

# **ANALISIS SENTIMEN ULASAN PENGGUNA MENGGUNAKAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE (SVM): STUDI KASUS APLIKASI M-PAJAK**

**Mas'udin Nur Hidayat**

Program Studi Sistem Informasi, Fakultas Sains dan Teknologi Universitas Terbuka  
Jalan Cabe Raya, Pondok Cabe, Pamulang, Tangerang Selatan 15437, Banten – Indonesia,  
Halo-UT 1500024

Email: [044090134@ecampus.ut.ac.id](mailto:044090134@ecampus.ut.ac.id)

## **ABSTRAK**

Transformasi digital sangat mendukung berbagai aspek kehidupan, termasuk pada sektor pemerintahan yang mulai mengembangkan *e-Government*. Salah satu penerapan *e-Government* adalah dikembangkannya aplikasi M-Pajak untuk memberikan kemudahan bagi wajib pajak dalam mengelola kewajibannya. Penelitian ini bertujuan untuk mengevaluasi sentimen pengguna terhadap aplikasi M-Pajak berdasarkan ulasan yang diambil dari Google Play Store. Analisis sentimen dilakukan dengan 1961 data ulasan yang dikelompokkan ke dalam kategori sentimen positif dan negatif. Pada penelitian ini juga dilakukan klasifikasi dengan menggunakan algoritma *Naïve Bayes* dan *Support Vector Machine*. Hasil evaluasi menunjukkan bahwa pengguna cenderung memiliki sentimen negatif terhadap aplikasi M-Pajak. Dari sisi algoritma, hasil menunjukkan bahwa model *Support Vector Machine* memiliki akurasi 96%, lebih unggul dibandingkan *Naïve Bayes* dengan akurasi 94%. Namun, analisis lebih lanjut menunjukkan adanya bias terhadap kelas mayoritas yang ditunjukkan dengan hasil *Recall* yang rendah. Sehingga diperlukan teknik penyeimbangan data untuk mengatasi masalah ketidakseimbangan kelas dan meningkatkan akurasi model.

*Kata kunci:* *Naïve Bayes*, *Sentimen Analisis*, *Support Vector Machine*

## **PENDAHULUAN**

Di era teknologi yang semakin beragam, transformasi digital mendukung berbagai aspek kehidupan dalam menciptakan inovasi, termasuk sektor pemerintahan. Salah satu bentuk penerapan transformasi digital ini adalah *e-Government*.

*e-Government* adalah inovasi yang diciptakan untuk membangun integrasi antara masyarakat, pemerintah, dan berbagai instansi melalui implementasi teknologi informasi dan komunikasi [1]. Dengan adanya *e-Government* diharapkan dapat memperkuat akuntabilitas pemerintah dengan meningkatkan akses masyarakat terhadap informasi dan layanan publik secara *real-time*.

Salah satu penerapan *e-Government* adalah dikembangkannya aplikasi M-Pajak oleh Direktorat Jenderal Pajak (DJP). Aplikasi M-Pajak dirancang untuk memberikan fasilitas kemudahan bagi masyarakat dalam mengelola kewajiban pajaknya [2]. Dengan menyediakan fitur seperti riwayat pajak, NPWP digital, dan pengingat untuk tenggat waktu pelaporan dan pembayaran [3], aplikasi ini berperan sebagai solusi praktis bagi masyarakat dalam melaksanakan kewajiban mereka.

*Google Play Store* menjadi platform yang dimanfaatkan publik untuk mengunduh aplikasi M-Pajak sekaligus memberikan ulasan terhadap penggunaan aplikasi ini [4]. Ulasan dari setiap pengguna di *Google Play Store* dapat mempengaruhi calon pengguna lain dalam mempertimbangkan penggunaan suatu aplikasi. Namun, karena jumlah data ulasan yang besar, pengolahan secara manual mungkin menjadi tidak efisien. Sehingga, diperlukan sebuah metode yang secara otomatis dapat mengidentifikasi kecenderungan ulasan pengguna terhadap aplikasi, apakah positif atau negatif [1].

