

Analisis Sentimen Ulasan Aplikasi Perbankan di Google Play Store menggunakan Algoritma Support Vector Machine

Sentiment Analysis of Banking Application Reviews on Google Play Store using Support Vector Machine Algorithm

¹Martinus Juan Prasetyo*, ²I Made Artha Agastya

^{1,2}Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta

^{1,2}Jl. Ring Road Utara, Condongcatur, Depok, Sleman, Yogyakarta, Indonesia, 55281

*e-mail: juanprasetyo000@gmail.com

(*received*: 28 August 2024, *revised*: 12 September 2024, *accepted*: 16 October 2024)

Abstrak

Aplikasi perbankan semakin penting dalam memfasilitasi transaksi keuangan sehari-hari. Namun, untuk memastikan kualitas layanan, pengembang perlu memahami umpan balik pengguna. Ulasan di Google Play Store memberikan wawasan penting terkait kepuasan, keluhan, dan saran. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan model analisis sentimen ulasan aplikasi perbankan menggunakan Support Vector Machine (SVM). Data yang dikumpulkan dari tiga bank populer di Indonesia digunakan untuk melatih dan menguji model. Penelitian ini juga berkontribusi dalam menyediakan dataset multi bank yang dapat menjadi *benchmark*. Berbagai skenario pembagian data latih dan uji dieksplorasi, serta pengujian berulang dilakukan dengan nilai *random state* yang berbeda untuk mendapatkan hasil yang stabil. Hasil penelitian menunjukkan bahwa model SVM mampu mencapai akurasi yang baik, dengan dataset BRI Mobile mencapai akurasi tertinggi sebesar 92,97%, diikuti oleh dataset Gabungan 90,05%, BCA Mobile 89,73%, dan Livin Mandiri 87,46%. Ulasan negatif didominasi oleh keluhan teknis, sementara ulasan positif menyoroti kemudahan dan keandalan aplikasi. Penelitian ini menunjukkan bahwa pendekatan yang digunakan berhasil menghasilkan performa yang kompetitif, dan pengembang aplikasi disarankan untuk fokus pada peningkatan aspek teknis, seperti memperbaiki masalah login, verifikasi, dan transaksi, guna meningkatkan kepuasan pengguna.

Kata kunci: aplikasi perbankan, sentimen analisis, *support vector machine*, ulasan pengguna

Abstract

Banking applications are increasingly important in facilitating daily financial transactions. However, to ensure service quality, developers need to understand user feedback. Reviews on the Google Play Store provide important insights related to satisfaction, complaints, and suggestions. Therefore, this study aims to develop the Sentiment Analysis Model for Banking Application Reviews Using Support Vector Machine (SVM). Data collected from three popular banks in Indonesia is used to train and test models. This research also contributes to providing multi -bank dataset which can be a benchmark. Various scenarios of the distribution of training and test data are explored, and repeated tests are carried out with different random state values to get stable results. The results showed that the SVM model was able to achieve good accuracy, with BRI Mobile dataset reaching the highest accuracy of 92.97%, followed by a combined dataset of 90.05%, BCA Mobile 89.73%, and Livin Mandiri 87.46%. Negative reviews are dominated by technical complaints, while positive reviews highlight the ease and reliability of the application. This study shows that the approach used has succeeded in producing competitive performance, and application developers are advised to focus on improving technical aspects, such as fixing login, verification, and transaction problems, in order to increase user satisfaction.

Keywords: banking apps, sentiment analysis, *support vector machine*, user reviews

1 Pendahuluan

Aplikasi perbankan telah menjadi alat penting bagi banyak orang dalam mengelola keuangan mereka sehari-hari. Aplikasi ini memungkinkan pengguna untuk melakukan berbagai transaksi seperti transfer uang, pembayaran tagihan, pengecekan saldo, dan berbagai layanan perbankan lainnya dengan cepat dan mudah [1]. Dengan kemajuan teknologi, aplikasi perbankan terus berkembang untuk memberikan pengalaman pengguna yang lebih baik dan lebih aman [1], [2]. Namun untuk memastikan kualitas dan efektivitas layanan, sangat penting bagi pengembang aplikasi perbankan untuk memahami *feedback* pengguna, karena pemahaman ini dapat secara langsung memengaruhi keberhasilan adaptasi fitur baru, perbaikan keamanan, dan kepuasan pelanggan secara keseluruhan.

Salah satu sumber utama *feedback* pengguna adalah ulasan aplikasi di Google Play Store. Ulasan pengguna di platform ini memberikan wawasan berharga tentang kepuasan pengguna, keluhan, dan saran untuk perbaikan. Ulasan ini dapat berfungsi sebagai indikator penting untuk mengevaluasi kinerja aplikasi dan mengidentifikasi area yang memerlukan peningkatan [3]. Namun, dengan jumlah ulasan yang sangat banyak, diperlukan metode yang efektif untuk menganalisis sentimen yang terkandung dalam ulasan-ulasan tersebut.

Analisis sentimen adalah teknik yang digunakan untuk mengidentifikasi dan mengkategorikan opini yang terkandung dalam teks [4]. Dalam konteks ulasan aplikasi, analisis sentimen membantu mengidentifikasi apakah ulasan pengguna bersifat positif atau negatif [5]. Metode ini sangat berguna untuk menganalisis *feedback* dalam jumlah besar dan memberikan gambaran umum tentang persepsi pengguna terhadap aplikasi perbankan. Dengan analisis sentimen, pengembang dapat lebih memahami kebutuhan dan ekspektasi pengguna, serta mengambil tindakan yang diperlukan untuk meningkatkan kualitas aplikasi.

Salah satu algoritma yang sering digunakan dalam analisis sentimen adalah *Support Vector Machine (SVM)*. *SVM* adalah metode pembelajaran mesin yang digunakan untuk klasifikasi dan regresi. Keunggulan *SVM* terletak pada kemampuannya untuk bekerja dengan baik dalam ruang dimensi tinggi dan efisiensinya dalam menemukan *hyperplane* yang optimal untuk memisahkan data [6], [7]. Beberapa penelitian sebelumnya telah menunjukkan bahwa *SVM* memiliki performa yang sangat baik dalam analisis sentimen, salah satunya pada penelitian Ziedhan Alifio Diekson *et al.* yang menganalisis sentimen terhadap ulasan aplikasi Traveloka menggunakan *SVM*, *Logistic Regression*, dan *Naive Bayes*, dan menemukan bahwa *SVM* mencapai akurasi tertinggi sebesar 84,58% [8].

