IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN LOGISTIC REGRESSION DALAM MENGANALISIS SENTIMEN TANGGAPAN MASYARAKAT TERHADAP TIKTOK SHOP DI SOCIAL MEDIA

SKRIPSI

DISUSUN OLEH

RISASTI DWI ARDINI

2109020062

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

MEDAN

2025

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN LOGISTIC REGRESSION DALAM MENGANALISIS SENTIMEN TANGGAPAN MASYARAKAT TERHADAP TIKTOK SHOP DI SOCIAL MEDIA

SKRIPSI

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

DISUSUN OLEH

RISASTI DWI ARDINI
2109020062

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

LEMBAR PENGESAHAN

Judul Skripsi	:	Implementasi Algoritma Support Vector Machine (Svm) Dan Logistic Regression Dalam Menganalisis Sentimen Tanggapan Masyarakat Terhadap Tiktok Shop Di Social Media
Nama Mahasiswa	:	Risasti Dwi Ardini
NPM	:	2109020062
Program Studi	:	Teknologi Informasi

Menyetujui
Komisi Pembimbing

(Indah Purnama Sari, S.T., M.Kom.)
NIDN. 0116049001

Ketua Program Studi	Dekan
(Fatma Sari Hutagalung, S.Kom., M.Kom.)	**(Dr. Al-Khowarizmi, S.Kom., M.Kom.)**
NIDN. 0117079301	**NIDN. 0127099201**

PERNYATAAN ORISINALITAS

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN LOGISTIC REGRESSION DALAM MENGANALISIS SENTIMEN TANGGAPAN MASYARAKAT TERHADAP TIKTOK SHOP DI SOCIAL MEDIA

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

RISASTI DWI ARDINI

NPM. 2109020062

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

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

Nama	: Risasti Dwi Ardini
NPM	: 2109020062
Program Studi	: Teknologi Informasi
Karya Ilmiah	: Skripsi

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

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN LOGISTIC REGRESSION DALAM MENGANALISIS SENTIMEN TANGGAPAN MASYARAKAT TERHADAP TIKTOK SHOP DI SOCIAL MEDIA

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

Risasti Dwi Ardini

NPM. 2109020062

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Risasti Dwi Ardini
Tempat dan Tanggal Lahir	: Labuhan Haji, 7 Oktober 2002
Alamat Rumah	: Dusun II desa Perkebunan labuhan haji
Telepon/Faks/HP	: 082241613460
E-mail	: risastidwiardini7@gmail.com
Instansi Tempat Kerja	:
Alamat Kantor	:

DATA PENDIDIKAN

SD	: SD Negeri 112284 Labuhan Haji	TAMAT: 2015
SMP	: SMP Negeri 3 Kualuh Selatan	TAMAT: 2018
SMA	: SMK Negeri 2 Kualuh Selatan	TAMAT: 2021

KATA PENGANTAR

Assalamuallaikum Warahmatullahi Wabarakatuh.

Puji dan syukur atas kehadirat Allah SWT yang mana telah memberikan nikmat dan hidayah, Sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Implementasi Algoritma Support Vector Machine (SVM) dan Logistic Regression dalam Menganalisis Sentimen Tanggapan Masyarakat terhadap TikTok Shop di Media Sosial"** Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana (S1) pada program studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi di 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 Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom. selaku Ketua Program Studi Teknologi Informasi.
4. Bapak Mhd. Basri, S.Si, M.Kom selaku Sekretaris Program Studi Teknologi Informasi.

5. Ibu Indah Purnama Sari, ST.,M.Kom selaku Dosen Pembimbing yang telah dengan sabar membimbing, mengarahkan, dan memberikan masukan yang sangat berarti selama proses penyusunan skripsi ini.
6. Bapak dan Ibu dosen Program Studi Teknologi Informasi yang telah memberikan ilmu selama perkuliahan.
7. Terima kasih yang mendalam kepada kedua orang tua tercinta, Bapak Risdian dan Ibu Asmawati, yang selalu memberikan doa, kasih sayang, dukungan serta menjadi sumber semangat yang tak pernah padam. Dari ketulusan, pengorbanan, dan kesabaran dari kedua orang tua tercinta, penulis mampu menyelesaikan pendidikan ini hingga tahap akhir.
8. Kepada saudara kandung saya Mbak Wanda Putri Antana dan keluarga kecilnya serta mbah uty terima kasih telah memberikan semangat, dukungan, kasih sayang, doa kepada penulis selama penyusunan skripsi ini.
9. Terima kasih untuk Nanda Rafikhi Azhari Lubis selaku penyemangat dan rekan seperjuangan selama perkuliahan tak hanya itu tetapi juga berbagi semangat, tawa, lelah, dan mimpi. Terima kasih telah menemani setiap langkah perjalanan ini, mulai dari duduk di bangku kuliah hingga akhirnya menyelesaikan skripsi bersama.
10. Untuk Nuraini sahabat terbaik saya yang selalu hadir memberi semangat, tawa, dan dukungan selama masa perkuliahan hingga penyusunan skripsi ini. Terima kasih atas kebersamaan, bantuan, serta kata-kata penyemangat yang begitu berarti di tengah rasa lelah dan ragu.
11. Untuk Dwi Arfi Ananda dan Amira Banem terima kasih atas kebersamaan, kerja sama, saling bantu, serta semangat yang terus mengalir selama masa perkuliahan dan proses penyusunan skripsi.
12. Terima kasih kepada diri saya sendiri yang telah bertahan, terus berjuang, dan tidak menyerah meskipun menghadapi berbagai rintangan selama proses perkuliahan hingga penyusunan skripsi ini. Terima kasih telah berani melangkah, menangis, bangkit, dan tetap percaya bahwa semua usaha ini akan membawa hasil yang indah pada waktunya.

13. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Dalam penulisan skripsi ini, Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna, baik pemilihan bahasa, penjelasan, dan isi dari skripsi ini. Oleh karena itu penulis menerima kritik dan saran yang membangun demi penyempurnaan skripsi ini di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan menjadi referensi bagi peneliti selanjutnya.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Medan, 14 Juli 2025

Risasti Dwi Ardini

NPM. 2109020062

IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DAN LOGISTIC REGRESSION DALAM MENGANALISIS SENTIMEN TANGGAPAN MASYARAKAT TERHADAP TIKTOK SHOP DI SOCIAL MEDIA

ABSTRAK

Perkembangan teknologi digital telah mendorong transformasi dalam aktivitas belanja masyarakat, salah satunya melalui platform TikTok Shop. Fitur ini memungkinkan pengguna untuk berbelanja langsung dalam aplikasi TikTok, yang memunculkan beragam tanggapan dari masyarakat, baik positif maupun negatif. Penelitian ini bertujuan untuk menganalisis sentimen masyarakat terhadap TikTok Shop dengan menggunakan dua algoritma klasifikasi teks, yaitu Support Vector Machine (SVM) dan Logistic Regression. Data yang digunakan berjumlah 500 komentar dari salah satu video TikTok yang membahas TikTok Shop. Komentar tersebut diproses melalui tahapan data cleaning, normalisasi, tokenisasi, stopword removal, stemming, serta ekstraksi fitur menggunakan metode TF-IDF. Model kemudian dievaluasi menggunakan metrik akurasi, presisi, recall, dan f1-score. Hasil penelitian menunjukkan bahwa algoritma SVM memiliki performa lebih baik dibandingkan Logistic Regression, dengan akurasi sebesar 89% dan f1-score sebesar 86%, sedangkan Logistic Regression memperoleh akurasi 86% dan f1-score 80%. Oleh karena itu, SVM dinilai lebih efektif dalam mengklasifikasikan sentimen masyarakat terhadap TikTok Shop di media sosial.

Kata Kunci: Analisis Sentimen; TikTok Shop; Support Vector Machine; Logistic Regression; TF-IDF.

IMPLEMENTATION OF SUPPORT VECTOR MACHINE (SVM) AND LOGISTIC REGRESSION ALGORITHMS IN ANALYZING PUBLIC SENTIMENT TOWARD TIKTOK SHOP ON SOCIAL MEDIA

ABSTRACT

The rapid development of digital technology has transformed the way people shop, one of which is through the TikTok Shop platform. This feature allows users to shop directly within the TikTok application, generating various public responses, both positive and negative. This study aims to analyze public sentiment toward TikTok Shop using two text classification algorithms: Support Vector Machine (SVM) and Logistic Regression. The dataset consists of 500 comments collected from a TikTok video discussing TikTok Shop. The comments were processed through several stages, including data cleaning, normalization, tokenization, stopword removal, stemming, and feature extraction using the TF-IDF method. The models were evaluated using accuracy, precision, recall, and F1-score metrics. The results show that the SVM algorithm outperformed Logistic Regression with an accuracy of 89% and an F1-score of 86%, while Logistic Regression achieved an accuracy of 86% and an F1-score of 80%. Therefore, SVM is considered more effective in classifying public sentiment toward TikTok Shop on social media.

Keywords: Sentiment Analysis; TikTok Shop; Support Vector Machine; Logistic Regression; TF-IDF.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan Teknologi dan digitalisasi telah mengubah cara masyarakat dalam berbelanja dan berinteraksi di dunia maya. Dengan semakin berkembangnya teknologi digital, aktivitas belanja manusia juga mengalami perubahan. Konsumen dapat membeli produk dan layanan melalui internet tanpa harus keluar rumah dan juga memberikan kemudahan dalam melakukan transaksi karena dapat dilakukan dengan cepat dan mudah. Salah satu contoh digitalisasi budaya berbelanja yang marak di Indonesia adalah belanja online atau e-commerce. Dalam beberapa tahun terakhir, belanja online semakin berkembang pesat di Indonesia dan semakin banyak platform e-commerce yang bermunculan.

Hadirnya e-commerce memberikan manfaat terhadap para konsumen, diantaranya yaitu menghemat waktu untuk berbelanja dan cukup menggunakan platform e-commerce. Konsumen dapat membandingkan kualitas barang maupun harga di dalam platform ecommerce, hal tersebut dikarenakan dalam platform e-commerce terdapat banyak toko yang dapat dipilih. Selain itu, konsumen juga dapat membeli barang yang diinginkan. Berdasarkan hasil *We Are Social* pada April 2021, Indonesia menempati urutan pertama tingkat dunia yang menggunakan layanan e-commerce yakni sebesar 88,1% dari pengguna internet (Lidwina, 2021). Salah satu platform yang berkembang pesat dalam dunia e-commerce adalah TikTok Shop, fitur belanja dalam aplikasi TikTok yang memungkinkan pengguna untuk membeli produk secara langsung melalui video dan siaran langsung.

Dengan pertumbuhan pesat ini, muncul berbagai tanggapan dari masyarakat bahwa TikTok Shop juga mendapat kritik positif maupun negatif dari beberapa pihak yang berpendapat bahwa fitur ini memberikan kemudahan berbelanja dan promosi menarik. Sementara yang lain menyampaikan keluhan terkait dapat merugikan pelaku usaha mikro, dan menengah (UMKM) di Indonesia karena kesulitan bersaing dengan produk impor yang harganya lebih murah. Selain itu, beberapa masyarakat juga khawatir bahwa fitur ini dapat merugikan UMKM.

Penelitian ini bertujuan untuk mengetahui opini dan sentimen masyarakat terhadap TikTok Shop. Data yang digunakan dalam penelitian ini diperoleh dari komentar pada postingan video di platform TikTok yang membahas TikTok Shop. Komentar-komentar tersebut dikumpulkan untuk dianalisis guna memahami opini serta sentimen masyarakat terhadap TikTok Shop. Sentimen yang terdapat dalam komentar dapat berupa tanggapan positif, negatif. Algoritma yang digunakan dalam penelitian ini adalah Algoritma Support Vector Machine (SVM) dan Logistic Regression merupakan dua pendekatan yang sering digunakan dalam klasifikasi teks karena efektivitasnya dalam membedakan sentimen positif dan negatif.

Dalam menganalisis sentimen tersebut algoritma Support Vector Machine (SVM) sebagai salah satu metode klasifikasi yang efektif dalam analisis sentimen, khususnya dalam mengelompokkan opini masyarakat sehingga sering memberikan hasil akurasi yang tinggi dalam analisis sentimen. Cara kerjanya dengan mencari garis pemisah terbaik (hyperplane) yang memaksimalkan jarak antar kelas data antara sentimen positif dan negatif. Pada Algoritma Logistic Regression dapat digunakan untuk analisis sentimen dengan memprediksi probabilitas suatu teks

memiliki sentimen positif atau negatif. Cara kerjanya menggunakan kombinasikan dengan fungsi sigmoid untuk menghasilkan output diskrit (0 atau 1) yang mewakili sentimen.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk menerapkan algoritma Support Vector Machine (SVM) dan Logistic Regression dalam menganalisis sentimen tanggapan masyarakat terhadap TikTok Shop di social media. Penelitian ini juga akan mengevaluasi performa kedua algoritma untuk menentukan metode yang lebih akurat dalam klasifikasi sentimen. Dengan adanya penelitian ini, diharapkan dapat memberikan wawasan bagi pelaku bisnis, pengembang platform, dan konsumen dalam memahami tren opini publik serta meningkatkan pengalaman pengguna dalam berbelanja di TikTok Shop.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana sentimen masyarakat terhadap TikTok Shop berdasarkan opini yang diperoleh dari social media TikTok?
2. Bagaimana proses implementasi algoritma Support Vector Machine (SVM) dan Logistic Regression dalam menganalisis sentimen komentar masyarakat terhadap TikTok Shop?
3. Seberapa efektif metode Support Vector Machine (SVM) dan Logistic Regression dalam menganalisis sentimen Masyarakat terhadap TikTok Shop?

1.3. Batasan Masalah

Batasan masalah yang akan dibahas dalam penelitian ini adalah sebagai berikut:

1. Data yang digunakan dalam penelitian ini hanya diperoleh dari social media TikTok, dengan sumber data berupa 500 komentar yang diambil dari satu video yang membahas TikTok Shop.
2. Analisis sentimen dalam penelitian ini akan dilakukan menggunakan dua metode yaitu Support Vector Machine (SVM) dan Logistic Regression.
3. Penelitian pada sentimen yang dianalisis dibatasi dalam kategori positif dan negatif.
4. Bahasa Pemrograman yang digunakan dalam mengelolah data adalah phyton

1.4. Tujuan Penelitian

Penelitian ini bertujuan untuk :

1. Menganalisis sentimen masyarakat terhadap TikTok Shop berdasarkan opini yang terdapat dalam komentar di social media TikTok.
2. Mengevaluasi hasil penerapan metode Support Vector Machine (SVM) dan Logistic Regression dalam mengklasifikasikan sentiment masyarakat terhadap TikTok Shop.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebaga berikut:

1. Dengan menganalisis sentimen masyarakat terhadap TikTok Shop di social media menggunakan algoritma Support Vector Machine (SVM) dan Logistic Regression, penelitian ini dapat memberikan respon positif, dan negatif dari pengguna tiktok shop.
2. Dengan membandingkan kinerja kedua algoritma ini, penelitian dapat menentukan metode mana yang lebih akurat dan efisien dalam mengklasifikasikan sentimen masyarakat terhadap TikTok Shop. Hasil ini dapat menjadi referensi bagi penelitian selanjutnya dalam memilih algoritma yang tepat untuk analisis sentimen.

BAB II

LANDASAN TEORI

2.1 Social Media

Social Media adalah sebuah media online, dengan para penggunanya bisa dengan mudah berpartisipasi, berbagi, dan menciptakan isi meliputi blog, jejaring sosial, wiki, forum dan dunia virtual. Blog, jejaring sosial dan wiki merupakan bentuk media sosial yang paling umum digunakan oleh masyarakat di seluruh dunia. Pendapat lain mengatakan bahwa social media adalah media online yang mendukung interaksi sosial dan media sosial menggunakan teknologi berbasis web yang mengubah komunikasi menjadi dialog interaktif (Liedfray et al., 2022).

Menurut (Zunianantа, 2021) Social media merupakan sebuah teknologi web berbasis internet yang memudahkan semua penggunanya untuk berkomunikasi, berbagi informasi dan membentuk sebuah kelompok secara virtual, sehingga dapat menyebarluaskan isi konten yang mereka hasilkan. Melalui media sosial, pengguna dapat berinteraksi secara interaktif dengan pengguna lain yang ada di seluruh dunia untuk berbagi informasi dan berkomunikasi. Fungsi utama dari social media yaitu agar pengguna dapat berkomunikasi dengan pengguna lain secara mudah dan efektif.

2.2 TikTok Shop

TikTok Shop adalah salah satu fitur yang diperkenalkan oleh aplikasi TikTok adalah TikTok Shop. Fitur ini memungkinkan pengguna untuk berbelanja di dalam aplikasi social media tanpa perlu berpindah aplikasi. TikTok meluncurkan fitur

terbarunya TikTok Shop, sebagai tanggapan atas popularitas global aplikasi tersebut. TikTok Shop menggabungkan media sosial dengan pasar. TikTok Shop adalah fitur aplikasi TikTok yang memudahkan pelaku bisnis dan penggunanya untuk menjual dan membeli produk. Dengan fitur tersebut, pembeli dapat dengan mudah melakukan pembelian di dalam platform media sosial TikTok tanpa harus beralih ke aplikasi belanja lainnya (Simanjuntak & Sari, 2023).

Menurut (Supriyanto et al., 2023) Tiktok Shop merupakan suatu e-commerce yang dapat dianggap sebagai sistem informasi bisnis karena penjualannya dilakukan melalui media elektronik yang menyediakan informasi khusus mengenai jual beli atau bisnis. Selain itu, Tiktok Shop juga menyediakan layanan yang mirip dengan marketplace dan e-commerce pada umumnya. Yang membedakan Tiktok Shop adalah harga yang sangat terjangkau, jauh lebih rendah daripada marketplace atau e-commerce lainnya.

