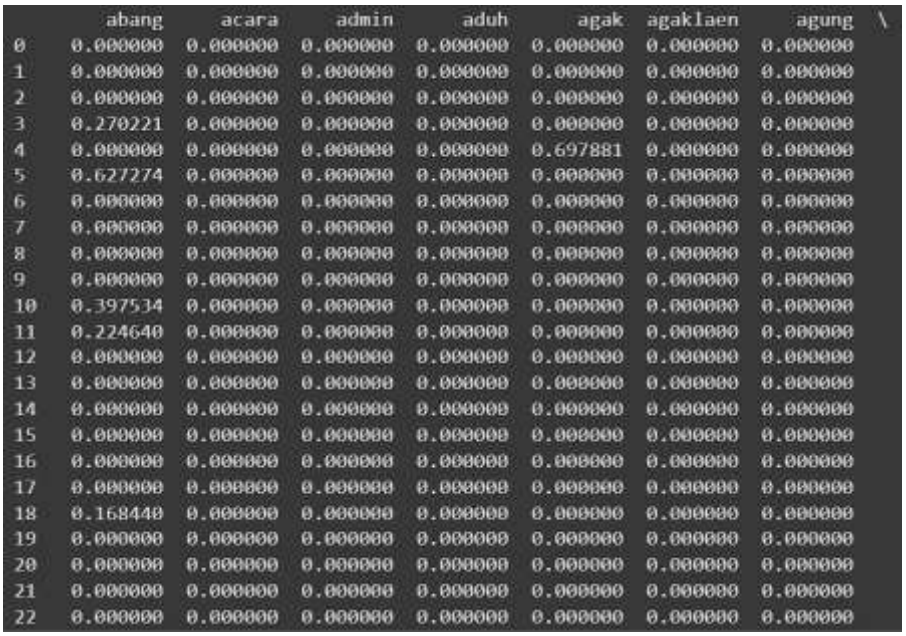
```
Shape X_train_tfidf: (900, 300)
Shape X_test_tfidf: (100, 300)
```

Gambar 4.22 Output Shape TF-IDF dari Data Latih dan Uji

Setelah hasil ekstraksi fitur TF-IDF dikonversi menjadi bentuk DataFrame, penulis menyesuaikan pengaturan tampilan output agar seluruh baris dan kolom dapat terlihat dengan jelas di lingkungan kerja. Hal ini dilakukan dengan tujuan untuk mempermudah proses verifikasi, eksplorasi, atau pengamatan langsung terhadap hasil vektorisasi dari data latih. Data TF-IDF yang awalnya berbentuk sparse matrix diubah menjadi array biasa agar dapat dianalisis lebih lanjut dan ditampilkan secara utuh dari indeks 0 hingga 999, seperti yang diperlihatkan script pada gambar 4.23 dan output pada gambar 4.24.

```
# Konversi hasil TF-IDF menjadi DataFrame agar terbaca
tfidf_df = pd.DataFrame(
    X_train_tfidf.toarray(),  # Ubah sparse matrix jadi array biasa
    columns=vectorizer.get_feature_names_out()  # Ambil nama-nama fitur (kata)
)

# Tampilkan seluruh data dari index 0 sampai 1000 (data latih = 1000)
pd.set_option('display.max_rows', 1000)      # (Opsional) agar baris tidak terpotong
pd.set_option('display.max_columns', None)  # Tampilkan semua kolom (kata)
pd.set_option('display.width', None)        # Hindari pemotongan tampilan lebar

# Tampilkan semua data vektorisasi dari 1-800
print(tfidf_df.iloc[:1000])
```

Gambar 4.23 Script TF-IDF Data Latih secara Keseluruhan



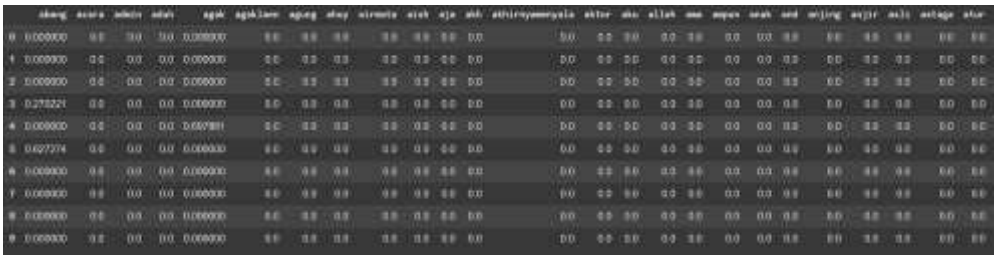
	abang	acara	admin	aduh	agak	agaklaen	agung	\
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
3	0.270221	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
4	0.000000	0.000000	0.000000	0.000000	0.697881	0.000000	0.000000	
5	0.627274	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
10	0.397534	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
11	0.224640	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
14	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
17	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
18	0.168440	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
20	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
21	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
22	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	

Gambar 4.24 Output Tampilan TF-IDF Data Latih secara Keseluruhan

Penulis juga menampilkan sebagian kecil hasil vektorisasi dalam bentuk DataFrame untuk memastikan bahwa proses TF-IDF telah berjalan dengan baik. Pada tahap ini, hanya ditampilkan 10 baris pertama dari hasil data latih yang telah dikonversi dari bentuk sparse matrix menjadi array. Hal ini bertujuan untuk melakukan pengecekan awal terhadap struktur data dan kata-kata (fitur) yang berhasil diambil oleh vectorizer. Tampilan ini disajikan secara ringkas agar lebih mudah diamati, seperti yang ditunjukkan pada Gambar 4.25 untuk script dan gambar 4.26 untuk output.

```
# Ambil daftar fitur (kata-kata unik) dari vectorizer
feature_names = vectorizer.get_feature_names_out()

# Konversi X_train_tfidf (sparse matrix) ke array
X_train_array = X_train_tfidf.toarray()

# Buat DataFrame TF-IDF
tfidf_df = pd.DataFrame(X_train_array, columns=feature_names)

# Tampilkan 5 baris pertama
tfidf_df.head(10)
```

Gambar 4.25 Script Tampilan TF-IDF pada 10 Baris Pertama Data Latih



Gambar 4.26 Output Tampilan TF-IDF pada 10 Baris Pertama Data Latih

Penulis menggunakan metode TF-IDF karena teknik ini mampu mengubah data teks menjadi representasi numerik yang memperhitungkan pentingnya suatu kata dalam suatu dokumen relatif terhadap seluruh korpus. TF-IDF juga mengurangi bobot dari kata-kata umum yang muncul di banyak dokumen. Dengan demikian, kata-kata yang lebih khas dan bermakna dalam konteks tertentu akan mendapatkan bobot lebih tinggi. Untuk menerapkan metode ini, penulis menggunakan "TfidfVectorizer" dari library Scikit-learn karena sudah menyediakan fungsi otomatis untuk melakukan tokenisasi, menghitung TF dan IDF, serta menghasilkan representasi vektor dalam format yang siap digunakan untuk pemodelan machine learning.

4.9. Implementasi Metode

Setelah seluruh tahapan pra-pemrosesan dan ekstraksi fitur selesai dilakukan, Penulis melakukan implementasi metode. tujuan dari implementasi metode adalah untuk menguji dan membandingkan performa dua algoritma klasifikasi, yaitu Logistic Regression dan Random Forest, dalam menganalisis sentimen pada ulasan film Agak Laen.

Pada tahap ini, kedua algoritma diterapkan terhadap data latih yang telah diubah ke bentuk numerik melalui metode TF-IDF, lalu diuji menggunakan data uji untuk melihat tingkat akurasi dan efektivitas masing-masing model dalam mengklasifikasikan komentar ke dalam kategori sentimen negatif, netral, atau positif. Berikut tahapan dari implementasi metode:

1. Penulis mengimpor library seperti “sklearn.linear_model” untuk Logistic Regression dan “sklearn.ensemble” untuk Random Forest yang dibutuhkan dalam proses pemodelan.
2. Model Logistic Regression dilatih menggunakan data latih hasil ekstraksi fitur TF-IDF. Setelah proses pelatihan, model digunakan untuk memprediksi data uji dan hasilnya dianalisis. Script yang digunakan penulis untuk proses ini ditampilkan pada Gambar 4.27.
3. Model Random Forest juga dilatih menggunakan data latih yang sama, kemudian digunakan untuk memprediksi sentimen dari data uji. Script yang digunakan penulis dalam tahap ini juga ditunjukkan pada Gambar 4.28.

Berikut merupakan script yang digunakan oleh penulis dalam proses implementasi metode, yaitu untuk melatih dan menguji model klasifikasi Logistic

Regression dan Random Forest berdasarkan data yang telah diekstraksi menggunakan TF-IDF, guna memprediksi sentimen dari ulasan secara otomatis.

```
#Logistic Regression✓
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

lr_model = LogisticRegression()
lr_model.fit(X_train_tfidf, y_train)
y_pred_lr = lr_model.predict(X_test_tfidf)
```

Gambar 4.27 Script Logistic Regression

```
#RANDOM FOREST✓

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_tfidf, y_train)
y_pred_rf = rf_model.predict(X_test_tfidf)
```

Gambar 4.28 Script Random Forest

4.10. Hasil Evaluasi Model

Setelah proses implementasi metode dilakukan dan model berhasil dilatih serta diuji, langkah selanjutnya adalah menampilkan hasil evaluasi dari masing-masing algoritma. Tahap ini bertujuan untuk melihat dan membandingkan performa model Logistic Regression dan Random Forest dalam mengklasifikasikan sentiment.