Oleh sebab itu, penelitian ini bertujuan untuk mengembangkan model yang efektif dalam mengkategorikan sentimen ulasan pengguna aplikasi perbankan, sehingga dapat memberikan informasi yang berguna bagi pengembang aplikasi untuk meningkatkan kualitas dan kinerja aplikasi perbankan mereka. Selain itu, penelitian ini juga berkontribusi dalam menyediakan dataset untuk multi bank yang dapat digunakan sebagai *benchmark*. Dengan adanya dataset ini, diharapkan dapat mendukung pengembangan model-model prediksi di industri perbankan secara lebih akurat dan relevan, serta menjadi acuan bagi penelitian dan implementasi di masa depan.

Untuk memperoleh model dengan akurasi tinggi, penelitian ini juga melakukan eksplorasi berbagai skenario pembagian data latih dan data uji. Selain itu, guna memastikan model yang lebih stabil dan konsisten, dilakukan pengujian berulang dengan menggunakan nilai *random state* yang berbeda. Dengan pendekatan ini, penelitian diharapkan dapat mengembangkan model yang efektif dalam mengkategorikan sentimen ulasan pengguna, sehingga dapat memberikan informasi yang berguna bagi pengembang aplikasi untuk meningkatkan kualitas dan kinerja aplikasi perbankan mereka.

2 Tinjauan Literatur

Berbagai penelitian telah mengkaji penerapan analisis sentimen di berbagai konteks dengan menggunakan algoritma *Support Vector Machine (SVM)* dan metode lainnya. Misalnya, penelitian oleh Natt Leelawat *et al.* meneliti sentimen wisata di Thailand selama pandemi COVID-19 dengan menggunakan *SVM*, *Random Forest*, dan *Decision Tree*, dan hasilnya menunjukkan bahwa *SVM* memberikan akurasi tertinggi sebesar 77,4% dalam analisis sentiment [9]. Selanjutnya, Andrea Perera *et al.* juga berkontribusi dalam bidang ini dengan meneliti deteksi *cyberbullying* di media sosial menggunakan *SVM*, *Naive Bayes*, dan *Logistic Regression*, di mana *SVM* menunjukkan akurasi tertinggi sebesar 75,5%, diikuti oleh *Logistic Regression* dengan 74,15% dan *Naive Bayes* dengan 72,81% [10]. Leonard Gunawan *et al.* dalam penelitiannya mengenai analisis sentimen ulasan restoran di Jakarta,

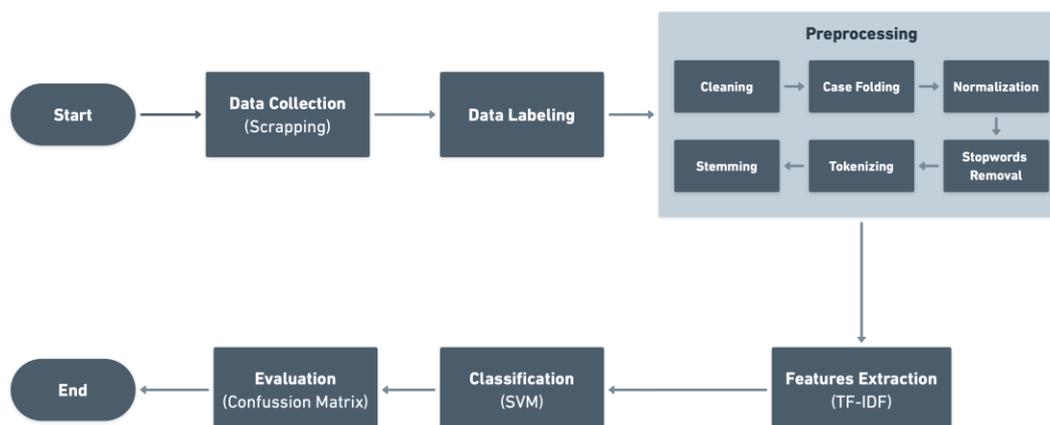
<http://sistemasi.ftik.unisi.ac.id>

menunjukkan bahwa SVM mencapai akurasi tertinggi sebesar 79%, lebih baik daripada Naive Bayes yang memiliki akurasi 77% [11]. Sementara itu, Handan Cam *et al.* mengkaji sentimen pada postingan Twitter terkait keuangan dengan enam algoritma pembelajaran mesin: Support Vector Machine (SVM), Multilayer Perceptron (MLP), Naive Bayes, Logistic Regression, K-Nearest Neighbors (KNN), dan Decision Trees (DTs), dan menemukan bahwa SVM dan MLP mencapai akurasi tertinggi masing-masing sebesar 89% dan 88%, lebih tinggi daripada algoritma lainnya yang mencapai sekitar 78,5% [12]. Terakhir, penelitian oleh Murat Demircan *et al.* yang mengembangkan model analisis sentimen berbahasa Turki menggunakan data ulasan *e-commerce* dan memanfaatkan lima algoritma pembelajaran mesin, yaitu Support Vector Machine (SVM), Random Forest, Decision Tree, Logistic Regression, dan K-Nearest Neighbors (KNN), menemukan bahwa model berbasis SVM dan Random Forest memberikan hasil terbaik dengan nilai *f1-score* tertinggi dibandingkan model lainnya [13].

Dari studi literatur, Support Vector Machine (SVM) telah terbukti sebagai algoritma yang efektif dalam analisis sentimen di berbagai konteks. Sepengetahuan penulis, penelitian yang fokus pada multi bank masih terbatas, sehingga memberikan peluang untuk melakukan analisis sentimen pada aplikasi perbankan menggunakan SVM. Pada penelitian ini, penulis melakukan eksplorasi berbagai skenario pembagian data latih dan data uji, serta pengujian berulang dengan menggunakan nilai random state yang berbeda untuk memperoleh hasil yang lebih stabil. Pada penelitian ini dilakukan pengumpulan data dari tiga bank populer di Indonesia, dan dataset ini belum tersedia di *repository* publik. Dengan demikian, penelitian ini turut berkontribusi dalam menyediakan dataset multi bank yang dapat digunakan sebagai *benchmark* bagi penelitian selanjutnya dalam bidang ini.

3 Metode Penelitian

Alur proses penelitian ini ditampilkan pada Gambar 1. Proses dimulai dengan pengumpulan dan pelabelan data, kemudian dilanjutkan dengan pra-pemrosesan, ekstraksi fitur, klasifikasi menggunakan algoritma SVM, dan diakhiri dengan evaluasi kinerja model.



Gambar 1. Flowchart klasifikasi sentiment pada ulasan aplikasi perbankan

3.1 Alat

Penelitian ini dijalankan di Google Colab dengan spesifikasi prosesor Intel Xeon CPU @ 2.00GHz, RAM sebesar 12 GB, dan GPU NVIDIA Tesla T4 untuk mempercepat komputasi. Google Colab dipilih karena menyediakan lingkungan yang efisien dan mendukung eksekusi kode berbasis *cloud* tanpa memerlukan perangkat keras lokal. Sedangkan *library* yang digunakan akan ditampilkan pada Tabel 1.