2.3 Analisis Sentimen

Analisis sentimen adalah salah satu proses menganalisis teks digital yang digunakan untuk menentukan apakah kata-kata atau kalimat yang disampaikan memiliki makna atau emosional pesan. Analisis sentiment bertujuan untuk menentukan apakah suatu teks mengandung sentimen positif, negatif, atau netral terhadap suatu topik (Yulia Kurniawati, 2023).

Gambar 2.1 Jenis Ekspresi Analisis Sentimen

Menurut penelitian (Saif M. Mohammad, 2021). Analisis sentimen adalah istilah umum untuk penentuan valensi, emosi, dan keadaan affektual lainnya dari teksatau ucapan secara otomatis menggunakan algoritma komputer. Paling umumnya, ini digunakan untuk merujuk pada tugas menentukan valensi suatu bagian secara otomatis teks, baik positif, negatif, atau netral, peringkat bintang suatu produk atau film ulasan, atau skor bernilai riil dalam rentang 0 hingga 1 yang menunjukkan tingkat kepositifan suatu sepotong teks.

2.4 Support Vector Machine (SVM)

Vapnik pada tahun 1992, memperkenalkan Support Vector Machine (SVM) sebagai model Machine Learning multifungsi yang dapat digunakan untuk menyelesaikan permasalahan klasifikasi, regresi, dan pendeteksian *outlier*. SVM adalah salah satu metode yang paling populer dalam machine learning (Dadan Dahman W., 2021). Support Vector Machine (SVM) memiliki keunggulan dalam memisahkan data sentimen positif dan negatif dan menghasilkan akurasi yang tinggi dalam klasifikasi sentiment (Irma Surya Kumala Idris et al., 2023)**.**

Menurut (Anshul, 2025) Cara kerja SVM didasarkan pada SRM atau Structural Risk Minimization yang dirancang untuk mengolah data menjadi Hyperplane yang mengklasifikasikan ruang input menjadi dua kelas. Support Vector Machine (SVM) diawali dengan pengelompokan kasus-kasus linier yang

dapat dipisahkan dengan hyperplane dan dibagi menurut kelasnya. Untuk kasus klasifikasi biner dengan dua fitur, rumus matematis dari hyperplane adalah (RB Fajriya Hakim, 2024):

$$w^T x + b = 0$$

Dimana:

w adalah vektor bobot (weight)

x adalah vector fitur input

b adalah bias

w^T adalah transpos dari vector bobot *w*

Konsep SVM diawali dengan masalah klasifikasi dua kelas sehingga membutuhkan set pelatihan positif dan negatif. SVM akan berusaha mendapatkan hyperplane (pemisah) sebaik mungkin untuk memisahkan kedua kelas dan memaksimalkan margin kedua kelas tersebut (Kendrew Huang, 2022).

2.5 Logistic Regression

Logistic Regression adalah algoritma pembelajaran mesin terbimbing yang menyelesaikan tugas klasifikasi biner dengan memprediksi probabilitas suatu hasil, peristiwa, atau pengamatan . Model tersebut memberikan hasil biner atau dikotomis yang dibatasi pada dua kemungkinan hasil: 0/1, ya/tidak, atau benar/salah (Vijay Kanade, 2022).

Menurut (Junifer Pangaribuan et al., 2021) Logistic Regression adalah analisis regresi yang tepat untuk dilakukan ketika variabel dependen adalah biner (dua kemungkinan). Logistic Regression digunakan untuk menggambarkan data dan untuk menjelaskan hubungan antara satu variabel biner dependen dan satu atau lebih variabel independen nominal, ordinal, interval atau rasio tingkat.

Logistic Regression dinyatakan dalam Persamaan berikut ini:

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

B_0 = Konstanta

B_1 = Koefisien dari masing -masing variable

Nilai p atau peluang (Y = 1) dapat dicari dengan persamaan berikut:

$$p = \frac{e^{(B_0+B_1X)}}{(1+e^{(B_0+B_1X)})}$$

Persamaan model yang digunakan untuk logistik regresi yang dapat disebut juga fungsi sigmoid adalah sebagai berikut:

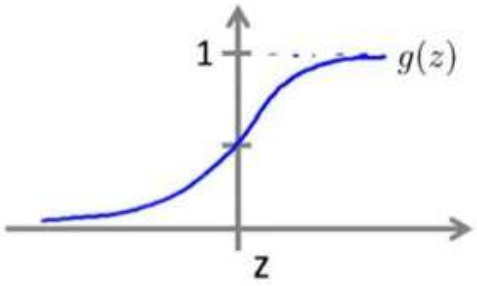
$$h_\theta = g(\theta^T x)$$

$$z = \theta^T x$$

$$g(z) = \frac{1}{1+e^{-z}}$$

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$h_\theta\,(x)$akan memberikan hasil diantara 0 dan 1, contohnya $h_\theta(x)$ memberikan nilai 0.7 dimana memberikan hasil probabilitas 70% terhadap output dari 1 dan untuk probabilitas hasil 0 adalah 30%. Fungsi dengan kurva yang berbentuk huruf S. Untuk setiap nilai x yang dipetakan ke dalam interval 0 sampai 1 dinamakan fungsi sigmoid biner, sedangkan output yang memiliki rentang antara -1 sampai dengan 1 disebut sigmoid tan. Berikut gambar 2.5 adalah gambar dari persamaan sigmoid yang berbentuk huruf S.



Gambar 2.2 Sigmoid

Misalnya, untuk memprediksi

y = 1, maka nilai dari $h_\theta(x) \geq 0.5$, dimana nilai dari $z \geq 0$

y = 0, maka nilai dari $h_\theta(x) < 0.5$, nilai dari $z < 0$,

Dari penjelasan di atas, dapat disimpulkan bahwa ketika memprediksi hasil dari sebuah persamaan, 0 ataupun 1, sama seperti memprediksi y = 1 ketika nilai dari $\theta^T x \geq 0$dan sebaliknya memprediksi nilai y = 0, $\theta^T x < 0$.

2.6 Confusion Matrix

Menurut (Dr.Maria Susan Anggreany, 2020) *Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Berikut ini keterangan pada *Confusion matrix* yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah:

True Positif (TP) = Perbandingan sampel bernilai true (benar) yang diprediksi secara benar.

True Negative (TN) = Perbandingan sampel bernilai false (salah) yang diprediksi secara benar.

False Positif (FP) = Perbandingan sampel bernilai false (salah) yang salah diprediksi sebagai sampel bernilai benar.

False Negative (FN) = Perbandingan sampel bernilai true (benar) yang salah diprediksi sebagai sampel bernilai salah.

Rumus confusion matrix untuk menghitung accuracy, precision, recall dan F-1 Score. Nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Accuracy menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. Berikut ini merupakan rumus perhitungan confusion matrix (Ferry Putrawansyah* dan Tri Susanti, 2024).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \text{ X } 100\%$$

$$Precision = \frac{TP}{TP + FP} \text{ X } 100\%$$

$$Recall = \frac{TP}{TP + FN} \text{ X } 100\%$$

$$F - 1\ Score = \frac{2 \text{ x } Recall \text{ x } Precision}{Recall + Precision}$$

2.7 Machine Learning

Machine Learning atau Pembelajaran Mesin adalah teknik pendekatan dari Artificial Intelligent (AI) yang digunakan untuk meniru untuk menggantikan peran manusia dalam melakukan aktivitas untuk memecahkan masalah. Singkatnya, Machine Learning adalah sebuah mesin yang dibuat untuk dapat belajar dan melakukan pekerjaan tanpa arahan dari penggunanya. Menurut Arthur Samuel, seorang perintis Amerika di bidang permainan komputer dan kecerdasan buatan, AI menyatakan bahwa pembelajaran mesin adalah cabang ilmu yang mempelajari

bagaimana memberi komputer kemampuan untuk belajar tanpa diprogram secara eksplisit (Wijoyo A et al., 2024).

Machine learning, telah menjadi pendorong utama inovasi teknologi di berbagai bidang. Teknologi ini memungkinkan sistem untuk mempelajari pola dan struktur dari data, yang kemudian digunakan untuk membuat prediksi atau keputusan secara otomatis.

Menurut (Pratama et al., 2025) Machine Learning dapat dibagi menjadi tiga kategori utama: supervised learning, unsupervised learning, dan reinforcement learning. Dalam supervised learning, model dilatih menggunakan dataset yang sudah diberi label, di mana input dan output diketahui. Contoh algoritma yang sering digunakan adalah regresi linier, regresi logistik, dan neural network. Teknologi ini banyak digunakan dalam aplikasi seperti prediksi pasar saham, klasifikasi email spam, dan diagnosis penyakit. Sebaliknya, unsupervised learning berfokus pada data tanpa label dan digunakan untuk mengidentifikasi pola tersembunyi. Sedangkan Reinforcement learning ini telah digunakan dalam aplikasi seperti pengendalian robotika dan pengembangan algoritma permainan seperti AlphaGo.

2.8 Google Colab

Google Colab atau Google Colaboratory, adalah sebuah executable document yang dapat digunakan untuk menyimpan, menulis, serta membagikan program yang telah ditulis melalui Google Drive. Google Colab sering digunakan karena memungkinkan pengguna untuk menjalankan kode Python tanpa perlu melakukan proses instalasi. Google Colab sebagai aplikasi yang diciptakan khusus operasi machine learning dan deep learning (Oliver, 2022).

Gambar 2.3 Logo Google Colab

Menurut (Isnanto, 2023) Melalui Google Colab Pengguna dapat mengakses GPU (Graphical Processing Unit) dan TPU (Tensor Processing Unit) secara gratis pada google colab, sehingga memungkinkan untuk melakukan pelatihan model machine learning yang rumit dan membutuhkan daya komputasi tinggi. Pengguna dapat mengimpor dan menginstall berbagai pustaka Python populer dengan mudah, misalnya TensorFlow, PyTorch, Pandas. Pengguna juga bisa mengakses sumber daya tambahan, seperti dataset dan model yang sudah tersedia.

2.9 Python

Python adalah bahasa pemrograman tingkat tinggi yang pertama kali dikembangkan oleh Guido van Rossum pada akhir 1980-an. Python didesain dengan fokus pada keterbacaan kode, sehingga mudah dipelajari dan digunakan oleh pemula maupun ahli. Fleksibilitasnya memungkinkan Python digunakan dalam berbagai aplikasi, mulai dari pengembangan web hingga analisis data. Python termasuk dalam kategori bahasa pemrograman yang mendukung berbagai gaya pemrograman seperti pemrograman fungsional, berorientasi objek, dan imperatif. Hal ini menjadikan Python sebagai bahasa yang sangat fleksibel untuk berbagai kebutuhan. Python digunakan dalam berbagai bidang mulai dari

pengembangan web, data science, hingga otomasi. Keberagaman pustaka yang tersedia membuat Python menjadi bahasa yang serbaguna (Muhammad Thoriq Al Fatih, 2024).

2.10 Tiktok Comment Scraper

Menurut (Storm_Scraper, n.d.) Tiktok Comment Scraper merupakan sebuah tools web scraping yang berjalan melalui situs Apify.com, digunakan untuk mengambil data komentar secara otomatis dari TikTok: teks komentar, ID pengguna, stempel waktu, jumlah balasan dan konten balasan, jumlah suka, dll. Untuk mendapatkan komentar TikTok, cukup masukkan URL video atau nama pengguna TikTok dan klik tombol Simpan & Mulai.

2.11 Penelitian Terdahulu

Penelitian terdahulu yang dilakukan oleh (Lidinillah et al., 2023) membahas tentang "analisis Sentimen Twitter terhadap Steam Menggunakan Algoritma Logistic Regression dan Support Vector Machine". Tujuannya adalah untuk mengetahui opini publik terhadap platform digital Steam melalui analisis sentimen pada social media Twitter. Penelitian ini menggunakan dua algoritma klasifikasi teks, yaitu Logistic Regression dan Support Vector Machine (SVM), untuk membandingkan efektivitas masing-masing dalam mengkategorikan sentimen menjadi dua kelas, yaitu sentimen positif dan negatif.

Hasil dari penelitian menunjukkan bahwa algoritma Logistic Regression memberikan performa yang lebih baik dibandingkan dengan SVM. Logistic Regression berhasil mencapai akurasi sebesar 87,5%, sedangkan SVM hanya mencapai 85%. Perbedaan ini menunjukkan bahwa Logistic Regression lebih

efektif dalam menangkap pola sentimen pada dataset yang digunakan dalam penelitian. Dengan demikian, dapat disimpulkan bahwa dalam konteks analisis sentimen terhadap Steam di Twitter, Logistic Regression merupakan metode yang lebih unggul dibandingkan dengan SVM.

Penelitian yang dilakukan oleh (Novantika, 2022) membahas tentang "*Analisis Sentimen Ulasan Pengguna Aplikasi Video Conference Google Meet menggunakan Metode SVM dan Logistic Regression",* diperoleh bahwa kedua metode klasifikasi, yaitu Support Vector Machine (SVM) dan Logistic Regression, mampu mengklasifikasikan sentimen ulasan pengguna secara cukup akurat. Metode Logistic Regression memberikan hasil dengan akurasi sebesar 85%, precision sebesar 0,87, recall sebesar 0,85, dan F1-score sebesar 0,86. Sementara itu, metode Support Vector Machine (SVM) menunjukkan performa yang lebih unggul, dengan akurasi mencapai 88%, precision sebesar 0,89, recall sebesar 0,88, dan F1-score sebesar 0,88. Berdasarkan perbandingan metrik evaluasi tersebut, dapat disimpulkan bahwa SVM lebih efektif dalam mengklasifikasikan sentimen ulasan pengguna terhadap aplikasi Google Meet dibandingkan dengan Logistic Regression. Hal ini mengindikasikan bahwa SVM mampu menangkap pola-pola dalam data teks dengan lebih baik, sehingga lebih akurat dalam memisahkan sentimen positif dan negatif.

Pada penelitian yang dilakukan oleh (Amalia et al., 2024) membahas tentang "*Analisis Sentimen Kebijakan Penyelenggara Sistem Elektronik Lingkup Privat Menggunakan Penalized Logistic Regression dan Support Vector Machine"*. Tujuannya adalah untuk mengklasifikasikan sentimen masyarakat terhadap kebijakan PSE lingkup privat dengan menggunakan dua metode pembelajaran

mesin, yaitu Penalized Logistic Regression (PLR) dan Support Vector Machine (SVM).

Hasil evaluasi menunjukkan bahwa metode SVM unggul dibandingkan PLR. SVM berhasil mencapai akurasi sebesar 82,60%, presisi 82,92%, recall 82,60%, dan F1-score sebesar 82,66%. Sementara itu, PLR mencatatkan akurasi sebesar 78,20%, presisi 78,33%, recall 78,20%, dan F1-score sebesar 78,23%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa SVM lebih efektif dalam mengklasifikasikan sentimen teks dengan fitur hasil ekstraksi TF-IDF dibandingkan dengan PLR, dan lebih mampu memisahkan data sentimen secara optimal. Penelitian ini memberikan kontribusi dalam pemanfaatan machine learning untuk analisis opini publik di social media, khususnya terkait isu kebijakan.

Pada penelitian yang dilakukan oleh (Simanjuntak & Sari, 2023) yang membahas *"Analisis Perbandingan Sentimen Corona Virus Disease-2019 (COVID-19) pada Twitter Menggunakan Metode Logistic Regression dan Support Vector Machine (SVM)"*. Tujuannya adalah untuk membandingkan kinerja dua algoritma klasifikasi populer, yaitu Logistic Regression dan Support Vector Machine (SVM), dalam menganalisis sentimen masyarakat terhadap isu COVID-19 berdasarkan data yang diperoleh dari platform Twitter.

Hasil dari eksperimen menunjukkan bahwa algoritma Support Vector Machine (SVM) menghasilkan akurasi yang lebih tinggi dibandingkan dengan Logistic Regression. Akurasi SVM mencapai 88,32%, sedangkan Logistic Regression mencatatkan akurasi sebesar 85,12%. Selain itu, nilai precision, recall, dan f1-score pada model SVM juga menunjukkan kinerja yang lebih baik secara

keseluruhan. Berdasarkan temuan ini, dapat disimpulkan bahwa algoritma SVM lebih efektif dalam menganalisis dan mengklasifikasikan sentimen pada data teks social media, khususnya dalam konteks isu pandemi COVID-19.

Tabel 2.1 penelitian Terdahulu

No	Judul	Metode	Hasil
1	Analisis Sentimen Twitter terhadap Steam Menggunakan Algoritma Logistic Regression dan Support Vector Machine	Logistic Regression dan Support Vector Machine	Metode Logistic Regression memberikan performa yang lebih baik dibandingkan dengan SVM. Logistic Regression berhasil mencapai akurasi sebesar 87,5%, sedangkan SVM hanya mencapai 85%.
2	Analisis Sentimen Ulasan Pengguna Aplikasi Video Conference Google Meet menggunakan Metode SVM dan Logistic Regression	Support Vector Machine dan Logistic Regression	SVM lebih efektif dalam mengklasifikasikan sentimen ulasan pengguna terhadap aplikasi Google Meet dibandingkan dengan Logistic Regression. Metode SVM mampu menangkap pola-pola dalam data teks dengan lebih baik, sehingga lebih akurat dalam memisahkan sentimen positif dan negatif.

