Journal of Information Technology and Computer Science (INTECOMS)
Volume 8 Nomor 5, Tahun 2025
e-ISSN : 2614-1574
p-ISSN : 2621-3249

ANALISIS SENTIMEN BERDASARKAN ULASAN GOOGLE PLAY STORE PADA APLIKASI DANA MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM)

SENTIMENT ANALYSIS BASED ON GOOGLE PLAY STORE REVIEWS OF THE DANA APPLICATION USING THE SUPPORT VECTOR MACHINE (SVM) METHOD

Ilyasa Garuda Putra Alva[1], Eri Zuliarso[2]
Universitas Stikubank[1,2]
ilyasa.garuda03@gmail.com[1]

ABSTARCT

This study aim to classify user sentiment from reviews of the DANA application collected from the Google Play Store using teh Support Vector Machine (SVM) algorithm. The data were gathered through web crawling and processed through preprocessing stages such as case folding, tokenization, filtering, stopword removal, and stemming. Reviews were then transformed into numerical representations using the TF-IDF method and classified into two sentiment categories: positive and negative. The model was evaluated using accuracy, precision, recall, and F1-score metrics. The result show that the SVM method delivers high classification performance with an accuracy of 88,5%, precision of 90,2%, recall 95,1%, and F1-score 92,5%. The findings were visualized using sentiment distribution histograms and WordCloud to highlight dominant words. This research provides insights into user perceptions and serves as a foundation for improving the DANA applications service quality.

Keywords*: Sentiment Analysis, DANA, Support Vector Machine, TF-IDF, User Review*

ABSTRAK

Penelitian ini bertujuan untuk mengklasifikasikan sentimen dari ulasan pengguna terhadap alikasi DANA yang diperoleh dari Google Play Store menggunakan algoritma Support Vector Machine (SVM). Data dikumpulkan melalui teknik web crawling dan diproses melalui tahapan preprocessing seperti case folding, tokenisasi, filtering, stopword removal, dan stemming. Ulasan kemudian dikonversi menjadi representasi numerik menggunakan metode TF-IDF dan diklasifikasikan ke dalam dua kategori sentimen: positif dan negatif. Evaluasi model dilakukan menggunakan metrik accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa metode SVM memberikan performa klasifikasi yang tinggi dengan nilai accuracy sebesar 88,5%, precision 90,2%, recall 95,1%, dan F1-score 92,5%. Visualisasi hasil dilakukan melalui hitogram distribusi sentimen dan WordCloud untuk menampilkan kata-kata dominan. Penelitian ini memberikan wawasan terhadap persepsi pengguna serta dapat dijadikan dasar pengembangan layanan aplikasi DANA.

Kata Kunci: Analisis Sentimen, DANA, Support Vector Machine, TF-IDF, Ulasan Pengguna.

PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah memberikan dampak besar terhadap berbagai aspek kehidupan, termasuk sektor keuangan. Salah satu inovasi yang muncul adalah aplikasi dompet digital *(e-wallet)* seperti DANA, yang memungkinkan pengguna melakukan transaksi secara cepat, aman, dan efisien[1]. DANA telah menjadi salah satu platform finansial digital terpopuler di Indonesia dengan jumlah unduhan tertinggi di Google Play Store [2]

Seiring bertambahnya jumlah pengguna, volume ulasan yang diberikan oleh pengguna di Google Play Store juga meningkat. Ulasan tersebut mencerminkan opini masyarakat terhadap layanan aplikasi, baik berupa pujian maupun keluhan. Namun, data ini bersifat tidak terstruktur dan berjumlah sangat besar, sehingga tidak memungkinkan untuk dianalisis secara manual. Oleh karena itu, diperlukan pendekatan otomatis menggunakan teknik *text mining*, khususnya *sentimen analysis*, untuk mengelompokkan opini menjadi sentimen positif dan negatif [3], [4].

Salah satu metode yang terbukti efektif dalam klasifikasi teks berdimensi tinggi adalah *Support Vector Machine (SVM).* SVM bekerja dengan membentuk *hyperline* optimal yang memisahkan data

ke dalam kelas tertentu dengan margin maksimum [5]. Metode ini telah banyak digunakan dalam peneliatian analisis sentimen terhadap ulasan aplikasi digital dan terbukti memberikan hasil yang kompetitif, terutama jika dikombinasikan dengan motode ekstraksi fitur seperti *Term Frequency-Inverse Document Frequency (TF-IDF)* [6], [7][8][9].

Penelitian ini bertujuan untuk mengevaluasi efektivitas algoritma SVM dalam mengklasifikasikan sentimen ulasan pengguna aplikasi DANA. Selain itu, penelitian ini juga berupaya memberikan gambaran umum mengenai persepsi masyarakat terhadap kualitas layanan aplikasi tersebut. Hasil dari penelitian diharapkan dapat memberi masukan yang bermanfaat bagi pengembang dalammeningkatkan kualitas dan keandalan layanan [10][11][12][13][14][15][16].

METODE

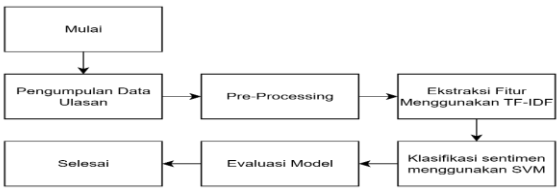
Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen yang bertujuan untuk menganaisis sentimen pengguna terhada aplikasi dompet digital DANA berdasarkan ulasan yang terdapat di Google Play Store. Metode yang digunakan dalam pengolahan data adalah Support Vector Machine (SVM) yang merupakan algoritma supervised learning dan terbukti efektif dalam klasifikasi data teks berdimensi tinggi [1].

