

Analisis Sentimen Pada Ulasan Aplikasi Home Credit dengan Metode SVM dan KNN

Arman Adiansyah¹, Wahyudin²

^{1,2}Universitas Bina Sarana Informatika, Teknologi Informasi
Jl. Kramat No. 98, Jakarta Pusat, Indonesia

e-mail korespondensi: armanadiansyah1@gmail.com

Abstrak - Dalam era teknologi, mencari pembiayaan finansial semakin mudah melalui aplikasi *mobile* seperti *Home Credit*. Aplikasi ini telah diunduh oleh lebih dari 10 juta pengguna Android dengan peringkat keseluruhan 4,4 di *Google Play Store*. Untuk membantu meninjau aplikasi, pengguna dapat memberikan ulasan dan penilaian di *Google Play Store*. Namun, dengan banyaknya ulasan, diperlukan analisis sentimen untuk mempermudah pemaha man. Dalam penelitian ini, dilakukan analisis sentimen menggunakan metode *Support Vector Machine (SVM)* dan *K-Nearest Neighbor (KNN)* pada data ulasan dari *Google Play Store*. Data yang diambil berjumlah 2.845 dengan informasi tentang skor dan komentar. Sentimen positif dan negatif ditentukan berdasarkan skor, dengan skor 4 dan 5 untuk sentimen positif, serta skor 1, 2, dan 3 untuk sentimen negatif. Setelah tahap *preprocessing* dan penghitungan *tf-idf*, dilakukan perhitungan menggunakan algoritma SVM dan KNN. Hasilnya menunjukkan bahwa metode SVM memiliki presisi 89%, *recall* 86%, F1-score 87%, dan akurasi 88%. Sementara metode KNN memiliki presisi 79%, *recall* 80%, F1-score 79%, dan akurasi 79%. Berdasarkan hasil tersebut, dapat disimpulkan bahwa metode *Support Vector Machine* lebih baik dalam melakukan analisis sentimen dalam penelitian ini.

Kata Kunci: Analisis Sentimen, *Home Credit*, *Support Vector Machine*, *K-Nearest Neighbor*

Abstracts - In the age of technology, finding finance has never been easier through mobile apps like *Home Credit*. The app has been downloaded by over 10 million Android users with an overall rating of 4.4 on the *Google Play Store*. To help review the app, users can leave reviews and ratings on the *Google Play Store*. However, with so many reviews, sentiment analysis is needed to facilitate understanding. In this study, sentiment analysis using the *Support Vector Machine (SVM)* and *K-Nearest Neighbor (KNN)* methods was conducted on review data from the *Google Play Store*. The data taken amounted to 2,845 with information about scores and comments. Positive and negative sentiments are determined based on scores, with scores of 4 and 5 for positive sentiments, and scores of 1, 2, and 3 for negative sentiments. After the preprocessing stage and *tf-idf* calculation, calculations are performed using the SVM and KNN algorithms. The results show that the SVM method has 89% precision, 86% recall, 87% F1-score, and 88% accuracy. While the KNN method has 79% precision, 80% recall, 79% F1-score, and 79% accuracy. Based on these results, it can be concluded that the *Support Vector Machine* method is better at performing sentiment analysis in this study.

Keywords: Sentiment Analysis, *Home Credit*, *Support Vector Machine*, *K-Nearest Neighbor*

1. Pendahuluan

Teknologi berkembang pesat pada setiap bidang kehidupan, salah satu bidang penting yang terdampak perkembangan teknologi ini adalah bidang finansial, sehingga hal-hal seperti pembayaran, bank, pinjaman dan lain-lain sudah bisa dilakukan dengan mudah. Sudah banyak perusahaan yang menawarkan jasa pembiayaan finansial berupa pinjaman dengan proses dan syarat yang sangat mudah, cukup mendaftar melalui aplikasi dan memenuhi persyaratannya, maka sudah bisa mendapatkan pinjaman, salah satu perusahaan tersebut yaitu PT *Home Credit* Indonesia. Supaya bisa menjangkau banyak pengguna di era modern ini, perusahaan membuat aplikasi *mobile Home Credit* agar pengguna jasa lebih mudah dalam mengajukan, mengelola dan membayar tagihan. Aplikasi *Home Credit* bisa di *download* di *Google Play Store*. [1]

