ANALISIS SENTIMEN UJARAN KEBENCIAN TERHADAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK

SKRIPSI

DISUSUN OLEH

NAZWA PUTRI ANANDA

2109010030

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

LEMBAR PENGESAHAN

Judul Skripsi	:	ANALISIS UJARAN KEBENCIAN TERHDAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK
Nama Mahasiswa	:	NAZWA PUTRI ANANDA
NPM	:	22109010030
Program Studi	:	SISTEM INFORMASI

Menyetujui
Komisi Pembimbing

(Dr. Firahmi Rizky, S.Kom., M.Kom.)
NIDN. 0116079201

Ketua Program Studi

(Dr. Firahmi Rizky, S.Kom., M.Kom.)
NIDN. 0128029302

Dekan

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

PERNYATAAN ORISINALITAS

ANALISIS SENTIMEN UJARAN KEBENCIAN TERHADAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK

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

Nazwa Putri Ananda

NPM. 2109010030

PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS

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

Nama	: Nazwa Putri Ananda
NPM	2109010030
Program Studi	: Sistem 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:

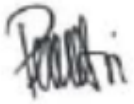
ANALISIS SENTIMEN UJARAN KEBENCIAN TERHADAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK

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



Nazwa Putri Ananda

NPM. 2109010030

PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS

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

Nama	: Nazwa Putri Ananda
NPM	2109010030
Program Studi	: Sistem 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 SENTIMEN UJARAN KEBENCIAN TERHADAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK

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

Nazwa Putri Ananda

NPM. 2109010030

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Nazwa Putri Ananda
Tempat dan Tanggal Lahir	: Medan, 09 Desember 2003
Alamat Rumah	: JL. Bahagia GG Angkir NO 68A Medan
Telepon/Faks/HP	087896886427
E-mail	: nazwaputriananda123@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SD Swasta Cerdas Bangsa	TAMAT: 2015
SMP	: SMP Swasta Cerdas Bangsa	TAMAT: 2018
SMA	: SMA Swasta Cerdas Bangsa	TAMAT: 2021

KATA PENGANTAR

Puji syukur Alhamdulillah, penulis panjatkan kehadiratnya kepada Allah SWT, yang telah melimpahkan banyak rahmat dan karunia-Nya serta memberi kekuatan kepada penulis untuk menyelesaikan tugas akhir dalam meraih strata 1 ini. Skripsi ini penulis sajikan dalam bentuk buku yang sederhana. Judul skripsi pada penelitian ini adalah sebagai berikut, **"ANALISIS SENTIMEN UJARAN KEBENCIAN PADA DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK"**.

Adapun tujuan penulisan skripsi ini dibuat sebagai salah satu syarat kelulusan Program Strata Satu (S1) Sistem Informasi Universitas Muhammadiyah Sumatera Utara. 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 Assoc. Prof. Dr. Al-Khowarizmi, M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Bapak Halim Maulana., ST., M.Kom sebagai Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
4. Bapak Lutfi Basit, S.Sos., M.I.Kom sebagai Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Ibu Dr. Firahmi Rizky, S.Kom, M.Kom sebagai Ketua Program Studi Sistem Informasi.
6. Ibu Dr. Firahmi Rizky, S.Kom, M.Kom selaku Dosen Pembimbing penulis yang selalu memberikan arahan dan semangat dalam mengerjakan skripsi. Bimbingan dan masukan yang sangat berharga telah memberikan kontribusi

besar untuk penulis dalam menyelesaikan skripsi dengan tepat waktu. Terimakasih sebesar-besarnya atas waktu, ilmu, kesabaran dalam setiap proses penulis. Semoga Allah SWT senantiasa melimpahkan Kesehatan, keberkahan, serta balasan yang terbaik atas kebaikan yang telah ibu berikan.

7. Teristimewa kepada papa dan mama terimakasih selalu berjuang untuk kehidupan penulis hingga saat ini, yang selalu mendidik, memotivasi dan memberikan dukungan yang sangat besar hingga penulis mampu menyelesaikan studi ini hingga akhir. Yang tidak pernah lelah mendoakan, mengusahakan dan memberi cinta dan kasih sayang yang tulus untuk kesuksesan penulis.
8. Untuk Rizky Wira Nanda Pasaribu terimakasih telah menjadi penyemangat, membantu dan mendengarkan semua keluh kesah penulis dari awal semester hingga akhir semester.
9. Kepada seluruh teman-teman KKN penulis, yang saling mendukung satu sama lain dalam penulisan skripsi.
10. Kepada diri sendiri terimakasih telah kuat sampai detik ini, berjuang sejauh ini dan berusaha keras agar bisa sampai di titik ini. Serta semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

ANALISIS UJARAN KEBENCIAN TERHDAP DRIVER GOJEK MENGGUNAKAN ALGORITMA NAÏVE BAYES DI PLATFORM FACEBOOK

ABSTRAK

Media sosial saat ini menjadi tempat banyak orang menyampaikan pendapat secara bebas, termasuk komentar negatif yang kadang mengarah pada ujaran kebencian. Salah satu pihak yang sering menjadi sasaran ujaran kebencian adalah driver Gojek. Penelitian ini dilakukan untuk mengelompokkan komentar pengguna Facebook menjadi dua kategori sentimen, yaitu positif dan negatif, dengan fokus utama pada komentar negatif. Data yang digunakan diperoleh dari komentar pada beberapa postingan publik di platform Facebook menggunakan bantuan tools scraping APIFY. Setelah data terkumpul, dilakukan proses preprocessing data, yaitu case folding, pembersihan data (cleaning), tokenisasi, normalisasi, penghapusan stopword, dan stemming. Selanjutnya, data diubah ke bentuk numerik menggunakan CountVectorizer. Algoritma klasifikasi yang digunakan adalah Naive Bayes dengan model MultinomialNB karena data yang digunakan berbentuk frekuensi kata. Hasil evaluasi model menunjukkan bahwa algoritma ini mampu mengklasifikasikan komentar negatif dengan cukup baik, terutama dalam mengenali pola kata yang sering muncul dalam ujaran kebencian terhadap driver Gojek.

Kata Kunci : Sentimen, Ujaran Kebencian, Faacebook, Driver Gojek, Naïve Bayes, MultinomialNB

ANALYSIS OF HATE SPEECH AGAINST GOJEK DRIVERS USING THE NAÏVE BAYES ALGORITHM ON THE FACEBOOK PLATFORM

ABSTRACT

Social media has become a space where many individuals express their opinions freely, including negative comments that may lead to hate speech. One group often targeted by such speech is Gojek drivers. This study aims to classify user comments on Facebook into two sentiment categories: positive and negative, with a primary focus on negative comments. The data was collected from public Facebook posts using the APIFY scraping tool. After the data was gathered, several preprocessing stages were carried out, including case folding, cleaning, tokenization, normalization, stopword removal, and stemming. The text data was then converted into numerical form using CountVectorizer. The classification algorithm used in this research is Naive Bayes with the MultinomialNB model, as the input data consists of word frequency. The results of the model evaluation show that this algorithm performs well in classifying negative comments, especially in identifying word patterns that commonly appear in hate speech directed toward Gojek drivers.

Keywords *: Sentiment, Hate Speech, Facebook, Gojek, Naïve Bayes, MultinomialNB*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Transportasi *online* merupakan layanan transportasi yang menggunakan internet dan dapat diakses melalui aplikasi yang ada di *smartphone*. Hal tersebut telah mengubah lanskap transportasi di Indonesia, terutama peralihan ojek pangkalan ke ojek *online*. Ojek pangkalan telah menjadi bagian penting dari layanan transportasi lokal di Indonesia. Namun, munculnya aplikasi ojek *online* membuat akses perjalanan menjadi lebih mudah dengan platform digital. Dengan transformasi besar ini, perusahaan teknologi memiliki peluang besar untuk digitalisasi layanan ojek. Salah satu contoh transformasi digital yang berhasil adalah Gojek. Gojek merupakan aplikasi yang dikembangkan oleh PT. Karya Anak Bangsa, mitra yang membantu Gojek dalam menjalankan bisnisnya. Gojek menawarkan layanan pesan antar makanan, transportasi roda dua dan roda empat, dan berbagai layanan lainnya. Dalam keberhasilannya Gojek mempunyai strategi kuat untuk mempertahankan kualitas dan kepercayaan pelanggan. Pengalaman pengguna yang optimal menjadi salah satu startegi utama yang Gojek terapkan. Pengguna Gojek dapat membagikan pengalaman mereka di berbagai platform media sosial. Salah satu platform media sosial yang paling populer adalah Facebook, di mana para pengguna dapat membagikan pengalaman mereka melalui postingan dan komentar. Mereka dapat menilai pengalaman mereka dengan nilai positif, *negative*, maupun netral. Ulasan positif, seperti pujian dan apresiasi, atau ulasan *negative*, seperti keluhan terhadap pelayanan aplikasi dan *driver*.

Driver Gojek adalah orang yang terdaftar sebagai mitra pengemudi di platform Gojek dan bertugas mengantar penumpang (GoRide,GoCar), mengirim makanan (GoFood), atau barang (GoSend). Mereka bekerja berdasarkan permintaan pelanggan melalui aplikasi Gojek. Menjadi *driver* Gojek memiliki banyak keuntungan, terutama dalam hal *fleksibilitas* dan peluang penghasilan yang lebih besar. *Driver* dapat menentukan sendiri waktu kerja mereka tanpa terikat oleh aturan kantor, yang membuatnya sesuai untuk orang yang ingin bekerja sambil melakukan aktivitas lain. Selain itu, agar *driver* dapat mencapai target tertentu, Gojek menawarkan berbagai bonus dan insentif. Ini memungkinkan Gojek untuk menghasilkan lebih banyak uang. Kemudahan pendaftaran dan ketersediaan berbagai layanan, seperti asuransi kecelakaan dan bantuan darurat, menambah rasa aman saat bekerja. Dengan meningkatnya permintaan untuk layanan pengantaran dan transportasi, menjadi *driver* Gojek mungkin menjadi karir yang menguntungkan. Namun, di balik berbagai keuntungan tersebut, menjadi *driver* Gojek juga memiliki tantangan tersendiri.

Salah satu tantangan yang cukup sulit bagi driver Gojek adalah ulasan *negative,* termasuk ujaran kebencian pelanggan. Setiap penilaian dan komentar pelanggan dalam sistem kerja berbasis aplikasi sangat memengaruhi reputasi *driver*. Sayangnya, tidak semua ulasan diberikan secara adil karena terkadang pelanggan memberikan rating rendah atau komentar buruk tanpa alasan yang jelas. Beberapa pelanggan bahkan melakukan ujaran kebencian dengan kata-kata kasar, hinaan, atau tuduhan yang tidak berdasar, yang dapat berdampak pada psikologis driver dan peluang mereka untuk mendapatkan orderan di masa depan. Meskipun Gojek memiliki sistem untuk meninjau laporan tentang ulasan yang tidak adil,

driver harus tetap profesional meskipun menghadapi ulasan yang tidak menyenangkan (Fani & Ardiansah, n.d.).

Menurut (Arfan et al., 2024) ujaran kebencian merupakan tindakan mengintimidasi atau perlakuan kasar yang mempunyai sifat intrinsik seperti karakter *bullying* yang berupa tindakan agresif, disengaja, dan berulang-ulang dari waktu ke waktu oleh seseorang atau kelompok kepada orang lain yang dilakukan di media elektronik secara berkelanjutan. Untuk memahami pola dan tingkat ujaran kebencian yang terjadi, diperlukan suatu metode yang dapat menganalisis sentimen dari komentar yang ditujukan kepada driver Gojek di platform Facebook. Salah satu pendekatan yang efektif adalah dengan menggunakan analisis sentimen berbasis algoritma Naïve Bayes, yang merupakan metode klasifikasi probabilistik dalam *machine learning*. Algoritma ini mampu mengklasifikasikan komentar atau ulasan menjadi sentimen positif, *negative*, atau netral dengan tingkat akurasi yang tinggi. Analisis sentimen adalah proses otomatis untuk mengelola data teks untuk menganalisis pendapat seseorang dalam bahasa tertulis. Hal ini dilakukan untuk mengidentifikasi ulasan terhadap subjek dan objek seperti individu, organisasi, dan produk layanan. Analisis sentimen dapat membantu memahami opini dan komunikasi masyarakat di media sosial (Hasri & Alita, 2022).

Analisis sentimen dilakukan dengan dua cara, yang pertama berbasis *machine learning* yang melatih *classifier* pada dataset yang sudah dilabelkan secara manual. Yang kedua berbasis leksikal, tidak memerlukan pelatihan dataset, mengukur polaritas dokumen atau kalimat berdasarkan sentimen kata-kata dan frasa menggunakan aturan Iinguistik tertentu (Arfan et al., 2024). Untuk melakukannya, ulasan harus diklasifikasikan menjadi kategori netral, positif, dan *negative*. Ulasan

negative mengandung unsur ujaran kebencian, sedangkan ulasan positif mengandung pujian atau dukungan.

Berdasarkan latar belakang tersebut, penelitian ini berfokus pada analisis sentimen ujaran kebencian terhadap driver Gojek di platform Facebook menggunakan algoritma Naïve Bayes. Hasil penelitian ini diharapkan dapat memberikan pemahaman lebih mendalam mengenai persepsi masyarakat terhadap *driver* Gojek serta menjadi bahan evaluasi bagi perusahaan dalam meningkatkan perlindungan terhadap mitra *driver* dari dampak *negative* media sosial.

1.1. Rumusan Masalah

Berdasarkan masalah yang diuraikan diatas maka penulis merumuskan masalah, yaitu :

1. Bagaimana menganalisis sentimen *assessment* terhadap *driver* gojek di platform Facebook?
2. Bagaimana menguji *negative* sentimen menggunakan metode Naïve Bayes?
3. Bagaimana implementasi metode Naïve Bayes dalam mengelompokan *negative* sentimen

1.2. Batasan Masalah

Dalam penelitian ini, terdapat beberapa batasan yang perlu ditetapkan agar dapat memberikan fokus yang jelas terhadap penelitian. Batasan tersebut adalah :

1. Penelitian ini hanya menganalisis komentar dari postingan di platform media sosial Facebook yang berkaitan dengan *driver* Gojek dari tahun 2024 sampai 2025.

1.3. Tujuan Penelitian

Penelitian ini bertujuan untuk mencapai beberapa target sebagai berikut :

1. Untuk menganalisis sentimen *assessment* terhadap *driver* ojek di platform Facebook
2. Untuk menguji *negative* sentimen menggunakan metode Naïve Bayes
3. Untuk mengimplementasikan metode Naïve Bayes dalam *negative* sentimen

1.4. Manfaat Penelitian

Adapun manfaat yang di hasilkan dari penelitian ini antaranya :

1. Membantu memahami pola dan tingkat ujaran kebencian yang terjadi terhadap *driver* Gojek di platform Facebook, sehingga dapat memberikan gambaran tentang sejauh mana komentar *negative* memengaruhi mitra *driver*.

BAB II

LANDASAN TEORI

2.1. Analisis Sentimen

Analisis sentimen, merupakan proses menganalisis sejumlah data teks besar untuk menentukan apakah mereka mengandung sentimen positif, *negative*, atau netral dari berbagai platform di media sosial. Analisis sentimen dikenal juga sebagai *opinion mining* atau *emotional artificial intelligence*, melihat dari bagaimana sikap, pemikiran, dan emosi di balik sebuah teks yang terdapat pada dokumen atau data (Elfansyah et al., 2024).

Analisis sentimen sangat penting bagi perusahaan, terutama untuk meningkatkan kualitas produk atau jasa layanan. Dengan memanfaatkan analisis sentimen, perusahaan bisa mengetahui opini masyarakat terhadap produk atau jasa layanan tersebut. Hal ini merupakan *feedback* bagi perusahaan agar dapat meningkatkan pelayanan maupun pendekatan khusus kepada *customer*.

Analisis sentimen menggunakan teknologi bahasa alami (NLP) dan *machine learning* (ML) untuk melatih perangkat lunak komputer agar dapat menganalisis teks dengan cara yang mirip dengan manusia. Perangkat lunak ini menggunakan salah satu dari dua pendekatan berbasis aturan atau ML, atau kombinasi dari keduanya yang dikenal sebagai hibrida. Setiap pendekatan berbasis aturan dapat memberikan hasil yang hampir *real-time*, pendekatan berbasis ML lebih mudah beradaptasi dan biasanya dapat menangani masalah yang lebih kompleks. Pendekatan hibrida menghasilkan hasil yang lebih cepat, akurat, dan dapat disesuaikan dengan perubahan bahasa dan konteks teks.

Menurut (Pratama et al., 2024) Analisis sentimen mengacu pada topik tertentu. Satu pernyataan bisa memiliki sentimen berbeda tergantung topiknya, oleh karena itu, topik sangat menentukan bagaimana sistem memahami suatu pernyataan. Dalam beberapa penelitian, pekerjaan dimulai dengan menentukan, terutama dalam *review* ulasan aplikasi.

2.2. Naïve Bayes

Algoritma Naïve Bayes adalah teorema probabilitas yang menghitung probabilitas kondisional secara terbalik, atau kemampuan teorema untuk memperbarui keyakinan tentang suatu peristiwa setelah melihat bukti atau informasi baru. Algoritma Naïve Bayes dikemukakan oleh ilmuan inggris Thomas Bayes yang memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya (Retnosari, 2021). Metode Bayes menganggap tolak ukur sebagai variable yang menunjukkan pengetahuan awal tentang tolak ukur sebelum pengamatan, yang ditunjukkan dalam bentuk nilai yang disebut distribusi prior (Cybertech et al., 2020). Algoritma ini sering digunakan dalam analisis teks karena kesederhanaannya dan efisiensinya dalam menangani data berukuran besar. Naïve Bayes sering digunakan dalam analisis sentimen untuk mengklasifikasikan teks menjadi kategori positif, *negative*, atau netral.

