JNATIA Volume 3, Nomor 2, Februari 2025 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Analisis Sentimen Ulasan Aplikasi Citilink menggunakan Metode Support Vector Machine dengan TF-IDF

David Brave Moarota Zebua[a1], Ida Bagus Gede Dwidasmara[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]davidzebua8@gmail.com
[2]dwidasmara@unud.ac.id

Abstract

In line with the advancement of the Industry 4.0 era, Indonesian society has been living side by side and is inseparable from the existing technological advancements. One of the conveniences experienced by today's society is that transactions no longer need to be conducted face-to-face in a particular place but can now be done online. In the context of air transportation, technological advancements have been very helpful to the public. Airline applications are one of the most widely used by passengers. In this study, the researchers focused on analyzing public sentiment towards the Citilink application, one of Indonesia's leading airlines. The researchers used the Support Vector Machine (SVM) method enhanced with TF-IDF (Term Frequency-Inverse Document Frequency) text representation to analyze sentiment from user reviews. The stages of this research began with data collection containing reviews from the Citilink application to analyze its sentiment. Then, it proceeded to the data preprocessing stage, where the collected data was cleaned until it became tokens ready for testing. After that, it moved to the weighting stage using Term Frequency-Inverse Document Frequency (TF-IDF). Then it continued to the stage of applying the Support Vector Machine (SVM) model. The last one is the evaluation to measure the accuracy level of the model used. Based on the results of this study, it can be concluded that the Support Vector Machine model that has been adapted to the dataset of Citilink application reviews from Google Playstore and supported by TF-IDF feature extraction successfully classified the sentiment of reviews with high accuracy, reaching 88%. Further evaluation also showed satisfactory values of precision, recall, and F1-Score, namely 90%, 83%, and 85%, respectively. This study shows that the Support Vector Machine model can be an effective instrument in understanding user responses to the performance of the Citilink application.

Keywords: *Sentiment Analysis, Citilink, Support Vector Machine, TF-IDF, Confusion Matrix*

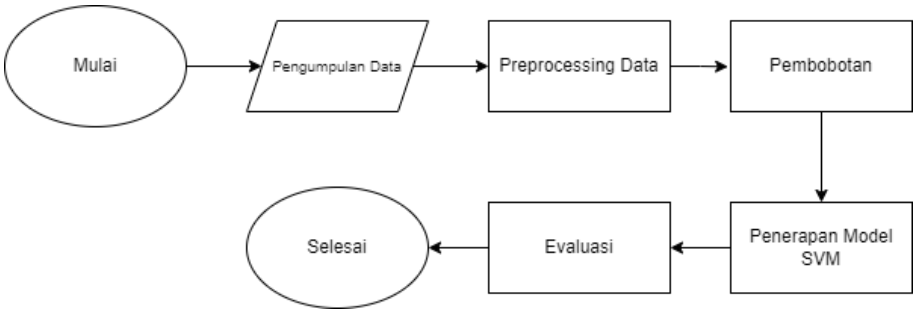
1. Pendahuluan

Seiring dengan perkembangan era industri 4.0, masyarakat Indonesia telah hidup berdampingan dan sudah tidak bisa dipisahkan lagi dengan kemajuan teknologi yang ada. Disamping itu, pemanfaatan teknologi yang sangat maju memberikan kemudahan dalam berbagai sektor kehidupan manusia, baik untuk melaksanakan kewajiban, maupun memenuhi kebutuhan. Adapun kemudahan yang dirasakan bagi masyarakat saat ini, dimana sekarang transaksi tidak perlu bertemu secara langsung dalam suatu tempat melainkan sekarang sudah bisa dilakukan secara online [1]. Kemudahan-kemudahan ini mempengaruhi cara masyarakat berperilaku dan memaksa kita untuk terus beradaptasi. Dalam konteks transportasi udara, adanya kemajuan teknologi sangat membantu masyarakat. Semuanya bermula dari munculnya peningkatan yang terjadi pada penerbangan domestik di Indonesia sehingga memunculkan ide untuk mengembangkan sebuah aplikasi online berbasis android yang berfungsi untuk memesan tiket pesawat [2]. Mulai dari pemesanan tiket secara *online* hingga memberikan informasi penerbangan secara *real-time* terus menerus dikembangkan. Aplikasi maskapai penerbangan menjadi salah satu yang paling banyak digunakan oleh penumpang. Di tengah persaingan yang

