



Perbandingan Model K-Nearest Neighbor dan Support Vector Machine untuk Klasifikasi Sentimen Pengguna Berdasarkan Ulasan Aplikasi GOPAY

Moh Adi Kurniawan¹, Gutti Zaidan Syauqi², Mohammad Fajar Sirullah³, Muhammad Arifin⁴

^{1, 2, 3, 4} Fakultas Teknik, Program Studi Sistem Informasi, Universitas Muria Kudus, Kudus, Indonesia

Email: ¹kurniawanmohadi@gmail.com, ²gutizidan@gmail.com, ³mohammadfajar.xtp1@gmail.com,
⁴arifin.m@umk.ac.id

(* : kurniawanmohadi@gmail.com)

Abstrak—Aplikasi digital seperti GOPAY memungkinkan pengguna memberikan ulasan yang berisi pengalaman dan penilaian terhadap layanan. Ulasan tersebut dapat digunakan untuk mengevaluasi kepuasan dan sentimen pengguna terhadap aplikasi. Tujuan dari penelitian ini adalah untuk melakukan perbandingan antara dua algoritma klasifikasi, yaitu K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM), dalam melakukan analisis sentimen pengguna berdasarkan ulasan aplikasi yang tersedia di Google Play Store. Data yang digunakan diperoleh dari platform Kaggle, kemudian diproses melalui tahap preprocessing teks serta ekstraksi fitur dengan metode TF-IDF. Kinerja model dievaluasi menggunakan metrik seperti akurasi, presisi, recall, dan F1-score. Berdasarkan hasil evaluasi, algoritma SVM menunjukkan performa terbaik dengan akurasi mencapai 86%, sedangkan KNN hanya mencapai 79%. Dengan demikian, metode SVM lebih direkomendasikan dalam klasifikasi sentimen berbasis teks.

Kata Kunci: K-Nearest Neighbors, Support Vector Machine, analisis sentimen, GOPAY, klasifikasi, pemrosesan bahasa alami.

Abstract—Digital applications such as GOPAY allow users to leave reviews that contain experiences and assessments of the service. These reviews can be used to evaluate user satisfaction and sentiment towards the application. The purpose of this study is to conduct a comparison between two classification algorithms, namely K-Nearest Neighbor (KNN) and Support Vector Machine (SVM), in conducting user sentiment analysis based on application reviews available on the Google Play Store. The data used was obtained from the Kaggle platform, then processed through text preprocessing and feature extraction using the TF-IDF method. Model performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. Based on the evaluation results, the SVM algorithm showed the best performance with an accuracy of 86%, while KNN only reached 79%. Thus, the SVM method is more recommended in text-based sentiment classification.

Keywords: K-Nearest Neighbors, Support Vector Machine, sentiment analysis, GOPAY, classification, natural language processing.

1. PENDAHULUAN

Kemajuan dalam teknologi informasi telah menjadi pendorong utama digitalisasi di berbagai bidang, salah satunya sektor keuangan. Di Indonesia, aplikasi dompet digital seperti GOPAY telah menjadi bagian penting dalam memfasilitasi transaksi harian, mulai dari pembayaran transportasi, belanja online, hingga pengiriman uang. Kemudahan yang ditawarkan membuat GOPAY diminati masyarakat dan memiliki jumlah pengguna yang sangat besar.

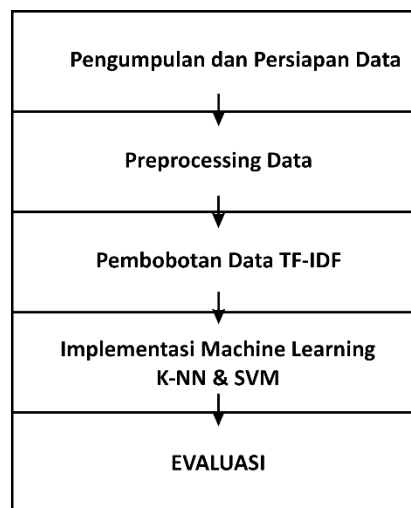
Seiring pertumbuhan pengguna, ulasan terhadap GOPAY di platform seperti Google Play Store juga meningkat pesat. Ulasan ini berisi informasi penting mengenai pengalaman pengguna, baik berupa kepuasan, keluhan, maupun saran. Data ini sangat berharga bagi pengembang untuk evaluasi layanan dan bagi calon pengguna untuk mendapatkan gambaran awal tentang aplikasi (Muhammadin & Sobari, 2021).

