


## Penerapan Algoritma *Support Vector Machine* Dalam Analisis Sentimen Terhadap Ulasan Pengguna Pada Aplikasi *NewSakpole*

DOI: <http://dx.doi.org/10.35889/jutisi.v15i2.3478>

Creative Commons License 4.0 (CC BY – NC) 

**Renaldi Irfan Firdaus<sup>1\*</sup>, Ahmad Abdul Chamid<sup>2</sup>, Ahmad Jazuli<sup>3</sup>**  
 Teknik Informatika, Universitas Muria Kudus, Jawa Tengah, Indonesia  
 \*e-mail *Corresponding Author*: 202051075@std.umk.ac.id

### Abstract

*To determine the success rate of a service, it is necessary to conduct sentiment analysis to understand public opinion and the level of public satisfaction, both positive, neutral, and negative. Sentiment analysis was used to improve the quality of service on the NewSakpole application in android-based vehicle tax payments. In this study, several stages were used such as data crawling, data preprocessing, word weighting using TF-IDF (Term Frequency Inverse Document Frequency) and SVM (Support Vector Machine) classification model for sentiment classification. By using the Confusion Matrix test with a data sharing of 20% for data training and 80% for data testing, the accuracy results were obtained of 81.08%. The accuracy value of each sentiment also showed a fairly good performance, with a positive sentiment of 81.11%, a neutral sentiment of 78.62%, and a negative sentiment of 83.33%. These results show that the SVM algorithm is able to provide a fairly stable classification performance in this case study.*

**Keywords:** *Sentimen Analysis; Google Play Store; NewSakpole; Support Vector Machine*

### Abstrak

Untuk mengetahui tingkat keberhasilan suatu pelayanan, perlu dilakukan analisis sentimen untuk memahami opini publik dan tingkat kepuasan publik baik positif, netral, maupun negatif. Analisis sentimen digunakan untuk meningkatkan kualitas pelayanan pada aplikasi *NewSakpole* dalam pembayaran pajak kendaraan yang berbasis android. Dalam penelitian ini menggunakan beberapa tahap seperti crawling data, preprocessing data, pembobotan kata menggunakan TF-IDF (*Term Frequency Inverse Document Frequency*) serta model klasifikasi SVM (*Support Vector Machine*) untuk klasifikasi sentimen. Dengan menggunakan pengujian *Confusion Matrix* dengan pembagian data 20% untuk data training dan 80% untuk data testing memperoleh hasil akurasi sebesar 81,08%. Nilai akurasi pada masing-masing sentimen juga menunjukkan performa yang cukup baik, dengan sentimen positif sebesar 81,11%, sentimen netral 78,62%, dan sentimen negatif 83,33%. Hasil ini menunjukkan bahwa algoritma SVM mampu memberikan performa klasifikasi yang cukup stabil pada studi kasus ini.

**Kata kunci:** *Analisis Sentimen; Google Play Store; NewSakpole; Support Vector Machine*

### 1. Pendahuluan

Pajak kendaraan bermotor atau PKB adalah pajak yang dikenakan atas kepemilikan atau penguasaan kendaraan bermotor. Pajak ini dibayar sebelumnya dan berlaku selama 12 bulan atau 1 tahun [1]. Pajak kendaraan bermotor termasuk Pendapatan Asli Daerah (PAD) akan tetapi, dalam pelaksanaannya masyarakat masih banyak yang mengalami kesulitan untuk mengakses informasi, melakukan pengecekan pajak, atau melakukan pembayaran karna keterbatasan waktu, jarak, dan prosedur yang dianggap rumit. Karena hal ini tidak sedikit masyarakat yang tidak patuh dalam pembayaran pajak kendaraan [2]. Pembayaran pajak kendaraan yang dilakukan secara teratur Memberikan dampak positif bagi pemerintah daerah dan masyarakat. Hal ini bisa dilihat karena sebagian besar pendapatan negara dalam Anggaran Pendapatan dan Belanja Negara (APBN) berasal dari pajak, sekitar 73,7 persen dari total pendapatan negara [2]. Untuk menunjang ketaatan masyarakat dalam pembayaran pajak, pemerintah memfasilitasi

aplikasi New Sakpole untuk mengecek dan membayar pajak kendaraan secara online untuk memudahkan masyarakat untuk taat pembayaran pajak. Akan tetapi, aplikasi New Sakpole tidak langsung menjadi jawaban dan solusi bagi masyarakat dalam ketaatan pembayaran pajak kendaraan, karna masih ada banyak celah yang tidak diketahui. Untuk itu dibuatlah analisis sentimen terhadap aplikasi *New Sakpole* agar dapat mengetahui kekurangan dan opini publik terhadap aplikasi New Sakpole. Data teks untuk analisis sentimen sendiri didapat melalui ulasan atau komentar yang diberikan pengguna pada ulasan aplikasi *New Sakpole* pada Google Play Store. *Google Play Store* sendiri adalah layanan penyedia konten digital dari Google yang menawarkan berbagai produk daring seperti aplikasi, permainan, film, musik, dan buku dalam berbagai genre [3].

Pembayaran pajak kendaraan dapat dilakukan melalui kantor SAMSAT (Sistem Manunggal Satu Atap) yang melayani registrasi kendaraan, pembayaran pajak kendaraan bermotor, bea balik nama, dan sumbangan wajib dana kecelakaan. Kantor yang biasa berada di perkotaan terkadang membuat alasan bagi masyarakat yang tinggal jauh dari kota enggan membayar pajak kendaraan. Meskipun SAMSAT keliling sudah berjalan, masih ada masyarakat yang enggan membayar karna tidak ada waktu. Untuk itu Pemprov Jawa Tengah membuat aplikasi untuk mempermudah bagi masyarakat yang ingin membayar pajak kendaraan, tetapi jarak dari SAMSAT jauh, maupun tidak ada waktu untuk membayar pajak. Aplikasi New Sakpole merupakan aplikasi informasi terkait dengan pembayaran pajak dan pengesahan STNK secara online berbasis android. Tetapi, aplikasi perlu diperbaiki secara berkala untuk memberikan pelayanan terbaik untuk masyarakat. Untuk itu analisis sentimen untuk aplikasi New Sakpole diperlukan agar pembuat aplikasi dapat mengetahui apa fitur yang perlu diperbaiki dan perlu dikembangkan. Data kritikan dan saran diambil dari ulasan penilaian pada komentar aplikasi New Sakpole.

