

ANALISIS SENTIMEN APLIKASI JAKONE MOBILE PADA GOOGLE PLAY STORE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE

Lukman Nur Haq, Astriana Mulyani

Teknologi Informasi, Universitas Nusa Mandiri
Jalan Kramat Raya No. 18, Senen, Jakarta Pusat, Indonesia
lukman.nurhaq@gmail.com

ABSTRAK

JakOne Mobile adalah aplikasi layanan perbankan digital dari Bank DKI yang dirancang untuk memudahkan nasabah dalam melakukan berbagai transaksi keuangan secara praktis dan aman. Seiring dengan banyaknya pengguna yang memberikan ulasan terhadap aplikasi JakOne Mobile, Bank DKI perlu memahami sentimen umum yang dimiliki oleh para pengguna. Hal ini penting untuk mengetahui bagaimana aplikasi diterima oleh nasabah dan mengidentifikasi faktor-faktor yang paling memengaruhi kepuasan atau ketidakpuasan mereka. Diperlukan metode otomatis yang mampu mengklasifikasikan sentimen pengguna secara efisien dan akurat. *Support Vector Machine* (SVM) adalah metode yang dianggap efektif untuk melakukan klasifikasi data teks, namun perlu dibuktikan apakah SVM dapat bekerja secara optimal dalam mengklasifikasikan sentimen pada ulasan aplikasi JakOne Mobile. Dari 5.000 data yang diambil kemudian data dibersihkan, lalu diperoleh sebanyak 4.611 data didapatkan hasil 1.718 untuk hasil positif, 1.583 untuk hasil negatif dan 1.310 untuk hasil netral. Hasil penerapan *Support Vector Machine* pada aplikasi JakOne Mobile mendapatkan akurasi tertinggi pada pembagian data latih dan data uji 80:20 sebesar 87,34%, Presisi 91%, *Recall* 91%, dan *F1 Score* sebesar 91%.

Kata kunci : Analisis Sentimen, JakOne Mobile, Support Vector Machine

1. PENDAHULUAN

Dalam beberapa tahun terakhir, industri perbankan dan keuangan mengalami perkembangan pesat, khususnya dengan hadirnya layanan aplikasi digital yang memungkinkan pengguna melakukan berbagai transaksi keuangan dengan lebih mudah dan efisien. Aplikasi perbankan dan pembayaran tagihan kini telah menjadi bagian penting dari kehidupan masyarakat modern, yang menawarkan kemudahan dalam melakukan transaksi seperti *transfer*, pembayaran tagihan, hingga pembelian produk digital. Penggunaan aplikasi ini semakin meningkat seiring dengan perubahan gaya hidup masyarakat yang lebih mengutamakan kenyamanan dan kecepatan dalam bertransaksi.

Di balik kemudahan yang ditawarkan, respons pengguna terhadap layanan aplikasi perbankan dan pembayaran tagihan bervariasi. Beberapa pengguna memberikan ulasan positif terkait kemudahan dan fitur yang disediakan, sementara yang lain mungkin menyampaikan keluhan mengenai masalah teknis, keamanan, atau pengalaman pengguna yang tidak memuaskan. Oleh karena itu, penting bagi perusahaan penyedia layanan untuk mengetahui sentimen pengguna terhadap aplikasi mereka agar dapat terus meningkatkan kualitas layanan sesuai dengan kebutuhan dan harapan pengguna.

JakOne Mobile adalah aplikasi layanan perbankan digital dari Bank DKI yang dirancang untuk memudahkan nasabah dalam melakukan berbagai transaksi keuangan secara praktis dan aman. Aplikasi ini menggabungkan berbagai fitur perbankan, seperti transfer antar bank, pembayaran tagihan, pembelian produk digital (pulsa, paket data, dan lainnya), serta

fitur keuangan lainnya yang memungkinkan pengguna melakukan transaksi kapan saja dan di mana saja.

Keunggulan JakOne Mobile adalah integrasi dengan program Pemprov DKI Jakarta. Sementara kekurangan JakOne Mobile yaitu keterbatasan jangkauan dan dukungan, stabilitas dan performa aplikasi, keterbatasan fitur dibandingkan kompetitor, keterbatasan sosialisasi dan edukasi, serta isu keamanan dan kepercayaan.

Analisis sentimen merupakan metode yang tepat untuk memahami opini dan persepsi pengguna melalui data ulasan atau komentar mereka. Dengan teknik ini, perusahaan dapat mengidentifikasi apakah sentimen pengguna terhadap aplikasi mereka cenderung positif, negatif, atau netral. Salah satu metode yang efektif untuk melakukan analisis sentimen adalah *Support Vector Machine* (SVM). Metode SVM dikenal andal dalam mengklasifikasikan data teks dan menangani data dengan dimensi tinggi, yang sering kali ditemui pada data ulasan pengguna aplikasi. SVM bekerja dengan mencari hyperplane yang optimal untuk memisahkan data ke dalam kelas yang berbeda, sehingga cocok untuk mengidentifikasi dan mengklasifikasi sentimen pengguna.[1]

2. TINJAUAN PUSTAKA

2.1. Analisis Sentimen

Analisis sentimen adalah cabang dari pemrosesan bahasa alami (*Natural Language Processing*, NLP) yang berfokus pada identifikasi dan klasifikasi opini atau perasaan dalam teks. Dalam era digital saat ini, di mana interaksi sosial dan komunikasi terjadi secara online, analisis sentimen menjadi alat penting bagi perusahaan, organisasi, dan individu untuk memahami

persepsi publik terhadap produk, layanan, atau isu tertentu.[2]