3	Analisis Sentimen Kebijakan Penyelenggara Sistem Elektronik Lingkup Privat Menggunakan Penalized Logistic Regression dan Support Vector Machine	Logistic Regression dan Support Vector Machine	Metode SVM lebih unggul dibandingkan Logistic Regression. Metode SVM lebih efektif dalam mengklasifikasikan sentimen teks dengan fitur hasil ekstraksi TF-IDF dibandingkan dengan PLR, dan lebih mampu memisahkan data sentimen secara optimal.
4	Analisis Perbandingan Sentimen Corona Virus Disease-2019 (COVID-19) pada Twitter Menggunakan Metode Logistic Regression dan Support Vector Machine (SVM)	Logistic Regression dan Support Vector Machine (SVM)	Support Vector Machine (SVM) menghasilkan akurasi yang lebih tinggi dibandingkan dengan Logistic Regression. Akurasi SVM mencapai 88,32%, sedangkan Logistic Regression mencatatkan akurasi sebesar 85,12%.

BAB III

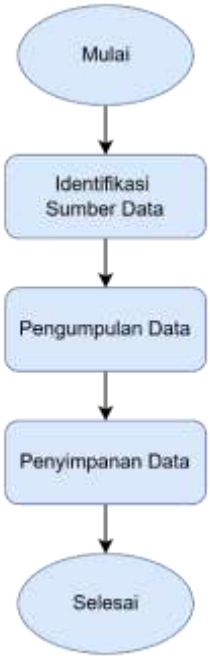
METODOLOGI PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini menggunakan metode Support Vector Machine (SVM) dan Logistic Regression dalam mengklasifikasikan sentimen tanggapan masyarakat terhadap tiktok shop di social media. Metode ini dipilih karena sesuai untuk mengukur dan menganalisis data, Hasil akhir dari penelitian ini mampu memberikan gambaran yang akurat tentang persepsi publik terhadap TikTok Shop di social media.

3.2. Tahap Penelitian

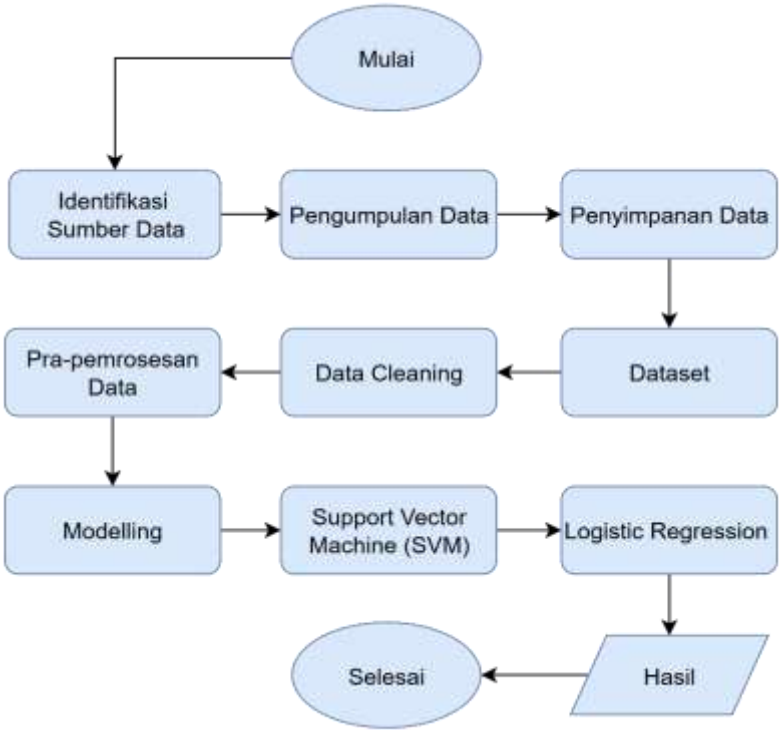
Berikut merupakan alur dari penelitian yang berjudul “Implementasi Algoritma Support Vector Machine (Svm) Dan Logistic Regression Dalam Menganalisis Sentimen Tanggapan Masyarakat Terhadap Tiktok Shop.



Gambar 3.1 Flowchart Alur Penelitian

Pada Gambar 3.1 flowchart di atas menggambarkan alur awal dari proses pengolahan data dalam penelitian analisis sentimen terhadap TikTok Shop. Proses dimulai dari simbol "Mulai", yang menandakan bahwa sistem atau alur kerja akan segera dijalankan. Langkah pertama adalah Identifikasi Sumber Data, dimana data akan diperoleh dari komentar atau opini pengguna social media TikTok yang berkaitan dengan TikTok Shop.

Selanjutnya ke tahap Pengumpulan Data, untuk proses pengumpulan data menggunakan metode web scraping. Data yang telah dikumpulkan kemudian masuk ke tahap Penyimpanan Data, data tersebut disimpan ke dalam format CSV. Terakhir, tahap ini ditutup dengan simbol "Selesai", yang menandai bahwa tahap awal proses data telah selesai dan siap dilanjutkan ke tahap berikutnya seperti pra-pemrosesan data.



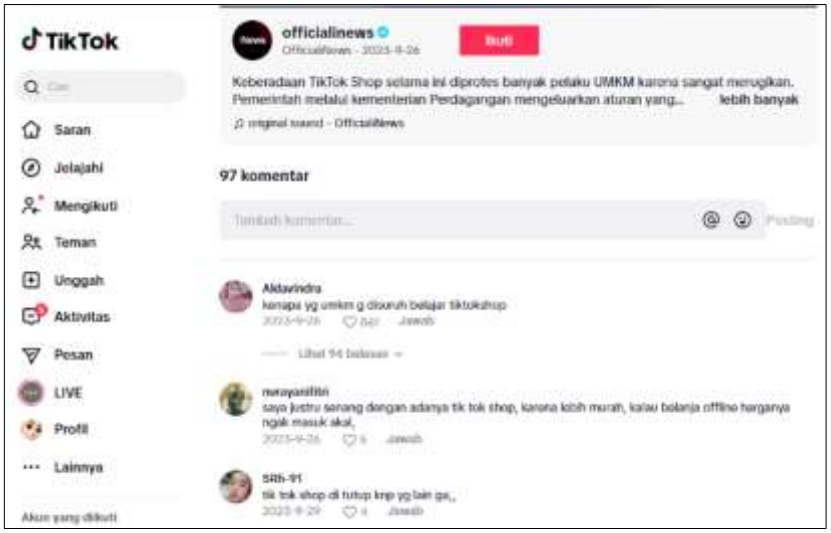
Gambar 3.5 Flowchart Sistem penelitian

Pada Gambar 3.2 di atas menunjukkan alur lengkap penelitian dalam menganalisis sentimen tanggapan masyarakat terhadap TikTok Shop menggunakan algoritma Support Vector Machine (SVM) dan Logistic Regression. Proses dimulai dari simbol "Mulai", yang menandai awal penelitian. Langkah pertama adalah Identifikasi Sumber Data menentukan platform yang akan dijadikan sumber, yaitu komentar-komentar pada social media TikTok. Selanjutnya, dilakukan Pengumpulan Data menggunakan teknik seperti web scraping, lalu data yang diperoleh disimpan pada tahap Penyimpanan Data ke dalam format CSV.

Setelah data terkumpul, selanjutnya masuk ketahap import dataset dilakukan Data Cleaning untuk menghapus karakter yang tidak diperlukan. Langkah berikutnya adalah Pra-pemrosesan Data, kemudian, masuk ke tahap Modelling, yaitu proses pelatihan dan pengujian data menggunakan dua algoritma, yaitu Support Vector Machine (SVM) dan Logistic Regression. Terakhir, proses ditutup dengan simbol "Selesai", yang menunjukkan akhir dari alur penelitian.

3.3. Identifikasi Sumber Data

Dalam penelitian ini, sumber data yang diperoleh dari komentar video di platform social media TikTok yang membahas TikTok Shop. Komentar-komentar ini mencerminkan opini dan sentimen masyarakat terhadap TikTok Shop, baik dalam bentuk tanggapan positif dan negatif.



Gambar 3.8 Tampilan Komentar dari platform social media

Gambar 3.3 mengambarkan tampilan komentar dari video pada platform social media TikTok yang membahas TikTok Shop. Dalam komentar tersebut terlihat beragam tanggapan dari masyarakat, mengenai dampaknya terhadap pengalaman berbelanja online. Data komentar seperti ini digunakan sebagai sumber utama dalam proses analisis sentimen pada penelitian ini.

3.4. Pengumpulan Data

Pengumpulan data yang digunakan dalam penelitian ini adalah web scraping, yaitu pengambilan data secara otomatis karena mempermudah proses ekstraksi data secara cepat, dan efisien dibandingkan dengan pengumpulan data secara manual. Pada penelitian ini, web scraping digunakan untuk mengumpulkan komentar dari platform TikTok yang membahas TikTok Shop. Komentar-komentar tersebut kemudian digunakan sebagai data utama dalam analisis sentimen terhadap tanggapan masyarakat terhadap TikTok Shop.

Tools yang digunakan dalam proses web scraping ini adalah Apify TikTok Comment Scraper, yaitu sebuah platform berbasis cloud yang menyediakan

berbagai actor (skrip otomatis) untuk mengekstraksi data dari situs web, termasuk TikTok. Apify TikTok Comment Scraper mempermudah penulis untuk mengumpulkan komentar dari video TikTok tanpa harus melakukan penyalinan secara manual satu per satu.

3.5. Penyimpanan Data

Data yang telah dikumpulkan melalui proses web scraping disimpan dalam format CSV atau Excel, yang berjumlah 5 kolom yang terdiri dari: Text, CreateTime, UniqueId, VideoWebUrl dan Sentimen. Penggunaan teknik web scraping ini telah banyak diterapkan dalam penelitian untuk mengumpulkan data dari social media secara efektif.

3.6. Dataset

Dataset yang disimpan berjumlah 500 komentar yang berkaitan dengan TikTok Shop, di mana masing-masing komentar mencerminkan opini positif dan negatif pengguna. Dataset yang telah disimpan kemudian diimpor ke dalam Google Colab menggunakan pustaka pandas.

3.7. Data Cleaning

Tahap selanjutnya setelah penyimpanan data dan import data adalah tahap *data cleaning dan normalisasi* atau pembersihan data bertujuan untuk menghapus, memperbaiki, atau memodifikasi data yang tidak lengkap, tidak akurat, atau terduplikasi dari kumpulan data. Seperti penghapusan simbol, angka, mention, tanda baca, spasi berlebihan, huruf berulang lebih dari 2 jadi 1 dan dan mengubah huruf kapital menjadi huruf kecil semua. Proses pembersihan data dilakukan dengan menggunakan bahasa pemrograman Python melalui platform Google

Colab. Data yang telah melewati tahap *cleaning* siap untuk ketahap pra-pemprosesan data.

3.8. Teknik Pra-pemrosesan Data

Dalam proses analisis sentimen terhadap TikTok Shop, tahap pra-pemrosesan dilakukan menggunakan platform Google Colab dengan bahasa pemrograman Python untuk pra-pemrosesan data teks.

Tahapan pertama adalah melakukan tokenisasi, yaitu pemisahan kalimat menjadi kata-kata terpisah agar dapat dianalisis satu per satu. Kata-kata tersebut lalu melalui proses Stopwords Removal, Stemming untuk mengembalikan ke bentuk dasarnya, seperti “menjual” atau “penjualan” yang diubah menjadi “jual”, Label Encoding. Dengan pra-pemrosesan data, algoritma SVM dan Logistic Regression dapat bekerja lebih efektif dalam memahami dan mengklasifikasikan sentimen tanggapan masyarakat terhadap TikTok Shop.

3.9. Modeling

Pada tahap ini, penulis melakukan proses modeling untuk membangun sistem klasifikasi sentimen berdasarkan komentar masyarakat yang telah melalui tahap data cleaning dan pra-pemrosesan. Sebelum dilakukan pelatihan model, data dibagi menjadi dua bagian, yaitu data latih (training data) dan data uji (testing data) dengan rasio 80:20. Selanjutnya, dilakukan proses ekstraksi fitur menggunakan metode TF-IDF (Term Frequency-Inverse Document Frequency) untuk mengubah data teks menjadi bentuk numerik yang dapat dikenali oleh algoritma. Model kemudian dilatih menggunakan data latih, dan hasil prediksi dari masing-masing model akan dievaluasi menggunakan data uji untuk menilai performa klasifikasi dari kedua

algoritma. Modeling dilakukan dengan menggunakan dua algoritma machine learning, yaitu Support Vector Machine (SVM) dan Logistic Regression. Kedua algoritma ini dipilih karena dikenal efektif dalam menangani permasalahan klasifikasi teks, khususnya dalam analisis sentimen.

Langkah selanjutnya melakukan evaluasi menggunakan metrik seperti akurasi, presisi, recall, F1-score confusion matrix untuk menilai seberapa baik model dapat mengklasifikasikan sentimen. Hasil dari kedua model dibandingkan untuk mengetahui metode mana yang memberikan kinerja terbaik dalam konteks data sentimen mengenai TikTok Shop. Proses ini dilakukan diplatform Google Colab, menggunakan pustaka Python seperti scikit-learn, pandas, numpy, dan matplotlib, sehingga memudahkan implementasi, visualisasi, dan pengujian model secara efisien.

3.10. Perangkat Penelitian

Perangkat Penelitian merupakan komponen penting yang mendukung kelancaran seluruh proses penelitian, karena digunakan untuk pengumpulan data dan analisis data. Perangkat penelitian yang digunakan dalam penelitian ini meliputi perangkat keras (*hardware)* dan perangkat lunak (*software)* yang dipilih sesuai dengan kebutuhan penelitian. Berikut adalah tabel perangkat keras dan perangkat lunak yang digunakan dalam penelitian:

Tabel 3.1 Kebutuhan Perangkat Keras

No	Perangkat Keras	Deskripsi
1	Laptop	Aspire A314-22
2	Processor	AMD 3020e with Radeon Graphics
3	Random Access Memory (RAM)	4,00 GB
4	System Type	64-bit

Tabel 3.2 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Deskripsi
1	Windows 11	Sistem operasi
2	Google Colab	Platform pemrograman Python dan eksekusi model Machine Learning
3	Python	Bahasa pemrograman utama untuk pengolahan data
4	Google Chrome	Membantu menyediakan tools untuk melakukan scraping data
5	Apify Tiktok Comment Scraper	Tools yang digunakan untuk Pengambilan komentar dari TikTok

3.11. Jadwal Penelitian

Adapun penelitian ini dilaksanakan dari bulan januari 2025 – Juli 2025 dan dapat dilihat pada tabel 3.3:

Tabel 3.3 Jadwal Penelitian

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

BAB IV

HASIL DAN PEMBAHASAN

4.1 Gambaran Umum

Data yang digunakan dalam penelitian ini berjumlah 500 komentar yang diambil dari media sosial TikTok, khususnya pada video yang membahas tentang TikTok Shop. Komentar-komentar ini dipilih karena mengandung opini positif dan negatif dari pengguna terhadap fitur belanja tersebut. Data ini disimpan dalam format CSV, yang berjumlah 5 kolom yang terdiri dari: Text, CreateTime, UniqueId, VideoWebUrl dan Sentimen.

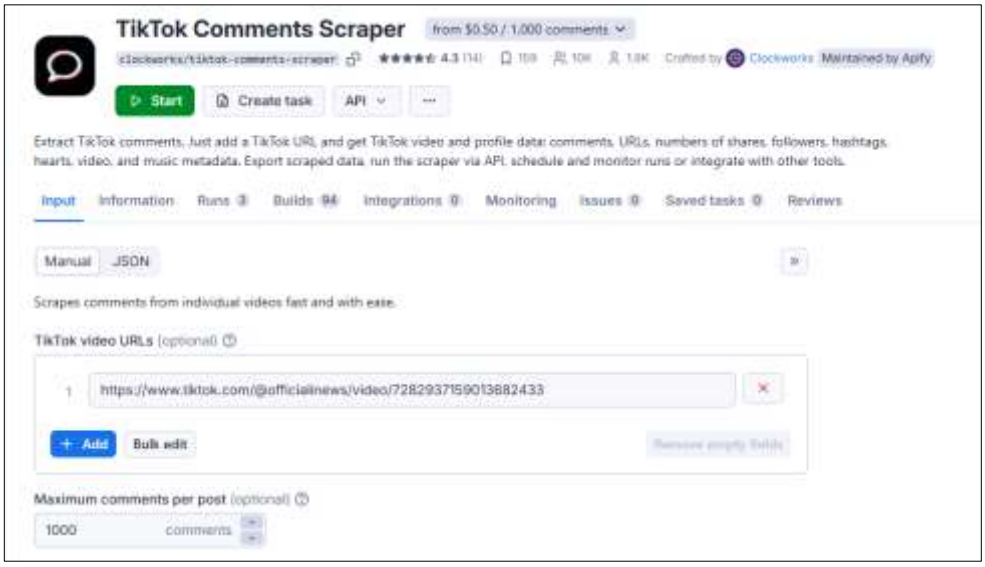
Dari total 500 data komentar yang telah dikumpulkan sebanyak 420 komentar termasuk dalam kategori sentimen positif dan 80 komentar termasuk dalam kategori sentimen negatif.

4.2 Pengumpulan Data

Langkah awal yang dilakukan dalam penelitian ini adalah proses pengambilan data komentar (scraping) dari media sosial TikTok. Pengambilan data dilakukan dengan menggunakan tools Apify TikTok Comment Scraper, yaitu alat berbasis web scraping yang dirancang untuk mengekstrak komentar dari sebuah video TikTok secara otomatis dan efisien. Berikut tahapan-tahapan scraping dalam pengumpulan data:

1. Langkah pertama adalah penulis mendaftar atau login ke platform Apify melalui situs https://apify.com. Setelah masuk, penulis akan diarahkan ke dashboard utama untuk memilih atau menjalankan aktor scraping.

2. Pada kolom pencarian, ketik TikTok Comment Scraper, lalu masukkan URL video TikTok yang akan diambil komentarnya ke dalam field input, tampilan input dapat ditunjukan pada gambar 4.1.