Beberapa studi sebelumnya telah menganalisis sentimen pada komentar di *Google Play Store*, salah satunya penelitian dari ini [4] yang menggunakan metode *knowledge discovery in database* atau KDD. Tahapannya meliputi data *selection*, *preprocessing*, *transformation*, data mining dan *evaluation*. Dan klasifikasi menggunakan algoritma *Random Forest Classifier*. Hasil penelitian menunjukkan akurasi sebesar 74% [4].

Riset-riset Salah satu penelitian lainnya yang dilakukan [5] menggunakan algoritma *Naïve Bayes Classifier* (NBC) dan mengambil hasil *scrapping* pada ulasan pada rentang waktu 1 juli 2022 – 31 juli 2022. Dataset mengklasifikasi pada rating yang diberikan yaitu ranting 4 dan 5 sebagai sentimen positif, ranting 1,2, dan 3 sebagai sentimen negatif. Menggunakan *tools Google Collabs* dengan bahasa pemrograman *Python* menghasilkan akurasi sebesar 87% [5].

Salah satu penelitian lainnya yang dilakukan [3] dengan metode yang digunakan meliputi *scrapping data* untuk mencari data, *text preprocessing* untuk mengolah data, pembobotan TF-IDF, dan modelling dengan algoritma *Naïve Bayes Classifier*. Data diambil dari ulasan pengguna kemudian diproses melalui beberapa tahap *preprocessing* seperti *casefolding*, *tokenization*, *stopword removal*, dan *stemming*. Hasil penelitian menunjukkan bahwa algoritma *Naïve Bayes Classifier* memiliki akurasi sebesar 89,67% [3].

Salah satu penelitian lainnya yang dilakukan [6] yang menggunakan *K-Nearest Neighbor* sebagai algoritma, data diambil dari ulasan di *Google Play Store* lalu diolah terlebih dahulu untuk membersihkan, yaitu dengan menghilangkan kata-kata tidak penting, memecah kalimat menjadi kata-kata, serta memperkecil bentuk kata. Setelah itu, algoritma *K-Nearest Neighbor* digunakan untuk mengklasifikasikan pendapat dalam ulasan menjadi positif atau negatif. Hasil yang didapat adalah akurasi sebesar 82% [6].

Salah satu penelitian lainnya [7] yang menggunakan metode *Support Vector Machine* dan data diambil dari ulasan di *Google Play Store*. Pengumpulan data menggunakan *crawling data* dari *library google-play-scraper*, dan metode pengujian yang digunakan adalah confusion matrix. dengan menggunakan algoritma SVM, mendapatkan hasil sebesar 82% [7].

Dari studi [4] menggunakan ulasan pengguna di *Google Play Store*. Kesenjangan yang didapat ialah hanya menggunakan 5.000 data tanpa disebutkan periode pengambilan, belum adanya perbandingan dengan algoritma lain seperti SVM, *Naïve Bayes* atau pendekatan berbasis kernel lainnya untuk performa yang lebih tinggi. Untuk studi [5] kesenjangan pelabelan menggunakan rating yang berpotensi menghasilkan bias karena tidak selalu mencerminkan isi teks ulasan secara aktual. Dan sentimen hanya menggunakan dua kelas sehingga belum mampu mempresentasikan opini netral yang muncul pada ulasan. Sedangkan studi [3] belum dilakukan komparasi dengan algoritma lain dalam klasifikasi teks berdimensi tinggi seperti *Support Vectore*

*Machine*. Penelitian lain dari [6] yang menggunakan *K-Nearest Neighbor* yang memiliki kelemahan pada data teks, selain itu juga sentimen hanya menggunakan dua kelas. Studi selanjutnya yaitu dari [7] kesenjangan data ulasan yang hanya 1.200 data ulasan mempengaruhi stabilitas dan kemampuan generalisasi klasifikasi. Dan hanya menggunakan dua kelas sentimen sebagai uji cobanya.

Dari beberapa riset, peneliti menggunakan analisis sentimen berbasis *machine learning* dengan menerapkan algoritma *Support Vector Machine (SVM)* dan metode pembobotan kata menggunakan *Term Frequency-Inverse Document Frequency*. SVM dipilih karena kemampuannya dalam melakukan klasifikasi data berdimensi tinggi secara optimal melalui pencarian *hyperplane* terbaik yang mampu memisahkan kelas sentimen secara cepat. Sementara TF-IDF digunakan untuk merepresentasikan teks ulasan dalam bentuk *vektore numerik* yang menekankan kata-kata penting dan mengurangi pengaruh kata umum, sehingga meningkatkan kualitas fitur yang digunakan dalam proses klasifikasi. Agar data ulasan dapat digunakan untuk mengetahui penilaian ulasan dan sentimen yang muncul terhadap aplikasi yang dipakai, maka analisis sentimen dapat digunakan untuk mengetahui suatu opini yang telah dituliskan dalam sebuah ulasan [8][9]. Oleh karena itu, diperlukan sistem yang mampu menganalisis sentimen pada aplikasi NewSakpole agar dapat menganalisis dan menyimpulkan ulasan pengguna serta memberikan saran untuk pengembangan aplikasi agar dapat berbenah dan mengevaluasi aplikasi NewSakpole. Setelah data didapatkan dan diolah, tahapan selanjutnya adalah pelabelan [10]. Penelitian menggunakan pelabelan untuk mengetahui sentimen pada ulasan pengguna, Proses pelabelan data multi-label sangat penting karena dengan proses yang baik diharapkan akan diperoleh model klasifikasi yang baik [11].

## 2. Metodologi

Pada metodologi penelitian ini, akan dijelaskan mengenai proses dan tahapan yang berurutan mengenai metodologi yang digunakan. Mulai tahapan awal dari pengumpulan data hingga tahapan terakhir yaitu evaluasi [12]. Proses dan tahapan dijelaskan berurutan pada Gambar 1. kerangka tahapan penelitian. Berikut alur penelitiannya :



Gambar 1. Kerangka Tahapan Penelitian

Dimulai dari tahapan pengumpulan data yang dilakukan dengan teknik *data crawling* di ulasan *Google Play Store*. Setelah itu *preprocessing data* yang meliputi *cleansing*, *case folding*, *tokenizing* dan *stopword removal*. Selanjutnya data akan diklasifikasi menggunakan SVM sehingga mendapatkan output sentimen positif, netral, atau negatif.

### 2.1 Data Crawling

Tahapan pertama pada penelitian ini adalah *crawling*, yang merupakan proses pengumpulan data mentah. Data diambil dari ulasan pengguna atau komentar pengguna pada aplikasi NewSakpole di *Google Play Store*, menggunakan metode *crawling* dengan bantuan *Library Google Play Scraper*. *Crawling* adalah metode mengumpulkan data dari halaman internet dengan menggunakan *Application Programming Interface* atau API. Dan data yang telah diambil akan disimpan pada *Google Spreadsheet* atau aplikasi Microsoft Exel dengan format *.csv* [13].