Dalam konteks klasifikasi, algoritma Naïve Bayes menggabungkan Teorema Bayes dengan asumsi bahwa setiap fitur dalam data independen satu sama lain, walaupun dalam kenyataannya hal ini tidak sepenuhnya benar. Algoritma Naïve Bayes sering digunakan untuk mengklasifikasikan dan menganalisis teks atau dokumen. Kata-kata atau token yang muncul dalam teks atau dokumen tersebut, disebut sebagai model "Bag-of-Words" atau "BoW" (Rahayu, 2023).

Seperti yang disebutkan sebelumnya, Naïve Bayes menggunakan Teorema Bayes, sebuah formula matematika sederhana, untuk probabilitas bersyarat atau probabilitas kondisional. Dengan kata lain, probabilitas bersyarat adalah ukuran kemungkinan bahwa suatu peristiwa akan terjadi karena peristiwa lain telah terjadi sebelumnya. Ini dapat berupa bukti, asumsi, praduga, atau pernyataan. Berikut adalah rumus Teorema Naïve Bayes :

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.1)$$

Dimana :

1) $P(H|X)$: Peluang kelas H (contoh: komentar *negative*) terjadi setelah diketahui data X (*posterior probability*).
2) P(X|H) : Seberapa besar kemungkinan data X muncul jika memang berasal dari kelas H (*likelihood*).
3) P(H) : Peluang awal suatu kelas sebelum melihat data (*prior probability*).
4) P(X) : Total probabilitas munculnya data X, tanpa memperhatikan kelasnya.

Perlu di ingat bahwa algoritma Naïve Bayes membuat asumsi yang sangat kuat jika setiap fitur bersifat *independent*. Artinya, kita bisa menghitung peluang berdasarkan kategori pada atribut terhadap kelasnya. Keuntungan algoritma Naïve Bayes adalah ketika digunakan pada kumpulan data yang besar, mereka memiliki tingkat kesalahan yang rendah. Selain itu, ketika digunakan pada kumpulan data yang lebih besar, mereka meningkatkan kecepatan dan akurasi (Dzikri et al., 2024). Selain itu, kelemahan algoritma Naïve Bayes terletak pada masalah probabilitas. Naïve Bayes memiliki masalah probabilitas nol, terutama saat menemukan kata-kata dalam data pengujian untuk kelas tertentu yang tidak ada dalam data pelatihan.

2.3. Ujaran Kebencian

Ujaran kebencian merupakan pernyataan atau ekspresi yang mengandung kebencian, permusuhan, atau diskriminasi terhadap individu atau kelompok. Hal ini bisa terjadi melalui platform digital, seperti media sosial, forum online, blog, dan *chat room*. Ujaran kebencian yang dimaksud dalam penelitian ini mencakup komentar sentimen *negative* yang mengandung unsur ujaran kebencian pada komentar terntentu di platform Facebook. Bentuk-bentuk ujaran kebencian sangat beragam dan bisa berdampak *negative* pada korbannya, termasuk driver Gojek yang kerap menerima komentar *negative* di platform seperti Facebook.

Penghinaan fisik dalam ujaran kebencian dapat mencakup berbagai bentuk, tergantung pada konteks dan frekuensi penghinaan, tetapi semuanya dapat berdampak buruk pada korban dari segi psikologis, sosial, dan profesional. Korban ujaran kebencian akan mengalami depresi sedang hingga berat, emosional yang tinggi, rasa tidak percaya diri, dan akhirnya percobaan pumbunuhan atau bunuh diri (Hutagalung et al., 2021). Ujaran kebencian meninggalkan jejak digital sebuah rekaman atau catatan yang dapat berguna dan memberikan bukti ketika membantu menghentikan perilaku salah ini. Menurut (Eleanora & Adawiah, 2021) ujaran kebencian sudah masuk ke dalam kategori tindak pidana, dengan unsur-unsur seperti subjek atau pelaku, perbuatan melanggar hukum, baik karena kelalaian atau kesengajaan, dan aturan yang mengatur waktu, tempat, dan keadaan. Oleh karena itu, penting untuk memahami ujaran kebencian agar kita dapat menggunakannya di masa depan.

2.4. Text Mining

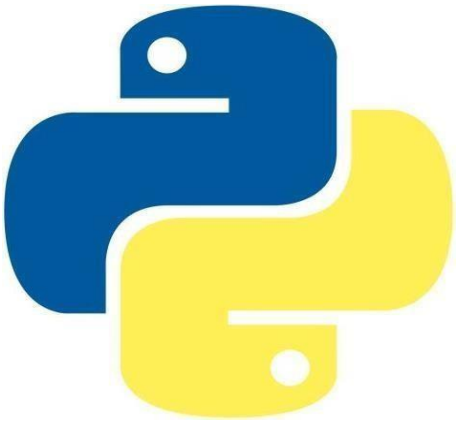
Text mining merupakan proses untuk mendapatkan informasi berkualitas tinggi dari teks dengan memperhatikan pola dan tren dengan mempelajari pola statistik. Proses ini juga melibatkan pembobotan kata, yang bertujuan untuk memberikan nilai atau bobot pada term tertentu dalam dokumen. Bobot yang diberikan pada term tergantung kepada metode yang digunakan. Dalam pembobotan kata banyak sekali terdapat algoritma-algoritma contohnya seperti TF, Idf, RF, TF-IDF, WIDF (Informatika & Amikom, 2019).

Menurut Sholih ’afif et al., (2021) Text mining dan Data mining memiliki tujuan yang hampir sama yaitu menemukan pola pada data sehingga dapat digunakan untuk membantu orang dalam pekerjaan mereka. Yang berbeda adalah sumber data yang akan digunakan, pada proses text mining sumber datanya adalah teks atau dokumen. Perbedaan yang lain adalah proses ekstraksi fitur untuk mengubah data teks menjadi data terstruktur.

2.5. Python

Python adalah bahasa pemograman interpretatif yang dapat menggunakan paradigma pemograman objek, fungsi, atau procedural (Reza et al., 2022). Python, yang dikembangkan oleh Guido van Rossum, sangat populer sebagai bahasa pemrograman Web dan skripting karena sintaksis kodenya yang mudah dipahami dan fitur pustaka standar. Sebagian besar, Python mendukung berbagai paradigma pemrograman. Ini termasuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional.

Python dapat digunakan sebagai bahasa skrip karena memiliki fitur sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis, memungkinkannya memenuhi berbagai keperluan pengembangan perangkat lunak dan mencapai tujuan yang biasanya tidak dapat dicapai dengan bahasa skrip. Salah satu bahasa tingkat tinggi adalah Python, bahasa lainnya seperti Pascal, c++, perl, java, dan sebagainya. Bahasa tingkat rendah seperti bahasa mesin atau bahasa *assembly*. Secara sederhana, komputer tidak dapat menjalankan program yang ditulis dalam bahasa mesin (Nurjanah & Insanudin, 2016).



Gambar 2. 1 Python

2.6. Facebook

Facebook merupakan salah satu media sosial yang paling banyak digunakan di dunia. Platform ini pertama kali diluncurkan pada tahun 2004 oleh Mark Zuckerberg bersama beberapa rekannya di Universitas Harvard. Awalnya hanya digunakan oleh mahasiswa Harvard, tetapi kemudian berkembang pesat dan kini digunakan secara global oleh berbagai kalangan. Facebook memungkinkan penggunanya untuk berbagi status, foto, video, tautan, serta berinteraksi dengan pengguna lain melalui komentar, pesan pribadi, dan fitur-fitur lainnya. Selain itu,

Facebook juga menyediakan fitur grup dan halaman yang memudahkan pengguna untuk bergabung atau mengikuti komunitas tertentu, serta menjalankan aktivitas bisnis dan promosi secara bebas. Namun, kebebasan ini sering kali disalahgunakan oleh sejumlah pengguna untuk menyampaikan ujaran kebencian (hate speech).

Ujaran kebencian di Facebook biasanya muncul dalam bentuk komentar yang menghina, merendahkan, mengandung ancaman, atau menyerang kelompok tertentu berdasarkan ras, agama, gender, orientasi seksual, atau latar belakang sosial. Konten seperti ini tidak hanya berpotensi menimbulkan konflik di dunia maya, tetapi juga bisa memicu kekerasan di dunia nyata. Sebagai platform yang memfasilitasi berbagai jenis konten dari pengguna, Facebook menyimpan beragam informasi yang dapat dijadikan alat bukti dalam kasus ujaran kebencian. Namun, dalam prakteknya, penggunaan alat bukti dari Facebook dalam ranah hukum seringkali menimbulkan perdebatan terkait keabsahan, privasi, dan kredibilitasnya (SaThierbach et al., 2015).

2.7. Google Colab

Google Colaboratory (Colab) adalah platform berbasis cloud yang banyak dimanfaatkan untuk menulis dan menjalankan kode Python secara daring. Karena Python cukup mudah dipahami, dan Colab memungkinkan pengguna bekerja sama serta mengelola kode secara *online*, kombinasi keduanya sering menjadi pilihan utama bagi para pendidik dalam memahami serta mengajarkan materi statistika. Dalam pelajaran matematika sendiri, statistika terbagi menjadi dua jenis, yaitu statistika deskriptif dan statistika inferensial (Handika, 2024).

Keunggulan utama dari Google Colab adalah kemudahannya dalam digunakan. Pengguna cukup memiliki akun Google dan koneksi internet untuk bisa mengaksesnya. File yang dikerjakan akan tersimpan otomatis di Google Drive, sehingga aman dan bisa diakses kapan saja. Selain itu, Colab juga menyediakan fasilitas untuk menjalankan program dengan bantuan GPU dan TPU secara gratis, yang biasanya dibutuhkan untuk pemrosesan data skala besar atau model pembelajaran mesin. Google Colab juga memungkinkan kerja kelompok karena beberapa orang bisa mengedit file notebook secara bersamaan. Hal ini sangat berguna dalam kerja tim atau tugas kolaboratif. Tidak hanya itu, Colab sudah mendukung banyak pustaka Python populer yang sering dipakai dalam analisis data, seperti Pandas, Numpy, Matplotlib, hingga Scikit-learn.

Gambar 2. 2 Google Colab

2.8. Penelitian Terdahulu

Berikut adalah tabel penelitian terdahulu yang mendukung kerangka teoritis pada penelitian ini :

Tabel 2. 1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Hasil Penelitian
1	(Mandasari et al., 2022)	Analisis Sentimen Pengguna Transportasi Online Terhadap	Hasil penelitian menunjukkan bahwa

		Layanan Grab Indonesia Menggunakan Multinomial Naive Bayes Classifier	Algoritma Multinomial Naïve Bayes dapat mengklasifikasikan tweet yang terkait dengan layanan Grab Indonesia dengan akurasi 86,57%. Dari tweet yang dikategorikan layanan GrabFood memiliki proporsi tertinggi sebesar 42,29%, sebagai sentimen netral, diikuti oleh GrabBike sebesar 32,57% dan GrabCar sebesar 25,14%.
2	(Ula & Fachrurrazi, 2023)	Analisis Sentimen Cyberbullying pada Media Sosial Twitter menggunakan Metode Support Vector Machine dan Naïve Bayes Classifier	Data diperoleh melalui API Twitter dengan mengumpulkan sekitar 100 tweet untuk setiap 10 kata kunci yang berpotensi menimbulkan

			cyberbullying. Setelah melalui proses prapemrosesan teks, data dianalisis menggunakan dua metode klasifikasi: Support Vector Machine (SVM) dan Naïve Bayes Classifier. Metode SVM mencapai akurasi sebesar 72% dan Metode Naïve Bayes Classifier mencapai akurasi sebesar 69%.
3	(Pratama et al., 2024)	Analisis Sentimen Aplikasi Indriver Pada Ulasan Google Playstore Dengan Metode Naive Bayes	Penelitian ini menunjukkan bahwa analisis sentimen Naive Bayes dapat digunakan untuk mengukur tingkat kepuasan pengguna terhadap aplikasi transportasi online seperti Indrive. Dengan hasil akurasi

			80%, model ini cukup andal untuk mengklasifikasikan ulasan sebagai positif atau negatif. Selain itu, penelitian ini mengeksplorasi aspek aplikasi Indrive yang paling dihargai dan dikeluhkan pengguna.
4	(Yuniar & Kismiantini, 2023)	Analisis Sentimen Ulasan pada Gojek Menggunakan Metode Naive Bayes	Hasilnya menunjukkan Mayoritas ulasan pengguna Gojek di Google Play Store bersifat positif (67,5%). Kata-kata yang dominan dalam ulasan positif: "baik", "driver", "bantu", "sejahtera ". Sementara Kata-kata yang dominan dalam ulasan negatif : "mahal", "eror", "Gopay", "lama",

			"kecewa".

Meskipun telah banyak penelitian yang membahas analisis sentimen dalam layanan transportasi *online*, tetapi sebagian besar penelitian lebih berfokus pada platform lain seperti Twitter dan YouTube, lebih fokus pada kepuasan pelanggan daripada dampak *negative* terhadap *driver*. Sebagian besar penelitian, seperti (Mandasari et al., 2022) dan (Pratama et al., 2024), menggunakan algoritma Naive Bayes untuk menganalisis sentimen terhadap layanan Grab dan Indriver, tetapi tidak berhasil menentukan *cyberbullying* yang dialami oleh driver. Selain itu, penelitian oleh (Ula & Fachrurrazi, 2023) membandingkan metode Naïve Bayes dan Support Vector Machine (SVM) dalam mendeteksi *cyberbullying* di Twitter, tetapi belum ada penelitian yang secara khusus menganalisis *cyberbullying* terhadap driver Gojek di Facebook. Penelitian ini mengisi celah tersebut dengan menganalisis intensitas sentimen *negative* dan ujaran kebencian terhadap *driver* Gojek menggunakan algoritma Naïve Bayes di platform Facebook, serta mengevaluasi dampak sosial dan profesional dari ujaran kebencian terhadap para mitra *driver*. Penelitian yang dilakukan oleh (Yuniar & Kismiantini, 2023) mengkategorikan ulasan pengguna tentang layanan Gojek menggunakan metode Naïve Bayes, tetapi tidak meneliti secara menyeluruh apakah komentar *negative* yang muncul memiliki unsur ujaran kebencian. Penelitian sebelumnya hanya membagi sentimen menjadi kategori umum positif, *negative*, dan netral, tanpa mempelajari pola dan jenis ujaran kebencian.

Oleh karena itu, penelitian ini berusaha menganalisis sentimen komentar pengguna Facebook tentang *driver* Gojek, serta mengidentifikasi intensitas dan pola ujaran kebencian yang terjadi. Penelitian ini akan mengklasifikasikan

komentar ke dalam kategori positif dan *negative*, serta meneliti sejauh mana komentar *negative* dapat dikategorikan sebagai bentuk ujaraan kebencian. Selain itu, penelitian ini bertujuan untuk mengeksplorasi bagaimana ujaran kebenciaan dapat memengaruhi kehidupan profesional dan psikologis *driver* Gojek dan bagaimana temuan ini dapat digunakan untuk membuat kebijakan yang lebih baik untuk melindungi mitra driver dari efek *negative* media sosial.

BAB III

METODOLOGI PENELITIAN

3.1. Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan eksploratif yang bertujuan untuk menganalisis sentimen *negative* pada komentar pengguna Facebook terhadap driver Gojek serta mengidentifikasi unsur ujaran kebencian dalam komentar tersebut. Penelitian kuantitatif dipilih karena memungkinkan pengolahan data dalam jumlah besar dan penyajian hasil dalam bentuk statistik yang lebih objektif. Penelitian ini menggunakan metode analisis sentimen berbasis *machine learning* menggunakan algoritma Naïve Bayes. Metode ini dipilih karena memiliki kemampuan untuk mengidentifikasi pola dalam data komentar yang dikumpulkan. Selain itu, penelitian ini bersifat deskriptif karena berusaha menggambarkan fenomena ujaran kebencian yang terjadi dalam komunitas pengguna Gojek di Facebook. Penelitian ini melibatkan proses *text mining*, yaitu teknik yang digunakan untuk mengekstrak informasi dari teks dalam jumlah besar untuk membersihkan, memproses, dan menganalisis data teks yang diperoleh dari komentar Facebook.

Penelitian ini secara khusus membahas pola-pola ujaran kebencian bermuatan *negative* yang muncul dalam komentar pengguna Facebook terhadap *driver* Gojek. Fokus analisis diarahkan pada kata-kata, frasa, dan konteks kalimat yang digunakan oleh pengguna dalam memberikan komentar, serta bagaimana intensitas ujaran tersebut dapat dikategorikan berdasarkan tingkat keparahan sentimen *negative*. Proses ini memiliki beberapa tahap yaitu *case folding*, *cleaning* data, tokenisasi,

normalisasi, *stopword removal*, dan *stemming*. Serta teknik untuk mengubah teks menjadi representasi numerik yang dapat di proses oleh model *machine learning* menggunakan metode *Vectorizer*. Selanjutnya, komentar yang telah diproses akan diklasifikasikan menggunakan algoritma Naïve Bayes untuk menentukan kategori sentimennya.

3.2. Data dan Sumber Data

Data penelitian berasal dari komentar pengguna Facebook tentang layanan driver Gojek. Data ini hanya berfokus pada komentar *negative* yang mengandung kritik, keluhan, atau ujaran kebencian yang berpotensi mengandung unsur ujaran kebencian. Selanjutnya, data tersebut dianalisis lebih lanjut untuk mengetahui pola dan intensitasnya. Jenis data yang digunakan dalam penelitian ini meliputi :

1. **Data Teks**

   Komentar *negative* dari pengguna facebook yang ditujukan kepada *driver* Gojek.

2. **Metadata**

   Informasi tambahan seperti nama pengguna, minggu komentar, dan tanggapan pengguna lain yang relevan.

3. **Data Historis**

   Riwayat komentar *negative* yang dapat menunjukkan tren dan pola sentiment *negative* terhadap *driver* Gojek dalam periode tertentu.

Sumber data utama dalam penelitian ini berasal dari platform media sosial Facebook, terutama dari:

1. **Grup dan Halaman Publik Facebook :** Grup diskusi atau halaman resmi dan tidak resmi yang membahas layanan Gojek.
2. **Kolom Komentar di Postingan Terkait:** Komentar pada postingan yang membahas pengalaman pengguna terhadap layanan Gojek.