Analisis sentimen dapat didefinisikan sebagai teknik yang digunakan untuk menentukan sentimen yang terkandung dalam teks, yang umumnya dikategorikan menjadi tiga kelas: positif, negatif, atau netral. Proses ini bertujuan untuk mendapatkan wawasan dari data tidak terstruktur, seperti ulasan produk, komentar di media sosial, atau artikel berita, dan mengubahnya menjadi informasi yang dapat diolah dan digunakan untuk pengambilan keputusan.[3]

Analisis sentimen dapat dilakukan dengan berbagai pendekatan. Jenis metode analisis sentiment adalah:

- Pendekatan Berbasis Kamus (*Lexicon-Based Approach*): Metode ini melibatkan penggunaan kamus atau daftar kata-kata yang telah dikategorikan berdasarkan sentimennya. Setiap kata dalam teks dievaluasi menggunakan kamus tersebut untuk menentukan sentimen keseluruhan.
- Pendekatan Berbasis Pembelajaran Mesin (*Machine Learning-Based Approach*): Pendekatan ini melibatkan pelatihan model menggunakan algoritma pembelajaran mesin untuk mengklasifikasikan teks berdasarkan sentimen. Model dilatih pada dataset yang telah diberi label, dan kemudian diterapkan pada data baru.
- Pendekatan Berbasis *Deep Learning*: Dengan perkembangan teknologi, pendekatan berbasis *deep learning* semakin populer. Metode ini menggunakan arsitektur jaringan saraf yang lebih dalam untuk menangkap konteks dan hubungan antar kata dalam teks.[4]

2.2. Google Play Store

Google Play Store, yang dikenal sebagai Play Store atau Google Play, adalah toko aplikasi online raksasa yang berfungsi sebagai pusat distribusi utama untuk perangkat Android. Sebelum Google Play Store, ada *Android Market* yang diluncurkan pada tanggal 22 Oktober 2008, tak lama setelah peluncuran sistem operasi Android itu sendiri, *Android Market* merupakan rumah bagi aplikasi pihak ketiga pertama untuk perangkat Android. Pada awalnya, hanya ada sekitar 13 aplikasi yang tersedia. Meskipun demikian, *Android Market* tumbuh pesat seiring dengan popularitas Android.

Pada tanggal 6 Maret 2012, Google mengumumkan perubahan besar. *Android Market*, bersama dengan layanan digital Google lainnya seperti *Google Music* dan *Google eBookstore*, disatukan di bawah satu payung merek yang baru dinamakan Google Play Store. Langkah ini menandai perubahan penting dalam strategi distribusi digital Google, dengan tujuan untuk menyederhanakan pengalaman pengguna dan menyediakan *platform* terpadu untuk berbagai jenis konten digital.

Sejak diluncurkan kembali sebagai Google Play Store, *platform* ini telah mengalami pertumbuhan dan evolusi yang luar biasa. Fitur-fitur baru terus ditambahkan, termasuk kemampuan untuk membeli

film, acara TV, musik, buku, dan bahkan berita. Google Play Store juga menjadi pusat bagi jutaan aplikasi dan *game* Android, baik yang gratis maupun berbayar.

Google Play Store telah mencapai statistik yang mengesankan. Pada tahun 2022, tercatat lebih dari 140 miliar unduhan konten dari platform ini. Dengan lebih dari 2,5 miliar pengguna aktif setiap bulan dan ketersediaan di 190 negara di seluruh dunia, Google Play Store adalah salah satu platform distribusi digital terbesar dan paling berpengaruh di dunia.

Dari awal yang sederhana sebagai *Android Market*, Google Play Store telah berkembang menjadi pusat konten digital yang sangat penting bagi ekosistem Android. Dengan jutaan aplikasi, game, film, acara TV, musik, buku, dan konten digital lainnya, Google Play Store terus menjadi sumber utama bagi pengguna Android untuk menemukan dan menikmati konten favorit mereka.

2.3. Support Vector Machine

Support Vector Machine (SVM) adalah salah satu metode dalam pembelajaran mesin (*machine learning*) yang digunakan untuk melakukan klasifikasi dan regresi. SVM termasuk ke dalam kategori *supervised learning*, yang artinya metode ini memerlukan data latih berlabel untuk membuat model prediksi. Metode ini sangat populer dan efektif terutama dalam tugas-tugas klasifikasi, seperti klasifikasi teks, analisis sentimen, pengenalan wajah, dan lainnya.[5], [6]

Pada dasarnya, SVM bekerja dengan mencari *hyperplane* atau garis pemisah terbaik yang dapat membedakan data ke dalam dua kelas atau lebih secara optimal. *Hyperplane* ini adalah batas keputusan yang memisahkan data dalam ruang multi-dimensi. SVM memiliki tujuan untuk menemukan *hyperplane* dengan *margin* terbesar antara data dari kelas yang berbeda, sehingga hasil klasifikasi lebih akurat dan model lebih *general*. Beberapa konsep dasar dalam SVM adalah sebagai berikut:

- Hyperplane*: Garis atau bidang pemisah di antara dua kelas. Dalam data dua dimensi, *hyperplane* berbentuk garis, sementara pada data tiga dimensi, *hyperplane* adalah bidang. Untuk data dengan dimensi lebih tinggi, *hyperplane* tetap ada tetapi sulit divisualisasikan.
- Support Vectors*: Titik-titik data yang berada paling dekat dengan *hyperplane* dan menentukan posisi serta orientasi *hyperplane*. *Support vectors* sangat berpengaruh dalam membentuk garis atau bidang pemisah yang optimal, karena mereka adalah titik-titik kritis yang mempersempit margin antara dua kelas.
- Margin*: Jarak antara *hyperplane* dan data titik terdekat dari masing-masing kelas. SVM berusaha untuk memaksimalkan *margin* ini, sehingga menghasilkan pemisahan yang optimal antara dua kelas. Semakin besar margin, semakin baik generalisasi model.[7]

Berikut adalah langkah-langkah umum untuk menerapkan *Support Vector Machine* (SVM) dalam

sebuah proyek klasifikasi, seperti analisis sentimen atau klasifikasi data lainnya:

1. Pengumpulan dan Persiapan Data

Mengumpulkan data yang akan digunakan, misalnya teks ulasan atau dataset yang sudah berlabel, lalu memastikan data memiliki label (contoh: positif, negatif, atau netral untuk analisis sentimen), kemudian melakukan data cleaning atau pembersihan data, seperti menghilangkan data yang tidak relevan, menangani data yang hilang, dan menormalkan data jika diperlukan.