### 2.2 Preprocessing Data

Tahapan kedua berisi data yang didapat dari proses *crawling* kemudian dibersihkan di data processing. *Preprocessing* ialah proses untuk mengolah data mentah menjadi data yang lebih bersih dan sesuai untuk dianalisis. Hal ini dilakukan untuk menerima data yang sudah dilatih yang baik serta fitur-fitur yang diekstrak akan sinkron dengan yang diinginkan [14][15].

Tahapan awal sebelum pemrosesan data mencakup normalisasi, yang merupakan prosedur untuk menstandarkan teks demi memastikan keseragaman data dan meminimalkan perbedaan kata-kata yang memiliki makna serupa.

- 1) *Normalization* : *Normalization* adalah menghapus tanda baca dan simbol dengan menghilangkan karakter yang tidak relevan seperti tanda baca, angka, dan simbol khusus, dan mengubah bahasa gaul atau bahasa non-standar menjadi bentuk standar [16].
- 2) *Casefolding* : *Casefolding* adalah tahap untuk mengubah semua kata alfabet dari huruf besar menjadi huruf kecil dengan fungsi `lower()` [17].
- 3) *Tokenizing*: Setelah proses *normalization* dan *case folding*, selanjutnya merupakan tahapan *tokenizing* yaitu pemecahan kalimat menjadi perkata yang disebut token. Proses ini menggunakan library NLTK yaitu `word_tokenize()`. Tujuannya adalah untuk memfasilitasi analisis dengan mengidentifikasi batas kata atau kalimat [16][18].
- 4) *Stopword Removal*: Setelah itu tahap *stopword removal* proses untuk membuang kata-kata umum pada dokumen. Daftar kata-kata umum atau khusus yang disimpan dalam sebuah basis pengetahuan bernama stoplist. Bertujuan untuk mengurangi kebisingan dalam data dan fokus pada kata-kata yang lebih bermakna [16][19].
- 5) *Stemming*: Tahapan terakhir adalah *stemming*, di mana data yang masuk akan melalui proses ini. Data yang memiliki imbuhan akan diubah menjadi bentuk kata dasarnya. Proses ini menggunakan library sastrawi khusus yang berfungsi sebagai kamus kata dasar [20].

### 2.3 Labelling Data

Setelah melakukan scraping data dan preprocessing, langkah selanjutnya adalah menganalisis ulasan pengguna. Pendekatannya melibatkan klasifikasi setiap kalimat ulasan sebagai positif, netral, atau negatif [16]. Pada tahap ini dilakukan labelling data yaitu proses memberikan tanda atau label yang informatif pada data yang telah diproses di *preprocessing*. Data yang sudah diperoleh akan dilebeli dengan kata positif, netral, atau negatif. Proses labelling dilakukan secara otomatis dengan menggunakan library `TextBlob`. Tetapi, library tersebut belum tersedia dalam bahasa Indonesia, sehingga mempengaruhi akurasi hasil akhir karena pengambilan data ulasan dilakukan menggunakan bahasa Indonesia. Hal ini memungkinkan identifikasi topik dan sentimen dalam satu kalimat ulasan, dengan hasil disimpan dalam format CSV.

### 2.4 Feature Extraction

Tahap selanjutnya yang dilakukan setelah proses preprocessing adalah proses feature extraction. Dokumen teks diubah menjadi bentuk vektor fitur dan kumpulan data yang ada digunakan untuk membuat fitur baru. Feature extraction menggunakan metode Term Frequency-Inverse Document Frequency atau TF-IDF. TF-IDF ialah metode menghitung bobot kata dengan cara menghitung frekuensi kemunculan kata dalam sebuah dokumen dan juga menghitung seberapa sering kata tersebut muncul di seluruh koleksi dokumen teks [21]. Term Frequency adalah jumlah kali sebuah kata muncul dalam satu data, semakin sering kata tersebut muncul, maka nilai Term Frequency semakin besar. Inverse Document Frequency adalah jumlah dokumen yang menggunakan sebuah kata berdasarkan semua dokumen dalam dataset, semakin jarang kata tersebut muncul, maka nilai Inverse Document Frequency semakin besar [22]. Berikut adalah tahapan pembobotan TF-IDF :

1. Menghitung Jumlah *Term Frequency* : Jumlah kata yang sering muncul didalam sebuah data, semakin sering kata yang muncul maka semakin besar tingkat kepentingan daripada kata lain yang ditunjukkan pada rumus (1).

$$TF(t, d) = \frac{f_{t,d}}{\text{total terms}} \quad (1)$$

Keterangan:

t	= kata(term) yang sedang dihitung
d	= dokumen tempat kata itu muncul
$f_{t,d}$	= jumlah kemunculan kata t dalam dokumen d
total terms	= jumlah seluruh kata dalam dokumen d

2. *Inverse Document Frequency*: Langkah selanjutnya adalah IDF, yaitu mengukur seberapa jarang kata muncul di seluruh data dokumen. Semakin jarang, semakin besar nilai IDF-nya ditunjukkan pada rumus (2).

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \quad (2)$$

Keterangan:

N = total jumlah dokumen dalam dataset  
 df(t) = jumlah dokumen yang mengandung kata t  
 log = logaritma

3. *Term Frequency-Inverse Document Frequency* : Langkah terakhir menghitung bobot pada tiap kata yang ada didalam dokumen ditunjukkan pada rumus (3).

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \quad (3)$$

Di tahap klasifikasi, menggunakan *Support Vector Machine* yang merupakan pembelajaran dengan analisis data dan mengenali sebuah pola. SVM mengambil himpunan data masukan dan memprediksi untuk setiap masukan yang diberikan. Dengan hasil yang didapatkan berupa salah satu sentimen dari tiga sentimen, yang mana membuat algoritma SVM sebagai penggolong *non probabilistic linier biner*, dikarenakan algoritma SVM sebagai pengklasifikasi. Algoritma SVM diberikan pelatihan yang mana masing-masing ditandai sebagai salah satu dari tiga kategori sentimen [23]. Klasifikasi SVM ditulis dengan rumus (4).

$$f(x) = W^t X + b \quad (4)$$

Keterangan:

F(x) = fungsi klasifikasi  
 W = vektor bobot  
 X = vektor fitur masukan  
 B = bias

Dan untuk rumus pada tiap tiap persamaan menurut Vapnik dan Cortes (1995) dapat dilihat pada rumus dibawah ini.

