

ANALISIS SENTIMEN ULASAN APLIKASI SMILE INDONESIA MENGGUNAKAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE (SVM)

Lina Rhomaningtias¹, Adenda Khairunisa², Shindi Shella May Wara³, Kartika Maulida Hindrayani⁴

^{1,2,3,4}Program Studi Sains Data, Universitas Pembangunan Nasional ‘Veteran’ Jawa Timur
 Jl. Rungkut Madya, Gn. Anyar, Kec. Gn. Anyar, Surabaya - Jawa Timur, Indonesia
 Email: ¹23083010077@student.upnjatim.ac.id, ²23083010075@student.upnjatim.ac.id,
³shindi.shella.fasilkom@upnjatim.ac.id, ⁴kartika.maulida.ds@upnjatim.ac.id

ABSTRAK

Tujuan studi ini adalah untuk mengevaluasi bagaimana pengguna memandang aplikasi digital SMILE Indonesia, sebuah *platform* layanan publik yang memantau penyampaian layanan kesehatan secara *real-time*. Menggunakan teknik *web scraping*, 383 ulasan pengguna dikumpulkan dari Google Play Store dan secara otomatis diklasifikasikan berdasarkan skor penilaian: ulasan dengan skor 1-2 dikategorikan sebagai negatif, ulasan dengan skor 4-5 sebagai positif, dan ulasan dengan skor 3 atau lebih rendah dikecualikan karena kemungkinan ambiguitas. Langkah-langkah *pre-processing* seperti *case folding*, pembersihan teks, tokenisasi, penghapusan kata, *stemming*, dan normalisasi diterapkan pada data yang telah dilabeli. Metode TF-IDF (Term Frequency–Inverse Document Frequency) kemudian digunakan untuk mewakili data secara numerik. Dua algoritma digunakan untuk klasifikasi: Naïve Bayes dan Support Vector Machine (SVM). Hasil evaluasi menunjukkan bahwa SVM mencapai 75% pada keempat metrik, sementara Naïve Bayes mencapai akurasi 79%, presisi 81%, recall 79%, dan F1-score 79%. Uji McNemar menunjukkan bahwa perbedaan kinerja antara kedua model tidak signifikan secara statistik ($p > 0.05$), meskipun Naïve Bayes memperoleh skor yang lebih tinggi. Penelitian sentimen ini memberikan wawasan tentang bagaimana masyarakat umum memandang layanan publik digital; sementara sikap negatif menekankan kesulitan teknis, sikap positif menyoroti aksesibilitas dan keuntungan praktis. Hasil ini dapat digunakan secara strategis oleh pengembang dan pembuat kebijakan untuk meningkatkan kualitas layanan digital berbasis *e-government*, terutama di bidang logistik kesehatan.

Kata kunci: SMILE Indonesia, analisis sentimen, SVM, Naïve Bayes, TF-IDF

ABSTRACT

The purpose of this study is to evaluate how users perceive the SMILE Indonesia digital application, a public service platform that monitors the delivery of health services in real time. Using web scraping techniques, 383 user reviews were collected from the Google Play Store and automatically classified based on rating scores: reviews with scores of 1-2 were categorized as negative, reviews with scores of 4-5 as positive, and reviews with scores of 3 or lower were excluded due to potential ambiguity. Pre-processing steps such as case folding, text cleaning, tokenization, word removal, stemming, and normalization were applied to the labeled data. The TF-IDF (Term Frequency–Inverse Document Frequency) method was then used to represent the data numerically. Two algorithms were used for classification: Naïve Bayes and Support Vector Machine (SVM). Evaluation results show that SVM achieved 75% on all four metrics, while Naïve Bayes achieved 79% accuracy, 81% precision, 79% recall, and 79% F1-score. The McNemar test indicates that the performance difference between the two models is not statistically significant ($p > 0.05$), although Naïve Bayes achieved higher scores. This sentiment analysis provides insights into how the general public perceives digital public services; while negative attitudes emphasize technical difficulties, positive attitudes highlight accessibility and practical benefits. These results can be strategically utilized by developers and policymakers to improve the quality of e-government-based digital services, particularly in the field of health logistics.