2. Pra-Pemrosesan Data

- Ekstraksi Fitur:** Dalam kasus data teks, seperti analisis sentimen, ubah data teks menjadi fitur numerik yang dapat diproses oleh SVM.
- Normalisasi Data:** Pada data numerik, dilakukan normalisasi atau penskalaan (misalnya min-max scaling) agar semua fitur berada dalam rentang yang sama. Hal ini penting untuk meningkatkan kinerja SVM.
- Pembagian Data:** Membagi data menjadi data latih (*training set*) dan data uji (*testing set*) agar model dapat dilatih dan diuji secara independen.[8]

3. Pemilihan Kernel dan Parameter SVM

Memilih jenis kernel SVM yang sesuai dengan karakteristik data. Kemudian menentukan parameter C (yang mengontrol trade-off antara margin maksimal dan kesalahan klasifikasi) dan parameter gamma (untuk kernel RBF atau γ). Parameter ini dapat dioptimalkan menggunakan teknik *grid search* atau *cross-validation*.

4. Pelatihan Model SVM

Menggunakan data latih untuk melatih model SVM. Model akan berusaha menemukan *hyperplane* atau garis pemisah optimal yang memisahkan data antar kelas. Dalam proses pelatihan, model SVM akan memilih *support vectors* dan memaksimalkan *margin* antar kelas.

5. Evaluasi Model

Menguji model menggunakan data uji yang sebelumnya disisihkan lalu mengukur performa model menggunakan metrik evaluasi seperti:

- Akurasi:** Persentase prediksi yang benar.
- Precision, Recall, dan F1-Score:** Metrik ini penting untuk mengukur performa pada data yang tidak seimbang (*imbalanced data*).
- Confusion Matrix:** Matriks yang menunjukkan jumlah prediksi benar dan salah untuk setiap kelas, memudahkan analisis performa model.[6], [9]

3. METODE PENELITIAN

Metode pengumpulan data dan metode analisis data merupakan dua pendekatan yang digunakan dalam penelitian ini. Tiga metode digunakan untuk memperoleh data untuk penelitian ini yaitu observasi, studi pustaka, dan mengumpulkan data primer dari ulasan pengguna di situs *web* Google Play Store. Langkah-langkah pemrosesan teks, pemodelan berbasis Lexicon dan SVM, matriks konfusi, dan visualisasi semuanya dimasukkan dalam analisis data untuk penelitian ini.

3.1. Metode Pengumpulan Data

Metode pengumpulan data berikut digunakan untuk mengumpulkan data dan informasi mengenai objek yang diteliti peneliti untuk penelitian ini

3.2. Observasi

Untuk mengumpulkan informasi untuk penelitian awal, peneliti memasang aplikasi JakOne Mobile, melihat ulasan Google Play Store, dan menggunakan aplikasi itu sendiri. Temuan yang peneliti peroleh dari pengamatannya adalah sebagai berikut.

- Mempelajari kendala terkait penggunaan aplikasi JakOne Mobile.
- Mempelajari lebih lanjut kelebihan dan kekurangan aplikasi JakOne Mobile.

3.3. Studi Pustaka

Untuk mengumpulkan informasi dan data yang berkaitan dengan judul penelitian, dilakukan tinjauan literatur. Data dan informasi yang terkumpul kemudian dijadikan sebagai bahan pendukung dan perbandingan bagi penelitian yang sedang berlangsung.

3.4. Scrapping Data

Python Scrapper digunakan untuk mengumpulkan data. Variabel metadata data ulasan mencakup nama pengguna pengguna, konten ulasan, jumlah bintang, serta tanggal dan waktu ulasan. Data dikumpulkan dari *review* di *website* Google Play. Setelah diproses, data diubah ke format *Comma Separated Value* (CSV).

3.5. Preprocessing Data

Preprocessing data adalah tahapan penting dalam analisis sentimen yang bertujuan untuk menyiapkan data teks mentah agar dapat diolah dan dianalisis dengan lebih baik. Data teks mentah seringkali tidak terstruktur dan mengandung banyak *noise*, seperti karakter khusus, tanda baca, kata-kata yang tidak relevan, dan lain sebagainya. *Preprocessing* data membantu membersihkan dan mentransformasi data teks menjadi format yang lebih terstruktur dan relevan untuk analisis sentimen.

3.6. Cleaning

Karakter spesial yang pengguna gunakan pada *review* JakOne Mobile di Google Play seperti angka, hashtag, mention username, URL Website, tag HTML, emoticon, koma, titik spasi berlebih akan dihapus dari data ulasan yang akan masuk ke tahap analisis. *Cleaning* ini bertujuan untuk mengurangi kesalahan acak (*noise*) pada data.

3.7. Tokenisasi

Tokenisasi adalah tahap persiapan teks yang dapat digunakan untuk memecah kalimat, paragraf, atau dokumen menjadi unit-unit yang lebih kecil, seperti token atau kata-kata independen, dan menghilangkan kalimat-kalimat yang identik.

Karena Python adalah bahasa pemrograman peka huruf besar-kecil yang membedakan huruf, semua ulasan akan diubah terlebih dahulu menjadi huruf kecil sebelum dipecah menjadi kata-kata. Hal ini akan menghilangkan perbedaan antara kata seperti "Aplikasi" dan "aplikasi", yang dapat membahayakan keakuratan hasil. Huruf kecil dan huruf kapital, yang

dapat menyebabkan perbedaan antara dua kata yang mempunyai arti yang sama jika tidak sama.