$$[(W^T \cdot x_i) \geq 1 \text{ untuk } y_i = +1] \quad (5)$$

$$[(W^T \cdot x_i) = 1] \quad (6)$$

$$[(W^T \cdot x_i) \leq 1 \text{ untuk } y_i = -1] \quad (7)$$

### 3.5 Evaluasi

Tahap terakhir yaitu evaluasi, dengan adanya tahap ini, peneliti dapat mengetahui setiap akurasi, presisi, bahkan *error* dari program yang sudah dibuat. Salah satu teknik untuk mengetahui hasil evaluasi dari program analisis sentimen menggunakan *Confusion Matrix* [23]. *Confusion Matrix* adalah tabel untuk menganalisis seberapa baik keakuratan metode klasifikasi untuk menganalisis benar dan salah. Tahap evaluasi melakukan pengukuran seperti perhitungan akurasi, presisi, *recall*, dan *F1 score* terhadap machine learning [24]. Akurasi bertujuan untuk mengetahui seberapa besar model dapat mengklasifikasi dengan benar. Sedangkan presisi untuk melihat sebesar apa “kesamaan” dari hasil prediksi. *Recall* berfungsi untuk seberapa besar sistem bisa mengelompokkan data uji dari salah satu kelas. Untuk *F1 score* merupakan kombinasi antara presisi dan *recall*. Maka rumusnya dapat dilihat sebagai berikut:

$$\text{Clasifikasi Akurasi} = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (8)$$

$$\text{Recall} = \frac{T_p}{T_p + F_p} \quad (9)$$

$$\text{Presisi} = \frac{T_p}{T_p + F_p} \quad (10)$$

$$F1 \text{ score} = 2 \times \text{presisi} \times \text{recall} \quad (11)$$

## 4. Hasil dan Pembahasan

Berdasarkan penelitian dengan pengambil 7000 data pada ulasan pengguna yang akan dilakukan proses analisis sentimen, data akan dibagi terlebih dahulu menjadi 80% data data latih dan 20% data uji coba.



**Gambar 2** Tampilan Website Analisis Sentimen

Diatas adalah tampilan beranda pada website analisis sentimen NewSakpole. Tampilan beranda menginformasikan tentang informasi dasar penulis dan aplikasi yang akan diteliti.

#### 4.1 Scapping Data

Pada menu *scapping data* disajikan informasi berapa jumlah data yang mau diambil dari ulasan pengguna aplikasi NewSakpole pada kolom komentar di *Google Play Store*. Data *scapping* didapat dari ulasan pengguna yang didapat dari kolom ulasan atau kolom komentar pada aplikasi NewSakpole di *Google Play Store*. Data hasil *scapping* dapat dilihat pada Tabel 2.

**Table 1** Data Mentah Hasil Crawling

<b>UserName</b>	<b>Ulasan</b>
supermarno1980	hanya bisa utk cek jumlah tagihan pajak, sdgkan utk bayar PKB tdk bisa ,coz hanya dimonopoli bank Jateng dan ternyata bawa nomor VA dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng.. gak gunaa deeh..
Pudji Widodo	Aplikasi jembut. katanya mau nyerap apbd. mau bayar pajak atas nama orag lain aja susah nya ampun, harus selfi lah, ktp lah. management nya tulus. saya juga punya plat T entah atas nama siapa, enak bayar nya gak ribed kayak gini
Mirza Almaruf	payah!!!! mau bayar gabisa, mau bayar juga ga keluar kode bayarnya payah!! aplikasi buang <sup>2</sup> Anggara ampas sekali!! katanya mempermudah bayar pajak. ini mau bayar aja dipersulit !!!!

#### 4.2 Preprocessing

Menu *preprocessing* memperlihatkan beberapa fitur dari memasukkan data *scapping* yang sudah diambil dan di upload untuk melakukan proses *preprocessing* yang didalamnya memuat beberapa tahapan seperti *normalization*, *casefolding*, *tokenizing*, *stopword removal*, dan *stemming*. *Preprocessing* berisi pengolahan data setelah data yang didapat dari proses *scapping*. Didalamnya ada beberapa tahapan, yaitu:

##### 4.2.1 Normalization

Tahap *normalization* adalah tahap untuk menghapus kata selain alfabet seperti angka, hastag, simbol, tanda baca, maupun mention. Hasil tahap ini bisa dilihat di Table 3.

**Table 2** Data Hasil Normalization Dari Ulasan Pengguna

<b>Ulasan</b>	<b>Normalization</b>
hanya bisa utk cek jumlah tagihan pajak, sdgkan utk bayar PKB tdk bisa ,coz hanya dimonopoli bank Jateng dan ternyata bawa nomor VA dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng.. gak gunaa deeh..	hanya bisa utk cek jumlah tagihan pajak sdgkan utk bayar PKB tdk bisa coz hanya dimonopoli bank Jateng dan ternyata bawa nomor VA dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng gak gunaa deeh
Aplikasi jembut. katanya mau nyerap apbd. mau bayar pajak atas nama orag lain aja susah nya	Aplikasi jembut katanya mau nyerap apbd mau bayar pajak atas nama orag

<i>Ulasan</i>	<i>Normalization</i>
ampun, harus selfi lah, ktp lah. management nya tulul. saya juga punya plat T entah atas nama siapa, enak bayar nya gak ribed kayak gini	lain aja susah nya ampun harus selfi lah ktp lah management nya tulul saya juga punya plat T entah atas nama siapa enak bayar nya gak ribed kayak gini
payah!!!! mau bayar gabisa, mau bayar juga ga keluar kode bayarnya payah!! aplikasi buang <sup>2</sup> Anggara ampas sekali!! katanya mempermudah bayar pajak. ini mau bayar aja dipersulit !!!!	payah mau bayar gabisa mau bayar juga ga keluar kode bayarnya payah aplikasi buang Anggara ampas sekali katanya mempermudah bayar pajak ini mau bayar aja dipersulit

**4.2.2 Casefolding**

Pada tahap *casefolding* bertujuan untuk mengubah kata besar menjadi kata kecil atau lower. Data casefolding dapat dilihat pada Table 4.