4.10.1. Evaluasi Logistic Regression

Setelah model Logistic Regression dilatih menggunakan data latih, penulis melanjutkan proses dengan menguji performa model terhadap data uji. Hal ini dilakukan dengan cara memprediksi label sentimen menggunakan model yang telah

dilatih, lalu membandingkannya dengan label aktual. Penulis menggunakan script evaluasi model yang dapat dilihat pada gambar 4.29.

```
print("Logistic Regression Results:")
print(classification_report(y_test, y_pred_lr))
print("Akurasi:", accuracy_score(y_test, y_pred_lr))
```

Gambar 4.29 Script Evaluasi Model Logistic Regression

Penulis menemukan bahwa hasil akurasi dari evaluasi Logistic Regression sebesar 96%, yang menunjukkan bahwa model memiliki kemampuan yang cukup baik dalam memprediksi sentimen secara keseluruhan. Namun, jika dilihat dari nilai precision, recall, dan f1-score untuk setiap kelas, terlihat bahwa model hanya berhasil mengklasifikasikan sentimen positif (label 2) dengan sangat baik, sementara untuk sentimen negatif (label 0) dan netral (label 1), model tidak mampu memberikan prediksi yang tepat. Hal ini kemungkinan besar disebabkan oleh distribusi data yang tidak seimbang, di mana kelas positif mendominasi. Rincian hasil klasifikasi Logistic Regression ditampilkan pada Gambar 4.30.

```
Logistic Regression Results:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.00      0.00      0.00         3
           2       0.96      1.00      0.98        96

    accuracy                           0.96       100
   macro avg       0.32      0.33      0.33       100
weighted avg       0.92      0.96      0.94       100

Akurasi: 0.96
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

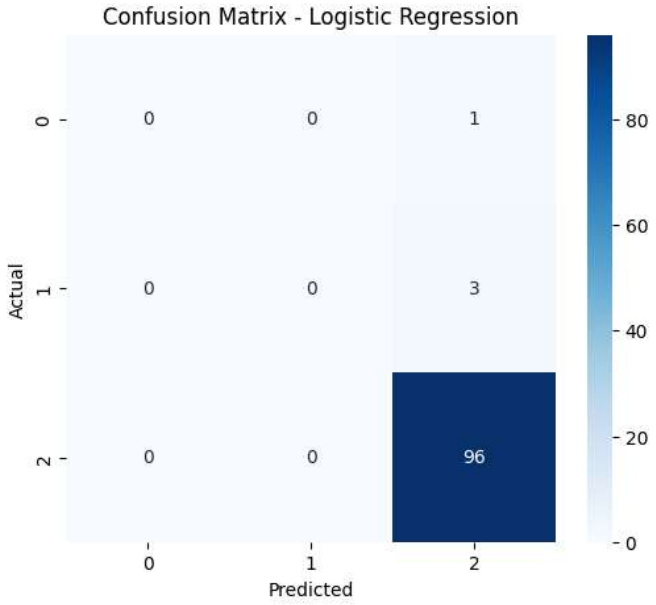
Gambar 4.30 Hasil Evaluasi Model Logistic Regression

Penulis juga menyajikan confusion matrix untuk model Logistic Regression yang ditampilkan pada Gambar 4.32. Confusion matrix berfungsi untuk

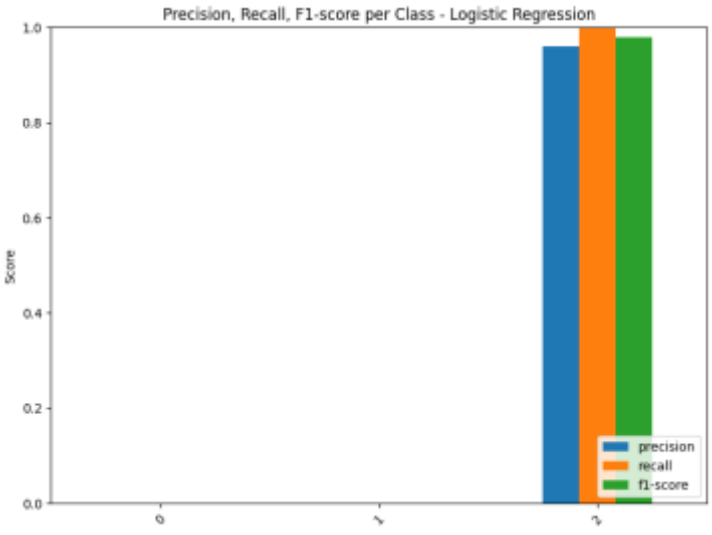
menggambarkan kinerja model klasifikasi dengan menunjukkan jumlah prediksi yang benar dan salah pada masing-masing kelas, sehingga memudahkan dalam mengevaluasi kesalahan klasifikasi secara spesifik. Selain itu, penulis juga membuat diagram yang terlihat pada Gambar 4.33 untuk memvisualisasikan nilai precision, recall, dan F1-score dari masing-masing kelas sentimen. Visualisasi ini berguna untuk melihat sejauh mana model mampu mengenali setiap kelas secara detail. Adapun script yang digunakan untuk menghasilkan evaluasi tersebut ditampilkan pada Gambar 4.31.

```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred_lr)
labels = lr_model.classes_

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Logistic Regression')
plt.show()

# Classification Report (bar chart)
report = classification_report(y_test, y_pred_lr, output_dict=True)
report_df = pd.DataFrame(report).transpose()

# Ambil hanya kelas (tanpa avg/accuracy)
report_df_class = report_df.iloc[:-3, :]

report_df_class[['precision', 'recall', 'f1-score']].plot(kind='bar', figsize=(8,6))
plt.title('Precision, Recall, F1-score per Class - Logistic Regression')
plt.ylim(0,1)
plt.ylabel('Score')
plt.xticks(rotation=45)
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

Gambar 4.31 Script Confution Martrix dan Grafik Evaluasi



Gambar 4.32 Confusion Matrix Logistic Regression



Gambar 4.33 Grafik Evaluasi Logistic Regression

4.10.2. Evaluasi Random Forest

Penulis melanjutkan tahap evaluasi dengan menguji kemampuan model dalam mengklasifikasikan data uji. Proses ini dilakukan dengan memprediksi kategori sentimen dari data uji menggunakan model yang telah dibangun, kemudian

hasil prediksi dibandingkan dengan label sebenarnya. Evaluasi performa model dilakukan menggunakan metrik yang ditampilkan melalui script pada Gambar 4.34.

```
print("Random Forest Results:")
print(classification_report(y_test, y_pred_rf))
print("Akurasi:", accuracy_score(y_test, y_pred_rf))
```