Seperti layanan aplikasi pada umumnya, layanan *Home Credit* juga memiliki layanan yang kurang sesuai dengan keinginan pengguna. Maka sebelum mengunduh dan mendaftar, bisa melihat terlebih dahulu ulasan yang diberikan oleh pengguna sebelumnya tentang kelebihan dan kekurangan aplikasi *Home Credit*. *Reviews* atau ulasan pengguna adalah salah satu bentuk umpan balik yang diberikan oleh pengguna terhadap produk atau layanan yang mereka gunakan. [2] Ulasan pengguna dapat berisi pendapat, kesan, atau saran yang berkaitan dengan kualitas, fitur, atau kinerja produk atau layanan tersebut. Tetapi dengan banyaknya jumlah ulasan yang ada pada *Google Play Store*, dan tidak sedikit pengguna yang tidak sesuai dalam memberikan rating dan komentar seperti berkomentar baik tetapi memberikan *rating* satu atau sebaliknya. Analisis sentimen sangat diperlukan untuk



menyaring komentar-komentar media sosial. Analisis sentimen pada komentar dilakukan untuk mengetahui komentar yang bersifat negatif dan komentar yang bersifat positif [3]. Karena memahami semua ulasan secara manual dapat menjadi tugas yang melelahkan dan memakan waktu, maka dari itu, analisis sentimen dengan bantuan *machine learning* dan teknik-teknik komputasi bisa digunakan untuk mengatasi masalah ini. Analisis sentimen digunakan untuk menganalisis dan mendapatkan informasi dari data teks yang ada, dan akan menghasilkan identifikasi teks apakah teks atau ulasan itu bersentimen negatif atau positif [3]. Tugas dasar dari analisis sentimen adalah mengklasifikasikan polaritas teks yang ada dalam kalimat, dokumen, maupun pendapat pasti memiliki aspek positif atau negatif [4].

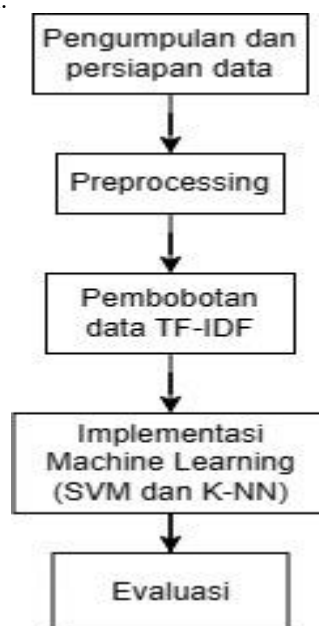
Algoritma SVM (*Support Vector Machine*) dan KNN (*K-Nearest Neighbors*) adalah dua pendekatan yang sering digunakan dalam analisis sentimen, masing-masing dengan keunggulan dan karakteristik unik. SVM dikenal dengan kemampuannya dalam menangani pemisahan sentimen dalam data yang tidak linier, serta kemampuannya untuk menangani outlier. Di sisi lain, KNN adalah algoritma berbasis instansi yang bersifat adaptif terhadap data baru dan lebih sederhana dalam konsepnya [5].

SVM adalah algoritma yang mempunyai *hyperplane* terbaik sebagai pemisah dua kelas data berbeda [6]. Algoritma *K-Nearest Neighbor* adalah metode untuk mengklasifikasikan objek-objek berdasarkan data yang memiliki jarak terdekat dengan data objek tersebut [7]. Dengan bantuan metode *machine learning* yang sudah ditentukan, analisis sentimen dapat dilakukan secara otomatis dan efisien dalam menganalisis besar volume teks dari ulasan pengguna, dan bisa dimanfaatkan untuk mendapatkan wawasan yang lebih mendalam dari data ulasan dan memahami sentimen pengguna dengan lebih efisien dan akurat.