Pengumpulan Data

Data dikumpulkan melalui teknik web scraping menggunakan library Google Play Scraper API berbasis Pyhton. Jumlah data yang diperoleh sebanyak 18.631 ulasan pengguna terhadap aplikasi DANA. Teknik ini serupa dengan pendekatan yang digunakan oleh [6], yang juga mengambil data ulasan aplikasi dompet digital untuk analisis sentimen.

Tahapan Penelitian

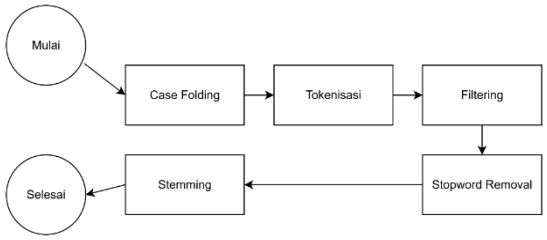
Adapun tahapan dalam penelitian ini adalah sebagai berikut:



Gambar 1. Tahapan Penelitian

Pra-Pemrosesan Data

Data yang diperoleh kemudian diproses melalui beberapa tahapan text processing untuk menghasilkan data yang bersih dan siap digunakan:



Gambar 2. Tahapan Pra-Pemrosesan Data

Ekstraksi Fitur TF-IDF

Data hasil preprocessing kemudia dikonversikan menjadi bentuk numerik menggunakan metode Term Frequency-Inverse (TF-IDF). TF menghitung frekuensi kemunculan kata dalam satu ulasan, sedangkan IDF mengukur seberapaumum kata tersebut dalam seluruh korpus [16]. Rumus yang digunakan adalah sebagai berikut:

a. Term Frequency (TF):

$$TF(t) = \frac{Jumlah\ kemunculan\ kata\ t\ dalam\ dokumen}{Jumlah\ seluruh\ kata\ dalam\ dokumen}$$

b. Inverse Document Frequency (IDF):

$$IDF(t) = log\left(\frac{Jumlah\ dokumen}{Jumlah\ dokumen\ mengandung\ kata\ t}\right)$$

c. TF-IDF:

TF-IDF(t)=TF(t)×IDF (t)

Klasifikasi Sentimen

Klasifikasi dilakukan menggunakan algoritma Support Vector Machine (SVM) dengan kernel linear. SVM digunakan untuk membagi data ke dalam dua kategori sentimen: positif dan negatif. Algoritma ini bekerja dengan membangun hyperlane

pemisah terbaik antar kelas data dan memiliki performa tinggi untuk data teks [4], [5].

Evaluasi Model

Untuk mengukur performa model klasifikasi, digunakan beberapa metrik evaluasi, yaitu:

a. *Accuracy* (Akurasi): Mengukur rasio prediksi yang benar terhadap keseluruhan data.

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

b. *Precision* (Presisi): Mengukur akurasi prediksi positif.

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

c. *Recall:* Mengukur rasio data positif yang berhasil terprediksi

$$Recall = \frac{TP}{TP + FP} \times 100\%$$

d. F1-Score: Kombinasi harmonis antara precision dan recal.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

.

HASIL DAN PEMBAHASAN

Pada bagian ini disajikan proses klasifikasi sentimen terhadap ulasan pengguna aplikasi DANA menggunakan metode *Support Vector Machine (SVM).*

Pengumpulan Data

Data dalam penelitian ini diperoleh dari ulasan pengguna aplikasi DANA yang tersedia di Google Play Store. Pengumpulan data dilakukan secara otomatis menggunakan teknik *web crawling*, yaitu proses mengambil data dari situs web secara terprogram dengan menggunakan Visual Studio Code sebagai alat bantu.

```
from google_play_scraper import Sort, reviews
import pandas as pd

# Crawl ulasan aplikasi DANA
result, _ = reviews(
    'id.dana',  # ID aplikasi DANA di Play Store
    lang='id',  # Bahasa Indonesia
    country='id',  # Negara Indonesia
    sort=Sort.NEWEST,  # Sortir ulasan terbaru
    count=1000,  # Jumlah ulasan yang diambil
    filter_score_with=None  # Ambil semua skor (1-5)
)

# Konversi ke DataFrame
df = pd.DataFrame(result)

# Simpan ke file CSV
df[['userName', 'score', 'content', 'at']].to_csv('ulasan_dana.csv', index=False)

print("✅ Data berhasil disimpan ke 'ulasan_dana.csv'")
```

Gambar 3. Kode *Crawling Data*

Gambar 3 adalah kode langkah awal dalam proses analisis sentimen, yang digunakan untuk mengambil data ulasan pengguna aplikasi DANA secara otomatis dan menyimpannya ke dalam format yang siap dianalisis lebih lanjut.