ketat di industri penerbangan, pendapat masyarakat terhadap aplikasi suatu maskapai dapat menjadi kunci keberhasilan maskapai tersebut. Dalam penelitian ini, peneliti berfokus dalam analisis sentimen pendapat masyarakat terhadap aplikasi Citilink, salah satu maskapai penerbangan terkemuka di Indonesia. Peneliti menggunakan metode Support Vector Machine (SVM) yang diperkuat dengan representasi teks TF-IDF (Term Frequency-Inverse Document Frequency) untuk menganalisis sentimen dari ulasan pengguna. Tujuan dari penelitian ini adalah untuk mengembangkan model yang dapat mengklasifikasikan ulasan pengguna menjadi kategori sentimen positif maupun negatif, sehingga maskapai penerbangan dapat merespons dengan cepat terhadap umpan balik pelanggan dan meningkatkan kualitas layanan mereka.

2. Metode Penelitian

2.1 Desain Penelitian



Gambar 1. Bagan Alur Penelitian

Tahap pertama dari penelitian ini yaitu pengumpulan data yang berisi ulasan dari aplikasi Citilink untuk menganalisis sentimennya. Kemudian, masuk ke tahap preprocessing data, dimana data-data yang sudah dikumpulkan dibersihkan hingga menjadi token yang siap diuji. Setelah itu, masuk ke tahap pembobotan menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF). Kemudian lanjut ke tahap penerapan model *Support Vector Machine* (SVM) . Yang terakhir yaitu evaluasi untuk mengukur tingkat akurasi model yang digunakan.

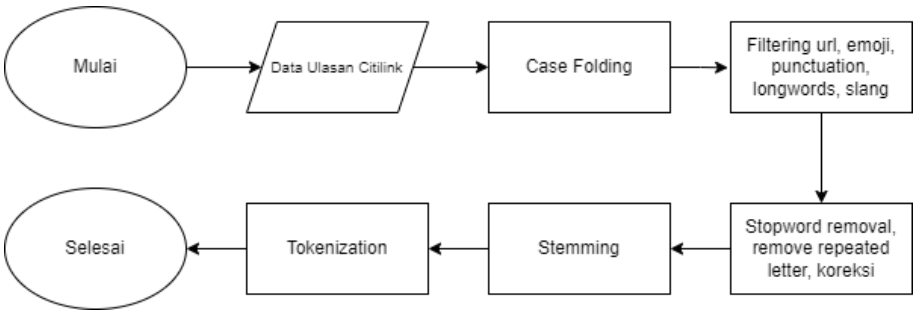
2.2 Pengumpulan Data

Teknik pengumpulan data yang digunakan pada penelitian ini yaitu menggunakan data sekunder yang diambil melalui *Kaggle*. Dataset yang digunakan merupakan dataset ulasan aplikasi Citilink yang terdapat di Google Playstore, dimana terdapat sebanyak 1471 data. Pada data *Kaggle* terdapat *score* yang dimana dapat diasumsikan score 1-3 merupakan ulasan buruk dan *score* 4-5 merupakan ulasan baik, namun tidak terdapat label yang menentukan sentimen.