Tabel 1. Library

Library	Versi
google-play-scraper	1.2.7
sklearn	1.5.2
pandas	2.2.2
nltk	3.8.1

indoNLP	0.3.4
sastrawi	1.2.0
numpy	1.26.4
matplotlib	3.7.1
wordcloud	1.9.3
joblib	1.4.2

3.2 Pengumpulan Data

Data diambil dari ulasan pengguna aplikasi BCA Mobile, BRI Mobile, Livin Mandiri di Google Play Store menggunakan *library* Google Play Scraper. Data yang dicari di *filter* berdasarkan konten terbaru dalam bahasa Indonesia dan di wilayah Indonesia. Data yang diambil *mencakup username, score atau rating, at (timestamp), dan content*. Hanya data *score* atau *rating* yang berada dalam rentang 1-5 dan konten ulasan yang digunakan untuk analisis.

3.3 Pelabelan Data

Pelabelan data dilakukan untuk mengkategorikan data menjadi kelas negatif dan positif, yang kemudian akan digunakan dalam pelatihan model *machine learning* analisis sentimen. Proses Pelabelan data dilakukan secara otomatis menggunakan program python. Label diberikan berdasarkan dari data *rating* [14]. Jika nilai *rating* adalah 1, 2 atau 3 maka akan diberi label 0 (negatif) dan jika nilai *rating* adalah 4 & 5 maka akan diberi label 1 (positif), seperti yang dirumuskan dalam persamaan (1).

$$Label = \begin{cases} 0, & \text{jika } rating \leq 3 \\ 1, & \text{jika } rating > 3 \end{cases} \quad (1)$$

3.4 Preprocessing Data

Preprocessing adalah serangkaian langkah awal yang dilakukan untuk mempersiapkan data mentah menjadi lebih bersih, terstruktur, dan siap digunakan dalam analisis atau pembuatan model *machine learning*. Proses ini diperlukan untuk meningkatkan kualitas data dan memastikan bahwa analisis yang dilakukan menghasilkan informasi yang akurat dan relevan. Berikut adalah langkah *preprocessing* yang dilakukan :

1. Cleaning Dataset

Cleaning dataset, yaitu proses membersihkan data dari *noise* atau informasi yang tidak relevan [15]. Proses ini dapat melibatkan penghapusan data yang tidak lengkap (*null* atau *missing values*), atau membuang entri yang tidak relevan dalam dataset.

2. Case Folding

Case folding, yaitu proses mengubah semua karakter teks menjadi huruf kecil (*lowercase*) [16]. Ini dilakukan untuk memastikan konsistensi dalam representasi teks, sehingga kata yang sama dengan penulisan huruf yang berbeda (besar kecilnya) dianggap sama.

3. Normalization

Normalization adalah proses mengubah teks menjadi format standar atau normal. Pada proses ini bisa melibatkan beberapa langkah seperti mengganti emoji dengan teks yang sesuai, menghilangkan karakter *non-alphabet*, menghapus spasi berlebih, menghapus tanda baca, mengganti kata slang dengan kata formal, dan menangani pengulangan huruf dalam kata (*word elongation*).

4. Stopwords Removal

Stopwords removal adalah proses menghapus kata-kata umum (*stopwords*) dari teks yang tidak memberikan nilai informasi yang signifikan dalam analisis teks. *Stopwords* biasanya terdiri dari kata-kata seperti "dan", "atau", "di", "yang", dan sebagainya. Penghapusan *stopwords* membantu memfokuskan analisis pada kata-kata kunci yang lebih informatif.

5. Tokenizing

Tokenizing atau proses membagi teks menjadi unit-unit yang lebih kecil seperti kata-kata (*tokens*) [17]. *Tokenizing* dapat dilakukan dengan menggunakan fungsi seperti *word_tokenize* dari NLTK

(*Natural Language Toolkit*) untuk memisahkan teks menjadi *token* berdasarkan spasi dan tanda baca.

6. Stemming

Stemming adalah proses mengubah kata-kata menjadi bentuk dasarnya atau kata dasar (*stem*) dengan menghapus imbuhan atau akhiran kata [18]. Tujuannya adalah untuk mengurangi kata-kata menjadi bentuk yang lebih sederhana sehingga variasi kata yang berbeda dapat dihitung sebagai satu kata. Dalam penelitian ini karena data menggunakan bahasa Indonesia, stemming dilakukan dengan menggunakan *library* Sastrawi.

3.5 Features Extraction

TF-IDF (*Term Frequency-Inverse Document Frequency*) adalah metode yang digunakan untuk mengukur pentingnya kata dalam sebuah dokumen teks dalam suatu koleksi dokumen [19]. Bobot *TF-IDF* diberikan kepada kata-kata yang sering muncul dalam dokumen tetapi jarang muncul di seluruh koleksi dokumen. *TF-IDF* dalam hal ini digunakan untuk mengekstraksi fitur-fitur kritis dari teks, memungkinkan *SVM* untuk membangun model yang dapat membedakan antara sentimen positif dan negatif berdasarkan kata-kata yang digunakan dalam ulasan. *Term Frequency* (*TF*) diukur berdasarkan frekuensi kemunculan suatu kata dalam dokumen tertentu, sementara *Inverse Document Frequency* (*IDF*) mengukur seberapa jarang kata tersebut muncul di seluruh koleksi dokumen. Dengan menggabungkan kedua ukuran ini, *TF-IDF* memberikan bobot yang lebih tinggi pada kata-kata yang umum di dalam dokumen tetapi tidak umum di seluruh koleksi, sehingga meningkatkan akurasi dalam klasifikasi sentimen. Rumus untuk menghitung *TF*, *IDF*, dan *TF-IDF* masing-masing diuraikan dalam persamaan (2), (3), dan (4).

$$TF(t, d) = \frac{\text{Jumlah kemunculan kata } t \text{ dalam dokumen } d}{\text{Total kata dalam dokumen } d} \quad (2)$$

$$IDF(t, D) = \log\left(\frac{\text{Total jumlah dokumen } D}{\text{jumlah dokumen yang mengandung kata } t}\right) \quad (3)$$

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (4)$$

Keterangan :

1. *t* adalah kata yang dianalisis.
2. *d* adalah dokumen yang sedang dipertimbangkan.
3. *D* adalah koleksi seluruh dokumen.