Penelitian sebelumnya yang dilakukan oleh M. Nurul Muttaqin dkk., dengan judul Analisis Sentimen Aplikasi Gojek Menggunakan Support Vector Machine dan K-Nearest Neighbor. Tujuan penelitian ini adalah melakukan klasifikasi pada ulasan aplikasi Gojek menggunakan SVM dan KNN. Sumber data yang digunakan diambil dari Google Play Store, karena data yang didapatkan murni opini pengguna tanpa ada promosi dan iklan. Hasilnya metode KNN dengan $K=22$ memperoleh nilai akurasi, presisi, dan recall dengan 82,14%, 82,28%, dan 95,43% dan SVM memperoleh nilai akurasi, presisi, dan recall berturut-turut sebesar 87,98%, 88,55%, dan 95,43%. Disimpulkan bahwa metode SVM melakukan klasifikasi dengan lebih baik dari metode KNN. [8]. Sedangkan penelitian lainnya dilakukan Atang Saepudin, dkk., dengan judul Optimasi Algoritma SVM Dan K-NN Berbasis *Particle Swarm Optimization* Pada Analisis *Sentiment* Fenomena Tagar #2019GantiPresiden, penelitian ini menggunakan algoritma SVM dan KNN, lalu ditambahkan optimasi yaitu *Particle Swarm Optimization*. Data yang digunakan sebanyak 400 *review* terdiri dari 200 *review* positif dan 200 *review* negatif. Hasilnya SVM menghasilkan akurasi 88,00% dengan AUC 0,964 (*Area Under Curve*) dan KNN menghasilkan akurasi 88,50% dengan AUC 0,948. Jika kedua perbandingan ditambahkan optimasi *Particle Swarm Optimization* (PSO) maka hasilnya akurasi SVM+PSO 92.75% dengan AUC 0.973 dan akurasi KNN+PSO 75.25% dengan AUC 0.768. Dari data tersebut akurasi SVM menggunakan optimasi terbukti lebih baik dalam pengklasifikasian data *tweet*. [9].

2. Metode Penelitian

Berikut merupakan alur dari penelitian ini.



Gambar 2. Alur Penelitian

Tahapan dan teori-teori yang mendasari penelitian ini adalah sebagai berikut:

2.1. Data Mining

Data *mining* adalah proses mencari sumber informasi untuk sebuah data. Data informasi ini bisa didapatkan dengan menggunakan teknik yang kompleks seperti kecerdasan buatan (AI), metode statistik, pembelajaran mesin (*machine learning*), matematika, dan lainnya. Teknik yang disebutkan nantinya berguna untuk melihat informasi dari *database* dengan volume yang besar. Data *mining* berasal dari metode pembelajaran mesin dan statistik, tetapi bisa digunakan juga untuk bidang ilmu komputer dan disiplin ilmu lainnya [11].

2.2. Analisis Sentimen

Analisis Sentimen yaitu proses menemukan sentimen (pendapat/pandangan) dari suatu kalimat atau teks apakah berperasaan negatif, positif atau netral. Pendekatan ini juga digunakan untuk mengetahui pandangan publik terhadap isu-isu, topik, kebijakan dan pelayanan berdasarkan data tekstual [12]. Analisis sentimen memiliki tahapan dalam pengolahan datanya yaitu: [12].

- 1) Pengumpulan data, proses awal untuk mendapatkan dan mengumpulkan data yang akan dianalisis.
- 2) *Preprocessing*, proses untuk memperoleh data yang sesuai dengan ketentuan klasifikasi. Seperti penghapusan data dari *noise*, menyamakan bentuk kata dan mengurangi ukuran data untuk memperoleh kumpulan data yang cocok dipakai saat pengklasifikasian.
- 3) Klasifikasi, proses untuk mengidentifikasi pola sentimen (pandangan atau pendapat) dari pesan teks yang sudah melalui proses *preprocessing*.
- 4) Evaluasi performa, proses untuk menghitung *accuracy*, presisi, *recall* dan *f-measure* atau *F1-score*.

2.3. Ulasan Pengguna

Ulasan atau *Review* merupakan kalimat atau teks berisikan penilaian dan komentar yang diberikan untuk hasil karya seseorang, ulasan bisa dimanfaatkan untuk dijadikan masukan untuk perbaikan [1].

2.4. Web Scraping

Web Scraping merupakan teknik otomatisasi untuk mengekstrak data dari sebuah website, database, atau sistem *legacy* yang belum terstruktur, kemudian hasilnya disimpan berupa file dengan format *comma-separated values* (CSV). Data yang digunakan untuk penelitian ini adalah data sekunder dari data ulasan yang diberikan pengguna *Home Credit* melalui *Google Play Store*. Jumlah populasi data yang digunakan yaitu 2.845 data, yang diambil dari tanggal 28 Maret 2023 sampai 24 Mei 2023 yang berisikan data ulasan negatif dan positif. Data ini didapatkan dari hasil *Web Scraping* dari *Google Play Store*.