**Table 3** Data Hasil Casefolding

<i>Normalization</i>	<i>Casefolding</i>
hanya bisa utk cek jumlah tagihan pajak sdgkan utk bayar PKB tdk bisa coz hanya dimonopoli bank Jateng dan ternyata bawa nomor VA dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng gak gunaa deeh	hanya bisa utk cek jumlah tagihan pajak sdgkan utk bayar pkb tdk bisa coz hanya dimonopoli bank jateng dan ternyata bawa nomor va dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng gak gunaa deeh
Aplikasi jembut katanya mau nyerap apbd mau bayar pajak atas nama orag lain aja susah nya ampun harus selfi lah ktp lah management nya tulul saya juga punya plat T entah atas nama siapa enak bayar nya gak ribed kayak gini	aplikasi jembut katanya mau nyerap apbd mau bayar pajak atas nama orag lain aja susah nya ampun harus selfi lah ktp lah management nya tulul saya juga punya plat t entah atas nama siapa enak bayar nya gak ribed kayak gini
payah mau bayar gabisa mau bayar juga ga keluar kode bayarnya payah aplikasi buang Anggara ampas sekali katanya mempermudah bayar pajak ini mau bayar aja dipersulit	payah mau bayar gabisa mau bayar juga ga keluar kode bayarnya payah aplikasi buang anggara ampas sekali katanya mempermudah bayar pajak ini mau bayar aja dipersulit

**4.2.3 Tokenizing**

Pada tahap ini setelah kata sudah bersih dan sudah diubah ke lower, kalimat ulasan akan dibagi tiap kata seperti token. Data dapat dilihat pada Tabel 5.

**Table 4** Data Hasil Tokenizing

<i>Casefolding</i>	<i>Tokenizing</i>
hanya bisa utk cek jumlah tagihan pajak sdgkan utk bayar pkb tdk bisa coz hanya dimonopoli bank jateng dan ternyata bawa nomor va dr sakpole utk bayar tunai di bank jateng jg tdk bs krn katanya blm bs sinkronisasi antara sakpole dan bank jateng gak gunaa deeh	[ 'hanya', 'bisa', 'utk', 'cek', 'jumlah', 'tagihan', 'pajak', 'sdgkan', 'utk', 'bayar', 'pkb', 'tdk', 'bisa', 'coz', 'hanya', 'dimonopoli', 'bank', 'jateng', 'dan', 'ternyata', 'bawa', 'nomor', 'va', 'dr', 'sakpole', 'utk', 'bayar', 'tunai', 'di', 'bank', 'jateng', 'jg', 'tdk', 'bs', 'krn', 'katanya', 'blm', 'bs', 'sinkronisasi', 'antara', 'sakpole', 'dan', 'bank', 'jateng', 'gak', 'gunaa', 'deeh' ]
aplikasi jembut katanya mau nyerap apbd mau bayar pajak atas nama orag lain aja susah nya ampun harus selfi lah ktp lah management nya tulul saya juga punya plat t entah atas nama siapa enak bayar nya gak ribed kayak gini	[ 'aplikasi', 'jembut', 'katanya', 'mau', 'nyerap', 'apbd', 'mau', 'bayar', 'pajak', 'atas', 'nama', 'orag', 'lain', 'aja', 'susah', 'nya', 'ampun', 'harus', 'selfi', 'lah', 'ktp', 'lah', 'management', 'nya', 'tulul', 'saya', 'juga', 'punya', 'plat', 't', 'entah', 'atas', 'nama', ]

<b>Casefolding</b>	<b>Tokenizing</b>
	'siapa', 'enak', 'bayar', 'nya', 'gak', 'ribed', 'kayak', 'gini']
payah mau bayar gabisa mau bayar juga ga keluar kode bayarnya payah aplikasi buang anggara ampas sekali katanya mempermudah bayar pajak ini mau bayar aja dipersulit	['payah', 'mau', 'bayar', 'gabisa', 'mau', 'bayar', 'juga', 'ga', 'keluar', 'kode', 'bayarnya', 'payah', 'aplikasi', 'buang', 'anggara', 'ampas', 'sekali', 'katanya', 'mempermudah', 'bayar', 'pajak', 'ini', 'mau', 'bayar', 'aja', 'dipersulit']

#### 4.2.4 Stopword Removal

Pada tahap ini, bertujuan untuk menghapus kata kata yang tidak pantas dan kata khusus yang sudah dilist didalam program. Data dapat dilihat pada Table 6.

**Table 5** Data Hasil Stopword Removal

<b>Tokenizing</b>	<b>Stopword</b>
['hanya', 'bisa', 'utk', 'cek', 'jumlah', 'tagihan', 'pajak', 'sdgkan', 'utk', 'bayar', 'pkb', 'tdk', 'bisa', 'coz', 'hanya', 'dimonopoli', 'bank', 'jateng', 'dan', 'ternyata', 'bawa', 'nomor', 'va', 'dr', 'sakpole', 'utk', 'bayar', 'tunai', 'di', 'bank', 'jateng', 'jg', 'tdk', 'bs', 'krn', 'katanya', 'blm', 'bs', 'sinkronisasi', 'antara', 'sakpole', 'dan', 'bank', 'jateng', 'gak', 'gunaa', 'deeh']	['utk', 'cek', 'tagihan', 'sdgkan', 'utk', 'bayar', 'pkb', 'coz', 'dimonopoli', 'bank', 'bawa', 'va', 'dr', 'utk', 'bayar', 'tunai', 'bank', 'bs', 'krn', 'bs', 'sinkronisasi', 'bank', 'gunaa', 'deeh']
['aplikasi', 'jembut', 'katanya', 'mau', 'nyerap', 'apbd', 'mau', 'bayar', 'pajak', 'atas', 'nama', 'orag', 'lain', 'aja', 'susah', 'nya', 'ampun', 'harus', 'selfi', 'lah', 'ktp', 'lah', 'management', 'nya', 'tulul', 'saya', 'juga', 'punya', 'plat', 't', 'entah', 'atas', 'nama', 'siapa', 'enak', 'bayar', 'nya', 'gak', 'ribed', 'kayak', 'gini']	['aplikasi', 'jembut', 'katanya', 'mau', 'nyerap', 'apbd', 'mau', 'bayar', 'pajak', 'atas', 'nama', 'orag', 'lain', 'aja', 'susah', 'nya', 'ampun', 'harus', 'selfi', 'lah', 'ktp', 'lah', 'management', 'nya', 'tulul', 'saya', 'juga', 'punya', 'plat', 't', 'entah', 'atas', 'nama', 'siapa', 'enak', 'bayar', 'nya', 'gak', 'ribed', 'kayak', 'gini']", ["nyerap", 'apbd', 'bayar', 'nama', 'susah', 'ampun', 'selfi', 'ktp', 'management', 't', 'nama', 'enak', 'bayar', 'ribed', 'gini']
['payah', 'mau', 'bayar', 'gabisa', 'mau', 'bayar', 'juga', 'ga', 'keluar', 'kode', 'bayarnya', 'payah', 'aplikasi', 'buang', 'anggara', 'ampas', 'sekali', 'katanya', 'mempermudah', 'bayar', 'pajak', 'ini', 'mau', 'bayar', 'aja', 'dipersulit']	['payah', 'bayar', 'gabisa', 'bayar', 'kode', 'bayarnya', 'payah', 'buang', 'anggara', 'ampas', 'mempermudah', 'bayar', 'bayar', 'dipersulit']

#### 4.2.5 Stemming

Tahap terakhir pada *preprocessing* adalah *stemming*, dimana kata imbuhan akan dihapus. Kata imbuhan di awal maupun diakhir, seperti 'ber', 'nya', dan 'me'. Tahap ini dilakukan agar mempermudah proses labelling dan pembobotan kata. Data stemming dapat dilihat di Table 7.