3.6 Klasifikasi

Pada tahap ini dilakukan pembuatan model *classification* menggunakan algoritma *SVM*. Dalam penelitian ini, *SVM* dengan kernel linear digunakan karena kemampuannya dalam menemukan *hyperplane* optimal yang memisahkan dua kelas (positif dan negatif) secara linier dalam ruang fitur. Inti dari *SVM* linear adalah untuk mencari garis atau permukaan yang memaksimalkan *margin*, yaitu jarak antara *hyperplane* dan titik data terdekat dari kedua kelas [20]. *Hyperplane* yang dihasilkan oleh *SVM* bertujuan untuk memastikan bahwa *margin* antara dua kelas data adalah yang terbesar, yang berarti model memiliki generalisasi yang baik pada data baru.

3.7 Evaluasi

Tahap terakhir adalah tahap evaluasi, setiap model yang telah dibuat di evaluasi seberapa baik model tersebut. Evaluasi dalam penelitian ini menggunakan *confusion matrix*. *Confusion matrix* adalah tabel yang digunakan untuk menggambarkan performa model klasifikasi dengan membandingkan hasil prediksi model dengan nilai sebenarnya dari data uji [21]. Gambar 2 menunjukkan visualisasi dari *confusion matrix*, yang terdiri dari empat komponen utama:

Actual	Negatif	TN	FP
	Positif	FN	TP
		Negatif	Positif
		Predicted	

Gambar 2. Visualisasi *confusion matrix*

Keterangan :

1. *True Negative* (TN) : jumlah data di mana model memprediksi kelas negatif, dan prediksi tersebut benar
2. *True Positive* (TP) : jumlah data di mana model memprediksi kelas positif, dan prediksi tersebut benar
3. *False Negative* (FN) : jumlah data di mana model memprediksi kelas negatif, tetapi sebenarnya data tersebut positif
4. *False Positive* (FP) : jumlah data di mana model memprediksi kelas positif, tetapi sebenarnya data tersebut negatif

Dari nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang diperoleh dari *confusion matrix*, kita dapat menghitung beberapa metrik evaluasi penting untuk menilai kinerja model klasifikasi, seperti Akurasi, Presisi, *Recall*, dan *F1-Score*. Metrik-metrik ini memberikan gambaran tentang seberapa baik model dalam memprediksi kelas dengan benar, baik pada kelas positif maupun negatif. Perhitungan metrik tersebut dapat dilakukan menggunakan persamaan (5), (6), (7), dan (8) berikut ini:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

$$Presisi = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

$$F1\ Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (8)$$

4 Hasil dan Pembahasan

Pada bagian ini, akan dijelaskan mengenai setiap langkah yang dilakukan dalam penelitian ini. Penelitian ini dilakukan menggunakan bahasa pemrograman Python dan diimplementasikan melalui platform Google Colab. Seluruh proses, mulai dari pengumpulan data, tahap preprocessing, pembangunan model, hingga evaluasi performa model, dilakukan secara terstruktur dengan memanfaatkan berbagai library.

4.1 Pengumpulan Data

Data ulasan berhasil diperoleh dari tiga aplikasi perbankan, yaitu BCA Mobile, BRI Mobile, dan Livin Mandiri. Setiap aplikasi diambil sebanyak 10.000 ulasan, sehingga total keseluruhan data yang terkumpul berjumlah 30.000 ulasan. Data yang berhasil dikumpulkan dapat diakses di <https://doi.org/10.17632/mvshyj7g67.1> untuk BCA Mobile, <https://doi.org/10.17632/8cnzj9h72v.1> untuk BRI Mobile dan <https://doi.org/10.17632/h8p5v6r6dn.1> untuk Livin Mandiri. Tabel 2 hingga Tabel 4 akan menunjukkan contoh data dari masing-masing dataset tersebut.

Tabel 2. Dataset BCA mobile

Username	Score	At	Content
Pengguna Google	5	2024-06-11 23:31:20	Aman dan terpercaya.
Pengguna Google	1	2024-06-11 21:15:09	Pengiriman CC gagal terus najis!
Pengguna Google	3	2024-06-11 7:39:27	Saat mau masuk ke apps bca mobile selalu lama ga seperti sebelumnya. Tolong diperbaiki lagi. Terimakasih

Tabel 3. Dataset BRI mobile

Username	Score	At	Content
Pengguna Google	1	2024-06-11 20:06:56	Terlalu banyak potongan
Pengguna Google	5	2024-06-11 19:46:24	Sangat Membantu dan Bermanfaat Mantap Min 😊
Pengguna Google	3	2024-06-11 19:32:36	Privikasi wajah punya saya ku lebih dari 5 menit gak masuk masuk

Tabel 4. Dataset livin mandiri

Username	Score	At	Content
Pengguna Google	5	2024-06-11 7:13:47	Baguss
Pengguna Google	1	2024-06-11 7:06:29	Kenapa aplikasi livin by mandiri saya gak bisa pas mau masukin pin saya
Pengguna Google	3	2024-06-11 7:03:24	Ok, cukup lumayan.

4.2 Pelabelan Data

Setelah data didapatkan, kemudian dilakukan pelabelan data. Dari proses pelabelan data, didapatkan data positif sebanyak 4892 untuk BCA Mobile, 8182 untuk BRI Mobile, dan 5353 untuk Livin Mandiri. Sedangkan untuk data negatif didapat 5108 untuk BCA Mobile, 1818 untuk BRI Mobile, dan 4647 untuk Livin Mandiri. Gambar 3 menunjukkan visualisasi persentase data berlabel positif dan negatif hasil pelabelan. Contoh hasil pelabelan pada masing-masing dataset disajikan pada Tabel 5, Tabel 6, dan Tabel 7.

Tabel 5. Pelabelan dataset BCA mobile

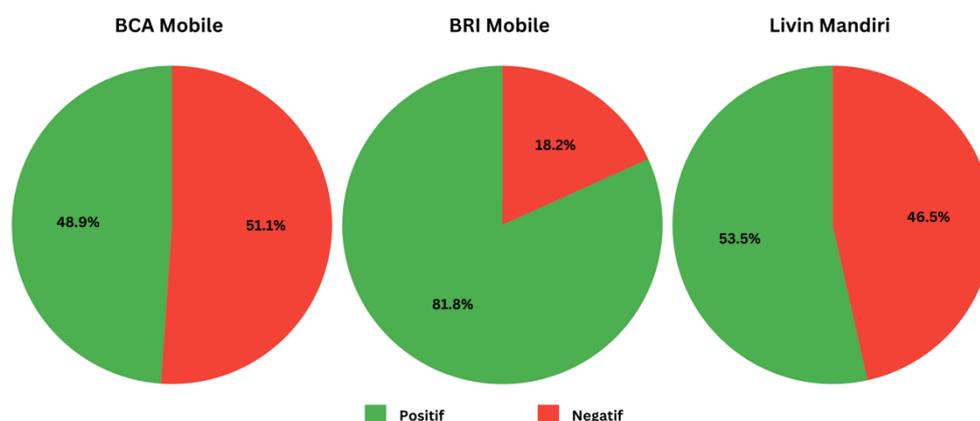
Content	Score	Label
Aman dan terpercaya.	5	1
Pengiriman CC gagal terus najis!	1	0
Saat mau masuk ke apps bca mobile selalu lama ga seperti sebelumnya. Tolong diperbaiki lagi. Terimakasih	3	0