Gambar 4.34 Script Evaluasi Model Random Forest

Berdasarkan hasil evaluasi model Random Forest, penulis memperoleh akurasi sebesar 98%, yang menunjukkan bahwa model ini memiliki performa yang sangat baik dalam mengklasifikasikan sentimen komentar. Model mampu memprediksi kelas positif (label 2) dengan sangat baik, dengan nilai precision 0.99, recall 1.00, dan f1-score 0.99. Sementara itu, untuk kelas netral (label 1), nilai precision dan recall masing-masing adalah 0.67, yang berarti model cukup baik namun belum sempurna dalam mengenali kelas ini. Untuk kelas negatif (label 0), model belum berhasil memberikan prediksi yang akurat, terlihat dari nilai precision dan recall sebesar 0.00. Hal ini menunjukkan bahwa meskipun performa keseluruhan tinggi, model masih mengalami kesulitan dalam menangani kelas dengan jumlah data yang sangat sedikit. Rincian hasil evaluasi model Random Forest dapat dilihat pada Gambar 4.35.

```
Random Forest Results:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.67      0.67      0.67         3
           2       0.99      1.00      0.99        96

    accuracy                           0.98       100
   macro avg       0.55      0.56      0.55       100
weighted avg       0.97      0.98      0.98       100

Akurasi: 0.98
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565:
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Gambar 4.35 Hasil Evaluasi Model Random Forest

Penulis juga menyajikan visualisasi pohon keputusan sebagai bagian dari interpretasi model Random Forest. Visualisasi ini berfungsi untuk menunjukkan bagaimana salah satu pohon dalam ensemble Random Forest mengambil keputusan berdasarkan nilai fitur tertentu dari data yang telah diekstraksi menggunakan TF-IDF. Pada Gambar 4.38, ditampilkan salah satu dari sepuluh pohon yang digunakan dalam model, di mana setiap simpul merepresentasikan keputusan yang dibuat berdasarkan nilai TF-IDF dari kata tertentu, seperti “gak”, “mau”, “silver”, “medan”, dan lainnya. Setiap percabangan dalam pohon menggambarkan kondisi logika (misalnya apakah nilai suatu fitur lebih kecil dari ambang tertentu), dan diikuti oleh distribusi label sentimen pada node tersebut dalam bentuk nilai value = [a, b, c], yang berarti jumlah sampel dalam masing-masing kelas sentimen (negatif, netral, positif). Nilai gini pada setiap simpul menunjukkan tingkat ketidakmurnian data di titik tersebut semakin kecil nilainya, semakin murni node tersebut (semakin seragam kelasnya).

Visualisasi ini sangat bermanfaat untuk memahami fitur apa saja yang dianggap penting oleh model, dan bagaimana pengaruh fitur tersebut terhadap keputusan klasifikasi. Selain itu, ini juga dapat membantu dalam mengidentifikasi kemungkinan overfitting atau bias dalam pohon. Pada bagian bawah gambar terdapat tanda (...) yang menunjukkan bahwa sebagian cabang pohon tidak ditampilkan secara lengkap untuk menjaga keterbacaan visual, karena struktur pohon dapat menjadi sangat besar dan kompleks.

Script yang digunakan untuk mengekstrak dan memvisualisasikan pohon keputusan ditampilkan pada Gambar 4.36. Dalam proses tersebut, diperoleh sebanyak 300 fitur unik hasil ekstraksi TF-IDF yang digunakan untuk

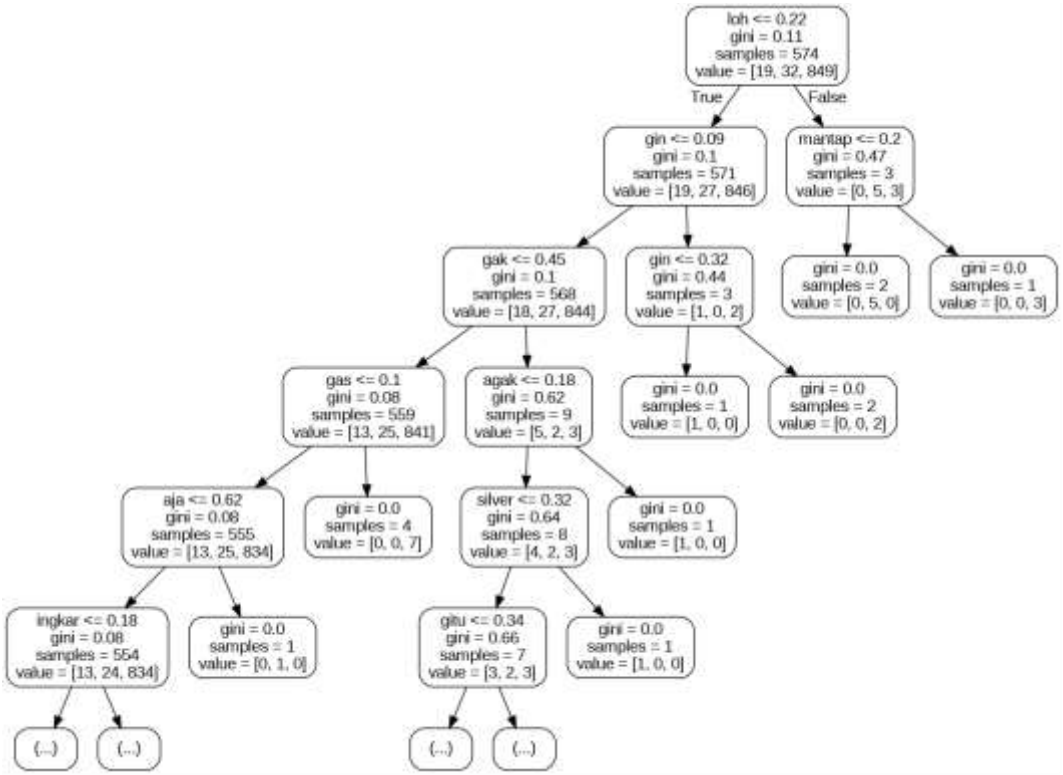
merepresentasikan teks pada data latih. Jumlah fitur tersebut telah sesuai antara data pelatihan dan daftar nama fiturnya, sebagaimana ditampilkan pada Gambar 4.37.

```
rf = RandomForestClassifier(n_estimators=10, random_state=42)
rf.fit(X_train_tfidf, y_train)
feature_list = vectorizer.get_feature_names_out().tolist()
train_features = pd.DataFrame(X_train_tfidf.toarray(), columns=feature_list)
tree = rf.estimators_[1]

# 11. Ekspor ke .dot
export_graphviz(tree,
                out_file='tree.dot',
                feature_names=feature_list,
                rounded=True,
                precision=2,
                max_depth=5)  # Agar tidak terlalu besar

# 12. Convert ke .png dan tampilkan

print("Jumlah fitur dalam train_features:", train_features.shape[1])
print("Jumlah nama fitur dalam feature_list:", len(feature_list))

(graph,) = pydot.graph_from_dot_file('tree.dot')
graph.write_png('tree.png')
display(Image(filename='tree.png'))
```

Gambar 4.36 Script Visualisasi Struktur Pohon Keputusan Random Forest

```
Jumlah fitur dalam train_features: 300
Jumlah nama fitur dalam feature_list: 300
```

Gambar 4.37 Jumlah Fitur pada Data TF-IDF Hasil Pelatihan



Gambar 4.38 Visualisasi Struktur Pohon Keputusan Random Forest

Keterangan:

1. Setiap node menunjukkan fitur (kata) yang digunakan untuk pemisahan, seperti gak, kenapa, capek, dll.
2. Gini menunjukkan nilai impuritas (semakin kecil, semakin "bersih" node tersebut).
3. Samples menunjukkan jumlah sampel (komentar) yang mencapai node itu.
4. Value menunjukkan distribusi jumlah komentar dalam masing-masing kelas sentimen (misalnya [7, 26, 827] artinya ada 7 negatif, 26 netral, dan 827 positif).
5. Cabang kiri berarti kondisi True (misalnya gak <= 0.37), sedangkan cabang kanan berarti kondisi False.

6. Dari struktur tersebut dapat dilihat bahwa pohon sangat condong pada kelas positif (label 2), karena dataset memang didominasi komentar positif.

4.11. Perbandingan Metode

Penulis melakukan perbandingan metode antara dua algoritma klasifikasi yang digunakan dalam penelitian ini, yaitu Logistic Regression dan Random Forest. Perbandingan dilakukan untuk melihat performa masing-masing model dalam mengklasifikasikan sentimen berdasarkan hasil evaluasi menggunakan metrik akurasi dan F1-score. Proses evaluasi dan perbandingan model ini dilakukan melalui script yang ditampilkan pada Gambar 4.39.

```
#Perbandingan Hasil✓

from sklearn.metrics import f1_score

print("Akurasi Logistic Regression:", accuracy_score(y_test, y_pred_lr))
print("F1-Score Logistic Regression:", f1_score(y_test, y_pred_lr, average='weighted'))

print("Akurasi Random Forest:", accuracy_score(y_test, y_pred_rf))
print("F1-Score Random Forest:", f1_score(y_test, y_pred_rf, average='weighted'))
```

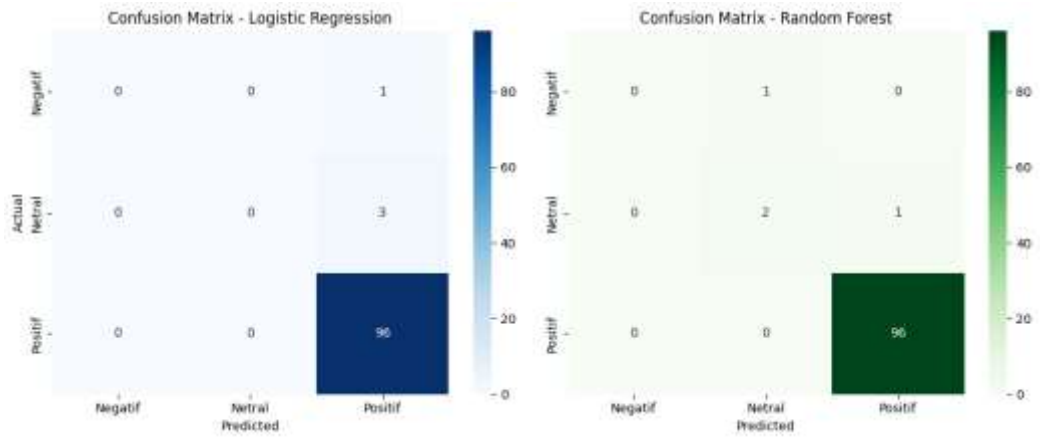
Gambar 4.39 Script Evaluasi dan Perbandingan Model

Berdasarkan hasil evaluasi, model Logistic Regression menghasilkan akurasi sebesar 0.96 dan F1-score sebesar 0.9404, sedangkan model Random Forest menunjukkan hasil yang lebih baik dengan akurasi sebesar 0.98 dan F1-score sebesar 0.9750. Hasil ini menunjukkan bahwa Random Forest memiliki performa klasifikasi yang lebih unggul dalam konteks data yang digunakan. Rincian perbandingan hasil evaluasi dari kedua metode tersebut dapat dilihat pada Gambar 4.40.

