

Perbandingan Metode Support Vector Machine dan K-Nearest Neighbors dalam Analisis Sentimen Ulasan Aplikasi MyXL

Raka Aji Nugroho¹⁾; Dwi Remawati²⁾; Teguh Susyanto³⁾; Wawan Laksito Yuly Saptomo⁴⁾

^{1,3)}Program Studi Sistem Informasi, Universitas Tiga Serangkai Surakarta

²⁾Program Studi Teknologi Informasi, Universitas Tiga Serangkai Surakarta

⁴⁾Program Studi Informatika, Universitas Tiga Serangkai Surakarta

¹⁾23430009.raka@tsu.ac.id; ²⁾wirema@tsu.ac.id; ³⁾teguh@tsu.ac.id; ⁴⁾wawanlaksito@tsu.ac.id

ABSTRACT

Sentiment analysis is an essential tool for understanding user perceptions of mobile applications, especially when reviews are unstructured text. This study aims to analyze user reviews of the MyXL application using Support Vector Machines (SVMs) and K-Nearest Neighbors (KNNs) and to compare their performance in classifying sentiment into positive, negative, and neutral categories. The dataset was obtained from Google Play Store via Kaggle and underwent text preprocessing, including case folding, removal of numbers and punctuation, tokenization, stopword removal, normalization, and stemming. Features were transformed into numerical representations using TF-IDF, and the data was split into training (70%) and testing (30%) sets. Evaluation using accuracy, precision, recall, and F1-score showed that SVM outperformed KNN with an accuracy of 0.743 versus 0.64, particularly in classifying neutral reviews. KNN exhibited higher misclassification rates in both the positive and negative classes, while SVM was more stable but tended to be biased toward the neutral class. These results provide application developers with insights into user satisfaction and guide service improvement and feature development.

Keywords: Sentiment Analysis, Support Vector Machine, K-Nearest Neighbor, MyXL

I. PENDAHULUAN

Salah satu fitur Google Play Store yang memungkinkan pengguna memberikan penilaian terhadap suatu aplikasi adalah ulasan, yang menjadi sumber informasi penting bagi pengembang untuk mengevaluasi kualitas layanan dan fitur aplikasi. Namun, ulasan umumnya berbentuk teks bebas yang tidak terstruktur sehingga sulit digunakan sebagai dasar pengambilan keputusan secara langsung [1].

PT XL Axiata Tbk menyediakan aplikasi layanan mandiri MyXL yang menerima banyak ulasan pengguna dengan beragam karakteristik. Ditemukan ketidaksesuaian antara nilai rating dan isi ulasan, seperti ulasan bernilai tinggi tetapi bermakna negatif, atau sebaliknya, yang berpotensi menyebabkan kesalahan interpretasi terhadap persepsi pengguna [2].

Untuk mengatasi permasalahan tersebut, analisis sentimen digunakan untuk mengelompokkan ulasan ke dalam sentimen positif, negatif, dan netral. Analisis sentimen merupakan bagian dari pemrosesan teks yang bertujuan mengekstraksi opini atau perasaan pengguna dari data tekstual. Penelitian sebelumnya menunjukkan bahwa pendekatan klasifikasi sentimen dapat

membantu pengembang memahami tingkat kepuasan pengguna secara lebih akurat [2], [3].

Dalam penelitian ini, data ulasan aplikasi MyXL dikumpulkan dari Google Play Store [2], kemudian diproses melalui tahap preprocessing teks dan direpresentasikan dalam bentuk numerik menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF) [4], [5], [6]. Selanjutnya, klasifikasi sentimen dilakukan menggunakan algoritma Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN), serta dievaluasi untuk menentukan metode dengan kinerja terbaik [7].

Meskipun metode Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN) telah banyak digunakan dalam analisis sentimen ulasan aplikasi, sebagian besar penelitian sebelumnya hanya berfokus pada perbandingan akurasi tanpa membahas stabilitas model serta pengaruh ketidakseimbangan kelas sentimen. Selain itu, penelitian khusus yang membandingkan kedua metode tersebut pada ulasan aplikasi MyXL masih terbatas dan umumnya hanya menggunakan satu metode klasifikasi [8]. Oleh karena itu, penelitian ini bertujuan untuk membandingkan kinerja SVM dan KNN pada

DOI : <https://doi.org/10.30646/tikomsin.v14i1.1062>
 ulasan aplikasi MyXL secara lebih komprehensif berdasarkan akurasi, F1-score, serta performa pada kelas mayoritas dan kelas minor. Kontribusi penelitian ini diharapkan dapat memberikan dasar pemilihan metode klasifikasi sentimen yang lebih tepat pada data ulasan aplikasi dengan distribusi kelas tidak seimbang [9], [10].

II. TINJAUAN PUSTAKA

2.1 Analisis Sentimen

Salah satu bagian dari pengolahan teks adalah analisis sentimen, yang bertujuan untuk menemukan dan menempatkan pendapat, emosi, atau sikap pengguna dalam teks ke dalam kategori tertentu, seperti sentimen positif, negatif, atau netral. Analisis sentimen banyak digunakan untuk data ulasan pengguna aplikasi di Google Play Store karena dapat secara otomatis menunjukkan tingkat kepuasan pengguna terhadap layanan atau produk tertentu. Analisis sentimen dapat membantu mengubah data yang tidak terstruktur menjadi informasi yang lebih relevan dan mudah dipahami [5].