Namun, jumlah ulasan yang besar dan bentuknya yang tidak terstruktur membuat analisis manual menjadi tidak efektif. Oleh karena itu, diperlukan pendekatan otomatis menggunakan *machine learning* untuk mengklasifikasikan sentimen pengguna (Wahyudi & Kusumawardana, 2021). Analisis sentimen menjadi solusi utama dalam mengidentifikasi opini dalam teks dan mengkategorikannya ke dalam sentimen positif, negatif, atau netral.

Penelitian ini membandingkan performa dua algoritma *machine learning*, yaitu *Support Vector Machine (SVM)* dan *K-Nearest Neighbor (KNN)*, dalam mengklasifikasikan sentimen ulasan pengguna GOPAY. SVM dikenal dengan kemampuannya memisahkan kelas secara optimal, sementara KNN menilai kedekatan data berdasarkan jarak. Untuk menilai efektivitas masing-masing model, dilakukan evaluasi melalui metrik akurasi, presisi, recall, dan f1-score dalam konteks penelitian ini.

2. METODE

Penelitian ini dilakukan melalui langkah-langkah berikut.



Gambar 1. Alur Penelitian

Penelitian ini didasarkan pada sejumlah tahapan dan konsep teoritis yang dirinci di bawah ini:

2.1 Data Mining

Data mining adalah kegiatan mengekstraksi informasi bernilai dari data, yang dilakukan menggunakan pendekatan seperti *artificial intelligence*, statistika, *machine learning*, dan teknik matematika lainnya. Teknik-teknik ini digunakan untuk mengidentifikasi pola atau wawasan tersembunyi dari kumpulan data berskala besar (Zahra et al., 2022). Data mining adalah cabang ilmu yang berkembang dari kecerdasan buatan dan rekayasa pengetahuan, dengan dasar utama pada pembelajaran mesin dan statistika, serta telah diterapkan di berbagai bidang seperti biologi, lingkungan, dan keuangan.

2.2. Analisis Sentimen

Analisis sentimen adalah proses untuk mendeteksi serta mengklasifikasikan opini atau perasaan dalam suatu teks ke dalam kategori positif, negatif, atau netral. Teknik ini sering digunakan untuk memahami persepsi publik terhadap isu, layanan, atau produk berdasarkan data teks yang diperoleh dari berbagai sumber seperti media sosial atau ulasan daring. Analisis sentimen memiliki tahapan dalam pengolahan datanya yaitu:

- 1) *Pengumpulan Data*, Pada tahap awal data teks diambil dari dataset yang sudah tersedia pada situs Kaggle. Data ini menjadi dasar untuk analisis lebih lanjut.
- 2) *Preprocessing Data*, Proses ini bertujuan untuk membersihkan dan mempersiapkan data agar siap untuk dianalisis. Langkah-langkahnya meliputi penghapusan karakter khusus, normalisasi teks, tokenisasi, penghapusan kata-kata umum (*stopwords*), dan stemming untuk mengembalikan kata ke bentuk dasarnya.