Dalam proses pengumpulan data, digunakan kata kunci "Driver Gojek" untuk mencari dan menyeleksi komentar yang relevan. Data tersebut berfungsi sebagai dasar untuk mengembangkan analisis sentimen yang lebih akurat. Data dikumpulkan melalui teknik web *scraping* dengan menggunakan APIFY untuk mengekstrak komentar dari halaman atau grup Facebook yang relevan dengan penelitian ini.

3.3. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

1. **Studi Kasus**

Pada tahap pengumpulan data, penulis mengumpulkan berbagai teori yang terkait dengan skripsi sebagai bahan untuk menyempurnakan penelitian ini. Teori-teori ini berasal dari berbagai referensi yang memberikan latar belakang konsep yang diperlukan, serta temuan penelitian terdahulu seperti jurnal dan tesis yang berkaitan dengan topik penelitian, serta artikel yang membahas beberapa aspek penelitian. Selain itu, penulis mengunjungi beberapa situs web yang menyediakan informasi dan metode penerapan menggunakan text mining, klasifikasi teks, dan

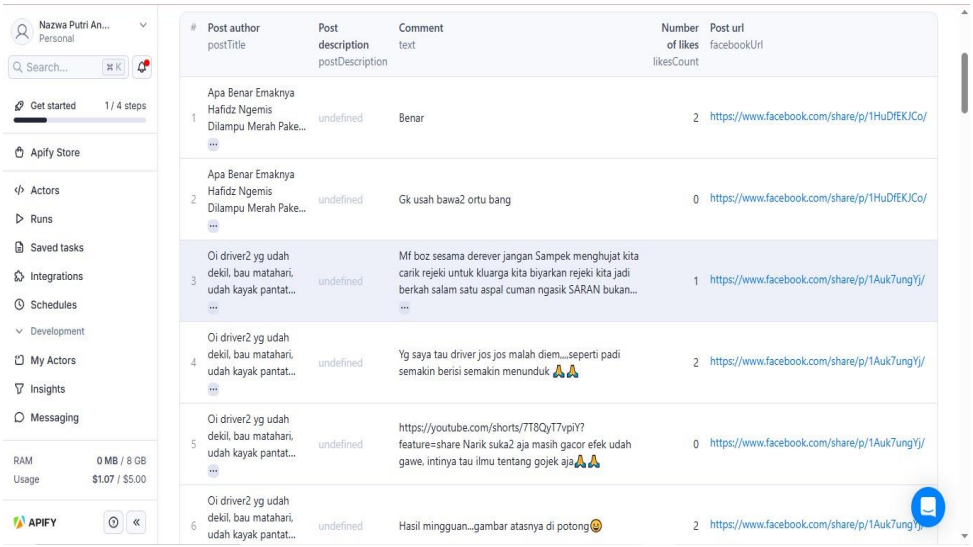
algoritma naïve bayes. Proses pengumpulan data ini tidak hanya membantu mengidentifikasi kerangka teori yang kuat, tetapi juga memberikan wawasan praktis tentang bagaimana menggunakan metode text mining dan klasifikasi algoritma naïve bayes untuk menganalisis ujaran kebencian terhadap *driver* Gojek.

2. Menggunakan APIFY

Dalam penelitian ini, proses pengumpulan data dilakukan menggunakan APIFY, yaitu sebuah platform otomatisasi berbasis *cloud* yang digunakan untuk web *scraping*, *crawling*, dan otomatisasi proses di web. Layanan ini memungkinkan pengguna untuk mengekstrak data dari berbagai situs web secara otomatis dan terstruktur. APIFY menyediakan lingkungan kerja berbasis JavaScript (Node.js) dan mendukung penggunaan pustaka populer seperti Puppeteer dan Cheerio. Salah satu keunggulan APIFY adalah kemampuannya dalam menjalankan agen (disebut actor) yang dapat bekerja di latar belakang dan dijadwalkan secara rutin, serta hasil ekstraksi datanya bisa langsung diekspor dalam format seperti JSON, CSV, atau Excel. Layanan ini banyak dimanfaatkan oleh peneliti, data analyst, dan pengembang yang membutuhkan data dari situs web yang tidak menyediakan API resmi, seperti platform media sosial, *e-commerce*, dan forum daring (Belghaouti et al., 2020).

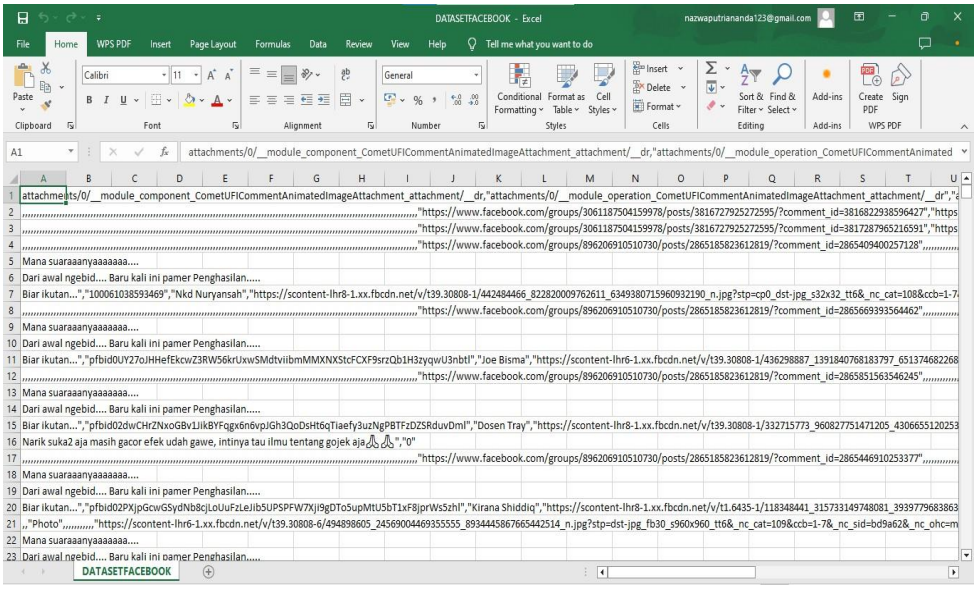
APIFY dimanfaatkan untuk mengumpulkan komentar publik dari halaman Facebook yang relevan dengan topik penelitian. Pemilihan APIFY didasarkan pada kemampuannya dalam mengekstrak data secara otomatis dan efisien. Dengan bantuan alat ini, komentar yang terdapat pada unggahan tertentu dapat dikumpulkan dalam jumlah besar tanpa harus dilakukan secara manual. Hal ini sangat membantu dalam proses pengumpulan data karena menghemat waktu dan tenaga, serta

memungkinkan peneliti untuk memperoleh data yang lebih lengkap dan *representative*.



Gambar 3. 1 Hasil scraping komentar facebook menggunakan APIFY

Hasil *scraping* yang ditampilkan dalam bentuk tabel terdiri dari beberapa kolom, yaitu kolom *post author* (nama pengguna), *comment* yang memuat isi komentar dari pengguna, *number of likes* yang mencatat jumlah tanda suka pada komentar tersebut, serta post URL yang merupakan tautan menuju postingan sumber di Facebook. Melalui struktur ini, komentar yang bersifat ujaran kebencian terhadap driver Gojek dapat diidentifikasi dan dianalisis lebih lanjut. Data hasil scraping kemudian diekspor ke dalam format CSV dan digunakan sebagai data mentah dalam proses *preprocessing* sebelum dianalisis dengan algoritma Naïve Bayes.



Gambar 3. 2 hasil scraping yang diekspor ke dalam format CSV

3. Penyaringan Data (*Filtering* Data)

Setelah data dikumpulkan melalui APIFY, dilakukan proses penyaringan untuk memastikan hanya komentar yang relevan dengan penelitian ini yang dimasukkan dalam dataset. Penyaringan ini melibatkan penghapusan komentar yang tidak terkait dengan layanan driver Gojek, komentar spam, serta duplikasi komentar. Data yang masih memiliki noise atau ketidak sesuaian dengan topik penelitian akan dieliminasi untuk menjaga validitas hasil analisis.

4. Validasi Data

Untuk memastikan akurasi dan kualitas dataset yang telah dikumpulkan, dilakukan validasi data dengan cara :

1) Mengecek keakuratan label sentimen dengan metode manual dan bantuan model prediksi awal.
2) Menggunakan uji coba pada sebagian dataset untuk melihat apakah klasifikasi sentimen telah berjalan dengan baik.

3) Melakukan penyesuaian jika terdapat ketidaksesuaian dalam pemberian label atau distribusi data yang tidak seimbang.

Dengan menggunakan APIFY sebagai teknik pengumpulan data, penelitian ini dapat mengakses data komentar secara efektif dan akurat sesuai dengan kebijakan privasi Facebook.

3.4. Metode Analisis Data

Analisis data dalam penelitian ini meliputi tahapan berikut :

1. ***Preprocessing*** **Data**

Membersihkan data dari karakter khusus, URL, dan kata-kata yang tidak relevan. Data yang telah dikumpulkan kemudian melewati tahapan preprocessing untuk meningkatkan kualitas analisis. Beberapa langkah preprocessing yang dilakukan antara lain :

1) **Case Folding :** Mengubah semua huruf dalam teks menjadi huruf kecil (*lowercase*) untuk memastikan konsistensi dalam analisis dan menghindari duplikasi kata yang sama namun berbeda dalam penulisan huruf besar dan kecil.
2) **Cleaning Data :** Proses pembersihan teks dari unsur-unsur yang tidak diperlukan dalam analisis. Proses pembersihan umumnya mencakup penghapusan tanda baca, angka, emotikon, simbol khusus, tautan, dan elemen lainnya.
3) **Tokenisasi :** Memecah teks menjadi bagian kata yang lebih kecil.
4) **Normalisasi :** Mengubah kata tidak baku atau slang menjadi kata baku yang lebih mudah dianalisis (KIKOY, 2023).

5) **Stopword Removal :** Menghapus kata-kata umum yang tidak memiliki makna signifikan dalam analisis sentimen, seperti 'dan', 'atau', 'saya' menggunakan library sastrawi.
6) **Stemming :** Mengubah kata-kata menjadi bentuk dasar untuk mengurangi variasi kata yang memiliki makna sama.

2. Labelling

Data yang telah melalui *preprocessing* selanjutnya dikategorikan menjadi komentar positif dan *negative*. *Labeling* dilakukan menggunakan pendekatan berbasis *Lexicon-Based* . Proses labeling ini bertujuan untuk memisahkan komentar positif dengan *negative* sehingga kata-kata positif diberi skor +1 dan kata-kata negative diberi skor -1. Seluruh kata dalam satu komentar kemudian dijumlahkan skornya. Jika hasil akhirnya bernilai positif, maka komentar tersebut diberi label positif, sedangkan jika hasilnya *negative*, maka komentar diberi label *negative*.

Seluruh komentar yang telah diberi label ini kemudian siap untuk diproses pada tahap selanjutnya, yaitu ekstraksi fitur menggunakan CountVectorizer. Pada tahap ini, setiap komentar diubah ke dalam bentuk representasi numerik, yaitu berdasarkan frekuensi kemunculan kata dalam komentar. Berbeda dengan TF-IDF yang memperhitungkan bobot kata berdasarkan frekuensi dalam seluruh dokumen, CountVectorizer hanya menghitung berapa kali suatu kata muncul dalam dokumen (komentar). Representasi ini digunakan untuk membangun model klasifikasi sentimen agar komputer dapat mengenali pola dari komentar positif dan *negative* berdasarkan data yang telah diberi label sebelumnya. Dengan proses pelabelan ini, komentar yang semula berbentuk teks bebas kini telah memiliki struktur yang jelas

dan siap dianalisis secara kuantitatif menggunakan metode klasifikasi yang telah ditentukan.

3. Spilt Data dengan CountVectorizer

Split data adalah langkah membagi data menjadi dua bagian, yaitu data latih dan data uji. Data latih digunakan untuk membangun model, sementara data uji digunakan untuk menguji apakah model yang dibuat bisa bekerja dengan baik pada data yang belum pernah dikenali sebelumnya. Tujuan dari pembagian ini adalah supaya proses pengujian model menjadi lebih jujur dan tidak hanya berdasarkan data yang sudah dikenal.

CountVectorizer digunakan untuk mengubah data teks menjadi data numerik dengan cara menghitung frekuensi kemunculan setiap kata dalam komentar. Setelah dilakukan pembagian data menjadi dua bagian, yaitu data latih dan data uji, dengan perbandingan 80 persen untuk data latih dan 20 persen untuk data uji. Setelah itu, CountVectorizer diterapkan pada data latih menggunakan fit_transform untuk menghasilkan vektor kata yang akan digunakan dalam pelatihan. Kemudian, data uji juga diubah menggunakan transform agar mengikuti struktur kata yang sama dengan data latih. Hasil dari proses ini berupa data angka yang bisa dibaca dan diproses oleh algoritma klasifikasi seperti Naive Bayes. Dengan begitu, komentar yang semula berupa teks sudah siap digunakan dalam proses pelatihan dan pengujian model.

Cara kerja CountVectorizer cukup sederhana, yaitu dengan menghitung jumlah kemunculan setiap kata dalam sebuah komentar. Setiap kata yang ada dalam kumpulan semua komentar akan dianggap sebagai fitur atau kolom dalam sebuah matriks. Kemudian, untuk setiap komentar, CountVectorizer mencatat berapa kali

masing-masing kata muncul. Misalnya, jika ada tiga komentar dan kata "*driver*" muncul dua kali di komentar pertama, satu kali di komentar kedua, dan tidak muncul sama sekali di komentar ketiga, maka nilai untuk kata "*driver*" dalam matriks tersebut akan tercatat sebagai 2, 1, dan 0. Setiap baris pada matriks mewakili satu komentar, sedangkan setiap kolom mewakili satu kata unik yang muncul di seluruh data. Jika sebuah kata tidak ada di dalam komentar tertentu, maka nilainya adalah 0. Sebaliknya, jika kata tersebut muncul, maka nilainya sesuai dengan jumlah kemunculannya. Berikut adalah rumus dasar CountVectorizer :

$$\mathrm{CV}(w, d) = \text{Jumlah kata w dalam dokumen d} \quad \text{.....(3.1)}$$

Dimana :

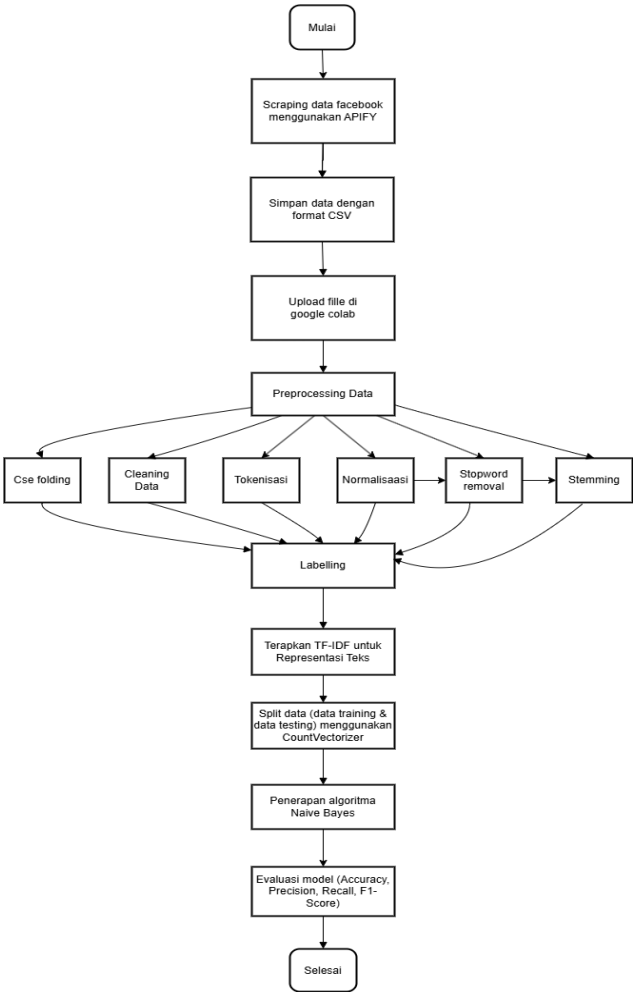
1) CV(w,d) = nilai CountVectorizer untuk kata *w* dalam dokumen *d*
2) w = kata (fitur) yang ada di dalam seluruh kumpulan komentar
3) d = dokumen atau komentar ke-n
4) Nilai akhir dari setiap elemen adalah jumlah berapa kali kata w muncul dalam dokumen d

4. Visualisasi Data

Visualisasi data merupakan cara menyajikan data dalam bentuk gambar, supaya lebih mudah dilihat dan dipahami. Daripada membaca data satu per satu dalam bentuk tabel atau angka, data ditampilkan dalam bentuk grafik, seperti diagram batang, diagram lingkaran, atau *word cloud*. Tujuannya supaya pembaca bisa langsung melihat pola atau kecenderungan dari data tersebut. Dalam analisis sentimen, visualisasi biasanya digunakan untuk menunjukkan jumlah komentar yang bersentimen positif dan *negative*.

5. Naïve Bayes

Sebelum menerapkan algoritma Naïve Bayes untuk melakukan klasifikasi, data komentar terlebih dahulu harus diubah dari bentuk teks menjadi bentuk numerik. Proses ini dilakukan dengan bantuan teknik ekstraksi fitur, salah satunya menggunakan CountVectorizer. CountVectorizer berfungsi untuk menghitung frekuensi kemunculan setiap kata dalam kumpulan komentar, lalu mengubahnya menjadi representasi vektor angka. Hasil dari proses ini berupa matriks yang merepresentasikan dokumen (komentar) dalam bentuk barisan angka sesuai dengan jumlah kata yang muncul. Matriks inilah yang kemudian digunakan sebagai input bagi algoritma Naïve Bayes untuk membangun model klasifikasi.