**Table 6** Data Hasil Stemming

<b>Stopword</b>	<b>Stemming</b>
['utk', 'cek', 'tagihan', 'sdgkan', 'utk', 'bayar', 'pkb', 'coz', 'dimonopoli', 'bank', 'bawa', 'va', 'dr', 'utk', 'bayar', 'tunai', 'bank', 'bs', 'krn', 'bs', 'sinkronisasi', 'bank', 'gunaa', 'deeh']	utk cek tagih sdgkan utk bayar pkb coz monopoli bank bawa va dr utk bayar tunai bank bs krn bs sinkronisasi bank gunaa deeh
['aplikasi', 'jembut', 'katanya', 'mau', 'nyerap', 'apbd', 'mau', 'bayar', 'pajak', 'atas', 'nama', 'orag', 'lain', 'aja', 'susah', 'nya', 'ampun', 'harus', 'selfi', 'lah', 'ktp', 'lah',	nyerap apbd bayar nama susah ampun selfi ktp management t nama enak bayar ribed gin

Stopword	Stemming
'management', 'nya', 'tulul', 'saya', 'juga', 'punya', 'plat', 't', 'entah', 'atas', 'nama', 'siapa', 'enak', 'bayar', 'nya', 'gak', 'ribed', 'kayak', 'gini']", "[nyerap', 'apbd', 'bayar', 'nama', 'susah', 'ampun', 'selfi', 'ktp', 'management', 't', 'nama', 'enak', 'bayar', 'ribed', 'gini']	
['payah', 'bayar', 'gabisa', 'bayar', 'kode', 'bayarnya', 'payah', 'buang', 'anggara', 'ampas', 'mempermudah', 'bayar', 'bayar', 'dipersulit']	payah bayar gabisa bayar kode bayar payah buang anggara ampas mudah bayar bayar sulit

### 4.3 Labelling

Di menu labelling tampilannya hampir sama dengan preprocessing data, tapi yang membedakan ialah prosesnya. Ketika sudah diupload data akan dilabeli dengan sentimen positif, netral, dan negatif dengan menggunakan TF-IDF sebagai labellingnya. Ditahap ini data yang sudah diolah di preprocessing akan dilabeli sesuai dengan data yang sudah diubah menjadi angkat melalui metode TF-IDF. Jika nilai diatas 0.1 maka diklasifikasikan sebagai sentimen positif, jika nilai sama dengan 0.0 maka diklasifikasikan sebagai sentimen netral, dan nilai kurang dari 0.0 diklasifikasikan menjadi sentimen negatif. Hasilnya bisa dilihat pada Tabel 8.

**Table 7** Data Hasil Labelling

Stopword	Labelling
utk cek tagih sdgkan utk bayar pkb coz monopoli bank bawa va dr utk bayar tunai bank bs krn bs sinkronisasi bank gunaa deeh	Netral
nyerap apbd bayar nama susah ampun selfi ktp management t nama enak bayar ribed gin	Positif
payah bayar gabisa bayar kode bayar payah buang anggara ampas mudah bayar bayar sulit	Negatif

### 4.4 Evaluasi

Model klasifikasi SVM menunjukkan performa yang cukup baik dengan akurasi 81.08% dari 7000 data uji. Berdasarkan hasil pada gambar 8 dapat dilihat bahwa presisi pada sentimen kelas negatif tercatat 83.33% kelas netral 78.61% dan kelas positif 81.1%. sementara nilai *recall* mendapatkan perbedaan yang cukup jauh, yakni 78.79% untuk kelas negatif, 86.46% untuk kelas netral, 78.74% untuk kelas positif. Sehingga mendapatkan hasil pada nilai *F1 score* sebesar 81% untuk kelas negatif, 82.35% untuk kelas netral, 79.91% untuk kelas positif.

	precision	recall	f1-score	support
Negatif	0.833333	0.787992	0.810029	533.000000
Netral	0.786177	0.864608	0.823529	421.000000
Positif	0.811060	0.787472	0.799092	447.000000
accuracy	0.810849	0.810849	0.810849	0.810849
macro avg	0.810190	0.813358	0.810883	1401.000000
weighted avg	0.812056	0.810849	0.810596	1401.000000

**Gambar 3** Klasifikasi *Support Vectore Machine*

Berdasarkan data klasifikasi Support Vectore Machine diatas, dapat diuji menggunakan *confusion matrix* untuk mengetahui seberapa baik metode klasifikasi yang digunakan. Dengan menggunakan *confusion matrix*, maka hasilnya dapat dilihat pada gambar 8.

	pred:Negatif	pred:Netral	pred:Positif
true:Negatif	420	53	60
true:Netral	35	364	22
true:Positif	49	46	352

**Gambar 4** Hasil Pengujian *Confusion Matrix*

Performa Algoritma SVM menggunakan akurasi, presisi, *recall*, dan *F1-Score* yang dibagi menjadi tiga kelas, yaitu positif, netral, dan negatif. Berikut hasil perhitungannya :

$$\text{Akurasi} = \frac{352 + 364 + 420}{352 + 364 + 420 + 60 + 22 + 53 + 46 + 35 + 49}$$

$$\text{Akurasi} = 0,8108$$

$$\text{Recall Positif} = \frac{352}{352 + 49 + 46}$$

$$\text{Recall Positif} = 0.788$$

$$\text{Presisi Positif} = \frac{352}{352 + 60 + 22}$$

$$\text{Presisi Positif} = 0.811$$

$$\text{F1 score Positif} = 2 \times \frac{0.811 \times 0.788}{0.811 + 0.788}$$

$$\text{F1 score Positif} = 0.799$$

Dari kelas positif, mendapatkan hasil nilai *recall* sebesar 78% untuk presisi sebesar 81% sedangkan untuk *F1-Score* sebesar 79%. Sedangkan untuk kelas netral, dapat dilihat dibawah :