3.8. Normalisasi

Normalisasi adalah tahapan mengubah kata yang tidak baku menjadi kata yang baku sesuai dengan Kamus Besar Bahasa Indonesia.

3.9. Stopword Removal

Stopword adalah kata-kata seperti konjungsi, preposisi, dan sebagainya yang tidak memiliki makna apa pun dalam sebuah kalimat. Ada juga kemungkinan untuk memperoleh banyak *stopword* dalam sebuah frasa. Menghilangkan *stopword* dari kalimat atau paragraf akan memudahkan pemrosesan data dan mencegah kesalahpahaman selama analisis lebih lanjut.

3.10. Stemming

Teknik *preprocessing* teks yang disebut *stemming* digunakan untuk mengidentifikasi kata dasar suatu istilah dengan menghilangkan semua imbuhan dari kata turunan, termasuk prefiks, infiks, sufiks, dan gabungan imbuhan. *Stemming* adalah proses mengubah bentuk kata menjadi kata dasar yang sesuai dengan struktur Bahasa Indonesia yang baik dan benar.

3.11. Pelabelan Data Metode Lexicon Based

Setiap tag kelas data komentar harus memiliki data ulasan yang digunakan untuk melatih *classifier* metode pembelajaran mesin yang terhubung. Terlepas dari apakah ulasannya positif atau negatif, setiap informasi yang telah melewati langkah *preprocessing* kata sekarang akan diberi tag atau label kelas. Label kelas teks dapat ditentukan dengan berbagai cara, seperti dengan menetapkan label kelas secara manual berdasarkan penilaian sendiri, mengevaluasi berdasarkan kuantitas nilai atau peringkat yang diberikan, atau menggunakan Metode *Lexicon Based* untuk kategorisasi teks.

Untuk menentukan apakah ulasan itu positif atau negatif untuk penelitian ini, peneliti akan menggunakan kamus *Lexicon*. Untuk menentukan label kategori data ulasan, peneliti pertama-tama menentukan sentimen yang ada dalam komentar. Kemudian, mereka menggunakan skor sentimen untuk mendapatkan label untuk setiap komentar.

Setiap ulasan sekarang dapat digunakan untuk melatih *classifier* dan berfungsi sebagai referensi untuk menentukan akurasi, presisi, *recall*, dan *f-measure* dari temuan analisis sentimen setelah diberi label sentimen dengan benar menggunakan kamus *Lexicon*. [10]

3.12. Model Algoritma Support Vector Machine

Tahapan berikutnya setelah mendapatkan hasil *Lexicon Based*, dapat dilakukan klasifikasi dengan metode *Support Vector Machine*. SVM adalah metode klasifikasi yang dilakukan dengan cara mencari term atau kondisi serupa dengan parameter tertentu agar dapat menentukan kategori akhirnya.

Untuk mencari persentase tertinggi dalam pengujian akurasi, pada penelitian ini data latih dan data uji dibagi sebesar 80% data latih dan 20% data uji.

Setelah set data latih dan data uji dibuat, maka dilanjutkan ke proses percobaan.

3.13. Confusion Matrix

Keempat jenis hasil kinerja machine learning yaitu akurasi, presisi, *recall*, dan *F-Measure* yang diperoleh dari analisis sentimen menggunakan metode *Support Vector Machine* pada ulasan JakoOne Mobile ditentukan dengan melakukan *Confusion Matrix* setelah prosedur pemodelan *Lexicon* dan *support vector machine* selesai. Untuk menemukan hasil akurasi maksimum, peneliti menghitung temuan analisis sentimen SVM tergantung pada setiap variabel.

3.14. Visualisasi

Untuk menentukan istilah mana yang paling sering ditemukan dalam data ulasan dan seberapa sering istilah tersebut muncul dalam ulasan, kata-kata yang sering muncul dalam ulasan sekarang akan divisualisasikan menjadi *Word Cloud*. Diagram batang yang terdiri dari beberapa karakteristik emosi adalah cara lain visualisasi digunakan untuk menunjukkan kuantitas emosi dalam data ulasan. [11]

4. HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data

Proses pengumpulan data dilakukan dengan menggunakan Bahasa *Python* dan *library python scrapper* yang dijalankan di Google Collaboratory. Berikut adalah tahapan-tahapan pengambilan data ulasan:

Install Library Google Play Scrapper kemudian lakukan impor *reviews* ke dalam *Notebook Google Collaboratory*.

```
[ ] !pip install google-play-scraper
Collecting google-play-scraper
  Downloading google_play_scraper-1.2.7-py3-none-any.whl.metadata (50 kB)
    -----
    google_play_scraper-1.2.7-py3-none-any.whl (28 kB)
Installing collected packages: google-play-scraper
Successfully installed google-play-scraper-1.2.7

from google_play_scraper import app, reviews
import pandas as pd
import datetime

google_play_scraper: Library ini digunakan untuk mengambil informasi aplikasi dan ulasan dari Google Play Store. pandas: Digunakan untuk bekerja dengan data dalam format DataFrame. datetime: Digunakan untuk menentukan rentang waktu (start_date dan end_date).
```