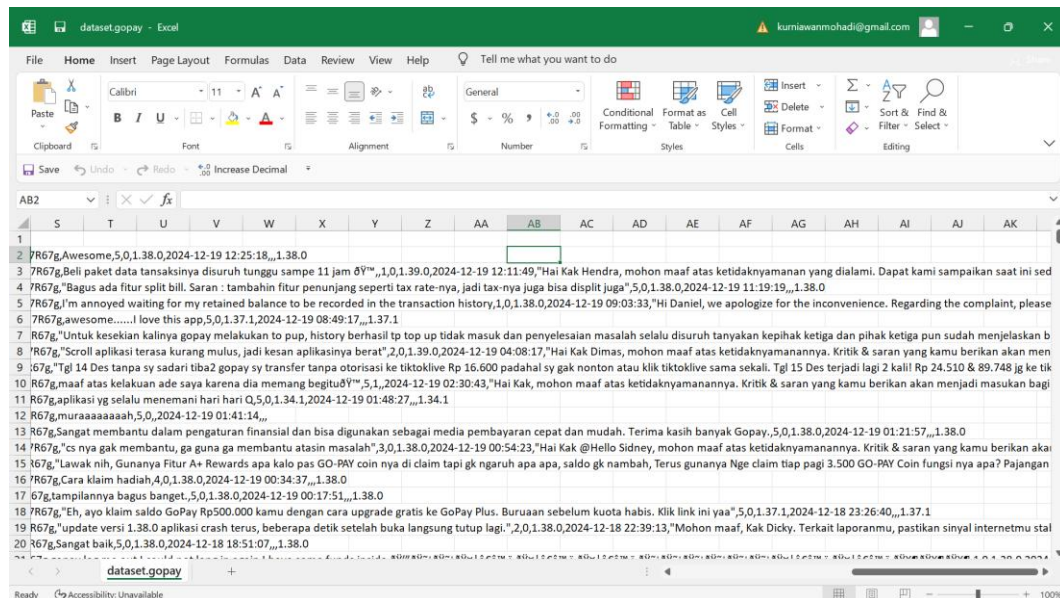
- 3) Klasifikasi Sentimen, Pada tahap ini, data yang telah diproses diklasifikasikan ke dalam kategori sentimen (positif, negatif, atau netral) menggunakan algoritma pembelajaran mesin seperti *Naive Bayes*, *Support Vector Machine (SVM)*, atau *K-Nearest Neighbor (KNN)*.
- 4) Evaluasi Performa Model, Langkah akhir adalah mengevaluasi kinerja model klasifikasi dengan menggunakan metrik seperti akurasi, presisi, recall, dan F1-score guna memastikan bahwa hasil analisis yang diperoleh dapat diandalkan.

2.3. Ulasan Pengguna

Ulasan atau *review* adalah bentuk tulisan atau pernyataan yang memuat penilaian serta tanggapan terhadap suatu karya atau hasil pekerjaan seseorang (Chohan et al., 2020). *Review* tersebut berperan sebagai sumber informasi yang dapat membantu dalam upaya peningkatan mutu dan penyempurnaan di masa depan.

2.4. Dataset

Dataset ulasan pengguna yang digunakan dalam penelitian ini diperoleh dari situs berbagi data publik Kaggle (www.kaggle.com).



Gambar 2. Dataset Ulasan Gopay

Dataset yang diambil memuat kumpulan ulasan teks dari pengguna aplikasi dompet digital, yang mencakup berbagai ekspresi sentimen seperti positif, negatif, dan netral. Dataset ini digunakan sebagai data latih dan uji dalam proses klasifikasi sentimen berbasis machine learning. Pemanfaatan data dari Kaggle memungkinkan peneliti untuk memperoleh sumber data yang terstruktur, relevan, dan telah digunakan secara luas dalam penelitian sejenis, sehingga mendukung validitas eksperimen yang dilakukan.

2.5. Preprocessing

Preprocessing data merupakan tahap awal yang penting dalam proses pemodelan, di mana data dipersiapkan melalui berbagai teknik pembersihan dan transformasi agar layak digunakan pada tahap analisis berikutnya (Alghifari & Juardi, 2021). Tahap ini bertujuan untuk memastikan bahwa data dalam kondisi bersih, konsisten, dan sesuai format yang dibutuhkan oleh algoritma machine learning.

Beberapa tahapan dalam *preprocessing* antara lain:



- 1) *Case Folding*, yaitu mengonversi seluruh huruf dalam teks menjadi huruf kecil untuk menyamakan format kata.
- 2) *Cleaning*, yang mencakup penghapusan elemen non-teks seperti angka, tanda baca, emoji, serta karakter khusus lainnya.
- 3) *Tokenization*, proses memisahkan kalimat atau paragraf menjadi satuan kata (token) individual.
- 4) *Normalization*, yaitu penyesuaian kata-kata tidak baku atau salah eja agar sesuai dengan ejaan yang disempurnakan (EYD).
- 5) *Stemming*, yakni mereduksi kata berimbuhan menjadi bentuk dasar untuk menyederhanakan representasi kata dalam analisis.