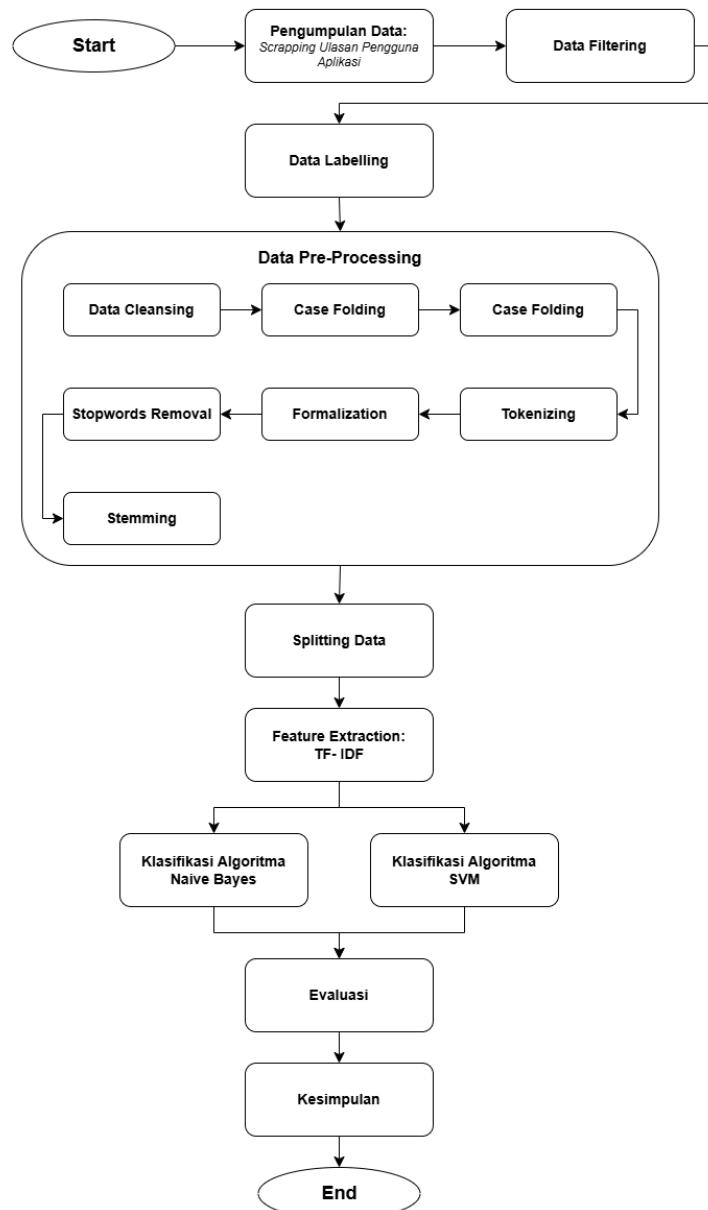
Melihat pentingnya ulasan pengguna, maka dilakukan analisis sentimen pada data ulasan aplikasi M-Pajak di *Google Play Store* untuk melihat kecenderungan pengguna terhadap aplikasi. Analisis sentimen, atau opinion mining, merupakan proses otomatisasi dalam mengolah data teks untuk menentukan apakah pendapat yang ada pada sebuah kalimat bersifat positif atau negatif [5].

Beberapa penelitian terkait analisis sentimen aplikasi menunjukkan berbagai hasil. Tanggraini dan Sitokdana (2022) menganalisis aplikasi Sentuh Tanahku menggunakan algoritma *Naïve Bayes*, dengan akurasi 89%, *precision* 83%, dan *recall* 87% [1]. Fitriyani dkk. (2023) melakukan analisis pada aplikasi Jamsostek Mobile dengan algoritma *Support Vector Machine (SVM)* yang menghasilkan akurasi 96%, *precision* 92%, *recall* 96%, dan *f1-score* 94% [6]. Indriyani dkk. (2022) melakukan analisis sentimen pada aplikasi Tiktok dengan membandingkan algoritma *Naïve Bayes* dan *SVM*, dan menemukan bahwa *SVM* memberikan performa lebih baik dengan akurasi 84%, dibandingkan dengan *Naïve Bayes* dengan akurasi 79% [7].

Dalam penelitian ini, dilakukan analisis sentimen terhadap aplikasi M-Pajak menggunakan algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) dengan tujuan mengetahui kualitas aplikasi dan kecenderungan masyarakat terhadap aplikasi, serta melakukan komparasi terhadap performa dari kedua algoritma.

## METODE PENELITIAN

Penelitian ini dilakukan dengan berbagai metode yang dibagi menjadi beberapa tahapan seperti yang ditunjukkan oleh Gambar 1.