Gambar 4. 1 *Install Library Google Play Scrapper*

Kemudian *Scraping* sesuai dengan app id Google Play Store dari JakoOne Mobile.

```
from google_play_scraper import reviews, sort

app_id = 'com.dev.jakone.nbanking'

def get_reviews(app_id, lang='id', count=5000, sort=sort.NEHEST, filter_score_with=None, filter_device_with=None, continuation_token=None):
    try:
        result, continuation_token = reviews(
            app_id,
            lang=lang,
            count=count,
            sort=sort,
            filter_score_with=filter_score_with,
            filter_device_with=filter_device_with,
            continuation_token=continuation_token
        )
        return result, continuation_token
    except Exception as e:
        print('Error: ', e)
        return None, None

reviews, continuation_token = get_reviews(app_id)

if reviews is not None:
    print('Jumlah ulasan: ', len(reviews))
    if len(reviews) > 0:
        print('Contoh ulasan:')
        print(reviews[0])
    else:
        print('Tidak dapat mengambil ulasan.')

Jumlah ulasan: 5000
```



```

u"\u0001FA00-\u0001FAGF" # Chess Symbols
u"\u0001A70-\u0001FAFF" # Symbols and Pictographs Extended-A
u"\u0001F00-\u0001FAFF" # Additional Emoticons
u"\u0001F10-\u0001F1FF" # Flags
return emoji_pattern.sub(r'', tweet)
else:
return tweet

# Fungsi untuk menghapus simbol
def remove_symbols(tweet):
if tweet is not None and isinstance(tweet, str):
tweet = re.sub(r'[\p{S}]+', '', tweet)
return tweet

# Fungsi untuk menghapus angka
def remove_numbers(tweet):
if tweet is not None and isinstance(tweet, str):
tweet = re.sub(r'\d+', '', tweet)
return tweet

df['cleaning'] = df['Review Text'].apply(lambda x: remove_urls(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_html(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_emoji(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_symbols(x))
df['cleaning'] = df['cleaning'].apply(lambda x: remove_numbers(x))
df.head(10)

```

Gambar 4. 8 Cleaning

4. Proses Case Folding

Pada proses ini huruf kapital pada data ulasan diganti menjadi huruf kecil (lowercase)

```

[] def case_folding(text):
if isinstance(text, str):
lowercase_text = text.lower()
return lowercase_text
else:
return text

df['case_folding'] = df['cleaning'].apply(case_folding)
df.head(10)

```

#	date	username	rating	Review Text	Cleaning	case_folding
0	2025-01-13 07:19:41	Peganga Google	5	Aplikasi jekone mobile sangat teren bagus kare...	Aplikasi jekone mobile sangat teren bagus kare...	aplikasi jekone mobile sangat teren bagus kare...
1	2025-01-13 06:24:20	Peganga Google	5	memudahkan transaksi kemanapun tanpa rpegly la...	memudahkan transaksi kemanapun tanpa rpegly la...	memudahkan transaksi kemanapun tanpa rpegly la...
2	2025-01-13 02:36:56	Peganga Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...
3	2025-01-13 02:27:30	Peganga Google	5	Aplikasinya bagus banget flynya memudahkan u...	Aplikasinya bagus banget flynya memudahkan u...	aplikasinya bagus banget flynya memudahkan u...
4	2025-01-13 00:28:38	Peganga Google	5	Aplikasinya bagus, flynya lengkap dan mudah d...	Aplikasinya bagus flynya lengkap dan mudah d...	aplikasinya bagus flynya lengkap dan mudah d...
5	2025-01-13 00:23:31	Peganga Google	5	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...
6	2025-01-12 08:01:24	Peganga Google	5	Transaksi disini lebih gampang, flynya juga...	Transaksi disini lebih gampang flynya juga...	transaksi disini lebih gampang flynya juga...
7	2025-01-12 05:58:56	Peganga Google	5	Aplikasinya praktis dan mudah dipkanan juga b...	Aplikasinya praktis dan mudah dipkanan juga b...	aplikasinya praktis dan mudah dipkanan juga b...
8	2025-01-12 05:23:10	Peganga Google	5	aplikasi nya bagus buat transaksi dan gampang...	aplikasi nya bagus buat transaksi dan gampang...	aplikasi nya bagus buat transaksi dan gampang...

Gambar 4. 9 Proses Case Folding

5. Normalisasi kata

Normalisasi kata adalah mengganti kata tidak baku pada ulasan menjadi kata yang baku sesuai kamus besar Bahasa Indonesia, pada tahap ini peneliti mengambil dataset dari kaggle.com untuk dataset kamus Bahasa Indonesia.

```

[] import pandas as pd

# Fungsi penggantian kata tidak baku
def replace_tidak_baku(text, kamus_tidak_baku):
if isinstance(text, str):
words = text.split()
replaced_words = []
kalimat_baku = []
kata_diganti = []
kata_tidak_baku_hash = []

for word in words:
if word in kamus_tidak_baku:
baku_word = kamus_tidak_baku[word]
if isinstance(baku_word, str) and all(char.isalpha()) for char in baku_word:
replaced_words.append(baku_word)
kalimat_baku.append(baku_word)
kata_diganti.append(word)
kata_tidak_baku_hash.append(hash(word))
else:
replaced_words.append("karakter eksplisit")
kalimat_baku.append(word)

replaced_words.append(word)
replaced_text = ' '.join(replaced_words)
else:
replaced_text = ''
kalimat_baku = []
kata_diganti = []
kata_tidak_baku_hash = []

```

Gambar 4. 10 Normalisasi kata

```

[] # Data Dataset
data = pd.DataFrame(df[['date', 'username', 'rating', 'Review Text', 'cleaning', 'case_folding']])
data.head(10)

