

## Analisis Sentimen Aplikasi Adiraku di *Google Play Store* Menggunakan Metode Support Vector Machine

Randi Afif<sup>1</sup>, Aji Supriyanto<sup>2</sup>, Rr. Fitri Damaryanti<sup>3</sup>, Wahyu Prasetya Adi<sup>4</sup>

<sup>1234</sup>Magister Teknologi Informasi, Fakultas Teknologi Informasi dan Industri, Universitas Stikubank Semarang

\*<sup>1</sup>randiafif0035@mhs.unisbank.ac.id, <sup>2</sup>ajisup@edu.unisbank.ac.id, <sup>3</sup>rrfitridmaryanti0038@mhs.unisbank.ac.id,

<sup>4</sup>wahyuprasetyaadi0037@mhs.unisbank.ac.id

### Abstract

The problem addressed in this study is the abundance of user reviews on the Adiraku application in the Google Play Store that have not been automatically analyzed to determine user sentiment, whether positive or negative. Therefore, this research aims to analyze the sentiment of Adiraku app user reviews using the Support Vector Machine (SVM) algorithm. The collected review data undergo preprocessing steps such as casefolding, filtering, tokenizing, and stemming to prepare the data for classification. The classification results show that the SVM model with a linear kernel can classify the reviews with an accuracy of ~91%. For positive sentiment, the model achieved a precision of 0.94 and a recall of 0.93, while for negative sentiment, it obtained a precision of 0.84 and a recall of 0.86. Data visualization using wordcloud revealed that words such as "easy", "help", and "good" dominate positive reviews, while "can't", "contract", and "error" appear more frequently in negative reviews. The evaluation using the classification report demonstrates that the SVM model performs well, although there is still room for improvement, especially in the negative sentiment category. Based on these findings, it is recommended that application development focus on improving accessibility features, contract transparency, and the overall user experience.

Keywords: sentiment, support vector Machine, google play store, wordcloud, adiraku app.

### Abstrak

Permasalahan dalam penelitian ini adalah banyaknya ulasan pengguna terhadap aplikasi Adiraku di *Google Play Store* yang belum dianalisis secara otomatis untuk mengetahui sentimen pengguna, baik positif maupun negatif. Oleh karena itu, penelitian ini bertujuan untuk menganalisis sentimen ulasan pengguna aplikasi Adiraku menggunakan algoritma *Support Vector Machine* (SVM). Data ulasan yang diperoleh kemudian diproses melalui tahapan *preprocessing* seperti *casefolding*, *filtering*, *tokenizing*, dan *stemming* untuk mempersiapkan data sebelum dilakukan klasifikasi. Hasil klasifikasi menunjukkan bahwa model SVM dengan kernel linear mampu mengklasifikasikan ulasan dengan akurasi sebesar ~91%. Pada sentimen positif, model memiliki *precision* sebesar 0.94 dan *recall* sebesar 0.93, sedangkan pada sentimen negatif, *precision* sebesar 0.84 dan *recall* sebesar 0.86. Visualisasi data menggunakan *wordcloud* menunjukkan bahwa kata-kata seperti "mudah", "bantu", dan "bagus" mendominasi ulasan positif, sementara kata-kata seperti "gak", "kontrak", dan "error" mendominasi ulasan negatif. Evaluasi menggunakan *classification report* menunjukkan bahwa model SVM memberikan performa yang baik, meskipun masih terdapat ruang untuk perbaikan khususnya pada kategori sentimen negatif. Berdasarkan hasil ini, disarankan agar pengembangan aplikasi difokuskan pada peningkatan fitur aksesibilitas, transparansi kontrak, dan pengalaman pengguna secara keseluruhan.

Kata kunci: sentimen, support vector machine, google play store, wordcloud, aplikasi adiraku

©This work is licensed under a Creative Commons Attribution -ShareAlike 4.0International License

### 1. Pendahuluan

Dalam era digital saat ini, aplikasi mobile telah menjadi bagian integral dari kehidupan sehari-hari, termasuk dalam konteks layanan keuangan. Salah satu aplikasi yang populer di Indonesia adalah Adiraku, yang memungkinkan pengguna untuk mengakses informasi pinjaman, melakukan pembayaran cicilan, serta mendapatkan layanan pelanggan secara lebih praktis. Dengan meningkatnya jumlah pengguna, aplikasi ini juga menerima beragam ulasan dari pengguna di platform *Google Play Store*. Ulasan ini tidak hanya mencerminkan pengalaman pengguna, tetapi juga memberikan wawasan berharga bagi pengembang untuk meningkatkan kualitas aplikasi. Oleh karena itu, analisis sentimen terhadap ulasan pengguna menjadi

penting untuk memahami persepsi dan kepuasan pengguna terhadap aplikasi tersebut [1-4].

Analisis sentimen merupakan teknik yang digunakan untuk mengidentifikasi dan mengklasifikasikan opini pengguna menjadi kategori positif, negatif, atau netral [1,2,5,6]. Metode ini sering diterapkan dalam bidang *Natural Language Processing* (NLP) untuk mengevaluasi pandangan masyarakat terhadap produk atau layanan tertentu. Dalam konteks aplikasi Adiraku, analisis sentimen dapat membantu pengembang memahami aspek mana yang perlu diperbaiki serta fitur apa yang paling dihargai oleh pengguna [3,7].