2.6. Term Frequency Inverse Document Frequency

Metode TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculannya dalam satu dokumen yang disesuaikan dengan seberapa jarang kata tersebut muncul di keseluruhan koleksi dokumen. Teknik ini diaplikasikan setelah proses *preprocessing* seperti tokenisasi, stemming, dan perhitungan frekuensi kata. TF-IDF menilai seberapa sering suatu kata muncul dalam dokumen tertentu (*term frequency*) dan menyeimbangkannya dengan frekuensi kemunculan kata tersebut di seluruh dokumen (*inverse document frequency*), sehingga kata-kata umum yang sering muncul secara global diberi bobot lebih rendah, sementara kata-kata yang lebih spesifik terhadap dokumen diberi bobot lebih tinggi.

2.7. K-Nearest Neighbor

K-Nearest Neighbor (KNN) merupakan salah satu metode klasifikasi yang menentukan kelas suatu objek berdasarkan kedekatannya dengan sejumlah data tetangga terdekat (*neighbor*) dalam ruang fitur. Kedekatan antar objek umumnya dihitung menggunakan rumus jarak Euclidean. Proses klasifikasi pada KNN dilakukan dengan mengidentifikasi sejumlah K data terdekat, kemudian menetapkan kelas mayoritas dari data tersebut sebagai prediksi kelas objek yang diuji (Windy Mardiyah et al., 2024). Metode ini memiliki keunggulan dalam kemampuannya menggeneralisasi data baru berdasarkan pola yang terbentuk dari data pelatihan. Namun, pemilihan nilai K yang optimal tidak dapat ditentukan secara matematis dan harus diperoleh melalui proses eksperimental, yaitu dengan mencoba beberapa nilai K hingga ditemukan hasil klasifikasi dengan akurasi terbaik.

2.8. Support Vector Machine

Support Vector Machine (SVM) adalah sebuah metode yang dapat diterapkan untuk menyelesaikan permasalahan klasifikasi maupun regresi, dengan prinsip utama membedakan dua kelas data melalui pencarian *hyperplane* optimal (Idris et al., 2023). *Hyperplane* tersebut berfungsi sebagai batas pemisah antara kelas-kelas, dan SVM bertujuan untuk memilih *hyperplane* yang memiliki margin maksimum, yaitu jarak terjauh antara *hyperplane* dan data terdekat dari masing-masing kelas. Data yang berada paling dekat dengan *hyperplane* dikenal sebagai *support vector*, dan peranannya sangat penting dalam menentukan posisi dan arah dari *hyperplane*. Dengan pendekatan ini, SVM mampu menghasilkan pemisahan kelas yang optimal, bahkan dalam ruang fitur berdimensi tinggi, sehingga banyak digunakan dalam pemodelan klasifikasi yang kompleks.

2.9. Confusion Matrix (Evaluasi)

Salah satu metode evaluasi yang dipakai untuk menilai tingkat akurasi dari teknik klasifikasi dalam memprediksi label data adalah *confusion matrix*. Pendekatan ini bekerja dengan cara membandingkan label aktual dari data dengan hasil prediksi yang diberikan oleh model (Romadloni et al., 2022). Berdasarkan perhitungan yang dilakukan pada *confusion matrix*, Metrik evaluasi seperti *akurasi*, *presisi*, *recall*, dan *F-measure* dapat dihitung untuk menilai seberapa baik model dalam mengklasifikasikan data secara akurat.