2.2 Ulasan Aplikasi di Google Play Store

Google Play Store menyediakan fitur ulasan, yang memungkinkan pengguna memberikan ulasan dan komentar terhadap aplikasi yang mereka gunakan. Untuk mengevaluasi kualitas aplikasi, pengembang dapat menggunakan ulasan ini sebagai sumber data penting. Namun, ketidaksesuaian sering terjadi antara penilaian numerik dan isi teks ulasan, yang dapat menyebabkan kesalahan interpretasi apabila hanya melihat penilaian saja tanpa memeriksa teks ulasan secara menyeluruh. Akibatnya, untuk menyelesaikan masalah tersebut, pendekatan analisis sentimen diperlukan [11].

2.3 Text Preprocessing

Preprocessing teks adalah langkah pertama yang sangat penting dalam analisis sentimen. Tujuan dari proses ini adalah untuk membersihkan dan menyiapkan data teks sehingga lebih terorganisir dan siap untuk digunakan dalam proses pembelajaran mesin. Preprocessing yang baik dapat meningkatkan kualitas fitur dan akurasi model klasifikasi sentimen dengan memotong case, menghilangkan tanda baca

dan angka, tokenisasi, menghilangkan stopword, normalisasi kata, dan stemming [6], [9].

2.4 Ekstraksi Fitur Menggunakan TF-IDF

Setelah proses preprocessing, data teks harus diubah menjadi representasi numerik sehingga algoritma pembelajaran mesin dapat memprosesnya. Salah satu metode yang paling umum digunakan adalah Term Frequency Inverse Document Frequency (TF-IDF). TF-IDF memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam sebuah dokumen dan keseluruhan kumpulan dokumen. Ini terbukti efektif dalam representasi teks dan sering digunakan dalam penelitian analisis sentimen berbasis ulasan aplikasi [10].

2.5 Support Virtual Machine

Banyak digunakan dalam klasifikasi teks, seperti analisis sentimen, Support Vector Machine (SVM). Hyperplane terbaik dipilih oleh SVM untuk memisahkan data ke dalam kelas-kelas tertentu secara optimal. SVM memiliki kemampuan untuk menangani data yang sangat besar dan menghasilkan klasifikasi data teks yang baik, yang merupakan salah satu keunggulannya. Akibatnya, SVM sering digunakan sebagai metode pembandingan dalam studi analisis sentiment [12].

2.6 K-Nearest Neighbor

Algoritma klasifikasi berbasis instance K-Nearest Neighbors (KNN) menentukan kelas data berdasarkan mayoritas kelas dari k tetangga terdekatnya. Konsep algoritma ini sederhana dan mudah digunakan. KNN sering digunakan sebagai metode pembandingan dalam analisis sentimen karena mampu memberikan hasil klasifikasi yang sangat baik, meskipun memiliki kelemahan dalam hal waktu komputasi pada dataset yang sangat besar [9],[13].

2.7 Penelitian Terdahulu

Tabel 1 merupakan penelitian terdahulu yang berisi nama peneliti, objek penelitian, metode yang digunakan, dan hasil penelitian.

Tabel 1. Ringkasan Penelitian Terdahulu

Penelitian	Objek	Metode	Hasil

DOI : <https://doi.org/10.30646/tikomsin.v14i1.1062>

ISSN Online : 2620-7532

Diandra Audiansyah	Ulasan Aplikasi MyXL	SVM	Akurasi tinggi, tetapi tidak membandingkan metode lain
Ramadhan Bagus Adi Nugroho	Aplikasi Perpustakaan	SVM vs KNN	SVM unggul, tetapi belum membahas stabilitas dan ketidakseimbangan kelas
Penelitian ini	Ulasan Aplikasi MyXL	SVM vs KNN	Analisis stabilitas dan performa pada kelas mayoritas dan minor

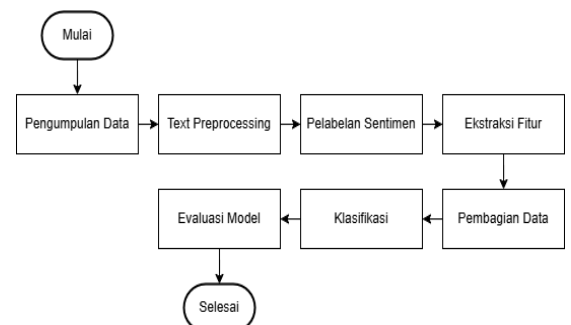
Berbagai penelitian terdahulu menunjukkan bahwa analisis sentimen ulasan aplikasi di Google Play Store dapat dilakukan secara efektif melalui tahapan preprocessing teks, ekstraksi fitur menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF), serta algoritma klasifikasi seperti Support Vector Machine (SVM) dan K-Nearest Neighbor (KNN) [7], [9], [10]. Berdasarkan ringkasan pada Tabel X, penelitian oleh Diandra Audiansyah [2] menerapkan metode SVM pada ulasan aplikasi MyXL dan menghasilkan akurasi yang baik, namun belum melakukan perbandingan dengan algoritma lain. Sementara itu, Ramadhan Bagus Adi Nugroho [8] membandingkan metode SVM dan KNN pada ulasan aplikasi perpustakaan digital dan menemukan bahwa SVM lebih unggul, tetapi belum membahas stabilitas model dan ketidakseimbangan kelas sentimen. Oleh karena itu, penelitian ini membandingkan kinerja SVM dan KNN pada ulasan aplikasi MyXL dengan fokus pada analisis stabilitas model serta performa pada kelas mayoritas dan minor untuk memberikan analisis yang lebih komprehensif.