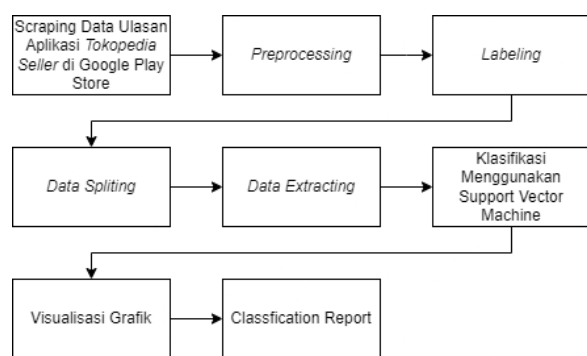
Penelitian sebelumnya menunjukkan bahwa algoritma *Support Vector Machine* (SVM) mampu memberikan performa tinggi dalam analisis sentimen, terutama

ketika digabungkan dengan teknik ekstraksi fitur seperti *TF-IDF (Term Frequency-Inverse Document Frequency)* atau *word embedding* [3,7,8,9]. *SVM* merupakan algoritma pembelajaran mesin yang efektif dalam klasifikasi, terutama pada data berlabel, dan dikenal memiliki kemampuan yang baik dalam menangani data berukuran besar serta kompleks [10-13]. Algoritma ini bekerja dengan mencari *hyperplane* optimal yang memisahkan kelas-kelas sentimen (positif, negatif, dan netral) secara maksimal dalam ruang fitur [12,13]. Dengan kekuatan generalisasi yang baik, *SVM* juga mampu mengatasi masalah ketidakseimbangan data dalam dataset ulasan pengguna, yang sering kali menjadi tantangan dalam analisis sentimen [3,5,6,7].

Penelitian ini berfokus pada penerapan algoritma *Support Vector Machine (SVM)* untuk menganalisis sentimen ulasan pengguna aplikasi Adiraku di *Google Play Store*. Dengan memanfaatkan pendekatan ini, penelitian bertujuan untuk memberikan pemahaman mendalam mengenai persepsi pengguna terhadap aplikasi tersebut, mengevaluasi performa algoritma *SVM* dalam klasifikasi sentimen, serta memberikan rekomendasi yang relevan bagi pengembangan aplikasi. Diharapkan hasil penelitian ini dapat mendukung peningkatan kualitas aplikasi Adiraku dan memberikan kontribusi positif dalam pengalaman pengguna layanan keuangan di Indonesia [3,5,6,7].

## 2. Metode Penelitian

### 2.1 Alur Penelitian



Gambar 1 Alur Penelitian

Gambar 1 menggambarkan alur penelitian ini, adapun alur yang akan dijelaskan mulai dari *Scraping* data ulasan Aplikasi Adiraku di *Google Play Store*, *Preprocessing*, *Casefolding*, *Filtering*, *Tokenizing*, *stemming*, *labelling*, *data splitting*, *data extracting*, klasifikasi menggunakan *Support Vector Machine*, visualisasi grafik, dan yang terakhir adalah *Classification Report*. Seluruh alur penelitian ini menggunakan tools Python 3.10.0.

### 2.2 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang tujuannya untuk menganalisa sekaligus mendapatkan data dan informasi yang berhubungan dengan penelitian kali ini. Data yang diperoleh merupakan

hasil *scraping* ulasan aplikasi Adiraku di *Google Play Store*. Pada penelitian kali ini data yang didapatkan ada 3 kategori yaitu ulasan positif (bintang 4 sampai 5), netral (bintang 3), dan negatif (bintang 1 sampai 2), namun yang akan digunakan hanya 2 kategori yaitu ulasan positif dan negatif saja. Tools yang digunakan dalam pengumpulan data ini yaitu *library google\_play\_scraper* yang tersedia di Python 3.10.0. Penggunaan *library google\_play\_scraper* dalam Python 3.10.0 memungkinkan pengumpulan data yang efisien, dengan target pengumpulan sebanyak 3000 data [14].

### 2.3 Preprocessing

*Preprocessing* adalah tahap yang sangat penting dalam pengolahan teks. Pada tahap ini, data ulasan yang telah dikumpulkan akan diproses agar siap digunakan dalam analisis sentimen. *Preprocessing* melibatkan beberapa langkah seperti *Casefolding*, *Filtering*, *Tokenizing*, dan *Stemming*. Sebelum dilakukan analisis lebih lanjut, data ulasan perlu melalui tahap *preprocessing* agar teks lebih bersih dan terstruktur. Tahapan *preprocessing* yang dilakukan meliputi *casefolding*, *filtering*, *tokenizing*, dan *stemming*.

*Casefolding* merupakan proses mengubah seluruh huruf dalam teks menjadi huruf kecil. Tujuan dari langkah ini adalah untuk menghindari perbedaan dalam analisis akibat kapitalisasi huruf, sehingga kata seperti "Bagus" dan "bagus" akan dianggap sama.

*Filtering* dilakukan untuk menghapus kata-kata yang tidak relevan atau tidak memiliki makna signifikan dalam proses analisis, seperti stopwords ("dan", "yang", "atau") serta karakter khusus, angka, dan tanda baca yang dapat mengganggu proses klasifikasi.

Selanjutnya, dilakukan *tokenizing*, yaitu memecah teks menjadi bagian-bagian yang lebih kecil berupa kata-kata individu (token). Misalnya, kalimat "aplikasi ini bagus" akan diubah menjadi ["aplikasi", "ini", "bagus"].

Tahap terakhir adalah *stemming*, yaitu mengubah setiap kata ke bentuk dasarnya atau akar katanya. Contohnya, kata "menggunakan" akan diubah menjadi "guna", dan "berjalan" menjadi "jalan" [15]. Proses ini penting untuk menyederhanakan variasi kata yang memiliki makna dasar yang sama.

### 2.4 Labeling

*Labeling* adalah proses memberi label pada data berdasarkan kategori yang sudah ditentukan, dalam hal ini apakah ulasan tersebut positif atau negatif. Ulasan dengan bintang 4-5 akan diberikan label positif, dan ulasan dengan bintang 1-2 akan diberikan label negatif. *Labeling* ini akan membantu dalam proses pelatihan model klasifikasi untuk mengenali pola dalam data ulasan. *Labeling* ini akan dilakukan otomatis menggunakan skrip, dengan memanfaatkan dataset yang sudah dikelompokkan berdasarkan rating bintang. *Labeling* yang tepat sangat penting untuk pelatihan

model klasifikasi, karena model harus dapat mengenali pola dalam data ulasan untuk melakukan prediksi yang akurat [15].