```
Akurasi Logistic Regression: 0.96
F1-Score Logistic Regression: 0.9404081632653061
Akurasi Random Forest: 0.98
F1-Score Random Forest: 0.9750259067357513
```

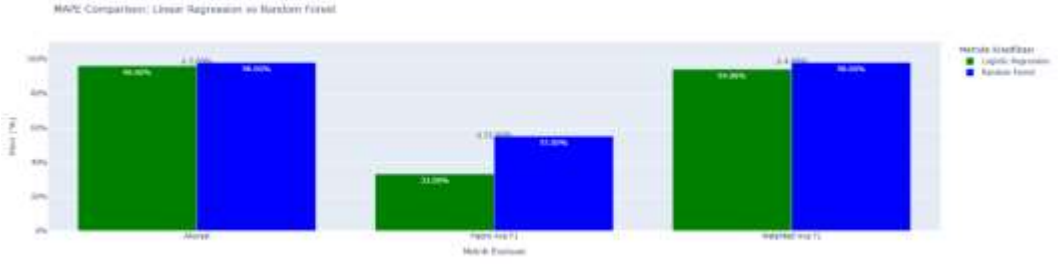
Gambar 4.40 Visualisasi Perbandingan Hasil Evaluasi Model

Penulis menyajikan perbandingan confusion matrix antara dua metode klasifikasi yang digunakan, berfungsi untuk menggambarkan kinerja model dalam mengklasifikasikan data uji berdasarkan label aktual dan prediksi. Pada confusion matrix Logistic Regression (kiri), terlihat bahwa seluruh prediksi diarahkan ke kelas positif, yaitu 96 prediksi benar dari total 100 data, sementara kelas negatif dan netral tidak berhasil diprediksi sama sekali. Sebaliknya, pada confusion matrix Random Forest (kanan), model mampu memprediksi semua kelas, termasuk netral dan negatif, meskipun masih terdapat beberapa kesalahan klasifikasi, seperti satu komentar netral yang diklasifikasikan sebagai positif. Hal ini menunjukkan bahwa Random Forest memiliki kemampuan generalisasi yang lebih baik terhadap kelas minoritas. Visualisasi ini bertujuan untuk memberikan gambaran yang lebih jelas mengenai akurasi dan kesalahan klasifikasi masing-masing model, dan dapat dilihat pada Gambar 4.41.

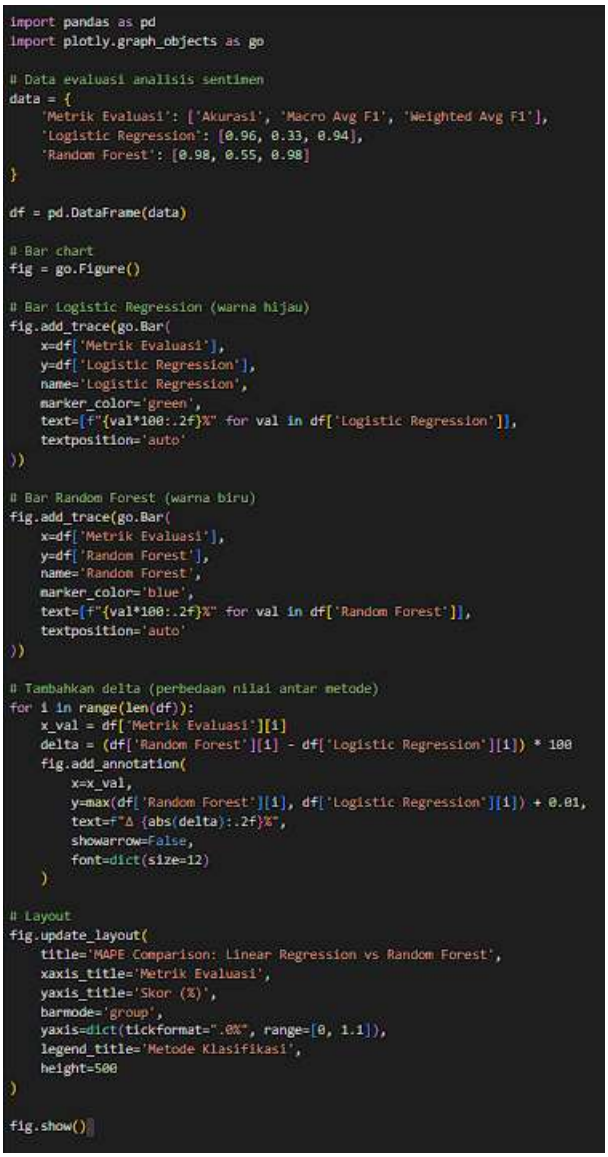


Gambar 4.41 Hasil Confusion Matrix untuk Perbandingan Model

Selain mengevaluasi model berdasarkan akurasi, f1-score, dan confusion matrix, penelitian ini juga membandingkan performa metode Logistic Regression dan Random Forest menggunakan metrik Mean Absolute Percentage Error (MAPE). Meskipun MAPE umumnya digunakan untuk mengukur kesalahan dalam prediksi numerik, dalam konteks ini MAPE digunakan untuk membandingkan selisih persentase performa antar metode berdasarkan metrik evaluasi utama. Visualisasi hasil perbandingan ini ditampilkan pada Gambar 4.42, sedangkan script yang digunakan untuk menghasilkan visualisasi tersebut disajikan pada Gambar 4.43. Berdasarkan hasil yang diperoleh, Random Forest menunjukkan performa yang lebih unggul dalam semua metrik yang dibandingkan. Pada metrik akurasi, Random Forest mencapai skor 98%, lebih tinggi 2% dibandingkan Logistic Regression yang memperoleh 96%. Perbedaan paling signifikan terjadi pada metrik macro average f1-score, dengan Random Forest mencapai 55% dan Logistic Regression hanya 33%, menghasilkan selisih sebesar 22%. Pada weighted average f1-score, Random Forest juga unggul dengan skor 98%, lebih tinggi 4% dari Logistic Regression yang memperoleh 94%. Hasil ini menunjukkan bahwa Random Forest tidak hanya lebih akurat, tetapi juga lebih konsisten dan andal dalam menangani data dengan distribusi kelas yang tidak seimbang.



Gambar 4.42 Diagram Perbandingan MAPE



```
import pandas as pd
import plotly.graph_objects as go

# Data evaluasi analisis sentimen
data = {
    'Metrik Evaluasi': ['Akurasi', 'Macro Avg F1', 'Weighted Avg F1'],
    'Logistic Regression': [0.96, 0.33, 0.94],
    'Random Forest': [0.98, 0.55, 0.98]
}

df = pd.DataFrame(data)

# Bar chart
fig = go.Figure()

# Bar Logistic Regression (warna hijau)
fig.add_trace(go.Bar(
    x=df['Metrik Evaluasi'],
    y=df['Logistic Regression'],
    name='Logistic Regression',
    marker_color='green',
    text=[f"{val*100:.2f}%" for val in df['Logistic Regression']],
    textposition='auto'
))

# Bar Random Forest (warna biru)
fig.add_trace(go.Bar(
    x=df['Metrik Evaluasi'],
    y=df['Random Forest'],
    name='Random Forest',
    marker_color='blue',
    text=[f"{val*100:.2f}%" for val in df['Random Forest']],
    textposition='auto'
))

# Tambahkan delta (perbedaan nilai antar metode)
for i in range(len(df)):
    x_val = df['Metrik Evaluasi'][1]
    delta = (df['Random Forest'][1] - df['Logistic Regression'][1]) * 100
    fig.add_annotation(
        x=x_val,
        y=max(df['Random Forest'][1], df['Logistic Regression'][1]) + 0.01,
        text=f"Δ {abs(delta):.2f}%",
        showarrow=False,
        font=dict(size=12)
    )