3. ANALISA DAN PEMBAHASAN

3.1 Uji Model

Dataset ulasan pengguna yang digunakan dalam penelitian ini diperoleh dari situs berbagi data publik Kaggle, dengan jumlah data sebanyak 8.766 (Delapan ribu tujuh ratus enam puluh enam).

```
# Import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from string import punctuation
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, confusion_matrix, classification_report

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!

[ ] # Upload file CSV
from google.colab import files
uploaded = files.upload()

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable
Saving dataset.gopay.csv to dataset.gopay.csv

[ ] # Load data
df = pd.read_csv('dataset.gopay.csv')

# Seleksi kolom dan hapus baris kosong
df = df[['content', 'score']].dropna()
df.head()
```

Gambar 3. Proses pengumpulan data

```
# Label sentiment: positif (>=4), negatif (<4)
df['sentiment'] = df['score'].apply(lambda x: 'positif' if x >= 4 else 'negatif')
df['content'] = df['content'].str.lower()
df = df[['content', 'sentiment']]
df.head()
```

| | content | sentiment |
|---|---------------------------------------------------|-----------|
| 0 | awesome | positif |
| 1 | beli paket data tansaksinya disuruh tunggu sam... | negatif |
| 2 | bagus ada fitur split bill. saran : tambahn f... | positif |
| 3 | i'm annoyed waiting for my retained balance to... | negatif |
| 4 | awesome.....i love this app | positif |

Gambar 4. Proses pemberian label sentimen.

Tahap *preprocessing* dilakukan untuk menyiapkan dan membersihkan data teks agar layak dianalisis lebih lanjut. Pada tahap ini, pembobotan kata diterapkan menggunakan metode TF-IDF (*Term Frequency-Inverse Document Frequency*), yang berfungsi untuk menentukan tingkat kepentingan kata dalam sebuah dokumen. Nilai bobot TF-IDF diperoleh dengan menggabungkan dua elemen utama, yaitu frekuensi kemunculan kata dalam dokumen (TF) dan frekuensi kebalikannya dalam seluruh korpus (IDF), yang nantinya akan digunakan dalam perhitungan *cosine similarity*.

Setelah menjalani tahap *preprocessing* dan penentuan bobot dengan TF-IDF, dataset yang telah diproses itu kemudian diterapkan dalam algoritma *machine learning*, seperti *K-Nearest Neighbor (KNN)* dan *Support Vector Machine (SVM)*, untuk melakukan klasifikasi sentimen.

```
[ ] # Split data: 80% train, 20% test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Uji model KNN dengan berbagai nilai K
for k in [1, 3, 5, 7, 9, 11]:
    model = KNeighborsClassifier(n_neighbors=k)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    prec = precision_score(y_test, y_pred, pos_label='positif')
    rec = recall_score(y_test, y_pred, pos_label='positif')
    print(f'K={k} --> Akurasi: {acc:.2f}, Presisi: {prec:.2f}, Recall: {rec:.2f}')

K=1 --> Akurasi: 0.79, Presisi: 0.81, Recall: 0.94
K=3 --> Akurasi: 0.79, Presisi: 0.79, Recall: 0.98
K=5 --> Akurasi: 0.77, Presisi: 0.77, Recall: 0.99
K=7 --> Akurasi: 0.77, Presisi: 0.77, Recall: 0.99
K=9 --> Akurasi: 0.76, Presisi: 0.76, Recall: 0.99
K=11 --> Akurasi: 0.76, Presisi: 0.75, Recall: 1.00
```

Gambar 5. Proses klasifikasi menggunakan Model K-NN

```
[ ] # 6. Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # 7. Penanganan Kelas Tidak Seimbang dengan SMOTE
#smote = SMOTE(random_state=42)
#X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

smote = SMOTE(random_state=42, k_neighbors=3, sampling_strategy='minority')
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# 8. Model SVM
svm = SVC(kernel='linear')
svm.fit(X_train_resampled, y_train_resampled) # Latih dengan data resampled
```