$$\text{Recall Netral} = \frac{364}{364 + 35 + 22}$$

$$\text{Recall Netral} = 0.865$$

$$\text{Presisi Netral} = \frac{352}{352 + 53 + 46}$$

$$\text{Presisi Netral} = 0.786$$

$$\text{F1 score Netral} = 2 \times \frac{0.865 \times 0.786}{0.865 + 0.786}$$

$$\text{F1 score Netral} = 0.824$$

Dari kelas netral, mendapatkan hasil nilai *recall* sebesar 86% untuk presisi sebesar 78% sedangkan untuk *F1-Score* sebesar 82%. Sedangkan kelas negatif, dapat dilihat sebagai berikut:

$$\text{Recall Negatif} = \frac{420}{420 + 53 + 60}$$

$$\text{Recall Negatif} = 0.788$$

$$\text{Presisi Negatif} = \frac{420}{420 + 35 + 49}$$

$$\text{Presisi Negatif} = 0.833$$

$$\text{F1 score Negatif} = 2 \times \frac{0.788 \times 0.833}{0.788 + 0.833}$$

$$\text{F1 score Negatif} = 0.810$$

Dari kelas negatif, mendapatkan hasil nilai *recall* sebesar 78% untuk presisi sebesar 83% sedangkan untuk *F1-Score* sebesar 81%. Performa klasifikasi memiliki performa yang baik dengan tingkat akurasi sebesar 81,09%. Kinerja untuk ketiga kelas relatif seimbang dengan nilai *F1-Score* berkisaran 79% - 82%, menunjukkan model mampu mengklasifikasi tiga sentimen dengan cukup akurat.

#### 4.5 Pembahasan

Penelitian menunjukkan bahwa penerapan *Support Vector Machine* (SVM) dengan pembobotan kata TF-IDF menghasilkan performa klasifikasi yang baik dalam tiga kelas sentimen, yaitu positif, netral, dan negatif. Dengan hasil tingkat akurasi sebesar 81.09% serta nilai *recall*, presisi, dan *F1-Score* yang relatif seimbang. Hal ini memperkuat pemahaman tentang algoritma SVM efektif digunakan pada data teks dengan jumlah banyak karena kemampuannya dalam membentuk *hyperplane* untuk pemisah antar kelas sentimen. Dengan demikian, penelitian menambah bukti empiris tentang keunggulan algoritma SVM dalam klasifikasi teks.

Pada penelitian aplikasi *Mobile JKN* menggunakan algoritma klasifikasi *Support Vector Machine* mendapatkan nilai tertinggi sebesar 95%, dari pada algoritma lain seperti *Naïve Bayes* dan KNN [25]. Sedangkan penelitian ini berhasil mengklasifikasi ulasan dengan kinerja yang baik, meskipun akurasinya 81%. Hal ini memperkuat konsep bahwa algoritma SVM memiliki performa yang baik dalam klasifikasi teks yang banyak dengan metode seperti TF-IDF. Tidak sedikit juga peneliti terdahulu yang hanya menggunakan dua kelas sentimen seperti kelas positif dan kelas negatif saja. Seperti penelitian dari Raihan Maulana tentang sentimen terhadap aplikasi

*MyPertamina* [5], tetapi sudah banyak juga peneliti yang menggunakan klasifikasi tiga kelas. Dan untuk penelitian ini, peneliti berkontribusi pada perkembangan yang lebih representatif dan memperkuat argumentasi literatur agar tidak fokus pada dua kelas saja. Sehingga kelas netral juga termasuk agar lebih akurat lagi opini nyata dari pengguna. Peneliti sebelumnya menggunakan data yang cukup variatif mulai dari 3.000 data – 10.000 data untuk proses klasifikasi. Seperti penelitian pada review film yang menggunakan 5.000 data dan menggunakan algoritma SVM sebagai klasifikasi [14]. Peneliti ini menggunakan 7.000 data memberikan bukti empiris tambahan bahwa ukuran dataset mempengaruhi performa model dan menandakan perlunya dataset besar untuk meningkatkan akurasi. Beberapa peneliti hanya fokus pada akurasi tanpa membahas metrix lainnya seperti presisi, *recall*, hingga *F1-Score* [25]. Penelitian ini menggunakan confusion matrix dengan tiga kelas, sehingga memberi peluang untuk evaluasi yang lebih detail. Sehingga analisis sentimen tidak hanya diukur hanya satu metrix saja, akan tetapi dari evaluasi *multimetrix* yang menjadi standar baik dalam NLP dan machine learning berbasis teks.

## 5. Simpulan

Pengguna *NewSakpole* memberikan ulasan yang beragam terkait fitur dan pelayanan digital. Hasil klasifikasi positif menunjukkan kemudahan dalam membayar dan mengakses informasi dalam membayar pajak yang sudah bisa dilakukan secara daring. Sentimen netral memberikan dukungan untuk meningkatkan pelayanan publik khususnya dalam pajak kendaraan. Sedangkan sentimen negatif memberikan kritik untuk sistem pembayaran yang terkadang susah dan jangkauan pembayaran yang masih terbatas di satu bank.

Faktor yang mempengaruhi adalah proses pelabelan secara otomatis yang menggunakan *liberary TextBlob* yang masih belum tersedia di bahasa Indonesia sehingga mempengaruhi hasil akhir. Penelitian aplikasi *NewSakpole* dari 7000 ulasan di *Google Play Store* menggunakan model *Support Vector Machine* dengan pembagian 80% data training dan 20% data testing mendapatkan akurasi 81,8% yang diperoleh setelah melakukan pengujian pada *Confusion matrix*. Sementara itu, model klasifikasi sentimen yang lebih komprehensif dapat dikembangkan dengan menggunakan metode *deep learning* terbaru dan menggunakan IndoBERD dalam mengartikan bahasa asing ke Indonesia.