Tabel 6. Pelabelan dataset BRI mobile

Content	Score	Label
Terlalu banyak potongan	1	0
Sangat Membantu dan Bermanfaat Mantap Min 😊	5	1
Privikasi wajah punya saya ku lebih dari 5 menit gak masuk masuk	3	0

Tabel 7. Pelabelan dataset livin mandiri

Content	Score	Label
Baguss	5	1
Kenapa aplikasi livin by mandiri saya gak bisa pas mau masukin pin saya	1	0
Ok, cukup lumayan.	3	0



Gambar 3. Presentase data positif dan negatif hasil labeling

4.3 Preprocessing Data

Setelah proses pelabelan selesai, data kemudian melalui tahapan *preprocessing*. Proses pertama adalah *cleaning* dataset, dan setelah dilakukan pengecekan, dipastikan bahwa data sudah lengkap tanpa adanya nilai yang *null*. Proses selanjutnya adalah *case folding*, *normalization*, *stopwords removal*, *tokenizing*, dan *stemming*. Pada langkah *normalization*, beberapa langkah yang dilakukan meliputi menerjemahkan emoji ke dalam teks, menghilangkan karakter *non-alphabet*, menghapus spasi berlebih, menghapus tanda baca, mengganti kata slang dengan kata formal serta menangani pengulangan huruf dalam kata (*word elongation*). Semua proses dari *case folding* hingga *stemming* akan divisualisasikan dalam tabel Tabel 8 sampai Tabel 17.

Tabel 8. Proses case folding

Sebelum	Sesudah
BCA Mobile emg mantulll, fitur2nya lengkap bgt. Gak nyesel pake ini. 👍	bca mobile emg mantulll, fitur2nya lengkap bgt. gak nyesel pake ini. 👍

Tabel 9. Proses translate emoji

Sebelum	Sesudah
bca mobile emg mantulll, fitur2nya lengkap bgt. gak nyesel pake ini. 👍	bca mobile emg mantulll, fitur2nya lengkap bgt. gak nyesel pake ini. !jempol_ke_atas!

Tabel 10. Proses menghilangkan karakter non-alphabet

Sebelum	Sesudah
bca mobile emg mantulll, fitur2nya lengkap bgt. gak nyesel pake ini. !jempol_ke_atas!	bca mobile emg mantulll fitur nya lengkap bgt gak nyesel pake ini jempol ke atas

Tabel 11. Proses menghapus spasi berlebih

Sebelum	Sesudah
bca mobile emg mantulll, fitur2nya lengkap bgt. gak nyesel pake ini. !jempol_ke_atas!	bca mobile emg mantulll fitur nya lengkap bgt gak nyesel pake ini jempol ke atas

Tabel 12. Proses menghapus tanda baca

Sebelum	Sesudah
bca mobile emg mantulll fitur nya lengkap bgt gak nyesel pake ini jempol ke atas	bca mobile emg mantulll fitur nya lengkap bgt gak nyesel pake ini jempol ke atas

Tabel 13. Proses mengganti kata slang dengan kata formal

Sebelum	Sesudah
bca mobile emg mantulll fitur nya lengkap bgt gak nyesel pake ini jempol ke atas	baca mobile memang mantulll fitur nya lengkap banget enggak menyesal pakai ini jempol ke atas

Tabel 14. Proses menangani pengulangan huruf dalam kata (*word elongation*)

Sebelum	Sesudah
baca mobile memang mantulll fitur nya lengkap banget enggak menyesal pakai ini jempol ke atas	baca mobile memang mantul fitur nya lengkap banget enggak menyesal pakai ini jempol ke atas

Tabel 15. Proses *stopwords removal*

Sebelum	Sesudah
baca mobile memang mantulll fitur nya lengkap banget enggak menyesal pakai ini jempol ke atas	baca mobile memang mantul fitur nya lengkap banget enggak menyesal pakai ini jempol ke atas

Tabel 16. Proses *tokenizing*

Sebelum	Sesudah
baca mobile mantul fitur nya lengkap banget menyesal pakai jempol	baca, mobile, mantul, fitur, nya, lengkap, banget, menyesal, pakai, jempol

Tabel 17. Proses *stemming*

Sebelum	Sesudah
baca, mobile, mantul, fitur, nya, lengkap, banget, menyesal, pakai, jempol	baca, mobile, mantul, fitur, nya, lengkap, banget, sesal, pakai, jempol

4.4 Features Extraction (TF-IDF)

Tahap *feature extraction* dalam penelitian ini dilakukan menggunakan metode *TF-IDF*, yang bertujuan untuk mengubah teks menjadi representasi numerik dengan menyoroti kata-kata paling penting berdasarkan frekuensi kemunculannya di dalam dokumen serta seberapa jarang kata tersebut muncul di dokumen lain. Proses ini membantu model pembelajaran mesin untuk lebih mudah mengenali fitur yang relevan dari teks. Untuk mempermudah pemahaman, hasil dari proses *TF-IDF* ini akan divisualisasikan dalam bentuk *word cloud*, di mana kata-kata dengan skor tertinggi akan ditampilkan lebih besar, sehingga memudahkan identifikasi kata kunci penting secara visual.



Gambar 4. Visualisasi wordcloud dataset BCA mobile

Pada Gambar 4, bisa dilihat bahwa ulasan negatif aplikasi BCA Mobile didominasi oleh kata-kata seperti "baca", "verifikasi", "aplikasi", "susah", dan "error". Hal ini menunjukkan bahwa banyak pengguna BCA Mobile mengalami masalah terkait proses verifikasi dan pembacaan informasi dalam aplikasi, serta sering menghadapi kesulitan dan kesalahan teknis. Sebaliknya, pada ulasan positif, kata-kata seperti "bagus", "ok", "bantu", "mantap", dan "mudah" mendominasi. Ini menandakan bahwa pengguna merasa puas dengan aplikasi, menemukan aplikasi ini bermanfaat dan membantu dalam melakukan transaksi perbankan. Kata "transaksi" juga muncul dengan frekuensi tinggi, menandakan bahwa fitur transaksi dalam aplikasi BCA Mobile mendapat apresiasi dari pengguna.