### 2.5 Data Splitting

Pada tahap ini, dataset yang telah melalui tahap *Preprocessing* akan dibagi menjadi dua bagian yaitu *Training Set* dan *Testing Set*. *Training Set* adalah data yang digunakan untuk melatih model klasifikasi, sedangkan *Testing Set* adalah data yang digunakan untuk menguji kemampuan model dalam melakukan klasifikasi. Proses ini dilakukan agar model yang dilatih dapat diuji dengan data yang tidak terlihat sebelumnya dan mengevaluasi akurasi serta generalisasi model. Data dibagi dengan rasio 80:20. Proses ini penting untuk memastikan bahwa model dapat diuji dengan data yang tidak terlihat sebelumnya, sehingga evaluasi akurasi dan generalisasi model dapat dilakukan dengan lebih baik [16].

### 2.6 Data Extracting

Pada tahap *data extracting*, data ulasan yang telah diproses akan menggunakan *TF-IDF* yang dihitung hanya pada data latih. Hal ini dilakukan untuk menghindari kebocoran informasi dari data uji. Setelah itu, data latih akan diubah menjadi representasi numerik dengan menggunakan *TF-IDF*. Proses yang sama akan diterapkan pada data uji, tetapi dengan menggunakan parameter *TF-IDF* yang dihitung dari data latih. Dengan cara ini, model *SVM* dan dilatih menggunakan data latih yang telah tertransformasi dan diuji menggunakan data uji dengan transformasi yang konsisten [17].

### 2.7 Support Vector Machine (SVM)

*Support Vector Machine (SVM)* adalah algoritma yang sangat efektif dalam klasifikasi, termasuk dalam analisis sentimen. Proses penerapan *SVM* dalam analisis sentimen melibatkan beberapa langkah penting yang dimulai dari pelatihan model hingga klasifikasi data baru. Dalam konteks analisis sentimen, *SVM* berfungsi untuk memisahkan data ulasan yang telah dilabeli menjadi dua kelas, yaitu positif dan negatif, dengan mencari *hyperplane* optimal yang memisahkan kedua kelas tersebut secara maksimal di ruang fitur.

Langkah pertama dalam penerapan *SVM* adalah pelatihan model menggunakan dataset yang telah dilabeli. Dataset ini biasanya terdiri dari ulasan pengguna yang telah dikategorikan sebagai positif atau negatif. *SVM* dapat digunakan untuk menganalisis sentimen pengguna aplikasi Google Meet dengan hasil yang memuaskan [3]. Selain itu, penelitian oleh Oktafani dan Prasetyaningrum juga menggarisbawahi pentingnya *SVM* dalam mengotomatisasi analisis sentimen pada komentar aplikasi, yang menunjukkan efisiensi dalam menangani jumlah data yang besar [12].

Setelah model dilatih, langkah selanjutnya adalah pencarian *hyperplane*. *Hyperplane* ini berfungsi sebagai batas pemisah antara kelas-kelas yang ada.

*SVM* berusaha untuk menemukan *hyperplane* yang memaksimalkan margin antara dua kelas, yang berarti jarak antara *hyperplane* dan titik data terdekat dari masing-masing kelas harus sebesar mungkin. *SVM* dapat mengevaluasi data pelatihan dan pengujian dengan baik, menggunakan teknik *preprocessing* seperti *case folding* dan *stopword removal* untuk meningkatkan akurasi klasifikasi [18].

Setelah proses pelatihan dan pencarian *hyperplane* selesai, model *SVM* siap untuk melakukan klasifikasi terhadap data baru yang belum diketahui sentimennya. *SVM* dapat digunakan untuk menganalisis lebih dari 1.000 ulasan pengguna aplikasi Grab dan menghasilkan akurasi sebesar 85,54% [19]. Hasil ini menunjukkan bahwa *SVM* tidak hanya efektif dalam pelatihan tetapi juga dalam aplikasi praktis untuk klasifikasi sentimen.

Dalam implementasinya, *SVM* dapat digunakan dengan berbagai kernel, seperti kernel linear dan RBF, yang dapat disesuaikan dengan karakteristik data yang digunakan. Pada penelitian ini, digunakan kernel linear karena dianggap sesuai dengan karakteristik data teks ulasan yang bersifat tinggi dimensi dan sparsity tinggi. Penelitian oleh Sholihah menunjukkan perbandingan kinerja antara kernel linear dan RBF dalam analisis sentimen, yang memberikan wawasan lebih lanjut tentang bagaimana pemilihan kernel dapat mempengaruhi hasil klasifikasi [13]. Dengan menggunakan *library Python* seperti *scikit-learn*, proses implementasi *SVM* menjadi lebih mudah dan efisien, memungkinkan peneliti untuk fokus pada pengembangan model dan analisis hasil.

### 2.8 Visualisasi Hasil

Visualisasi hasil analisis dan evaluasi model dilakukan untuk memberikan gambaran yang lebih jelas mengenai performa model. Beberapa jenis visualisasi yang dapat digunakan antara lain *Word Cloud* dan *Confusion matrix*. *Word Cloud* digunakan untuk menampilkan kata-kata yang sering muncul dalam ulasan, dengan ukuran kata yang lebih besar menunjukkan frekuensi kemunculan yang lebih tinggi, sedangkan *Confusion matrix* digunakan untuk menampilkan jumlah prediksi yang benar dan salah dalam klasifikasi model, selain itu *Confusion matrix* mengungkapkan pola kesalahan klasifikasi yang signifikan, membandingkan tingkat akurasi dan mengukur performa model dalam klasifikasi [20,21]. Visualisasi ini akan dilakukan menggunakan *matplotlib* dan *seaborn*.