## Daftar Referensi

- [1]. N. K. A. Juliantari, I. M. Sudiartana, dan N. L. G. M. Dcriani, "Pengaruh kesadaran wajib pajak, kualitas pelayanan, kewajiban moral, sanksi pajak, dan sosialisasi pajak terhadap kepatuhan wajib pajak dalam membayar pajak kendaraan bermotor di kantor samsat gianyar," *J. Kharisma*, vol. 3, no. 1, pp. 128–139, 2021.
- [2]. J. Cong dan A. Sukrisno, "Faktor-Faktor Yang Mempengaruhi Kepatuhan Wajib Pajak Dalam Membayar Pajak Kendaraan Bermotor," *J. Paradig. Akunt.*, vol. 1, no. 2, pp. 292–299, 2019.
- [3]. B. Suseti, G. Wicaksono, A. A. Riadi, U. M. Kudus, dan G. P. Store, "Analisis Sentimen Terhadap Ulasan Aplikasi Bsi Mobile," *Bina Inform. dan Komput.*, vol. 2, no. 1, 2024.
- [4]. M. F. Y. Herjanto dan Carudin, "Sirekap Pada Play Store Menggunakan," *JITET (Jurnal Inform. dan Tek. Elektro Ter.)*, vol. 12, no. 2, pp. 1204–1210, 2024.
- [5]. R. Maulana, A. Vountama, dan T. Ridwan, "Analisis Sentimen Ulasan Aplikasi Mypertamina Pada Google Play Store Menggunakan Algoritma NBC," *J. Teknol. Terpadu*, vol. 9, no. 1, pp. 42–48, 2023.
- [6]. I. H. Kusuma dan N. Cahyono, "Analisis Sentimen Masyarakat Terhadap Penggunaan E-Commerce Menggunakan Algoritma K-Nearest Neighbor," *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, pp. 302–307, 2023.
- [7]. N. R. Setiawan dan E. R. Kaburuan, "Sentimen Analisis Review Aplikasi Digital Korlantas Pada Google Play Store Menggunakan Metode SVM," *J. SISFOKOM (Sistem Inf. dan Komputer)*, vol. 12, pp. 105–116, 2023.
- [8]. A. A. Chamid, Widowati, dan R. Kusumaningrum, "Graph-Based Semi-Supervised Deep Learning for Indonesian Aspect-Based Sentiment Analysis," *Big Data Cogn. Comput.*, 2023.
- [9]. A. Jazuli dan R. Kusumaningrum, "Enhancing Aspect-based Sentiment Analysis in Student Reviews Using Bi-directional Autoencoder and Index Generator Algorithm," *TEM J.*, 2023, doi: 10.18421/TEMxx-xx.
- [10]. A. Jazuli, A. A. Chamid, dan R. Kusumaningrum, "Transformer-based semantic indexing for aspect-based sentiment analysis using an enhanced index generation algorithm with BERT,"

- Int. J. Adv. Technol. Eng. Explor.*, vol. 12, no. 127, 2025.
- [11]. A. A. Chamid dan R. Kusumaningrum, "Labeling Consistency Test of Multi-Label Data for Aspect and Sentiment Classification Using the Cohen Kappa Method," *Ing. des Syst. d'Information*, vol. 29, no. 1, pp. 161–167, 2024.
- [12]. M. Iranda dan N. Huda, "Analisis Kinerja Algoritma SVM dan Naïve Bayes untuk Klasifikasi Sentimen Program Makan Gratis," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, pp. 1452–1464.
- [13]. M. Kesuma dan R. Iskandar, "Analisis Toko dan Asal Toko Fashion Pria di Shopee Menggunakan Data Scrapping dan Exploratory Data Analysis," *Maj. Ilm. Teknol. Elektro*, vol. 21, no. 1, 2022.
- [14]. O. I. Gifari, M. Adha, I. R. Hendrawan, F. Freddy, dan S. Durrand, "Analisis Sentimen Review Film Menggunakan TF-IDF dan Support Vector Machine," *JIFOTECH (JOURNAL Inf. Technol.)*, vol. 2, no. 1, pp. 36–40, 2022.
- [15]. A. Jazuli dan R. Kusumaningrum, "Enhancing Aspect-Based Sentiment Analysis through Data Labeling Classification on Student Reviews Using a Text Sampling Approach," *TEM J.*, vol. 8984, pp. 801–810, 2023.
- [16]. A. A. Chamid, R. Nindyasari, N. Azizah, dan A. Hariyadi, "Analysis of public opinion on the governor candidate debate using LDA and IndoBERT," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, 2025.
- [17]. A. Sujjada, J. Nurfazri, I. Griha, dan T. Isa, "Analisis Sentimen Terhadap Review Bank Digital Pada Google Play Store Menggunakan Metode Support Vector Machine ( SVM )," *J. Rekayasa Teknol. Nusa Putra*, vol. 9, no. 2, pp. 122–135, 2023.
- [18]. V. Westley, D. Thomas, dan F. Rumaisa, "Analisis Sentimen Ulasan Hotel Bahasa Indonesia Menggunakan Support Vector Machine dan TF-IDF," *J. MEDIA Inform. BUDIDARMA*, vol. 6, pp. 1767–1774, 2022, doi: 10.30865/mib.v6i3.4218.
- [19]. P. M. Prihatini, "Implementasi ekstraksi fitur pada pengolahan dokumen berbahasa indonesia," *J. MATRIX*, vol. 6, no. 3, pp. 174–178, 2016.
- [20]. Musfiroh, A. Tholib, dan Z. Arifin, "Analisis Sentimen Terhadap Ulasan Aplikasi Shopee di Google Play Store Menggunakan Metode TF-IDF dan Long Short-Term Memory ( LSTM )," *J. Electr. Eng. Comput.*, vol. 6, no. 2, pp. 371–381, 2024, doi: 10.33650/jeecom.v4i2.
- [21]. M. H. Mahendra, D. T. Murdiansyah, dan K. M. Lhaksmana, "Analisis Sentimen Tweet COVID-19 Menggunakan Metode K-Nearest Neighbors dengan Ekstraksi Fitur TF-IDF dan CountVectorizer Dike : Jurnal Ilmu Multidisiplin," *J. Ilmu Multidisiplin*, vol. 1, pp. 37–43, 2023.
- [22]. M. F. Muzakki, R. F. Umbara, F. Informatika, dan U. Telkom, *Analisis Sentimen Mahasiswa Terhadap Fasilitas Universitas Telkom Menggunakan Metode Jaringan Saraf Tiruan Dan TF-IDF*, vol. 6, no. 2. 2019.
- [23]. A. A. Munandar dkk., "Sentimen Analisis Aplikasi Belajar Online Menggunakan Klasifikasi SVM," *JOINTECS (Journal Inf. Technol Comput. Sci.)*, vol. 7, no. 1, pp. 1–7, 2026.
- [24]. A. Jazuli dan R. Kusumaningrum, "Auto Labeling to Increase Aspect-Based Sentiment Analysis Using K-Nearest Neighbors Method," *E3S Web Confrences*, vol. 05001, pp. 1–12, 2022.
- [25]. N. Bethry dan B. Tjikdaphia, "Comparison of nbc, svm, knn classification results in sentiment analysis of mobile jkn," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. IX, no. 4, 2023.