Gambar 4.1 Tampilan Input URL

3. Setelah URL dimasukkan klik tombol Start, Lalu aktor akan mulai scraping secara otomatis data komentar dari video TikTok yang telah diinput dan tunggu proses scraping hingga selesai
4. Setelah proses scraping selesai, hasil data akan ditampilkan pada halaman output. Data tersebut kemudian diunduh dalam format CSV. Pada kolom sentimen dikerjakan secara manul, tampilan hasil data dalam format CSV dapat dilihat pada Gambar 4.2.

	Text,CreateTime,UniqueId,VideoWebUrl,Sentimen
2	padahal tiktok shop barangnya real pick ðŸ˜, sekarang selalu pake tt shop dari pada oren atau ijo ðŸ¥²,1695694841,kyaa_here,https://www.tiktok.co
3	baahhh itu bukan jln keluar ya kami dgn ada tik tok shop malah jdi lebih mudah blanja,1695694885,rahmawatywaty565,https://www.tiktok.com/@offici
4	entar gue pindah jualan ke snack vidio ajalah ðŸ˜,,1695694654,ivans_gasvol,https://www.tiktok.com/@officialinews/video/7282937159013682433,Negat
5	aku jadi sedih karena aku udah nyaman belanja di tiktok shop ðŸ¥²,1696349672,nartisunarti2483,https://www.tiktok.com/@officialinews/video/728293
6	aslinya tt membantu untuk orang yang gak ada waktu,1696371854,enipandawa,https://www.tiktok.com/@officialinews/video/7282937159013682433,Po
7	sedihhh krna blnjaku nyaman ditiktok GK pernah kecewa,1696346857,ekayadong,https://www.tiktok.com/@officialinews/video/7282937159013682433,
8	belanja di pasar mahal ya luar biasa barang ya sama di tiktok. murah ya luar biasa.,,1695702773,jasimsembiring,https://www.tiktok.com/@officialinews/
9	pindah ke shopie,1695709950,looloocygnyhasby,https://www.tiktok.com/@officialinews/video/7282937159013682433,Negatif
10	harusnyaaaa dgn adanya Tiktok shop inii bisa menjadi peluang besar untuk pedagang. kenapa gak coba dimanfaatkan saja?,1695707254,fadillahind,https
11	karena tiktok shop gak nyumbang pajak mungkin.,1695710742,duffamb278,https://www.tiktok.com/@officialinews/video/7282937159013682433,Negat
12	semua yg aku butuh ada di tiktok,1695694857,reniarisma,https://www.tiktok.com/@officialinews/video/7282937159013682433,Positif
13	lo saya udh terlanjur love sama tiktok krn saya gk bisa nawar kalau di pasar,1695709357,ikasoliha2,https://www.tiktok.com/@officialinews/video/72829
14	mungkin karna bayaran ke negara kurang. bisa di koreksi kalo salah.,1695705866,ismetpelor,https://www.tiktok.com/@officialinews/video/728293715
15	tiktok shop di hati soalnya barang nya murah murah,1695710587,srindah7508,https://www.tiktok.com/@officialinews/video/7282937159013682433,Pos
16	ya Allah cuma di tiktok sama Nemu baju gede2 buat saya yg gndut bisa spil2ðŸ˜-plis jgn di tutup,1695721566,umma_bany,https://www.tiktok.com/@offi
17	tiktok shop yg jual juga orang orang UMKM pak.,1695711615,dinapuspitasari062,https://www.tiktok.com/@officialinews/video/7282937159013682433,P
18	waaah.. pdhl enak.. duduk manis dirumah barang datang sendiri,1695694740,riniswd,https://www.tiktok.com/@officialinews/video/728293715901368
19	baru aja laku jualan di tiktokshopðŸ˜- di Shopee gk laku,1695694821,cdy.jo,https://www.tiktok.com/@officialinews/video/7282937159013682433,Positif
20	dikeluhkan UMKM yg mna?? Di Tiktok bnyak UMKM modal kcil 1-2 juta bisa jualan.. g sanggup sewa toko lbu2 bs jualan sambil jaga anak...,1695697399,all

Gambar 4.2 format CSV

Pada tanggal 2 juni 2025 penulis melakukan pengumpulan data. Dari hasil scraping tersebut, terkumpul sebanyak 1.313 komentar, Setelah itu data disimpan dan penulis melalukan proses seleksi dan penyaringan sebanyak 500 komentar dipilih untuk dianalisis dalam penelitian ini.

4.3 Import Dataset

Setelah proses pengumpulan dan penyimpanan data selesai dilakukan, tahap selanjutnya adalah mengimpor dataset ke dalam Google Colab. Dataset yang digunakan dalam penelitian ini berisi komentar-komentar pengguna TikTok yang berkaitan dengan TikTok Shop dan telah disimpan dalam format CSV. Proses import dilakukan menggunakan pustaka pandas yang berfungsi untuk membaca dan menampilkan data dalam bentuk tabel agar mudah diolah dan dianalisis pada tahap selanjutnya.

```
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Skripsi/500komentiktok.csv')
```

Gambar 4.3 Proses Untuk Pemanggilan Dataset

Pada gambar 4.3 menunjukkan proses pemanggilan dataset menggunakan pustaka pandas di Google Colab. Pada tahap ini, file 500komentiktok.csv yang telah disimpan di Google Drive dimuat ke dalam variabel df menggunakan fungsi pd.read_csv().

Adapun tampilan data yang berhasil diimpor ke dalam Google Colab dapat dilihat pada Gambar 4.4 berikut.

```
df.head() # Menampilkan 5 data teratas
```

	Text	CreateTime	UniqueId	VideoWebUrl	Sentimen
0	padahal tiktok shop barangnya real pick 😭 😭 , se...	1695694841	kyaa_here	https://www.tiktok.com/@officialinews/video/72...	Positif
1	baahhh itu bukan jln keluar ya kami dgn ada ti...	1695694885	rahmawatywaty565	https://www.tiktok.com/@officialinews/video/72...	Positif
2	entar gue pindah jualan ke snack vidio ajalah 😂	1695694654	ivans_gasvol	https://www.tiktok.com/@officialinews/video/72...	Negatif
3	aku jadi sedih karena aku udah nyaman belanja ...	1696349672	nartisunarti2483	https://www.tiktok.com/@officialinews/video/72...	Positif
4	aslinya tt membantu untuk orang yang gak ada w...	1696371854	enipandawa	https://www.tiktok.com/@officialinews/video/72...	Positif

Gambar 4.4 Menampilkan 5 data teratas

Pada gambar 4.4 menggunakan pustaka pandas untuk menampilkan isi data dalam bentuk tabel. Fungsi df.head() digunakan untuk melihat lima baris pertama dari dataset, yang berisi kolom seperti text, createtime, uniqueid, videoweburl dan sentimen.

4.4 Data (Cleaning)

Langkah selanjutnya yang dilakukan adalah proses data cleaning. Data cleaning bertujuan untuk membersihkan data teks dari elemen-elemen yang tidak relevan Berikut ini merupakan tahapan dan *script* dalam proses *Cleaning*:

1. Mengimpor pustaka re untuk mengenali pola teks dalam proses pembersihan komentar.
2. Penulis melakukan pembersihan teks secara menyeluruh dengan Mengubah semua huruf menjadi huruf kecil, menghapus mention, tanda baca, spasi

berlebihan, huruf berulang lebih dari 2 jadi 1. *Script* tersebut dilihat pada gambar 4.5.

3. Penulis membuat kolom baru bernama “cleaned_text” yang berisi hasil pembersihan dari kolom “text”.

```
import re

#Data Cleaning
def clean_text(text):
    text = text.lower()                          # huruf kecil semua
    text = re.sub(r'@\w+', '', text)             # hapus mention
    text = re.sub(r'[^\w\s]', '', text)          # hapus tanda baca
    text = re.sub(r'\s+', ' ', text).strip()     # hapus spasi berlebihan
    text = re.sub(r'(.)\1{2,}', r'\1', text)     # huruf berulang lebih dari 2 jadi 1
    text = re.sub(r'(.)\1{1}$', r'\1', text)     # Ubah sisa akhir yang dobel jadi 1
    text = re.sub(r'(.)\1+', r'\1', text)        # semua huruf berulang → satu huruf
    return text

df['cleaned_text'] = df['Text'].astype(str).apply(clean_text)

# Tampilkan hasil cleaning
pd.set_option('display.max_colwidth', None)  # agar kolom panjang terlihat penuh
df[['Text','UniqueId', 'Sentimen', 'cleaned_text']].head(10)  # tampilkan 10 baris pertama
```

Gambar 4.5 Script Data Cleaning

Berikut contoh hasil ulasan sebelum dan sesudah melalui proses pembersihan yang terdapat pada table 4.1.

Table 4.1 Contoh Hasil Proses Cleaning

No	UniqueId	Sebelum (Text)	Sesudah (Cleaned_Text)
1	kyaa_here	padahal tiktok shop barangnya real pick 😭😭, sekarang selalu pake tt shop dari pada oren atau ijo □	padahal tiktok shop barangnya real pick sekarang selalu pake t shop dari pada oren atau ijo
2	rahmawatywaty565	baahhh itu bukan jln keluar ya kami dgn ada tik tok shop malah jdi lebih mudah blanja	bah itu bukan jln keluar ya kami dgn ada tik tok shop malah jdi lebih mudah blanja

3	ivans_gasvol	entar gue pindah jualan ke snack vidio ajalah 😂	entar gue pindah jualan ke snack vidio ajalah
4	nartisunarti2483	aku jadi sedih karena aku udah nyaman belanja di tiktok shop	aku jadi sedih karena aku udah nyaman belanja di tiktok shop
6	enipandawa	aslinya tt membantu untuk orang yang gak ada waktu	aslinya t membantu untuk orang yang gak ada waktu
7	ekayadong	sedihhh krna blnjaku nyaman ditiktok GK pernah kecewa	sedih krna blnjaku nyaman ditiktok gk pernah kecewa
8	jasimsembiring	belanja di pasar mahal ya luar biasa barang ya sama di tiktok. murah ya luar biasa.	belanja di pasar mahal ya luar biasa barang ya sama di tiktok murah ya luar biasa
9	looloocygnyhasby	pindah ke shopie	pindah ke shopie
10	fadillahind	harusnyaaaa dgn adanya Tiktok shop inii bisa menjadi peluang besar untuk pedagang. kenapa gak coba dimanfaatkan saja?	harusnya dgn adanya tiktok shop ini bisa menjadi peluang besar untuk pedagang kenapa gak coba dimanfatkan saja

4.4.1 Normalisasi

Setelah proses pembersihan data (data cleaning) dilakukan tahap selanjutnya adalah normalisasi teks. Berikut ini merupakan tahapan dan *script* dalam proses *Normalisasi*:

1. Penulis memisahkan teks menjadi kata-kata menggunakan *text.split().*
2. Kemudian penulis mengubah kata-kata tidak baku (*slang*) menjadi bentuk kata baku. Sehingga kata-kata seperti “tt” menjadi “tiktok”, “blanja”

menjadi “belanja”, “gk” menjadi “gak”. *Script* tersebut dilihat pada gambar 4.6.

3. Penulis membuat kolom baru bernama “normalize_text” yang berisi hasil perbaikan dari kolom “cleaned_text”.

```
def normalize_slang(text):
    # Pisahkan kata-kata
    words = text.split()
    # Kamus slang
    slang_dict = {
    "tt": "tiktok", "t": "tiktok", "tts": "tiktok shop", "tktk": "tiktok", "tiktik": "tiktok", "gk": "gak", "gx": "gak",
    "krna": "karena", "dgn": "dengan", "bgt": "banget", "syg": "sayang", "bnyk": "banyak", "pd": "pada", "belnja": "belan
    "jln": "jalan,", "bgtt": "banget", "shopie": "shoppe", "pdhl": "padahal", "d": "di", "y": "ya", "blnj": "belanja", "m
    "lbh": "lebih", "hrg": "harga", "sgt": "sangat", "psr": "pasar", "gw": "hrs", "bs": "bisa", "karna": "karena", "blnja
    "entar": "nanti","jdi": "jadi", "krn": "karena", "ad": "ada", "aduuh": "aduh", "agr": "agar", "aj": "aje", "ajja": "a
    "apl": "aplikasi", "aplgi": "apalagi", "aplksi": "aplikasi", "aq": "aku", "artis2": "artis", "artist": "artis", "ato"
    "app": "aplikasi", "3rts": "3 ratus", "akyu": "aku", "baikmenghindarin": "baik menghindarin", "bulanja": "belanja", "
    "barangxa": "barangnya", "baranya": "barangnya", "baring2": "baring", "bayak": "banyak", "belaja": "belanja", "belanj
    "apik": "bagus", "berdesakkan": "berdesakan", "bgi": "bagi", "bgus": "bagus", "bikij": "buat", "bikin": "buat", "bisa
    "blanj": "belanja", "blanjanya": "belanjanya", "blh": "boleh", "bljr": "belajar", "blm": "belum", "blnja": "belanja",
    "oren": "shoppe", "ijo": "tokopedia", "ak": "aku", "aing": "aku", "ap": "apa", "bm": "minyak", "bnyak": "banyak", "bn
    "br": "baru", "brang": "barang", "brati": "berarti", "brg": "barang", "brg2": "barang", "brng": "barang", "brng2x": "
    "byk": "banyak", "canggih": "canggih", "cape": "capek", "capet": "capat", "bah": "apa", "gue": "aku", "dimanfatkan": "
}
    normalized_words = [slang_dict.get(word, word) for word in words]
    return ' '.join(normalized_words)

# Terapkan ke DataFrame
df['normalize_text'] = df['cleaned_text'].astype(str).apply(normalize_slang)

# Tampilkan hasilnya
pd.set_option('display.max_colwidth', None)
df[['cleaned_text', 'normalize_text']].head(10)
```

Gambar 4.6 Script Normalisasi

Berikut contoh hasil sebelum dan sesudah melalui proses *Normalisasi* yang terdapat pada table 4.2.

Table 4.2 Contoh Hasil Proses Normalisasi

No	UniqueId	Sebelum (Cleaned_Text)	Sesudah (Normalize_Text)
1	kyaa_here	padahal tiktok shop barangnya real pick sekarang selalu pake t shop dari pada oren atau ijo	padahal tiktok shop barangnya real pick sekarang selalu pakai tiktok shop dari pada shoppe atau tokopedia
2	rahmawatywaty565	bah itu bukan jln keluar ya kami dgn ada tik tok	apa itu bukan jalan, keluar ya kami dengan ada tik tok shop malah

		shop malah jdi lebih mudah blanja	jadi lebih mudah belanja
3	ivans_gasvol	entar gue pindah jualan ke snack vidio ajalah	nanti aku pindah jualan ke snack vidio ajalah
4	nartisunarti2483	aku jadi sedih karena aku udah nyaman belanja di tiktok shop	aku jadi sedih karena aku udah nyaman belanja di tiktok shop
6	enipandawa	aslinya t membantu untuk orang yang gak ada waktu	aslinya tiktok membantu untuk orang yang gak ada waktu
7	ekayadong	sedih krna blnjaku nyaman ditiktok gk pernah kecewa	sedih karena belanjaku nyaman ditiktok gak pernah kecewa
8	jasimsembiring	belanja di pasar mahal ya luar biasa barang ya sama di tiktok murah ya luar biasa	belanja di pasar mahal ya luar biasa barang ya sama di tiktok murah ya luar biasa
9	looloocygnyhasby	pindah ke shopie	pindah ke shoppe
10	fadillahind	harusnya dgn adanya tiktok shop ini bisa menjadi peluang besar untuk pedagang kenapa gak coba dimanfatkan saja	harusnya dengan adanya tiktok shop ini bisa menjadi peluang besar untuk pedagang kenapa gak coba dimanfaatkan saja

4.5 Pra-Pemrosesan Data

Setelah data dibersihkan melalui proses data *cleaning* dan *normalisasi*, langkah selanjutnya adalah pra-pemrosesan data, proses pra-pemrosesan dilakukan melalui beberapa tahapan, yaitu: *tokenisasi, stopwords, stemming, labeling*. Berikut ini merupakan tahapan yang dilakukan dalam pra-pemrosesan data:

4.5.1 Tokenisasi

Tokenisasi merupakan tahap awal dalam proses pra-pemrosesan teks yang bertujuan untuk memecah kalimat menjadi kata-kata tunggal (token). Tokenisasi dilakukan terhadap komentar yang telah melalui tahap pembersihan (cleaning) dan normalisasi. Berikut adalah tahapan dan *script* proses tokenisasi:

1. Penulis mengimport word_tokenize untuk memisahkan kalimat menjadi daftar kata, *script* tersebut dilihat pada gambar 4.7.
2. Kemudian penulis membuat kolom baru Bernama "tokens" yang berisi hasil perbaikan dari kolom "normalize_text".

```
from nltk.tokenize import word_tokenize

# Tokenisasi (ubah menjadi daftar kata)
df['tokens'] = df['normalize_text'].astype(str).apply(word_tokenize)

pd.set_option('display.max_rows', None)
df[['normalize_text', 'tokens']]
```

Gambar 4.7 Script Tokenisasi

Berikut contoh hasil sebelum dan sesudah melalui proses *Tokenisasi* yang terdapat pada table 4.3.