userName	score	content	at
Sugeng Ariyanto	5	Sangat membantu	2025-07-12 07:31:49
Daniel Sebastian Sinaga	3	Kemarin Saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	2025-07-12 07:31:34
dewi anggraini	5	istimewa	2025-07-12 07:31:08
sopiantri marlinda	5	mndhn mmbntu 🤣🤣🤣🤣🤣	2025-07-12 07:28:52
Jamal Anjasmara	5	mantappppp	2025-07-12 07:27:26
Cahaya Fatiha	5	baru pakai dana lagi, jadi awal2 agak ribet. tp oke lah biar LBH protec	2025-07-12 07:25:23
Ahdi Cikahuripanpurwakarta	5	ok	2025-07-12 07:24:55
Rusdi Anton	5	mantappppp	2025-07-12 07:24:35
ambia army	5	saya sangat senang dengan aplikasi ini, namun saya tidak bisa masuk ke akun DANA Saya.	2025-07-12 07:24:30
Tecu Roberto	5	good	2025-07-12 07:24:23

Gambar 4. Hasil *Crawling Data*

Gambar 4 adalah hasil *crawling data* ulasan pengguna aplikasi DANA dari Google Play Store yang sudah disimpan dalam format CSV. Tabel ini memuat informasi dari pengguna mengenai pengalaman mereka menggunakan aplikasi.

Hasil *Pre-Processing*

Tahapan *Pre-processing* data bertujuan untuk membersihkan dan menyiapkan data teks agar dapat diproses lebih lanjut dalam analisis sentimen. Proses ini dilakukan melalui beberapa tahap, yaitu: *case folding*, *tokenizing*, *filtering*, *stopword removal*, dan *stemming*. Hasil dari setiap tahap dirinci sebagai berikut:

a. ***Case Folding***

```
import pandas as pd

# Load data
df = pd.read_csv('ulasan_dana.csv')

# Terapkan case folding pada kolom ulasan (misal: kolom 'content')
df['casefolded'] = df['content'].astype(str).str.lower()

# Simpan hasilnya ke file baru (opsional)
df.to_csv('ulasan_dana_casefolded.csv', index=False)

print("✅ Case folding selesai. Kolom baru 'casefolded' ditambahkan.")
```

Gambar 5. Kode *Case Folding*

Gambar 5 adalah kode yang bertujuan untuk menstandarkan format tulisan ulasan sebelum dilanjutkan ke proses *tokenizing*, *stopword removal*, dan *stemming*.

userName	score	content	at	casefolded
Sugeng Ariyanto	5	Sangat membantu	2025-07-12 07:31:49	sangat membantu
Daniel Sebastian Sinaga	3	Kemarin Saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	2025-07-12 07:31:34	kemarin saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil
dewi anggraini	5	istimewa	2025-07-12 07:31:08	istimewa
sopiantri marlinda	5	mndhn mmbntu 🤣🤣🤣🤣🤣	2025-07-12 07:28:52	mndhn mmbntu 🤣🤣🤣🤣🤣
Jamal Anjasmara	5	mantappppp	2025-07-12 07:27:26	mantappppp

Gambar 6. Hasil *Case Folding*

Gambar 6 adalah tabel hasil yang menunjukkan bahwa proses *case folding* telah berhasil diterapkan dan hasilnya disimpan ke dalam kolom baru bernama *casefolded.*

b. ***Tokenizing***

```
import pandas as pd

# Load data hasil case folding (asumsinya sudah disimpan sebelumnya)
df = pd.read_csv('ulasan_dana_casefolded.csv')

# Fungsi tokenisasi: memecah teks menjadi list kata
def tokenize(text):
    return text.split()

# Terapkan ke kolom 'casefolded'
df['token'] = df['casefolded'].apply(tokenize)

# Simpan ke file baru (opsional)
df.to_csv('ulasan_dana_tokenized.csv', index=False)

print("✅ Tokenisasi selesai. Kolom baru 'token' ditambahkan.")
```

Gambar 7. Kode *Tokenizing*

Gambar 7 adalah kode *tokenizing.* Proses ini merupakan langkah penting dalam *pre-processing* data teks sebelum dilakukan proses analisis sentimen lebih lanjut, seperti *stopword removal*, dan *stemming.*

userName	score	content	at	casefolded	token
Sugeng Ariyanto	5	Sangat membantu	2025-07-12 07:31:49	sangat membantu	['sangat', 'membantu']
Daniel Sebastian Sinaga	3	Kemarin Saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	2025-07-12 07:31:34	kemarin saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	['kemarin', 'saya', 'transfer', 'disini', 'uangnya', 'tidak', 'masuk', 'padahal', 'bukti', 'transfer', 'nya', 'berhasil']
dewi anggraini	5	istimewa	2025-07-12 07:31:08	istimewa	['istimewa']
sopiantri marlinda	5	mndhn mmbntu 🤔🤔🤔🤔🤔	2025-07-12 07:28:52	mndhn mmbntu 🤔🤔🤔🤔🤔	['mndhn', 'mmbntu 🤔🤔🤔🤔🤔']
Jamal Anjasmara	5	mantappppp	2025-07-12 07:27:26	mantappppp	['mantappppp']

Gambar 8. Hasil *Tokenizing*

Gambar 8 adalah tabel bahwa proses *tokenizing* berhasil dilakukan dan setiap ulasan telah diubah menjadi kumpulan kata yang siap di proses dalam tahap analisis selanjutnya.