**Gambar 1.** Metodologi Penelitian

## **Pengumpulan Dataset**

Tahap pertama yang dilakukan adalah mengumpulkan data yang diperlukan dalam analisis sentimen. Data yang digunakan adalah data ulasan pengguna aplikasi M-Pajak yang diperoleh melalui proses scraping menggunakan library *google play scraper* dengan jumlah dataset sebanyak 1961 data ulasan.

```
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
 ---  --  
 0   Unnamed: 0        2000 non-null    int64  
 1   reviewId          2000 non-null    object  
 2   userName          2000 non-null    object  
 3   userImage          2000 non-null    object  
 4   content           2000 non-null    object  
 5   score              2000 non-null    int64  
 6   thumbsUpCount     2000 non-null    int64  
 7   reviewCreatedVersion 1591 non-null  object  
 8   at                2000 non-null    object  
 9   replyContent       0 non-null      float64 
 10  repliedAt         0 non-null      float64 
 11  appVersion         1591 non-null  object  
 dtypes: float64(2), int64(3), object(7)  
 memory usage: 187.6+ KB
```

**Gambar 2.** Data Hasil Scraping

## **Data Filtering**

Proses data filtering dilakukan untuk memilih variabel dari data hasil scraping yang sesuai untuk proses analisis sentimen. Pada penelitian ini hanya digunakan kolom *content* kemudian dibuat variabel baru pada saat proses pelabelan data dengan nama variabel *label*.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1961 entries, 0 to 1960
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
 ---  --  
 0   content  1961 non-null    object  
 1   label    1961 non-null    object  
 dtypes: object(2)  
memory usage: 30.8+ KB
```

**Gambar 3.** Variabel Hasil Data Filtering

## **Pelabelan Data**

Proses pelabelan data merupakan proses pemberian label atau kategori pada data mentah untuk memberikan konteks atau informasi yang diperlukan agar data dapat digunakan dalam proses pemodelan machine learning. Pada tahap ini, dilakukan pelabelan secara manual dengan membagi label menjadi 2 yaitu positif dan negatif.

**Tabel 1.** Hasil *Data Labeling*

<i>Content</i>	<i>Label</i>
Saya udah muter-muter buka aplikasi maupun di web tetap gak bisa. Katanya akun di blokir padahal baru saja di buat. Ini gimana ceritanya saya mau urus NPWP kok gak bisa. 😞	Negative
Sangat membantu dalam pembayaran pajak tanpa perlu ke kantornya lagi	Positive

### **Proses Pre-Processing**

#### **1. Data Cleansing**

Data cleansing merupakan proses pembersihan data yang meliputi penghapusan karakter yang tidak diperlukan seperti simbol, tanda baca, emoji, dan karakter lainnya. Contoh data hasil dari penerapan *data cleansing* adalah sebagai berikut.

**Tabel 2.** Hasil *Data Cleansing*

<i>Sebelum Cleansing</i>	<i>Setelah Cleansing</i>
Saya udah muter-muter buka aplikasi maupun di web tetap gak bisa. Katanya akun di blokir padahal baru saja di buat. Ini gimana ceritanya saya mau urus NPWP kok gak bisa. 😞	Saya udah muter-muter buka aplikasi maupun di web tetap gak bisa. Katanya akun di blokir padahal baru saja di buat Ini gimana ceritanya saya mau urus NPWP kok gak bisa