Berdasarkan ringkasan penelitian terdahulu tersebut, dapat disimpulkan bahwa belum terdapat penelitian yang secara khusus membandingkan metode SVM dan KNN pada ulasan aplikasi MyXL dengan menitikberatkan pada kestabilan model serta performa pada kelas mayoritas

dan minor, sehingga penelitian ini menempati posisi sebagai pengembangan dari penelitian-penelitian sebelumnya.

III. METODE PENELITIAN

Alur metode penelitian pada penelitian ini seperti pada Gambar 1, dimulai dengan pengumpulan data berupa ulasan aplikasi MyXL yang diperoleh dari Kaggle. Dataset yang digunakan merupakan data sekunder yang berasal dari penelitian sebelumnya dan berisi ulasan pengguna aplikasi MyXL di Google Play Store. Setelah data diperoleh, selanjutnya data tersebut diberi label berdasarkan sentimen, yaitu positif, negatif, dan netral.



Gambar 1. Alur Metode Penelitian

Kemudian dilakukan proses pengolahan teks untuk membersihkan dan mempersiapkan data teks agar siap digunakan. Data teks yang telah diproses selanjutnya diubah menjadi bentuk numerik dengan metode *Term Frequency-Inverse Document Frequency* (TF-IDF). Dataset yang telah siap dibagi menjadi dua bagian, yaitu data latih sebesar 70% dan data uji sebesar 30%. Proses klasifikasi dilakukan dengan menggunakan dua algoritma, yaitu K-Nearest Neighbors (KNN) dengan nilai $k = 3$ dan Support Vector Machine (SVM) dengan kernel linear. Pada tahap terakhir, kinerja model diuji menggunakan beberapa metrik penilaian seperti akurasi, precision, recall, dan F1-score, serta hasil klasifikasi ditampilkan dalam bentuk visualisasi [2], [3].

3.1 Pengumpulan Data

Data dikumpulkan menggunakan dataset ulasan aplikasi MyXL yang diperoleh dari situs Kaggle. Dataset ini merupakan data sekunder yang berasal dari penelitian sebelumnya dan berisi ulasan pengguna aplikasi MyXL di Google Play Store. Teks

DOI : <https://doi.org/10.30646/tikomsin.v14i1.1062>
 ulasan, serta informasi tambahan yang berkaitan dengan proses analisis sentimen, digunakan dalam pengumpulan data. Penggunaan Kaggle dipilih karena kelengkapan data yang telah terstruktur dan kemudahan akses.

3.2 Text Preprocessing

Tujuan dari proses preprocessing teks adalah untuk membersihkan dan menyiapkan data teks agar lebih terstruktur dan siap diolah. Proses ini termasuk case folding, penghapusan angka dan tanda baca, tokenisasi, penghapusan stopword, normalisasi kata, dan stemming. Metode ini digunakan untuk meningkatkan kualitas fitur yang dihasilkan dan mengurangi suara data.

Gambar 2 menunjukkan potongan kode yang digunakan untuk melakukan pembersihan teks dalam penelitian ini.

```
#text preprocessing
stemmer =
StemmerFactory().create_stemmer()
stopwords =
set(StopWordRemoverFactory().get_stop_words())

def clean_text(text):
    text = re.sub(r'http\S+|@[A-Za-z0-9]+|##[A-Za-z0-9]+', ' ', text)
    text = re.sub(r'[0-9]+', ' ', text)
    text = re.sub(r'^\w\s', ' ', text)
    text = re.sub(r'(\.)\{2,\}', r'\1', text)
    text = text.lower().strip()

    tokens = text.split()
#tokenisasi
    tokens =
[normalisasi_dict.get(k, k) for k in tokens] #normalisasi
    tokens = [k for k in tokens if k not in stopwords] #stopword
    tokens = [stemmer.stem(k) for k in tokens] #stemming

    return ' '.join(tokens)

df['Cleaned'] =
df['Ulasan'].apply(clean_text)
```

Gambar 2. Kode Proses Text Preprocessing

Gambar 3 menunjukkan hasil dari proses preprocessing teks, yang menunjukkan bahwa teks ulasan menjadi lebih bersih dan seragam. Teks ulasan sebelum dan sesudah pembersihan berbeda.

Ulasan	Cleaned
Harusnya dikasih bintang 4 bilang terimakasih....	harus kasih bintang bilang terimakasih mau nga...
Tolong dong masalah jaringan hampir setiap har...	dong masalah jaring hampir hari leg parahh jan...
Saya mau komen lagi,miminnn kenapa akhir akhi...	mau komen mimin akhir akhir jaring xl sangat l...
Bonus kouta tiktok 13gb tidak diaktifkan/tidak...	bonus kouta tiktok gb aktif beri jadi kouta yg...
Sejauh ini bagusA? aja sih apk nya, tpi udh bb...	jauh bagus aja sih apk nya tpi udh bbrp hari s...
...	...
Xl sekarang beda sama dulu, sekarang xl boros...	xl sekarang beda sama dulu sekarang xl boros b...
Sebagai Pelanggan Setia Xl. Bertahun tahun , Ke...	langgan setia xl tahun tahun kecewa layan xl s...
Suka boros kalo pake sekali beli mahal bgt kal...	suka boros kalo pake sekali beli mahal banget ...