### 2.9 Classification Report

*Classification Report* adalah laporan yang berisi metrik evaluasi dari performa model klasifikasi, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. *Accuracy* yaitu ukuran persentase prediksi yang benar dibandingkan dengan seluruh data uji. Akurasi memberikan gambaran umum tentang seberapa baik model dapat memprediksi dengan benar, meskipun

tidak mempertimbangkan distribusi kelas yang tidak seimbang. Berikut rumusnya:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad [1]$$

*Precision* yaitu ukuran ketepatan model dalam mengklasifikasikan data positif atau negatif. Berikut rumusnya:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad [2]$$

*Recall* yaitu ukuran kemampuan model dalam menangkap seluruh data yang seharusnya diklasifikasikan sebagai positif atau negatif. Berikut rumusnya:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad [3]$$

*F1-Score* yaitu rata-rata harmonis antara *Precision* dan *Recall*, yang memberikan gambaran tentang keseimbangan model dalam melakukan klasifikasi. Berikut rumusnya:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad [4]$$

*Classification Report* ini dihasilkan menggunakan fungsi `classification_report` dari `scikit-learn` untuk memberikan gambaran yang lebih jelas mengenai kualitas model [22].

### 3. Hasil dan Pembahasan

#### 3.1 Pengumpulan Data

Data ulasan aplikasi Adiraku dikumpulkan dari *Google Play Store* menggunakan *library google-play-scraper* pada Python. Ulasan yang dikumpulkan dikategorikan berdasarkan rating menjadi tiga kategori: positif (bintang 4–5), netral (bintang 3), dan negatif (bintang 1–2). Namun, penelitian ini hanya menggunakan ulasan positif dan negatif.

Sebanyak 3000 data ulasan berhasil diperoleh untuk dianalisis lebih lanjut, penggunaan dataset yang lebih besar dapat meningkatkan akurasi dan keandalan klasifikasi [23]. Kolom yang diambil yaitu Full Text. Hasil pengumpulan data dapat dilihat pada Tabel 1.

Tabel 1 Hasil Pengumpulan Data

Full Text
Ini adira gimana sih,saya mau lihat kontrak saya aja ko susah banget,dibikin gampang lah ,kaya tetangga , login,masukan no kotrak udah transparan tuh
Oke
Buruk

tahun 2024 sampai 2025 bukannya di perbaiki ni aplikasi malah tetap eror belum ada kontrak aktif ,udah kasih tau petugasnya malah bilang eror aja 🤔

kontrak tidak muncul di aplikasi

#### 3.2 Preprocessing

Tahap *Preprocessing* dilakukan untuk membersihkan dan menyiapkan data teks agar lebih terstruktur dan sesuai untuk analisis.

##### 3.2.1 Casefolding

Pada tahap ini, semua huruf pada teks ulasan diubah menjadi huruf kecil. Tujuannya adalah untuk menghilangkan perbedaan akibat kapitalisasi, sehingga kata seperti "Bagus" dan "bagus" dianggap sama. Langkah ini penting dalam analisis teks untuk memastikan konsistensi dalam pengolahan data [24].

##### 3.2.2 Filtering

*Filtering* dilakukan untuk menghapus kata-kata yang tidak relevan (stopwords) seperti "dan", "atau", "yang", serta tanda baca, angka, dan karakter khusus lainnya. Tujuannya adalah untuk mengurangi kebisingan data dan hanya menyisakan kata-kata yang bermakna penting.

##### 3.2.3 Tokenizing

Teks ulasan yang telah difilter kemudian dipecah menjadi kata-kata individu (token). Misalnya, ulasan "Aplikasi ini sangat bagus" diubah menjadi ["aplikasi", "ini", "sangat", "bagus"]. Proses-proses ini penting dalam analisis teks karena dapat meningkatkan akurasi model klasifikasi, seperti yang dijelaskan dalam literatur yang menunjukkan bahwa *Preprocessing* yang baik dapat meningkatkan performa *Support Vector Machine* [25].

##### 3.2.4 Stemming

*Stemming* dilakukan untuk mengubah kata-kata menjadi bentuk dasarnya. Misalnya, kata "berjalan" diubah menjadi "jalan". Proses ini bertujuan untuk menyamakan variasi kata dengan arti yang sama. Hasil *Preprocessing* dapat dilihat pada Tabel 2.

Tabel 2 Hasil Preprocessing

Full Text	Text Clean	Text Filtering	Text Tokenizin g	Text Stemmin g
Ini adira gimana sih,saya mau lihat kontrak saya aja ko susah banget,di bikin	ini adira gimana sih,saya mau lihat kontrak saya aja ko susah bangetdi bikin	adira gimana sihsaya lihat kontrak aja ko susah bangetdi bikin	['adira', 'gimana', 'sihsaya', 'lihat', 'kontrak', 'aja', 'ko', 'susah', 'bangetdibikin',	adira gimana sihsaya lihat kontrak aja ko susah bangetdi bikin

gampang lah ,kaya tetangga , login,maskan no kotrak udah transpara n tuh	gampang lah kaya tetangga loginmas ukan no kotrak udah transpara n tuh	gampang kaya tetangga loginmas ukan no kotrak udah transpara n tuh	'gampang', 'kaya', 'tetangga', 'loginmas ukan', 'no', 'kotrak', 'udah', 'transpara n', 'tuh']	gampang kaya tetangga loginmas ukan no kotrak udah transpara n tuh
Oke	oke	oke	['oke']	oke
Buruk	buruk	buruk	['buruk']	buruk
tahun 2024 sampai 2025 bukannya di perbaiki ni aplikasi malah tetap eror belum ada kontrak aktif ,udah kasih tau petugasn ya bilang eror aja	tahun 2024 sampai 2025 bukannya di perbaiki ni aplikasi malah tetap eror belum ada kontrak aktif ,udah kasih tau petugasn ya bilang eror aja	2024 2025 perbaiki ni aplikasi eror kontrak aktif udah kasih tau petugasn ya bilang eror aja	['2024', '2025', 'perbaiki', 'ni', 'aplikasi', 'eror', 'kontrak', 'aktif', 'udah', 'kasih', 'tau', 'petugasn ya', 'bilang', 'eror', 'aja']	2024 2025 baik ni aplikasi eror kontrak aktif udah kasih tau tugas bilang eror aja
kontrak tidak muncul di aplikasi	kontrak tidak muncul di aplikasi	kontrak muncul aplikasi	['kontrak', 'muncul', 'aplikasi']	kontrak muncul aplikasi