#### **2. Case Folding**

Case folding merupakan proses mengubah huruf dalam sebuah teks menjadi huruf kecil guna mengurangi variasi kata yang muncul karena perbedaan huruf kapital atau kecil, sehingga dapat mempermudah proses analisis teks dan pemodelan data. Contoh data hasil dari penerapan *case folding* adalah sebagai berikut.

**Tabel 3.** Hasil *Case Folding*

<i>Sebelum Case Folding</i>	<i>Setelah Case Folding</i>
Saya udah muter-muter buka aplikasi maupun di web tetap gak bisa Katanya akun di blokir padahal baru saja di buat Ini gimana ceritanya saya mau urus NPWP kok gak bisa	saya udah muter-muter buka aplikasi maupun di web tetap gak bisa katanya akun di blokir padahal baru saja di buat ini gimana ceritanya saya mau urus npwp kok gak bisa

### 3. *Tokenizing*

Tokenizing merupakan proses pemisahan teks utuh menjadi bagian – bagian kecil yang dapat memudahkan proses analisa dan pemodelan data. Contoh data hasil dari penerapan *tokenizing* adalah sebagai berikut.

**Tabel 4.** Hasil *Tokenizing*

<i>Sebelum Tokenizing</i>	<i>Setelah Tokenizing</i>
saya udah muter-muter buka aplikasi maupun di web tetap gak bisa katanya akun di blokir padahal baru saja di buat ini gimana ceritanya saya mau urus npwp kok gak bisa	[saya, udah, muter, muter, buka, aplikasi, maupun, di, web, tetap, gak, bisa, akun, di, blokir, padahal, baru, ini, gimana, ceritanya, saya, mau, urus, npwp, kok, bisa]

### 4. *Formalization*

Formalisasi merupakan proses mengubah kalimat atau kata tidak baku menjadi bentuk baku. Contoh data hasil dari penerapan *formalization* adalah sebagai berikut.

**Tabel 5.** Hasil *Normalization*

<i>Sebelum Formalization</i>	<i>Setelah Formalization</i>
[saya, udah, muter, muter, buka, aplikasi, maupun, di, web, tetap, gak, bisa, akun, di, blokir, padahal, baru, saja, di, buat, ini, gimana, ceritanya, saya, mau, urus, npwp, kok, bisa]	[saya, sudah, berputar, berputar, buka, aplikasi, maupun, di, situs, tetap, tidak, katanya, akun, di, blokir, padahal, bisa, katanya, akun, di, blokir, padahal, baru, saja, di, buat, ini, bagaimana, ceritanya, saya, mau, mengurus, npwp, kok, tidak, bisa]

## 5. Stopwords Removal

*Stopwords Removal* merupakan proses penghapusan terhadap kata dasar yang tidak memiliki makna. Contoh data hasil dari penerapan *stopwords removal* adalah sebagai berikut.

**Tabel 6.** Hasil *Stopwords Removal*

<b>Sebelum Stopwords Removal</b>	<b>Setelah Stopwords Removal</b>
[saya, sudah, berputar, berputar, buka, aplikasi, situs, maupun, di, tetap, tidak, bisa, akun, blokir, ceritanya, katanya, akun, di, blokir, padahal, mengurus, npwp, baru, saja, di, buat, ini, bagaimana, ceritanya, saya, mau, mengurus, npwp, kok, tidak, bisa]	[berputar, berputar, buka, aplikasi, situs, tidak, bisa, akun, blokir, ceritanya, bisa, akun, blokir, cerita, urus, npwp, mengurus, npwp, tidak, bisa]