Gambar 3. Hasil Text Preprocessing

3.3 Pelabelan Sentimen

Pelabelan sentimen dilakukan untuk mengelompokkan setiap ulasan ke dalam tiga kelas, yaitu sentimen positif, negatif, dan netral. Proses pelabelan dalam penelitian ini dilakukan secara otomatis menggunakan pendekatan berbasis kamus kata (lexicon-based), dengan menetapkan kriteria tertentu berdasarkan kemunculan kata kunci sentimen untuk menghasilkan kelas target yang digunakan dalam proses klasifikasi [3], [7]. Pendekatan ini memiliki keterbatasan karena hanya mengandalkan kemunculan kata kunci tanpa mempertimbangkan konteks kalimat secara menyeluruh, seperti negasi, sarkasme, atau struktur kalimat kompleks, serta dilakukan tanpa validasi manual, sehingga berpotensi menimbulkan bias dalam penentuan kelas sentimen dan memengaruhi distribusi kelas pada data. Kode pelabelan seperti pada Gambar 4.

```
#menambahkan label sentimen
positif_words =
['mantap','cepat','mudah','ramah','pu as','bagus','stabil','lancar']
negatif_words =
['jelek','lemot','kesal','buruk','lam bat','susah','tidak nyaman']
netral_words =
['biasa','standar','lumayan','oke','t idak buruk','tidak terlalu']

def label_sentimen(text):
    pos_count = sum(1 for word in positif_words if word in text)
    neg_count = sum(1 for word in negatif_words if word in text)
    net_count = sum(1 for word in netral_words if word in text)

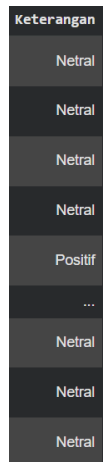
    if pos_count > neg_count and pos_count > net_count:
        return 'Positif'
    elif neg_count > pos_count and neg_count > net_count:
        return 'Negatif'
```

```
elif net_count > pos_count and
net_count > neg_count:
    return 'Netral'
else:
    return 'Netral'

df['Keterangan'] =
df['Cleaned'].apply(label_sentimen)
df
```

Gambar 4. Kode Proses Pelabelan Sentimen

Gambar 5 menunjukkan hasil dari pelabelan sentimen.



Gambar 5. Hasil Pelabelan Sentimen

3.4 Ekstraksi Fitur

Ekstraksi fitur digunakan untuk mengubah data teks menjadi representasi numerik yang dapat diproses oleh algoritma pembelajaran mesin. Metode *Term Frequency Inverse Document Frequency* (TF-IDF) digunakan. TF-IDF memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam kedua dokumen dan dataset secara keseluruhan. Gambar 6 merupakan proses ekstraksi fitur.

```
#menggabungkan fitur untuk metode
KNN dan SVM
vectorizer = TfidfVectorizer()
X_text =
vectorizer.fit_transform(df['Cleaned
'])
X_rating =
df['Score'].values.reshape(-1, 1)

#menggabungkan TF-IDF dengan numeric
X_combined = hstack([X_text,
X_rating])
y = df['Sentimen_Num']
```

Gambar 6. Kode Ekstraksi Fitur

Tingkat kepentingan kata dalam dokumen ulasan ditentukan dengan TF-IDF. Metode stacking horizontal

menggabungkan vektor TF-IDF dengan fitur teks dan skor penilaian pengguna. Tujuan dari integrasi fitur ini adalah untuk meningkatkan jumlah informasi yang digunakan oleh algoritma KNN dan SVM untuk melakukan klasifikasi sentiment.

3.5 Pembagian Data

Data set yang telah diekstraksi kemudian dibagi menjadi dua bagian: data latihan (training) dan data uji (testing). Sebagian besar dari data uji digunakan untuk membangun model klasifikasi, sedangkan sebagian besar data latihan digunakan untuk mengukur kinerja model. Proses pembagian data yang digunakan dalam penelitian ini ditunjukkan pada Gambar 7.

```
#memisahkan data menjadi 70%
training dan 30% testing
X_train, X_test, y_train, y_test =
train_test_split(
    X_combined, y, test_size=0.3,
    random_state=42
)
```

Gambar 7. Kode Pembagian Data

Dalam penelitian ini, fungsi `train_test_split` digunakan untuk membagi data menjadi 70% data latihan dan 30% data uji. Nilai `random_state` digunakan untuk memastikan bahwa proses pembagian data konsisten dan dapat diperduplikasi.

3.6 Klasifikasi

Dua algoritma digunakan untuk melakukan tahap klasifikasi: K-Nearest Neighbors (KNN) dan Support Vector Machine (SVM). Metode KNN menentukan kelas sentimen berdasarkan tetangga terdekat dengan menggunakan nilai $k = 3$. Sebaliknya, metode SVM memisahkan data ke dalam kelas sentimen dengan menggunakan kernel linier. Untuk mendapatkan hasil yang dapat dibandingkan, kedua pendekatan digunakan pada data yang sama. Untuk menghasilkan hasil klasifikasi yang dapat dibandingkan secara adil, kedua algoritma digunakan pada dataset yang sama. Gambar 8 menunjukkan metode klasifikasi sentimen menggunakan metode KNN dan SVM [8], [14].

```
#model KNN K3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)

#SVM dengan kernel linear
svm = SVC(kernel='linear')
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
```

Gambar 8. Kode Klasifikasi

3.7 Evaluasi Model

Evaluasi model dilakukan untuk mengetahui kinerja algoritma KNN dan SVM dalam mengklasifikasikan sentimen ulasan. Metode yang dievaluasi termasuk akurasi, ketepatan, recall, dan skor F1. Hasil evaluasi digunakan untuk membandingkan kinerja kedua algoritma dan menentukan yang memberikan hasil terbaik. Pada Gambar 9 merupakan proses untuk evaluasi model.

```
#menampilkan akurasi metode KNN
print("Akurasi KNN:",
accuracy_score(y_test, y_pred_knn))
print("Classification Report KNN:\n",
classification_report(y_test,
y_pred_knn))
print("Confusion Matrix KNN:\n",
confusion_matrix(y_test, y_pred_knn))

#menampilkan akurasi metode SVM
print("Akurasi SVM:",
accuracy_score(y_test, y_pred_svm))
print("Classification Report SVM:\n",
classification_report(y_test,
y_pred_svm))
print("Confusion Matrix SVM:\n",
confusion_matrix(y_test, y_pred_svm))
```