c. ***Filtering***

```
import pandas as pd
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

# Load data hasil tokenisasi
df = pd.read_csv('ulasan_dana_tokenized.csv')

# Inisialisasi stopword list dari Sastrawi
stop_factory = StopWordRemoverFactory()
stopwords = set(stop_factory.get_stop_words())

# Pastikan kolom token berupa list (jika data CSV tersimpan sebagai string list)
import ast
df['token'] = df['token'].apply(lambda x: ast.literal_eval(x))  # ubah dari string ke list

# Fungsi filtering
def filter_tokens(tokens):
    return [word for word in tokens if word not in stopwords]

# Terapkan filtering
df['filtered'] = df['token'].apply(filter_tokens)

# Simpan hasil ke file baru (opsional)
df.to_csv('ulasan_dana_filtered.csv', index=False)

print("✅ Filtering selesai. Kolom baru 'filtered' ditambahkan.")
```

Gambar 9. Kode *Filtering*

Gambar 9 adalah kode yang menjalankan proses *filtering* dengan menghapus kata-kata umum dari token ulasan pengguna.

userName	score	content	at	casefolded	token	filtered
Sugeng Ariyanto	5	Sangat membantu	2025-07-12 07:31:49	sangat membantu	['sangat', 'membantu']	['sangat', 'membantu']
Daniel Sebastian Sinaga	3	Kemarin Saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	2025-07-12 07:31:34	kemarin saya transfer disini uangnya tidak masuk padahal bukti transfer nya berhasil	['kemarin', 'saya', 'transfer', 'disini', 'uangnya', 'tidak', 'masuk', 'padahal', 'bukti', 'transfer', 'nya', 'berhasil']	['kemarin', 'transfer', 'disini', 'uangnya', 'masuk', 'padahal', 'bukti', 'transfer', 'nya', 'berhasil']
dewi anggraini	5	istimewa	2025-07-12 07:31:08	istimewa	['istimewa']	['istimewa']
sopiantri marlinda	5	mndhn mmbntu 🤔🤔🤔🤔🤔	2025-07-12 07:28:52	mndhn mmbntu 🤔🤔🤔🤔🤔	['mndhn', 'mmbntu 🤔🤔🤔🤔🤔']	['mndhn', 'mmbntu 🤔🤔🤔🤔🤔']
Jamal Anjasmara	5	mantappppp	2025-07-12 07:27:26	mantappppp	['mantappppp']	['mantappppp']

Gambar 10. Hasil *Filtering*

Gambar 10 adalah hasil dari *filtering* yang menampilkan kata-kata penting tersisa setelah proses *filtering* dilakukan.

d. ***Stopword Removal***

```
import pandas as pd
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

# Load data hasil case folding
df = pd.read_csv('ulasan_dana_casefolded.csv')

# Inisialisasi stopword dari Sastrawi
stop_factory = StopWordRemoverFactory()
stopwords = set(stop_factory.get_stop_words())

# Fungsi untuk menghapus stopword
def remove_stopwords(text):
    words = text.split()
    filtered_words = [word for word in words if word not in stopwords]
    return ' '.join(filtered_words)

# Terapkan stopword removal
df['no_stopwords'] = df['casefolded'].apply(remove_stopwords)

# Simpan hasilnya (opsional)
df.to_csv('ulasan_dana_no_stopwords.csv', index=False)

print("✅ Stopword removal selesai. Kolom 'no_stopwords' ditambahkan.")
```

Gambar 11. Kode *Stopword Removal*

Gambar 11 adalah kode untuk melakukan proses *stopword removal.* Tahapan ini berperan penting dalam *preprocessing* karena dapat mengurangi noise pada data dan meningkatkan akurasi model klasifikasi seperti SVM pada tahap selanjutnya.

userName	score	content	at	casefolded	no_stopwords
Farhan Q	1	banyak scam aplikasi dajjal dilarang download!!!	2025-07-13 04:36:26	banyak scam aplikasi dajjal dilarang download!!!	banyak scam aplikasi dajjal dilarang download!!!
kang inay	2	penggantian nomor dana yg terdaftar kenapa tidak ada bisa, alasan sy ganti nomor karena nomor yg terdaftar sudah tidak aktif, terimakasih	2025-07-13 04:34:05	penggantian nomor dana yg terdaftar kenapa tidak ada bisa, alasan sy ganti nomor karena nomor yg terdaftar sudah tidak aktif, terimakasih	penggantian nomor dana yg terdaftar bisa, alasan sy ganti nomor nomor yg terdaftar aktif, terimakasih
Akbar Rizqi	5	ok	2025-07-13 04:31:19	ok	
Muhammad Aisar	5	terbaik	2025-07-13 04:30:57	terbaik	terbaik
Achmad Sobiq	5	sangat besar jasanya	2025-07-13 04:30:54	sangat besar jasanya	sangat besar jasanya

Gambar 12. Hasil *Stopword Removal*

Gambar 12 adalah hasil dari *stopword removal*, tahapan ini ini bertujuan membersihkan tekas agar siap untuk analisis sentimen dengan cara menghilangkan kata-kata tidak ppenting yang menyamakan bentuk huruf.

e. ***Stemming***

```
import pandas as pd
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

# Load data
df = pd.read_csv('ulasan_dana_no_stopwords.csv')

# Bersihkan data NaN dan pastikan string
df['no_stopwords'] = df['no_stopwords'].fillna('').astype(str)

# Inisialisasi stemmer
stem_factory = StemmerFactory()
stemmer = stem_factory.create_stemmer()

# Fungsi stemming
def stem_text(text):
    return stemmer.stem(text)

# Terapkan stemming
df['stemmed'] = df['no_stopwords'].apply(stem_text)

# Simpan hasil
df.to_csv('ulasan_dana_stemmed.csv', index=False)

print("✅ Stemming selesai tanpa error.")
```