## 6. Stemming

*Stemming* merupakan proses pengubahan kata yang berimbuhan menjadi kata dasar. Proses ini dilakukan dengan menggunakan library *Sastrawi* dan *Swifter*.

**Tabel 7.** Hasil *Stemming*

<b>Sebelum Stemming</b>	<b>Setelah Stemming</b>
[berputar, berputar, buka, aplikasi, situs, tidak, bisa, akun, blokir, ceritanya, mengurus, npwp, tidak, bisa]	[putar, putar, buka, aplikasi, situs, tidak, bisa, akun, blokir, cerita, urus, npwp, tidak, bisa]

## *Splitting Data*

Proses splitting data dilakukan untuk membagi dataset menjadi beberapa bagian untuk keperluan training dan testing. Pada penelitian ini, data dibagi menjadi 75% untuk latih dan 25% untuk data uji.

## **Feature Extraction *TF-IDF***

Proses feature extraction merupakan proses identifikasi dan konversi data mentah menjadi representasi yang lebih informatif untuk memudahkan model machine learning memahami pola data. TF-IDF adalah metode statistik yang digunakan untuk mengukur pentingnya suatu data dalam dokumen tertentu. TF (*Term Frequency*) menunjukkan seberapa sering data muncul dalam dokumen, sedangkan IDF (*Inverse Document Frequency*) merepresentasikan tingkat kelangkaan data tersebut di seluruh kumpulan dokumen [8].

## ***Naïve Bayes***

*Naïve Bayes* adalah algoritma yang biasa digunakan untuk melakukan klasifikasi berbasis *teorema Bayesian* dengan bantuan perhitungan probabilitas [9]. Algoritma ini bekerja dengan mengasumsikan bahwa setiap fitur pada data bersifat independen satu sama lain.

Persamaan dari *teorema Bayesian* adalah sebagai berikut:

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad [14]$$

Di mana:

$X$  = data dengan *class* yang belum diketahui

$H$  = hipotesis data  $X$  merupakan suatu *class* spesifik

$P(H|X)$  = probabilitas hipotesis  $H$  berdasarkan kondisi  $X$  (*posteriori probability*)

$P(H)$  = probabilitas hipotesis  $H$  (*prior probability*)

$P(X|H)$  = probabilitas  $X$  berdasarkan kondisi pada hipotesis  $H$

$P(X)$  = probabilitas dari  $X$

## ***Support Vector Machine (SVM)***

*Support Vector Machine* adalah model yang menggunakan vektor pendukung sebagai titik batas untuk memisahkan kelas, bertujuan untuk menemukan bidang pemisah optimal yang membagi dua kelompok data pada tugas klasifikasi untuk data tanpa label [10]. SVM bekerja dengan cara mencari *hyperplane* yang memaksimalkan margin antara dua kelas yang berbeda.

SVM memiliki tiga jenis pendekatan atau kernel, yaitu radial, linier, dan sigmoid. Dalam penelitian ini, pendekatan kernel linier digunakan untuk memisahkan data dengan yang lebih

sederhana, di mana data yang ada dipisahkan dengan garis lurus atau bidang datar. Pendekatan dengan model linier membantu model untuk lebih cepat dalam melakukan proses pelatihan dan memberikan hasil yang memadai, terutama pada data yang dapat dipisahkan dengan garis lurus.