Gambar 3. 3 Alur Penelitian

Pada penelitian ini, model yang digunakan adalah Multinomial Naïve Bayes karena jenis data yang dianalisis merupakan representasi jumlah kemunculan kata dari komentar pengguna, yang telah diubah menjadi bentuk vektor menggunakan CountVectorizer. Model MultinomialNB sangat cocok untuk menangani data dalam bentuk frekuensi atau jumlah kata, karena model ini menghitung peluang setiap kelas berdasarkan seberapa sering suatu kata muncul dalam kelas tersebut. Oleh karena itu, MultinomialNB menjadi pilihan yang tepat untuk menyelesaikan permasalahan klasifikasi teks pada data komentar di *platform* Facebook.

3.5. Evaluasi Model

Evaluasi model merupakan tahapan penting dalam proses klasifikasi untuk mengetahui sejauh mana kinerja algoritma dalam memprediksi data dengan benar. Dalam penelitian ini, evaluasi dilakukan menggunakan empat metrik utama, yaitu akurasi, presisi, *recall*, dan f1-*score*.

1. **Akurasi**

Akurasi menunjukkan seberapa banyak data yang diklasifikasikan dengan benar oleh model dibandingkan dengan seluruh data yang diuji. Rumusnya adalah :

$$\text{Akurasi} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \quad (3.2)$$

2. **Presisi**

Presisi pada kelas *negative* menunjukkan ketepatan model dalam memprediksi komentar sebagai *negative* artinya, dari seluruh komentar yang diprediksi *negative*, seberapa banyak yang benar-benar *negative*. Rumusnya adalah :

$$\text{Presisi} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (3.3)$$

3. **Recall**

Recall pada kelas *negative* menunjukkan kemampuan model dalam menemukan seluruh komentar *negative* yang sebenarnya ada di dalam data. Rumusnya adalah :

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (3.4)$$

4. **F1-Score**

F1-*score* digunakan untuk menyeimbangkan antara nilai presisi dan *recall*, terutama ketika model mungkin baik dalam satu sisi tetapi kurang pada sisi lainnya. Rumusnya adalah :

$$\text{F1} - \text{Score} = 2\text{x}\ \frac{\text{Presisi x Recall}}{\text{Presisi} + \text{Recall}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.5)$$

Dimana :

1) TP (*True Positif*) : jumlah data yang benar-benar positif dan berhasil diprediksi sebagai positif oleh model.
2) TN (*True Negative*) : data *negative* yang berhasil diprediksi *negative*.
3) FP (*False Positive*) : data *negative* yang salah diprediksi sebagai positif.
4) FN (*False Negative*) : data positif yang salah diprediksi sebagai *negative*.

Dalam penelitian ini, evaluasi dilakukan untuk mengukur kemampuan algoritma Naïve Bayes dalam mengklasifikasikan komentar pengguna di Facebook menjadi dua kategori, yaitu positif dan *negative*, dengan fokus utama pada komentar *negative* yang mengandung ujaran kebencian terhadap *driver* Gojek. Melalui evaluasi ini, peneliti dapat menilai apakah model yang digunakan sudah mampu mengenali komentar *negative* secara konsisten dan akurat.

3.6. Waktu dan Tempat Penelitian

1. Waktu Penelitian

Tabel 3. 1 Waktu Penelitian

No	Tahapan	Desember 2024	Januari 2025	Februari 2025	Maret 2025	April 2025	Mei 2025	Juni 2025	Juli 2025	Agustus 2025
1	Pengajuan judul									
2	Pengumpulan data									
3	Penyusunan prosposal									
4	Seminar proposal									
5	Analisi dan data penelitian									
6	Penyusunan skripsi									
7	Sidang meja hijau									
8	Penyempurnaan skripsi dan									

	penulisan artikel									

2. Tempat Penelitian

Penelitian ini dilakukan secara daring dengan menggunakan perangkat laptop yang memiliki spesifikasi memadai untuk menjalankan proses ekstraksi data, *preprocessing*, analisis sentimen, dan klasifikasi menggunakan metode Naïve Bayes. Data dikumpulkan secara *online* melalui Facebook Graph API, yang memungkinkan peneliti mengakses komentar pada postingan dan grup publik yang relevan dengan penelitian ini. Seluruh proses pengolahan data, mulai dari ekstraksi hingga analisis sentimen, dilakukan di laptop dengan spesifikasi sebagai berikut :

Tabel 3. 2 Perangkat Keras

No	Nama Perangkat	Deskripsi
1	Laptop	LENOVO
2	Processor	AMD Ryzen 3
3	RAM	8GB RAM
4	Penyimpanan	512 GB SSD

Tabel 3. 3 Perangkat Lunak

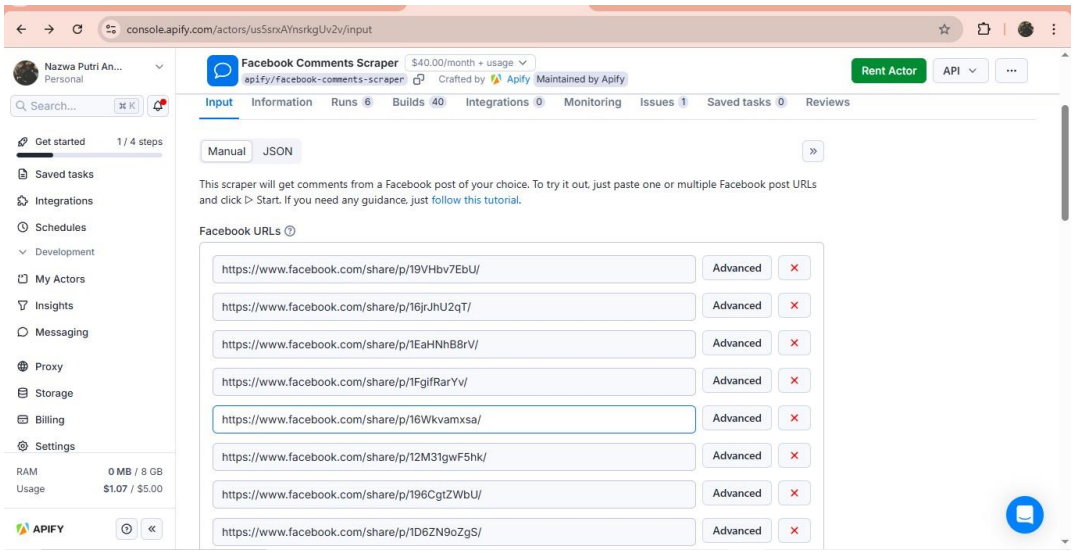
No	Nama	Deskripsi
1	Windows 10 64-bit	*Operating System*
2	*Python* 3.10.0	*Tools* untuk membangun dan melatih (training) model
3	*Scikit-learn*	Pustaka *Python* yang menyediakan algoritma dan fungsi untuk analisis data, termasuk pengelompokkan data

4	*Matplotlib* dan *Seaborn*	Pustaka *Python* untuk visualisasi data
5	Microsoft Office 2010	*Tools* untuk membuat hasil laporan Penelitian

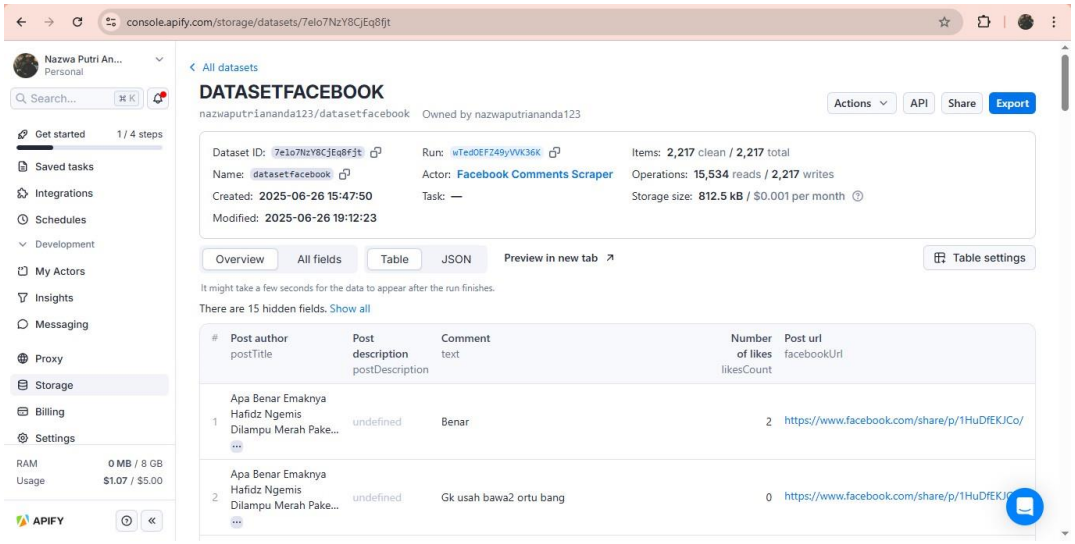
BAB IV
HASIL DAN PEMBAHASAN

4.1. Hasil Web Scraping

Pengumpulan data dilakukan dengan metode *scraping* menggunakan platform APIFY. Dalam prosesnya, APIFY mengakses link-link postingan yang relevan terhadap *driver* Gojek. Setelah link-link yang relevan ditentukan, APIFY secara otomatis mengambil komentar dari setiap postingan tersebut dan menyusunnya ke dalam format terstruktur. Data yang diambil meliputi nama pengguna, isi postingan, dan komentar pengguna.

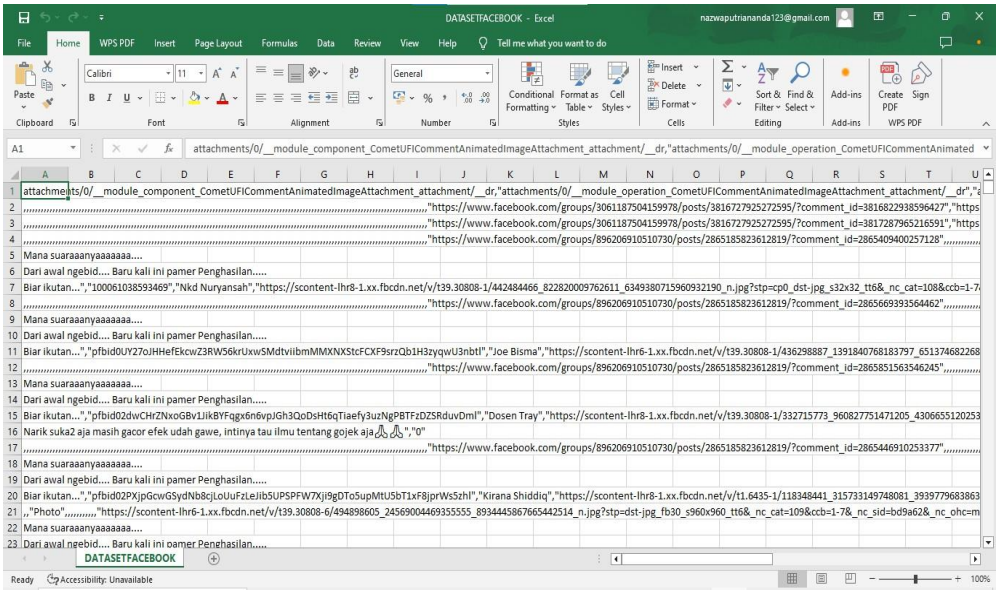


Gambar 4. 1 Mengakses link-link postingan yang relevan



Gambar 4. 2 Hasil scraping terdapat 2217 data

Data hasil scraping tersebut kemudian disimpan dalam format CSV (*Comma Separated Values*). Pemilihan format CSV dilakukan karena format ini bersifat sederhana dan ringan untuk perangkat lunak pengolahan data seperti Microsoft Excel, Google Sheets, Google Colab, maupun bahasa pemrograman Python. Setiap baris dalam file CSV merepresentasikan satu data komentar, sedangkan setiap kolom berisi atribut yang diambil dari hasil *scraping*, yaitu nama pengguna (*post author*), isi postingan (*post content*), dan komentar pengguna (*comment*). Dengan menggunakan format CSV, proses pengolahan data selanjutnya seperti pembersihan data, pelabelan sentimen, dan ekstraksi fitur dapat dilakukan dengan lebih mudah.



Gambar 4. 3 Format file CSV

Selanjutnya file CSV tersebut di upload kedalam Goole Colab, dengan perintah from google.colab import files yang berfungsi untuk memanggil fungsi upload bawaan Colab. Setelah itu, files.upload() dijalankan agar pengguna dapat memilih file. Setelah file berhasil diunggah, file CSV tersebut dibaca menggunakan pd.read_csv('DATASETFACEBOOK.csv'), lalu disimpan ke dalam variabel df. Kemudian, hanya beberapa kolom yang dibutuhkan saja yang dipilih, yaitu 'profileName', 'postTitle', dan 'text'. Ketiga kolom ini kemudian diberi nama baru yang lebih mudah dibaca, yaitu 'Nama Pengguna', 'Postingan', dan 'Komentar'. Terakhir, df.head(10) digunakan untuk menampilkan sepuluh data pertama dari DataFrame sebagai langkah verifikasi bahwa data sudah berhasil dimuat dan diolah dengan benar.

```
from google.colab import files
import pandas as pd

uploaded = files.upload()

# Lalu baca filenya
df = pd.read_csv('DATASETFACEBOOK.csv')  # Nama file

# Pilih beberapa kolom
df = df[['profileName' ,'postTitle', 'text']]

# Ganti nama kolom
df = df.rename(columns={'profileName' : 'Nama Pengguna', 'postTitle' : 'Postingan',
'text' : 'Komentar'})

#Tampilkan hasil
df.head(10)
```

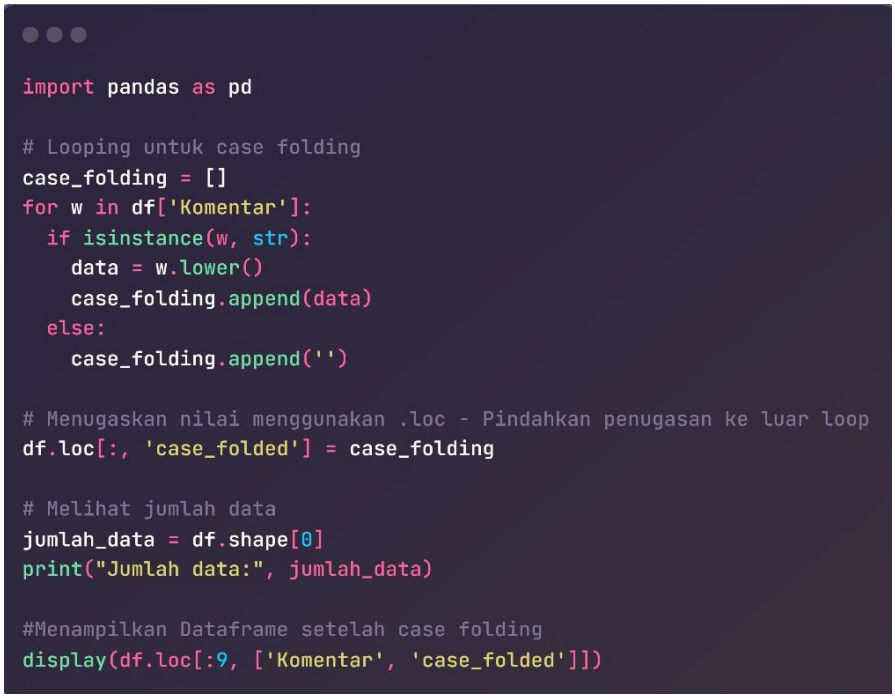
Gambar 4. 4 Upload File-Code

4.2. Preprocessing Data

Berdasarkan tampilan file CSV, terlihat bahwa data yang diperoleh masih mengandung elemen-elemen yang tidak dibutuhkan seperti tautan gambar, baris kosong, ID metadata, serta karakter-karakter khusus yang tidak relevan dengan isi komentar. Oleh karena itu, dilakukan tahapan *preprocessing* data untuk membersihkan dan menormalkan isi komentar sebelum dianalisis lebih lanjut. *Preprocessing* dimulai dengan menghapus baris kosong dan kolom yang tidak diperlukan, seperti kolom tautan gambar atau metadata teknis dari Facebook.

1. Case Folding

Case folding merupakan tahap awal dalam *preprocessing* data teks yang bertujuan untuk menyeragamkan bentuk huruf dengan cara mengubah seluruh huruf dalam komentar menjadi huruf kecil (*lowercase*).



```python
import pandas as pd

# Looping untuk case folding
case_folding = []
for w in df['Komentar']:
  if isinstance(w, str):
    data = w.lower()
    case_folding.append(data)
  else:
    case_folding.append('')

# Menugaskan nilai menggunakan .loc - Pindahkan penugasan ke luar loop
df.loc[:, 'case_folded'] = case_folding

# Melihat jumlah data
jumlah_data = df.shape[0]
print("Jumlah data:", jumlah_data)

#Menampilkan Dataframe setelah case folding
display(df.loc[:9, ['Komentar', 'case_folded']])
```

Gambar 4. 5 Case Folding-Code

Source code di atas ditulis didalam Google Colab menggunakan bahasa pemrograman Python dengan bantuan library pandas yang digunakan untuk mengelola data dalam bentuk tabel atau DataFrame. Pada awal kode, import pandas as pd digunakan untuk memanggil library pandas. Kemudian, proses case folding dilakukan dengan cara membuat list kosong bernama case_folding = []. Selanjutnya dilakukan looping untuk setiap elemen dalam kolom ‘Komentar’ pada DataFrame df. Pada setiap baris, kode if isinstance(w, str) digunakan untuk memastikan bahwa nilai tersebut adalah string. Maka nilai komentar tersebut diubah menjadi huruf kecil dengan fungsi lower() dan dimasukkan ke dalam list case_folding. Setelah seluruh komentar diproses, list case_folding yang telah berisi hasil huruf kecil disimpan ke dalam kolom baru pada DataFrame dengan nama 'case_folded'. Terakhir, hasil case folding ditampilkan menggunakan display() hanya untuk 9 baris pertama agar bisa dilihat perbandingan antara komentar asli dengan hasil yang sudah dikonversi ke huruf kecil.