Tabel 1. Tabel Gambaran Dataset

Content	Score
Makin hari pelayanan makin buruk,tiket makin mahal,,m	1
Worst airplan ever....	1
Tidak bisa mngajukan refund, saat maskapai merubah jdwal penerbangan	1
Tidak dapat mengubah destinasi Banyak bug	1
Mantap sih aplikasinya	5

2.3 Text Preprocessing



Gambar 2. Bagan Alur Tahap Preprocessing

Pada tahap preprocessing data, tahap pertama yang dilakukan yaitu melakukan *case folding* dimana semua huruf yang ada diubah menjadi huruf kecil. Tahap selanjutnya ialah filtering yang berfungsi untuk menghilangkan URL, simbol, emoji, nomor, kata panjang, dan juga *slang* yang ada pada dataset. Kemudian, masuk ke tahap *stopword removal* untuk menghapus *stopword*, mengahpus huruf yang berulang, dan juga koreksi beberapa kesalahan penulisan maupun singkatan. Selanjutnya masuk ke tahap *stemming* terlebih dahulu untuk mengubah tiap kata menjadi kata dasar, dan yang terakhir yaitu tokenisasi untuk mengubah data menjadi token-token terpisah.

2.4 Term-Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF merupakan metode yang umum digunakan untuk merepresentasikan dokumen teks sebagai vektor numerik. Vektor ini kemudian dapat digunakan dalam berbagai tugas pemrosesan bahasa alami (NLP), seperti klasifikasi teks, pengelompokan teks, dan analisis sentimen. *Term Frequency* yaitu frekuensi kemunculan term i pada dokumen j dibagi dengan total term pada dokumen j, dituliskan dalam bentuk [3],

$$tf_{ij} = \frac{f_d(i)}{\underset{j \in d}{max} f_d(j)} \quad (1)$$

Inverse Document Frequency berfungsi mengurangi bobot suatu term jika kemunculannya banyak tersebar diseluruh dokumen, dituliskan dalam bentuk [4],

$$IDF(w) = log\left(\frac{N}{DF(w)}\right) \quad (2)$$

2.5 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu algoritma *machine learning* yang memiliki kinerja sangat baik dalam mengklasifikasikan data. Hal yang luar biasa dari SVM ini yaitu dapat digunakan tanpa bergantung pada dimensi dari ruang fitur [5]. SVM mengukur kompleksitas hipotesis berdasarkan margin yang memisahkan bidang dan bukan jumlah fitur. SVM juga dikatakan sebagai pengklasifikasi linear yang didasarkan pada prinsip memaksimalkan margin. SVM menggunakan *hyperplane* secara optimal untuk mengklasifikasikan data ke dalam dua kelompok di ruang dimensi yang lebih tinggi. Margin adalah jarak antara *hyperplane* dan data terdekat dari setiap kelas. Data terdekat ini disebut vektor pendukung. *Hyperplane* adalah pemisah terbaik antara dua kelas yang telah ditentukan sebelumnya [6]. Prinsip dasar dari SVM adalah pengklasifikasi linear, dan kemudian dikembangkan agar dapat bekerja pada masalah non-linear, yaitu dengan menggabungkan konsep trik kernel dalam ruang kerja berdimensi tinggi.

2.6 Confusion Matrix

Confussion matrix dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan

analisis apakah *classifier* tersebut baik dalam mengenali *tuple* dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi kepada *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika classifier salah dalam melakukan klasifikasi data [7]. Berikut merupakan rumus untuk melakukan perhitungan *confusion matrix* .

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 - Score = \frac{2.Precision.Recall}{Precision+Recall} \quad (6)$$

3. Hasil dan Pembahasan

3.1 Preprocessing Data

Jumlah data yang akan melalui tahap *preprocessing* terdiri dari 974 data positif, dan 497 data negatif. Tahap pertama yaitu *Case Folding*, yang dilakukan untuk mengkonversi semua huruf dalam dataset menjadi huruf kecil. Kemudian terdapat penghapusan URL, simbol, emoji, nomor, kata berukuran panjang, dan bahasa gaul. Selanjutnya dilakukan penghapusan *stopwords* dengan *library* Sastrawi untuk menghapus *stopwords* berbahasa Indonesia, juga koreksi terhadap pengulangan huruf serta kesalahan penulisan. Kemudian yaitu tahap stemming yang dilakukan dengan *library* Sastrawi untuk mengubah setiap kata menjadi bentuk dasarnya. Tahap terakhir yaitu tokenisasi dimana hasil *stemming* dibagi menjadi token per kata Adapun contoh data yang telah melewati tahap *preprocessing* terdapat pada tabel 2.