### 3.3 Labeling

Pada tahap ini, ulasan diberi label berdasarkan kategorinya. Ulasan dengan rating bintang 4–5 diberi label positif, sedangkan ulasan dengan rating bintang 1–2 diberi label negatif. Proses *Labeling* dilakukan secara otomatis menggunakan skrip Python untuk memastikan konsistensi. Hasil *Labeling* dapat dilihat pada Tabel 3.

Tabel 3 Hasil *Labeling*

Text Stemming	Labeling
adira gimana sihsaya lihat kontrak aja ko susah bangetdibikin gampang kaya tetangga loginmasukan no kotrak udah transpara n tuh	negatif
oke	positif

buruk negatif  
2024 2025 baik ni aplikasi eror kontrak aktif udah kasih tau tugas bilang eror aja negatif  
kontrak muncul aplikasi negatif

### 3.4 Klasifikasi menggunakan *Support Vector Machine*

Dalam penelitian ini, proses pemodelan menggunakan algoritma *Support Vector Machine (SVM)*. Pemodelan dilakukan dengan memproses data *train\_set* dan *test\_set* yang masing-masing berjumlah 2272 dan 569 data, serta label *y\_train* dan *y\_test* yang juga berjumlah 2272 dan 569 data. Algoritma *SVM* dengan *kernel linear* digunakan, yang diimpor dari modul *sklearn.svm*. Adapun sintaks yang digunakan dapat dilihat pada Gambar 2.

```

from sklearn.svm import LinearSVC
from sklearn.metrics import accuracy_score

for c in [0.01, 0.05, 0.25, 0.5, 0.75, 1]:
    svm = LinearSVC(C=c)
    svm.fit(train_set_tf_id, y_train)
    print("Accuracy for C=%s: %s" % (c, accuracy_score(y_train, svm.predict(train_set_tf_id))))

svm = LinearSVC(C=1)
svm.fit(train_set_tf_id, y_train)

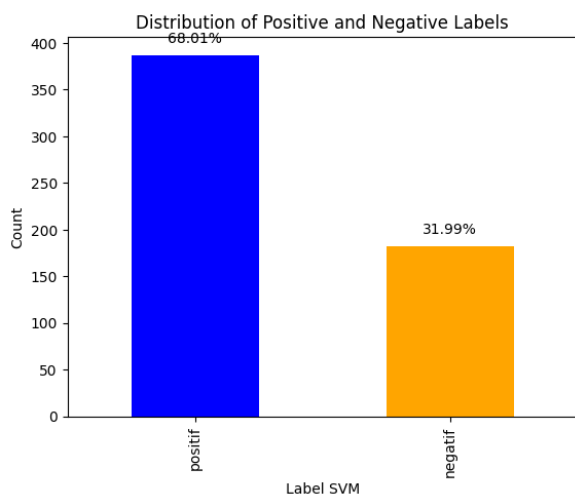
Accuracy for C=0.01: 0.7733274647887324
Accuracy for C=0.05: 0.818373963669719
Accuracy for C=0.25: 0.8661091549295775
Accuracy for C=0.5: 0.975792233211268
Accuracy for C=0.75: 0.8980833823189034
Accuracy for C=1: 0.981954235321126

LinearSVC(C=1)

y_pred = svm.predict(test_set_tf_id)
data_clean_baru = pd.DataFrame({'test_set': test_set, 'label SVM': y_pred})
data_clean_baru.to_csv('Dataset/ulasanapp_3000_hasil_sentimen_SVM_test.csv', index=False)
data_clean_baru.head()
    
```

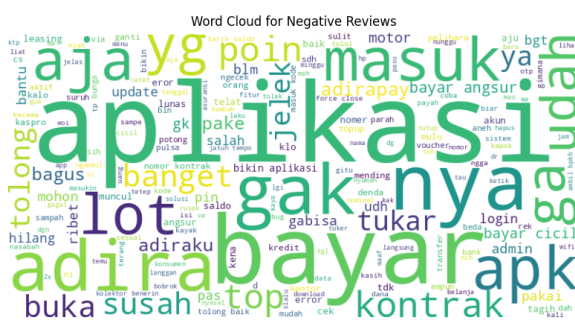
Gambar 2 Klasifikasi *Support Vector Machine*

Langkah selanjutnya dalam proses analisis adalah melakukan visualisasi hasil klasifikasi untuk mendapatkan gambaran distribusi data secara visual. Visualisasi ini dilakukan dalam bentuk diagram batang (bar plot), yang menunjukkan perbandingan jumlah data yang termasuk dalam kategori sentimen positif, negatif, dan netral yang dapat dilihat pada gambar 3. Dengan adanya visualisasi ini, pembaca dapat lebih mudah memahami kecenderungan opini pengguna terhadap aplikasi yang diteliti.



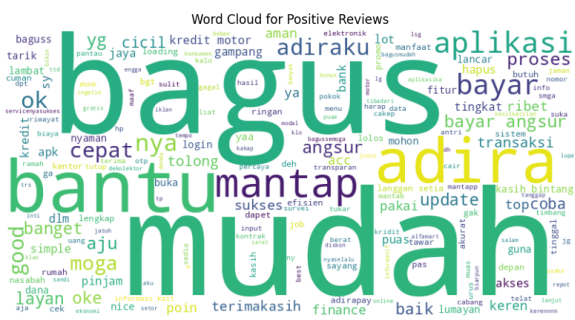
Gambar 3 Visualisasi data hasil klasifikasi

Selanjutnya adalah akan ditampilkan *Wordcloud* untuk mengetahui kata apa saja yang muncul saat pemodelan menggunakan *Support Vector Machine*. *Wordcloud* bisa dilihat pada gambar 4 dan gambar 5.