### **Evaluasi**

Proses evaluasi dilakukan dengan menggunakan *confusion matrix* yang dihasilkan dari tahap pemodelan klasifikasi menggunakan algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM). *Confusion matrix* adalah tabel yang menyajikan informasi tentang perbandingan antara hasil klasifikasi yang dihasilkan oleh sistem dengan hasil klasifikasi yang sebenarnya [12].

Terdapat 4 istilah dalam *confusion matrix* yaitu nilai *True Positive* (*TP*), *True Negative* (*TN*), *False Positive* (*FP*), dan *False Negative* (*FN*). Nilai *True Positive* (*TP*) dan *True Negative* (*TN*) menunjukkan hasil klasifikasi yang benar. Sedangkan nilai *False Positive* (*FP*) adalah kondisi di mana nilai hasil diprediksi *positive* namun nilai yang sebenarnya adalah *negative*, dan *False Negative* (*FN*) adalah kondisi di mana nilai hasil prediksi *negative* namun nilai sebenarnya adalah *positive*.

**Tabel 7.** Tabel *Confusion Matrix*

	<b><i>Positive</i></b>	<b><i>Negative</i></b>
<b><i>Positive</i></b>	<i>TP</i>	<i>FN</i>
<b><i>Negative</i></b>	<i>FP</i>	<i>TN</i>

Hasil dari nilai *confusion matrix* tersebut akan digunakan sebagai dasar pengukuran algoritma melalui 4 faktor, yaitu *accuracy*, *precision*, *recall*, dan *f1-score* [13] dengan persamaan sebagai berikut [12]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

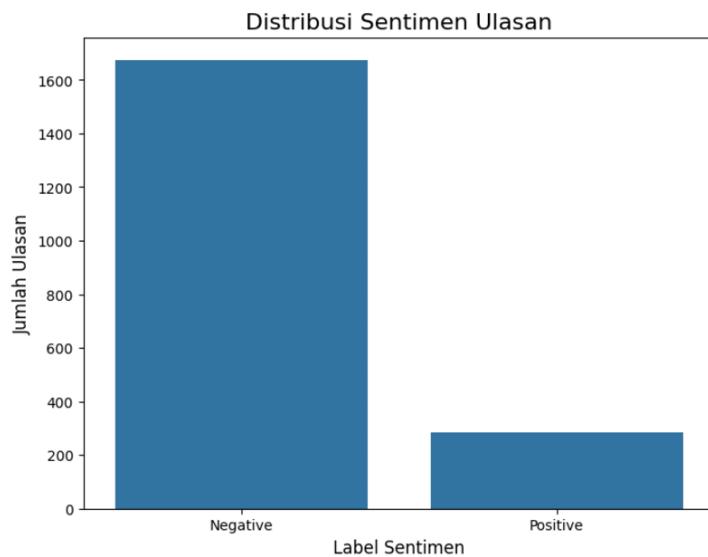
$$precision = \frac{TP}{TP + FP}$$

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

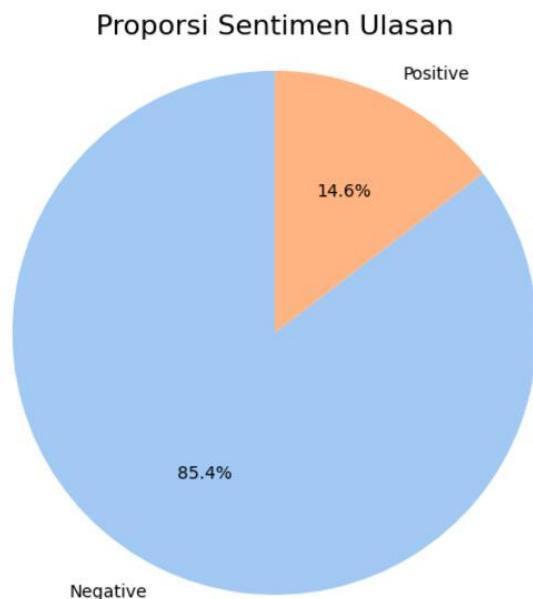
$$f1 - score = \frac{2 \times precision \times recall}{precision \times recall}$$

## HASIL DAN PEMBAHASAN

Berdasarkan hasil dari proses pelabelan secara manual, didapatkan hasil yang menunjukkan bahwa aplikasi M-Pajak cenderung mendapatkan sentimen negatif dari pengguna. Hal ini dibuktikan dengan lebih banyaknya proporsi sentimen negatif sebanyak 1675 ulasan atau 85.4% dibandingkan sentimen positif sebanyak 286 ulasan atau 14.6%. Hasil tersebut dapat dilihat pada gambar 4 dan 5.



**Gambar 4.** Perbandingan Jumlah Sentimen Negatif dan Positif.



**Gambar 5.** Perbandingan Proporsi Jumlah Sentimen Negatif dan Positif.

Setelah dilakukan proses *data preprocessing*, selanjutnya dilakukan pembagian data dengan proporsi 75% data train dan 25% data test. Kemudian dilakukan *feature extraction* menggunakan TF-IDF untuk menghitung frekuensi kata dalam setiap dokumen dan memberikan bobot berdasarkan seberapa penting kata tersebut terhadap keseluruhan dokumen. Semakin sering kata tersebut muncul, maka semakin tinggi bobotnya, seperti ditunjukkan pada gambar 6.

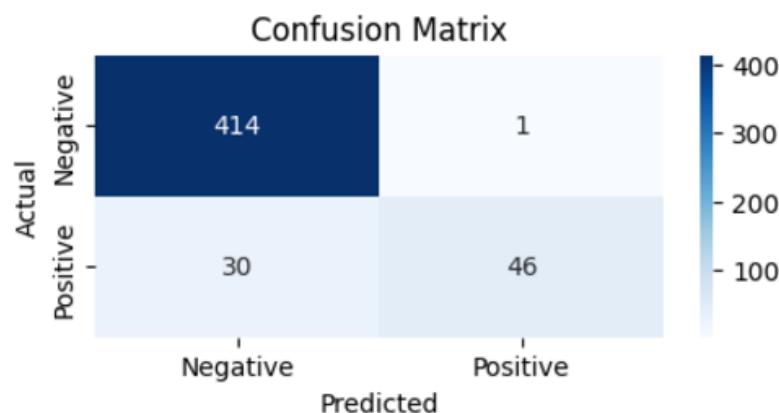


**Gambar 6.** Hasil Pembobotan TF-IDF

Klasifikasi dengan algoritma *Naïve Bayes* dan *Support Vector Machine* (SVM) dilakukan pada 1470 data latih. Proses klasifikasi dibantu dengan *library scikit-learn* dan dilakukan percobaan sebanyak 15 kali.