Tabel 2. Contoh Data Preprocessing

Content	Case folding	Cleaned	Stemmed	Token
Tidak bisa mngajukan refund, saat maskapai merubah jdwal penerbangan	tidak bisa mngajukan refund, saat maskapai merubah jdwal penerbangan	tidak bisa mengajukan refund maskapai merubah jadwal penerbangan	tidak bisa aju refund maskapai rubah jadwal terbang	[‘tidak’ ‘bisa’ ‘aju’ ‘refund’ ‘maskapai’ ‘rubah’ ‘jadwal’ ‘terbang’]

3.2 Perhitungan Ekstraksi Fitur Term Frequency-Inverse Document Frequency (TF-IDF)

Sesudah tahap *preprocessing* dilakukan, langkah berikutnya yaitu perhitungan ekstraksi fitur TF-IDF yang dimana data akan diubah menjadi vektor numerik. Ekstraksi fitur ini dilakukan menggunakan *library scikit-learn*. Pertama-tama, *TfidfVectorizer* akan mempelajari semua kata unik (vocabulary) yang muncul di seluruh data teks. Kemudian *TfidfVectorizer* akan menghitung berapa kali setiap kata (term) muncul dalam dokumen tersebut. Setelah itu, *TfidfVectorizer* akan menghitung seberapa umum kata tersebut muncul di seluruh kumpulan dokumen, dimana kata yang jarang muncul di banyak dokumen akan memiliki nilai IDF yang lebih tinggi. Kemudian *TfidfVectorizer* akan menggabungkan nilai TF dan IDF untuk setiap kata dalam setiap dokumen, menghasilkan skor TF-IDF. Terakhir, *TfidfVectorizer* mengubah setiap dokumen menjadi vektor numerik yang berisi skor TF-IDF untuk semua kata dalam vocabulary. Dokumen yang serupa akan memiliki vektor yang lebih mirip karena kata-kata penting yang sama akan memiliki skor TF-IDF yang tinggi di kedua dokumen.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

```
tfidf_vectorizer = TfidfVectorizer()
x_train_tfidf = tfidf_vectorizer.fit_transform(x_train)
x_test_tfidf = tfidf_vectorizer.transform(x_test)
```

Gambar 3. Implementasi Ekstraksi Fitur Term Frequency-Inverse Document Frequency

3.3 Pemodelan Support Vector Machine

Setelah ekstraksi fitur TF-IDF, langkah selanjutnya yaitu menerapkan metode klasifikasi SVM. Metode ini merupakan salah satu metode unggulan dalam klasifikasi teks. Sebelum melakukan pemodelan, terlebih dahulu dilakukan *splitting* data menjadi data latih dan juga data uji dengan perbandingan data uji sebesar 30%, dan data latih sebesar 70% yang dapat dilihat pada gambar 4.

```
x_train, x_test, y_train, y_test = train_test_split(df['token'], df['sentiment'], test_size=0.3, random_state=42)

print(f'jumlah data latih: {len(x_train)}')
print(f'jumlah data uji: {len(x_test)}')

jumlah data latih: 1029
jumlah data uji: 442
```

Gambar 4. Splitting Data Latih dan Data Uji

Kemudian pemodelan data *Support Vector Machine* dilakukan dengan *library sci-kit learn*. Kali ini, fungsi kernel yang akan digunakan oleh SVM yaitu linear. Dalam hal ini, kernel linear dipilih karena kernel linear berguna untuk data yang dapat dipisahkan secara linier dalam ruang fitur. Kemudian masing-masing nilai *hyperparameter* C, *intercept*, dan juga *support vector* dicetak untuk mengontrol *trade-off* antara kerentanan model terhadap *overfitting* dan margin antara kelas serta membantu dalam pembuatan keputusan klasifikasi.