Gambar 4 *Wordcloud* Negatif

Pada Gambar 4, *Wordcloud* negatif ditampilkan kata-kata yang sering muncul dengan label sentiment negative.



Gambar 5 *Wordcloud* Positif

Demikian juga pada Gambar 5, ditampilkan kata-kata yang sering keluar dengan label sentiment positif.

*Wordcloud* memiliki kegunaan untuk menampilkan kata dominan pada sebuah data, data yang akan diproses ada 2 yaitu data positif dan data negatif. Pada *Wordcloud* kata-kata yang sering muncul pada data positif adalah bagus, bantu, adira, mantap, mudah,

proses, gampang, aman, dst. Adapun pada data negatif kata-kata yang dominan muncul adalah aplikasi, gak, bayar, poin, jelek, bikin aplikasi, gabisa, kontrak, dst..

Langkah selanjutnya adalah *confusion matrix*. Total data yang digunakan dalam *confusion matrix* adalah 569 data. Hasil *confusion matrix* klasifikasi *SVM* dapat dilihat pada Tabel 4.

Tabel 4 *Confusion matrix* Klasifikasi *SVM*

Actual/Predicted	Predicted Negative	Predicted Positive
Actual Negative	153 (TN)	24 (FP)
Actual Positive	29 (FN)	363 (TP)

Berdasarkan *confusion matrix* di atas, metrik evaluasi dapat dihitung. Pertama yaitu *Accuracy*, berikut perhitungannya:

$$Accuracy = \frac{363 + 153}{363 + 153 + 24 + 29} = 90.68\% \approx 91\%$$

Kemudian yang kedua yaitu *Precision*, berikut perhitungannya:

$$Precision = \frac{363}{363 + 24} = 93.80\%$$

Kemudian yang ketiga yaitu *Recall*, berikut perhitungannya:

$$Recall = \frac{363}{363 + 29} = 92.60\%$$

Kemudian yang keempat yaitu *F1-Score*, berikut perhitungannya:

$$F1 - Score = 2 \times \frac{0.9380 \times 0.9260}{0.9380 + 0.9260} = 93.19\%$$

Setelah diketahui nilai *confusion matrix* dan metrik evaluasi, berikutnya adalah evaluasi model menggunakan *classification report*. *Classification report* dari hasil pengujian menggunakan metode *SVM* dapat dilihat pada tabel 5.

Tabel 5 *Classification Report* Klasifikasi *SVM*

	Precision	Recall	F1-Score	Support
Negatif	0.84	0.86	0.85	177
Positif	0.94	0.93	0.93	392

<i>Accuracy</i>			0.91	569
<i>Macro avg</i>	0.89	0.90	0.89	569
<i>Weighted avg</i>	0.91	0.91	0.91	569

Tabel 5 menunjukkan hasil evaluasi model *Support Vector Machine* dalam mengklasifikasikan sentimen ulasan menjadi positif dan negatif. Secara keseluruhan, model ini memiliki akurasi sebesar 91%, yang mencerminkan bahwa model mampu mengklasifikasikan sebagian besar ulasan dengan benar.

Pada ulasan dengan sentimen negatif, *Precision* sebesar 0.84 menunjukkan bahwa sebagian besar prediksi negatif sesuai dengan kategori sebenarnya, meskipun tidak sempurna. *Recall* sebesar 0.86 mengindikasikan bahwa model mampu menangkap sebagian besar ulasan yang benar-benar tergolong negatif. Namun, masih ada sejumlah kecil ulasan negatif yang tidak terklasifikasi dengan benar. *F1-Score* untuk kategori ini mencapai 0.85, yang menunjukkan keseimbangan yang baik antara *Precision* dan *Recall*.

Sebaliknya, pada ulasan dengan sentimen positif, model menunjukkan performa yang sangat baik dengan *Precision* sebesar 0.94 dan *Recall* sebesar 0.93. Hal ini menunjukkan bahwa model sangat andal dalam memprediksi ulasan positif dan mampu menangkap hampir seluruh ulasan yang tergolong positif. *F1-Score* untuk kategori ini adalah 0.93, menegaskan kualitas prediksi yang konsisten dan unggul pada sentimen positif.

Secara keseluruhan, akurasi model mencapai 0.91, menunjukkan bahwa 91% prediksi model sesuai dengan kategori sebenarnya. Rata-rata nilai *Precision*, *Recall*, dan *F1-Score* masing-masing berada di kisaran 0.89 hingga 0.91, menegaskan bahwa model memiliki performa yang baik dalam mengklasifikasikan sentimen ulasan secara keseluruhan.[22].

#### 4. Kesimpulan

Penelitian ini dilakukan untuk menjawab permasalahan terkait belum adanya sistem otomatis yang dapat mengklasifikasikan sentimen ulasan pengguna terhadap aplikasi Adiraku di *Google Play Store*. Banyaknya ulasan pengguna yang tidak terorganisir menyebabkan kesulitan bagi pengembang aplikasi dalam mengetahui persepsi pengguna secara cepat dan akurat. Oleh karena itu, penelitian ini bertujuan untuk membangun sistem analisis sentimen menggunakan

algoritma *Support Vector Machine (SVM)* sebagai solusi untuk mengatasi permasalahan tersebut.

Urgensi dari penelitian ini terletak pada kebutuhan akan sistem analitik yang mampu menyajikan informasi sentimen pengguna secara otomatis dan efisien, sehingga pengembang aplikasi dapat mengambil keputusan berbasis data dalam meningkatkan kualitas layanan dan pengalaman pengguna.