Jumlah data: 2217

	Komentar	case_folded
0	Benar	benar
1	Gk usah bawa2 ortu bang	gk usah bawa2 ortu bang
2	Mf boz sesama derever jangan Sampek menghujat ...	mf boz sesama derever jangan sampek menghujat ...
3	Yg saya tau driver jos jos malah diem,,,,seper...	yg saya tau driver jos jos malah diem,,,,seper...
4	https://youtube.com/shorts/7T8QyT7vpiY?feature...	https://youtube.com/shorts/7t8qyt7vpiy?feature...
5	Hasil mingguan...gambar atasnya di potong😀	hasil mingguan...gambar atasnya di potong😀
6	Kesimpulan apa yg didapat dari postingan saya ...	kesimpulan apa yg didapat dari postingan saya ...
7	Sehat?	sehat?
8	Segol dong	segol dong
9	Sigitu aja udah pamer.	sigitu aja udah pamer.

Gambar 4. 6 Hasil case folding

2. Cleaning Data

Cleaning data merupakan salah satu tahap penting dalam proses *preprocessing* yang bertujuan untuk membersihkan data komentar dari elemen-elemen yang tidak relevan atau mengganggu proses analisis. Dalam penelitian ini, data komentar yang diperoleh dari hasil *scraping* Facebook masih mengandung berbagai komponen yang tidak diperlukan seperti tanda baca, angka, karakter khusus, emoji, *mention*, hashtag, serta tautan atau URL, dan lain-lain.

```python
import re
import pandas as pd

# Contoh daftar kata-kata tidak penting (banned words)
banned_words = [
    'wkwk', 'hehe', 'oh', 'kkkkkkkkk', 'cok', 'ngab', 'wkwkwkwk', 'wkwk', 'wkwkw',
'wkwkwkw', 'wkwkwk', 'hihi', 'hihihii', 'hihihi', 'hehehe', 'heheheh', 'huhu',
'anjay', 'yah', 'yaelah'
]

# Buat regex dari daftar banned
re_banned_words = re.compile(r"\b(" + "|".join(map(re.escape, banned_words)) +
r")\b", re.IGNORECASE)

# Fungsi cleaning lengkap
def clean_text(text):
    if not isinstance(text, str):
        return ""
    text = re.sub(r'@\w+', '', text)            # hapus mention
    text = re.sub(r'#\w+', '', text)            # hapus hashtag
    text = re.sub(r'[^\w\s]', '', text)         # hapus tanda baca
    text = re.sub(r'\d+', '', text)             # hapus angka
    text = text.lower()                         # ubah ke lowercase
    text = re_banned_words.sub('', text)        # hapus banned words
    text = re.sub(r'http\S+|www\S+', '', text)  # hapus link
    text = re.sub(r'\s+', ' ', text)            # hapus spasi berlebih
    return text.strip()                         # hapus spasi depan/belakang


# Terapkan fungsi cleaning ke kolom Komentar
df['Komentar_Cleaning'] = df['Komentar'].apply(clean_text)

# Tampilkan hasil
display(df[['Komentar', 'Komentar_Cleaning']].head(10))
```

Gambar 4. 7 Cleaning Data-Code

Dalam proses ini, *cleaning* data dilakukan dengan menggunakan bantuan pustaka pandas dan re (*regular expression*). Langkah pertama yaitu membuat daftar kata-kata yang dianggap tidak penting (*banned words*) seperti “wkwk”, “hehe”, “oh”, “yah”, dan lain-lain. Kata-kata ini sering muncul di komentar media sosial, namun tidak memiliki nilai dalam proses klasifikasi sentimen. Selanjutnya, daftar tersebut digabungkan menjadi pola regex agar dapat dihapus sekaligus dalam satu proses. Fungsi clean_text() dibuat untuk menjalankan proses cleaning secara menyeluruh, termasuk menghapus *mention*, hashtag, tanda baca, angka, tautan URL, serta mengganti seluruh huruf menjadi huruf kecil. Selain itu, fungsi ini juga menghapus spasi ganda dan membersihkan komentar dari kata-kata tidak penting berdasarkan daftar *banned* yang telah ditentukan. Fungsi tersebut kemudian

diterapkan ke kolom komentar dan hasilnya disimpan dalam kolom baru bernama Komentar_Cleaning.

	Komentar	Komentar_Cleaning
0	Benar	benar
1	Gk usah bawa2 ortu bang	gk usah bawa ortu bang
2	Mf boz sesama derever jangan Sampek menghujat ...	mf boz sesama derever jangan sampek menghujat ...
3	Yg saya tau driver jos jos malah diem,,,,seper...	yg saya tau driver jos jos malah diemseperti p...
4	https://youtube.com/shorts/7T8QyT7vpiY?feature...	narik suka aja masih gacor efek udah gawe inti...
5	Hasil mingguan...gambar atasnya di potong😀	hasil mingguangambar atasnya di potong
6	Kesimpulan apa yg didapat dari postingan saya ...	kesimpulan apa yg didapat dari postingan saya ...
7	Sehat?	sehat
8	Segol dong	segol dong
9	Sigitu aja udah pamer.	sigitu aja udah pamer

Gambar 4. 8 Hasil Cleaning Data

3. Tokenisasi

Tokenisasi merupakan salah satu tahap penting dalam proses *preprocessing* data teks yang bertujuan untuk memecah kalimat atau paragraf menjadi satuan-satuan kata yang lebih kecil, yang disebut dengan token. Dalam konteks penelitian ini, tokenisasi diterapkan pada komentar-komentar yang telah melalui proses *cleaning* agar setiap komentar dapat diuraikan menjadi deretan kata tunggal yang siap dianalisis. Proses ini diperlukan karena algoritma klasifikasi seperti Naïve Bayes tidak dapat bekerja langsung dengan teks utuh dalam bentuk kalimat, melainkan membutuhkan input berupa kumpulan kata (token) yang dapat dihitung frekuensinya. Sebagai contoh, kalimat “driver gojek sering dihina” akan diubah menjadi empat token: [“driver”, “gojek”, “sering”, “dihina”].

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

# Download the 'punkt_tab' resource
nltk.download('punkt_tab')

# Tokenisasi
df['Tokenized_Komentar'] = df['Komentar_Cleaning'].astype(str).apply(lambda x:
word_tokenize(x.lower()))

# Filter hasil kosong
df = df[df['Tokenized_Komentar'].apply(lambda x: isinstance(x, list) and len(x) >
0)]
df = df.reset_index(drop=True)

# Gabungkan token-token dalam satu string untuk tampilan yang rapi
df['Tokenized_Komentar_Join'] = df['Tokenized_Komentar'].apply(lambda x: ',
'.join(x))

# Hapus baris yang kolom 'Komentar'-nya kosong (NaN)
df = df.dropna(subset=['Komentar_Cleaning'])
df = df.reset_index(drop=True)

# Tampilkan kolom Komentar dan hasil Tokenized_Komentar_Join
from IPython.display import display
display(df[['Komentar_Cleaning', 'Tokenized_Komentar_Join']].head(10))
```

Gambar 4. 9 Tokenisasi-Code

Tahap tokenisasi dilakukan dengan bantuan pustaka Natural Language Toolkit (nltk) dalam Python untuk memecah komentar menjadi satuan kata atau token. Proses diawali dengan mengimpor fungsi word_tokenize dari modul nltk.tokenize serta mengunduh data pendukung berupa ‘punkt’ dan ‘punkt_tab’ yang dibutuhkan untuk mendeteksi batas kata. Tokenisasi diterapkan pada kolom komentar yang telah melalui tahap *cleaning* (Komentar_Cleaning). Dalam proses ini, setiap komentar diubah ke dalam bentuk string, kemudian diubah ke huruf kecil (lower()), dan selanjutnya diproses menggunakan word_tokenize() untuk dipecah menjadi daftar kata (token). Hasil tokenisasi disimpan ke dalam kolom baru bernama Tokenized_Komentar. Selanjutnya, dilakukan filter untuk menghapus data kosong atau yang token-nya tidak valid, serta dilakukan reset_index agar data tetap terstruktur dengan rapi. Untuk keperluan visualisasi yang lebih sederhana, token-token yang dihasilkan dari proses sebelumnya digabung kembali ke dalam

satu kalimat melalui fungsi join() dan disimpan di kolom Tokenized_Komentar_Join. Terakhir, ditampilkan perbandingan antara komentar hasil *cleaning* dan komentar hasil tokenisasi gabungan.

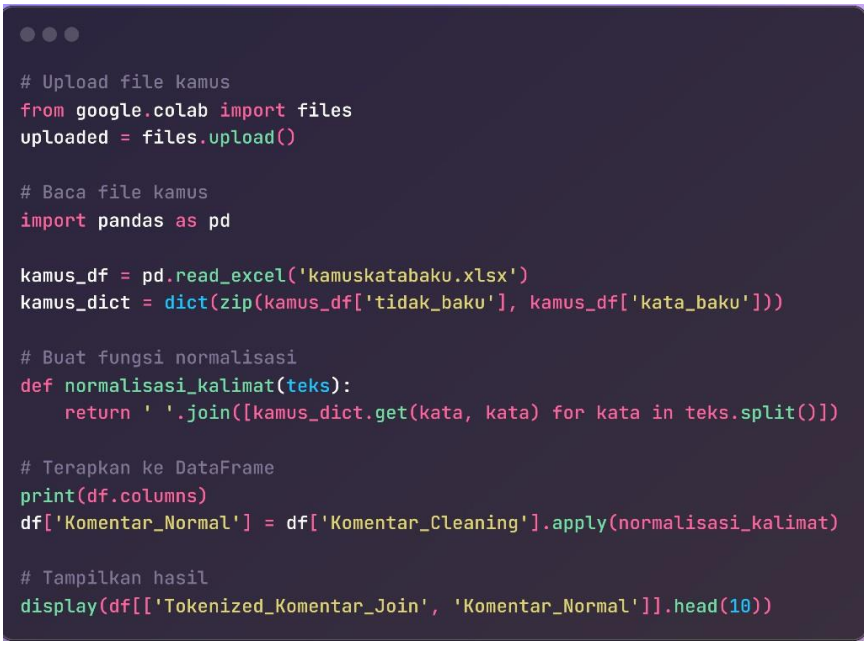
	Komentar_Cleaning	Tokenized_Komentar_Join
0	benar	benar
1	gk usah bawa ortu bang	gk, usah, bawa, ortu, bang
2	mf boz sesama derever jangan sampek menghujat ...	mf, boz, sesama, derever, jangan, sampek, meng...
3	yg saya tau driver jos jos malah diemseperti p...	yg, saya, tau, driver, jos, jos, malah, diemse...
4	narik suka aja masih gacor efek udah gawe inti...	narik, suka, aja, masih, gacor, efek, udah, ga...
5	hasil mingguangambar atasnya di potong	hasil, mingguangambar, atasnya, di, potong
6	kesimpulan apa yg didapat dari postingan saya ...	kesimpulan, apa, yg, didapat, dari, postingan,...
7	sehat	sehat
8	segol dong	segol, dong
9	sigitu aja udah pamer	sigitu, aja, udah, pamer

Gambar 4. 10 Hasil Tokenisasi

4. Normalisasi

Normalisasi merupakan salah satu tahap penting dalam *preprocessing* data teks yang bertujuan untuk menyamakan bentuk kata yang berbeda tetapi memiliki makna yang sama. Dalam konteks analisis komentar pada media sosial seperti Facebook, sering kali ditemukan berbagai variasi penulisan kata yang tidak baku, seperti kata “gk”, “nggak”, atau “ga” yang semuanya memiliki makna sama yaitu “tidak”. Jika tidak dinormalisasi, kata-kata tersebut akan dianggap berbeda oleh sistem, padahal maknanya sama. Oleh karena itu, dalam proses normalisasi ini dilakukan penggantian kata tidak baku atau kata gaul dengan bentuk kata baku yang sesuai Kamus Besar Bahasa Indonesia (KBBI) atau sesuai konteks penggunaannya. Proses normalisasi dilakukan dengan menggunakan kamus kata tidak baku yang

telah dibuat sebelumnya dalam bentuk *dictionary* di Python, lalu diterapkan pada setiap kata dalam komentar yang telah melalui tahap tokenisasi.



```
# Upload file kamus
from google.colab import files
uploaded = files.upload()

# Baca file kamus
import pandas as pd

kamus_df = pd.read_excel('kamuskatabaku.xlsx')
kamus_dict = dict(zip(kamus_df['tidak_baku'], kamus_df['kata_baku']))

# Buat fungsi normalisasi
def normalisasi_kalimat(teks):
    return ' '.join([kamus_dict.get(kata, kata) for kata in teks.split()])

# Terapkan ke DataFrame
print(df.columns)
df['Komentar_Normal'] = df['Komentar_Cleaning'].apply(normalisasi_kalimat)

# Tampilkan hasil
display(df[['Tokenized_Komentar_Join', 'Komentar_Normal']].head(10))
```

Gambar 4. 11 Normalisasi-Code

Proses ini diawali dengan mengunggah file kamus kata tidak baku dalam format Excel (‘kamuskatabaku.xlsx’) menggunakan fungsi files.upload() dari pustaka google.colab. Peneliti menggunakan sebuah kamus kata tidak baku yang diperoleh dari situs Kaggle, sebuah platform berbagi dataset yang banyak digunakan dalam penelitian data *science*. Kamus ini berisi daftar pasangan kata tidak baku dan kata baku yang umum digunakan dalam percakapan di media sosial, seperti “gk” yang dinormalisasi menjadi “tidak”, atau “aja” menjadi “saja”. File kamus tersebut diunduh dalam format Excel (kamuskatabaku.xlsx) dan kemudian diolah menggunakan library Python pandas untuk diubah ke dalam format dictionary (kamus_dict) dengan fungsi zip, sehingga setiap kata tidak baku dipasangkan dengan kata bakunya sebagai pasangan key-value. Setelah kamus siap, dibuatlah fungsi normalisasi_kalimat(teks) yang berfungsi untuk memeriksa setiap kata dalam kalimat, kemudian mencocokkannya dengan kamus. Jika ditemukan

kesesuaian kata tidak baku, maka kata tersebut diganti dengan bentuk bakunya. Jika tidak ada yang sesuai, maka kata dibiarkan tetap. Proses penggantian ini dilakukan dengan menggunakan list comprehension dan fungsi .get() dari dictionary. Fungsi normalisasi ini kemudian diterapkan ke setiap baris komentar pada kolom Komentar_Cleaning dan hasilnya disimpan dalam kolom baru bernama Komentar_Normal. Terakhir, ditampilkan perbandingan antara komentar hasil tokenisasi (Tokenized_Komentar_Join) dengan hasil normalisasi (Komentar_Normal) untuk menunjukkan bagaimana kata-kata tidak baku berhasil dikonversi ke bentuk baku.

	Tokenized_Komentar_Join	Komentar_Normal
0	benar	benar
1	gk, usah, bawa, ortu, bang	tidak usah bawa orang tua abang
2	mf, boz, sesama, derever, jangan, sampek, meng...	maaf bos sesama derever jangan sampai menghuja...
3	yg, saya, tau, driver, jos, jos, malah, diemse...	yang saya tau driver jos jos malah diemseperti...
4	narik, suka, aja, masih, gacor, efek, udah, ga...	menarik suka saja masih gacor efek sudah gawe ...
5	hasil, mingguangambar, atasnya, di, potong	hasil mingguangambar atasnya di potong
6	kesimpulan, apa, yg, didapat, dari, postingan,...	kesimpulan apa yang didapat dari postingan say...
7	sehat	sehat
8	segol, dong	segol dong
9	sigitu, aja, udah, pamer	sigitu saja sudah pamer

Gambar 4. 12 Hasil Normalisasi

5. Stopword Removal

Stopword removal merupakan salah satu tahap penting dalam *preprocessing* teks yang bertujuan untuk menghapus kata-kata yang dianggap tidak memiliki makna signifikan dalam proses analisis. Kata-kata tersebut disebut sebagai stopword, yaitu kata-kata umum yang sering muncul dalam kalimat namun tidak memberikan informasi khusus terhadap konteks atau makna keseluruhan, seperti “yang”, “di”, “dan”, “ke”, “untuk”, dan sebagainya. Dalam penelitian ini, stopword

removal dilakukan setelah data komentar melewati tahap tokenisasi dan normalisasi, sehingga kata-kata tidak penting dapat dihilangkan sebelum dilakukan ekstraksi fitur. Penghapusan stopword bertujuan untuk mengurangi *noise* dalam data dan memastikan bahwa hanya kata-kata yang mengandung makna penting saja yang dipertahankan. Proses ini biasanya dilakukan menggunakan daftar stopword dari pustaka NLP seperti Sastrawi atau NLTK, yang sudah menyediakan daftar kata umum dalam Bahasa Indonesia.