Gambar 9. Kode Evaluasi Model

Gambar 10 merupakan hasil evaluasi model KNN. Dengan akurasi 0.64 dan performa terbaik pada kelas 1 (sentimen netral atau positif, tergantung label), KNN memiliki precision dan recall yang lebih rendah pada kelas 0 dan 2, seperti yang ditunjukkan oleh confusion matrix, di mana banyak prediksi kelas 0 dan 2 yang salah diklasifikasikan.

```
*** Akurasi KNN: 0.64
Classification Report KNN:
precision    recall  f1-score   support

0           0.25     0.33     0.29         30
1           0.76     0.76     0.76        201
2           0.50     0.42     0.46         69

accuracy          0.64         300
macro avg         0.50         300
weighted avg      0.65         300

Confusion Matrix KNN:
[[ 10  18   2]
 [ 21 153  27]
 [   9  31  29]]
```

Gambar 10. Akurasi Metode KNN

Gambar 11 merupakan hasil akurasi model SVM, yang memiliki akurasi lebih tinggi, yaitu 0.743, dengan precision dan recall terbaik pada kelas 1, serta performa lebih stabil pada kelas 2 dibandingkan dengan KNN. Confusion matrix menunjukkan bahwa SVM lebih mampu membedakan kelas dengan jumlah data besar (kelas 1), meskipun kelas minor (kelas 0) masih sering salah klasifikasi.

```
Akurasi SVM: 0.7433333333333333
Classification Report SVM:
precision    recall  f1-score   support

0           0.40     0.07     0.11         30
1           0.74     0.95     0.83        201
2           0.79     0.45     0.57         69

accuracy          0.74         300
macro avg         0.65         300
weighted avg      0.72         300

Confusion Matrix SVM:
[[  2  28   0]
 [  3 190   8]
 [  0  38  31]]
```

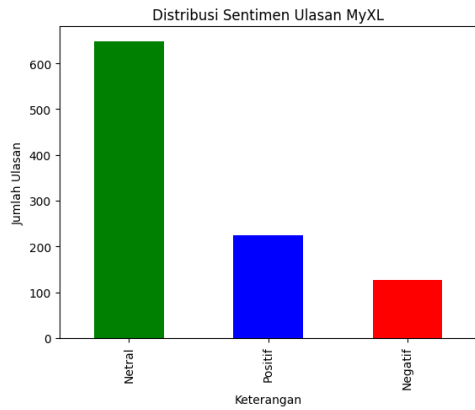
Gambar 11. Akurasi Metode SVM

IV. HASIL DAN PEMBAHASAN

Untuk membuat interpretasi dan analisis hasil penelitian lebih mudah, visualisasi hasil menampilkan grafik perbandingan akurasi metode KNN dan SVM, confusion matrix, dan distribusi sentimen ulasan pengguna aplikasi MyXL.

4.1 Distribusi Sentimen Ulasan

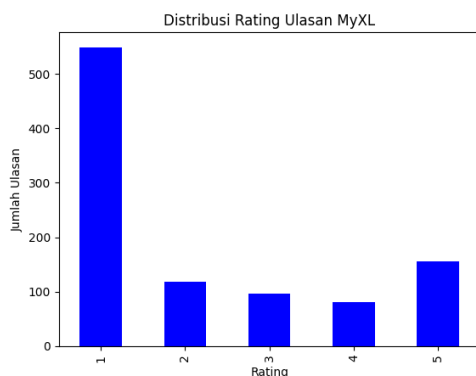
Gambar 12 merupakan distribusi sentimen ulasan MyXL yang menunjukkan bahwa sekitar 650 ulasan menunjukkan sentimen netral, dengan 220 ulasan positif dan 130 ulasan negatif, yang menunjukkan bahwa sebagian besar pengguna aplikasi MyXL tidak sepenuhnya puas atau tidak tidak puas. Namun, lebih banyak ulasan positif daripada ulasan negatif menunjukkan persepsi. Hasil ini dapat digunakan sebagai dasar evaluasi untuk meningkatkan pengalaman pengguna dan kualitas layanan.



Gambar 12. Visualisasi Berdasarkan Sentimen

4.2 Distribusi Rating Ulasan

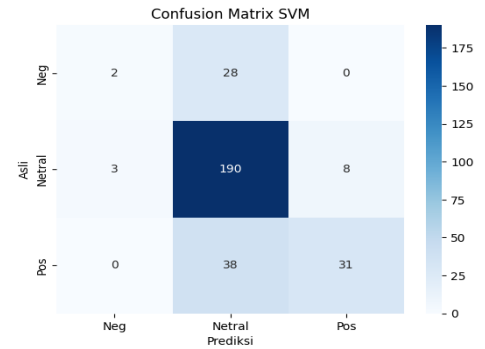
Gambar 13 merupakan distribusi rating ulasan MyXL yang menunjukkan bahwa rating 1 mendominasi dengan sekitar 550 ulasan, diikuti oleh rating 5 dengan 160 ulasan, dan rating 2, 3, dan 4 masing-masing memiliki jumlah yang relatif lebih sedikit dan tersebar secara merata. Dominasi rating rendah menunjukkan bahwa aplikasi MyXL memiliki tingkat ketidakpuasan pengguna yang cukup tinggi, meskipun masih ada beberapa pengguna yang memberikan rating yang sangat baik.