Hasil pengujian menunjukkan bahwa sistem yang dibangun mampu mengklasifikasikan sentimen ulasan dengan tingkat akurasi sebesar ~91%, *precision* untuk sentimen positif sebesar 0.94 dan *recall* 0.93, serta *precision* untuk sentimen negatif sebesar 0.84 dan *recall* 0.86. Visualisasi data menggunakan *Wordcloud* juga membantu dalam mengidentifikasi kata-kata dominan pada masing-masing kategori sentimen. Berdasarkan hasil pengujian ini, dapat disimpulkan bahwa sistem berhasil mengatasi permasalahan utama, yaitu menyediakan klasifikasi sentimen otomatis yang akurat dan informatif terhadap ulasan pengguna aplikasi Adiraku.

Berdasarkan analisis *Wordcloud* untuk ulasan negatif dan positif pada aplikasi Adiraku, berikut adalah rekomendasi *Decision Support System (DSS)* untuk pengembangan aplikasi:

- a. Perbaikan Isu Utama Berdasarkan *Wordcloud* Negatif
  1. Masalah Aksesibilitas dan Login: Kata-kata seperti masuk, gak, susah, buka, apk, dan error menunjukkan keluhan pengguna terkait kesulitan dalam mengakses aplikasi. Rekomendasi: Tingkatkan stabilitas sistem untuk memastikan pengguna dapat login tanpa gangguan.
  2. Tambahkan panduan atau *help center* di halaman login untuk membantu pengguna yang mengalami masalah.
  3. Sediakan opsi pemulihan akun yang mudah dan cepat, seperti menggunakan kode OTP atau pemulihan melalui email.
  4. Fitur Kontrak dan Transaksi: Kata-kata seperti kontrak, bayar, saldo, dan angsur menunjukkan keluhan terkait transparansi informasi kontrak dan proses pembayaran. Rekomendasi: Pastikan informasi kontrak tampil secara lengkap dan akurat di aplikasi.
  5. Optimalkan fitur pembayaran untuk meminimalkan *error*, misalnya dengan menyediakan beberapa opsi pembayaran yang lebih aman dan mudah.
  6. Tampilkan notifikasi atau status pembayaran secara real-time.