```
# Import Library
!pip install Sastrawi
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.StopWordRemover.StopWordRemover import StopWordRemover

stopwords_tambahan = ['yang', 'nya', 'yg', 'lu', 'gua', 'krn', 'dong', 'kan', 'sih',
'mah', 'iya', 'lah', 'hayo', 'ah', 'ya', 'apa', 'dalam', 'kali', 'mau', 'kok',
'make', 'tangerang', 'wah', 'wkwkwkwk', 'ton', 'itu', 'kau', 'nah', 'dan', 'di',
'ke', 'pada', 'adalah', 'gang', 'lo', 'rmhnya', 'yan', 'malhotra', 'wkk', 'totok',
'anjay',  'mending', 'situ', 'mungkin', 'sebagai']

# Buat factory dan ambil stopword list bawaan
factory = StopWordRemoverFactory()
stopword_list = factory.get_stop_words()

# Gabungkan dengan list tambahan
stopword_list += stopwords_tambahan
stopword_list = list(set(stopword_list))  # Buat jadi unik

# Fungsi remove stopword baru
import string

def remove_stopwords_custom(text):
    # Hapus tanda baca
    for punct in string.punctuation:
        text = text.replace(punct, "")
    return ' '.join([word for word in text.split() if word not in stopword_list])

# Apply the function to create the 'Komentar_Stopword' column
df['Komentar_Stopword'] = df['Komentar_Normal'].apply(remove_stopwords_custom)

display(df[['Komentar_Normal', 'Komentar_Stopword']].head(10))
```

Gambar 4. 13 Stopword Removal-Code

Pada *source code* ini, proses penghapusan stopword diawali dengan mengimpor library Sastrawi, yaitu pustaka pemrosesan bahasa alami Bahasa Indonesia yang menyediakan modul StopWordRemoverFactory untuk mengakses daftar *stopword* bawaan. Selain itu, peneliti juga menambahkan daftar *stopword*

tambahan secara manual ke dalam variabel stopwords_tambahan, yang terdiri dari kata-kata informal atau slang yang sering muncul di media sosial seperti “yang”, “gua”, “mah”, “dong”, “iya”, dan sebagainya. Daftar *stopword* dari Sastrawi kemudian digabungkan dengan daftar tambahan tersebut, lalu dikonversi ke dalam bentuk set untuk memastikan tidak ada duplikasi. Selanjutnya, dibuat fungsi remove_stopwords_custom(text) yang berfungsi untuk membersihkan tanda baca dari teks menggunakan modul string.punctuation, kemudian membuang kata-kata yang terdapat di dalam stopword_list. Proses penghapusan dilakukan dengan membandingkan setiap kata dalam teks dan hanya mempertahankan kata yang tidak termasuk dalam daftar *stopword*. Fungsi ini diterapkan pada kolom Komentar_Normal menggunakan fungsi .apply() dari pandas, dan hasilnya disimpan dalam kolom baru Komentar_Stopword.

	Komentar_Normal	Komentar_Stopword
0	benar	benar
1	tidak usah bawa orang tua abang	usah bawa orang tua abang
2	maaf bos sesama derever jangan sampai menghuja...	maaf bos sesama derever jangan menghujat carik...
3	yang saya tau driver jos jos malah diemseperti...	tau driver jos jos malah diemseperti padi sema...
4	menarik suka saja masih gacor efek sudah gawe ...	menarik suka gacor efek gawe intinya tau ilmu ...
5	hasil mingguangambar atasnya di potong	hasil mingguangambar atasnya potong
6	kesimpulan apa yang didapat dari postingan say...	kesimpulan didapat postingan commentan orang a...
7	sehat	sehat
8	segol dong	segol
9	sigitu saja sudah pamer	sigitu pamer

Gambar 4. 14 Hasil Stopword Removal

6. Stemming

Stemming merupakan proses mengubah kata turunan ke dalam bentuk dasarnya (*root word* atau kata dasar). Tujuannya adalah untuk menyederhanakan

variasi morfologi kata sehingga model analisis dapat memproses data secara lebih konsisten. Dalam konteks penelitian ini, stemming sangat penting karena komentar yang diambil dari media sosial seperti Facebook sering mengandung berbagai bentuk kata turunan, misalnya “lari”, “berlari”, dan “berlarian” yang semuanya berasal dari kata dasar “lari”. Tanpa proses *stemming*, sistem akan menganggap ketiga kata tersebut sebagai entitas yang berbeda, padahal memiliki makna yang sama. Oleh karena itu, dalam penelitian ini digunakan *library* Sastrawi, yaitu pustaka NLP Bahasa Indonesia yang menyediakan algoritma *stemming* berbasis kamus. Proses *stemming* dilakukan setelah tahapan *stopword removal*, agar hanya kata-kata penting saja yang diproses.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

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

# Fungsi stemming
def stemming_kalimat(teks):
    return stemmer.stem(teks)

# Terapkan ke kolom hasil normalisasi
df['Komentar_Stemmed'] = df['Komentar_Stopword'].apply(stemming_kalimat)

# Tampilkan hasil
display(df[['Komentar_Cleaning', 'Tokenized_Komentar_Join', 'Komentar_Normal',
'Komentar_Stopword', 'Komentar_Stemmed']].head(10))
```

Gambar 4. 15 Stemming-Code

Source code di atas menunjukkan penerapan proses *stemming* pada data komentar yang telah melalui tahap normalisasi dan *stopword removal*. Langkah pertama adalah mengimpor modul StemmerFactory dari pustaka Sastrawi, yang merupakan *library* pemrosesan bahasa alami khusus Bahasa Indonesia. Kemudian dibuat objek *stemmer* menggunakan perintah factory.create_stemmer(), yang nantinya berfungsi untuk mengubah kata turunan menjadi kata dasar. Selanjutnya, dibuat fungsi bernama stemming_kalimat(teks) yang menggunakan

method .stem(teks) untuk menerapkan proses *stemming* pada setiap baris teks. Fungsi tersebut kemudian diterapkan ke kolom Komentar_Stopword, yaitu kolom hasil penghapusan *stopword* sebelumnya, dengan menggunakan method .apply(). Hasil *stemming* disimpan ke dalam kolom baru bernama Komentar_Stemmed. Terakhir, ditampilkan lima kolom secara berdampingan: Komentar_Cleaning, Tokenized_Komentar_Join, Komentar_Normal, Komentar_Stopword, dan Komentar_Stemmed, untuk memperlihatkan transformasi data komentar dari tahap awal hingga menjadi kata-kata dasar.

	Komentar_Cleaning	Tokenized_Komentar_Join	Komentar_Normal	Komentar_Stopword	Komentar_Stemmed
0	benar	benar	benar	benar	benar
1	gk usah bawa ortu bang	gk, usah, bawa, ortu, bang	tidak usah bawa orang tua abang	usah bawa orang tua abang	usah bawa orang tua abang
2	mf boz sesama derever jangan sampek menghujat ...	mf, boz, sesama, derever, jangan, sampek, meng...	maaf bos sesama derever jangan sampai menghuja...	maaf bos sesama derever jangan menghujat carik...	maaf bos sama derever jangan hujat carik rejek...
3	yg saya tau driver jos jos malah diemseperti p...	yg, saya, tau, driver, jos, jos, malah, diemse...	yang saya tau driver jos jos malah diemseperti...	tau driver jos jos malah diemseperti padi sema...	tau driver jos jos malah diemseperti padi maki...
4	narik suka aja masih gacor efek udah gawe inti...	narik, suka, aja, masih, gacor, efek, udah, ga...	menarik suka saja masih gacor efek sudah gawe ...	menarik suka gacor efek gawe intinya tau ilmu ...	tarik suka gacor efek gawe inti tau ilmu gojek
5	hasil mingguangambar atasnya di potong	hasil, mingguangambar, atasnya, di, potong	hasil mingguangambar atasnya di potong	hasil mingguangambar atasnya potong	hasil mingguangambar atas potong
6	kesimpulan apa yg didapat dari postingan saya ...	kesimpulan, apa, yg, didapat, dari, postingan,...	kesimpulan apa yang didapat dari postingan say...	kesimpulan didapat postingan commentan orang a...	simpul dapat postingan commentan orang atas in...
7	sehat	sehat	sehat	sehat	sehat
8	segol dong	segol, dong	segol dong	segol	gol
9	sigitu aja udah pamer	sigitu, aja, udah, pamer	sigitu saja sudah pamer	sigitu pamer	sigitu pamer

Gambar 4. 16 Hasil Stemming

4.3. Labelling

Setelah data komentar melalui tahap *preprocessing*, tahap berikutnya adalah *labelling* atau pemberian label sentimen pada setiap komentar. Dalam penelitian ini, klasifikasi sentimen dibagi menjadi dua kategori, yaitu positif dan *negative*, dengan fokus utama diarahkan pada komentar yang mengandung unsur ujaran kebencian terhadap *driver* Gojek. Proses pelabelan dilakukan secara manual dengan membaca dan menganalisis isi setiap komentar, lalu menentukan apakah komentar

tersebut tergolong sebagai ujaran kebencian (*negative*) atau tidak (positif). Komentar yang mengandung hinaan, caci maki, provokasi, atau sindiran kasar kepada driver Gojek diberi label "*Negative*", sedangkan komentar yang mendukung, netral, atau memberi pembelaan kepada driver diberi label "Positif".

```
import pandas as pd
import random

# Masukkan kamus leksikon positif dan negatif, yang diambil dari GitHub
positive_df = pd.read_csv('positive.tsv', sep='\t')
negative_df = pd.read_csv('negative.tsv', sep='\t')

# Assuming the lexicon files have a column named 'word'
positive_lexicon = set(positive_df['word'])
negative_lexicon = set(negative_df['word'])

# Fungsi untuk menentukan sentimen dan menghitung skornya
def determine_sentiment(text):
  if isinstance(text, str):
    positive_count = sum(1 for word in text.split() if word in positive_lexicon)
    negative_count = sum(1 for word in text.split() if word in negative_lexicon)
    sentiment_score = positive_count - negative_count

# Jika skor 0, ubah menjadi positif atau negatif secara acak
    if sentiment_score == 0:
      sentiment_score = random.choice([1, -1])

    if sentiment_score > 0:
      sentiment = "Positif"
    else:
      sentiment = "Negatif"

    return sentiment_score, sentiment
  return 0, "Netral" # Return Netral for non-string inputs

# Tentukan sentimen dan skor untuk setiap ulasan
df[['score', 'Sentiment']] = df['Komentar_Stemmed'].apply(lambda x:
pd.Series(determine_sentiment(x)))

# Tampilkan hasilnya
display(df[['Komentar_Stemmed', 'score', 'Sentiment']].head(10))
```

Gambar 4. 17 Labelling-Code

Source code tersebut menunjukkan proses pelabelan data komentar secara otomatis berdasarkan pendekatan *lexicon-based*, yaitu dengan memanfaatkan dua kamus kata (*lexicon*) berisi daftar kata positif dan *negative* yang masing-masing dimuat dari file positive.tsv dan negative.tsv. File tersebut dipisahkan oleh tab (\t) dan diasumsikan memiliki kolom bernama 'word'. Setiap kata dari kedua kamus ini kemudian dimasukkan ke dalam struktur set() untuk mempercepat proses pencocokan. Selanjutnya, dibuat sebuah fungsi bernama determine_sentiment()

yang menerima masukan berupa teks komentar. Fungsi ini akan memecah kalimat menjadi kata-kata, lalu menghitung jumlah kata yang termasuk ke dalam daftar kata positif maupun *negative*. Nilai sentimen kemudian dihitung dengan mengurangkan jumlah kata *negative* dari jumlah kata positif. Jika nilai akhirnya positif, maka komentar diberi label "Positif"; jika *negative*, diberi label "Negatif". Namun, jika hasilnya nol (artinya tidak ditemukan kata dari kedua kategori tersebut), maka label ditentukan secara acak antara positif atau *negative* menggunakan fungsi random.choice([1, -1]) sebagai metode penyeimbang. Fungsi ini kemudian diterapkan pada kolom Komentar_Stemmed menggunakan apply() dan hasilnya disimpan ke dalam dua kolom baru, yaitu score untuk nilai numerik sentimen dan sentiment untuk label kategorikal.

1 to 10 of 10 entries Filter

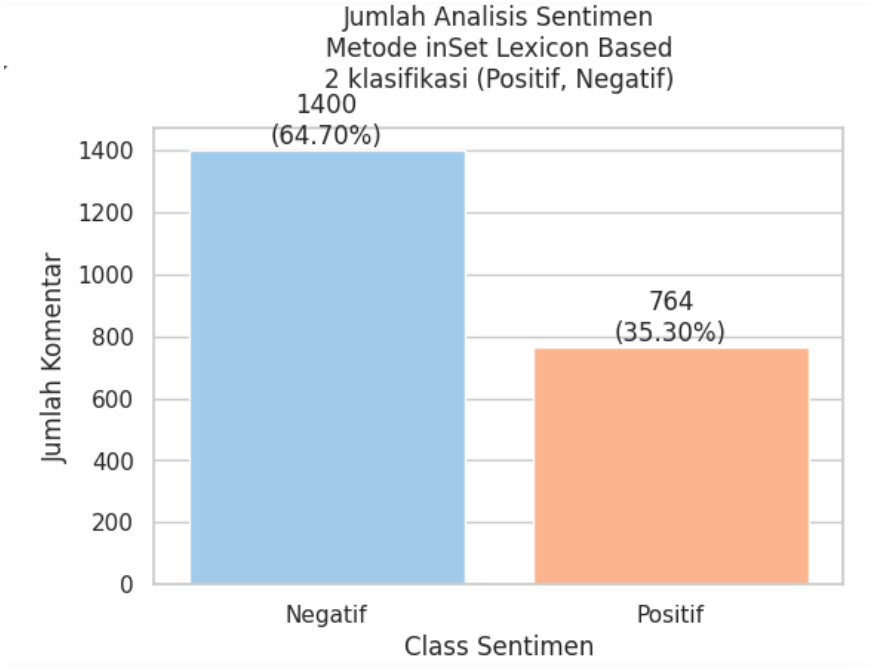
index	Komentar_Stemmed	score	Sentiment
0	benar	1	Positif
1	usah bawa orang tua abang	-2	Negatif
2	maaf bos sama derever jangan hujat carik rejeki keluarga biyarkan rejeki jadi berkah salam satu aspal cuman ngasik saran bukan kritik	1	Positif
3	tau driver jos jos malah diemseperti padi makin isi makin tunduk	1	Positif
4	tarik suka gacor efek gawe inti tau ilmu gojek	2	Positif
5	hasil mingguangambar atas potong	-1	Negatif
6	simpul dapat postingan commentan orang atas inijebakan betmen padahal postingpun sandar milu jangan makan isu hoax potong vidio bau suku ras agama biar indonesia pecah belah saat pamer hasil situ banyak manusia sombong congkak aku langit hehehhehe saat posting driver kena musibah order fiktif mixue juta driver pura buta bisu tuli padahal donasi bantu ringan driver sebut slogan salam satu aspal hanya buah wacana belaka padahal banyak pamer hasil tipping bagai rumput tetangga selalu lihat lebih hijau jangan nyinyir iri padahal jumlah hatters medsos banyak habis banyak comment hapus malu baca comment an kalau akun asli lanjut sendiri gan otw pulang dulu	-15	Negatif
7	sehat	1	Positif
8	gol	1	Positif
9	sigitu pamer	-1	Negatif

Gambar 4. 18 Hasil Labelling

Dari hasil *labelling* tersebut terdapat komentar pada nomor 2 "usah bawa orang tua bang", diklasifikasikan sebagai *Negative* dengan skor –2, dan klasifikasi ini masuk akal karena kalimat ini bernada menyindir atau menyalahkan seseorang dengan membawa urusan pribadi (orang tua) ke dalam pembicaraan publik. Kata "usah" merupakan bentuk tidak baku dari "jangan", yang sering digunakan dalam konteks menyuruh orang berhenti atau tidak melakukan sesuatu. Jika kata "usah" tidak masuk dalam leksikon *negative,* maka kata ini sebenarnya tidak berkontribusi

terhadap skor, padahal punya nada larangan yang cukup kuat. Kata “bawa orang tua” meskipun tidak mengandung kata-kata *negative* secara langsung, tapi dalam konteks sosial, membawa-bawa orang tua dalam diskusi bisa bernuansa menyerang pribadi, dan biasanya dianggap kurang sopan atau menyakitkan.

Komentar nomor 5 “hasil mingguangambar atas potong” diklasifikasikan sebagai *Negative* dengan skor –1. Dari sisi kalimat, hasil ini cukup masuk akal, karena komentarnya bernada keluhan atau komplain, meskipun penyampaiannya tidak kasar. Kata “potong” mungkin masuk dalam daftar leksikon negatif, karena bisa diasosiasikan dengan pemutusan, pengurangan, atau sesuatu yang dipotong secara tidak menyenangkan. Kata lainnya seperti “hasil”, “minggu”, “gambar” kemungkinan tidak berpengaruh (netral), jadi skor –1 kemungkinan hanya berasal dari kata “potong”. Komentar nomor 9 "sigitu pamer" diklasifikasikan sebagai *Negative* dengan skor -1. Hasil ini masuk akal, karena dari sisi makna dan konteks, komentar ini memang memiliki nada menyindir atau mengejek secara halus. Kata "pamer" secara umum memiliki konotasi *negative* dalam percakapan bahasa Indonesia, karena merujuk pada sifat membanggakan diri secara berlebihan. Kata "sigitu" sendiri memang tidak baku, tapi karena "pamer" cukup kuat, komentar ini langsung terklasifikasi sebagai *Negative*.



Gambar 4. 19 Hasil visualisasi dari analisis sentimen menggunakan metode inSet Lexicon Based

Diagram batang di atas menunjukkan hasil analisis sentimen terhadap komentar pengguna di Facebook dengan menggunakan metode inSet *Lexicon Based*. Metode inSet adalah cara untuk menentukan sentimen suatu komentar dengan melihat apakah kata-kata dalam komentar itu ada di dalam daftar kata yang sudah dikelompokkan sebagai kata positif atau kata *negative*. Disebut "inSet" karena prosesnya berdasarkan pengecekan apakah kata itu “ada di dalam kumpulan” (set) kata positif atau *negative*. Berdasarkan grafik batang tersebut, terlihat bahwa komentar dengan sentimen *negative* mendominasi jumlah data, yaitu sebanyak 1.400 yang berarti komentar-komentar tersebut berisi kritik, ujaran kasar, atau ketidaksukaan terhadap topik yang dibahas. Jumlah ini setara dengan 64,70% dari total data yang dianalisis. Sementara itu, sebanyak 764 komentar diklasifikasikan sebagai Positif, yang artinya komentar tersebut bernada dukungan, pujian, atau menunjukkan perasaan yang baik terhadap topik yang disampaikan. Presentase komentar positif ini hanya sebesar 35,30% dari total data. Visualisasi ini

memperlihatkan bahwa mayoritas komentar yang dianalisis memiliki nada *negative*, menunjukkan kecenderungan adanya sentimen *negative* yang cukup dominan pada data yang dikumpulkan.