	A	B	C
	content	score	sentimen
1	Tdinya mau coba2 eh ditolak.. pas liat review banyak yg komplek, uninstal aja lah, tolong data2 saya jangan di salah gunakan..	1	0
2	Kok saya tidak bisa mengajukan pinjaman padahal saya baru pakai pinjaman online di home kredit ini doang.	1	0
3	Semoga ada limitnya	5	1
4	apk, gak jelas	1	0
5	Iloveeyou	5	1
6	Ok	5	1
7	Agar lebih berhati hati bagi pengguna home kredit.	1	0
8	sekarang selalu di tolak kalo ngajuin cicilan, terakhir ngambil AC lancar sampe lunas dulu pun pernah pembiayaan multiguna 10jt lancar history ngambil barang lancar laptop, sepeda, hape dll lancar sekarang selalu di tolak padahal limit ada di aplikasi	1	0
9	ok jg cuma.ga bs mengajukan cicilan barang.percuma .	5	1
10	Jos	5	1
11	Tiap hari diteleponin Sehari 2x pada hal g punya cicilan/sangkutan seperti diteror..apa caranya seperti ini cari nasabah ..tiap hari ditelpon Mengganggu aktifitas orang yg lagi kerja, yg slalu ditelpon terus	1	0
12	Baughs♥♥♥	5	1

Gambar 1. Dataset

Dataset yang sudah didapatkan dan sudah dibersihkan dari kolom-kolom yang tidak digunakan untuk pengklasifikasian, langkah selanjutnya yaitu pelabelan sentimen berdasarkan *score* komentar. *Score* yang diberikan kurang dari sama dengan tiga (1, 2, 3), maka hasil sentimennya berlabel negatif =, dan jika *score* yang diberikan lebih dari tiga (4, 5) maka hasil sentimennya berlabel positif [10].

2.5. Preprocessing

Preprocessing data adalah tahapan-tahapan untuk melakukan persiapan pemrosesan data yang akan dilakukan pemodelan. Pada tahapan ini dilakukan beberapa kegiatan untuk membentuk data dan juga melakukan pembersihan data sehingga data bisa diproses ke tahap selanjutnya [13]. Tahapan-tahapan ketika melakukan *preprocessing* data [14], sebagai berikut:

- 1) *Case Folding*, proses mengganti semua huruf menjadi huruf kecil.
- 2) *Cleaning*, proses menghapus angka, tanda baca, emoji, dan sebagainya.
- 3) *Tokenization*, proses untuk memotong *string* berdasarkan dari setiap kata.
- 4) *Normalize*, proses memperbaiki kata yang belum sesuai ejaan yang disesuaikan (EYD).
- 5) *Stemming*, proses mengubah kata berimbuhan menjadi bawaan kata dasar.

2.6. Term Frequency Inverse Document Frequency

Term Frequency-Inverse Document Frequency adalah teknik perhitungan dan pemberian nilai bobot pada setiap kata dalam *dataset*, berfungsi untuk menunjukkan pentingnya setiap kata dalam *dataset* tersebut setelah dilakukan proses *preprocessing* seperti *tokenize*, *stemming* dan frekuensi munculnya kata dalam dokumen [12].

2.7. Machine Learning

Machine learning adalah teknologi kecerdasan buatan (AI) yang memungkinkan sistem untuk belajar secara otomatis dari pengalaman dan menjadi lebih baik tanpa perlu dirancang secara eksplisit. *Machine learning* memberikan perspektif baru dan membantu dalam pemecahan masalah bagi organisasi penelitian [15].