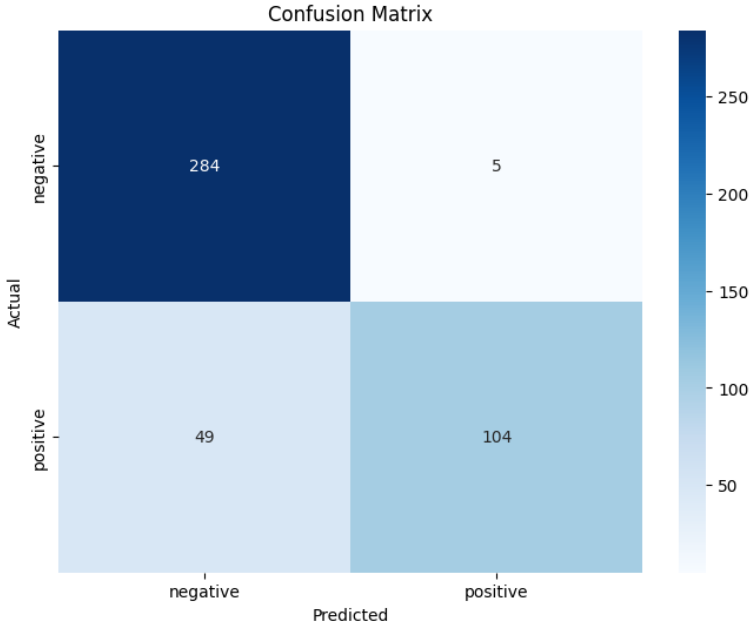
```
svm_model = SVC(kernel='linear', random_state=42)

svm_model.fit(x_train_tfidf, y_train)

print("Parameter model SVM:")
print(f"Kernel: {svm_model.kernel}")
print(f"C: {svm_model.C}")
print(f"Intercept: {svm_model.intercept_}")
print(f"Support Vectors: {svm_model.support_vectors_}")

Parameter model SVM:
Kernel: linear
C: 1.0
Intercept: [-0.01213951]
Support Vectors:   (0, 147)     0.11840844543151888
  (0, 177)      0.2192686062728066
  (0, 202)      0.16036921698949588
  (0, 215)      0.1413647744818176
  (0, 462)      0.13541252856014313
  (0, 482)      0.2192686062728066
```

Gambar 5. Pelatihan ModelSupport Vector Machine dan Sebagian Hasil Cetaknya

Setelah melatih model Support Vector Machine, langkah selanjutnya yaitu menguji model

p-ISSN: 2986-3929
e-ISSN: 3032-1948

menggunakan data uji yang telah diproses dengan ekstraksi fitur TF-IDF sebelumnya. Setelah melakukan prediksi terhadap data uji, hasil prediksi akan dibandingkan dengan label aslinya dari data uji yang akan menghasilkan *confusion matrix*. *Confusion matrix* adalah tabel uji yang menyajikan jumlah prediksi benar dan salah yang telah dibuat oleh model klasifikasi pada setiap kelas.