4.4. Split Data Menggunakan CountVectorizer

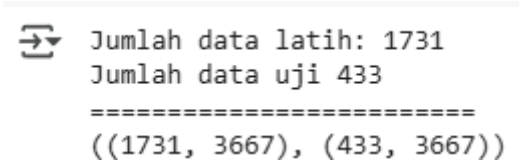
Split data adalah proses membagi data mentah menjadi dua bagian utama, yaitu data latih dan data uji. Data latih digunakan untuk melatih model agar model tersebut bisa mengenali pola atau struktur dalam data. Sementara itu, data uji digunakan untuk menguji seberapa baik model yang telah dilatih tersebut mampu memprediksi data baru yang belum pernah ia lihat sebelumnya. Tujuannya agar hasil analisis tidak hanya bagus pada data yang sudah dikenali, tapi juga akurat saat digunakan pada data lain. Biasanya, pembagian ini menggunakan perbandingan 80:20 atau 70:30, artinya 80% data dijadikan data latih dan sisanya sebagai data uji. Setelah data dibagi, data komentar yang masih berbentuk teks tidak bisa langsung digunakan oleh model, karena model hanya bisa membaca data dalam bentuk angka. Di sinilah CountVectorizer digunakan, CountVectorizer akan mengubah kata-kata dalam komentar menjadi angka berdasarkan seberapa sering kata tersebut muncul.

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer

x = df['Komentar_Stemmed']
y = df['Sentiment']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,
random_state=42)

print("Jumlah data latih:", len(x_train))
print("Jumlah data uji", len(x_test))
print('=========================')

vectorizer = CountVectorizer()
x_train_vec = vectorizer.fit_transform(x_train)
x_test_vec = vectorizer.transform(x_test)

x_train_vec.shape, x_test_vec.shape
```

Gambar 4. 20 Split Data-Code

Source code di atas merupakan proses pembagian data (split data) dan konversi teks ke bentuk numerik (*vectorisasi*) menggunakan CountVectorizer. Pertama-tama, data teks hasil *stemming* dari kolom 'Komentar_Stemmed' disimpan ke dalam variabel x, sedangkan label sentimennya (Positif/Negatif) dari kolom 'Sentiment' disimpan ke dalam variabel y. Fungsi train_test_split digunakan untuk membagi data menjadi dua bagian, yaitu data latih (*training*) dan data uji (*testing*). Parameter test_size=0.2 berarti 20% dari total data digunakan sebagai data uji, sedangkan 80% sisanya sebagai data latih. random_state=42 digunakan agar hasil pembagian data tetap konsisten setiap kali dijalankan. Setelah dibagi, program mencetak jumlah data latih dan data uji ke layar. Selanjutnya, CountVectorizer digunakan untuk mengubah teks ke dalam bentuk angka. fit_transform() digunakan pada data latih untuk membangun model dan langsung menerapkannya ke data latih, sedangkan transform() digunakan pada data uji menggunakan model vektor yang sama, tanpa membangunnya kembali. Hasilnya adalah data dalam bentuk matriks fitur, yang siap digunakan untuk pelatihan model *machine learning*.

Terakhir, kode x_train_vec.shape, x_test_vec.shape digunakan untuk menampilkan ukuran data latih dan uji setelah proses vektorisasi.



Gambar 4. 21 Hasil Split Data

Dari hasil tersebut, terlihat bahwa proses pembagian data berhasil menghasilkan 1731 data latih dan 433 data uji. Ini artinya, sekitar 80% dari total data digunakan untuk melatih model, sementara sisanya, yaitu 20%, digunakan untuk menguji seberapa baik model tersebut bekerja. Kemudian, hasil transformasi dari teks menjadi bentuk numerik dengan CountVectorizer menunjukkan dimensi (1731, 3667) untuk data latih dan (433, 3667) untuk data uji. Angka 3667 menunjukkan jumlah fitur atau kata unik yang berhasil diidentifikasi dari seluruh kumpulan komentar. Setiap baris pada bentuk vektor tersebut mewakili satu komentar, dan setiap kolom mewakili frekuensi atau keberadaan sebuah kata tertentu di komentar tersebut. Bentuk data seperti ini sangat penting sebagai masukan bagi algoritma pembelajaran mesin yang tidak dapat memproses teks secara langsung, melainkan membutuhkan data dalam bentuk angka.

4.5. Visualisasi Data

Visualisasi data adalah cara untuk menampilkan data dalam bentuk gambar atau grafik supaya lebih gampang dibaca dan dimengerti. Daripada lihat angka-angka di tabel, lebih mudah memahami data kalau ditampilkan dengan grafik batang, garis, atau *pie chart*. Dalam konteks pembagian data seperti data *training*

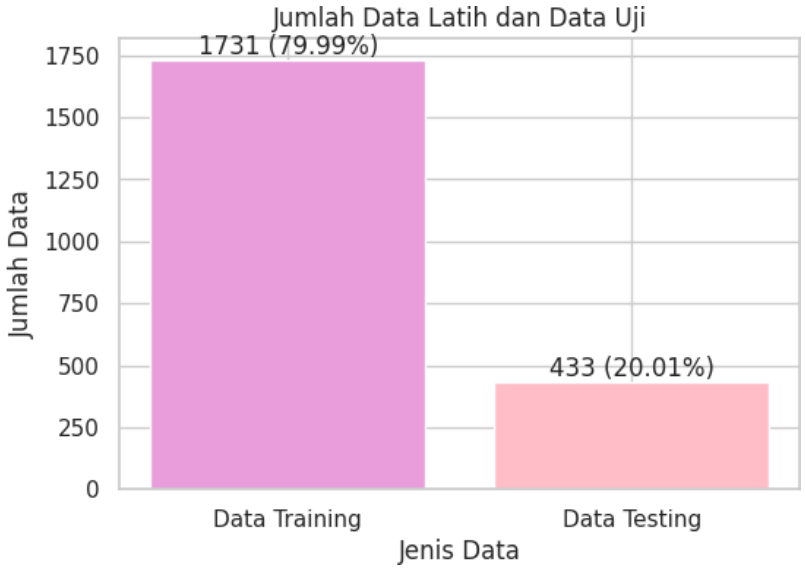
dan data *testing*, visualisasi ini dipakai buat menunjukkan seberapa banyak jumlah masing-masing data setelah dibagi.

```
import matplotlib.pyplot as plt


train_size = len(x_train)
test_size = len(x_test)

plt.figure(figsize=(6, 4))
bars = plt.bar(['Data Training', 'Data Testing'], [train_size, test_size], color=
['plum', 'pink'])

for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 1, f'{height} ({height /
(train_size + test_size) * 100:.2f}%)',
             ha='center', va='bottom')

plt.title('Jumlah Data Latih dan Data Uji')
plt.xlabel('Jenis Data')
plt.ylabel('Jumlah Data')
plt.show()
```

Gambar 4. 22 Visualisasi Data-Code

Source code tersebut digunakan untuk memvisualisasikan jumlah data latih dan data uji dalam bentuk diagram batang. Visualisasi ini dibuat dengan menggunakan library matplotlib.pyplot, yang populer untuk membuat grafik di Python. Pertama, nilai train_size dan test_size dihitung menggunakan fungsi len(), yang masing-masing menyimpan jumlah data latih (x_train) dan data uji (x_test). Kemudian plt.figure(figsize=(6, 4)) digunakan untuk mengatur ukuran gambar. Selanjutnya, plt.bar() membuat grafik batang dengan label "Data Training" dan "Data Testing", serta menampilkan jumlahnya sesuai dengan nilai train_size dan test_size. Warna batang diatur menggunakan parameter color, dalam hal ini warna ungu muda ('plum') untuk data training dan pink untuk data testing. Bagian for bar in bars: bertujuan menambahkan teks label ke atas setiap batang grafik. Label tersebut berisi jumlah data pada masing-masing batang serta persentasenya terhadap total data, dengan format seperti: 1731 (80.00%). Angka ini diposisikan

secara horizontal di tengah batang dan vertikal di atas batang. Lalu plt.title(), plt.xlabel(), dan plt.ylabel() digunakan untuk menambahkan judul dan label sumbu pada grafik. Terakhir, plt.show() digunakan untuk menampilkan grafik ke layar.



Gambar 4. 23 Hasil Visualisasi Data

Hasil tersebut menunjukkan visualisasi pembagian data menjadi dua bagian, yaitu data *training* dan data *testing*. Pada grafik batang terlihat bahwa sebanyak 1.731 data digunakan sebagai data *training*, yang mewakili 79,99% dari total data yang tersedia. Sementara itu, sebanyak 433 data digunakan sebagai data *testing*, atau sebesar 20,01%. Pembagian ini dilakukan berdasarkan parameter test_size=0.2, yang berarti 20 persen dari seluruh data digunakan untuk menguji model, sedangkan sisanya digunakan untuk melatih model.

4.6. Naïve Bayes

Naïve Bayes yang digunakan dalam penelitian ini adalah algoritma klasifikasi yang cocok untuk data teks seperti komentar, yang menggunakan model MultinomialNB dilatih dengan data hasil transformasi CountVectorizer dan

digunakan untuk memprediksi sentimen komentar, apakah termasuk positif atau *negative*. Selanjutnya akan dievaluasi menggunakan *confusion matrix*, yang divisualisasikan dalam bentuk *heatmap*. Dari visualisasi tersebut, bisa diketahui seberapa banyak komentar yang berhasil diklasifikasikan dengan benar atau salah oleh model.

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Inisialisasi model Naive Bayes
models = {
    "Naive Bayes": MultinomialNB(),
}

# Train models
results = {}
for model_name, model in models.items():
    model.fit(x_train_vec, y_train)
    y_pred = model.predict(x_test_vec)
    results[model_name] = {
        "accuracy": accuracy_score(y_test, y_pred),
        "classification_report": classification_report(y_test, y_pred,
output_dict=True),
        "confusion_matrix": confusion_matrix(y_test, y_pred)
    }
```

Gambar 4. 24 Proses penerapan algoritma Naive Bayes-Code

Source code di atas merupakan proses penerapan algoritma Naïve Bayes, tepatnya model MultinomialNB dari *library* sklearn. Model ini digunakan untuk mengklasifikasikan data teks yang sebelumnya sudah diubah menjadi bentuk numerik dengan CountVectorizer. Model Naïve Bayes diinisialisasi dan disimpan dalam *dictionary* bernama models. Kemudian, dilakukan proses pelatihan model (fit) menggunakan data latih x_train_vec dan y_train. Setelah model dilatih, model digunakan untuk memprediksi hasil klasifikasi pada data uji (x_test_vec), dan hasilnya disimpan dalam variabel y_pred. Selanjutnya, hasil evaluasi model disimpan ke dalam *dictionary* results. Ada tiga jenis evaluasi yang dilakukan, yaitu *Accuracy Score* untuk mengukur tingkat akurasi prediksi model, *Classification Report* untuk menampilkan metrik evaluasi seperti *precision*, *recall*, dan f1-*score.*

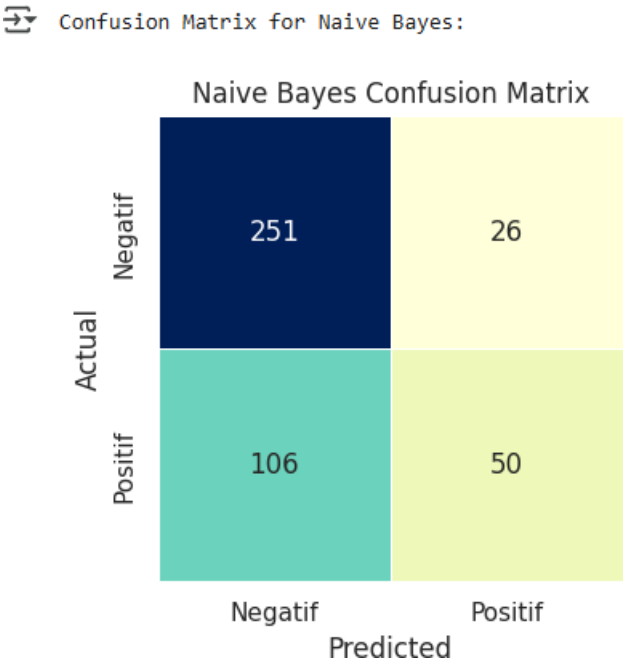
Dan *Confusion Matrix* untuk melihat perbandingan antara hasil prediksi dan label asli.

```
 import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Inisialisasi model Naive Bayes
models = {
    "Naive Bayes": MultinomialNB(),
}

# Train models
results = {}
for model_name, model in models.items():
    model.fit(x_train_vec, y_train)
    y_pred = model.predict(x_test_vec)
    results[model_name] = {
        "accuracy": accuracy_score(y_test, y_pred),
        "classification_report": classification_report(y_test, y_pred,
output_dict=True),
        "confusion_matrix": confusion_matrix(y_test, y_pred)
    }

for model_name, result in results.items():
    fig, ax = plt.subplots(figsize=(4, 4))
    sns.heatmap(
        result["confusion_matrix"],
        annot=True,
        fmt="d",
        cmap="YlGnBu",
        cbar=False,
        xticklabels=['Negatif', 'Positif'],
        yticklabels=['Negatif', 'Positif'],
        ax=ax,
        square=True,
        linewidths=0.5
)
print(f"\nConfusion Matrix for {model_name}:\n")
ax.set_title(f"{model_name} Confusion Matrix", fontsize=12)
ax.set_xlabel("Predicted", fontsize=12)
ax.set_ylabel("Actual", fontsize=12)

plt.tight_layout()
plt.show()
```

Gambar 4. 25 Menampilkan ConfusionMatrix dalam bentuk visual (heatmap)-Code

Sorce code tersebut berfungsi untuk menampilkan *confusion matrix* dalam bentuk visual (*heatmap*) menggunakan *library* matplotlib dan seaborn. *Confusion matrix* sendiri adalah bagian penting dari evaluasi hasil klasifikasi dalam proses penerapan algoritma Naïve Bayes sebelumnya. Secara alur, setelah model Naïve Bayes dilatih dan menghasilkan prediksi terhadap data uji, langkah selanjutnya adalah mengevaluasi seberapa tepat hasil klasifikasinya. confusion_matrix yang sebelumnya sudah disimpan ke dalam dictionary results pada Naïve Bayes,

sekarang digunakan untuk divisualisasikan agar lebih mudah dibaca dan dianalisis. sns.heatmap() digunakan untuk menggambar confusion matrix dalam bentuk warna biru kehijauan karena menggunakan cmap="YlGnBu". annot=True menampilkan angka di dalam setiap kotak. fmt="d" memastikan angka ditampilkan sebagai bilangan bulat (integer). Label sumbu x dan y disesuaikan dengan kelas 'Positif' dan 'Negatif'. ax.set_xlabel() dan ax.set_ylabel() digunakan untuk memberi label sumbu. ax.set_title() memberi judul untuk confusion matrix-nya.



Gambar 4. 26 Hasil Confusion Matrix

Berdasarkan *confusion matrix* di atas, dapat dilihat bahwa dari seluruh data uji, model Naïve Bayes berhasil mengklasifikasikan sebanyak 251 komentar *negative* dengan benar, sementara 26 komentar *negative* salah diklasifikasikan sebagai positif. Untuk komentar positif sebanyak 50 komentar berhasil dikenali dengan benar sebagai positif, tetapi ada 106 komentar positif yang justru diklasifikasikan secara keliru sebagai *negative*. Dari hasil ini dapat disimpulkan bahwa model cenderung lebih akurat dalam mengenali komentar positif

dibandingkan dengan komentar *negative*. Kelemahan model tampak jelas pada bagian komentar *negative* yang banyak salah klasifikasi, sehingga perlu dievaluasi lebih lanjut pada sisi data atau teknik pemodelannya.

Untuk memastikan kesesuaian *confusion matrix* tersebut, dilakukan pula perhitungan manual terhadap metrik evaluasi pada kelas Negatif yaitu akurasi, precision, recall, dan f1-score. Perhitungan ini bertujuan untuk memverifikasi bahwa nilai-nilai yang dihasilkan model memang sesuai dengan logika evaluasi berdasarkan data pada *confusion matrix* sebelumnya. Berikut perhitungannya :

Tabel 4. 1 Hail evaluasi

	Predicted Positif	Predicted Negatif
Actual Positif	50 (TP)	106 (FN)
Actual Negatif	26 (FP)	251 (TN)

1. Akurasi

$$\text{Akurasi} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = \frac{50 + 251}{50 + 251 + 26 + 106} = \frac{301}{433}$$

$$\approx 0.695 \text{ atau } 69{,}5\%$$

2. Precision per kelas

1) Positif

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{50}{50 + 26} = \frac{50}{76} \approx 0{,}658 \text{ atau } 65{,}8\%$$

2) Negative

$$\text{Precision} = \frac{\text{TN}}{\text{TN} + \text{FN}} = \frac{251}{251 + 106} = \frac{251}{357} \approx 0{,}703 \text{ atau } 70{,}3\%$$

3. Recall per kelas

1) Positif

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{50}{50 + 106} = \frac{50}{156} \approx 0{,}321 \text{ atau } 32{,}1\%$$

2) Negative

$$\text{Recall} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{251}{251 + 26} = \frac{251}{277} \approx 0{,}906 \text{ atau } 90{,}6\%$$

4. F1-Score per kelas

1) Positif

$$\text{F1}_{\text{Positif}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0{,}658 \times 0{,}321}{0{,}658 + 0{,}321}$$

$$\approx 0{,}431 \text{ atau } 43{,}1\%$$

2) Negative

$$\text{F1}_{\text{Negatif}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0{,}703 \times 0{,}906}{0{,}703 + 0{,}906} = 2 \times \frac{0{,}636}{1{,}609}$$

$$\approx 0{,}792 \text{ atau } 79{,}2\%$$

4.7. Evaluasi Model

Kelemahan model yang telah dijelaskan sebelumnya, terutama pada ketidakseimbangan dalam mengenali sentimen positif, perlu dievaluasi lebih lanjut baik dari sisi data maupun teknik pemodelannya. Untuk itu, dilakukan evaluasi model, yaitu tahap penting setelah pelatihan model. Evaluasi model bertujuan untuk mengetahui seberapa baik model dalam melakukan prediksi pada data uji, serta mengidentifikasi bagian mana dari model atau data yang perlu ditingkatkan. Evaluasi ini menggunakan berbagai metrik seperti akurasi, *precision*, *recall*, dan f1-*score*.

```
rom IPython.display import display

for model_name, result in results.items():
    print(f"\nClassification Report for {model_name}:")

    report_df = pd.DataFrame(result['classification_report']).transpose()

    styled_df = report_df.style.background_gradient(cmap="coolwarm")
    styled_df = styled_df.format(precision=3)
    display(styled_df)
```

Gambar 4. 27 Evaluasi Model-Code

Kode from IPython.display import display digunakan untuk menampilkan objek DataFrame secara interaktif di notebook seperti Google Colab atau Jupyter Notebook. Kemudian dilakukan perulangan dengan for model_name, result in results.items():, yang berarti jika ada beberapa model, maka semua hasil evaluasinya bisa ditampilkan satu per satu berdasarkan nama model. Baris report_df = pd.DataFrame(result['classification_report']).transpose() mengubah hasil classification report yang sebelumnya dalam bentuk dictionary menjadi sebuah DataFrame dan ditranspose agar setiap metrik seperti *precision*, *recall*, dan f1-*score* berada di baris yang berbeda, bukan kolom. Ini membuat tampilannya lebih mudah dibaca. Lalu, report_df.style.background_gradient(cmap="coolwarm") adalah styling visual agar hasil evaluasi memiliki gradasi warna semakin tinggi nilainya, semakin merah atau biru tergantung colormap, ini hanya untuk kejelasan visual. Baris styled_df = styled_df.format(precision=3) dipakai untuk mengatur presisi angka, yaitu membulatkan nilai metrik evaluasi hingga 3 angka di belakang koma. Terakhir, display(styled_df) berfungsi untuk menampilkan hasil evaluasi tersebut secara interaktif dan berwarna.