2.8. Support Vector Machine

Support Vector Machine (SVM) adalah teknik yang digunakan untuk melakukan prediksi, baik dalam konteks regresi maupun klasifikasi. Untuk membedakan antara dua kelas yang berbeda, prinsip kerja SVM adalah memilih *hyperplane* terbaik dengan margin maksimum yang diizinkan [8]. Tujuan SVM yaitu mencari *hyperplane* yang optimal. *Hyperplane* yang bisa membagi kedua class dengan jarak margin terjauh antar kelas. Margin adalah jarak antara *hyperplane* dan pola masing-masing class. *Instance* terdekat disebut *support vector*. Di bawah adalah contoh *hyperplane* dengan keterangan garis merah yang di atas garis hitam tebal dengan tanda “+” sebagai *support vector* kelas *men*, dan garis merah di bawah garis hitam tebal dengan tanda “o” sebagai *support vector* kelas *women* [16].

2.9. K-Nearest Neighbor

K-Nearest Neighbor (k-NN/KNN) adalah sebuah teknik atau metode dalam melakukan klasifikasi terhadap objek yang berdasar pada data pembelajaran (*neighbor*) yang jaraknya paling dekat dengan objek tersebut. Dekat atau jauhnya *neighbor* dihitung berdasarkan jarak *Euclidean* [7]. Metode KNN bekerja dengan mencari pola K yang terdekat kemudian menentukan kelas keputusan yang berdasar pada jumlah pola terbanyak. Prosesnya menghasilkan K dengan akurasi tertinggi dalam menggeneralisasi data yang akan datang. Masalahnya, k ini tidak dapat ditentukan secara matematik, sehingga proses pelatihannya dengan observasi terhadap sebanyak k sampai mendapatkan hasil k yang paling optimum [16].

2.10. K-fold cross validation

K-Fold Cross Validation adalah salah satu teknik pengujian data yang melipat gandakan data menjadi bagian-bagian n dan membaginya menjadi bagian-bagian n dengan jumlah yang sama (misalnya, bagian K, L, M, dan N), masing-masing memiliki data yang berbeda, kemudian diterapkan pada data [12].

K-Fold Cross Validation merupakan suatu metode dari *Cross Validation* yang banyak digunakan, penggunaannya metodenya dengan cara melipat data sebanyak k dan mengulangi (iterasi) eksperimennya sebanyak k [17].

2.11. Confusion matrix (Evaluasi)

Metode yang digunakan untuk mengevaluasi seberapa akurat metode klasifikasi dalam memprediksi kelas data di mana teknik ini menghasilkan perbedaan nilai kelas aslinya dengan nilai prediksi. Pada *confusion matrix* dilakukan perhitungan yaitu, Akurasi, Presisi, *Recall*, dan *F-measure* [12].

3. Hasil dan Pembahasan

3.1. Uji Model

Data yang digunakan berupa ulasan pengguna aplikasi *Home Credit* pada kolom *review* di *Google Play Store*. Pengambilan data ini dilakukan dengan teknik *Scraping* menggunakan *Google Colab* dan bahasa pemrograman *python*, dan hasil *scraping* ini diimpor menjadi file dengan ekstensi CSV (*Comma Separated Values*) dan XLSX. Prosesnya dijalankan secara otomatis dengan menggunakan *library Google-play-scrapers* dari pemrograman *python* yang dijalankan di *Google Colab*, prosesnya terhitung sangat cepat dibandingkan dengan mengambil data secara manual. data yang didapatkan adalah data dari tanggal 28 Maret 2023 sampai tanggal 24 Mei 2023, data ini berjumlah 2.845 (Dua ribu delapan ratus empat puluh lima).

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[2] import pandas as pd
hcredit = pd.read_csv("/content/drive/My Drive/analisis_sentimen/homecredit.csv", index_col = 0)

[3] # Ubah data tanggal menjadi data datetime
hcredit["at"] = pd.to_datetime(hcredit["at"])

[4] # Ambil hanya date-nya saja
hcredit["date"] = hcredit["at"].dt.date
hcredit["date"].head()

[5] a = pd.date_range(start = "2023-03-28", end = "2023-05-24")

from datetime import date, timedelta
import numpy as np

start_date = date(2023, 3, 28)
end_date = date(2023, 5, 24)
delta = timedelta(days = 1)
hcredit_pakai = pd.DataFrame(columns = list(hcredit.columns))
while start_date <= end_date:
    x = hcredit[hcredit["date"] == start_date]
    hcredit_pakai = pd.concat([hcredit_pakai, x])
    start_date += delta