Table 4.3 Contoh Hasil Proses Tokenisasi

No	UniqueId	Sebelum (Normalize_Text)	Sesudah (Tokens)
1	kyaa_here	padahal tiktok shop barangnya real pick sekarang selalu pakai tiktok shop dari pada shoppe atau tokopedia	[padahal, tiktok, shop, barangnya, real, pick, sekarang, selalu, pakai, tiktok, shop, dari, pada, shoppe, atau, tokopedia]
2	rahmawatywaty565	apa itu bukan jalan, keluar ya kami dengan ada tik tok	[apa, itu, bukan, jalan, ,, keluar, ya, kami, dengan,

		shop malah jadi lebih mudah belanja	ada, tik, tok, shop, malah, jadi, lebih, mudah, belanja]
3	ivans_gasvol	nanti aku pindah jualan ke snack vidio ajalah	[nanti, aku, pindah, jualan, ke, snack, vidio, ajalah]
4	nartisunarti2483	aku jadi sedih karena aku udah nyaman belanja di tiktok shop	[aku, jadi, sedih, karena, aku, udah, nyaman, belanja, di, tiktok, shop]
6	enipandawa	aslinya tiktok membantu untuk orang yang gak ada waktu	[aslinya, tiktok, membantu, untuk, orang, yang, gak, ada, waktu]
7	ekayadong	sedih karena belanjaku nyaman ditiktok gak pernah kecewa	[sedih, karena, belanjaku, nyaman, ditiktok, gak, pernah, kecewa]
8	jasimsembiring	belanja di pasar mahal ya luar biasa barang ya sama di tiktok murah ya luar biasa	[belanja, di, pasar, mahal, ya, luar, biasa, barang, ya, sama, di, tiktok, murah, ya, luar, biasa]
9	looloocygnyhasby	pindah ke shoppe	[pindah, ke, shoppe]
10	fadillahind	harusnya dengan adanya tiktok shop ini bisa menjadi peluang besar untuk pedagang kenapa gak coba dimanfaatkan saja	[harusnya, dengan, adanya, tiktok, shop, ini, bisa, menjadi, peluang, besar, untuk, pedagang, kenapa, gak, coba, dimanfaatkan, saja]

4.5.2 Stopword Removal

Setelah proses tokenisasi dilakukan, tahap selanjutnya adalah *Stopword Removal*, Berikut ini merupakan tahapan dan *script* dalam proses *stopword removal*:

1. Penulis menghapus kata-kata umum yang tidak memiliki makna penting dalam proses analisis sentimen. seperti: yang, dengan, dan, ke, menggunakan "remove_stopwords()". *Script* tersebut dapat dilihat pada gambar 4.8.
2. Kemudian penulis membuat kolom baru Bernama "no_stopwords" yang berisi hasil perbaikan dari kolom "token".

```
import nltk
from nltk.corpus import stopwords

# Stopwords Removal (Bahasa Indonesia)
stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(tokens):
    filtered = [word for word in tokens if word not in stop_words]
    return filtered

pd.set_option('display.max_colwidth', None)
df['no_stopwords'] = df['tokens'].apply(remove_stopwords)
```

Gambar 4.8 Script Stopword

Berikut contoh hasil sebelum dan sesudah melalui proses *Tokenisasi* yang terdapat pada table 4.4.

Table 4.4 Contoh Hasil Proses Stopwords

No	UniqueId	Sebelum (Tokens)	Sesudah (Stopwords)
1	kyaa_here	[padahal, tiktok, shop, barangnya, real, pick, sekarang, selalu, pakai, tiktok, shop, dari, pada, shoppe, atau, tokopedia]	[tiktok, shop, barangnya, real, pick, pakai, tiktok, shop, shoppe, tokopedia]

2	rahmawatywaty565	[apa, itu, bukan, jalan, ,, keluar, ya, kami, dengan, ada, tik, tok, shop, malah, jadi, lebih, mudah, belanja]	[jalan, ,, ya, tik, tok, shop, mudah, belanja]
3	ivans_gasvol	[nanti, aku, pindah, jualan, ke, snack, vidio, ajalah]	[pindah, jualan, snack, vidio, ajalah]
4	nartisunarti2483	[aku, jadi, sedih, karena, aku, udah, nyaman, belanja, di, tiktok, shop]	[sedih, udah, nyaman, belanja, tiktok, shop]
6	enipandawa	[aslinya, tiktok, membantu, untuk, orang, yang, gak, ada, waktu]	[aslinya, tiktok, membantu, orang, gak]
7	ekayadong	[sedih, karena, belanjaku, nyaman, ditiktok, gak, pernah, kecewa]	[sedih, belanjaku, nyaman, ditiktok, gak, kecewa]
8	jasimsembiring	[belanja, di, pasar, mahal, ya, luar, biasa, barang, ya, sama, di, tiktok, murah, ya, luar, biasa]	[belanja, pasar, mahal, ya, barang, ya, tiktok, murah, ya]
9	looloocygnyhasby	[pindah, ke, shoppe]	[pindah, shoppe]
10	fadillahind	[harusnya, dengan, adanya, tiktok, shop, ini, bisa, menjadi, peluang, besar, untuk, pedagang, kenapa, gak, coba, dimanfaatkan, saja]	[tiktok, shop, peluang, pedagang, gak, coba, dimanfaatkan]

4.5.3 Stemming

Setelah proses penghapusan stopword dilakukan, langkah selanjutnya adalah *stemming*, Berikut ini merupakan tahapan dan *script* dalam proses *Stemming*:

1. Penulis mengubah setiap kata dalam komentar ke bentuk dasarnya atau kata dasar seperti: kata “berjualan”, “dijual”, dan “barangnya”, “barang”, “membantu”, ”bantu, menggunakan “stemmer.stem()”. Script tersebut dapat dilihat pada gambar 4.9.
2. Kemudian penulis membuat kolom bernama “stemmed_text” yang berisi hasil perbaikan dari kolom “no_stopwords”.

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Inisialisasi Stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Fungsi stemming
def stem_text(text):
    return stemmer.stem(str(text))  # pastikan teks dalam bentuk string

# Terapkan stemming ke kolom hasil stopwords removal (misalnya: 'no_stopwords')
df['stemmed_text'] = df['no_stopwords'].apply(stem_text)

# Tampilkan hasil
pd.set_option('display.max_colwidth', None)
df[['no_stopwords', 'stemmed_text']].head(10)
```

Gambar 4.9 Script Stemming

Berikut contoh hasil sebelum dan sesudah melalui proses *Stemming* yang terdapat pada table 4.5.

Table 4.5 Contoh Hasil Proses Stemming

No	UniqueId	Sebelum (Stopwords)	Sesudah (stemmed_text)
1	kyaa_here	[tiktok, shop, barangnya, real, pick, pakai, tiktok, shop, shoppe, tokopedia]	tiktok shop barang real pick pakai tiktok shop shoppe tokopedia

2	rahmawatywaty565	[jalan, ,, ya, tik, tok, shop, mudah, belanja]	jalan ya tik tok shop mudah belanja
3	ivans_gasvol	[pindah, jualan, snack, vidio, ajalah]	pindah jual snack vidio aja
4	nartisunarti2483	[sedih, udah, nyaman, belanja, tiktok, shop]	sedih udah nyaman belanja tiktok shop
6	enipandawa	[aslinya, tiktok, membantu, orang, gak]	asli tiktok bantu orang gak
7	ekayadong	[sedih, belanjaku, nyaman, ditiktok, gak, kecewa]	sedih belanja nyaman ditiktok gak kecewa
8	jasimsembiring	[belanja, pasar, mahal, ya, barang, ya, tiktok, murah, ya]	belanja pasar mahal ya barang ya tiktok murah ya
9	looloocygnyhasby	[pindah, shoppe]	pindah shoppe
10	fadillahind	[tiktok, shop, peluang, pedagang, gak, coba, dimanfaatkan]	tiktok shop peluang dagang gak coba manfaat

4.5.4 Labeling

Pada tahap ini, proses labeling data sentimen dilakukan secara manual. Komentar yang mengandung tanggapan baik atau mendukung terhadap TikTok Shop diberi label positif, sedangkan komentar yang menunjukkan ketidakpuasan diberi label negatif. Berikut ini tahapan dan *script* proses *labeling*:

1. Penulis melakukan proses Label Encoding untuk mengubah data pada kolom sentimen menjadi bentuk numerik agar dapat diproses oleh algoritma machine learning.

2. Kemudian penulis membuat dua kategori yaitu: kategori positif diberi label 1 dan negatif diberi label 0. Script tersebut dapat dilihat pada gambar 4.10.
3. Hasil labeling kemudian disimpan kembali dalam kolom Sentimen pada DataFrame. Proses ini memastikan bahwa data dapat digunakan sebagai input pada model klasifikasi SVM atau Logistic Regression, yang memerlukan nilai numerik.

```
#LABEL ENCODING✓

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sentimen'] = le.fit_transform(df['Sentimen'])  # pastikan kolomnya benar
#0 = negatif, 1 = positif (bisa dicek lewat `le.classes_`)
```

Gambar 4.10 Script Labeling

Berikut contoh hasil labeling data yang dimana kategori positif diberi label 1 dan negatif diberi label 0, yang terdapat pada Tabel table 4.6.

Table 4.6 Contoh Hasil Proses Labeling

No	UniqueId	Stemmed_text	Sentimen
1	kyaa_here	tiktok shop barang real pick pakai tiktok shop shoppe tokopedia	1
2	rahmawatywaty565	jalan ya tik tok shop mudah belanja	1
3	ivans_gasvol	pindah jual snack vidio aja	0
4	nartisunarti2483	sedih udah nyaman belanja tiktok shop	1
6	enipandawa	asli tiktok bantu orang gak	1
7	ekayadong	sedih belanja nyaman ditiktok gak kecewa	1
8	jasimsembiring	belanja pasar mahal ya barang ya tiktok murah ya	1

9	looloocygnyh asby	pindah shoppe	0
10	fadillahind	tiktok shop peluang dagang gak coba manfaat	1

Pada gambar 4.11 merupakan *Script* untuk menampilkan hasil jumlah sentimen dan menampilkan visualisasi dari sentiment.

```
# Hitung jumlah tiap jenis sentimen
print("Jumlah sentimen (berdasarkan teks):")
print(df['Sentimen'].value_counts())

# Hitung jumlah masing-masing sentimen
label_counts = df['Sentimen'].value_counts()
labels = ['Positif', 'Negatif']
jumlah = [label_counts[1], label_counts[0]]  # asumsi 1 = positif, 0 = negatif
colors = ['skyblue', 'lightcoral']

# Buat diagram batang
plt.figure(figsize=(6, 4))
plt.bar(labels, jumlah, color=colors)
plt.title('Distribusi Sentimen Komentar')
plt.xlabel('Kategori Sentimen')
plt.ylabel('Jumlah Komentar')
plt.grid(axis='y', linestyle='--', linewidth=0.5)
plt.show()
```

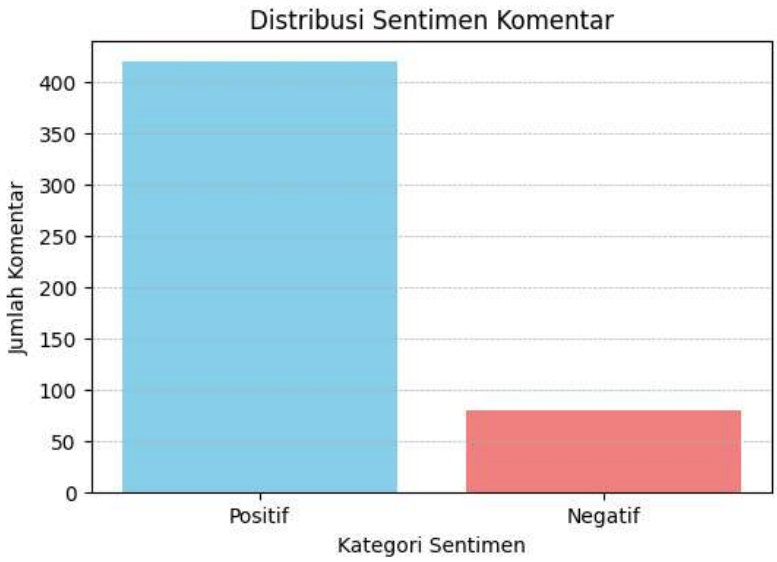
Gambar 4.11 *Script* Jumlah sentimen

Pada gambar 4.12 menunjukkan hasil perhitungan jumlah komentar berdasarkan kategori sentimen di mana terdapat 420 komentar yang dikategorikan sebagai sentimen positif (dilabeli dengan angka 1) dan 80 komentar yang termasuk dalam sentimen negatif (dilabeli dengan angka 0).

Hasil ini menunjukkan bahwa mayoritas komentar yang dianalisis dalam dataset ini bersifat positif terhadap TikTok Shop, sedangkan sebagian kecil menunjukkan opini negatif.

```
Jumlah sentimen (berdasarkan teks):
Sentimen
1    420
0     80
Name: count, dtype: int64
```

Gambar 4.12 Jumlah sentimen



Gambar 4.13 Visualisasi Sentimen

Pada gambar 4.13 meruapakan tampilan visualisasi distribusi sentimen komentar dalam bentuk diagram batang. Visualisasi ini membantu menggambarkan kecenderungan opini masyarakat terhadap TikTok Shop secara lebih jelas.

4.6 Modeling

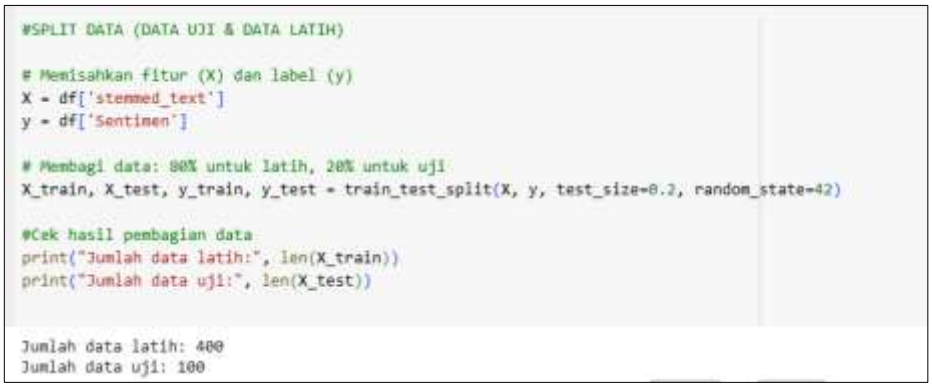
Pada tahap ini, data yang telah melalui proses pra-pemrosesan dibagi menjadi dua bagian, yaitu pembagiana data latih dan data uji Selanjutnya, dilakukan ekstraksi fitur menggunakan metode TF-IDF untuk mengubah data teks menjadi bentuk numerik. Hasil transformasi TF-IDF kemudian digunakan sebagai input untuk membangun model klasifikasi menggunakan algoritma Support Vector

Machine (SVM) dan Logistic Regression. Berikut ini merupakan tahapan yang dilakukan dalam modeling:

4.6.1. Pembagian Data Latih dan Data Uji

Setelah data selesai melalui tahapan pra-pemrosesan dan pelabelan, langkah selanjutnya adalah membagi data ke dalam dua bagian, yaitu data latih (*training data*) dan data uji (*testing data*). Berikut ini tahapan dan *script* proses Pembagian data latih dan data uji:

1. Penulis memisahkan antara fitur label X, yang berisi kolom "stemmed_text", dan label Y, yang berisi kolom "Sentimen". Selanjutnya, data dibagi dengan 80% untuk data latih dan 20% untuk data uji. Script tersebut dapat dilihat pada gambar 4.14.
2. Kemudian penulis menampilkan Hasil dari proses pembagian menggunakan "print()" untuk mengecek jumlah data pada masing-masing bagian. Berdasarkan output yang ditampilkan, diperoleh sebanyak 400 data latih dan 100 data uji, dari total 500 data komentar yang digunakan dalam penelitian.



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

# Memisahkan fitur (X) dan label (y)
X = df['stemmed_text']
y = df['Sentimen']

# Membagi data: 80% untuk latih, 20% untuk uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

```
Jumlah data latih: 400
Jumlah data uji: 100
```

Gambar 4.14 Script Pembagian Data

Pada gambar 4.15 dibawah merupakan *script* proses pembuatan visualisasi distribusi sentimen pada data latih dan data uji. Langkah pertama adalah menghitung jumlah sentimen positif dan negatif pada kedua jenis data, lalu digabungkan ke dalam satu tabel. Kemudian divisualisasikan menggunakan diagram batang untuk melihat perbandingan antara data latih dan uji. Visualisasi ini membantu memahami keseimbangan kelas sebelum proses pelatihan model.