Gambar 13. Kode *Stemming*

Gambar 13 adalah tahapan untuk mengubah setiap kata menjadi bentuk dasarnya agar lebih representatif dan konsisten saat dianalisis oleh model klasifikasi.

userName	score	content	at	casefolded	no_stopwords	stemmed
Farhan Q	1	banyak scam aplikasi dajjal dilarang download!!!	2025-07-13 04:36:26	banyak scam aplikasi dajjal dilarang download!!!	banyak scam aplikasi dajjal dilarang download!!!	banyak scam aplikasi dajjal larang download
kang inay	2	penggantian nomor dana yg terdaftar kenapa tidak ada bisa, alasan sy ganti nomor karena nomor yg terdaftar sudah tidak aktif, terimakasih	2025-07-13 04:34:05	penggantian nomor dana yg terdaftar kenapa tidak ada bisa, alasan sy ganti nomor karena nomor yg terdaftar sudah tidak aktif, terimakasih	penggantian nomor dana yg terdaftar bisa, alasan sy ganti nomor nomor yg terdaftar aktif, terimakasih	ganti nomor dana yg daftar bisa alas sy ganti nomor nomor yg daftar aktif terimakasih
Akbar Rizqi	5	ok	2025-07-13 04:31:19	ok		
Muhammad Aisar	5	terbaik	2025-07-13 04:30:57	terbaik	terbaik	baik
Achmad Sobiq	5	sangat besar jasanya	2025-07-13 04:30:54	sangat besar jasanya	sangat besar jasanya	sangat besar jasa

Gambar 14. Hasil *Stemming*

Gambar 14 adalah proses *stemming* yang berhasil mengubah kata ke bentuk dasarnya, sehingga teks ulasan menjadi lebih sederhana dan siap untuk analisis sentimen.

Labeling Data

Labeling data adalah proses memberikan kategori sentimen (misalnya positif atau negatif) pada setiap data ulasan, sehingga bisa digunakan untuk pelatihan model klasifikasi (seperti SVM).

a. *Score* 4 atau 5 = label (1) positif
b. *Score* 1 atau 2 = label (0) negative
c. *Score* 3 bisa diabaikan (netral)

```
import pandas as pd

# Load dataset
df = pd.read_csv('ulasan_dana.csv')

# Buat kolom label otomatis: 1 = positif, 0 = negatif
def label_sentiment(score):
    if score >= 4:
        return 1  # Positif
    elif score <= 2:
        return 0  # Negatif
    else:
        return None  # Netral (bisa dihapus nanti)

df['label'] = df['score'].apply(label_sentiment)

# Hapus data netral jika hanya ingin 2 kelas
df = df.dropna(subset=['label'])

# Simpan ke file baru
df.to_csv('ulasan_dana_labeled.csv', index=False)

print("✅ Labeling selesai. File disimpan sebagai 'ulasan_dana_labeled.csv'")
```

Gambar 15. Kode *Labeling Data*

Hasil Analisis

Setelah melalui tahapan *preprocessing* dan pelatihan model klasifikasi menggunakan Support Vector Machine (SVM), langkah berikutnya adalah menerapkan model untuk mengklasifikasikan ulasan pengguna aplikasi DANA menjadi dua kategori sentimen, yaitu positif dan negatif. Hasil analisis ini memberikan gambaran umum terhadap persepsi pengguna terhadap aplikasi DANA berdasarkan ulasan mereka di Google Play Store.

Histogram Disrtibusi Sentimen

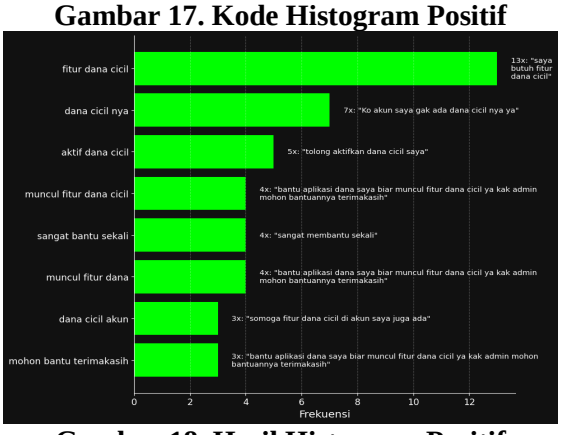
Histogram Distribusi Sentimen adalah gambaran jumlah dan persentase data ulasan yang termasuk dalam masing-masing kategori sentimen, seperti positif dan negatif.

```
# prompt: buatkan histogram sentimen positif berdasarkan 1 data tambahkan kalimat atau isi  per data nya

import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter

# 1. Load data
df = pd.read_csv('ulasan_dana_clean_labeled.csv')
df = df[df['clean_review'].notnull()]

# Convert 'label' column to numerical representation if needed
# Assuming 'positif' maps to 1 and 'negatif' maps to 0
if df['label'].dtype == 'object':
    df['label'] = df['label'].map({'positif': 1, 'negatif': 0}).fillna(-1) # Map 'positif' to 1, 'negatif' to 0, others to -1
    df = df[df['label'] != -1] # Remove rows that were not 'positif' or 'negatif'
else:
    # Ensure label is 0 for negative and 1 for positive if it's already numerical
    df['label'] = df['label'].map({0: 0, 1: 1}).fillna(-1) # Assuming 0 is negative, 1 is positive
    df = df[df['label'] != -1]


# Filter hanya review dengan label positif (1)
positive_reviews_df = df[df['label'] == 1].copy()

# Fungsi untuk ekstrak frasa 3-5 kata
def extract_phrases_with_content(df_row, n_min=3, n_max=5):
    text = df_row['clean_review']
    original_content = df_row['content'] # Get the original content
    words = text.split()
    phrases_info = []
    for n in range(n_min, n_max + 1):
        for i in range(len(words) - n + 1):
            phrase = ' '.join(words[i:i + n])
            phrases_info.append({'phrase': phrase, 'content': original_content}) # Store phrase and original content
    return phrases_info

# Kumpulkan frasa dan konten asli dari ulasan positif
all_positive_phrases_info = []
for index, row in positive_reviews_df.iterrows():
    all_positive_phrases_info.extend(extract_phrases_with_content(row))
```