[6] hcredit_pakai = hcredit_pakai.reset_index(drop = True)
hcredit_pakai.tail(2)
```

Gambar 3. Pengambilan Data

Setelah pengumpulan data dan pembuatan label sentimen, tahap selanjutnya yaitu *preprocessing*, gunanya agar membersihkan dan mempersiapkan data teks sebelum dianalisis lebih lanjut. Pembobotan data *tf-idf* adalah tahap setiap kata (*term*) dari seluruh dokumen yang akan digunakan untuk perhitungan *cosine similarity* akan dilakukan perhitungan bobot setiap dalam dokumen atau kalimat. Perhitungan bobot yang digunakan, yaitu *term frequency (tf)*, *inverse document frequency (idf)*, dan *tf-idf*. Perhitungan *term frequency (tf)* pada proses analisis sentimen berlangsung secara otomatis karena menggunakan bantuan *library* bahasa pemrograman, tetapi cara perhitungannya sama seperti menghitung manual. Setelah melakukan tahapan-tahapan *preprocessing* dan tahapan TF-IDF, selanjutnya *dataset* diimplementasikan ke dalam algoritma *machine learning* yaitu *K-Nearest Neighbor* dan *support vector machine learning*.

```
[ ] from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

[ ] X = content_isi['content_clean']
Y = content_isi['sentimen']

[ ] # INI BUAT YANG K-FOLD CROSS VALIDATION
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()

X = vectorizer.fit_transform(X.values.astype('U'))

print(X.shape) # Banyak kolom = Banyak kata yang ada dalam keseluruhan dokumen (content)

[ ] # Split into train and test data

train_X, test_X, train_Y, test_Y = model_selection.train_test_split(X, Y, test_size = 0.2, random_state = 0)
# random_state = 0 menyatakan tidak ada pengacakan pada data yang di split yang artinya urutannya masih sama
```

Gambar 4. Proses otomatisasi

Proses perubahan dataset menjadi numerik menggunakan library *scikit-learn* *tfidf* vectorizer secara otomatis, kemudian data akan dibagi menjadi data latih sebesar 80% dan data uji 20%, tujuannya agar penggunaan model lebih optimal dan diharapkan bisa menganalisis dengan baik. Selanjutnya dataset sudah siap untuk dilakukan evaluasi model.

3.2. Evaluasi Model

Evaluasi model dilakukan setelah proses uji model selesai dilakukan. Evaluasi model dilaksanakan untuk menghitung performa metode yang dipilih. Pada uji model yang dilakukan menghasilkan *confusion matrix* dengan dimensi 2x2, ada dua gambar *confusion matrix* yang di hasilkan dari uji model, yaitu tabel *confusion* untuk algoritma *support vector machine* dan *K-Nearest Neighbor*.

```
# Accuracy, Precision, Recall, f1-score

from sklearn.metrics import classification_report, f1_score, recall_score, precision_score, confusion_matrix

print("\nHere is the classification report: \n")
print(conf_mat)
print(classification_report(test_Y, y_pred))
# Confusion matrix SVM
tn, fp, fn, tp = confusion_matrix(test_Y, y_pred).ravel()
print("True Positive : ", tp)
# Accuracy score
print("True Negative : ", tn)
# Precision score
print("False Positive : ", fp)
# Recall score
print("False Negative : ", fn)
```

Gambar 5. Proses Evaluasi

Selanjutnya, performa metode klasifikasi setiap kelasnya dapat dilihat melalui nilai *presisi*, *recall*, *f1-score* pada setiap kelasnya. Hasil nilai *presisi*, *recall*, dan *f1-score* memiliki nilai sebesar 0 – 1. Semakin tinggi nilainya mendekati 1 maka hasilnya semakin baik.

```
Here is the classification report SVM:

[[328 13]
 [ 55 173]]

              precision    recall  f1-score   support

     0       0.86      0.96      0.91       341
     1       0.93      0.76      0.84       228

 accuracy          0.88          569
 macro avg         0.89          569
 weighted avg      0.89          569

 True Positive : 173
 True Negative : 328
 False Positive : 13
 False Negative : 55
```

Gambar 6. Hasil *presisi*, *recall*, *accuracy*, *f1-score* SVM

Terdapat hasil *True Positive* 173, *True Negative* 328, *False Positive* 13, *False Negative* 55. Setelah mengetahui hasil confusion matrix dari data uji SVM, selanjutnya dilakukan perhitungan untuk mencari *accuracy*:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

$$accuracy = \frac{173 + 328}{173 + 328 + 55 + 13}$$

$$accuracy = \frac{501}{569} = 0,8804$$

$$accuracy = 0,88 \times 100\% = 88,00\%$$