Gambar 6. Confusion Matrix Bag-of-Words

Berdasarkan *confusion matrix* pada gambar 6, model SVM berhasil memprediksi sebanyak 284 sentimen negatif, namun terdapat 5 data bersentimen negatif yang diprediksi positif. Model ini juga berhasil memprediksi sebanyak 104 sentimen positif, namun terdapat 49 sentimen positif yang diprediksi negatif. Berdasarkan confusion matrix diatas didapatkan nilai *Accuracy*, *Precision*, F1-*Score* pada model Support Vector Machine sebagai berikut:

Classification Report:

	precision	recall	f1-score	support
negative	0.85	0.98	0.91	289
positive	0.95	0.68	0.79	153
accuracy			0.88	442
macro avg	0.90	0.83	0.85	442
weighted avg	0.89	0.88	0.87	442

Gambar 7. Hasil Evaluasi

Dari hasil akurasi yang didapatkan model Support Vector Machine yang telah dilatih dengan ekstraksi fitur TF-IDF, didapatkan nilai *accuracy* sebesar 88.%, nilai precision 90%, nilai recall 83%, dan juga nilai F1-Score 85%

4. Kesimpulan

Dari hasil penelitian ini, dapat disimpulkan bahwa model Support Vector Machine yang telah disesuaikan dengan dataset ulasan aplikasi Citilink dari Google Playstore dan ditunjang dengan ekstraksi fitur TF-IDF berhasil mengklasifikasikan sentimen ulasan dengan akurasi yang tinggi, mencapai 88%. Evaluasi lanjutan juga menunjukkan nilai *precision*, *recall*, dan F1-*Score* yang memuaskan, yaitu masing-masing sebesar 90%, 83%, dan 85%. Penelitian ini menunjukkan bahwa model Support Vector Machine dapat menjadi instrumen yang efektif dalam memahami tanggapan pengguna terhadap performa aplikasi Citilink.

Daftar Pustaka

[1] D. Septiansari and T. Handayani, “Pengaruh Belanja Online Terhadap Perilaku Konsumtif pada Mahasiswa di Masa Pandemi Covid-19,” *Jurnal Ekonomi dan Manajemen Teknologi*, vol. 5, no. 1, pp. 53–65, 2021, doi: 10.35870/emt.v5i1.372

[2] A. Tristiaratri, A. H. Brata, and L. Fanani, “Perbandingan User Interface Aplikasi Mobile Pemesanan Tiket Pesawat Online dengan Design Thinking,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 6, pp. 2113-2120, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1509/550

[3] M. Yunus, “TF-IDF (Term Frequency-Inverse Document Frequency) : Representasi Vector Data Text,” Medium, 30 April 2020, [Online]. Tersedia: https://yunusmuhammad007.medium.com/tf-idf-term-frequency-inverse-document-frequency-representasi-vector-data-text-2a4eff56cda

[4] R. Kosasih and A. Alberto, “Sentiment analysis of game product on shopee using the TF-IDF method and naive bayes classifier,” *ILKOM Jurnal Ilmiah*, vol. 13, no. 2, pp. 101–109, 2021, doi: 10.33096/ilkom.v13i2.721.101-109

[5] G. Patil, V. Galande, V. Kekan et al., “Sentiment Analysis Using Support Vector Machine,” *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 2607-2612, 2014, [Online]. Available: https://www.semanticscholar.org/paper/Sentiment-Analysis-Using-Support-VectorMachine-Patil-Gal/759f15464a7ad372cba9c38a0f8c5caff6d85cf1

[6] Styawati, A. R. Isnain, N. Hendrastuty et al., “Comparison of Support Vector Machine and Naïve Bayes on Twitter Data Sentiment Analysis,” *Jurnal Informatika: Jurnal pengembangan IT (JPIT),* vol. 6, no. 1, pp. 56-60, 2021, doi: 10.30591/jpit.v6i1.3245

[7] A. Mulyani, D. Kurniadi, M. R. Nashrulloh et al., “The Prediction Of Ppa And Kip-Kuliah Scholarship Recipients Using Naive Bayes Algorithm,” *Jurnal Teknik Informatika (JUTIF),* vol. 3, no. 4, pp. 821-827, 2022, doi: 10.20884/1.jutif.2022.3.4.297

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong