

# ANALISIS SENTIMEN APLIKASI X PADA GOOGLE PLAY STORE MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM)

Eskiyaturrofikoh<sup>1)</sup>, Ryan Randy Suryono<sup>\*2)</sup>

1. Universitas Teknokrat Indonesia, Indonesia
2. Universitas Teknokrat Indonesia, Indonesia

## Article Info

**Kata Kunci:** *Aplikasi X; Google PlayStore; Naïve Bayes; Support Vector Machine*

**Keywords:** *Application X; Google PlayStore; Naïve Bayes; Support Vector Machine*

## Article history:

Received 19 June 2024

Revised 27 July 2024

Accepted 7 August 2024

Available online 1 September 2024

## DOI :

<https://doi.org/10.29100/jupi.v9i3.5392>

\* Corresponding author.

Ryan Randy Suryono

E-mail address:

[ryan@teknokrat.ac.id](mailto:ryan@teknokrat.ac.id)

## ABSTRAK

Dalam era digital saat ini, *Google Play Store* telah menjadi salah satu platform terkemuka bagi pengguna Android untuk mengakses dan mengunduh berbagai aplikasi. Oleh karena itu, ulasan yang dipublikasikan di platform ini memberikan gambaran yang berharga tentang sentimen pengguna terhadap aplikasi tertentu. Tujuan penelitian ini adalah untuk menganalisis sentimen terhadap aplikasi X di *Google Play Store* dengan menggunakan dua metode klasifikasi yang berbeda, yakni *Naïve Bayes* dan *Support Vector Machine* (SVM). Dataset yang terdiri dari 4087 ulasan telah dikumpulkan dan dibagi menjadi dua bagian, yaitu data training (70%) dan data testing (30%). Hasil penelitian menunjukkan bahwa sebelum penerapan teknik SMOTE, akurasi SVM adalah 75,5%, sedangkan akurasi *Naïve Bayes* adalah 75%. Namun, setelah penerapan SMOTE, akurasi SVM meningkat menjadi 81%, sementara akurasi *Naïve Bayes* tetap pada 75,5%. Hasil ini menunjukkan bahwa penggunaan teknik SMOTE dapat meningkatkan kinerja model klasifikasi, terutama dalam hal mengenali sentimen positif dan negatif pada ulasan aplikasi. Analisis sentimen ini memberikan pemahaman yang lebih dalam tentang preferensi pengguna dan membantu pengembang aplikasi untuk meningkatkan pengalaman pengguna mereka dengan lebih baik.

## ABSTRACT

In the current digital era, *Google Play Store* has become one of the leading platforms for Android users to access and download various applications. Therefore, reviews published on this platform provide valuable insights into user sentiments towards specific applications. This study aims to conduct sentiment analysis on application X in the *Google Play Store* using two different classification methods, namely *Naïve Bayes* and *Support Vector Machine* (SVM). A dataset consisting of 4087 reviews has been collected and divided into two parts, namely training data (70%) and testing data (30%). The research results indicate that before the application of the SMOTE technique, the accuracy of SVM is 75.5%, while the accuracy of *Naïve Bayes* is 75%. However, after applying SMOTE, the accuracy of SVM increased to 81%, while the accuracy of *Naïve Bayes* remained at 75.5%. These results indicate that the use of the SMOTE technique can improve the classification model's performance, especially in identifying positive and negative sentiments in application reviews. This sentiment analysis provides a deeper understanding of user preferences and helps application developers enhance their users' experience more effectively.

## I. PENDAHULUAN

DALAM era digital yang terus berkembang penggunaan aplikasi *mobile* telah menjadi suatu kebutuhan dalam kehidupan sehari-hari, dengan berbagai macam aplikasi yang dapat diakses melalui berbagai *platform* yang tersedia [1]. *Google Play Store* merupakan *platform* terbesar untuk aplikasi android yang memberikan kemudahan akses kepada pengguna untuk mengunduh jutaan aplikasi yang tersedia. Dengan meningkatnya jumlah aplikasi, sangat penting untuk menganalisis sentimen terhadap ulasan pengguna semakin bertambah untuk memahami respons dan *feedback* yang diberikan oleh masyarakat [2]. Analisis sentimen adalah metode yang digunakan untuk mengidentifikasi dan memahami perasaan atau pendapat yang terdapat dalam

teks. Ulasan pengguna merupakan sumber informasi yang penting karena dapat memberikan pemahaman tentang tingkat kepuasan atau ketidakpuasan pengguna terhadap suatu aplikasi [3].

Google Play Store menyediakan berbagai aplikasi untuk pengguna untuk diunduh. Penilaian pengguna dan ulasan sangat penting untuk menentukan popularitas dan penerimaan sebuah aplikasi. *Google Play Store* telah menjadi forum diskusi di antara pengguna Aplikasi X karena perubahan Namanya. Pada 27 Oktober 2022, Elon Musk resmi menjadi pemilik Twitter, yang kemudian berganti nama menjadi X Corp. Pada 23 Juli 2023, Musk melakukan rebranding aplikasi, mengubah logo dari yang sebelumnya berlogo burung dan merubah nama aplikasi menjadi X. Keunikan dari aplikasi X adalah kepemilikan oleh Elon Musk dan potensi perubahan yang signifikan dalam strategi dan fitur aplikasi. Dikenal sebagai tokoh yang inovatif dan ambisius, Elon Musk menciptakan ekspektasi akan transformasi menarik di bawah kepemilikan barunya. Meskipun tujuannya tetap sama dengan Twitter, yaitu memberikan pengguna platform untuk berinteraksi dan berdiskusi, kepemilikan baru oleh Elon Musk dan rebranding menjadi X Corp membawa harapan akan pengalaman pengguna yang lebih baik dan fitur-fitur yang lebih inovatif. Analisis sentimen terhadap ulasan pengguna di Google Play Store bertujuan untuk memahami pandangan pengguna terhadap pembaruan aplikasi ini dan tingkat kepuasan mereka terhadap perubahan yang diperkenalkan oleh X Corp di bawah kepemilikan Elon Musk. Perubahan aplikasi ini menghasilkan banyak ulasan pada media sosial X di Google Play Store yang menggambarkan sentimen pengguna terhadap berbagai aspek aplikasi, termasuk kegunaan, kinerja, fitur, dan dukungan pengguna.

Penelitian ini menggunakan metode SVM untuk menentukan aplikasi novel online terbaik di Google Play Store berdasarkan 4.137 ulasan pengguna. Akurasi yang diperoleh adalah 88,60%. *Dream* unggul dibandingkan *Wattpad* berdasarkan ulasan pengguna, dengan 26,12% pengguna sangat menyukai *Dream* dari 3.090 ulasan, sedangkan *Wattpad* hanya mendapat 22,82% dari 4.137 ulasan. Hal ini menjadi relevan terutama di masa pandemi COVID-19, di mana banyak orang mencari hiburan dan mengisi waktu luang dengan membaca novel secara online [4]. Ulasan aplikasi MOLA di sebuah platform streaming video di *Google Play Store* dengan 5 juta unduhan dan rating 3,2, menjadi subjek analisis sentimen menggunakan algoritma SVM. Dari 520 ulasan, terbagi menjadi 312 positif dan 208 negatif. Hasil terbaik, dengan skenario 1 (90:10) dan kernel RBF, mencapai akurasi 92,31%, presisi 96,3%, recall 89,66%, dan f1-score 92,86% [5]. Analisis sentimen terkait Covid-19 menggunakan 2 metode yaitu *Support Vector Machine* (SVM) dan *Naïve Bayes*. Penelitian ini menunjukkan bahwa SVM mencapai tingkat akurasi yang lebih tinggi, yakni sebesar 88,8%, sedangkan *Naïve Bayes* memiliki akurasi sebesar 82,51% [6]. Pada penelitian membahas tentang analisis sentimen untuk *Google Play Store* menunjukkan bahwa penggunaan metode *Naïve Bayes* dan *Support Vector Machine* (SVM) menghasilkan akurasi yang berbeda. Metode SVM memberikan tingkat akurasi lebih tinggi, mencapai 89,49%, sementara metode *Naïve Bayes* mencatat akurasi sebesar 83,87% [7]. *Support Vector Machine* (SVM) adalah algoritma pembelajaran mesin yang memisahkan kelas dengan menciptakan garis atau hiperplane dengan margin maksimum, efektif untuk analisis sentimen dengan fitur-fitur kompleks; sementara *Naïve Bayes*, sebuah metode klasifikasi probabilistik, menggunakan probabilitas untuk menentukan kelas sentimen, sering digunakan dalam kasus analisis sentimen dengan dataset besar karena kecepatan pelatihan dan klasifikasi yang cepat [8]. Meskipun SVM memiliki kelebihan dalam menangani data dengan dimensi tinggi, ia cenderung memerlukan waktu yang lebih lama untuk melatih model, sedangkan *Naïve Bayes*, meskipun sederhana dan mudah diimplementasikan, mengasumsikan independensi antara fitur-fiturnya yang mungkin tidak selalu berlaku dalam analisis sentimen yang kompleks [9]. Dalam konteks analisis sentimen aplikasi, pemilihan antara kedua metode ini tergantung pada kompleksitas data, kebutuhan kecepatan, dan interpretasi model yang diinginkan.

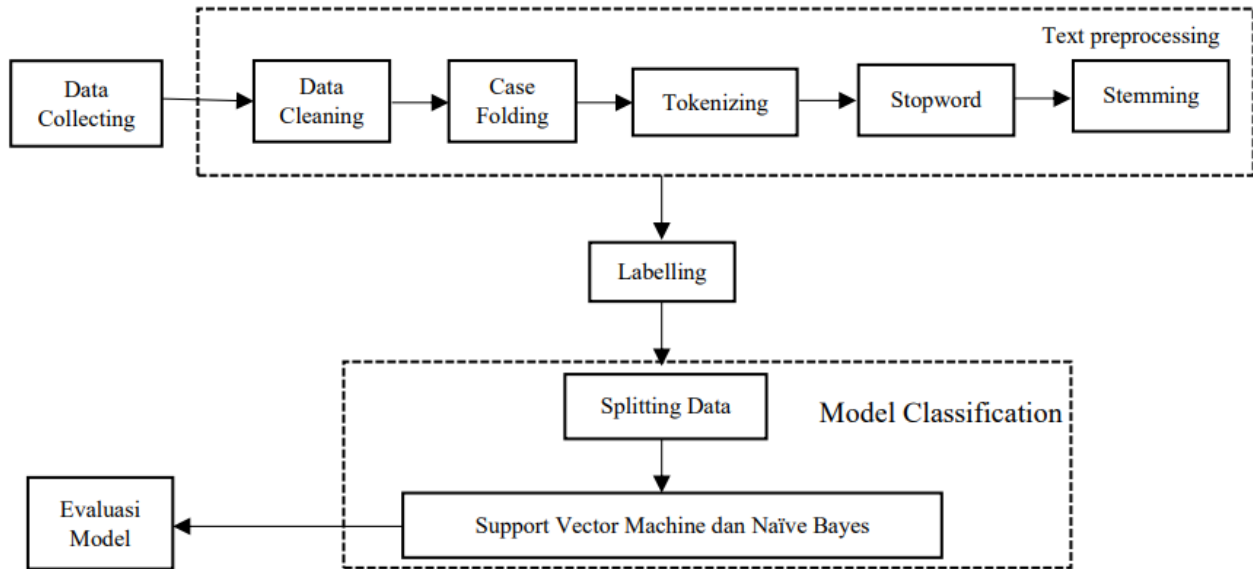
Berdasarkan latar belakang masalah tersebut, Penelitian ini bertujuan membandingkan kinerja algoritma *Support Vector Machine* (SVM) dan *Naïve Bayes* dalam menganalisis sentimen ulasan pengguna Aplikasi X di Google Play Store. Dengan memperbandingkan kedua metode, diharapkan dapat mengidentifikasi kelebihan dan kelemahan masing-masing, serta memberikan wawasan bagi pengembang aplikasi untuk meningkatkan pengalaman pengguna dan kualitas produk di pasar yang kompetitif. Data diambil dari *Google Play Store*, dan hasil penelitian akan disajikan dalam laporan yang menyeluruh, menyediakan pemahaman yang lebih baik tentang efektivitas klasifikasi sentimen dalam konteks aplikasi mobile.

## II. METODE PENELITIAN

### A. Metode Penelitian

Dalam penelitian ini dibagi menjadi beberapa tahap, yaitu pengumpulan data, dilanjutkan dengan pra-pemrosesan data, dan kemudian menerapkan teknik SMOTE, lalu data dibagi menjadi data uji dan data latih untuk

selanjutnya tahap klasifikasi model menggunakan algoritma *Support Vector Machine* (SVM) dan *Naïve Bayes* [10]. Tahap terakhir yaitu Evaluasi model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score* untuk membandingkan performakedua algoritma dalam menganalisis sentimen pada platform *Google PlayStore* terkait Aplikasi X. Gambar 1 merupakan ilustrasi dari alur penelitian ini.



Gambar 1. Tahapan Penelitian

### B. Data Collecting

Dataset diambil melalui API Google Play Store dengan menggunakan metode crawling data menggunakan bahasa pemrograman Python, dimana tujuannya adalah untuk mengumpulkan tweet yang mengandung kata kunci "aplikasi X". Sebanyak 4087 tweet berhasil dikumpulkan dalam rentang waktu antara tahun 2023 dan 2024. Dataset ini berisi teks berbahasa Indonesia yang akan digunakan untuk melakukan analisis sentimen terhadap aplikasi X di Google Play Store menggunakan metode *Naïve Bayes* dan *Support Vector Machine* (SVM). Evaluasi dilakukan dengan membagi dataset menjadi 70% untuk pelatihan (training) dan 30% untuk pengujian (testing), yang akhirnya menunjukkan peningkatan signifikan dalam akurasi model klasifikasi, terutama pada metode SVM, setelah penerapan teknik SMOTE. Setelah pengumpulan data selesai, data tersebut disimpan dalam format CSV untuk kemudian dapat dianalisis lebih lanjut.

### C. Data Preprocessing

Pengumpulan data dari Twitter melibatkan proses ekstraksi informasi tentang pengguna dan tweet dari platform Twitter, dengan menggunakan kata kunci tertentu. Ekstraksi data ini dilakukan melalui pemanfaatan *Application Programming Interface* (API). Dataset untuk penelitian ini diperoleh melalui penerapan teknik web scraping dengan menggunakan kata kunci "Aplikasi X". Pengumpulan data dilakukan dengan memanfaatkan teknik scraping data, menggunakan library tweet yang tersedia dalam bahasa pemrograman Python. Sejumlah 4087 data berhasil terkumpul. Dataset tersebut kemudian dilakukan preprocessing, seperti membersihkan data, menghapus karakter yang tidak perlu, dan pelabelan data.

#### 1) Cleaning Data

*Cleaning* bertujuan untuk menghilangkan karakter atau simbol seperti tautan URL, nama pengguna atau mention (@username), hashtag (#), retweet, dan emotikon dari teks.

Tabel 1. *Cleaning Data*

| <i>Tweet</i> | <i>Cleaning Data</i> |
|--------------|----------------------|
|--------------|----------------------|

|   |   |
|---|---|
| <p>Aplikasi Sosial Media yg sangat berfaedah bagi Saya. Karena saya sebagai konsumen beberapa layanan sangat terbantu untuk terhubung dengan penyedia layanan dgn pengalaman yg jauh lebih memudahkan, sederhana, responsif &amp; cepat, demi solusi permasalahan yang Saya hadapi untuk seger ditindaklanjuti ke depannya.</p>   | <p>Aplikasi Sosial Media yg sangat berfaedah bagi Saya Karena saya sebagai konsumen beberapa layanan sangat terbantu untuk terhubung dengan penyedia layanan dgn yg jauh lebih sederhana responsif cepat demi solusi yang Saya hadapi untuk seger ke depannya</p> |
| <p>Ini twittet knapa jadi lemot dan gak respon..? Setiap buka aplilasi ini pasti lemot gak muncul tweet baru ditimeline cm ada tweet lama yg sebelumnya, direfresh berkali2pun gak ada responnya, masa iya harus logout login terus..? seharusnya pembaharuan itu mempermudah bukan malah mempersulit dan bikin ribet... tolong diperhatikan demi kenyamanan bersama.... Terima kasih..</p> | <p>Ini twittet knapa jadi lemot dan gak respon Setiap buka aplilasi ini pasti lemot gak muncul tweet baru cm ada tweet lama yg direfresh gak ada responnya masa iya harus logout login terus itu bukan malah dan bikin ribet tolong demi bersama Terima kasih</p> |

## 2) Case Folding

Case folding mengubah semua teks dalam dokumen menjadi huruf kecil karena tidak semua teks memiliki keseragaman dalam penggunaan huruf kapital. Proses ini memperhatikan sensitivitas terhadap huruf besar dan kecil.

Tabel 2. Case Folding

| <i>Tweet</i>  | <i>Case Folding</i>   |
|---|---|
| <p>Aplikasi Sosial Media yg sangat berfaedah bagi Saya Karena saya sebagai konsumen beberapa layanan sangat terbantu untuk terhubung dengan penyedia layanan dgn yg jauh lebih sederhana responsif cepat demi solusi yang Saya hadapi untuk seger ke depannya</p> | <p>aplikasi sosial media yg sangat berfaedah bagi saya karena saya sebagai konsumen beberapa layanan sangat terbantu untuk terhubung dengan penyedia layanan dgn yg jauh lebih sederhana responsif cepat demi solusi yang saya hadapi untuk seger ke depannya</p> |
| <p>Ini twittet knapa jadi lemot dan gak respon Setiap buka aplilasi ini pasti lemot gak muncul tweet baru cm ada tweet lama yg direfresh gak ada responnya masa iya harus logout login terus itu bukan malah dan bikin ribet tolong demi bersama Terima kasih</p> | <p>ini twittet knapa jadi lemot dan gak respon setiap buka aplilasi ini pasti lemot gak muncul tweet baru cm ada tweet lama yg direfresh gak ada responnya masa iya harus logout login terus itu bukan malah dan bikin ribet tolong demi bersama terima kasih</p> |

## 3) Tokenizing

Setelah proses pembersihan, tahapan selanjutnya adalah *tokenizing*. Pada tahap ini, data akan diproses dengan menghilangkan tanda baca sehingga menghasilkan kumpulan kata atau kalimat yang berdiri sendiri.

Tabel 3. Tokenizing

| <i>Tweet</i>  | <i>Tokenizing</i>   |
|---|---|
| <p>aplikasi sosial media yg sangat berfaedah bagi saya karena saya sebagai konsumen beberapa layanan sangat terbantu untuk terhubung dengan penyedia layanan dgn yg jauh lebih sederhana responsif cepat demi solusi yang saya hadapi untuk seger ke depannya</p> | <p>['aplikasi', 'sosial', 'media', 'yg', 'sangat', 'berfaedah', 'bagi', 'saya', 'karena', 'saya', 'sebagai', 'konsumen', 'beberapa', 'layanan', 'sangat', 'terbantu', 'untuk', 'terhubung', 'dengan', 'penyedia', 'layanan', 'dgn', 'yg', 'jauh', 'lebih', 'sederhana', 'responsif', 'cepat', 'demi', 'solusi', 'yang', 'saya', 'hadapi', 'untuk', 'seger', 'ke', 'depannya']</p>                     |
| <p>ini twittet knapa jadi lemot dan gak respon setiap buka aplilasi ini pasti lemot gak muncul tweet baru cm ada tweet lama yg direfresh gak ada responnya masa iya harus logout login terus itu bukan malah dan bikin ribet tolong demi bersama terima kasih</p> | <p>['ini', 'twittet', 'knapa', 'jadi', 'lemot', 'dan', 'gak', 'respon', 'setiap', 'buka', 'apilasi', 'ini', 'pasti', 'lemot', 'gak', 'muncul', 'tweet', 'baru', 'cm', 'ada', 'tweet', 'lama', 'yg', 'direfresh', 'gak', 'ada', 'responnya', 'masa', 'iya', 'harus', 'logout', 'login', 'terus', 'itu', 'bukan', 'malah', 'dan', 'bikin', 'ribet', 'tolong', 'demi', 'bersama', 'terima', 'kasih']</p> |

#### 4) Stopword

Tahapan *stopword* bertujuan untuk menghapus kata-kata umum yang sering muncul dalam jumlah besar dan tidak memiliki makna penting. Hal ini dilakukan menggunakan *algoritma stoplist* untuk membuang kata-kata yang kurang relevan atau menggunakan *wordlist* untuk menyimpan kata-kata yang penting.

Tabel 4. *StopWord*

| <i>Tweet</i>   | <i>Stop Word</i>  |
|--|---|
| [ 'aplikasi', 'sosial', 'media', 'yg', 'sangat', 'berfaedah', 'bagi', 'saya', 'karena', 'saya', 'sebagai', 'konsumen', 'beberapa', 'layanan', 'sangat', 'terbantu', 'untuk', 'terhubung', 'dengan', 'penyedia', 'layanan', 'dgn', 'yg', 'jauh', 'lebih', 'sederhana', 'responsif', 'cepat', 'demi', 'solusi', 'yang', 'saya', 'hadapi', 'untuk', 'seger', 'ke', 'depannya']                      | [ 'aplikasi', 'sosial', 'media', 'yg', 'berfaedah', 'konsumen', 'layanan', 'terbantu', 'terhubung', 'penyedia', 'layanan', 'dgn', 'yg', 'sederhana', 'responsif', 'cepat', 'solusi', 'yang', 'hadapi', 'seger', 'depannya']           |
| [ 'ini', 'twwitet', 'knapa', 'jadi', 'lemot', 'dan', 'gak', 'respon', 'setiap', 'buka', 'apiliasi', 'ini', 'pasti', 'lemot', 'gak', 'muncul', 'tweet', 'baru', 'cm', 'ada', 'tweet', 'lama', 'yg', 'direfresh', 'gak', 'ada', 'responnya', 'masa', 'iya', 'harus', 'logout', 'login', 'terus', 'itu', 'bukan', 'malah', 'dan', 'bikin', 'ribet', 'tolong', 'demi', 'bersama', 'terima', 'kasih'] | [ 'twwitet', 'knapa', 'lemot', 'gak', 'respon', 'buka', 'apiliasi', 'lemot', 'gak', 'muncul', 'tweet', 'cm', 'tweet', 'yg', 'direfresh', 'gak', 'responnya', 'iya', 'logout', 'login', 'bikin', 'ribet', 'tolong', 'terima', 'kasih'] |

#### 5) Stemming

Dalam proses *stemming*, setiap kata akan diubah menjadi bentuk dasarnya dengan menghilangkan imbuhan. *Stemming* yang digunakan untuk melakukan *stemming* adalah *library sastrawi*.

Tabel 5. *Stemming*

| <i>Tweet</i>  | <i>Stemming</i>  |
|---|--|
| [ 'aplikasi', 'sosial', 'media', 'yg', 'berfaedah', 'konsumen', 'layanan', 'terbantu', 'terhubung', 'penyedia', 'layanan', 'dgn', 'yg', 'sederhana', 'responsif', 'cepat', 'solusi', 'yang', 'hadapi', 'seger', 'depannya']           | aplikasi sosial media yg faedah konsumen layan bantu hubung sedia layan dgn yg sederhana responsif cepat solusi yang hadap seger depan               |
| [ 'twwitet', 'knapa', 'lemot', 'gak', 'respon', 'buka', 'apiliasi', 'lemot', 'gak', 'muncul', 'tweet', 'cm', 'tweet', 'yg', 'direfresh', 'gak', 'responnya', 'iya', 'logout', 'login', 'bikin', 'ribet', 'tolong', 'terima', 'kasih'] | twwitet knapa lot gak respon buka apiliasi lot gak muncul tweet cm tweet yg direfresh gak responnya iya logout login bikin ribet tolong terima kasih |

#### D. Labelling

Pada tahap ini, dataset akan diberi label positif dan negatif. Penentuan label dilakukan otomatis menggunakan library Python yang memberikan skor pada setiap sentimen terkait komentar di *Google Play Store*. Proses ini memungkinkan pengklasifikasian dataset berdasarkan sentimen yang terkandung dalam komentar, memudahkan analisis dan penggunaan data tersebut dalam pembangunan model.

#### E. Handle Imbalance

*Handle imbalance* adalah istilah pada situasi ketidakseimbangan antara jumlah sampel di setiap kelas atau kategori yang akan diprediksi dalam konteks klasifikasi atau pembelajaran mesin [11]. Ketidakseimbangan kelas adalah masalah dalam analisis sentimen karena dapat menyebabkan *bias* (kecenderungan) dalam model. Dengan kelas mayoritas mendominasi, model cenderung kurang sensitif terhadap sentimen yang kurang umum atau jarang terjadi, yang akhirnya mengurangi kemampuan model untuk mengenali dan memprediksi sentimen dengan akurat. Hal ini dapat menghasilkan hasil yang tidak seimbang atau tidak dapat diandalkan, mengurangi efektivitas analisis sentimen dalam memahami pandangan pengguna atau tren yang mungkin ada dalam data. Oleh karena itu, penanganan ketidakseimbangan kelas menjadi penting untuk memastikan bahwa model dapat mengenali dan mengatasi dengan benar sentimen dari semua kelas dalam dataset [12]. SMOTE (*Synthetic Minority Over-sampling Technique*) adalah teknik yang digunakan untuk mengatasi masalah ketidakseimbangan kelas ini. SMOTE membantu meningkatkan representasi kelas minoritas dengan menciptakan sampel sintesis baru dari kelas tersebut, yang dihasilkan dengan menggabungkan atau menginterpolasi sampel yang sudah ada. Teknik ini membantu mempertahankan distribusi kelas mayoritas sambil meningkatkan keseimbangan dataset

secara keseluruhan [13]. Dengan demikian, penggunaan SMOTE dapat meningkatkan kinerja model dalam mengidentifikasi kelas minoritas.

#### F. Splitting Data

Dataset telah dibagi menjadi dua bagian: data training dan data uji. Pembagian dilakukan dengan rasio 70% untuk data training dan 30% untuk data testing. Dengan total dataset berjumlah 4087, ini berarti 70% dari total data, yaitu 2860 data training, akan digunakan untuk melatih model, sedangkan 30% sisanya, yaitu 1227 data testing, akan digunakan untuk menguji performa model. Proses pembagian ini penting untuk menguji keandalan dan konsistensi model dalam menangani data baru.

#### G. Model Classification

Penerapan model klasifikasi melibatkan dua algoritma utama, yaitu *Support Vector Machine* (SVM) dan *Naïve Bayes*. Proses klasifikasi ini ditujukan untuk memperoleh tingkat akurasi dari kedua model tersebut, yang kemudian akan dibandingkan untuk mengevaluasi kinerja relatif masing-masing algoritma [14]. Dengan demikian, pemilihan model yang paling sesuai dengan data dan tujuan analisis dapat dilakukan dengan lebih baik.

##### 1. Support Vector Model (SVM)

*Support Vector Machine* (SVM) adalah sebuah algoritma pembelajaran mesin yang digunakan untuk tugas-tugas klasifikasi dan regresi. SVM berfungsi dengan cara membangun sebuah *hyperplane* dalam sebuah ruang berdimensi tinggi yang memisahkan instance-instance dari kelas yang berbeda [15]. *Hyperplane* ini dipilih sedemikian rupa sehingga jarak antara *hyperplane* dan instance terdekat dari setiap kelas (yang disebut sebagai *support vectors*) maksimal. SVM sangat efektif dalam menangani dataset dengan banyak fitur, dan memiliki keunggulan dalam menangani data yang tidak terstruktur atau memiliki kompleksitas tinggi [16]. Algoritma ini digunakan secara luas dalam berbagai aplikasi, termasuk klasifikasi teks, pengenalan wajah, pengenalan tulisan tangan, dan banyak lagi. Dalam *Support Vector Machine*, rumus untuk memprediksi kelas dari suatu sampel  $x$  menggunakan persamaan berikut.

$$f(x) = \text{sign}(w \cdot x + b) \quad (1)$$

Dimana

- $f(x)$  adalah fungsi keputusan yang mengklasifikasikan sampel  $x$
- $w$  adalah vektor bobot
- $x$  adalah vector fitur input
- $b$  adalah bias
- $\text{sign}(\cdot)$  adalah fungsi tanda yang menghasilkan label kelas

##### 2. Naive Bayes

*Naive Bayes classifier* adalah algoritma klasifikasi yang didasarkan pada teorema Bayes dengan asumsi independensi antara fitur-fitur yang diberikan. Ini adalah pendekatan yang cepat dan sederhana untuk klasifikasi yang sering digunakan dalam pemrosesan teks, klasifikasi email spam, dan sistem rekomendasi [17]. *Naive Bayes* memprediksi kelas atau label untuk data baru dengan menghitung probabilitas kelas berdasarkan distribusi fitur-fitur yang diamati. Dengan asumsi independensi antar fitur, algoritma ini menghitung probabilitas kelas dengan menggunakan *teorema Bayes* [18]. Meskipun sederhana, *Naive Bayes* dapat memberikan hasil yang baik dalam banyak situasi dan sering digunakan sebagai baseline untuk membandingkan kinerja model klasifikasi yang lebih kompleks.

$$P(X|Y) = \frac{P(Y|X) \times P(X)}{P(Y)} \quad (2)$$

$P(X|Y)$  = probabilitas kondisional dari X diberikan Y

$P(Y|X)$  = probabilitas kondisional dari Y diberikan X

$P(X)$  = probabilitas prior dari X

$P(Y)$  = probabilitas prior dari Y

### H. Evaluasi Model

Evaluasi model dilakukan untuk menentukan kinerja proses dengan menggunakan confusion matrix. Confusion matrix adalah cara untuk menilai seberapa baik model pembelajaran mesin dalam mengklasifikasikan data. Matriks ini terdiri dari tabel 2x2 yang memberikan informasi tentang kebenaran dan ketidakbenaran prediksi yang dihasilkan oleh model [19]. Dengan confusion matrix, kita dapat melihat seberapa baik model memprediksi kelas positif dan negatif, serta seberapa sering model melakukan kesalahan prediksi. Tabel 2 berisi keterangan untuk setiap elemen dalam *Confusion Matrix*:

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (3)$$

$$Presisi = \frac{TP}{TP+FP}$$

(4)

$$Recall = \frac{TP}{TP+FN}$$

(5)

$$f1 - score = \frac{2 \times Presisi \times Recall}{Presisi + Recall}$$

(6)

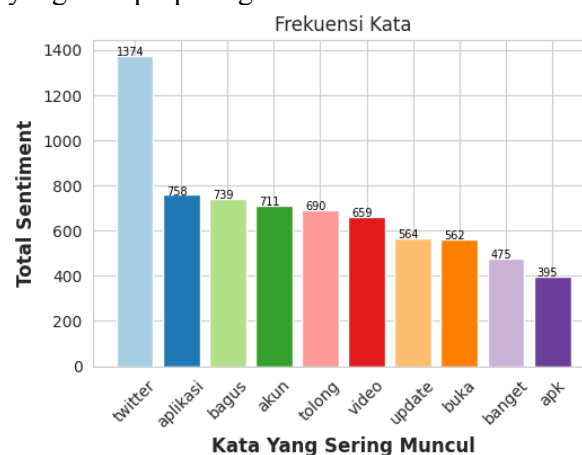
Keterangan:

- *True Positive (TP)* : Jumlah data yang berhasil diidentifikasi dengan benar sebagai positif oleh model.
- *False Positive (FP)* : Jumlah data yang model salah mengklasifikasikan sebagai positif
- *True Positive (TP)* : Jumlah data yang berhasil diidentifikasi dengan benar sebagai negative oleh model
- *True Positive (TP)* : Jumlah data yang model salah mengklasifikasikan sebagai negative

## III. HASIL DAN PEMBAHASAN

### A. Dataset

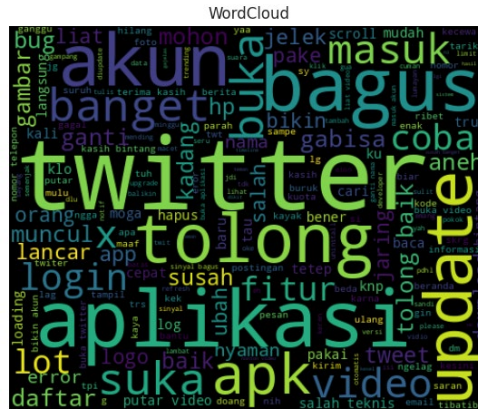
Dataset diperoleh dari pengguna pada Google Play Store melalui proses crawling, menghasilkan sebanyak 4087 tweets. Selanjutnya, dilakukan tahap preprocessing teks pada dataset, diikuti dengan pelabelan atau pembuatan kelas. Dalam data tersebut, terdapat 10 kata dengan frekuensi tertinggi, antara lain "twitter", "aplikasi", "bagus", "akun", "tolong", "video", "update", "buka", "banget", dan "apk". Kalimat sebelumnya dapat memiliki sentimen positif atau negatif tergantung pada konteks penggunaannya dalam ulasan. Misalnya, kata "bagus" dan "banget" cenderung menyatakan kepuasan, sementara "tolong" menunjukkan ketidakpuasan atau permintaan bantuan. Dengan memahami distribusi dan konteks penggunaan kata-kata ini, dapat memberikan pemahaman yang lebih baik tentang karakteristik dataset dan sentimen umum terhadap Aplikasi X di Google Play Store. Berikut adalah frekuensi kata yang terdapat pada gambar 2.



Gambar 2. Frekuensi Kata

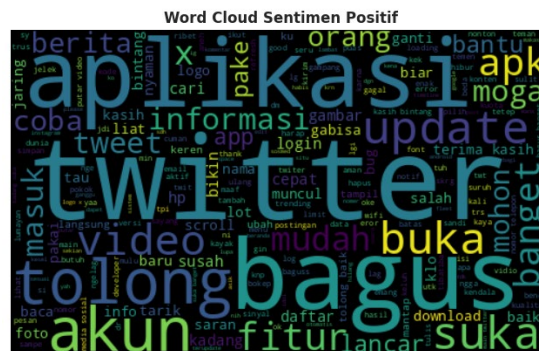
## B. Tahap Visualisasi *Word Cloud*

Penggunaan *Word Cloud* bertujuan untuk memberikan visualisasi tentang dataset, memungkinkan pengidentifikasian kata-kata yang paling sering muncul dalam dokumen. Proses pembuatan *WordCloud* dilakukan dengan menggunakan library *matplotlib* dalam bahasa pemrograman Python. Gambar 6 merupakan hasil visualisasi *WordCloud*. Dari hasil visualisasi pada Gambar 3, terlihat bahwa kata "Aplikasi X" memiliki ukuran yang besar, menunjukkan bahwa kata tersebut sering muncul dalam dataset.



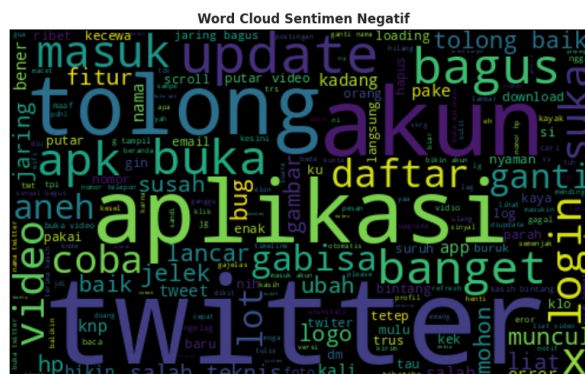
Gambar 3. WordCloud Semua Sentiment

Pada word cloud sentimen positif, beberapa kata yang sering muncul adalah "twitter", "aplikasi", "bagus", "tolong", "video", "mudah", "akun", "suka", "update", dan "apk". Ini menunjukkan bahwa dalam ulasan positif, pengguna cenderung menyebutkan aspek-aspek seperti kualitas aplikasi, kemudahan penggunaan, dukungan yang diberikan, serta kepuasan terhadap pembaruan dan fitur-fitur baru.



Gambar 4. WordCloud Sentiment Positif

Pada word cloud sentimen positif, beberapa kata yang sering muncul adalah "twitter", "aplikasi", "buka", "tolong", "daftar", "gabis", "banget", "masuk", "update", dan "bagus". Ini menunjukkan bahwa dalam ulasan positif, pengguna cenderung menyebutkan aspek-aspek seperti kemudahan akses, permintaan bantuan atau dukungan, pengalaman pendaftaran yang lancar, serta kepuasan terhadap pembaruan dan kualitas aplikasi secara umum.

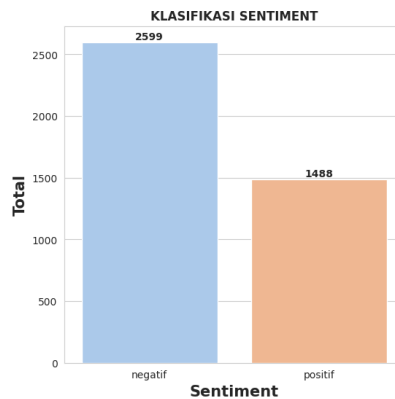


Gambar 5. WordCloud Sentiment Negatif



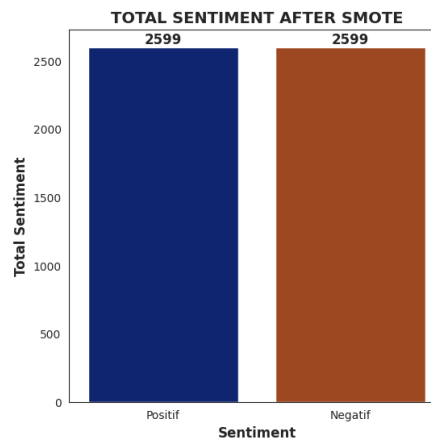
### C. Tahap Pengujian

Penelitian yang dilakukan oleh Dewi dan Mailoa (2023) mengenai klasifikasi sentimen dari opini masyarakat di media sosial Twitter tentang Mixue menunjukkan bahwa analisis menggunakan teknik SMOTE dan algoritma SVM meningkatkan hasil akurasi sebesar 4,27% menjadi 73,67% [20]. Dalam penelitian ini, terdeteksi ketidakseimbangan kelas, sehingga diterapkan metode SMOTE untuk menangani perbedaan jumlah antar kelas. Langkah ini diambil untuk memastikan bahwa model dapat belajar dari kelas minoritas dengan lebih baik dan menghindari bias yang mungkin timbul akibat ketidakseimbangan tersebut. Setelah melalui tahap pre-processing dan penerapan SMOTE, jumlah dataset bertambah menjadi 5198 tweets. Klasifikasi sentiment negatif sebanyak 2599 data dan sentiment positif sebanyak 1488 data. Perbandingan antara data sebelum penerapan SMOTE dan setelahnya dapat dilihat pada Gambar 6 dan Gambar 7.



Gambar 6. Jumlah klasifikasi sentiment sebelum menggunakan SMOTE

Setelah menerapkan teknik SMOTE maka jumlah data dalam kategori sentiment positif dan negatif berhasil disamakan menjadi 2599 data.



Gambar 7. Jumlah klasifikasi sentiment setelah menggunakan SMOTE

### D. Klasifikasi Model

Dalam perbandingan kinerja SVM dan Naive Bayes dalam analisis sentimen, SVM memiliki keunggulan dalam kemampuannya untuk mengatasi masalah klasifikasi dalam ruang berdimensi tinggi dan memiliki fleksibilitas dalam penggunaan fungsi kernel untuk menangani data yang kompleks. Namun, kelemahannya termasuk rentan terhadap overfitting pada dataset yang kompleks dan sulitnya interpretasi model yang dihasilkan. Di sisi lain, Naive Bayes memiliki kelebihan dalam ketahanan terhadap overfitting dan kemampuan interpretasi yang tinggi karena asumsi sederhana yang digunakan. Namun, kelemahannya adalah kurang mampu menangani hubungan yang kompleks antara fitur dan label, karena asumsi bahwa semua fitur independen satu sama lain. Dalam penelitian ini komparasi dilakukan menggunakan dua metode, yaitu *Support Vector Machine* dan *Naive Bayes*. Sebanyak 70% dari data digunakan sebagai data training, sementara sisanya 30% digunakan sebagai data testing. Dalam tahap pengujian, nilai accuracy, F1-score, precision, recall, dan Confusion Matrix akan dibandingkan antara kedua metode untuk mengevaluasi. Hasil pengujian confusion matrix sebelum SMOTE dapat dilihat pada tabel 7, dan sesudah SMOTE pada tabel 8.

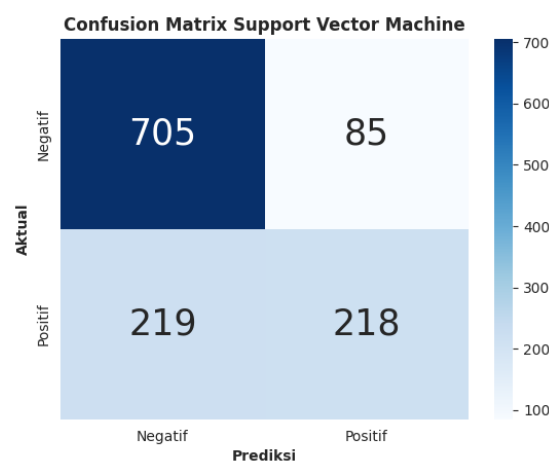
Tabel 7. Hasil Confusion Matrix Sebelum SMOTE

| Model       | Accuracy | Precision | Recall | f1-score |
|-------------|----------|-----------|--------|----------|
| SVM         | 75,5%    | 74%       | 69,5%  | 70,5%    |
| Naïve Bayes | 75%      | 76%       | 76%    | 75,5%    |

Tabel 8. Hasil Confusion Matrix Sesudah SMOTE

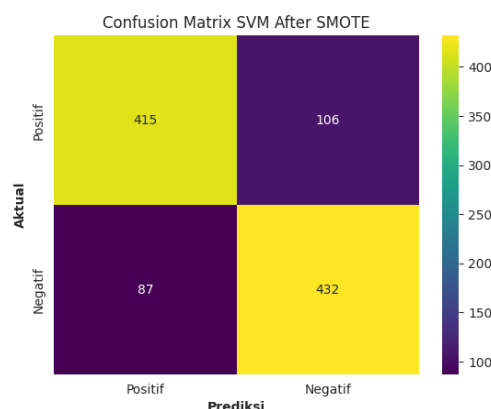
| Model       | Accuracy | Precision | Recall | f1-score |
|-------------|----------|-----------|--------|----------|
| SVM         | 81%      | 81,5%     | 81%    | 81,5%    |
| Naïve Bayes | 75,5%    | 76%       | 76%    | 75,5%    |

Berdasarkan hasil Confusion Matrix yang diperoleh sebelum menggunakan SMOTE pada algoritma SVM, terdapat kecenderungan bahwa model kurang akurat dalam mengenali sentiment positif terkait aplikasi X. Dengan 705 *True Positives* dan 85 *False Negatives*, model cenderung lebih baik dalam mengidentifikasi sentiment positif. Namun, terdapat 219 *False Positives* yang menunjukkan bahwa model cenderung melakukan kesalahan dengan mengklasifikasikan ulasan negatif sebagai positif. Selain itu, terdapat 218 *True Negatives* yang mengindikasikan bahwa model dapat mengenali dengan baik ulasan yang benar-benar negatif. Dengan demikian, hasil ini menunjukkan bahwa model perlu ditingkatkan kemampuannya dalam mengenali sentiment positif dengan lebih akurat.



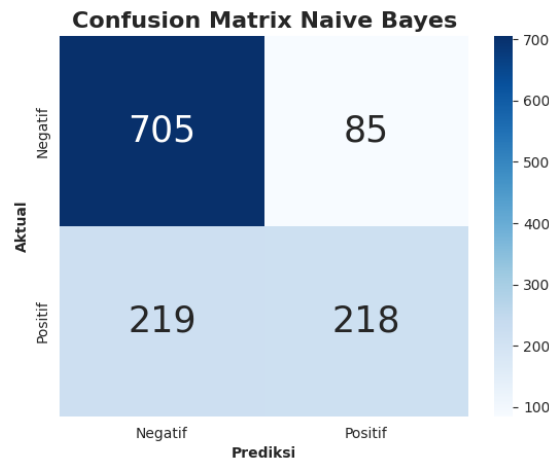
Gambar 8. Hasil *Confusion Matrix* Algoritma *Support Vector Machine* sebelum optimasi SMOTE

Berdasarkan hasil *Confusion Matrix* Algoritma *Support Vector Machine* sesudah optimasi SMOTE, dengan 415 *True Positives* dan 106 *False Negatives*, model cenderung lebih baik dalam mengidentifikasi sentimen positif terkait aplikasi X. Namun, perlu diperhatikan bahwa terdapat 87 *False Positives*, yang menunjukkan model cenderung melakukan kesalahan dengan mengklasifikasikan ulasan negatif sebagai positif. Selain itu, terdapat 432 *True Negatives*, yang mengindikasikan bahwa model dapat mengenali dengan baik ulasan yang benar-benar negatif. Dengan demikian model memiliki performa yang baik dalam mengenali sentimen positif.



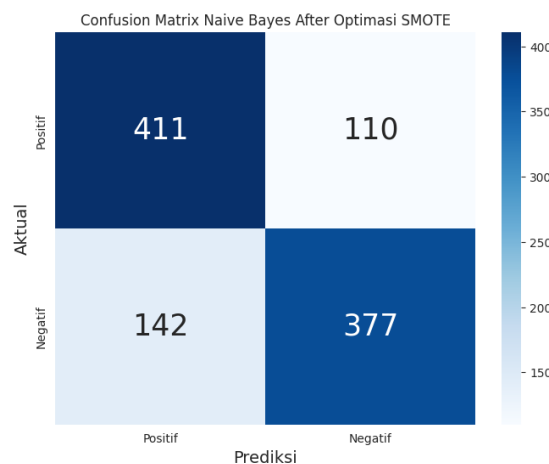
Gambar 9. Hasil *Confusion Matrix* Algoritma *Support Vector Machine* sesudah optimasi SMOTE

Berdasarkan hasil Confusion Matrix Algoritma *Support Vector Machine* sesudah optimasi SMOTE yang diperoleh, terdapat kecenderungan yang baik dalam pengenalan sentimen positif dengan 705 *True Positives*, menunjukkan model mampu mengenali ulasan yang sebenarnya positif. Namun, masih terdapat kesalahan dengan 85 *False Negatives*, yang menunjukkan beberapa ulasan positif salah diklasifikasikan sebagai negatif. Selain itu, 219 *False Positives* menandakan bahwa ada ulasan yang salah diklasifikasikan sebagai positif, padahal seharusnya negatif, dan 218 *True Negatives* menunjukkan kemampuan model dalam mengenali ulasan yang benar-benar negatif.



Gambar 8. Hasil Confusion Matrix Algoritma *Naive Bayes* sebelum optimasi SMOTE

Berdasarkan hasil Confusion Matrix Algoritma *Naive Bayes* sebelum optimasi SMOTE terdapat 411 *True Positives* (TP), menunjukkan bahwa model secara akurat mengidentifikasi ulasan yang positif. Namun, terdapat 110 *False Negatives* (FN), yang mengindikasikan bahwa ada beberapa ulasan positif yang salah diklasifikasikan sebagai negatif. Selain itu, terdapat 142 *False Positives* (FP), menunjukkan kesalahan dalam mengklasifikasikan ulasan negatif sebagai positif. Dengan 377 *True Negatives* (TN), model mampu mengenali ulasan yang benar-benar negatif.



Gambar 9. Hasil Confusion Matrix Algoritma *Naive Bayes* sesudah optimasi SMOTE

Setelah menerapkan teknik SMOTE, terjadi peningkatan yang signifikan dalam akurasi model SVM, naik dari 75,5% menjadi 81%. Peningkatan tersebut menunjukkan bahwa SMOTE berhasil meningkatkan kemampuan SVM dalam mengklasifikasikan data, terutama dalam menangani ketidakseimbangan kelas. Namun, tidak ada perubahan yang mencolok dalam akurasi model *Naive Bayes* setelah penerapan SMOTE, yang tetap berada pada 75,5%. Hal ini menunjukkan bahwa *Naive Bayes* mungkin memiliki keterbatasan dalam menangani ketidakseimbangan kelas, dan SMOTE tidak memberikan perbaikan yang signifikan dalam hal ini. Secara keseluruhan, hasil tersebut menegaskan efektivitas SMOTE dalam meningkatkan kinerja model SVM dalam mengatasi ketidakseimbangan kelas, sementara menyoroti tantangan yang mungkin dihadapi oleh model *Naive Bayes* dalam konteks yang sama.

#### IV. KESIMPULAN

Berdasarkan penelitian ini yang membahas tentang analisis sentimen terhadap aplikasi X di Google Play Store dengan menggunakan metode Naïve Bayes dan Support Vector Machine (SVM). Dataset yang digunakan terdiri dari 4087 data, yang dibagi menjadi 70% data untuk pelatihan (training) dan 30% data untuk pengujian (testing). Hasil akurasi sebelum SMOTE pada metode SVM adalah 75,5%, sedangkan pada algoritma Naive Bayes adalah 75%. Setelah penerapan SMOTE, akurasi meningkat menjadi 81% untuk SVM dan tetap 75,5% untuk algoritma Naive Bayes. Hal ini menunjukkan bahwa penerapan SMOTE memberikan peningkatan signifikan pada akurasi model klasifikasi, terutama pada metode SVM.

#### DAFTAR PUSTAKA

- [1] G. A. Mursianto, D. Widiyanto, and B. T. Wahyono, "Analisis Sentimen Ulasan Pengguna Pada Aplikasi Google Classroom Menggunakan Metode SVM Dan Seleksi Fitur PSO," *Inform. J. Ilmu Komput.*, vol. 18, no. 3, p. 221, 2022, doi: 10.52958/itk.v18i3.4685.
- [2] A. S. Rahayu, A. Fauzi, and R. Rahmat, "Komparasi Algoritma Naïve Bayes Dan Support Vector Machine (SVM) Pada Analisis Sentimen Spotify," *J. Sist. Komput. dan Inform.*, vol. 4, no. 2, p. 349, 2022, doi: 10.30865/json.v4i2.5398.
- [3] I. Aida Sapitri and M. Fikry, "Pengklasifikasian Sentimen Ulasan Aplikasi Whatsapp Pada Google Play Store Menggunakan Support Vector Machine," *J. TEKINKOM*, vol. 6, no. 1, pp. 1–7, 2023, doi: 10.37600/tekinkom.v6i1.773.
- [4] S. I. Nurhafida and F. Sembiring, "Analisis Sentimen Aplikasi Novel Online Di Google Play Store Menggunakan Algoritma Support Vector Machine (SVM)," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 6, no. 1, pp. 317–327, 2022.
- [5] M. D. Hendriyanto, A. A. Ridha, and U. Enri, "Analisis Sentimen Ulasan Aplikasi Mola Pada Google Play Store Menggunakan Algoritma Support Vector Machine," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 1, pp. 1–7, 2022, doi: 10.31539/intecom.v5i1.3708.
- [6] S. Styawati, A. R. Isnain, N. Hendrastuty, and L. Andraini, "Comparison of Support Vector Machine and Naïve Bayes on Twitter Data Sentiment Analysis," *J. Inform. J. Pengemb. IT*, vol. 6, no. 1, pp. 56–60, 2021, doi: 10.30591/jpit.v6i1.3245.
- [7] L. B. Ilmawan and E. Winarko, "Aplikasi Mobile untuk Analisis Sentimen pada Google Play," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 9, no. 1, p. 53, 2015, doi: 10.22146/ijccs.6640.
- [8] A. F. Setyaningsih, D. Septiyani, and S. R. Widiasari, "Implementasi Algoritma Naïve Bayes untuk Analisis Sentimen Masyarakat pada Twitter mengenai Kepopuleran Produk Skincare di Indonesia," *J. Teknol. Inform. dan Komput.*, vol. 9, no. 1, pp. 224–235, 2023, doi: 10.37012/jtik.v9i1.1409.
- [9] P. Fremmuzar and A. Baita, "Uji Kernel SVM dalam Analisis Sentimen Terhadap Layanan Telkomsel di Media Sosial Twitter," *Komputika J. Sist. Komput.*, vol. 12, no. 2, pp. 57–66, 2023, doi: 10.34010/komputika.v12i2.9460.
- [10] I. S. K. Idris, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, no. 1, pp. 32–35, 2023, doi: 10.37905/jjee.v5i1.16830.
- [11] L. Qadrini, H. Hikmah, and M. Megasari, "Oversampling, Undersampling, Smote SVM dan Random Forest pada Klasifikasi Penerima Bidikmisi Sejava Timur Tahun 2017," *J. Comput. Syst. Informatics*, vol. 3, no. 4, pp. 386–391, 2022, doi: 10.47065/josyc.v3i4.2154.
- [12] F. D. Astuti and F. N. Lenti, "Implementasi SMOTE untuk Mengatasi Imbalance Class pada Klasifikasi Car Evolution Menggunakan K-NN," *JUPITER J. Penelit. Ilmu Dan Teknol. Komput.*, vol. 13, no. 1, pp. 89–98, 2021.
- [13] N. S. Ramadhanti, W. A. Kusuma, and A. Annisa, "Optimasi Data Tidak Seimbang pada Interaksi Drug Target dengan Sampling dan Ensemble Support Vector Machine," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 6, p. 1221, 2020, doi: 10.25126/jtik.2020762857.
- [14] P. K. Sari and R. R. Suryono, "Komparasi Algoritma Support Vector Machine, Naïve," *J. Mnemon.*, vol. 2, no. 2, pp. 7–13, 2024.
- [15] T. T. Widowati and M. Sadikin, "Analisis Sentimen Twitter terhadap Tokoh Publik dengan Algoritma Naive Bayes dan Support Vector Machine," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 11, no. 2, pp. 626–636, 2021, doi: 10.24176/simet.v11i2.4568.
- [16] W. W. Kamal, "Analisis Sentimen Ulasan Produk Skincare Menggunakan Metode Support Vector Machine (Studi Kasus: Forum Female Daily)," p. 65, 2021.
- [17] T. Safitri, Y. Umaidah, and I. Maulana, "Analisis Sentimen Pengguna Twitter Terhadap Grup Musik BTS Menggunakan Algoritma Support Vector Machine," *J. Appl. Informatics Comput.*, vol. 7, no. 1, pp. 28–35, 2023, doi: 10.30871/jaic.v7i1.5039.
- [18] C. Kurniawan and H. Irsyad, "Perbandingan Metode K-Nearest Neighbor Dan Naïve Bayes Untuk Klasifikasi Gender Berdasarkan Mata," *J. Algoritm.*, vol. 2, no. 2, 2022.
- [19] Rina Noviana and Isram Rasal, "Penerapan Algoritma Naive Bayes Dan Svm Untuk Analisis Sentimen Boy Band Bts Pada Media Sosial Twitter," *J. Tek. dan Sci.*, vol. 2, no. 2, pp. 51–60, 2023, doi: 10.56127/jts.v2i2.791.
- [20] T. A. Dewi and E. Mailoa, "Perbandingan Implementasi Metode Smote Pada Algoritma Support Vector Machine (Svm) Dalam Analisis Sentimen Opini Masyarakat Tentang Mixue," *J. Indones. Manaj. Inform. dan Komun.*, vol. 4, no. 3, pp. 849–855, 2023, doi: 10.35870/jimik.v4i3.289.