```
Here is the classification report:

[[253 88]
 [ 33 195]]

              precision    recall  f1-score   support

     0       0.88      0.74      0.81       341
     1       0.69      0.86      0.76       228

 accuracy          0.79          569
 macro avg         0.79          569
 weighted avg      0.81          569

 True Positive : 195
 True Negative : 253
 False Positive : 88
 False Negative : 33
```

Gambar 6. Hasil *presisi*, *recall*, *accuracy*, *f1-score* KNN

Hasil yang diperoleh *True Positive* 195, *True Negative* 253, *False Positive* 33, *False Negative* 88. Setelah mengetahui hasil confusion matrix dari data uji KNN, selanjutnya dilakukan perhitungan untuk mencari *accuracy*:

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (2)$$

$$accuracy = \frac{195 + 253}{195 + 253 + 88 + 33}$$

$$accuracy = \frac{448}{569} = 0,7873$$

$$accuracy = 0,79 \times 100\% = 79.00\%$$

Dari perhitungan SVM dan KNN di atas, nilai akurasi yang diperoleh adalah 0.7873 untuk *K-NN* dan 0.8804 untuk SVM, yang dihitung berdasarkan nilai dari diagonal *confusion matrix* dibagi dengan jumlah seluruh data.

Tabel 1. Perbandingan *accuracy*

Metode	Akurasi
<i>Support Vector machine (K-Fold CV+kernel linear)</i>	88.00 %
<i>K-Nearest Neighbor (K-Fold CV+K=3)</i>	79.00 %

Berdasarkan Tabel 1, metode SVM yang digunakan mendapatkan skor lebih dari 80%, sedangkan untuk KNN mendapatkan hasil kurang dari 80%, hasil ini bisa dikategorikan metode KNN kurang baik dalam melakukan klasifikasi pada penelitian ini. Kemudian untuk perbandingannya *Support Vector Machine* dengan *kernel Linear* mendapatkan *Accuracy* 88.00% ini merupakan hasil yang lumayan baik dalam pengklasifikasian. Hasil yang didapatkan sesudah melakukan analisis sentimen dengan bantuan machine learning yaitu, terjarungnya konten atau komentar, berdasarkan pemberian sentimen negatif dan positif. Diketahuinya jumlah score dan sentimen yang didapatkan dari tanggal 28 Maret – 24 Mei 2023.

4. Kesimpulan

Setelah dilakukannya pengujian algoritma *Support Vector Machine* dan *K-Nearest Neighbor* ada beberapa hal yang dihasilkan. Sistem penganalisa sentimen yang digunakan sudah cukup baik dalam melakukan tugasnya mengidentifikasi sentimen positif dan negatif. Hasil dari penelitian ini yaitu nilai akurasi 0.88 atau 88.0% untuk SVM dan 0.79 atau 79.0% untuk KNN. Hasil ini diketahui bahwa SVM lebih baik dalam proses pengklasifikasian ini. Kelebihan penelitian ini adalah menghasilkan dan memiliki nilai akurasi, presisi dan recall yang tinggi, sehingga sudah bisa digunakan dalam sebuah sistem. Kekurangan penelitian ini adalah nilai antar presisi dan recall tidak merata, hal ini disebabkan karena jumlah data sentimen positif dan negatif tidak sama.

Referensi

- [1] N. Herlinawati *et al.*, “Analisis Sentimen Zoom Cloud Meetings Di Play Store Menggunakan Naïve Bayes Dan Support Vector Machine,” *Journal of Computer Engineering System and Science*, vol. 5, no. 2, pp. 2502–714, 2020.
- [2] E. Syahputra, “Bertahan saat-pandemi pengguna-home credit tembus 98 juta,” CNBC Indonesia. Accessed: Jun. 07, 2023. [Online]. Available: <https://www.cnbcindonesia.com/news/202110141613364-283946/bertahan-saat-pandemi-pengguna-home-credit-tembus-98-juta>