# Layout
fig.update_layout(
    title='MAPE Comparison: Linear Regression vs Random Forest',
    xaxis_title='Metrik Evaluasi',
    yaxis_title='Skor (%)',
    barmode='group',
    yaxis=dict(tickformat=".0%", range=[0, 1.1]),
    legend_title='Metode Klasifikasi',
    height=500
)

fig.show()
```

Gambar 4.43 Script Visualisasi Perbandingan MAPE

Penulis juga melakukan analisis secara kualitatif terhadap kedua metode yang digunakan, yaitu Random Forest dan Logistic Regression. Berdasarkan pengamatan terhadap performa model, Random Forest menunjukkan kemampuan yang lebih unggul dalam menangani distribusi data yang tidak seimbang, dengan memberikan prediksi yang lebih akurat dan stabil di berbagai kelas sentimen. Model ini juga mampu mengenali pola yang lebih kompleks dalam data, yang membuatnya lebih fleksibel dalam klasifikasi. Di sisi lain, Logistic Regression

memiliki kelebihan dalam hal kecepatan dan interpretabilitas hasil, serta cukup efektif untuk data dengan pola linier yang sederhana. Namun, model ini kurang mampu menangkap hubungan non-linier dalam data dan performanya cenderung menurun ketika dihadapkan dengan distribusi kelas yang tidak merata. Secara keseluruhan, dari hasil evaluasi dan karakteristik masing-masing model, Random Forest dinilai lebih andal dalam menyelesaikan permasalahan analisis sentimen pada ulasan film Agak Laen.

Selain melakukan evaluasi berdasarkan metrik performa seperti akurasi dan f1-score, penulis juga menyusun sebuah tabel yang berisi keunggulan dan kelemahan dari masing-masing metode yang digunakan, dapat dilihat pada tabel 4.8.

Tabel 4. 8 Perbandingan Keunggulan dan Kelemahan Metode

Metode	Keunggulan	Kelemahan
Logistic Regression	1. Algoritma yang mudah diimplementasikan dan cepat dalam proses pelatihan. 2. Sangat cocok digunakan ketika hubungan antara fitur dan target bersifat linear. 3. Logistic Regression cenderung stabil pada data yang tidak terlalu kompleks.	1. Tidak optimal jika data memiliki hubungan non-linear yang kuat. 2. Outlier atau data ekstrem bisa sangat memengaruhi hasil model. 3. Kinerja menurun jika fitur tidak berhubungan secara linier dengan target.
Random Forest	1. Mampu menangani data kompleks dan berukuran besar dengan performa tinggi. 2. Penggabungan banyak pohon membuat model lebih stabil dan minim overfitting. 3. Dapat mengidentifikasi fitur yang paling berpengaruh dalam pengambilan Keputusan.	1. Sulit untuk memahami dan menjelaskan hasil prediksi karena terdiri dari banyak pohon. 2. Dibandingkan dengan Logistic Regression, proses pelatihan lebih lambat, terutama jika jumlah pohon besar. 3. Karena banyaknya pohon yang dibangun, model ini bisa lebih memakan memori.

4.12. Pembahasan

Berdasarkan hasil evaluasi yang telah dilakukan, terlihat bahwa metode Random Forest memberikan performa yang lebih unggul dibandingkan Logistic Regression dalam mengklasifikasikan sentimen ulasan film *Agak Laen*. Random Forest mencapai akurasi hingga 98% dengan nilai F1-score yang tinggi dan seimbang di hampir semua kelas sentimen, sedangkan Logistic Regression hanya memperoleh akurasi 96% dan mengalami kesulitan dalam memprediksi kelas negatif dan netral akibat ketidakseimbangan data. Hal ini sesuai dengan teori yang menyatakan bahwa Random Forest, sebagai metode ensemble, lebih tangguh dalam menangani data yang kompleks dan tidak seimbang karena menggabungkan banyak pohon keputusan.

Di sisi lain, Logistic Regression cenderung lebih sederhana dan sensitif terhadap distribusi kelas yang tidak merata. Dalam proses eksperimen, salah satu tantangan utama adalah dominasi sentimen positif yang menyebabkan distribusi data menjadi timpang, sehingga berdampak pada kinerja model tertentu. Meskipun demikian, hasil yang diperoleh sejalan dengan harapan dan teori sebelumnya, di mana Random Forest memang dikenal lebih kuat dalam menangani variabilitas data. Implikasi dari penelitian ini menunjukkan bahwa pemilihan metode klasifikasi yang tepat sangat penting dalam analisis sentimen, terutama ketika diterapkan dalam konteks publikasi atau promosi film di media sosial, karena dapat memberikan pemahaman yang lebih akurat terhadap opini masyarakat.

Dalam proses pengerjaan penelitian ini, penulis mengalami beberapa kendala utama yang memengaruhi alur dan waktu pelaksanaan analisis sentimen.

Berikut merupakan tiga hambatan atau tantangan signifikan yang dihadapi selama penelitian berlangsung:

1. Penulis harus memilah secara manual dari 2.400 komentar menjadi 1.000 komentar yang relevan dan dapat digunakan sebagai data penelitian.
2. Proses pelabelan sentimen dilakukan secara manual terhadap 1.000 komentar, yang memerlukan ketelitian agar klasifikasinya akurat.
3. Ditemukan banyak kata tidak baku atau tidak sesuai, sehingga perlu dilakukan normalisasi untuk menyamakan bentuk kata sebelum diproses oleh model.

Meskipun dalam proses penelitian ini penulis menghadapi sejumlah kendala, seperti pemilahan data komentar, pemberian label sentimen secara manual, serta penanganan kata-kata tidak baku pada tahap normalisasi, seluruh tahapan tetap dapat diselesaikan dengan baik. Dengan melakukan penyesuaian dan upaya secara bertahap, penulis berhasil menyelesaikan proses analisis sentimen menggunakan dua metode klasifikasi, yaitu Logistic Regression dan Random Forest. Hasil evaluasi dan perbandingan dari kedua model menunjukkan performa yang cukup memuaskan dan sesuai dengan tujuan penelitian.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan maka dapat disimpulkan, bahwa:

1. Penelitian ini berhasil mengimplementasikan dua algoritma klasifikasi, yaitu Logistic Regression dan Random Forest, dalam menganalisis sentimen publik dari komentar Instagram terhadap film Agak Laen. Data yang digunakan berjumlah 1.000 komentar yang telah melalui proses data cleaning, preprocessing, labeling, dan ekstraksi fitur menggunakan metode TF-IDF.
2. Hasil evaluasi menunjukkan bahwa kedua metode memiliki performa yang baik, namun terdapat perbedaan tingkat akurasi dan kemampuan dalam menangani distribusi data yang tidak seimbang. Random Forest menunjukkan akurasi sebesar 98% dan F1-Score sebesar 0.975, sementara Logistic Regression memperoleh akurasi 96% dan F1-Score 0.94. Hal ini menunjukkan bahwa Random Forest memiliki kinerja yang lebih unggul dalam mengklasifikasikan sentimen komentar.
3. Visualisasi seperti confusion matrix, classification report, dan pohon keputusan memberikan interpretasi lebih lanjut terhadap kemampuan masing-masing model dalam mengenali pola data. Random Forest dinilai lebih tangguh karena kemampuannya dalam menggabungkan banyak pohon keputusan sehingga lebih stabil terhadap noise dan variasi data.

5.2. Saran

Adapun saran dari penulis untuk penelitian selanjutnya, yaitu sebagai berikut:

1. Menambahkan jumlah data komentar pada Instagram Agak Laen.
2. Menambahkan data komentar yang bervariatif, seperti deteksi sentiment melalui emoji.
3. Menggunakan teknik data balancing atau resampling untuk menangani distribusi data yang tidak seimbang agar hasil prediksi model terhadap semua kelas sentimen lebih merata.
4. Menerapkan metode lain seperti SVM (Support Vector Machine) atau Deep Learning untuk mengetahui perbandingan performa yang lebih luas.

DAFTAR PUSTAKA

Agustia, D. N., & Suryono, R. R. (2025). *COMPARISON OF NAÏVE BAYES, RANDOM FOREST, AND LOGISTIC REGRESSION ALGORITHMS FOR SENTIMENT ANALYSIS ONLINE GAMBLING KOMPARASI ALGORITMA NAÏVE BAYES, RANDOM FOREST, DAN LOGISTIC REGRESION UNTUK ANALISIS SENTIMEN JUDI ONLINE*. *10*(1), 2025. https://jurnal.polbeng.ac.id/index.php/ISI/article/view/356