Gambar 13. Rating atau Score

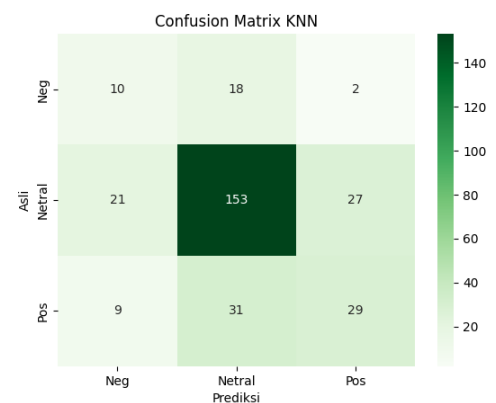
4.3 Hasil Confusion Matrix

Hasil Confusion Matrix SVM menunjukkan performa yang lebih baik dalam mengklasifikasikan sentimen netral, dengan 190 data terklasifikasi secara benar. Akan tetapi, SVM juga mengalami kesalahan klasifikasi pada sentimen negatif dan positif, yang sebagian besar diprediksi sebagai netral, menunjukkan kecenderungan bias terhadap kelas netral. Ditunjukkan pada Gambar 14.



Gambar 14. Confusion Matrix SVM

Gambar 15 merupakan hasil Confusion Matrix KNN yang menunjukkan bahwa model mampu mengklasifikasikan sentimen netral dengan benar sebanyak 153 data, sedangkan klasifikasi benar untuk sentimen negatif dan positif masing-masing sebanyak 10 dan 29 data. Namun, masih terdapat kesalahan klasifikasi yang cukup tinggi, terutama pada sentimen negatif dan positif yang sering diprediksi sebagai netral.

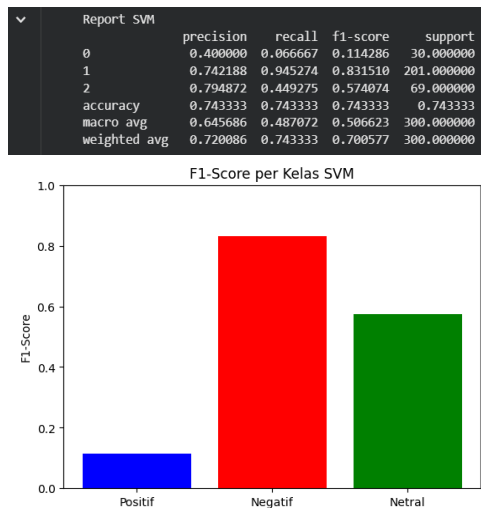


Gambar 15. Confusion Matrix KNN

Ketika kedua model dibandingkan, terlihat bahwa SVM lebih baik dalam klasifikasi daripada KNN, terutama dalam menemukan sentimen netral. Namun, kedua model masih menghadapi masalah dalam membedakan sentimen positif dan negatif, sehingga diperlukan pengembangan tambahan untuk meningkatkan kinerja klasifikasi, seperti penyeimbangan data atau optimasi parameter.

4.4 Report dan F1 Score SVM

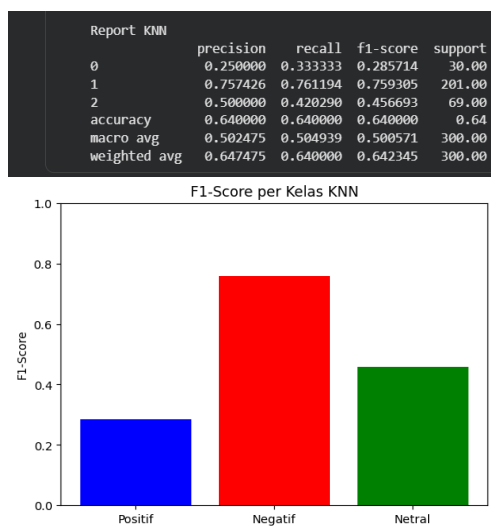
Gambar 10 menampilkan nilai F1-Score model Support Vector Machine (SVM) untuk masing-masing kelas, yaitu Positif, Negatif, dan Netral. Berdasarkan hasil evaluasi, kelas Negatif memperoleh nilai F1-Score tertinggi ($\approx 0,81$), diikuti kelas Netral ($\approx 0,58$), sedangkan kelas Positif memiliki nilai F1-Score paling rendah ($\approx 0,11$).



Gambar 10. F1 Score SVM

4.5 Report dan F1 Score KNN

Gambar 11 menunjukkan nilai F1-Score model K-Nearest Neighbor (KNN) untuk masing-masing kelas, yaitu Positif, Negatif, dan Netral. Berdasarkan hasil evaluasi, kelas Negatif memperoleh nilai F1-Score tertinggi ($\approx 0,76$), diikuti oleh kelas Netral ($\approx 0,46$), sedangkan kelas Positif memiliki nilai F1-Score paling rendah ($\approx 0,29$).



Gambar 11. F1 Score KNN

Tabel 2. Perbandingan F1 Score SVM dan F1 Score KNN

Metode	Positif	Netral	Negatif	Avg
SVM	0.114	0.832	0.574	0.507
KNN	0.286	0.759	0.457	0.501

Tabel 2 menunjukkan perbandingan hasil confusion matrix. Analisis F1-Score per Kelas sebagai berikut :

a. Kelas Positif

Meskipun KNN memiliki skor F1 yang lebih tinggi pada kelas positif, nilai keduanya relatif rendah, menunjukkan bahwa kedua model menghadapi masalah dalam mengklasifikasikan data positif yang termasuk kelas minor. Karena SVM cenderung bias terhadap kelas yang lebih dominan, mereka tidak berhasil pada kelas ini.

b. Kelas Netral

Dengan dominasi kelas netral dalam dataset, SVM lebih stabil dalam memprediksi kelas netral. SVM juga berhasil pada kelas netral dengan skor F1 yang lebih tinggi daripada KNN.

c. Kelas Negatif

SVM unggul dibandingkan dengan KNN pada kelas negatif, meskipun skor F1 tidak setinggi kelas netral. Ini menunjukkan bahwa kelas minor masih sulit diklasifikasikan karena jumlah datanya lebih sedikit daripada kelas mayoritas.

Hasil evaluasi menunjukkan bahwa metode Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN) memiliki karakteristik performa yang berbeda dalam mengklasifikasikan sentimen ulasan. SVM unggul dalam mengklasifikasikan kelas mayoritas (Netral), yang ditunjukkan oleh nilai precision, recall, dan F1-score yang lebih tinggi dibandingkan dengan KNN, sebagaimana terlihat pada Confusion Matrix SVM dan Tabel Perbandingan F1-Score (Tabel 2). Selain itu, SVM juga menunjukkan performa yang lebih baik pada kelas negatif, dengan nilai F1-score yang lebih tinggi dibandingkan dengan KNN. Keunggulan ini menunjukkan tingkat stabilitas model SVM yang lebih baik, yang tercermin dari selisih

DOI : <https://doi.org/10.30646/tikomsin.v14i1.1062>

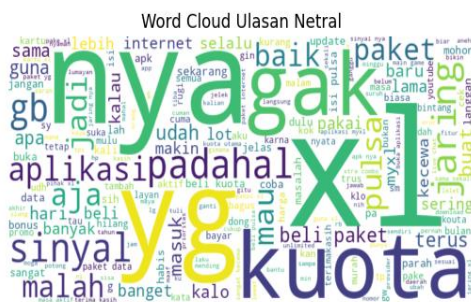
ISSN Online : 2620-7532

nilai precision dan recall antarkelas yang relatif lebih kecil dibandingkan dengan KNN.

Meskipun dataset memiliki ketidakseimbangan kelas, performa SVM tidak mengalami penurunan yang signifikan, sedangkan KNN menunjukkan peningkatan kesalahan klasifikasi pada kelas minor yang sering diprediksi sebagai kelas netral, sebagaimana ditunjukkan pada Confusion Matrix KNN. Hal ini menunjukkan bahwa SVM lebih efektif dalam menangani distribusi data yang tidak seimbang pada dataset ulasan MyXL.

Di sisi lain, KNN menunjukkan ketergantungan yang lebih besar terhadap distribusi data karena proses klasifikasinya didasarkan pada mayoritas tetangga terdekat. Meskipun KNN relatif sederhana dan mudah diimplementasikan, kestabilan dan akurasi model secara keseluruhan cenderung menurun ketika dihadapkan pada dominasi kelas mayoritas. Oleh karena itu, pemilihan metode klasifikasi perlu disesuaikan dengan karakteristik data dan tujuan analisis sentimen.

Visualisasi Word Cloud untuk Sentimen Netral dibuat untuk memahami kata-kata dominan yang muncul dalam ulasan netral aplikasi MyXL yang ditampilkan pada Gambar 12. Word Cloud ini menampilkan kata-kata yang paling sering digunakan oleh pengguna, tanpa membawa nilai emosional yang jelas (tidak positif maupun negatif).



Gambar 12. World Cloud

V. PENUTUP

5.1 Kesimpulan

Hasil penelitian menunjukkan bahwa Support Vector Machine (SVM) dan K-Nearest Neighbors (KNN) keduanya mampu

digunakan untuk melakukan analisis sentimen pada ulasan aplikasi MyXL. SVM menunjukkan kinerja yang lebih baik dengan nilai akurasi sebesar 0,743 dibandingkan dengan KNN sebesar 0,64, serta lebih stabil dalam mengklasifikasikan kelas mayoritas (Netral), meskipun masih terdapat kesalahan klasifikasi pada kelas minor (Positif dan Negatif). KNN menunjukkan performa yang relatif lebih baik pada sebagian kelas minor, khususnya kelas positif, namun kinerjanya cenderung menurun pada kondisi distribusi data yang tidak seimbang. Berdasarkan distribusi sentimen dan analisis Word Cloud, sebagian besar ulasan pengguna bersifat netral, sehingga interpretasi sentimen perlu dilakukan secara hati-hati agar mencerminkan persepsi pengguna secara akurat. Secara keseluruhan, hasil penelitian menegaskan pentingnya pemilihan metode klasifikasi yang sesuai dengan karakteristik distribusi data sentimen.

Secara ilmiah, penelitian ini berkontribusi dalam memberikan analisis komparatif kinerja dan stabilitas metode SVM dan KNN pada data sentimen dengan distribusi kelas yang tidak seimbang, khususnya pada ulasan aplikasi MyXL, sehingga dapat menjadi referensi dalam pemilihan metode klasifikasi sentimen pada penelitian sejenis.