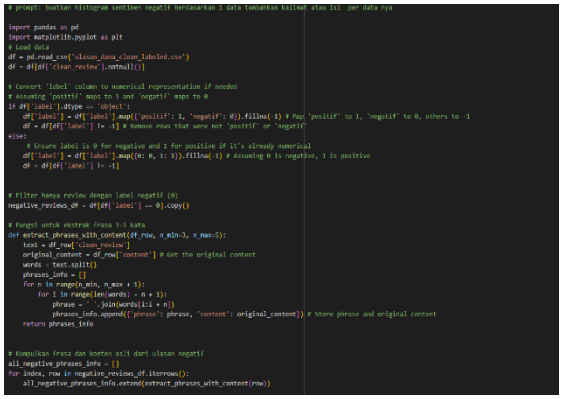
Gambar 16. Kode Histogram Positif

```
# Hapus duplikasi frasa, simpan salah satu contoh konten
unique_phrases_df = phrases_df.drop_duplicates(subset=['phrase']).sort_values(by='frequency', ascending=False)

# Ambil top 10 frasa
top_pos_phrases_with_content = unique_phrases_df.head(10)

# 🔧 Plot dengan teks besar dan tema hitam
plt.style.use('dark_background')
fig, ax = plt.subplots(figsize=(10, 8))

# Histogram Frasa Positif
phrases = top_pos_phrases_with_content['phrase'].tolist()
freq = top_pos_phrases_with_content['frequency'].tolist()
contents = top_pos_phrases_with_content['content'].tolist() # Original content for each phrase

# Reverse for plotting from highest to lowest frequency from bottom up
phrases.reverse()
freq.reverse()
contents.reverse()

bars = ax.barh(phrases, freq, color='lime')

ax.set_title("Top 10 Kalimat Positif (3-5 kata)", color='white', fontsize=18)
ax.set_xlabel("Frekuensi", color='white', fontsize=14)
ax.tick_params(colors='white', labelsize=12)
ax.grid(True, axis='x', linestyle='--', alpha=0.3)

# Ubah warna background
ax.set_facecolor('#111111')
fig.patch.set_facecolor('#111111')

# Tambahkan teks (frekuensi dan contoh kalimat) di sebelah bar
for bar, phrase, frequency, content in zip(bars, phrases, freq, contents):
    ax.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2,
            f'{frequency}x: "{content}"', # Display frequency and original content
            va='center', ha='left', color='white', fontsize=10, wrap=True) # Use wrap=True

# Adjust x-axis limits to accommodate text labels
ax.set_xlim(0, max(freq) + max(freq)*1.5) # Adjust multiplier as needed

plt.tight_layout()
```