SVC

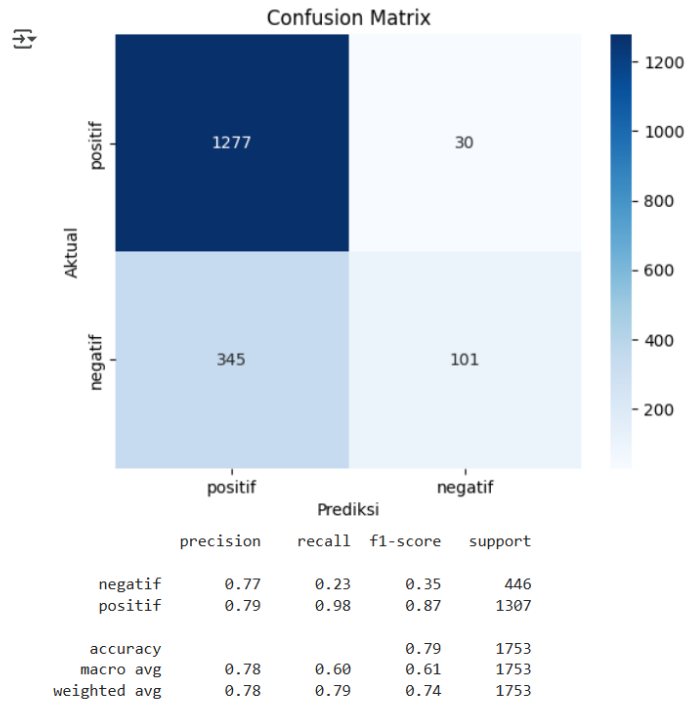
SVC(kernel='linear')

Gambar 6. Proses klasifikasi menggunakan Model SVM

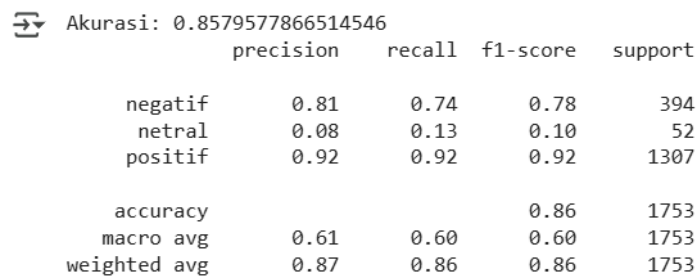
Proses perubahan dataset menjadi numerik menggunakan library scikit-learn tfidf vectorizer secara otomatis, Data tersebut kemudian dipisahkan menjadi 80 persen untuk proses pelatihan dan 20 persen untuk pengujian, tujuannya agar penggunaan model lebih optimal dan diharapkan bisa menganalisis dengan baik. Selanjutnya dataset sudah siap untuk dilakukan evaluasi model.

3.2. Evaluasi Model

Tahap evaluasi model dilakukan sebagai langkah lanjutan setelah proses pengujian untuk menilai efektivitas kinerja model yang digunakan, dengan tujuan untuk mengukur kinerja dari metode klasifikasi yang digunakan. Evaluasi ini dilakukan melalui perhitungan *confusion matrix* berdimensi 2x2, yang merepresentasikan performa klasifikasi dengan mengamati jumlah prediksi yang tepat dan yang keliru pada setiap kategori. Untuk menilai efektivitas metode klasifikasi secara lebih rinci, digunakan metrik seperti *precision*, *recall*, dan *F1-score* pada masing-masing kelas. Ketiga indikator ini memberikan gambaran sejauh mana model mampu mengelompokkan data secara akurat. Nilai metrik tersebut berada dalam skala antara 0 hingga 1, di mana nilai yang mendekati 1 mencerminkan kemampuan prediksi model yang lebih baik.



Gambar 7. Hasil Evaluasi model K-NN



Gambar 8. Hasil Evaluasi model SVM

Dari hasil evaluasi model di atas, nilai akurasi yang diperoleh adalah 0.79 untuk K-NN dan 0.86 untuk SVM, Perhitungan dilakukan dengan membagi jumlah nilai pada diagonal utama dari *confusion matrix* dengan total keseluruhan data yang diuji.

Tabel 1. Perbandingan Akurasi

| Metode | Akurasi |
|---------------------------------------------------|---------|
| <i>K-Nearest Neighbor (K3 + Confusion Matrix)</i> | 79% |
| <i>Support Veector Machine (Kernel Linear)</i> | 86% |

Berdasarkan Tabel 1, metode *Support Vector Machine (SVM)* berhasil memperoleh akurasi sebesar 86%, sementara metode *K-Nearest Neighbors (KNN)* hanya mencapai 79%. Hal ini menunjukkan bahwa KNN kurang efektif dalam melakukan klasifikasi pada penelitian ini. Di sisi lain, SVM dengan kernel Linear memberikan hasil akurasi 86%, yang tergolong cukup baik dalam proses klasifikasi. Setelah dilakukan analisis sentimen menggunakan algoritma machine learning, sistem mampu mengidentifikasi komentar atau konten berdasarkan kategori sentimen positif dan negatif.