```
# Hitung jumlah label di data latih
train_counts = y_train.value_counts().rename(index={1: 'Positif', 0: 'Negatif'})

# Hitung jumlah label di data uji
test_counts = y_test.value_counts().rename(index={1: 'Positif', 0: 'Negatif'})

# Gabungkan ke satu tabel untuk ditampilkan
summary_df = pd.DataFrame({
    'Data Latih': train_counts,
    'Data Uji': test_counts
}).fillna(0).astype(int)

print(summary_df)

# Visualisasi Bar Chart
summary_df.plot(kind='bar', color=['skyblue', 'salmon'])
plt.title('Distribusi Sentimen pada Data Latih dan Uji')
plt.ylabel('Jumlah')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# Visualisai Data Latih
train_counts.plot.pie(autopct='%1.1f%%', startangle=140, colors=['lightgreen', 'lightcoral'])
plt.title('Distribusi Sentimen Data Latih')
plt.ylabel('')
plt.show()

# Visualisasi Data Uji
test_counts.plot.pie(autopct='%1.1f%%', startangle=140, colors=['lightblue', 'orange'])
plt.title('Distribusi Sentimen Data Uji')
plt.ylabel('')
plt.show()
```

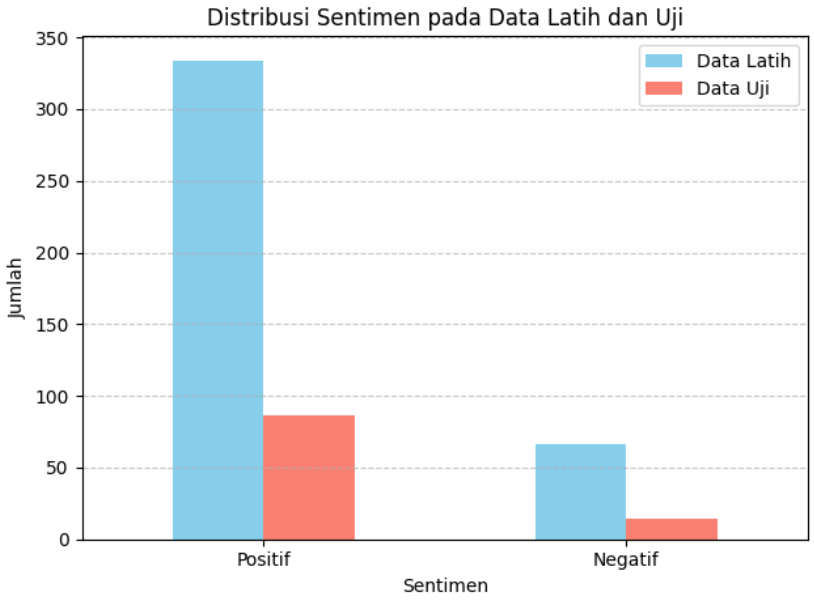
Gambar 4.15 Script Visualisasi Distribusi

Pada table 4.6 dibawah menunjukkan hasil distribusi sentimen pada data latih dan data uji yang telah dibagi sebelumnya. Sentimen terbagi menjadi dua kategori, yaitu positif dan negatif, yang masing-masing ditampilkan dalam dua warna berbeda: biru untuk data latih dan merah untuk data uji. Visualisasi tersebut dapat dilihat pada gambar 4.16

Table 4.7 Jumlah data Latih dan data Uji

Sentimen	Data Latih	Data Uji
Positif	334	86
Negatif	66	14

Dari visualisasi tersebut, terlihat bahwa pada data latih, jumlah komentar dengan sentimen positif berjumlah 334 komentar, sedangkan komentar dengan sentimen negatif berjumlah 66 komentar. Sementara itu, pada data uji, komentar positif berjumlah 86 komentar, dan komentar negatif berjumlah sekitar 14 komentar. Hasil tersebut dapat dilihat pada table 4.7.



Gambar 4.16 Visualisasi Data

4.6.2. TF-IDF

Langkah selanjutnya penulis mengubah data teks ke dalam bentuk numerik agar dapat diproses oleh algoritma machine learning. Proses ini dilakukan dengan menggunakan metode TF-IDF. Berikut ini tahapan dan *script* proses TF-IDF:

1. Penulis menggunakan TfidfVectorizer dari pustaka sklearn untuk mengubah data teks menjadi bentuk numerik. Sebanyak 1.000 kata dengan bobot tertinggi yang akan digunakan sebagai fitur. Script tersebut dapat dilihat pada gambar 4.17.
2. Selanjutnya penulis menggunakan fungsi Shape untuk melihat ukuran (dimensi) dari matriks hasil TF-IDF, baik untuk data latih (X_train_tfidf) maupun data uji (X_test_tfidf).
3. Setelah data berhasil dikonversi ke dalam format numerik data siap digunakan dalam proses pelatihan dan pengujian model klasifikasi.

```
#Ekstraksi Fitur: TF-IDF✓

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=1000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

#Melihat data uji setelah TF-IDF✓

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

Gambar 4.17 Script Ekstrasi Fitur TF-IDF

```
Shape X_train_tfidf: (400, 978)
Shape X_test_tfidf: (100, 978)
```

Gambar 4.18 Hasil TF-IDF

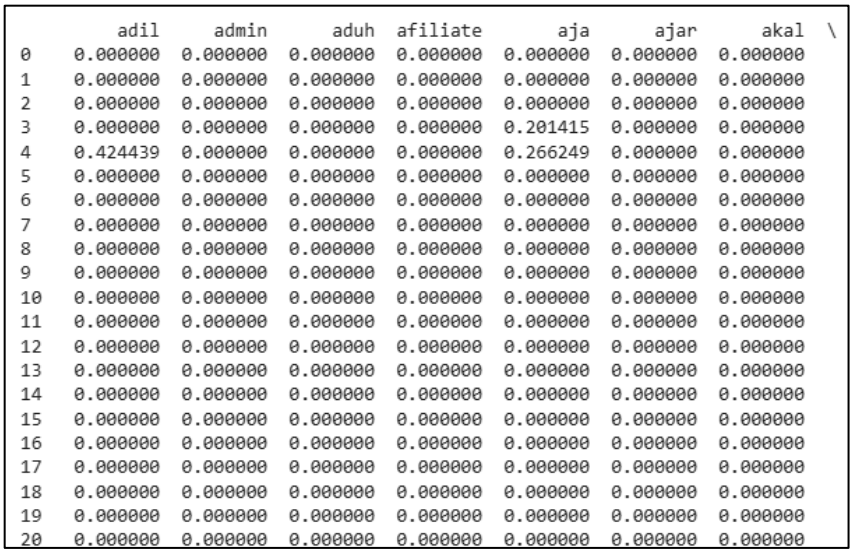
Pada gambar 4.18 di atas menampilkan hasil dari fungsi shape setelah proses transformasi TF-IDF. Berdasarkan hasil tersebut, diketahui bahwa data latih (X_train_tfidf) memiliki hasil (400, 978), yang berarti terdapat 400 data latih dan 978 fitur unik yang dihasilkan dari proses ekstraksi teks. Sementara itu, data uji (X_test_tfidf) memiliki hasil (100, 978), menandakan terdapat 100 data uji dengan jumlah fitur yang sama, yaitu 978. Kesamaan jumlah fitur antara data latih dan data uji menunjukkan bahwa transformasi TF-IDF berhasil dilakukan dengan konsisten, sehingga data siap digunakan dalam proses pelatihan dan pengujian model klasifikasi.

```
tfidf = TfidfVectorizer()
X_train_tfidf = tfidf.fit_transform(X_train)
# Ambil daftar fitur (kata-kata)
feature_names = tfidf.get_feature_names_out()

# Ubah hasil TF-IDF menjadi DataFrame
tfidf_df = pd.DataFrame(X_train_tfidf.toarray(), columns=feature_names)

# Tampilkan seluruh data dari index 0 sampai 399 (data latih = 400)
pd.set_option('display.max_rows', None)      # (Opsional) agar baris tidak terpotong
pd.set_option('display.max_columns', None)  # Tampilkan semua kolom (kata)
pd.set_option('display.width', None)        # Hindari pemotongan tampilan lebar

# Tampilkan semua data vektorisasi dari 0-4
print(tfidf_df.iloc[:400])
```

Gambar 4.19 Script TF-IDF Data Latih

Pada gambar 4.19 di atas menunjukkan proses untuk mengubah hasil transformasi TF-IDF menjadi bentuk DataFrame, agar dapat dianalisis dan ditampilkan dengan lebih jelas. Langkah pertama penulis mentransformasikan data latih (X_train) menggunakan TfidfVectorizer. Selanjutnya hasil TF-IDF yang awalnya berbentuk sparse matrix diubah ke dalam array lalu dikonversi menjadi DataFrame dengan kolom-kolomnya mewakili kata-kata fitur. Kemudian hasil vektorisasi TF-IDF dari baris indeks 0 hingga 400 ditampilkan untuk representasi

data latih dalam bentuk numerik berdasarkan bobot kata dari TF-IDF. Hasil Tersebut dapat dilihat pada gambar 4.20.



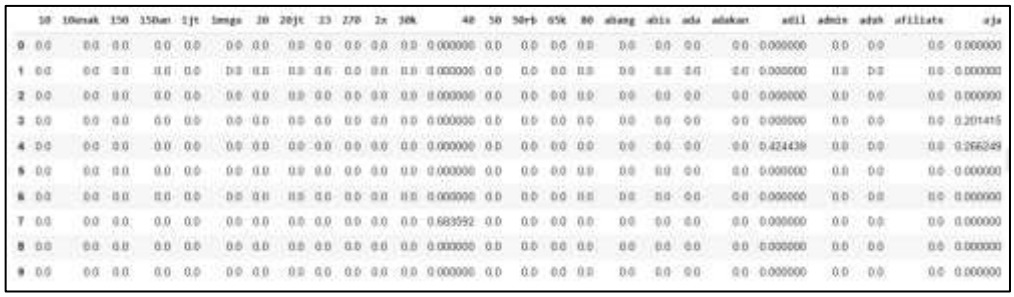
	adil	admin	aduh	afiliate	aja	ajar	akal	\
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
2	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
3	0.000000	0.000000	0.000000	0.000000	0.201415	0.000000	0.000000	
4	0.424439	0.000000	0.000000	0.000000	0.266249	0.000000	0.000000	
5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
14	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
16	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
17	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
18	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
19	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
20	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	

Gambar 4.20 Hasil TF-IDF Data Latih

Tahap selanjutnya penulis juga menampilkan hasil TF-IDF menjadi bentuk tabel (DataFrame) agar lebih mudah dibaca dan dianalisis. *Script* tersebut dapat dilihat pada gambar 4.20, Untuk hasil TF-IDF dalam bentuk table dapat dilihat pada gambar 4.22.

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

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

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

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

Gambar 4. 21 Script Menampilkan Table TF-IDF



Gambar 4.22 Hasil Table TF-IDF

4.7 Algoritma Support Vector Machine (SVM)

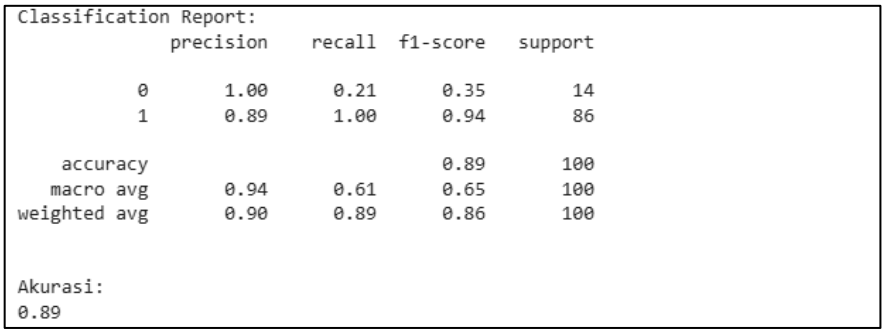
Pada tahap ini, penulis menerapkan algoritma Support Vector Machine (SVM) sebagai salah satu metode klasifikasi untuk menganalisis sentimen komentar masyarakat terhadap TikTok Shop. SVM digunakan karena kemampuannya dalam memisahkan data ke dalam dua kelas, yaitu positif dan negatif. Model dilatih menggunakan data latih yang telah diproses.

```
# Import library
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Melatih model menggunakan data latih
svm_model.fit(X_train_tfidf, y_train)

# Melakukan prediksi terhadap data uji
y_pred_svm = svm_model.predict(X_test_tfidf)

# Evaluasi hasil prediksi
print("\nClassification Report:")
print(classification_report(y_test, y_pred_svm))

print("\nAkurasi:")
print(accuracy_score(y_test, y_pred_svm))
```

Gambar 4.23 Script Penerapan Algoritma Support Vector Machine

Pada gambar 4.23 merupakan *script* penerapan algoritma Support Vector Machine (SVM). Penulis melakukan import library, yaitu SVC dari sklearn.svm untuk membangun model SVM, untuk melakukan pelatihan model menggunakan data latih dan prediksi menggunakan data uji. Bagian akhir dari script diatas adalah evaluasi hasil prediksi, yang ditampilkan dalam bentuk Classification Report yang

menampilkan precision, recall, dan f1-score untuk masing-masing kelas dan akurasi model, yang menunjukkan persentase prediksi yang benar terhadap total data uji.



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

           0       1.00      0.21      0.35        14
           1       0.89      1.00      0.94        86

    accuracy                           0.89       100
   macro avg       0.94      0.61      0.65       100
weighted avg       0.90      0.89      0.86       100


Akurasi:
0.89
```

Gambar 4. 24 Hasil Algoritma Support Vector Machine

Pada Gambar 4.24 tersebut menampilkan hasil evaluasi performa model Support Vector Machine (SVM) yang digunakan untuk analisis sentimen, ditunjukkan melalui classification report dan nilai akurasi. Berdasarkan laporan klasifikasi, terdapat dua kelas sentimen, yaitu kelas 0 (negatif) dan kelas 1 (positif).

Untuk kelas 0, model memiliki precision sebesar 1.00, yang artinya semua prediksi untuk kelas negatif benar. Namun, recall hanya sebesar 0.21, menunjukkan bahwa dari seluruh data negatif yang seharusnya terdeteksi, hanya 21% yang berhasil diprediksi oleh model. Hal ini membuat nilai f1-score untuk kelas 0 cukup rendah, yaitu 0.35, karena f1-score mempertimbangkan keseimbangan antara precision dan recall. Jumlah data (support) untuk kelas 0 sebanyak 14.

Sementara itu, untuk kelas 1 (positif), model menunjukkan performa yang sangat baik dengan precision sebesar 0.89, recall 1.00, dan f1-score 0.94. Artinya, sebagian besar data sentimen positif berhasil dikenali dan diklasifikasikan dengan tepat oleh model. Jumlah data untuk kelas ini adalah 86. Secara keseluruhan, akurasi model adalah 0.89, yang berarti 89% dari total data uji diklasifikasikan

dengan benar. Nilai macro average (rata-rata untuk tiap kelas tanpa mempertimbangkan jumlah data) menunjukkan precision 0.94, recall 0.61, dan f1-score 0.65, sedangkan weighted average (rata-rata tertimbang berdasarkan jumlah data per kelas) lebih tinggi, dengan precision 0.90, recall 0.89, dan f1-score 0.86.

Dari hasil ini, dapat disimpulkan bahwa model bekerja sangat baik untuk mendeteksi sentimen positif, namun kurang optimal dalam mendeteksi sentimen negatif, dikarena distribusi data yang tidak seimbang (kelas positif jauh lebih banyak dari kelas negatif).

4.8 Logistic Regression

Langkah selanjutnya dalam penelitian ini adalah menerapkan algoritma Logistic Regression karena merupakan salah satu metode klasifikasi yang sering digunakan dalam pemodelan data kategorik, termasuk dalam analisis sentimen. Pada tahap ini, data latih yang telah melalui proses ekstraksi fitur dengan metode TF-IDF akan digunakan untuk membangun model, kemudian dilakukan prediksi terhadap data uji untuk mengevaluasi performa dari algoritma Logistic Regression.

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

# Inisialisasi model
logreg = LogisticRegression()

# Latih model
logreg.fit(X_train_tfidf, y_train)

# Prediksi pada data uji
y_pred_logreg = logreg.predict(X_test_tfidf)

# Akurasi
print("Akurasi Logistic Regression:", accuracy_score(y_test, y_pred_logreg))

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred_logreg, target_names=['Negatif:0', 'Positif:1']))
```

Gambar 4.25 Script Penerapan Algoritma Logistic Regression

Pada gambar 4.25 merupakan *script* penerapan algoritma Logistic Regression dalam proses analisis sentimen. Penulis mengimpor library yang LogisticRegression dari sklearn.linear_model serta classification_report dan accuracy_score. Selanjutnya, model Logistic Regression diinisialisasi melalui variabel logreg, kemudian dilatih menggunakan data latih hasil ekstraksi TF-IDF (X_train_tfidf dan y_train).

```
Akurasi Logistic Regression: 0.86

Classification Report:
              precision    recall  f1-score   support

   Negatif:0       0.00      0.00      0.00        14
   Positif:1       0.86      1.00      0.92        86

    accuracy                           0.86       100
   macro avg       0.43      0.50      0.46       100
weighted avg       0.74      0.86      0.80       100
```

Gambar 4.26 Hasil Algoritma Logistic Regression

Pada gambar 4.26 menunjukkan hasil evaluasi performa model Logistic Regression dalam mengklasifikasikan sentimen masyarakat terhadap TikTok Shop berdasarkan data yang diperoleh dari media sosial. Dari output tersebut, terlihat bahwa model menghasilkan akurasi sebesar 0.86 atau 86%, yang berarti 86 dari 100 data uji berhasil diklasifikasikan dengan benar oleh model. Namun, akurasi ini tidak menunjukkan performa yang ideal, karena perlu dilihat dari hasil precision, recall, dan f1-score pada masing-masing kelas.

Dari classification report, Untuk kelas Negatif (label 0), precision, recall, dan f1-score semuanya bernilai 0.00, dengan jumlah dukungan (support) sebanyak 14 data. Ini berarti model sama sekali tidak mampu mengenali data yang tergolong negatif. Untuk kelas Positif (label 1) yang memiliki dukungan 86 data, model memiliki precision sebesar 0.86, recall 1.00, dan f1-score sebesar 0.92,

menunjukkan bahwa model sangat baik dalam mengenali sentimen positif. Namun, performa ini menunjukkan adanya ketidak seimbangan dalam pembelajaran model.

Nilai macro average untuk precision, recall, dan f1-score masing-masing adalah 0.43, 0.50, dan 0.46. Macro average menghitung rata-rata metrik untuk tiap kelas tanpa mempertimbangkan proporsi datanya, sehingga nilai ini mencerminkan ketidak seimbangan performa model terhadap kelas minoritas (negatif). Sebaliknya, nilai weighted average yang memperhitungkan jumlah data per kelas memberikan nilai precision sebesar 0.74, recall 0.86, dan f1-score sebesar 0.80, yang masih menunjukkan performa dominan terhadap kelas positif.