Agustina, M. P., & Hendry. (2021). Sentimen Masyarakat Terkait Perpindahan Ibukota Via Model Random Forest dan Logistic Regression. *AITI: Jurnal Teknologi Informasi*, *18*(2), 111–124. https://ejournal.uksw.edu/aiti/article/view/5459

Alfarizi, M. R. sirfatullah, Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNINGDAN DEEP LEARNING. *KARIMAH TAUHID (Karya Ilmiah Mahasiswa Bertauhid)*, *2*(1). https://ojs.unida.ac.id/karimahtauhid/article/view/7518

Amaliah, S., Nusrang, M., & Aswi. (2022). PENERAPAN METODE RANDOM FOREST UNTUK KLASIFIKASI VARIAN MINUMAN KOPI DI KEDAI KOPI KONIJIWA BANTAENG. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, *4*(2). https://jurnalvariansi.unm.ac.id/index.php/variansi/article/view/31

Ardika, I. N. A. K., & Wibawa, I. G. A. W. (2022). *Analisis Sentimen Ulasan Pengguna Aplikasi Pelayanan Masyarakat Dengan Menggunakan Algoritma Random Forest*. https://ojs.unud.ac.id/index.php/jnatia/article/download/92585/47039

Averina, A., Hadi, H., & Siswantoro, J. (2022). Analisis Sentimen Multi-Kelas Untuk Film Berbasis Teks Ulasan Menggunakan Model Regresi Logistik. *Teknika*, *11*(2), 123–128. https://doi.org/10.34148/teknika.v11i2.461

Basit. (2024, April 15). *Logistic Regression*. BPTSI Unisa Yogyakarta. https://bptsi.unisayogya.ac.id/logistic-regression/

Britanthia, L., Tanujaya, C., Susanto, B., & Saragih, A. (2020). Perbandingan Metode Regresi Logistik dan Random Forest untuk Klasifikasi Fitur Mode

Audio Spotify. *Indonesian Journal of Data and Science (IJODAS), 1*(3), 68–78. https://www.jurnal.yoctobrain.org/index.php/ijodas/article/view/16

Digital Skola. (2023, October 14). *Panduan Menggunakan Google Colab dengan Mudah*. Digital Skola. https://digitalskola.com/blog/home/google-colab

Dwi, A. (2024, March 29). *10 Film Indonesia Terlaris Sepanjang Masa, Terbaru Film Agak Laen*. TEMPO. https://www.tempo.co/teroka/10-film-indonesia-terlaris-sepanjang-masa-terbaru-film-agak-laen-72664

Export Instagram Followers with IGExport. (n.d.). LinkedRadar. Retrieved May 9, 2025, from https://linkedradar.com/blog/export-instagram-followers

Fadhillah, O. S. D., Jaman, J. H., & Carudin. (2025). PERBANDINGAN NAIVE BAYES, SUPPORT VECTOR MACHINE, LOGISTIC REGRESSION DAN RANDOM FOREST DALAM MENGANALISIS SENTIMEN MENGENAI TIKTOKSHOP. *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, *13*(1). https://journal.eng.unila.ac.id/index.php/jitet/article/view/5746

Junianto, H., Saputro, R. E., Kusuma, B. A., Intan, D., & Saputra, S. (2024). COMPARISON OF LOGISTIC REGRESSION AND RANDOM FOREST IN SENTIMENT ANALYSIS OF DISDUKCAPIL APPLICATION REVIEWS. *Jurnal Teknik Informatika (JUTIF)*, *5*(6). https://doi.org/10.52436/1.jutif.2024.5.6.1802

Kurniawati, Y. (2023, November 24). *Analisis Sentimen dan Jenisnya*. Binus University. https://sis.binus.ac.id/2023/11/24/analisis-sentimen-dan-jenisnya/

M. Azhar N.H. (2024, November 21). *Pengertian Python: Bahasa Pemrograman Serbaguna dan Populer*. Telkom University. https://bse.telkomuniversity.ac.id/pengertian-python-bahasa-pemrograman-serbaguna-dan-populer/

Meilana, E. (2025, January 15). *Tren Terbaru Pembelajaran Mesin (Machine Learning) dan Dampaknya pada Dunia Industri*. UNESA (Universitas Negeri Surabaya). https://terapan-ti.vokasi.unesa.ac.id/post/tren-terbaru-pembelajaran-mesin-machine-learning-dan-dampaknya-pada-dunia-industri

Milagista, A. (2024, February 5). *Sinopsis Film Agak Laen, Kisah Horor-Komedi Empat Penjaga Rumah Hantu*. Detikjateng.

https://www.detik.com/jateng/budaya/d-7178370/sinopsis-film-agak-laen-kisah-horor-komedi-empat-penjaga-rumah-hantu/

Oliver, A. (2022, January 25). *Mengenal Google Colab: Mulai dari Definisi, Cara Menggunakan, hingga Manfaatnya*. Glints. https://glints.com/id/lowongan/google-colab-adalah/

Permana, N. A., & Bunyamin, H. (2024). Perbandingan Logistic Regression dengan Random Forest dalam Memprediksi Sentimen Pada IMDb Moview Review. *Jurnal Strategi*, *6*(2). http://www.strategi.it.maranatha.edu/index.php/strategi/article/view/538

Putri, S. (2024, July 2). *Tujuan Hingga Proses Sentiment Analysis untuk Sebuah Brand*. Kelas.Work. https://kelas.work/blogs/tujuan-hingga-proses-sentiment-analysis-untuk-sebuah-brand

Radjah, E. G., & Talakua, A. C. (2024). *Analisis Sentimen Komentar Terhadap Konten Tenun NTT di Youtube Menggunakan Metode SMOTE dan Logistic Regression: Vol. XIII* (Issue 2). https://ojs.unkriswina.ac.id/index.php/transformatif/article/view/1005

Saragih, A. S. (2024, February 23). *Agak Laen: Pengemasan Permasalahan Sosial yang Dikemas dengan Apik*. Wacana. https://wacana.org/agak-laen-pengemasan-permasalahan-sosial-orang-sumut-yang-dikemas-dengan-apik/

Septian, B. (2023, February 16). *Apa Itu Analisis Sentimen : Pengertian, Tipe, dan Cara Kerjanya*. KAZEE. https://blog.kazee.id/apa-itu-analisis-sentimen

Setyawan, N. H., & Wakhidah, N. (2025). ANALISIS PERBANDINGAN METODE LOGISTIC REGRESSION, RANDOM FOREST, GRADIENT BOOSTING UNTUK PREDIKSI DIABETES. *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *10*(1), 150–162. https://doi.org/10.29100/jipi.v10i1.5743

Stihec, J. (2024, June 13). *Random Forests in Machine Learning for Advanced Decision-Making*. Shelf. https://shelf.io/blog/random-forests-in-machine-learning/

Triyantono, Y. S., Faraby, S. Al, & Dwifebri, M. (2021). Analisis Sentimen Terhadap Ulasan Film Menggunakan Word2vec Dan Svm. *EProceedings of Engineering*, *8*(4).

Wijaya, E. C. D., Wacana, K. S., & Diponegoro, J. (2023). *Analisis Sentimen Terhadap Film Sri Asih Dengan CNN, KNN, dan Logistic Regression Sentiment Analysis of Film Sri Asih with CNN, KNN, and Logistic Regression* (Vol. 15, Issue 3). https://csridjournal.potensi-utama.org/index.php/CSRIDjournal/article/download/44/27

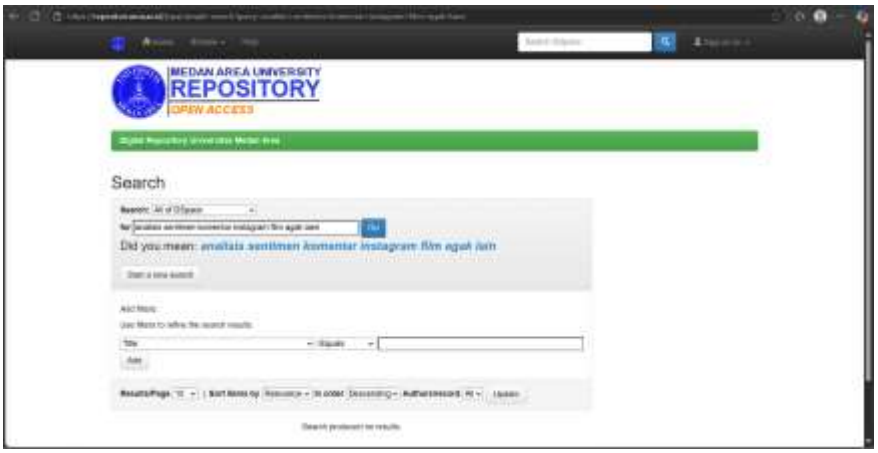
Yonatan, A. Z. (2022, December 23). *Mengenal Google Colab, Cara Menggunakan, dan Keuntungannya*. Detikbali. https://www.detik.com/bali/berita/d-6476973/mengenal-google-colab-cara-menggunakan-dan-keuntungannya

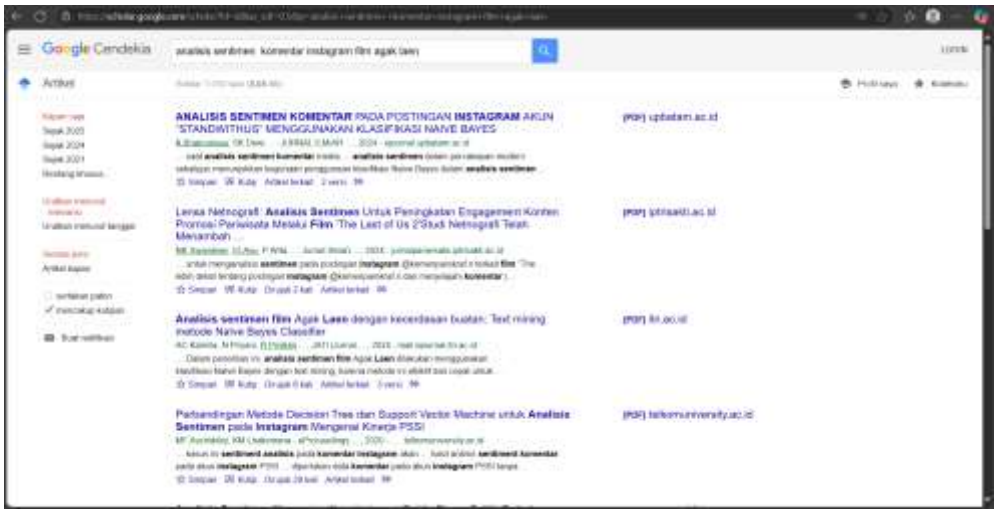
(Agustia & Suryono, 2025; Agustina & Hendry, 2021; Alfarizi et al., 2023; Amaliah et al., 2022; Ardika & Wibawa, 2022; Averina et al., 2022; Basit, 2024; Britanthia et al., 2020; Digital Skola, 2023; Dwi, 2024; *Export Instagram Followers with IGExport*, n.d.; Fadhillah et al., 2025; Junianto et al., 2024; Kurniawati, 2023; M. Azhar N.H, 2024; Meilana, 2025; Milagista, 2024; Oliver, 2022; Permana & Bunyamin, 2024; Putri, 2024; Radjah & Talakua, 2024; Saragih, 2024; Septian, 2023; Setyawan & Wakhidah, 2025; Stihec, 2024; Triyantono et al., 2021; Wijaya et al., 2023; Yonatan, 2022)

LAMPIRAN

Lampiran 1.

Screenshoot menunjukkan belum adanya kajian tentang analisis sentimen terhadap film Agak Laen pada komentar Instagram (12 Juli 2025).



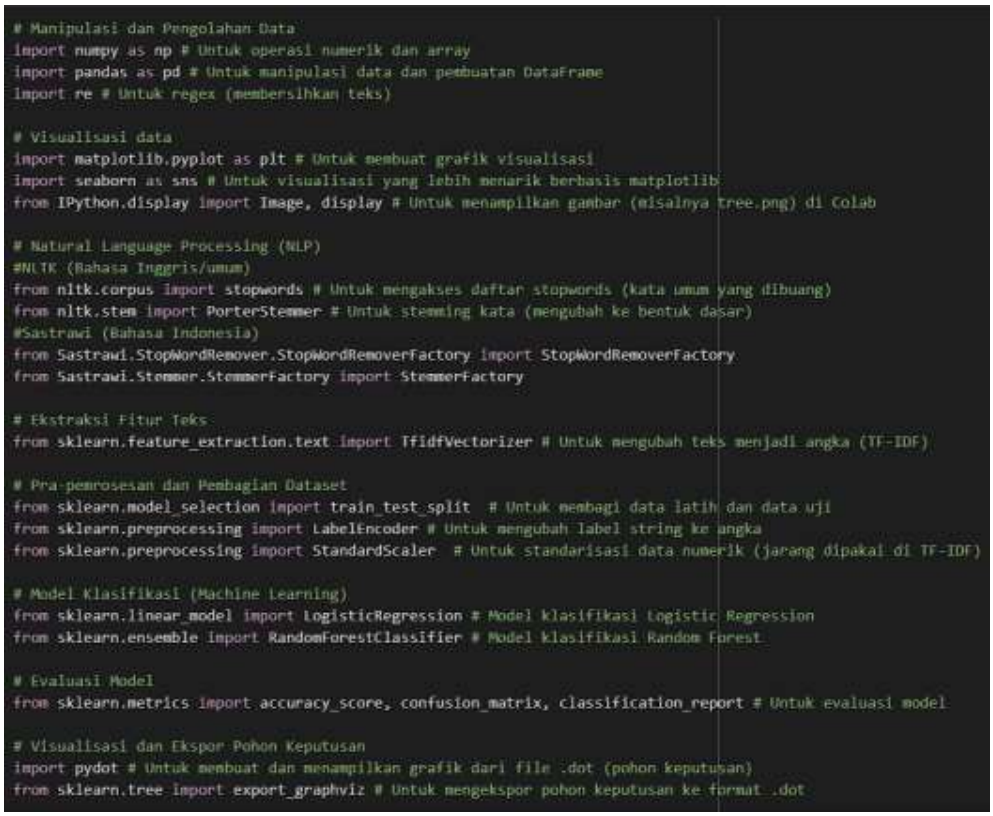

Google Cendekia
analisis sentimen komentar instagram film agak laen
Artikel
ANALISIS SENTIMEN KOMENTAR PADA POSTINGAN INSTAGRAM AKUN 'STANDWITHUS' MENGGUNAKAN KLASIFIKASI NAIVE BAYES
Analisis Sentimen Untuk Peningkatan Engagement Konten Promosi Pariwisata Melalui Film 'The Last of Us 2'Studi Netnografi Terah Menambah
Analisis sentimen film Agak Laen dengan kecerdasan buatan: Text mining metode Naive Bayes Classifier
Perbandingan Metode Decision Tree dan Support Vector Machine untuk Analisis Sentimen pada Instagram Mengenai Kinerja PSSI


JMS
JURNAL
MATEMATIKA KOMPUTASI DAN STATISTIKA
ISSN : 2503-2984
HOME
CURRENT
ARCHIVES
ABOUT
Search
analisis sentimen komentar instagram film agak laen
ADVANCED FILTERS
By Author
ADDITIONAL MENUS

Lampiran 2.

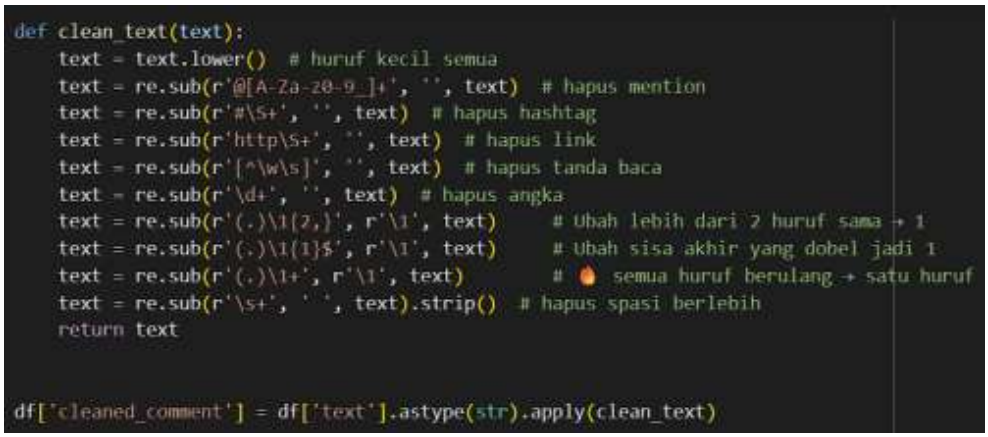
Source code atau script pemrosesan data menggunakan Bahasa pemrograman python.

1. **Library yang digunakan**