7. Pengalaman Pengguna (UX): Kata-kata seperti susah, gak, dan ribet mengindikasikan bahwa antarmuka aplikasi perlu lebih ramah pengguna. Rekomendasi: Redesign antarmuka dengan fokus pada pengalaman pengguna yang sederhana, cepat, dan intuitif.
  8. Tambahkan fitur search untuk mempermudah pengguna menemukan menu yang mereka butuhkan.
- b. Memperkuat Sentimen Positif Berdasarkan Wordcloud Positif
1. Peningkatan Kemudahan dan Bantuan: Kata-kata seperti bantu, mudah, cepat, mantap, dan bagus menunjukkan apresiasi pengguna terhadap kemudahan aplikasi. Rekomendasi: Pertahankan dan promosikan fitur yang membantu pengguna seperti pengajuan kredit atau pembayaran cepat.
  2. Tambahkan fitur chatbot atau layanan pelanggan langsung yang responsif untuk memberikan solusi instan.
  3. Peningkatan Inovasi dan Fitur Tambahan: Kata update menunjukkan pengguna menyadari adanya pembaruan aplikasi. Rekomendasi: Lakukan pembaruan aplikasi secara berkala dengan menambahkan fitur yang relevan, seperti pengingat tagihan atau promo khusus.
  4. Pastikan semua pembaruan memiliki dokumentasi lengkap agar pengguna dapat memahami perubahan dan manfaatnya.
  5. Promosi dan Reward: Kata-kata seperti bantu dan bagus dapat dimanfaatkan untuk mendorong lebih banyak ulasan positif. Rekomendasi:
  6. Adakan program reward untuk pengguna yang memberikan umpan balik atau ulasan positif, seperti poin yang dapat digunakan untuk pembayaran.
  7. Berikan apresiasi kepada pengguna setia melalui diskon atau fitur eksklusif.
- Daftar Rujukan**
- [1] A. Kovari, "AI for Decision Support: Balancing Accuracy, Transparency, and Trust Across Sectors," *Inf.*, vol. 15, no. 11, 2024, doi: 10.3390/info15110725.
  - [2] C. Collins, D. Dennehy, K. Conboy, and P. Mikalef, "Artificial intelligence in information systems research: A systematic literature review and research agenda," *Int. J. Inf. Manage.*, vol. 60, no. November 2020, p. 102383, 2021, doi: 10.1016/j.ijinfomgt.2021.102383.
  - [3] D. Angraina and A. Putri, "Analisis Sentimen Pengguna Aplikasi Google Meet Menggunakan Algoritma Support Vector Machine," vol. 3, no. 3, pp. 472–478, 2022, doi: 10.37859/coscitech.v3i3.4260.
  - [4] D. Safrida Putri and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi Pospay Dengan Algoritma Support Vector Machine," vol. 11, no. 01, pp. 32–40, 2023, doi: 10.33884/jif.v11i01.6611.
  - [5] L. Rajaoarisoa, R. Randrianandraina, G. J. Nalepa, and J. Gama, "Decision-making systems improvement based on explainable artificial intelligence approaches for predictive maintenance," *Eng. Appl. Artif. Intell.*, vol. 139, no. PB, p. 109601, 2025, doi: 10.1016/j.engappai.2024.109601.
  - [6] M. Asrol, M. Yani, Machfud, P. Papilo, S. Mursida, and Marimin, "Design of intelligent decision support system for supply chain sustainability assessment," *Procedia Comput. Sci.*, vol. 227, pp. 659–669, 2023, doi: 10.1016/j.procs.2023.10.570.
  - [7] I. Maulana, W. Apriandari, and A. Pambudi, "Analisis Sentimen Berbasis Aspek Terhadap Ulasan Aplikasi MyPertamina Menggunakan Support Vector Machine," vol. 6, no. 2, pp. 172–181, 2023, doi: 10.36080/idealisis.v6i2.3022.
  - [8] S. Behmel, M. Damour, R. Ludwig, and M. J. Rodriguez, "Intelligent decision-support system to plan, manage and optimize water quality monitoring programs: design of a conceptual framework," *J. Environ. Plan. Manag.*, vol. 64, no. 4, pp. 703–733, 2021, doi: 10.1080/09640568.2020.1782858.
  - [9] M. Yazdani, S. Shahriari, and M. Haghani, "Progress in Disaster Science Real-time decision support model for logistics of emergency patient transfers from hospitals via an integrated optimisation and machine learning approach," *Prog. Disaster Sci.*, vol. 25, no. May 2024, p. 100397, 2025, doi: 10.1016/j.pdisas.2024.100397.
  - [10] K. Coussement and D. F. Benoit, "Interpretable data science for decision making," *Decis. Support Syst.*, vol. 150, no. August, p. 113664, 2021, doi: 10.1016/j.dss.2021.113664.
  - [11] M. A. Mochinski *et al.*, "Developing an Intelligent Decision Support System for large-scale smart grid communication network planning," *Knowledge-Based Syst.*, vol. 283, no. October 2023, p. 111159, 2024, doi: 10.1016/j.knosys.2023.111159.
  - [12] M. Oktafani and P. T. Prasetyaningrum, "Implementasi Support Vector Machine Untuk Analisis Sentimen Komentar Aplikasi Tanda Tangan Digital," vol. 15, no. 1, pp. 10–19, 2022, doi: 10.33005/sibc.v15i1.4.
  - [13] E. Rizqi Mar'atus Sholihah, I. G. Susrama Mas Diyasa, and E. Yulia Puspaningrum, "Perbandingan Kinerja Kernel Linear Dan RBF Support Vector Machine Untuk Analisis Sentimen Ulasan Pengguna Kai Access Pada Google Play Store," vol. 8, no. 1, pp. 728–733, 2024, doi: 10.36040/jati.v8i1.8800.
  - [14] S. Wang, J. Ren, and R. Bai, "A Regularized Attribute Weighting Framework for Naive Bayes," *IEEE Access*, vol. 8, pp. 225639–225649, 2020, doi: 10.1109/ACCESS.2020.3044946.
  - [15] D. Li, J. Sun, H. Yang, and X. Wang, "An Enhanced Naive Bayes Model for Dissolved Oxygen Forecasting in Shellfish Aquaculture," *IEEE Access*, vol. 8, pp. 217917–217927, 2020, doi: 10.1109/ACCESS.2020.3042180.
  - [16] S. Shitharth, P. R. Kshirsagar, P. K. Balachandran, K. H. Alyoubi, and A. O. Khadidos, "An Innovative Perceptual Pigeon Galvanized Optimization (PPGO) Based Likelihood Naïve Bayes (LNB) Classification Approach for Network Intrusion Detection System," *IEEE Access*, vol. 10, pp. 46424–46441, 2022, doi: 10.1109/ACCESS.2022.3171660.
  - [17] S. Ruan, H. Li, C. Li, and K. Song, "Class-specific deep feature weighting for naïve bayes text classifiers," *IEEE Access*, vol. 8, pp. 20151–20159, 2020, doi: 10.1109/ACCESS.2020.2968984.
  - [18] N. Rachmawati Oktaria Mardiyanto, K. Kusriani, and N. Ferry Wahyu Wibowo, "Analisis Sentimen Pengguna Aplikasi Bank Syariah Indonesia Dengan menggunakan Algoritma Support Vector Machine (Svm)," vol. 4, no. 1, pp. 9–15, 2023, doi: 10.46764/teknimedia.v4i1.85.
  - [19] R. Wahyudi and G. Kusumawardana, "Analisis Sentimen Pada Aplikasi Grab Di Google Play Store Menggunakan Support Vector Machine," vol. 8, no. 2, pp. 200–207, 2021, doi: 10.31294/ji.v8i2.9681.
  - [20] J. Fasilkom, "Peningkatan Akurasi Klasifikasi Tutupan

- 
- Lahan Menggunakan Random Forest pada Data Sentinel-2 di Jambi Author : Akhiyar Waladi,” vol. 15, no. 1, pp. 17–24, 2025.
- [21] B. C. Octariadi, “Penerapan Algoritma ( Naïve Bayes ) Untuk Memprediksi Penyakit Diare,” vol. 15, no. 1, pp. 49–56, 2025.
- [22] J. Sanjaya, B. Priyatna, and S. S. Hilabi, “Analisis Sentimen Terhadap Opini Proyek Kereta Cepat Menggunakan Metode Naïve Bayes Classifier,” *J. Fasilkom*, vol. 14, no. 1, pp. 263–270, 2024.
- [23] S. Naiem, A. E. Khedr, A. M. Idrees, and M. I. Marie, “Enhancing the Efficiency of Gaussian Naïve Bayes Machine Learning Classifier in the Detection of DDOS in Cloud Computing,” *IEEE Access*, vol. 11, no. October, pp. 124597–124608, 2023, doi: 10.1109/ACCESS.2023.3328951.
- [24] B. Han, H. Shin, Y. Kim, J. Choi, and Y. Lee, “HEaaN-NB: Non-Interactive Privacy-Preserving Naive Bayes Using CKKS for Secure Outsourced Cloud Computing,” *IEEE Access*, vol. 12, no. August, pp. 110762–110780, 2024, doi: 10.1109/ACCESS.2024.3438161.
- [25] A. Abraham *et al.*, “Naïve Bayes Approach for Word Sense Disambiguation System with a Focus on Parts-of-Speech Ambiguity Resolution,” *IEEE Access*, vol. 12, no. August, pp. 126668–126678, 2024, doi: 10.1109/ACCESS.2024.3453912.