Hasil dari pemodelan yang didapatkan dari proses klasifikasi kemudian diterapkan pada data uji untuk dilakukan evaluasi terhadap klasifikasi dengan kedua algoritma pada proses analisis sentimen ulasan pengguna aplikasi M-Pajak. Proses evaluasi didasarkan pada *confusion matrix* yang menampilkan nilai *Accuracy*, *Precision*, dan *Recall* [11]. Hasil *confusion matrix* pada kedua algoritma ditunjukkan pada gambar 7 dan 8.

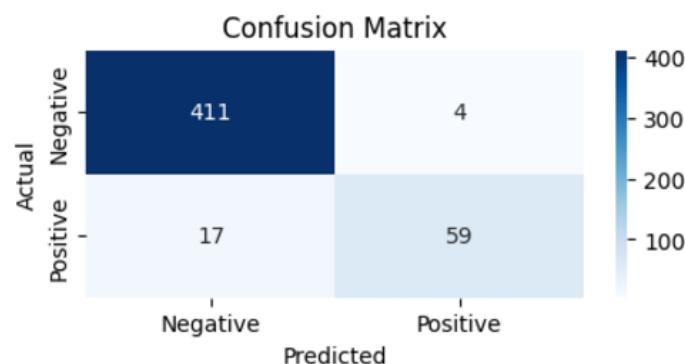
	precision	recall	f1-score	support
0	0.93	1.00	0.96	415
1	0.98	0.61	0.75	76
accuracy			0.94	491
macro avg	0.96	0.80	0.86	491
weighted avg	0.94	0.94	0.93	491



**Gambar 7.** Confusion Matrix dengan Algoritma *Naïve Bayes*

Hasil klasifikasi menggunakan algoritma *Naïve Bayes* menunjukkan akurasi yang tinggi yakni sebesar 94%. Namun, model menunjukkan kinerja yang kurang efektif dalam mendeteksi kelas positive dengan recall yang rendah (0.61). Hal ini menunjukkan bahwa model cenderung bias terhadap kelas negative yang memiliki lebih banyak data.

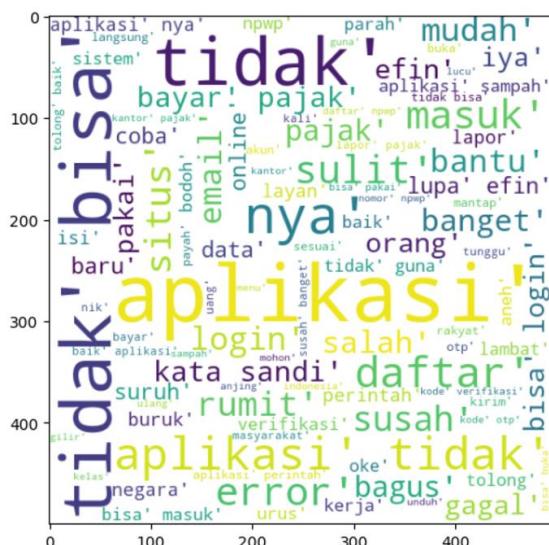
	precision	recall	f1-score	support
0	0.96	0.99	0.98	415
1	0.94	0.78	0.85	76
accuracy			0.96	491
macro avg	0.95	0.88	0.91	491
weighted avg	0.96	0.96	0.96	491



**Gambar 8.** Confusion Matrix dengan Algoritma *Support Vector Machine (SVM)*

Hasil klasifikasi menggunakan algoritma *Support Vector Machine (SVM)* menunjukkan akurasi yang lebih tinggi dari algoritma *Naïve Bayes* yakni sebesar 96%. Namun, sama seperti model sebelumnya, masih terdapat kegagalan dalam mendeteksi sebagian besar kelas positive ditunjukkan oleh recall yang rendah (0.78). Hal ini juga menunjukkan bahwa model cenderung bias terhadap kelas negative yang memiliki lebih banyak data.

Pada tahap akhir, dilakukan proses visualisasi untuk mengidentifikasi kata yang sering muncul dalam sentimen ulasan pengguna aplikasi M-Pajak. Hasil visualisasi tersebut disajikan dalam bentuk *wordcloud*.



**Gambar 9.** *Wordcloud* ulasan.

Pada gambar 9, hasil visualisasi dengan *wordcloud* menunjukkan kata yang sering muncul dalam sentimen, antara lain “aplikasi”, “tidak”, “bisa”, “daftar”, dan “error”.

## KESIMPULAN DAN SARAN

Berdasarkan hasil analisis sentimen pada 1961 data ulasan pengguna aplikasi M-Pajak menunjukkan bahwa aplikasi tersebut cenderung mendapat sentimen negatif dari pengguna. Kemudian dari hasil evaluasi dengan *confusion matrix*, menunjukkan bahwa algoritma *Support Vector Machine* memiliki kinerja yang lebih unggul dengan akurasi sebesar 96%, lebih tinggi dibandingkan dengan algoritma *Naïve Bayes* yang mencapai akurasi 94%. Dari hasil evaluasi, dinyatakan bahwa model masih cenderung bias terhadap kelompok yang memiliki porsi data lebih banyak.

Penelitian ini memiliki keterbatasan terkait dengan ketidakseimbangan data antara kelas positif dan kelas negatif. Oleh karena itu, diharapkan penelitian selanjutnya dapat dilakukan

metode untuk mengatasi masalah ketidakseimbangan data dengan menggunakan teknik seperti oversampling atau undersampling. Selain itu, dapat juga dipertimbangkan untuk menggunakan algoritma lain yang lebih mampu menangani data yang tidak seimbang, seperti Random Forest atau XGBoost.