```
# Manipulasi dan Pengolahan Data
import numpy as np # Untuk operasi numerik dan array
import pandas as pd # Untuk manipulasi data dan pembuatan DataFrame
import re # Untuk regex (membersihkan teks)

# Visualisasi data
import matplotlib.pyplot as plt # Untuk membuat grafik visualisasi
import seaborn as sns # Untuk visualisasi yang lebih menarik berbasis matplotlib
from IPython.display import Image, display # Untuk menampilkan gambar (misalnya tree.png) di Colab

# Natural Language Processing (NLP)
#NLTK (Bahasa Inggris/umum)
from nltk.corpus import stopwords # Untuk mengakses daftar stopwords (kata umum yang dibuang)
from nltk.stem import PorterStemmer # Untuk stemming kata (mengubah ke bentuk dasar)
#Sastrawi (Bahasa Indonesia)
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Ekstraksi Fitur Teks
from sklearn.feature_extraction.text import TfidfVectorizer # Untuk mengubah teks menjadi angka (TF-IDF)

# Pra-pemrosesan dan Pembagian Dataset
from sklearn.model_selection import train_test_split  # Untuk membagi data latih dan data uji
from sklearn.preprocessing import LabelEncoder # Untuk mengubah label string ke angka
from sklearn.preprocessing import StandardScaler  # Untuk standarisasi data numerik (jarang dipakai di TF-IDF)

# Model Klasifikasi (Machine Learning)
from sklearn.linear_model import LogisticRegression # Model klasifikasi Logistic Regression
from sklearn.ensemble import RandomForestClassifier # Model klasifikasi Random Forest

# Evaluasi Model
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report # Untuk evaluasi model

# Visualisasi dan Ekspor Pohon Keputusan
import pydot # Untuk membuat dan menampilkan grafik dari file .dot (pohon keputusan)
from sklearn.tree import export_graphviz # Untuk mengekspor pohon keputusan ke format .dot
```

2. **Data Cleaning**



```
def clean_text(text):
    text = text.lower()  # huruf kecil semua
    text = re.sub(r'@[A-Za-z0-9_]+', '', text)  # hapus mention
    text = re.sub(r'#\S+', '', text)  # hapus hashtag
    text = re.sub(r'http\S+', '', text)  # hapus link
    text = re.sub(r'[^\w\s]', '', text)  # hapus tanda baca
    text = re.sub(r'\d+', '', text)  # hapus angka
    text = re.sub(r'(.)\1{2,}', r'\1', text)      # Ubah lebih dari 2 huruf sama → 1
    text = re.sub(r'(.)\1{1}$', r'\1', text)      # Ubah sisa akhir yang dobel jadi 1
    text = re.sub(r'(.)\1+', r'\1', text)         # 🔥 semua huruf berulang → satu huruf
    text = re.sub(r'\s+', ' ', text).strip()  # hapus spasi berlebih
    return text


df['cleaned_comment'] = df['text'].astype(str).apply(clean_text)
```

3. Data Cleaning - Normalisasi Data