Secara keseluruhan, hasil dari model Logistic Regression dalam penelitian ini cenderung mengabaikan sentimen negatif, sehingga meskipun akurasinya terlihat tinggi, model tidak cukup andal dalam mengklasifikasikan opini masyarakat secara menyeluruh. Hal ini bisa disebabkan oleh ketidak seimbangan jumlah data sentimen, di mana kelas positif mendominasi dataset. Dengan demikian, model dapat memberikan hasil analisis sentimen yang lebih akurat dan representatif terhadap persepsi masyarakat terhadap TikTok Shop di media sosial.

4.9 Evaluasi Model

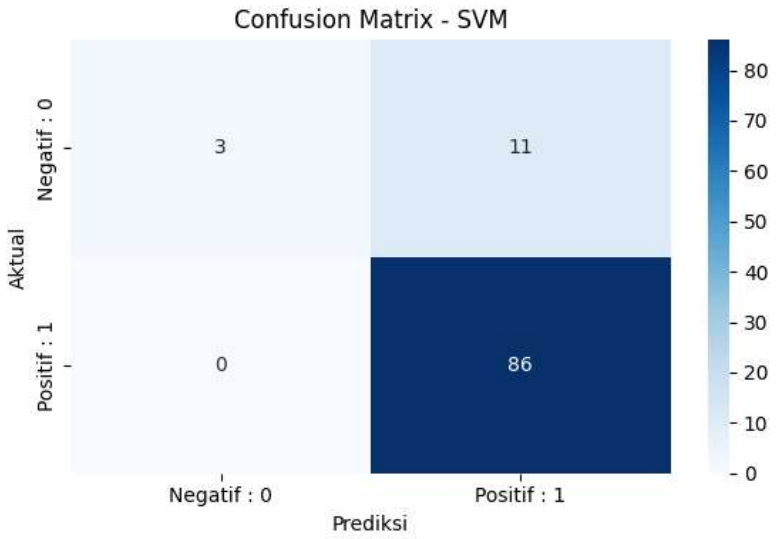
Pada tahap ini penulis melakukan evaluasi terhadap kinerja dua algoritma yang telah diterapkan, yaitu Support Vector Machine (SVM) dan Logistic Regression. Evaluasi dilakukan dengan menggunakan data uji. Tujuan dari evaluasi ini adalah untuk mengetahui seberapa efektif masing-masing algoritma dalam mengklasifikasikan sentimen pada komentar masyarakat terhadap TikTok Shop. Berikut ini merupakan evaluasi *Confusion Matrix dan Classification Report*:

4.9.1. Confusion Matrix Support Vector Machine (SVM)

Penulis menampilkan confusion matrix untuk algoritma Support Vector Machine (SVM), Confusion matrix digunakan untuk melihat jumlah prediksi yang benar dan salah dari masing-masing kelas. Berikut adalah *script* dan hasil *confusion matrix Support Vector Machine SVM*:

```
cm = confusion_matrix(y_test, y_pred_svm)

class_names = ['Negatif : 0', 'Positif : 1']
plt.figure(figsize=(6, 4))
sns.heatmap(pd.DataFrame(cm), annot=True, cmap="Blues", fmt='g', xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix - SVM')
plt.tight_layout()
plt.show()
```

Gambar 4.27 Script Confusion Matrix Support Vector Machine SVM



Gambar 4.28 Hasil Confusion Matrix SVM

Pada gambar 4.28 menunjukkan *confusion matrix* dari hasil klasifikasi menggunakan algoritma Support Vector Machine (SVM) terhadap dua kelas: Negatif (0) dan Positif (1). Confusion matrix ini menggambarkan performa model dalam mengenali masing-masing kelas berdasarkan hasil prediksi terhadap data uji.

Dari confusion matrix tersebut, diketahui bahwa model berhasil memprediksi 86 data yang sebenarnya positif dengan benar sebagai positif (True Positive), dan 3 data yang sebenarnya negatif juga diprediksi dengan benar sebagai negatif (True Negative). Namun, terdapat 11 data negatif yang salah diprediksi sebagai positif (False Positive), dan tidak ada data positif yang salah diprediksi sebagai negatif (False Negative = 0).

Secara keseluruhan, akurasi model dapat dihitung sebagai jumlah prediksi benar dibagi total data, yaitu:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \text{ X } 100\%$$

$$= \frac{86 + 3}{86 + 3 + 11 + 0} \text{ X } 100\%$$

$$= \frac{89}{100} \text{ X } 100\%$$

$$= 0{,}89$$

Kemudian, precision untuk kelas positif (seberapa banyak prediksi positif yang benar-benar positif) adalah:

$$Precision = \frac{TP}{TP + FP} \text{ X } 100\%$$

$$= \frac{86}{86 + 11} \text{ X } 100\%$$

$$= \frac{86}{97} \text{ X } 100\%$$

$$= 0{,}88$$

Sedangkan recall (seberapa banyak data positif yang berhasil dikenali) adalah:

$$Recall = \frac{TP}{TP + FN} \text{ X } 100\%$$

$$= \frac{86}{86 + 0} \text{ X } 100\%$$

$$= \frac{86}{86} \text{ X } 100\%$$

$$= 100$$

F-1 Score menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan

$$F - 1\ Score = \frac{2 \text{ x } Recall \text{ x } Precision}{Recall + Precision}$$

$$= \frac{2 \text{ x } 1 \text{ x } 0{,}88}{1 + 0{,}88}$$

$$= \frac{1{,}76}{1{,}88}$$

$$= 93$$

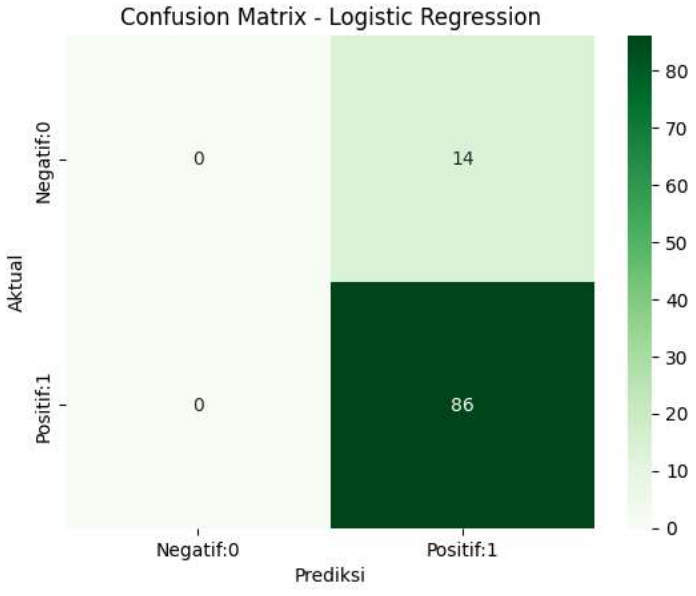
Dari hasil ini, dapat disimpulkan bahwa model SVM memiliki kemampuan yang sangat baik dalam mengenali kelas positif karena tidak ada false negative. Namun, model masih memiliki kelemahan dalam mengenali kelas negatif, terlihat dari 11 kesalahan prediksi pada data negatif (false positive), yang menurunkan tingkat precision. Model SVM secara umum menunjukkan performa yang cukup baik dengan tingkat akurasi dan recall yang tinggi.

4.9.2. Confusion Matrix Logistic Regression

Pada tahap ini penulis menampilkan confusion matrix algoritma Logistic Regression untuk mengetahui jumlah prediksi yang benar dan salah dari masing-masing kelas, serta menjadi dasar dalam perhitungan metrik evaluasi seperti precision, recall, dan f1-score. Berikut adalah script dan hasil confusion matrix Logistic Regression:

```
# Confusion matrix
cm = confusion_matrix(y_test, y_pred_logreg)

sns.heatmap(cm, annot=True, fmt='d', cmap='Greens', xticklabels=['Negatif:0' , 'Positif:1'], yticklabels=['Negatif:0', 'Positif:1'])
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix - Logistic Regression')
plt.show()

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

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

report_df_class[['precision', 'recall', 'f1-score']].plot(kind='bar', figsize=(8,4))
plt.title('Precision, Recall, F1-score per Class - Logistic Regression')
plt.ylim(0,1)
plt.ylabel('Score')
plt.xticks(rotation=45)
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

Gambar 4.29 Script Confusion Matrix Logistic Regression



Gambar 4.30 Hasil Confusion Matrix Logistic Regression

Pada gambar 4.30 di atas menunjukkan confusion matrix dari hasil klasifikasi menggunakan algoritma Logistic Regression pada dua kelas: Negatif (0) dan Positif (1). Matrix ini digunakan untuk mengevaluasi kinerja model klasifikasi dengan membandingkan hasil prediksi model terhadap data yang sebenarnya.

Dari confusion matrix tersebut, kita dapat melihat bahwa model memprediksi semua data sebagai kelas positif (1). Terdapat 86 data aktual positif yang berhasil diklasifikasikan dengan benar sebagai positif (True Positive), dan 14 data aktual negatif yang salah diklasifikasikan sebagai positif (False Positive). Tidak ada satu pun data negatif yang diklasifikasikan dengan benar (True Negative = 0), dan tidak ada data positif yang salah diklasifikasikan sebagai negatif (False Negative = 0). Berikut ini perhitungan metrix evaluasi berdasarkan data tersebut:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \text{ X } 100\%$$

$$= \frac{86 + 0}{86 + 0 + 14 + 0} \text{ X } 100\%$$

$$= \frac{86}{100} \text{ X } 100\%$$

$$= 0{,}86$$

Kemudian, precision untuk kelas positif (seberapa banyak prediksi positif yang benar-benar positif) adalah:

$$Precision = \frac{TP}{TP + FP} \text{ X } 100\%$$

$$= \frac{86}{86 + 14} \text{ X } 100\%$$

$$= \frac{86}{100} \text{ X } 100\%$$

$$= 0{,}86$$

Sedangkan recall (seberapa banyak data positif yang berhasil dikenali) adalah:

$$Recall = \frac{TP}{TP + FN} \text{ X } 100\%$$

$$= \frac{86}{86 + 0} \text{ X } 100\%$$

$$= \frac{86}{86} \text{ X } 100\%$$

$$= 100$$

F-1 Score menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan

$$F - 1\ Score = \frac{2 \text{ x } Recall \text{ x } Precision}{Recall + Precision}$$

$$= \frac{2 \text{ x } 1 \text{ x } 0{,}86}{1 + 0{,}86}$$

$$= \frac{1{,}72}{1{,}86}$$

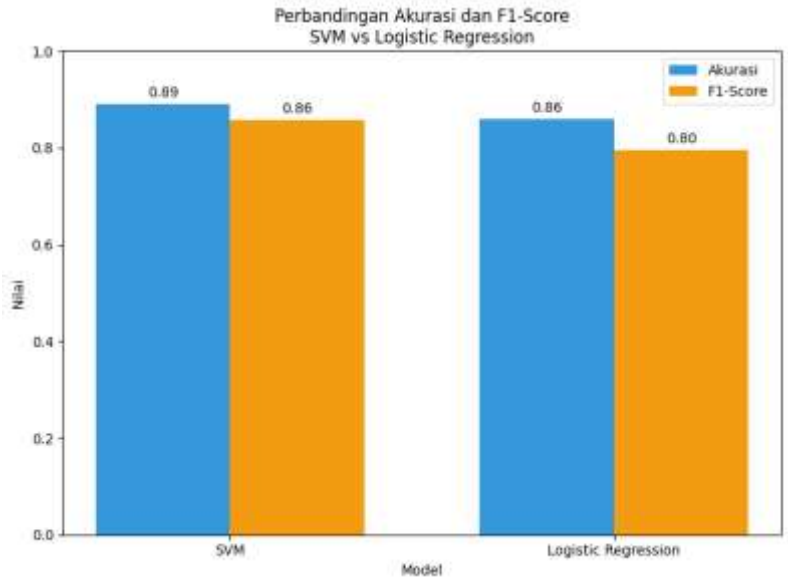
$$= 92$$

Hasil confusion matrix ini menunjukkan bahwa model sepenuhnya cenderung terhadap kelas positif, karena tidak mampu mengklasifikasikan satu pun data sebagai kelas negatif. Meskipun recall-nya tinggi (100%) karena semua data positif dikenali dengan benar, presisi menurun karena adanya 14 false positive. Hal ini terjadi dikarenakan model dilatih pada data yang tidak seimbang, di mana jumlah data positif jauh lebih banyak dibandingkan data negatif.

4.10 Analisis Hasil

Berdasarkan hasil evaluasi peforma model, diperoleh bahwa algoritma Support Vector Machine (SVM) memiliki akurasi sebesar 0.89 dan f1-score sebesar 0.86 (dibulatkan). Sementara itu, algoritma Logistic Regression memiliki akurasi sebesar 0.86 dan f1-score sebesar 0.80 (dibulatkan).

Berikut ini merupakan hasil visualisasi perbandingan antara kedua algoritma Support Vector Machine (SVM) dan Logistic Regression:



Gambar 4.31 Hasil Visualisasi Perbandingan

Akurasi menunjukkan proporsi prediksi yang benar terhadap seluruh data uji, sedangkan f1-score merupakan metrik yang mempertimbangkan keseimbangan antara precision dan recall, ketika data tidak seimbang.

Dari hasil di atas, dapat disimpulkan bahwa Support Vector Machine (SVM) memiliki kinerja yang lebih baik dibandingkan Logistic Regression, baik dari segi akurasi maupun f1-score. Nilai f1-score Support Vector Machine (SVM) yang lebih tinggi menunjukkan bahwa model ini lebih seimbang dalam mengenali kedua kelas

(positif dan negatif), serta lebih akurat dalam memprediksi data uji. Perbedaan nilai f1-score yang cukup signifikan menunjukkan bahwa Support Vector Machine (SVM) lebih efektif dalam menangani tantangan klasifikasi sentimen, khususnya dalam konteks data yang cenderung tidak seimbang.

Sementara itu, Logistic Regression masih memberikan performa yang cukup baik, tetapi terlihat lebih sensitif terhadap data yang tidak seimbang, yang dapat menyebabkan model terlalu fokus pada kelas mayoritas. Hal ini dapat menyebabkan penurunan presisi atau recall, terutama pada kelas minoritas.

Secara keseluruhan, berdasarkan metrik evaluasi yang digunakan, Support Vector Machine(SVM) lebih unggul dalam mengklasifikasikan sentimen komentar masyarakat terhadap TikTok Shop dibandingkan Logistic Regression. Dengan demikian, Support Vector Machine (SVM) dapat dianggap sebagai algoritma yang lebih optimal untuk digunakan dalam penelitian ini.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan mengenai analisis sentimen masyarakat terhadap TikTok Shop menggunakan algoritma Support Vector Machine (SVM) dan Logistic Regression, maka dapat disimpulkan bahwa:

1. Algoritma Support Vector Machine (SVM) menunjukkan performa yang lebih unggul dibandingkan Logistic Regression dalam mengklasifikasikan sentimen komentar masyarakat di TikTok terhadap TikTok Shop. Hal ini ditunjukkan oleh nilai akurasi dan f1-score yang lebih tinggi, yaitu akurasi sebesar 0,89 dan f1-score sebesar 0,86 untuk SVM, dibandingkan dengan Logistic Regression yang memiliki akurasi 0,86 dan f1-score 0,80.
2. Support Vector Machine (SVM) lebih seimbang dalam mengenali kedua kelas sentimen (positif dan negatif), terutama dalam konteks data yang tidak seimbang.
3. Penelitian ini menunjukkan bahwa algoritma Support Vector Machine (SVM) lebih optimal dan efisien untuk digunakan dalam tugas klasifikasi sentimen terhadap data teks dari media sosial TikTok, khususnya dalam konteks e-commerce TikTok Shop.
4. Jumlah data yang digunakan sebanyak 500 komentar, dengan 420 komentar termasuk dalam kategori positif dan 80 komentar termasuk kategori negatif. Data ini diproses melalui tahapan cleaning, preprocessing, ekstraksi fitur

menggunakan TF-IDF, hingga evaluasi model menggunakan metrik seperti akurasi, precision, recall, dan f1-score.

5.2 Saran

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

1. Penelitian ini hanya menggunakan 500 komentar dari satu video TikTok. Untuk meningkatkan ketepatan hasil yang lebih luas, disarankan agar penelitian selanjutnya menggunakan jumlah data yang lebih besar dan berasal dari berbagai sumber atau video yang berbeda.
2. Penelitian ini hanya fokus pada TikTok. Peneliti selanjutnya disarankan menerapkan pendekatan yang sama terhadap platform lain seperti Twitter, Instagram, atau YouTube guna mendapatkan pemahaman opini masyarakat dari berbagai media sosial.
3. Selain Support Vector Machine (SVM) dan Logistic Regression, penelitian selanjutnya dapat mencoba algoritma lain seperti Naive Bayes, Random Forest, atau model deep learning untuk membandingkan dan mengevaluasi performa dalam klasifikasi sentimen.

DAFTAR PUSTAKA

Amalia, N. A., Utami, I. T., & Wilandari, Y. (2024). Analisis Sentimen Kebijakan Penyelenggara Sistem Elektronik Lingkup Privat Menggunakan Penalized Logistic Regression Dan Support Vector Machine. *Jurnal Gaussian*, *12*(4), 560–569. https://doi.org/10.14710/j.gauss.12.4.560-569

Anshul. (2025). *Support Vector Machine (SVM)*. Analyticsvidhya. https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/?

Dadan Dahman W. (2021). *Support Vector Machine (SVM)*. Medium. https://medium.com/sysinfo/support-vector-machine-svm-5d95a7d7a547

Dr.Maria Susan Anggreany. (2020, November 1). *Confusion Matrix*. Binus. https://socs.binus.ac.id/2020/11/01/confusion-matrix/

Ferry Putrawansyah* dan Tri Susanti. (2024). Penerapan Metode Support Vector Machine Terhadap Klasifikasi Jenis Jambu Biji. *JIKO (Jurnal Informatika Dan Komputer)*, *8*(1), 193. https://doi.org/10.26798/jiko.v8i1.988

Irma Surya Kumala Idris, Yasin Aril Mustofa, & Irvan Abraham Salihi. (2023). Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Mengunakan Algoritma Support Vector Machine (SVM). *Journal of Information Science*, *5*(6), 823–848. https://doi.org/10.1177/0165551510388123

Isnanto, B. A. (2023). *Google Colab: Definisi, Kelebihan, dan Cara Menggunakan*. DetikInet. https://inet.detik.com/cyberlife/d-6992733/google-colab-definisi-kelebihan-dan-cara-menggunakan

Junifer Pangaribuan, J., Tanjaya, H., & Kenichi, K. (2021). Mendeteksi Penyakit Jantung Menggunakan Machine Learning Dengan Algoritma Logistic Regression. *Journal Information System Development (ISD)*, *06*(02), 1–10.

Kendrew Huang, E. P. P. (2022). *Support Vector Machine Algorithm*. Binus. https://sis.binus.ac.id/2022/02/14/support-vector-machine-algorithm/

Lidinillah, E. R., Rohana, T., & Juwita, A. R. (2023). Analisis sentimen twitter terhadap steam menggunakan algoritma logistic regression dan support vector machine. *TEKNOSAINS : Jurnal Sains, Teknologi Dan Informatika*, *10*(2),

154–164. https://doi.org/10.37373/tekno.v10i2.440

Lidwina, A. (2021). *Penggunaan E-Commerce Indonesia Tertinggi di Dunia*. Databoks. https://databoks.katadata.co.id/teknologi-telekomunikasi/statistik/b3c62c475783be9/penggunaan-e-commerce-indonesia-tertinggi-di-dunia

Liedfray, T., Waani, F. J., & Lasut, J. J. (2022). Peran Media Sosial Dalam Mempererat Interaksi Antar Keluarga Di Desa Esandom Kecamatan Tombatu Timur Kabupaten Tombatu Timur Kabupaten Minasa Tenggara. *Jurnal Ilmiah Society*, *2*(1), 2. https://ejournal.unsrat.ac.id/v3/index.php/jurnalilmiahsociety/article/download/38118/34843/81259

Muhammad Thoriq Al Fatih. (2024). *Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya*. Telkomuniversity. https://dif.telkomuniversity.ac.id/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/

Novantika, A. (2022). Analisis sentimen ulasan pengguna aplikasi video conference google meet menggunakan metode svm dan logistic regression. *PRISMA, Prosiding Seminar Nasional Matematika*, *5*, 808–813. https://journal.unnes.ac.id/sju/index.php/prisma/

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

Pratama, A. R., Wabula, F., Ilmandry, H., Isabela, M. L., & Sianipar, R. (2025). Literature Review The Impact of Machine Learning in Modern Industries. *Nian Tana Sikka : Jurnal Ilmiah Mahasiswa*, *3*(2021).

RB Fajriya Hakim. (2024, April 29). *SVM (Support Vector Machine)*. Medium. https://medium.com/@986110101/svm-support-vector-machine-6ee9fccd4222

Saif M. Mohammad. (2021). *Analisis Sentimen: Mendeteksi Valensi, Emosi, dan Keadaan Afektif Lainnya Secara Otomatis dari Teks*. Researchgate. https://www.researchgate.net/publication/350981895_Sentiment_analysis_Automatically_Detecting_Valence_Emotions_and_Other_Affectual_States_fro

m_Text

Simanjuntak, K., & Sari, R. P. (2023). Analisis Sistem S-Commerce pada Tiktok Shop untuk Meningkatkan Daya Saing Menggunakan Metode SWOT. *Jurnal Unitek, 16*(1), 1–6. https://doi.org/10.52072/unitek.v16i1.476

Storm_Scraper. (n.d.). *TikTok Comments Scraper*. https://console.apify.com/actors/Bg8VmZCJXlIPsQPpo/input

Supriyanto, A., Chikmah, I. F., Salma, K., & Tamara, A. W. (2023). Penjualan Melalui Tiktok Shop dan Shopee: Menguntungkan yang Mana? *BUSINESS: Scientific Journal of Business and Entrepreneurship*, *1*, 1–16. https://journal.csspublishing/index.php/business

Vijay Kanade. (2022). *Apa itu Regresi Logistik? Persamaan, Asumsi, Jenis, dan Praktik Terbaik*. Spiceworks. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

Wijoyo, A., Saputra, A. Y., Ristanti, S., Sya'ban, R., Amalia, M., & Febriansyah, R. (2024). Pembelajaran Machine Learning. *OKTAL : Jurnal Ilmu Komputer Dan Science*, *3*(2). https://journal.mediapublikasi.id/index.php/oktal/article/download/2305/2526/10017

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

Zunianantа, L. E. (2021). Penggunaan Media Sosial sebagai Media Komunikasi Informasi Di Perpustakaan. *Jurnal Ilmu Perpustakaan*, *10*(4), 37–42.

LAMPIRAN

Lampiran ini berisi script Python yang digunakan dalam analisis sentimen menggunakan algoritma Support Vector Machine (SVM) dan Logistic Regression.

1. Library yang digunakan