- [3] A. Muhammadin and I. A. Sobari, “Analisis Sentimen Pada Ulasan Aplikasi Kredivo Dengan Algoritma SVM Dan NBC,” *Jurnal Rekayasa Perangkat Lunak*, vol. 2, no. 2, 2021, [Online]. Available: <http://jurnal.bsi.ac.id/index.php/reputasi>
- [4] 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>
- [5] M. Rangga, A. Nasution, and M. Hayaty, “Perbandingan Akurasi dan Waktu Proses Algoritma K-NN dan SVM dalam Analisis Sentimen Twitter,” *JURNAL INFORMATIKA*, vol. 6, no. 2, pp. 212–218, 2019, [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- [6] 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,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 8, pp. 3987–3994, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id>

- [7] I. A. Angreni, S. A. Adisasmita, M. I. Ramli, and S. Hamid, "Pengaruh Nilai K Pada Metode K-Nearest Neighbor (Knn) Terhadap Tingkat Akurasi Identifikasi Kerusakan Jalan," *Rekayasa Sipil*, vol. 7, no. 2, p. 63, Jan. 2019, doi: 10.22441/jrs.2018.v07.i2.01.
- [8] M. N. Muttaqin and I. Kharisudin, "Analisis Sentimen Pada Ulasan Aplikasi Gojek Menggunakan Metode Support Vector Machine dan K Nearest Neighbor," *UNNES Journal of Mathematics*, vol. 10, no. 2, pp. 22–27, 2021, [Online]. Available: <http://journal.unnes.ac.id/sju/index.php/ujm>
- [9] A. Saepudin, R. Aryanti, and E. Fitriani, "Optimasi Algoritma SVM Dan K-NN Berbasis Particle Swarm Optimization Pada Analisis Sentimen Fenomena Tagar #2019GantiPresiden," *Jurnal Teknik Komputer AMIK BSI*, 2020, doi: 10.31294/jtk.v4i2.
- [10] I. S. I. Kumala, Y. A. Mustofa, and I. A. Salihi, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura Journal of Electrical and Electronics Engineering*, vol. 5, no. 1, pp. 32–35, Dec. 2023, doi: 10.1177/0165551510388123.
- [11] I. Iwandini, A. Triayudi, and G. Soepriyono, "Sawo Manila No.61, RW.7, Pejaten Bar," *Journal of Information System Research*, vol. 4, no. 2, 2023, doi: 10.47065/josh.v4i2.2937.
- [12] H. Aulia, N. Syifa, A. Nugroho, R. Firliana, P. Algoritma, and N. Bayes, "Habibi Aulia Nur Syifa Perbandingan Algoritma Naïve Bayes Classifier Dan K-Nearest Neighbors Untuk Analisis Sentimen Covid-19 Di Twitter," *Jurnal Ilmiah Informatika*, 2023.
- [13] S. Chohan, A. Nugroho, A. Maezar Bayu Aji, W. Gata, and S. Nusa Mandiri, "Analisis Sentimen Aplikasi Duolingo Menggunakan Metode Naïve Bayes dan Synthetic Minority Over Sampling Technique," *Jurnal Informatika dan Komputer*, vol. 22, no. 2, 2020, doi: 10.31294/p.v21i2.
- [14] D. S. Putri, A. Sentimen, U. Aplikasi, and T. Ridwan, "Analisis Sentimen Ulasan Aplikasi Pospay dengan Algoritma Support Vector Machine," *Jurnal Ilmiah Informatika*, 2023.
- [15] K. Anwar, "KLIK: Kajian Ilmiah Informatika dan Komputer Analisa sentimen Pengguna Instagram Di Indonesia Pada Review Smartphone Menggunakan Naive Bayes," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 2, no. 4, pp. 148–155, 2022, [Online]. Available: <https://djournals.com/klik>
- [16] F. S. Pamungkas, B. D. Prasetya, and I. Kharisudin, "Perbandingan Metode Klasifikasi Supervised Learning pada Data Bank Customers Menggunakan Python," *PRISMA, Prosiding Seminar Nasional Matematika*, vol. 3, pp. 689–694, 2019, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/>
- [17] R. Fajar, S. Program, P. Rekayasa, N. Lunak, and R. Bengkalis, "Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter," *Jurnal Inovtek Polbeng-Seri Informatika*, vol. 3, no. 1, 2018.