Classification Report for Naive Bayes:

	precision	recall	f1-score	support
Negatif	0.703	0.906	0.792	277.000
Positif	0.658	0.321	0.431	156.000
accuracy	0.695	0.695	0.695	0.695
macro avg	0.680	0.613	0.611	433.000
weighted avg	0.687	0.695	0.662	433.000

Gambar 4. 28 Hasil Evaluasi Model

Berdasarkan hasil tersebut, model Naive Bayes pada analisis sentimen ujaran kebencian, diketahui bahwa model ini memiliki tingkat akurasi sebesar 0.695 atau setara dengan 69,5%. Jika dilihat lebih lanjut, performa model untuk kelas Negatif cukup baik, dengan nilai *precision* sebesar 0.703, *recall* 0.906, dan f1-*score* mencapai 0.792. Ini menunjukkan bahwa model lebih mampu mengenali komentar-komentar *negative* secara konsisten dan cukup akurat. Sebaliknya, performa untuk kelas Positif masih tergolong rendah, dengan nilai *precision* hanya 0.658, *recall* 0.321, dan f1-*score* sebesar 0.431. Rendahnya nilai *recall* pada kelas positif menandakan bahwa model sering gagal mendeteksi komentar yang sebenarnya positif. Hal ini dapat mengindikasikan ketidakseimbangan data atau kurang optimalnya model dalam memahami ciri-ciri dari komentar positif. Sementara itu, nilai rata-rata makro (macro avg) menunjukkan *precision* sebesar 0.680, *recall* 0.613, dan f1-*score* sebesar 0.611, sedangkan *weighted average* memperlihatkan nilai yang sedikit lebih tinggi, yaitu *precision* 0.687, *recall* 0.695, dan f1-*score* 0.662. Nilai-nilai ini secara umum menggambarkan bahwa performa model masih lebih condong ke kelas *negative* dan belum cukup seimbang dalam mengenali kedua kelas. Hasil perhitungan manual sebelumnya dapat disimpulkan bahwa

model memang lebih unggul dalam mendeteksi komentar *negative* dan sudah bekerja cukup baik pada kelas tersebut.

4.8. Word Cloud

Tahapan terakhir adalah visualisasi data menggunakan word cloud untuk mengetahui kata yang sering muncul pada setiap sentimen.

Gambar 4. 29 Sentimen Positif

Word cloud sentimen positif di atas memperlihatkan kata-kata yang paling sering muncul dalam komentar yang tergolong ke dalam kategori positif. Kata yang paling dominan adalah “sama”, “buat”, “orang”, “abang”, dan “kerja”, yang ditampilkan dalam ukuran huruf paling besar. Ini menunjukkan bahwa dalam komentar yang bernada positif, pengguna sering menggunakan kata-kata yang berkaitan dengan rasa kebersamaan, aktivitas bekerja, serta sapaan atau bentuk penghormatan seperti “abang”. Selain itu, kata seperti “polisi”, “jasa”, “pak”, “customer”, dan “driver” juga sering muncul, yang dapat mencerminkan dukungan terhadap pihak tertentu atau bentuk pelayanan. *Word cloud* ini memberikan

gambaran umum tentang pola kata yang sering digunakan dalam komentar positif tanpa perlu membaca satu per satu isi komentar.

Gambar 4. 30 Sentimen Negative

Word cloud sentimen *negative* di atas menunjukkan kumpulan kata yang sering muncul dalam komentar bernada negatif. Kata “kalau”, “abang”, “driver”, “kamu”, dan “ojol” tampak paling dominan karena ditampilkan dengan ukuran paling besar. Ini menandakan bahwa kata-kata tersebut paling sering digunakan dalam komentar negatif. Kata lain seperti “jangan”, “order”, “kerja”, “polisi”, dan “akun” juga muncul cukup menonjol. Umumnya, kata-kata ini digunakan dalam konteks keluhan, sindiran, atau protes terhadap kejadian yang dianggap merugikan atau tidak sesuai harapan. Kata seperti “anjing” dan “babi” umumnya digunakan sebagai bentuk makian. Sementara “bau”, “gendut”, dan “sombong” bisa menjadi bentuk ejekan terhadap fisik atau kepribadian. Sedangkan kata “pecat” mengindikasikan komentar yang menyarankan atau menuntut tindakan terhadap seseorang (misalnya *driver* atau pihak Gojek). Kehadiran kata-kata ini memperkuat bahwa dalam analisis sentimen, sentimen *negative* tidak hanya bermakna

ketidakpuasan, tetapi juga bisa berisi ekspresi kebencian, hinaan, atau kekerasan verbal.

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai analisis sentimen terhadap komentar pengguna Facebook yang berkaitan dengan driver Gojek, dapat disimpulkan bahwa :

1. Komentar yang mengandung ujaran kebencian terhadap driver Gojek dapat dideteksi dan diklasifikasikan menggunakan algoritma Naïve Bayes, khususnya model MultinomialNB, dengan cukup baik. Model ini efektif dalam mengenali pola kata yang sering muncul dalam komentar *negative*.
2. Proses *preprocessing* yang terdiri dari *case folding*, *cleaning* data, tokenisasi, normalisasi, *stopword removal*, dan *stemming* terbukti berperan penting dalam meningkatkan kualitas data sebelum dilakukan klasifikasi.
3. Konversi data teks ke dalam bentuk numerik menggunakan CountVectorizer memungkinkan algoritma Naïve Bayes untuk melakukan proses pelatihan dan prediksi berdasarkan frekuensi kata.
4. Hasil evaluasi model menunjukkan bahwa sistem dapat mengidentifikasi sentimen *negative* secara lebih dominan dibandingkan sentimen positif, sesuai dengan fokus penelitian ini, yaitu mengamati potensi ujaran kebencian dalam komentar netizen terhadap *driver* Gojek.
5. Pembuatan visualisasi *Word Cloud* berhasil memperlihatkan kata-kata yang sering muncul dalam komentar negatif, seperti “anjing”, “babi”, “pecat”, dan sebagainya. Hal ini memperkuat bahwa komentar di media sosial masih

menyimpan potensi ujaran yang merendahkan dan kasar terhadap profesi tertentu.

5.2. Saran

Adapun saran yang dapat disampaikan penulis untuk penelitian selanjutnya adalah :

1. Penelitian ini hanya menggunakan dua label sentimen, yaitu positif dan *negative*. Untuk penelitian berikutnya, disarankan menambahkan kategori netral atau membuat pembobotan tingkat kebencian (misalnya ringan, sedang, kuat) agar hasil klasifikasi lebih rinci.
2. Jumlah data yang digunakan masih terbatas dan hanya berasal dari satu platform, yaitu Facebook. Penelitian selanjutnya dapat memperluas sumber data dari platform lain seperti Twitter, TikTok, atau YouTube agar hasil analisis menjadi lebih menyeluruh.
3. Untuk hasil yang lebih akurat, dapat dipertimbangkan penggunaan metode klasifikasi lain seperti Support Vector Machine (SVM) atau model berbasis *deep learning* seperti LSTM.
4. Leksikon kata positif dan *negative* yang digunakan untuk pelabelan bisa lebih diperluas atau diperbarui agar mencakup lebih banyak ragam kata yang digunakan oleh netizen, termasuk kata-kata gaul atau slang yang sedang tren.
5. Perlu adanya kolaborasi lebih lanjut dengan pihak Gojek atau lembaga terkait untuk menindaklanjuti temuan ujaran kebencian ini, sebagai bentuk perlindungan terhadap profesi *driver* ojek online di ruang digital.

DAFTAR PUSTAKA

Arfan, I. S., Fauziah, S., & Nawangsih, I. (2024). Analisis Sentimen Terhadap Cyber Bullying di X Menggunakan Algoritma Naïve Bayes: Sentiment Analyst of Cyber Bullying in X Using Naïve Bayes Algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(4), 1411–1419.

Belghaouti, O., Handouzi, W., & Tabaa, M. (2020). Improved traffic sign recognition using deep convnet architecture. *Procedia Computer Science*, *177*, 468–473. https://doi.org/10.1016/j.procs.2020.10.064

Cybertech, J., Widyanti, A., Nofriansyah, D., & Rizky, F. (2020). *SISTEM PAKAR DALAM MENGANALISA PENYAKIT SINUSITIS*. *September*.

Dzikri, M. H., Setiawan, I. R., & Indrayana, D. (2024). PENERAPAN ALGORITMA NAIVE BAYES UNTUK MENDETEKSI PENIPUAN LOWONGAN PEKERJAAN. *Simtek: Jurnal Sistem Informasi Dan Teknik Komputer*, *9*(2), 102–109.

Eleanora, F. N., & Adawiah, R. Al. (2021). Sosialisasi Bahaya Dan Dampak Cyberbullying (Perundungan Dunia Maya) Bagi Pelajar Di Sekolah Menengah Kejuruan (SMK) Negeri 3 Bekasi. *Jurnal Pengabdian Barelang*, *3*(01), 70–72. https://doi.org/10.33884/jpb.v3i01.2685

Elfansyah, M. R., Perdana, M. R., Nabawi, I. N. T. I., & Rudiman, R. (2024). Analisis Sentiment Cyberbullying pada media Youtube menggunakan Algoritma Naïve Bayes. *KOMPUTEK : Jurnal Teknik Universitas Muhammadiyah Ponorogo*, *8*(1), 29–35. http://studentjournal.umpo.ac.id/index.php/komputek

Fani, M., & Ardiansah, B. (n.d.). *The Relationship between Religiosity and Emotional Regulation in Online OJek Drivers in Sidoarjo [Hubungan Antara Religiusitas Dengan Regulasi Emosi Pada Driver Ojek Online di Sidoarjo]*. 1–12.

Handika. (2024). Pemanfaatan Python dan Google Colab Dalam Pembelajaran Statistika Deskriptif. *Edumatnesia: Prosiding Seminar Nasional Matematika Dan Pendidikan*

Matematika, 379–389.

Hasri, C. F., & Alita, D. (2022). Penerapan Metode Naïve Bayes Classifier Dan Support Vector Machine Pada Analisis Sentimen Terhadap Dampak Virus Corona Di Twitter. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *3*(2), 145–160.

Hutagalung, A. S., Negara, A. B. P., & Pratama, E. E. (2021). Aplikasi Pendeteksi Cyberbullying Terhadap Komentar Postingan Media Sosial Instagram dengan Metode Naïve Bayes Classifier Berbasis Website. *Jurnal Sistem Dan Teknologi Informasi (Justin)*, *9*(3), 364. https://doi.org/10.26418/justin.v9i3.44843

Informatika, M. T., & Amikom, U. (2019). *ANALISIS PEMBOBOTAN KATA PADA KLASIFIKASI TEXT MINING*. *3*(2), 179–184.

KIKOY. (2023). *Analisis Sentimen - Unsupervised Lexical*. https://www.kaggle.com/code/rizkia14/analisis-sentimen-unsupervised-lexical

Mandasari, S., Hayadi, B. H., & Gunawan, R. (2022). Analisis Sentimen Pengguna Transportasi Online Terhadap Layanan Grab Indonesia Menggunakan Multinomial Naive Bayes Classifier. *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, *5*(2), 118. https://doi.org/10.53513/jsk.v5i2.5635

Nurjanah, T. S., & Insanudin, E. (2016). *Hack Database Website Menggunakan Python dan Sqlmap Pada Windows*.

Pratama, R., Studi, P., Informasi, T., Ilmu, F., Dan, K., Informasi, T., Muhammadiyah, U., & Utara, S. (2024). *GOOGLE PLAYSTORE DENGAN METODE NAIVE BAYES*.

Rahayu, R. (2023). *Abstrak*. *December*.

Retnosari, R. (2021). Analisa kelayakan kredit usaha mikro berjalan pada perbankan dengan metode naive bayes. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, *8*(1), 53–59.

Reza, M., I. A. Q. Maududi, M. Rifki, A. Mujaddid, F. Ikhsanudin, Y. Adharani, S. N. Ambo, & N. Rosanti. (2022). Artifical Intelligence : Image Processing &

Application with Python. *Seminar Nasional Pengabdian Masyarakat LPPM UMJ*, *1*(1), 1–8. http://jurnal.umj.ac.id/index.php/semnaskat

SaThierbach, K., Petrovic, S., Schilbach, S., Mayo, D. J., Perriches, T., Rundlet, E. J. E. J. E. J., Jeon, Y. E., Collins, L. N. L. N., Huber, F. M. F. M., Lin, D. D. H. D. H., Paduch, M., Koide, A., Lu, V. T., Fischer, J., Hurt, E., Koide, S., Kossiakoff, A. A., Hoelz, A., Hawryluk-gara, L. A., … Hoelz, A. (2015). No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title. *Proceedings of the National Academy of Sciences*, *3*(1), 1–15. http://dx.doi.org/10.1016/j.bpj.2015.06.056%0Ahttps://academic.oup.com/bioinformatics/article-abstract/34/13/2201/4852827%0Ainternal-pdf://semisupervised-3254828305/semisupervised.ppt%0Ahttp://dx.doi.org/10.1016/j.str.2013.02.005%0Ahttp://dx.doi.org/10.10

Sholih 'afif, M., Muzakir, M., Al, M. I., & Al Awalaien, G. (2021). Text Mining Untuk Mengklasifikasi Judul Berita Online Studi Kasus Radar Banjarmasin Menggunakan Metode Naïve Bayes. *Kumpulan JurnaL Ilmu Komputer (KLIK)*, *08*(2), 199–208.

Ula, M., & Fachrurrazi, S. (2023). Analisis Sentimen Cyberbullying pada Media Sosial Twitter menggunakan Metode Support Vector Machine dan Naïve Bayes Classifier. *TECHSI - Jurnal Teknik Informatika*, *14*(2), 91. https://doi.org/10.29103/techsi.v14i2.12103

Yuniar, P., & Kismiantini. (2023). Analisis Sentimen Ulasan pada Gojek Menggunakan Metode Naive Bayes. *Statistika*, *23*(2), 164–175. https://doi.org/10.29313/statistika.v23i2.2353

LAMPIRAN

MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK BAN-PT/Akred/PT/III/2019
Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
https://fikti.umsu.ac.id fikti@umsu.ac.id umsumedan umsumedan umsumedan umsumedan

Berita Acara Pembimbingan Skripsi

Nama Mahasiswa : Nazwa Putri Amanda
NPM : 2109010030
Nama Dosen Pembimbing : Dr. Firahmi Rizky S.Kom, M.Kom
Program Studi : Sistem Informasi
Konsentrasi :
Judul Penelitian :

Item	Hasil Evaluasi	Tanggal	Paraf Dosen
	Revisi Judul	14/01-2025	
	Revisi Bab 1-2	04/02-2025	
	Revisi Bab 3	19/02-2025	
	ACC Seminar Proposal	21/02-2025	
	Revisi Judul	11/03-2025	
	Revisi Bab 4	11/07-2025	
	Revisi Bab 5	12/07-2025	
	ACC SIDANG	14/07-2025	

Medan, 11 Juli 2025

Diketahui oleh :
Ketua Program Studi
Sistem Informasi

(Martiano S.Pd, S.Kom.,M.Kom)

Disetujui oleh :
Dosen Pembimbing

(Dr. Firahmi Rizky,S.Kom.,M.Kom)

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 965/II.3-AU/UMSU-09/F/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

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

Program Studi	**: Sistem Informasi**
Pada tanggal	**: 16 Desember 2024**

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama	**: Nazwa Putri Ananda**
NPM	**: 2109010030**
Semester	**: VII (Tujuh)**
Program studi	**: Sistem Informasi**
Judul Proposal / Skripsi	**: Pendeteksian Bullying Pada Layanan Gojek Melalui Analisis Sentimen Dengan Algoritma Naive Bayes**
Dosen Pembimbing	**: Dr. Firahmi Rizky, S.Kom.,M.Kom.**

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi** dinyatakan " **BATAL** " bila tidak selesai sebelum Masa Kadaluarsa tanggal : **16 Desember 2025**
4. Revisi judul……

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
Pada Tanggal : 15 Jumadil Akhir 1446 H
16 Desember 2024 M

Dekan

Dr. Al-Khowarizmi,S.Kom.,M.Kom
NIDN : 0127099201

Cc. File