Gambar 5. Visualisasi wordcloud dataset BRI mobile

Pada Gambar 5, ulasan negatif BRI Mobile didominasi oleh kata-kata seperti "login", "brimo", "gagal", dan "error". Hal ini menunjukkan bahwa banyak pengguna mengalami masalah dalam proses masuk ke aplikasi dan sering kali menghadapi kegagalan teknis. Kata "susah" dan "ganggu" juga muncul, mengindikasikan adanya ketidakpuasan terhadap pengalaman pengguna yang sulit dan terganggu. Sebaliknya, pada ulasan positif, kata-kata seperti "bantu", "bagus", "mantap", dan "mudah" mendominasi. Ini menandakan bahwa pengguna merasa puas dengan aplikasi, menemukan aplikasi ini sangat membantu dan mudah digunakan untuk melakukan transaksi perbankan. Kata "transaksi" juga muncul dengan frekuensi tinggi, menandakan bahwa fitur transaksi dalam aplikasi BRI Mobile mendapat apresiasi dari pengguna.



Gambar 6. Visualisasi wordcloud dataset livin mandiri

Pada Gambar 6, ulasan Livin Mandiri negatif didominasi oleh kata-kata seperti "aplikasi", "update", "buka", dan "susah". Hal ini menunjukkan bahwa banyak pengguna mengalami masalah terkait pembaruan aplikasi dan kesulitan dalam membuka aplikasi tersebut. Kata "transaksi" dan "error" juga muncul, mengindikasikan adanya kendala dalam melakukan transaksi dan seringnya muncul kesalahan teknis. Sebaliknya, pada ulasan positif, kata-kata seperti "bagus", "bantu", "mantap", dan "mudah" mendominasi. Ini menandakan bahwa pengguna merasa puas dengan aplikasi, menemukan aplikasi ini sangat membantu dan mudah digunakan untuk melakukan transaksi perbankan. Kata

"transaksi" juga muncul dengan frekuensi tinggi, menandakan bahwa fitur transaksi dalam aplikasi Livin Mandiri mendapat apresiasi dari pengguna.

4.5 Klasifikasi

Penelitian ini memanfaatkan empat dataset, yaitu ulasan dari BCA Mobile, BRI Mobile, Livin Mandiri, serta kombinasi dari ketiga dataset tersebut. Setiap dataset dianalisis secara individual, sekaligus digabungkan untuk memperoleh *insight* yang lebih komprehensif terkait sentimen pengguna aplikasi perbankan. Pembuatan model dibuat menggunakan skenario pembagian data latih dan data uji dalam rasio 60:40, 70:30, 80:20, dan 90:10. Setiap skenario diujicobakan sebanyak 10 kali dengan nilai *random state* yang berbeda, menghasilkan total 160 model. Untuk membangun model klasifikasi sentimen, digunakan metode SVM dengan kernel linear. SVM diterapkan dengan pengaturan *class_weight='balanced'* untuk menangani ketidakseimbangan data, sehingga model dapat lebih akurat dalam mengklasifikasikan sentimen pengguna, terutama dalam mengenali kelas minoritas pada setiap dataset.

4.6 Evaluasi

Setelah pembuatan model, langkah selanjutnya adalah evaluasi model. Dengan melakukan perhitungan rata-rata hasil evaluasi untuk setiap dataset berdasarkan skenario yang digunakan, diperoleh hasil seperti yang disajikan pada Tabel 18 hingga Tabel 21 berikut ini:

Tabel 18. Rata-rata hasil evaluasi dari dataset BCA mobile

Skenario	Accuracy	Precision	Recall	F1-Score
60:40	88.60%	88.76%	88.60%	88.58%
70:30	88.58%	88.72%	88.58%	88.55%
80:20	88.56%	88.69%	88.56%	88.53%
90:10	88.45%	88.53%	88.45%	88.42%

Tabel 19. Rata-rata hasil evaluasi dari dataset BRI mobile

Skenario	Accuracy	Precision	Recall	F1-Score
60:40	91.25%	92.52%	91.25%	91.62%
70:30	91.22%	92.45%	91.22%	91.58%
80:20	91.40%	92.63%	91.40%	91.75%
90:10	91.57%	92.94%	91.57%	91.95%

Tabel 20. Rata-rata hasil evaluasi dari dataset livin mandiri

Skenario	Accuracy	Precision	Recall	F1-Score
60:40	85.06%	85.52%	85.06%	85.06%
70:30	85.09%	85.57%	85.09%	85.10%
80:20	84.94%	85.44%	84.94%	84.95%
90:10	85.51%	85.96%	85.51%	85.51%

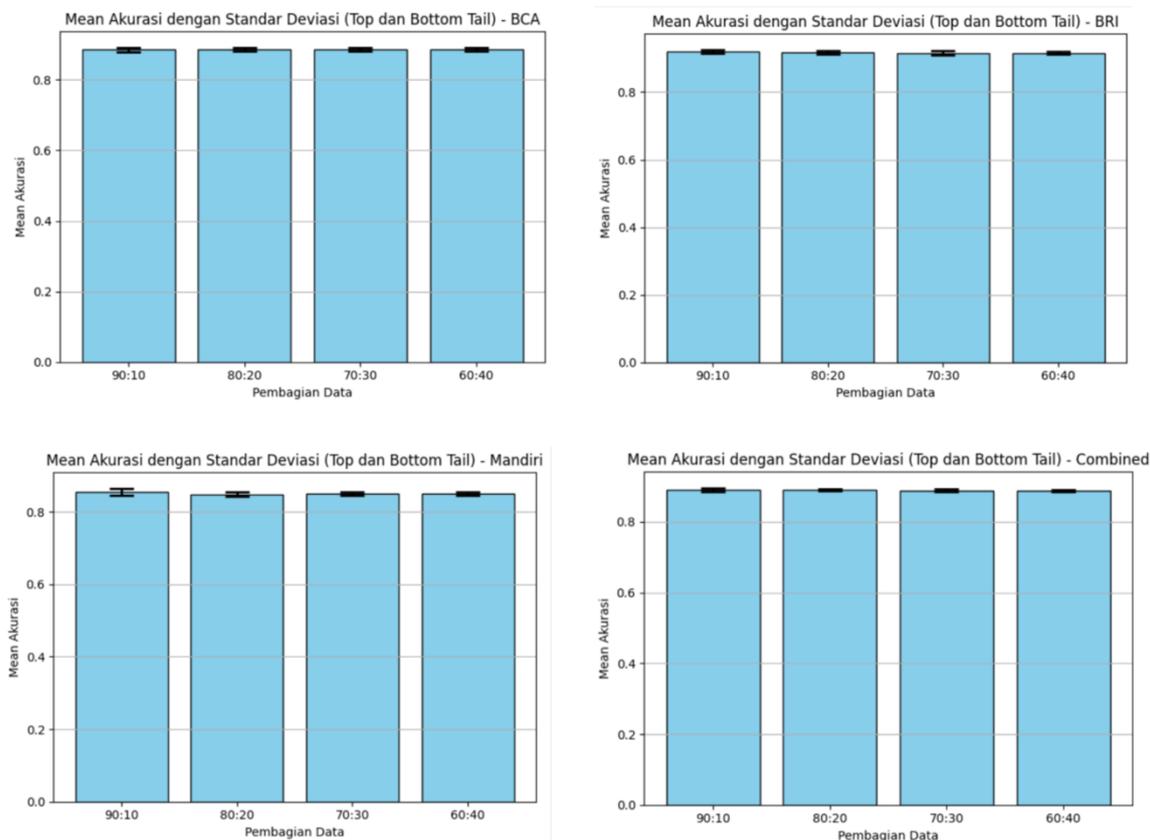
Tabel 21. Rata-rata hasil evaluasi dari dataset gabungan