5.2 Saran

Agar model menjadi lebih akurat, terutama untuk penelitian mendatang, disarankan untuk menggunakan dataset yang lebih besar dan seimbang. Ini terutama berlaku untuk kelas minor (positif dan negatif). Untuk meningkatkan kinerja, metode klasifikasi tambahan seperti Random Forest, Naive Bayes, atau deep learning dapat dicoba. Selain itu, metadata ulasan, seperti waktu pembuatan, lokasi, dan jenis perangkat, dapat diperkaya dengan menambah informasi yang digunakan model. Untuk membuat model lebih responsif terhadap berbagai bahasa ulasan pengguna, disarankan agar proses preprocessing teks dioptimalkan serta penyesuaian hiperparameter SVM dan KNN dilakukan. Terakhir, pengembang aplikasi dapat menggunakan sistem pemantauan sentimen secara real-time untuk membantu mereka membuat keputusan cepat.

DAFTAR PUSTAKA

- [1] I. Aida Sapitri, M. Fikry, F. Sains dan Teknologi, and U. Islam Negeri Sultan Syarif Kasim Riau, "Pengklasifikasian Sentimen Ulasan Aplikasi Whatsapp Pada Google Play Store Menggunakan Support Vector Machine," *Jurnal TEKINKOM*, vol. 6, no. 1, 2023, doi: 10.37600/tekinkom.v6i1.773.
- [2] D. Diandra Audiansyah, D. Eka Ratnawati, and B. Trias Hanggara, "Analisis Sentimen Aplikasi MyXL menggunakan Metode Support Vector Machine berdasarkan Ulasan Pengguna di Google Play Store," 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [3] I. Garuda, P. Alva, E. Zuliarso, and U. Stikubank, "Analisis Sentimen Berdasarkan Ulasan Google Play Store Pada Aplikasi Dana Menggunakan Metode Support Vector Machine (SVM)," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 8, no. 5, 2025.
- [4] S. F. Kadir and A. Fairuzabadi, "Analisis Sentimen Ulasan Shopee di Google Play dengan TF-IDF dan Logistic Regression," *RIGGS: Journal of Artificial Intelligence and Digital Business*, vol. 4, no. 2, pp. 7940–57945, Jul. 2025, doi: 10.31004/riggs.v4i2.2850.
- [5] N. Putri Husain, A. Febriana Syam, and R. Mustikosari, "Analisis Sentimen Ulasan Pengguna Tiktok pada Google Play Store Berbasis TF-IDF dan Support Vector Machine," 2024. [Online]. Available: <https://images.app.goo.gl/hC6494uW637VmYVW9>
- [6] A. Suharman and M. Kamayani Sulaeman, "Analisis Sentimen Pengguna Aplikasi Livin' by Mandiri Menggunakan Metode Support Vector Machine (SVM) dengan Ekstraksi Fitur TF-IDF dan Word2Vec," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 8, pp. 2201–2212, Aug. 2025, doi: 10.52436/1.jpti.941.
- [7] "Analisis Sentimen Pada Ulasan Aplikasi Home Credit dengan Metode SVM dan KNN", [Online]. Available: <https://ejournal.mediaantartika.id/index.php/jka>
- [8] Ramadhan Bagus Adi Nugroho, Mukh Taofik Chulkamdi, and Rizki Dwi Romadhona, "Perbandingan Performa Algoritma Support Vector Machine dan K-Nearest Neighbors terhadap Analisis Sentimen Aplikasi Perpustakaan Digital," *Jurnal Ilmiah Teknik Informatika dan Komunikasi*, vol. 5, no. 3, pp. 115–126, Aug. 2025, doi: 10.55606/juitik.v5i3.1522.
- [9] D. Fajar Nawulansih, N. Ceisa Santi, and I. Aristia Sa'ida, "Analisis Sentimen Ulasan Aplikasi DANA di Google Play Store: Penerapan Support Vector Machine dan Synthetic Minority Over-sampling Technique," *Jurnal Pendidikan dan Teknologi Indonesia*, vol. 5, no. 9, pp. 2660–2671, Sep. 2025, doi: 10.52436/1.jpti.1053.
- [10] D. Remawati, E. Noersasongko, A. Marjuni, and Pujiono, "Mental Health Detection with TF-IDF Feature Extraction," *Int. Conf. Artif. Intell. Mechatronics Syst. AIMS 2024*, pp. 1–6, 2024, doi: 10.1109/AIMS61812.2024.10512480.
- [11] D. Remawati, H. Wijayanto, Y. Retno, W. Utami, and B. D. Raharja, "Pengelompokan Film Trending di Youtube Menggunakan TF-IDF," *Jurnal Sistem Informasi TGD*, vol. 4, pp. 65–74, 2025.
- [12] Y. Y. Zandroto, A. V. Vitianingsih, A. L. Maukar, N. K. Hikmawati, and R. Hamidan, "Sentiment Analysis of BCA Mobile App Reviews Using K-Nearest Neighbour and Support Vector Machine Algorithm," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 8, no. 2, p. 448, Aug. 2025, doi: 10.24014/ijaidm.v8i2.37773.
- [13] R. B. Afandi, T. Fajar Nurdiansyah, A. N. Ramadhani, and A. P. Sari, "JIP (Jurnal Informatika Polinema) Implementasi Support Vector Machine Untuk Analisis Sentimen Aplikasi 'MPStore-Super App UMKM'".
- [14] A. Syaifudin Agil Rafsanjani *et al.*, "Sistemasi: Jurnal Sistem Informasi Analisis Sentimen Ulasan Pengguna

DOI : <https://doi.org/10.30646/tikomsin.v14i1.1062>

ISSN Online : 2620-7532

Aplikasi KitaLulus pada Google Play Store menggunakan algoritma Support Vector Machine (SVM) Sentiment Analysis of User Reviews of the KitaLulus Application on Google Play Store using the Support Vector Machine (SVM) Algorithm." [Online]. Available:
<http://sistemasi.ftik.unisi.ac.id>