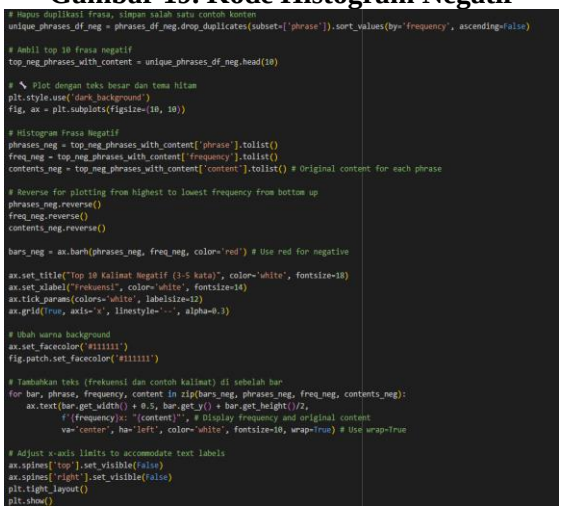
Gambar 17. Kode Histogram Positif



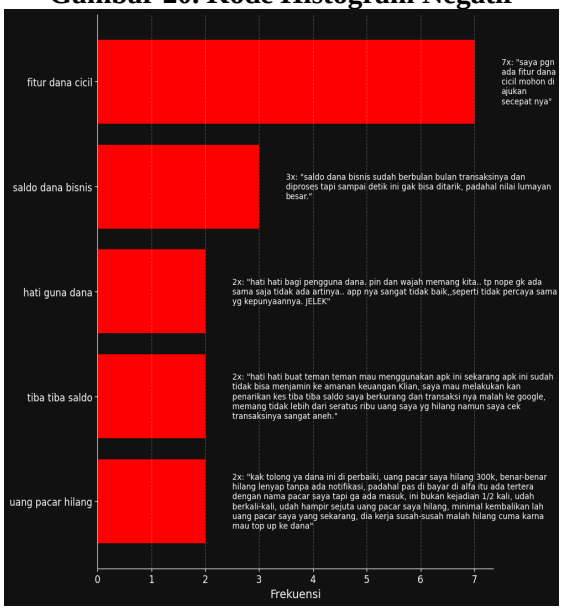
Gambar 18. Hasil Histogram Positif



Gambar 19. Kode Histogram Negatif



Gambar 20. Kode Histogram Negatif

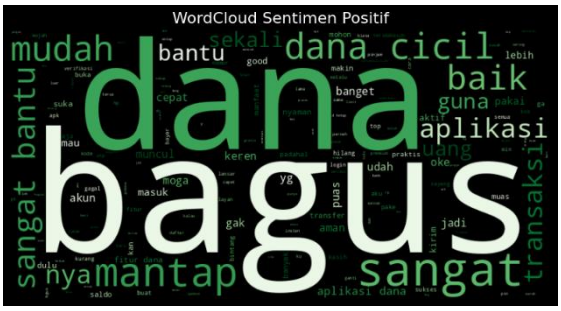


Gambar 21. Hasil Histogram Negatif

Wordcloud Sentimen

WordCloud sentimen adalah visualisasi berbentuk kumpulan kata yang menunjukkan kata-kata yang paling sering muncul dalam data ulasan berdasarkan kategori sentimen, yaitu positif dan negatif.

Gambar 22. Kode Wordcloud Positif



Gambar 23. Hasil Wordcloud Positif

Gambar 24. Kode Wordcloud Negatif

Gambar 25. Hasil Wordcloud Negatif

Klasifikasi Analisis Sentimen Menggunakan Support Vector Machine

Pada tahapan ini, klasifikasi analisis sentimen dilakukan menggunakan algoritma Support Vector Machine (SVM). Model diuji mengunakan data uji dan hasil klasifikasi divisualisasikan dan dievaluasi

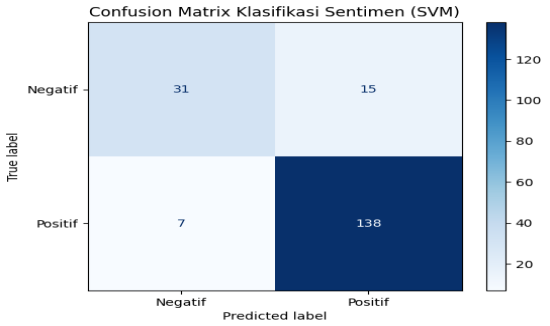
menggunakan akurasi, presisi, recall, dan F1-Score.

Evaluasi Model

Evaluasi model dilakukan untuk mengukur kinerja algoritma Support Vector Machine (SVM) dalam mengklasifikasikan sentimen ulasan pengguna aplikasi DANA menjadi sentimen positif dan negatif. Evaluasi dilakukan dengan menggunakan akurasi, presisi, recall, dan F1-Score.

Confusion Matrix

Confusion Matrix digunakan untuk memvisualisasikan hasil klasifikasi model pada data pengujian dan menampilkan sebaran prediksi benar dan salah.



Gambar 26. *Confusion Matrix*

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

     Negatif       0.82      0.67      0.74        46
     Positif       0.90      0.95      0.93       145

    accuracy                           0.88       191
   macro avg       0.86      0.81      0.83       191
weighted avg       0.88      0.88      0.88       191