Skenario	Accuracy	Precision	Recall	F1-Score
60:40	88.65%	89.42%	88.65%	88.76%
70:30	88.64%	89.41%	88.64%	88.74%
80:20	88.70%	89.48%	88.70%	88.81%
90:10	88.84%	89.65%	88.84%	88.95%

Untuk mengevaluasi konsistensi performa model dalam berbagai skenario pembagian data, dapat dilihat dari visualisasi standar deviasi pada masing-masing dataset yang ditampilkan pada Gambar 7.

<http://sistemasi.ftik.unisi.ac.id>

Hasil visualisasi menunjukkan bahwa standar deviasi akurasi pada setiap rasio pembagian data sangat kecil, terlihat dari garis hitam tipis di atas batang akurasi. Standar deviasi ini diperoleh dari sepuluh kali percobaan pembuatan model untuk setiap skenario, dengan nilai *random state* yang berbeda-beda. Meskipun ada variasi dalam *random state*, jarak garis hitam pada setiap dataset tetap berdekatan, yang mengindikasikan bahwa variasi akurasi dari berbagai percobaan pembuatan model sangat rendah. Hal ini menunjukkan bahwa model yang telah dibuat memiliki performa yang stabil dan tidak terlalu terpengaruh oleh perubahan dalam inialisasi *random state*.



Gambar 7. Visualisasi standar deviasi

Dari keseluruhan model tersebut, dataset BCA Mobile berhasil mencapai akurasi tertinggi sebesar 89.53% dengan *random state* 404, dataset BRI Mobile berhasil mencapai akurasi tertinggi 92.87% dengan *random state* 99, dataset Livin Mandiri mencapai akurasi tertinggi 87.66% dengan *random state* 123, dan dataset gabungan mencapai akurasi tertinggi 90.32% dengan *random state* 0. Semua hasil tersebut diperoleh pada skenario 90:10.

Tabel 22. Model dengan akurasi tertinggi dari masing-masing dataset

Dataset	Skenario	Random State	Accuracy	Precision	Recall	F1 Score
BCA Mobile	90:10	404	89.73%	89.76%	89.73%	89.72%
BRI Mobile	90:10	99	92.97%	93.98%	92.97%	93.24%
Livin Mandiri	90:10	123	87.46%	88.01%	87.46%	87.48%
Gabungan	90:10	0	90.05%	90.57%	90.05%	90.13%

Hasil ini menunjukkan bahwa penggunaan *SVM* dengan skenario pembagian data latih dan data uji 90:10 memberikan performa terbaik untuk semua dataset. Model untuk dataset BRI Mobile menunjukkan performa tertinggi secara keseluruhan dengan akurasi 92.97%. Penggunaan nilai *random state* yang berbeda memungkinkan untuk mengidentifikasi performa optimal dari model yang dibangun.

Hal ini penting untuk memastikan bahwa hasil yang diperoleh tidak bergantung pada satu nilai *random state* tertentu.

4.7 Komparasi Penelitian Terdahulu

Berdasarkan Tabel 23, hasil penelitian ini menunjukkan bahwa performa model SVM pada analisis sentimen aplikasi perbankan menunjukkan hasil yang unggul. Dataset BRI Mobile mencapai akurasi tertinggi sebesar 92.97%, melampaui analisis sentimen pariwisata di Thailand (77.4%) [8], cyberbullying di Twitter (75.5%) [9], dan ulasan restoran di Jakarta (79%) [10]. Gabungan ulasan dari aplikasi BCA Mobile, BRI Mobile, dan Livin Mandiri juga mencapai akurasi tinggi sebesar 90.05%, melebihi beberapa penelitian terdahulu, sedangkan dataset BCA Mobile (89.73%) dan Livin Mandiri (87.46%) mendekati hasil analisis sentimen Twitter terkait keuangan (89%) [11]. Temuan ini menunjukkan efektivitas analisis sentimen dalam memahami feedback pengguna aplikasi perbankan, serta menunjukkan bahwa pendekatan eksplorasi skenario pembagian data dan pengujian *random state* yang dilakukan dalam penelitian ini berhasil menghasilkan model yang stabil.

Tabel 23. Komparasi dengan penelitian sebelumnya

Referensi	Dataset	Akurasi
[9]	Pariwisata di Thailand selama pandemi Covid 19	77.4%
[10]	Cyberbullying di Twitter	75.5%
[11]	Ulasan Restoran di Jakarta	79%
[12]	Postingan Twitter terkait keuangan	89%
Penulis	Ulasan Aplikasi BCA Mobile	89.73%
Penulis	Ulasan Aplikasi BRI Mobile	92.97%
Penulis	Ulasan Aplikasi Livin Mandiri	87.46%
Penulis	Gabungan Ulasan Aplikasi BCA Mobile, BRI Mobile dan Livin Mandiri	90.05%

5 Kesimpulan

Berdasarkan visualisasi dari *word cloud*, ulasan negatif aplikasi perbankan didominasi oleh keluhan teknis seperti kesulitan *login*, pembaruan, verifikasi, dan kegagalan transaksi, menunjukkan perlunya perbaikan di area ini. Sebaliknya, ulasan positif menyoroti kemudahan penggunaan, manfaat, dan keandalan aplikasi, yang sudah diapresiasi oleh pengguna. Oleh karena itu, pengembang perlu fokus memperbaiki aspek teknis sambil mempertahankan dan meningkatkan fitur-fitur yang sudah memberikan pengalaman positif. Penelitian ini juga menunjukkan bahwa SVM dapat menghasilkan model yang baik untuk memprediksi ulasan pengguna aplikasi perbankan. Hasil analisis menunjukkan dataset BRI Mobile memiliki akurasi tertinggi (92.97%), disusul dataset Gabungan (90.05%), BCA Mobile (89.73%), dan Livin Mandiri (87.46%). Dibandingkan penelitian sebelumnya, metode SVM dalam penelitian ini terbukti lebih efektif untuk analisis sentimen ulasan aplikasi perbankan, dengan hasil akurasi yang lebih tinggi. Pendekatan yang digunakan, seperti eksplorasi skenario pembagian data dan pengujian berulang, berhasil memberikan performa yang kompetitif. Untuk penelitian mendatang, beberapa pengembangan dapat dilakukan, seperti mengeksplorasi metode *deep learning*, menganalisis fitur spesifik dalam ulasan, mengembangkan model dengan analisis aspek tertentu, dan memperluas penelitian ke aplikasi perbankan lainnya termasuk aplikasi bank digital seperti SeaBank, Bank Jago, dan Bank Neo Commerce.