4. KESIMPULAN

Setelah dilakukan evaluasi terhadap dua metode klasifikasi, yakni *Support Vector Machine* (SVM) dan *K-Nearest Neighbor* (KNN), diperoleh sejumlah hasil yang dapat dijadikan dasar evaluasi performa masing-masing metode. Sistem ini menunjukkan hasil yang baik dalam mengenali apakah suatu sentimen bersifat positif atau negatif. Berdasarkan hasil pengujian, algoritma SVM mampu mencapai tingkat akurasi sebesar 0,86 atau 86%, sedangkan algoritma KNN hanya memperoleh akurasi sebesar 0,79 atau 79%. Temuan ini mengindikasikan bahwa SVM lebih unggul dibandingkan KNN dalam menangani proses klasifikasi data sentimen dalam konteks penelitian ini.

Secara keseluruhan, temuan ini mengindikasikan bahwa penggunaan algoritma SVM dalam sistem analisis sentimen memiliki potensi yang baik untuk diterapkan dalam studi sejenis, khususnya ketika dibutuhkan tingkat akurasi yang tinggi dalam proses klasifikasi. Namun, perlu diingat bahwa keseimbangan data menjadi elemen krusial yang harus dipertimbangkan selama proses berlangsung pelatihan model, agar performa algoritma dapat lebih maksimal di seluruh kategori sentimen yang dianalisis.

REFERENCES

- Alghifari, F., & Juardi, D. (2021). Penerapan Data Mining Pada Penjualan Makanan Dan Minuman Menggunakan Metode Algoritma Naïve Bayes. *Jurnal Ilmiah Informatika*, 9(02), 75–81. <https://doi.org/10.33884/jif.v9i02.3755>
- Chohan, S., Nugroho, A., Aji, A. M. B., & Gata, W. (2020). Analisis Sentimen Pengguna Aplikasi Duolingo Menggunakan Metode Naïve Bayes dan Synthetic Minority Over Sampling Technique. *Paradigma - Jurnal Komputer Dan Informatika*, 22(2), 139–144. <https://doi.org/10.31294/p.v22i2.8251>
- Idris, I. S. K., Mustofa, Y. A., & Salihi, I. A. (2023). Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM). *Jambura Journal of Electrical and Electronics Engineering*, 5(1), 32–35. <https://doi.org/10.37905/jjee.v5i1.16830>
- Muhammadin, A., & Sobari, I. A. (2021). Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma Svm Dan Nbc. *Reputasi: Jurnal Rekayasa Perangkat Lunak*, 2(2), 85–91. <https://doi.org/10.31294/reputasi.v2i2.785>
- Romadloni, P., Adhi Kusuma, B., & Maulana Baihaqi, W. (2022). Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(2), 622–628. <https://doi.org/10.36040/jati.v6i2.5238>
- Wahyudi, R., & Kusumawardana, G. (2021). Analisis Sentimen pada Aplikasi Grab di Google Play Store Menggunakan Support Vector Machine. *Jurnal Informatika*, 8(2), 200–207. <https://doi.org/10.31294/ji.v8i2.9681>
- Windy Mardiyah, N., Rahaningsih, N., & Ali, I. (2024). Penerapan Data Mining Menggunakan Algoritma K-Nearest Neighbor Pada Prediksi Pemberian Kredit Di Sektor Finansial. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(2), 1491–1499. <https://doi.org/10.36040/jati.v8i2.9010>
- Zahra, F., Ridla, M. A., & Azise, N. (2022). Implementasi Data Mining Menggunakan Algoritma Apriori Dalam Menentukan Persediaan Barang (Studi Kasus : Toko Sinar Harahap). *JUSTIFY : Jurnal Sistem Informasi Ibrahimy*, 3(1), 55–65. <https://doi.org/10.35316/justify.v3i1.5335>