```
# Manipulasi data
import pandas as pd
import numpy as np

# Visualisasi
import matplotlib.pyplot as plt
import seaborn as sns

# Preprocessing
import re # untuk regex (pembersihan teks)
from nltk.tokenize import word_tokenize # untuk memecah teks menjadi token (kata-kata)
from nltk.corpus import stopwords # Untuk mengubah kata-kata umum yang tidak memiliki makna penting
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory # untuk mengubah kata berimbuhan menjadi kata

# Konversi teks ke angka
from sklearn.feature_extraction.text import TfidfVectorizer

# Split dan normalisasi
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler

# Machine Learning Models
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC

# Evaluasi
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

2. Data Celaning

```
import re

#Data Cleaning
def clean_text(text):
    text = text.lower()                         # huruf kecil semua
    text = re.sub(r'@\w+', '', text)            # hapus mention
    text = re.sub(r'[^\w\s]', '', text)         # hapus tanda baca
    text = re.sub(r'\s+', ' ', text).strip()    # hapus spasi berlebihan
    text = re.sub(r'(.)\1{2,}', r'\1', text)    # huruf berulang lebih dari 2 jadi 1
    text = re.sub(r'(.)\1{1}$', r'\1', text)    # Ubah sisa akhir yang dobel jadi 1
    text = re.sub(r'(.)\1+', r'\1', text)       # semua huruf berulang → satu huruf
    return text

df['cleaned_text'] = df['Text'].astype(str).apply(clean_text)

# Tampilkan hasil cleaning
pd.set_option('display.max_colwidth', None)  # agar kolom panjang terlihat penuh
df[['Text','UniqueId', 'Sentimen', 'cleaned_text']].head(10)  # tampilkan 10 baris pertama
```

3. Normalisasi

```
def normalize_slang(text):
    # Pisahkan kata-kata
    words = text.split()
    # Kamus slang
    slang_dict = {
    "tt": "tiktok", "t": "tiktok", "tts": "tiktok shop", "tktk": "tiktok", "tiktik": "tiktok", "gk": "gak", "ga": "gak", "g": "gak",
    "krna": "karena", "dgn": "dengan", "bgt": "banget", "syg": "sayang", "bnyk": "banyak", "pd": "pada", "belnja": "belanja", "d": "
    "jln": "jalan,", "bgtt": "banget", "shopie": "shoppe", "pdhl": "padahal", "d": "di", "y": "ya", "blnj": "belanja", "mhl": "mahal"
    "lbh": "lebih", "hrg": "harga", "sgt": "sangat", "psr": "pasar", "gu": "hrs", "bs": "bisa", "karna": "karena", "blnjaku": "belan
    "entar": "nanti","jdi": "jadi", "krn": "karena", "ad": "ada", "aduuh": "aduh", "agr": "agar", "aj": "aja", "ajja": "aja", "ajaga"
    "apl": "aplikasi", "aplgi": "apalagi", "apliksi": "aplikasi", "aq": "aku", "artis2": "artis", "artist": "artis", "ato": "atau", "t
    "app": "aplikasi", "3rts": "3 ratus", "akyu": "aku", "balkmenghindarin": "balk menghindarin", "balanja": "belanja", "barangnyala
    "barangxa": "barangnya", "baranya": "barangnya", "baring2": "baring", "bayak": "banyak", "belaja": "belanja", "belanjas": "belan
    "apik": "bagus", "berdesakkan": "berdesakan", "bgi": "bagi", "bgus": "bagus", "bikij": "buat", "bikin": "buat", "bismilah": "bism
    "blanj": "belanja", "blanjanya": "belanjanya", "blh": "boleh", "bljr": "belajar", "blm": "belum", "blnja": "belanja", "blnja2": "
    "oren": "shoppe", "ijo": "tokopedia", "ak": "aku", "aing": "aku", "ap": "apa", "bm": "minyak", "bnyak": "banyak", "bnyk": "banya
    "br": "baru", "brang": "barang", "brati": "berarti", "brg": "barang", "brg2": "barang", "brng": "barang", "brng2x": "barang", "br
    "byk": "banyak", "cangih": "canggih", "cape": "capek", "cepet": "capat", "bah": "apa", "gue": "aku", "dimonfatkan": "dimonfaatkan
}
    normalized_words = [slang_dict.get(word, word) for word in words]
    return ' '.join(normalized_words)

# Terapkan ke DataFrame
df['normalize_text'] = df['cleaned_text'].astype(str).apply(normalize_slang)

# Tampilkan hasilnya
pd.set_option('display.max_colwidth', None)
df[['cleaned_text', 'normalize_text']].head(10)
```

4. Tokenisasi

```
from nltk.tokenize import word_tokenize

# Tokenisasi (ubah menjadi daftar kata)
df['tokens'] = df['normalize_text'].astype(str).apply(word_tokenize)

pd.set_option('display.max_rows', None)
df[['normalize_text', 'tokens']]
```

5. Stopwords

```
import nltk
from nltk.corpus import stopwords

# Stopwords Removal (Bahasa Indonesia)
stop_words = set(stopwords.words('indonesian'))

def remove_stopwords(tokens):
    filtered = [word for word in tokens if word not in stop_words]
    return filtered

pd.set_option('display.max_colwidth', None)
df['no_stopwords'] = df['tokens'].apply(remove_stopwords)
```

6. Stemming

```
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Inisialisasi Stemmer
factory = StemmerFactory()
stemmer = factory.create_stemmer()

# Fungsi stemming
def stem_text(text):
    return stemmer.stem(str(text))  # pastikan teks dalam bentuk string

# Terapkan stemming ke kolom hasil stopwords removal (misalnya: 'no_stopwords')
df['stemmed_text'] = df['no_stopwords'].apply(stem_text)

# Tampilkan hasil
pd.set_option('display.max_colwidth', None)
df[['no_stopwords', 'stemmed_text']].head(10)
```

7. Labeling

```
#LABEL ENCODING✓

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['Sentimen'] = le.fit_transform(df['Sentimen'])  # pastikan kolomnya benar
# 0 = negatif, 1 = positif (bisa dicek lewat `le.classes_`)
```

8. Menghitung Jumlah Sentimen Positif & Negatif

```
# Hitung jumlah tiap jenis sentimen
print("Jumlah sentimen (berdasarkan teks):")
print(df['Sentimen'].value_counts())

# Hitung jumlah masing-masing sentimen
label_counts = df['Sentimen'].value_counts()
labels = ['Positif', 'Negatif']
jumlah = [label_counts[1], label_counts[0]]  # asumsi 1 = positif, 0 = negatif
colors = ['skyblue', 'lightcoral']

# Buat diagram batang
plt.figure(figsize=(6, 4))
plt.bar(labels, jumlah, color=colors)
plt.title('Distribusi Sentimen Komentar')
plt.xlabel('Kategori Sentimen')
plt.ylabel('Jumlah Komentar')
plt.grid(axis='y', linestyle='--', linewidth=0.5)
plt.show()
```

9. Pembagaian Data Latih dan Data Uji

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

# Memisahkan fitur (X) dan label (y)
X = df['stemmed_text']
y = df['Sentimen']

# Membagi data: 80% untuk latih, 20% untuk uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

10. Menghitung Jumlah Label Data Latih & Data Uji

```
# Hitung jumlah label di data latih
train_counts = y_train.value_counts().rename(index={1: 'Positif', 0: 'Negatif'})

# Hitung jumlah label di data uji
test_counts = y_test.value_counts().rename(index={1: 'Positif', 0: 'Negatif'})

# Gabungkan ke satu tabel untuk ditampilkan
summary_df = pd.DataFrame({
    'Data Latih': train_counts,
    'Data Uji': test_counts
}).fillna(0).astype(int)

print(summary_df)

# Visualisasi Bar Chart
summary_df.plot(kind='bar', color=['skyblue', 'salmon'])
plt.title('Distribusi Sentimen pada Data Latih dan Uji')
plt.ylabel('Jumlah')
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()

# Visualisai Data Latih
train_counts.plot.pie(autopct='%1.1f%%', startangle=140, colors=['lightgreen', 'lightcoral'])
plt.title('Distribusi Sentimen Data Latih')
plt.ylabel('')
plt.show()

# Visualisasi Data Uji
test_counts.plot.pie(autopct='%1.1f%%', startangle=140, colors=['lightblue', 'orange'])
plt.title('Distribusi Sentimen Data Uji')
plt.ylabel('')
plt.show()
```

11. TF-IDF

```
#Ekstraksi Fitur: TF-IDF✓

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=1000)
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

#Melihat data uji setelah TF-IDF✓

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

12. Implementasi Model Support Vestor Machine

```
# Import library
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Membuat model SVM
svm_model = SVC(kernel='linear')

# Melatih model menggunakan data latih
svm_model.fit(X_train_tfidf, y_train)

# Melakukan prediksi terhadap data uji
y_pred_svm = svm_model.predict(X_test_tfidf)

# Evaluasi hasil prediksi
print("\nClassification Report:")
print(classification_report(y_test, y_pred_svm))

print("\nAkurasi:")
print(accuracy_score(y_test, y_pred_svm))
```

13. Implementasi Model Logistic Regression

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

# Inisialisasi model
logreg = LogisticRegression()

# Latih model
logreg.fit(X_train_tfidf, y_train)

# Prediksi pada data uji
y_pred_logreg = logreg.predict(X_test_tfidf)

# Akurasi
print("Akurasi Logistic Regression:", accuracy_score(y_test, y_pred_logreg))

# Classification report
print("\nClassification Report:")
print(classification_report(y_test, y_pred_logreg, target_names=['Negatif:0', 'Positif:1']))
```

14. Evaluasi Model Support Vestor Machine

```
cm = confusion_matrix(y_test, y_pred_svm)

class_names = ['Negatif : 0', 'Positif : 1']
plt.figure(figsize=(6, 4))
sns.heatmap(pd.DataFrame(cm), annot=True, cmap="Blues", fmt='g', xticklabels=class_names, yticklabels=class_names)
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix - SVM')
plt.tight_layout()
plt.show()
```

15. Evaluasi Model Logistic Regression

```
# Confusion matrix
cm = confusion_matrix(y_test, y_pred_logreg)

sns.heatmap(cm, annot=True, fmt='d', cmap='Greens', xticklabels=['Negatif:0' , 'Positif:1'], yticklabels=['Negatif:0', 'Positif:1'])
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion matrix - Logistic Regression')
plt.show()

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

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

report_df_class[['precision', 'recall', 'f1-score']].plot(kind='bar', figsize=(8,4))
plt.title('Precision, Recall, F1-score per Class - Logistic Regression')
plt.ylim(0,1)
plt.ylabel('Score')
plt.xticks(rotation=45)
plt.legend(loc='lower right')
plt.tight_layout()
plt.show()
```

16. Hasil Perbandingan

```
#Perbandingan Hasil✓

from sklearn.metrics import f1_score

print("Akurasi Support Vector Machine:", accuracy_score(y_test, y_pred_svm))
print("F1-Score Support Vector Machine:", f1_score(y_test, y_pred_svm, average='weighted'))

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

17. Visualisasi Perbandingan

```
import matplotlib.pyplot as plt
import numpy as np

# Nilai evaluasi model
akurasi_svm = 0.89
f1_svm = 0.8577

akurasi_lr = 0.86
f1_lr = 0.7952

# Data untuk visualisasi
model = ['SVM', 'Logistic Regression']
akurasi = [akurasi_svm, akurasi_lr]
f1_score = [f1_svm, f1_lr]

x = np.arange(len(model))  # posisi label x
width = 0.35  # lebar batang

# Membuat bar chart
fig, ax = plt.subplots(figsize=(8, 6))
bars1 = ax.bar(x - width/2, akurasi, width, label='Akurasi', color='#3498DB')
bars2 = ax.bar(x + width/2, f1_score, width, label='F1-Score', color='#F39C12')
```

```
# Menambahkan keterangan
ax.set_ylabel('Nilai')
ax.set_xlabel('Model')
ax.set_title('Perbandingan Akurasi dan F1-Score\nSVM vs Logistic Regression')
ax.set_xticks(x)
ax.set_xticklabels(model)
ax.set_ylim(0, 1)  # nilai 0 sampai 1
ax.legend()

# Menambahkan label nilai di atas batang
for bar in bars1 + bars2:
    height = bar.get_height()
    ax.annotate(f'{height:.2f}',
                xy=(bar.get_x() + bar.get_width() / 2, height),
                xytext=(0, 3),  # offset atas
                textcoords="offset points",
                ha='center', va='bottom')

# Tampilkan plot
plt.tight_layout()
plt.show()
```