```

#	date	username	rating	Review Text	Cleaning	case_folding
0	2025-01-13 07:19:41	Peganga Google	5	Aplikasi jekone mobile sangat teren bagus kare...	Aplikasi jekone mobile sangat teren bagus kare...	aplikasi jekone mobile sangat teren bagus kare...
1	2025-01-13 06:24:20	Peganga Google	5	memudahkan transaksi kemanapun tanpa rpegly la...	memudahkan transaksi kemanapun tanpa rpegly la...	memudahkan transaksi kemanapun tanpa rpegly la...
2	2025-01-13 02:36:56	Peganga Google	5	Aplikasi ini membantu banget dengan tampilan y...	Aplikasi ini membantu banget dengan tampilan y...	aplikasi ini membantu banget dengan tampilan y...
3	2025-01-13 02:27:30	Peganga Google	5	Aplikasinya bagus banget flynya memudahkan u...	Aplikasinya bagus banget flynya memudahkan u...	aplikasinya bagus banget flynya memudahkan u...
4	2025-01-13 00:28:38	Peganga Google	5	Aplikasinya bagus, flynya lengkap dan mudah d...	Aplikasinya bagus flynya lengkap dan mudah d...	aplikasinya bagus flynya lengkap dan mudah d...
5	2025-01-13 00:23:31	Peganga Google	5	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...
6	2025-01-12 08:01:24	Peganga Google	5	Transaksi disini lebih gampang, flynya juga...	Transaksi disini lebih gampang flynya juga...	transaksi disini lebih gampang flynya juga...
7	2025-01-12 05:58:56	Peganga Google	5	Aplikasinya praktis dan mudah dipkanan juga b...	Aplikasinya praktis dan mudah dipkanan juga b...	aplikasinya praktis dan mudah dipkanan juga b...
8	2025-01-12 05:23:10	Peganga Google	5	aplikasi nya bagus buat transaksi dan gampang...	aplikasi nya bagus buat transaksi dan gampang...	aplikasi nya bagus buat transaksi dan gampang...
9	2025-01-12 01:59:50	Peganga Google	5	Aplikasi nya jelas dan sangat recomendedee bu...	Aplikasi nya jelas dan sangat recomendedee bu...	aplikasi nya jelas dan sangat recomendedee bu...

Gambar 4. 11 Kata-kata yang tidak baku sudah berubah

6. Tokenisasi

Tokenisasi bertujuan untuk memecah teks pada ulasan menjadi potongan-potongan kata, dan potongan kata tersebut disebut token ke dalam bentuk token.

```

[] def tokenize(text):
tokens = text.split()
return tokens

df['tokenize'] = df['normalisasi'].apply(tokenize)
df.head(10)

```

Gambar 4. 12 Tokenisasi

7. Stopword Removal

Stopword removal berfungsi menghilangkan kata yang tidak memiliki makna dalam ulasan.

```

[] from nltk.corpus import stopwords
nltk.download('stopwords')
stop_words = stopwords.words('Indonesian')

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

def remove_stopwords(text):
return [word for word in text if word not in stop_words]

df['stopword_removal'] = df['tokenize'].apply(lambda x: remove_stopwords(x))
df.head(10)

```

#	date	username	rating	Review Text	Cleaning	case_folding	normalisasi	stopword_removal
0	2025-01-13 07:19:41	Peganga Google	5	Aplikasi jekone mobile sangat teren bagus kare...				
1	2025-01-13 06:24:20	Peganga Google	5	memudahkan transaksi kemanapun tanpa rpegly la...				
2	2025-01-13 02:36:56	Peganga Google	5	Aplikasi ini membantu banget dengan tampilan y...				
3	2025-01-13 02:27:30	Peganga Google	5	Aplikasinya bagus banget flynya memudahkan u...				
4	2025-01-13 00:28:38	Peganga Google	5	Aplikasinya bagus, flynya lengkap dan mudah d...	Aplikasinya bagus flynya lengkap dan mudah d...	aplikasinya bagus flynya lengkap dan mudah d...	aplikasinya bagus flynya lengkap dan mudah d...	aplikasinya bagus flynya lengkap dan mudah d...
5	2025-01-13 00:23:31	Peganga Google	5	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...	aplikasi ini rekomenadae lgt buat topu game m...
6	2025-01-12 08:01:24	Peganga Google	5	Transaksi disini lebih gampang, flynya juga...	Transaksi disini lebih gampang flynya juga...	transaksi disini lebih gampang flynya juga...	transaksi disini lebih gampang flynya juga...	transaksi disini lebih gampang flynya juga...
7	2025-01-12 05:58:56	Peganga Google	5	Aplikasinya praktis dan mudah dipkanan juga b...				
8	2025-01-12 05:23:10	Peganga Google	5	aplikasi nya bagus buat transaksi dan gampang...				
9	2025-01-12 01:59:50	Peganga Google	5	Aplikasi nya jelas dan sangat recomendedee bu...				

Gambar 4. 13 Stopword Removal

8. Stemming Data

Stemming data bertujuan mengganti kata dalam ulasan ke dalam kata dasar sesuai kaidah Bahasa Indonesia. Pada tahap ini semua kata dalam ulasan telah ternormalisasi dan data yang dibutuhkan sudah bersih, sehingga proses selanjutnya sudah bisa dilanjutkan.

```

[] !pip install Sastrawi

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
from nltk.stem import PorterStemmer
from nltk.stem.snowball import SnowballStemmer

Requirement already satisfied: sastrawi in /usr/local/lib/python3.10/dist-packages (3.0.0)

[] factory = StemmerFactory()
stemmer = factory.create_stemmer()

def stem_text(text):
return [stemmer.stem(word) for word in text]

df['stemming_data'] = df['stopword_removal'].apply(lambda x: ' '.join(stem_text(x)))
df.head(10)

```

Gambar 4. 14 Stemming data

```

[] date username rating Review Text Cleaning case_folding normalisasi stemming_data stopword_removal stemming_data
0 2025-01-13 07:19:41 Peganga Google 5 Aplikasi jekone mobile sangat teren bagus kare... Aplikasi jekone mobile sangat teren bagus kare... aplikasi jekone mobile sangat teren bagus kare...
1 2025-01-13 06:24:20 Peganga Google 5 memudahkan transaksi kemanapun tanpa rpegly la... memudahkan transaksi kemanapun tanpa rpegly la... memudahkan transaksi kemanapun tanpa rpegly la...
2 2025-01-13 02:36:56 Peganga Google 5 Aplikasi ini membantu banget dengan tampilan y... Aplikasi ini membantu banget dengan tampilan y... aplikasi ini membantu banget dengan tampilan y...
3 2025-01-13 02:27:30 Peganga Google 5 Aplikasinya bagus banget flynya memudahkan u... Aplikasinya bagus banget flynya memudahkan u... aplikasinya bagus banget flynya memudahkan u...
4 2025-01-13 00:28:38 Peganga Google 5 Aplikasinya bagus, flynya lengkap dan mudah d... Aplikasinya bagus flynya lengkap dan mudah d... aplikasinya bagus flynya lengkap dan mudah d...
5 2025-01-13 00:23:31 Peganga Google 5 aplikasi ini rekomenadae lgt buat topu game m... aplikasi ini rekomenadae lgt buat topu game m... aplikasi ini rekomenadae lgt buat topu game m...
6 2025-01-12 08:01:24 Peganga Google 5 Transaksi disini lebih gampang, flynya juga... Transaksi disini lebih gampang flynya juga... transaksi disini lebih gampang flynya juga...
7 2025-01-12 05:58:56 Peganga Google 5 Aplikasinya praktis dan mudah dipkanan juga b... Aplikasinya praktis dan mudah dipkanan juga b... aplikasinya praktis dan mudah dipkanan juga b...
8 2025-01-12 05:23:10 Peganga Google 5 aplikasi nya bagus buat transaksi dan gampang... aplikasi nya bagus buat transaksi dan gampang... aplikasi nya bagus buat transaksi dan gampang...
9 2025-01-12 01:59:50 Peganga Google 5 Aplikasi nya jelas dan sangat recomendedee bu... Aplikasi nya jelas dan sangat recomendedee bu... aplikasi nya jelas dan sangat recomendedee bu...

```

Gambar 4. 15 Hasil stemming data

9. Word Cloud Setelah Preprocessing

Menampilkan Wordcloud setelah preprocessing


```
[ ] from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

[ ] # Mendefinisikan vectorizer
vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

[ ] # Menampilkan hasil vektorisasi
print("Matriks Vektorisasi untuk Data Latih:")
print(X_train_vectorized.toarray())

# menampilkan sebagian kecil matriks
print("\nSebagian kecil Matriks Vektorisasi untuk Data Latih:")
print(X_train_vectorized[15, :].toarray())

Matriks Vektorisasi untuk Data Latih:
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

[ ] Sebagian kecil Matriks Vektorisasi untuk Data Latih:
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

[ ] # Membuat dan melatih model SVM
svm = SVC(kernel='linear')
svm.fit(X_train_vectorized, y_train)

SVC
SVC(kernel='linear')

[ ] # Evaluasi model
y_pred_svm = svm.predict(X_test_vectorized)

[ ] # Menampilkan confusion matrix dalam angka
cm_svm = confusion_matrix(y_test, y_pred_svm)
print("SVM Confusion Matrix:")
print(cm_svm)

SVM Confusion Matrix:
[[288 29 3]
 [30 215 25]
 [ 2 27 297]]

[ ] # Fungsi untuk plot confusion matrix
def plot_confusion_matrix(model_name, y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)

    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
                xticklabels=['positif', 'negatif', 'netral'],
                yticklabels=['positif', 'negatif', 'netral'])

    plt.title(f"{model_name} Confusion Matrix")
    plt.xlabel("Predicted Label")
    plt.ylabel("True Label")
    plt.show()

plot_confusion_matrix("SVM", y_test, y_pred_svm)
```

Gambar 4. 18 Support Vector Machine dan Confusion Matrix

Gambar 4. 19 Hasil Confusion Matrix

Dari hasil *confusion matrix* dapat dijelaskan bahwa terdapat 288 data positif yang diprediksi termasuk ke dalam sentimen positif, 30 data positif yang diprediksi termasuk ke dalam sentimen negatif, dan 2 data positif yang termasuk ke dalam sentimen netral. Selain itu terdapat 215 data negatif yang diprediksi termasuk ke

dalam sentimen negatif, 29 data negatif yang diprediksi termasuk kedalam sentiment positif, dan 27 data negatif yang termasuk kedalam sentimen netral. Kemudian terdapat 297 data netral yang termasuk ke dalam sentimen netral, 25 data netral yang termasuk ke dalam sentimen negatif, dan 3 data netral yang termasuk kedalam sentimen positif.

```
[ ] # Menampilkan accuracy untuk SVM
accuracy_svm = accuracy_score(y_test, y_pred_svm)
print("SVM Accuracy:", accuracy_svm)
print("")

accuracy_svm_percentage = accuracy_svm * 100
print("SVM Accuracy:", "{:.2f}%".format(accuracy_svm_percentage))
print("")

# Menampilkan classification report untuk SVM
print("SVM Classification Report:")
print(classification_report(y_test, y_pred_svm))

SVM Accuracy: 0.8733624454148472

SVM Accuracy: 87.34%

SVM Classification Report:
precision recall f1-score support
Negatif 0.90 0.90 0.90 320
Netral 0.79 0.80 0.79 270
Positif 0.91 0.91 0.91 326

accuracy 0.87 916
macro avg 0.87 0.87 0.87 916
weighted avg 0.87 0.87 0.87 916
```

Gambar 4. 20 Nilai akurasi Support Vector Machine

Nilai akurasi *Support Vector Machine* menunjukkan angka 0.8733624454148472, angka ini berarti model berhasil memprediksi sentiment dengan benar sebanyak 87,34% dari total data.

Laporan klasifikasi ini memberikan informasi lebih detail tentang kinerja model untuk setiap kategori sentimen. Berikut penjelasannya:

1. *Precision* (Presisi): Presisi mengukur seberapa sering model benar dalam memprediksi suatu kategori sentimen dari semua data yang diprediksi memiliki sentimen tersebut.
 - a. Negatif: 0.90 (90% dari data yang diprediksi negatif benar-benar negatif)
 - b. Netral: 0.79 (79% dari data yang diprediksi netral benar-benar netral)
 - c. Positif: 0.91 (91% dari data yang diprediksi positif benar-benar positif)
2. *Recall*: *Recall* mengukur seberapa sering model benar dalam memprediksi suatu kategori sentimen dari semua data yang sebenarnya memiliki sentimen tersebut.
 - a. Negatif: 0.90 (90% dari data negatif yang sebenarnya berhasil diprediksi sebagai negatif)
 - b. Netral: 0.80 (80% dari data netral yang sebenarnya berhasil diprediksi sebagai netral)
 - c. Positif: 0.91 (91% dari data positif yang sebenarnya berhasil diprediksi sebagai positif)
3. *F1-Score*: *F1-score* adalah rata-rata harmonik antara presisi dan *recall*. Ini memberikan ukuran keseimbangan antara presisi dan *recall*.
 - a. Negatif: 0.90
 - b. Netral: 0.79
 - c. Positif: 0.91
4. *Support* (Jumlah Data): Jumlah data aktual untuk setiap kategori sentimen.
 - a. Negatif: 320
 - b. Netral: 270
 - c. Positif: 326

5. KESIMPULAN DAN SARAN

Kesimpulan akhir dari penelitian ini adalah bahwa pendekatan analisis sentimen berbasis machine learning dapat digunakan untuk mengolah data ulasan secara lebih akurat dibandingkan metode berbasis rating bintang saja. Implementasi model ini diharapkan dapat membantu perusahaan dalam mengambil keputusan strategis berdasarkan sentimen pengguna terhadap produk atau layanan mereka. Penelitian lebih lanjut juga dapat memperluas cakupan analisis dengan mengeksplorasi faktor-faktor yang mempengaruhi sentimen pengguna terhadap aplikasi fintech, seperti aspek keamanan, kecepatan transaksi, pengalaman pengguna (UX/UI), serta kualitas layanan pelanggan. Studi ini akan memberikan wawasan yang lebih mendalam mengenai aspek yang paling berpengaruh terhadap kepuasan pengguna dan bagaimana perusahaan dapat meningkatkan daya saing aplikasinya di pasar fintech. Selain itu, penelitian di masa depan dapat mempertimbangkan analisis penerapan sentimen berbasis multimodal, yang tidak hanya mengandalkan teks tetapi juga memperhitungkan elemen lain seperti ekspresi wajah dalam ulasan video, nada suara dalam rekaman, atau pola interaksi pengguna dengan aplikasi.

DAFTAR PUSTAKA

- [1] F. Z. Junaedy *et al.*, “KOMPARASI ALGORITMA SUPPORT VECTOR MACHINE DAN NAÏVE BAYES PADA ANALISIS SENTIMEN FORMULA-E JAKARTA TAHUN 2022.” [Online]. Available: <https://youtu.be/mhAEiHuVFxY>
- [2] R. C. Rivaldi, T. D. Wismarini, J. T. Lomba, and J. Semarang, “Analisis Sentimen Pada Ulasan Produk Dengan Metode Natural Language Processing (NLP) (Studi Kasus Zalika Store 88 Shopee),” vol. 17, no. 1, pp. 120–128, 2024, doi: 10.51903/elkom.v17i1.1680.
- [3] O. Manullang, C. Prianto, and N. H. Harani, “Analisis Sentimen Untuk Memprediksi Hasil Calon Pemilu Presiden Menggunakan Lexicon Based dan Random Forest.”
- [4] S. Alpin Rizaldi, S. Alam, and I. Kurniawan, “ANALISIS SENTIMEN PENGGUNA APLIKASI JMO (JAMSOSTEK MOBILE) PADA GOOGLE PLAY STORE MENGGUNAKAN METODE NAIVE BAYES 1),” vol. 2, no. 3, pp. 109–117, 2023, doi: 10.55123.
- [5] S. I. Nurhafida and F. Sembiring, “Analisis Sentimen Aplikasi Novel Online Di Google Play Store Menggunakan Algoritma Support Vector Machine (SVM),” 2022.
- [6] M. Khoirul, U. Hayati, and O. Nurdiawan, “ANALISIS SENTIMEN APLIKASI BRIMO PADA ULASAN PENGGUNA DI GOOGLE PLAY MENGGUNAKAN ALGORITMA NAIVE BAYES,” 2023.
- [7] E. R. Kaburuan and N. R. Setiawan, “Sentimen Analisis Review Aplikasi Digital Korlantas Pada Google Play Store Menggunakan Metode SVM,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 12, no. 1, pp. 105–116, Mar. 2023, doi: 10.32736/sisfokom.v12i1.1614.
- [8] T. Elizabeth, “Analisis Sentimen Ulasan Aplikasi PrimaKu Menggunakan Metode Support Vector Machine,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 4, 2022, [Online]. Available: <http://jurnal.mdp.ac.id>
- [9] R. Wahyudi *et al.*, “Analisis Sentimen pada review Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine,” *JURNAL INFORMATIKA*, vol. 8, no. 2, 2021, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [10] “ANALISIS SENTIMEN TERHADAP KOMENTAR SARKASME MENGGUNAKAN ALGORITMA SUPPORT VECTOR MACHINE (SVM).”
- [11] D. Untuk, M. Persyaratan, M. Gelar, S. Komputer, D. Oleh, and A. Fatihin, “ANALISIS SENTIMEN TERHADAP ULASAN APLIKASI MOBILE MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (SVM) DAN PENDEKATAN LEXICON BASED.”