Referensi

- [1] H. A. Alnemer, "Determinants of digital banking adoption in the Kingdom of Saudi Arabia: A technology acceptance model approach," *Digital Business*, vol. 2, no. 2, Jan. 2022, doi: 10.1016/j.digbus.2022.100037.
- [2] F. Mi Alnaser, S. Rahi, M. Alghizzawi, and A. H. Ngah, "Does artificial intelligence (AI) boost digital banking user satisfaction? Integration of expectation confirmation model and antecedents

<http://sistemasi.ftik.unisi.ac.id>

- of artificial intelligence enabled digital banking,” *Heliyon*, vol. 9, no. 8, Aug. 2023, doi: 10.1016/j.heliyon.2023.e18930.
- [3] S. J and K. U, “Sentiment analysis of amazon user reviews using a hybrid approach,” *Measurement: Sensors*, vol. 27, Jun. 2023, doi: 10.1016/j.measen.2023.100790.
- [4] A. A. Ilham, A. Bustamin, and E. Wahyudiarto, “Customer Satisfaction Assessment System on Transactions E-commerce Product Purchases Using Sentiment Analysis,” *International Journal on Advanced Science Engineering Information Technology*, vol. 13, no. 3, 2023.
- [5] O. Alsemaree, A. S. Alam, S. S. Gill, and S. Uhlig, “Sentiment analysis of Arabic social media texts: A machine learning approach to deciphering customer perceptions,” *Heliyon*, vol. 10, no. 9, May 2024, doi: 10.1016/j.heliyon.2024.e27863.
- [6] J. S. Pimentel, R. Ospina, and A. Ara, “A novel fusion Support Vector Machine integrating weak and sphere models for classification challenges with massive data,” *Decision Analytics Journal*, vol. 11, Jun. 2024, doi: 10.1016/j.dajour.2024.100457.
- [7] A. Wahyu Nugroho and N. Norhikmah, “Sentiment Analysis using the Support Vector Machine Algorithm on Covid_19,” *SISTEMASI: Jurnal Sistem Informasi*, vol. 13, no. 4, pp. 1758–1772, 2024, doi: <https://doi.org/10.32520/stmsi.v13i4.3778>.
- [8] Z. A. Diekson, M. R. B. Prakoso, M. S. Q. Putra, M. S. A. F. Syaputra, S. Achmad, and R. Sutoyo, “Sentiment analysis for customer review: Case study of Traveloka,” in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 682–690. doi: 10.1016/j.procs.2022.12.184.
- [9] N. Leelawat *et al.*, “Twitter data sentiment analysis of tourism in Thailand during the COVID-19 pandemic using machine learning,” *Heliyon*, vol. 8, no. 10, Oct. 2022, doi: 10.1016/j.heliyon.2022.e10894.
- [10] A. Perera and P. Fernando, “Cyberbullying Detection System on Social Media Using Supervised Machine Learning,” *Procedia Comput Sci*, vol. 239, pp. 506–516, 2024, doi: 10.1016/j.procs.2024.06.200.
- [11] L. Gunawan, M. S. Anggreainy, L. Wihan, Santy, G. Y. Lesmana, and S. Yusuf, “Support vector machine based emotional analysis of restaurant reviews,” in *Procedia Computer Science*, Elsevier B.V., 2022, pp. 479–484. doi: 10.1016/j.procs.2022.12.160.
- [12] H. Cam, A. V. Cam, U. Demirel, and S. Ahmed, “Sentiment analysis of financial Twitter posts on Twitter with the machine learning classifiers,” *Heliyon*, vol. 10, no. 1, Jan. 2024, doi: 10.1016/j.heliyon.2023.e23784.
- [13] M. Demircan, A. Seller, F. Abut, and M. F. Akay, “Developing Turkish sentiment analysis models using machine learning and e-commerce data,” *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 202–207, Jun. 2021, doi: 10.1016/j.ijcce.2021.11.003.
- [14] A. A. Syed, F. L. Gaol, A. Boediman, and W. Budiharto, “Airline reviews processing: Abstractive summarization and rating-based sentiment classification using deep transfer learning,” *International Journal of Information Management Data Insights*, vol. 4, no. 2, p. 100238, Nov. 2024, doi: 10.1016/j.jjime.2024.100238.
- [15] D. Chicco, L. Oneto, and E. Tavazzi, “Eleven quick tips for data cleaning and feature engineering,” *PLoS Comput Biol*, vol. 18, no. 12, Dec. 2022, doi: 10.1371/journal.pcbi.1010718.

- [16] H. Mustafidah, S. Suwarsito, and T. Pinandita, "Natural Language Processing for Mapping Exam Questions to the Cognitive Process Dimension," *International Journal of Emerging Technologies in Learning*, vol. 17, no. 13, pp. 4–16, 2022, doi: 10.3991/ijet.v17i13.29095.
- [17] S. Choo and W. Kim, "A study on the evaluation of tokenizer performance in natural language processing," *Applied Artificial Intelligence*, vol. 37, no. 1, 2023, doi: 10.1080/08839514.2023.2175112.
- [18] T. H. J. Hidayat, Y. Ruldeviyani, A. R. Aditama, G. R. Madya, A. W. Nugraha, and M. W. Adisaputra, "Sentiment analysis of twitter data related to Rinca Island development using Doc2Vec and SVM and logistic regression as classifier," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 660–667. doi: 10.1016/j.procs.2021.12.187.
- [19] W. N. Ibrahim Al-Obaydy, H. A. Hashim, Y. AbdulKhaleq Najm, and A. A. Jalal, "Document classification using term frequency-inverse document frequency and K-means clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 3, pp. 1517–1524, Sep. 2022, doi: 10.11591/ijeecs.v27.i3.pp1517-1524.
- [20] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput Oper Res*, vol. 152, Apr. 2023, doi: 10.1016/j.cor.2022.106131.
- [21] H. Yun, "Prediction model of algal blooms using logistic regression and confusion matrix," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2407–2413, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2407-2413.