## DAFTAR PUSTAKA

- [1] Tanggraeni, A. I., & Sitokdana, M. N. (2022). Analisis Sentimen Aplikasi E-Government pada Google Play Menggunakan Algoritma Naïve Bayes. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, 9(2), 785-795.
- [2] Directorate General of Taxes, Indonesia, DJP Online. Google Play. [Online]. Available:<https://play.google.com/store/apps/details?id=id.go.pajak.djp&hl=id>. [Accessed: 09-Nov-2024].
- [3] Faisol, A. (2024). Analisis Sentimen M-Pajak Menggunakan Algoritma Support Vector Machine, Naive Bayes dan KNN. *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, 7(5).
- [4] Jayanti, N. R. (2024). Analisis Sentimen Review Aplikasi Identitas Kependudukan Digital Menggunakan Algoritma Support Vector Machine. *GLOBAL: Jurnal Lentera BITEP*, 2(04), 132-138.
- [5] Insan, M. K. K., Hayati, U., & Nurdiawan, O. (2023). Analisis Sentimen Aplikasi Brimo Pada Ulasan Pengguna Di Google Play Menggunakan Algoritma Naive Bayes. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), 478-483.
- [6] Fitriyana, V., Hakim, L., Novitasari, D. C. R., & Asyhar, A. H. (2023). Analisis Sentimen Ulasan Aplikasi Jamsostek Mobile Menggunakan Metode Support Vector Machine. *Jurnal Buana Informatika*, 14(01), 40-49.
- [7] Indriyani, F. A., Fauzi, A., & Faisal, S. (2023). Analisis sentimen aplikasi tiktok menggunakan algoritma naïve bayes dan support vector machine. *TEKNOSAINS: Jurnal Sains, Teknologi dan Informatika*, 10(2), 176-184.
- [8] Syafrizal, S., Afdal, M., & Novita, R. (2024). Analisis Sentimen Ulasan Aplikasi PLN Mobile Menggunakan Algoritma Naïve Bayes Classifier dan K-Nearest Neighbor: Sentiment Analysis of PLN Mobile Application Review Using Naïve Bayes Classifier

and K-Nearest Neighbor Algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 10-19.

- [9] Praneswara, A. O., & Cahyono, N. (2023). Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma Naive Bayes. *The Indonesian Journal of Computer Science*, 12(6).
- [10] Hasibuan, E., & Heriyanto, E. A. (2022). Analisis Sentimen Pada Ulasan Aplikasi Amazon Shopping Di Google Play Store Menggunakan Naive Bayes Classifier. *Jurnal Teknik dan Science*, 1(3), 13-24.
- [11] Fritama, S. D., Ramadhan, Y. R., & Komara, M. A. (2023). Analisis Sentimen Review Produk Acne Spot Treatment di Female Daily Menggunakan Algoritma K-Nearest Neighbor. *KLIK: Kajian Ilmiah Informatika dan Komputer*, 4(1), 134-143.
- [12] Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Perbandingan metode naïve bayes dan support vector machine pada analisis sentimen twitter. *SMATIKA Jurnal: STIKI Informatika Jurnal*, 10(02), 71-76.
- [13] Safitri, T., Umaidah, Y., & Maulana, I. (2023). Analisis Sentimen Pengguna Twitter Terhadap BTS Menggunakan Algoritma Support Vector Machine. *Journal of Applied Informatics and Computing (JAIC)*, 7(1), 34-41.
- [14] Pradnyana, G. (2022). *Data Mining*. Universitas Terbuka