```
normalization_dict = {
    "anj": "anjing", "gk": "gak", "ga": "gak", "yg": "yang", "ato": "atau", "bgt": "banget
    "abg": "abang,", "silper": "silver", "prlu": "perlu", "abangabang": "abang", "abangaba
    "ahahaha": "hahaha", "ahahahahahah": "hahaha", "ahahahahahahaha": "hahaha", "akwkwkwkw
    "wkwkw": "wkwk", "wkwkwk": "wkwk", "wkwkwkw": "wkwk", "wkwkwkwk": "wkwk", "wkwkwkwkw":
    "kwkwkw": "wkwk", "wkakwkwkwkwk": "wkwk", "wkakaka": "wkwk", "wkokokoko": "wkwk", "wkw
    "bangat": "banget", "asyekmantap": "mantap", "mantab": "mantap", "congrats": "sukses",
    "ytb" : "youtube", "anjeng": "anjing", "aprl": "april", "anying": "anjing","awaq":"awa
    "bg": "abang", "beneran": "benar", "bener": "benar", "bikinin": "bikin", "bjir": "anji
    "inih":"ini", "asalamualaikum": "assalamualaikum", "wkwkwkmenyala": "wkwk", "wkwkwkwkw
    "hahahahahahaha ": "haha", "hahahahahahah": "haha", "hahahahahahahahah": "haha", "
    "hahahahahahahahahahahah": "haha", "hahahahahahahahahahaha  ": "haha", "jgn": "jan
}

def normalize_text(text):
    words = text.split()
    normalized_words = [normalization_dict.get(word, word) for word in words]
    return ' '.join(normalized_words)

df['normalized'] = df['cleaned_comment'].astype(str).apply(normalize_text)
```

4. Preprocessing Data - Tokenizing

```
# Fungsi tokenisasi sederhana (split berdasarkan spasi)
def tokenize(text):
    return text.split()

# Terapkan ke kolom 'stemmed'
df['tokens'] = df['normalized'].astype(str).apply(tokenize)
```

5. Preprocessing Data - Stopword Removal

```
import nltk
nltk.download('stopwords')

from nltk.corpus import stopwords
stopwords_ind = set(stopwords.words('indonesian'))
stopwords_ind.discard('agak')

#nama_orang = ['boris', 'bokir', 'ernest', 'jegel', 'indra', 'sadana', 'agung', 'bene']
kata_tak_bermakna = {'ach', 'aci', 'ah', 'al', 'amago', 'aw', 'be', 'an', 'apas', 'akh', 'weh'}

def remove_stopwords(tokens):
    tokens = [word.lower() for word in tokens]  # pastikan lowercase
    tokens = [word for word in tokens if word not in stopwords_ind]
    #tokens = [word for word in tokens if word not in nama_orang]
    tokens = [word for word in tokens if word not in kata_tak_bermakna]
    return ' '.join(tokens)

df['no_stopwords'] = df['tokens'].apply(remove_stopwords)
```

6. **Preprocessing Data – Stemming**

```
# Buat stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
    return stemmer.stem(text)

df['stemmed'] = df['no_stopwords'].astype(str).apply(stem_text)
```

7. **Labeling Data**

```
#LABEL ENCODING✓

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sentimen'] = le.fit_transform(df['Sentimen'])  # pastikan kolomnya benar
#0 = negatif, 1 = netral, 2 = positif
```

8. **Pembagian Data Latih dan Data Uji**

```
#SPLIT DATA (DATA UJI & DATA LATIH)✓

from sklearn.model_selection import train_test_split

X = df['stemmed'].apply(lambda x: ' '.join(x) if isinstance(x, list) else (x if isinstance(x, str) else ''))
y = df['Sentimen']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42, stratify=y)
#Untuk mengatur jumlah data, dimana yg direkomendasikan 80%:20% tetapi jika mengikuti itu, hasilnya sama.

# Cek jumlah data
print("Jumlah data latih:", len(X_train))
print("Jumlah data uji:", len(X_test))
```

9. **Ekstraksi Fitur**

```
#Ekstraksi Fitur: TF-IDF✓

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=300)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

#Melihat data uji setelah TF-IDF✓

print("Shape X_train_tfidf:", X_train_tfidf.shape)
print("Shape X_test_tfidf:", X_test_tfidf.shape)
```

10. Implementasi Metode – Logistic Regression

```
#Logistic Regression✓
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

lr_model = LogisticRegression()
lr_model.fit(X_train_tfidf, y_train)
y_pred_lr = lr_model.predict(X_test_tfidf)
```

11. Implementasi Metode – Random Forest

```
#RANDOM FOREST✓

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train_tfidf, y_train)
y_pred_rf = rf_model.predict(X_test_tfidf)
```

12. Evaluasi Metode – Logistic Regression

```
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred_lr)
labels = lr_model.classes_

plt.figure(figsize=(6,5))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Logistic Regression')
plt.show()

# Classification Report (bar chart)
report = classification_report(y_test, y_pred_lr, output_dict=True)
report_df = pd.DataFrame(report).transpose()

# Ambil hanya kelas (tanpa avg/accuracy)
report_df_class = report_df.iloc[:-3, :]

report_df_class[['precision', 'recall', 'f1-score']].plot(kind='bar', figsize=(8,6))
plt.title('Precision, Recall, F1-score per Class - Logistic Regression')
plt.ylim(0,1)
plt.ylabel('Score')
plt.xticks(rotation=45)
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

13. Evaluasi Metode – Random Forest

```
rf = RandomForestClassifier(n_estimators=10, random_state=42)
rf.fit(X_train_tfidf, y_train)
feature_list = vectorizer.get_feature_names_out().tolist()
train_features = pd.DataFrame(X_train_tfidf.toarray(), columns=feature_list)
tree = rf.estimators_[1]

# 11. Ekspor ke .dot
export_graphviz(tree,
                out_file='tree.dot',
                feature_names=feature_list,
                rounded=True,
                precision=2,
                max_depth=5)  # Agar tidak terlalu besar

# 12. Convert ke .png dan tampilkan

print("Jumlah fitur dalam train_features:", train_features.shape[1])
print("Jumlah nama fitur dalam feature_list:", len(feature_list))

(graph,) = pydot.graph_from_dot_file('tree.dot')
graph.write_png('tree.png')
display(Image(filename='tree.png'))
```

14. Perbandingan Hasil Metode

```
#Perbandingan Hasil√

from sklearn.metrics import f1_score

print("Akurasi Logistic Regression:", accuracy_score(y_test, y_pred_lr))
print("F1-Score Logistic Regression:", f1_score(y_test, y_pred_lr, average='weighted'))

print("Akurasi Random Forest:", accuracy_score(y_test, y_pred_rf))
print("F1-Score Random Forest:", f1_score(y_test, y_pred_rf, average='weighted'))
```

15. Perbandingan Hasil MAPE

```python
import pandas as pd
import plotly.graph_objects as go

# Data evaluasi analisis sentimen
data = {
    'Metrik Evaluasi': ['Akurasi', 'Macro Avg F1', 'Weighted Avg F1'],
    'Logistic Regression': [0.96, 0.33, 0.94],
    'Random Forest': [0.98, 0.55, 0.98]
}

df = pd.DataFrame(data)

# Bar chart
fig = go.Figure()

# Bar Logistic Regression (warna hijau)
fig.add_trace(go.Bar(
    x=df['Metrik Evaluasi'],
    y=df['Logistic Regression'],
    name='Logistic Regression',
    marker_color='green',
    text=[f"{val*100:.2f}%" for val in df['Logistic Regression']],
    textposition='auto'
))

# Bar Random Forest (warna biru)
fig.add_trace(go.Bar(
    x=df['Metrik Evaluasi'],
    y=df['Random Forest'],
    name='Random Forest',
    marker_color='blue',
    text=[f"{val*100:.2f}%" for val in df['Random Forest']],
    textposition='auto'
))

# Tambahkan delta (perbedaan nilai antar metode)
for i in range(len(df)):
    x_val = df['Metrik Evaluasi'][1]
    delta = (df['Random Forest'][1] - df['Logistic Regression'][1]) * 100
    fig.add_annotation(
        x=x_val,
        y=max(df['Random Forest'][1], df['Logistic Regression'][1]) + 0.01,
        text=f"Δ {abs(delta):.2f}%",
        showarrow=False,
        font=dict(size=12)
    )

# Layout
fig.update_layout(
    title='MAPE Comparison: Linear Regression vs Random Forest',
    xaxis_title='Metrik Evaluasi',
    yaxis_title='Skor (%)',
    barmode='group',
    yaxis=dict(tickformat=".0%", range=[0, 1.1]),
    legend_title='Metode Klasifikasi',
    height=500
)

fig.show()
```