Confusion Matrix:
[[ 31  15]
 [  7 138]]
```

Gambar 27. Hasil Akurasi

Berikut ini adalah penjelasan nilai berdasarkan gambar 4.16:

1. TN (True Negative) = 31
   Ada 31 data yang benar-benar negatif, dan model juga memprediksinya sebagai negatif.
2. FP (False Positive) = 15
   Ada 15 data yang sebenarnya negatif, tapi model salah memprediksi sebagai positif.
3. FN (False Negative) = 7
   Ada 7 data yang sebenarnya positif, tapi model salah memprediksi sebagai negatif.
4. TP (True Positive) = 138
   Ada 138 data yang benar-benar positif, dan model juga memprediksinya sebagai positif.

Total Data Uji

Total=TP+TN+FP+FN=31+15+7+138=191

Evaluasi Model

1. Akurasi = (TP + TN) / Total
   = (138 + 31) / 191 ≈ 0.885 atau 88.5%
2. *Precision* (positif) = TP / (TP + FN)
   = 138 / (138 + 15) ≈ 0.902
3. *Recall* (positif) = TP / (TP + FN)
   = 138 / (138 + 7) ≈ 0.951
4. *F1-Score* = 2 × (*Precision* × *Recall) / (Precision* + *Recall)*

SIMPULAN

Penelitian ini berhasi menerapkan metode *Support Vector Machine (SVM)* untuk mengklasifikasikan sentimen pengguna terhadap aplikasi dompet digital DANA berdasarkan ulasan yang diambil dari Google Play Store. Proses dimulai dengan pengumpulan data ulasan, kemudian dilanjutkan dengan tahapan pra-pemrosesan data seperti *case folding, tokenizing, filtering, stopword removal*, dan, *stemming* untuk membersihkan serta menyederhanakan teks. Setelah itu, dilakukan ekstraksi fitur menggunakan metode Term Frequency-Inverse Document Frequency (TF-IDF) guna mengubah teks menjadi bentuk numerik yang dapat diolah oleh model machine learning.

Model klasifikasi menggunakan SVM kemudian dibangun dan diuji menggunakan metrik evaluasi seperti akurasi, presisi, recall dan f1-*score*. Hasil evaluasi menunjukkan bahwa SVM mampu memberikan performa yang cukup baik dalam membedakan ulasan dengan sentimen positif dan negatif. Hal ini membuktikan bahwa pendekatan ini efektif untuk menangani data teks berdimensi tinggi seperti ulasan pengguna aplikasi. Temuan dari penelitian ini diharapkan dapat menjadi masukan bagi pengembang aplikasi dalam memahami persepsi dan

pengalaman pengguna secara lebih mendalam, serta membantu meningkatkan kualitas layanan berbasis data sentimen.

DAFTAR PUSTAKA

[1] T. Sugihartono, R. Rian, and C. Putra, "Penerapan Metode Support Vector Machine dalam Classifikasi Ulasan Pengguna Aplikasi Mobile JKN," vol. 7, no. 2, pp. 144–153, 2024.

[2] Harenanda Alfigo, "ANALISIS SENTIMEN PENGGUNA APLIKASI DANA PADA ULASAN GOOGLE PLAY MENGGUNAKAN METODE K-FOLD CROSS VALIDATION DAN SUPPORT VECTOR MACHINE (SVM)," 2023.

[3] B. Filemon,) Viny, C. Mawardi,) Novario, and J. Perdana, "Jurnal Ilmu Komputer dan Sistem Informasi PENGGUNAAN METODE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI SENTIMEN E-WALLET."

[4] A. B. Muzayyanah, R. E. Pawening, and Z. Arifin, "ANALISIS SENTIMEN PADA ULASAN APLIKASI EHADRAH DI GOOGLE PLAYSTORE MENGGUNAKAN SUPPORT VECTOR MACHINE," 2024. [Online]. Available: http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/indexAyuBasirotulMuzayyanah|http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index|

[5] C. M. Putri, M. Afdal, R. Novita, and M. Mustakim, "Perbandingan Evaluasi Kernel Support Vector Machine dalam Analisis Sentimen Chatbot AI pada Ulasan Google Play Store," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 7, no. 3, pp. 1236–1245, Jul. 2024, doi: 10.32493/jtsi.v7i3.41354.

[6] R. AL Anshari *et al.*, "MENGGUNAKAN METODE SUPPORT VECTOR MACHINE 1) INFO ARTIKEL ABSTRAK," vol. 2, no. 3, pp. 118–128, 2023, doi: 10.55123.

[7] M. R. Nadhif, D. Wisnu Brata, and B. Rahayudi, "Analisis Sentimen Data Ulasan Pengguna Aplikasi TIX ID di Indonesia pada Google Play Store menggunakan Support Vector Machine," 2022. [Online]. Available: http://j-ptiik.ub.ac.id

[8] W. E. Saputro, H. Yuana, and W. D. Puspitasari, "ANALISIS SENTIMEN PENGGUNA DOMPET DIGITAL DANA PADA KOLOM KOMENTAR GOOGLE PLAY STORE DENGAN METODE KLASIFIKASI SUPPORT VECTOR MACHINE," 2023.

[9] A. Athallah Muhammad *et al.*, "ANALISIS SENTIMEN PENGGUNA APLIKASI DANA BERDASARKAN ULASAN PADA GOOGLE PLAY MENGGUNAKAN METODE SUPPORT VECTOR MACHINE Prodi S1 Informatika / Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta".

[10] G. G. Warow and H. Pandia, "Analisis Sentimen Aplikasi Dana Menggunakan Naïve Bayes Classifier dan Support Vector Machine".

[11] S. Nadhifah, F. N. Aini, H. H. Kusumawardhani, and M. Y. Febrianto, "Analisis Sentiment Ulasan Aplikasi Gopay Pada Google Play Store Menggunakan Algoritma Support Vector Machine," vol. 14, no. 1, 2024.

[12] A. Zaiem Praghakusma and N. Charibaldi, "Komparasi Fungsi Kernel Metode Support Vector Machine untuk Analisis Sentimen Instagram dan Twitter (Studi Kasus : Komisi Pemberantasan Korupsi)," vol. 9, no. 2, pp. 33–42, 2021, doi: 10.12928/jstie.v8i3.xxx.

[13] M. Resa *et al.*, "PERBANDINGAN METODE SUPPORT VECTOR MACHINE DENGAN METODE LEXICON DALAM ANALISIS

SENTIMEN BAHASA INDONESIA," *Jurnal Teknologi Informasi)*, vol. 6, no. 1, [Online]. Available: https://github.com/fajri91/InSet.

[14] Supian Acuan, Revaldo Bagus Tri, Marhadi Nanda, Rahmaddeni, and Efrizoni Lusiana, "Penerapan SVM dan Word2Vec untuk Analisis Sentimen Ulasan Pengguna Aplikasi DANA," *Jurnal Ilmiah Komputasi*, vol. 23, no. 3, Sep. 2024, doi: 10.32409/jikstik.23.3.3642.

[15] A. Nurian and B. Nurina Sari, "ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI GOOGLE PLAY MENGGUNAKAN NAÏVE BAYES," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 11, no. 3, pp. 2830–7062, doi: 10.23960/jitet.v11i3%20s1.3348.

[16] Sudrajat Muhammad Riky, Atika Prima Dina, and Herlawati, "Implementasi Support Vector Machine (SVM) dan Naive Bayes untuk Analisis Sentimen Aplikasi KAI Access," *Jurnal ICT : Information Communication & Technology*, 2021.