Keywords: SMILE Indonesia, sentiment analysis, SVM, Naïve Bayes, TF-IDF

1. PENDAHULUAN

Penggunaan aplikasi seluler telah menjadi aspek mendasar dari masyarakat kontemporer di era teknologi digital yang berkembang pesat. Berbagai sektor mulai dari pendidikan, keuangan, hingga layanan publik telah mengadopsi teknologi digital guna meningkatkan efisiensi dan kualitas layanan [1]. Salah satu aplikasi yang hadir sebagai solusi dalam bidang layanan publik adalah SMILE (Sistem Monitoring Inventaris Logistik Kesehatan secara Elektronik). Aplikasi ini dikembangkan oleh UNDP Indonesia bekerja sama dengan Kementerian



Kesehatan Republik Indonesia, sebagai sistem digital untuk mencatat, memantau, dan memperkuat rantai pasok logistik vaksin, obat-obatan, dan perbekalan kesehatan secara *real-time* [2].

SMILE Indonesia hadir sebagai inovasi digital yang menjawab tantangan pelayanan publik yang selama ini identik dengan proses manual, antrian panjang, dan keterbatasan waktu operasional. Dengan ketersediaan di *platform* seperti Google Play Store, aplikasi ini telah digunakan oleh ribuan pengguna dan mendapatkan berbagai ulasan yang mencerminkan beragam pengalaman, baik negatif maupun positif. Seiring meningkatnya jumlah pengguna, penting bagi pengembang dan pemangku kepentingan untuk memahami persepsi masyarakat terhadap aplikasi tersebut, guna meningkatkan kualitas layanan dan memperkuat kepercayaan publik terhadap sistem digital pemerintah.

Salah satu metode untuk mengklasifikasikan dan mengidentifikasi sentimen, opini atau perasaan dalam teks adalah analisis sentimen, yang merupakan cabang dari pemrosesan bahasa alami (NLP) [3]. Salah satu metode untuk memahami kesan pengguna adalah analisis sentimen. Tingkat kepuasan, keluhan, dan harapan pengguna terhadap aplikasi SMILE Indonesia dapat diketahui dengan menganalisis evaluasi pengguna di Google Play Store. Informasi ini digunakan untuk membuat keputusan tentang pengembangan dan peningkatan fungsionalitas aplikasi.

Analisis sentimen telah menggunakan sejumlah metode, seperti Naïve Bayes dan SVM, yang terkenal karena keefektifannya dalam menangani data tekstual. Klasifikasi sentimen yang akurat dari data ulasan pengguna merupakan tujuan dari kedua pendekatan ini. Teknik-teknik ini dipilih karena keefektifannya dalam sejumlah studi sebelumnya yang meneliti opini publik berbasis teks, terutama dalam aplikasi untuk layanan digital.

Dalam studi tahun 2024, analisis terhadap ulasan pengguna aplikasi Halo BCA di Google Play Store dilakukan dengan menerapkan algoritma Naive Bayes, SVM, dan Random Forest. Hasilnya menunjukkan bahwa SVM mencapai akurasi tertinggi sebesar 87%, diikuti oleh Naive Bayes dengan akurasi 81% [4]. Sementara itu, penelitian lain pada tahun yang sama menelaah respons pengguna terhadap aplikasi Samsat Digital Nasional (SIGNAL) menggunakan metode Naive Bayes. Ulasan pengguna diklasifikasikan ke dalam tiga kategori utama, yaitu positif, netral, dan negatif [5].

Penelitian ini bertujuan untuk menilai sentimen ulasan pengguna terhadap aplikasi SMILE Indonesia dengan menggunakan dua metode klasifikasi yakni Naive Bayes dan SVM. Penelitian ini akan memberikan ringkasan menyeluruh tentang seberapa baik kinerja masing-masing metode dalam kategorisasi sentimen dengan membandingkan kinerja kedua sistem. Penelitian ini juga akan memberikan rekomendasi yang berguna bagi para pengembang aplikasi yang ingin meningkatkan dukungan pelanggan. Meskipun analisis sentimen terhadap aplikasi layanan publik telah banyak dilakukan, seperti pada aplikasi Halo BCA maupun SIGNAL, belum banyak studi yang secara spesifik membahas persepsi masyarakat terhadap sistem logistik kesehatan digital. Penelitian ini menjadi salah satu studi awal yang fokus pada aplikasi SMILE Indonesia, yang memiliki peran penting dalam distribusi vaksin dan perbekalan medis nasional. Dengan demikian, studi ini mengisi celah penelitian (*research gap*) dalam konteks evaluasi layanan publik berbasis *e-government*, khususnya dalam sektor kesehatan digital di Indonesia.

2. METODE PENELITIAN

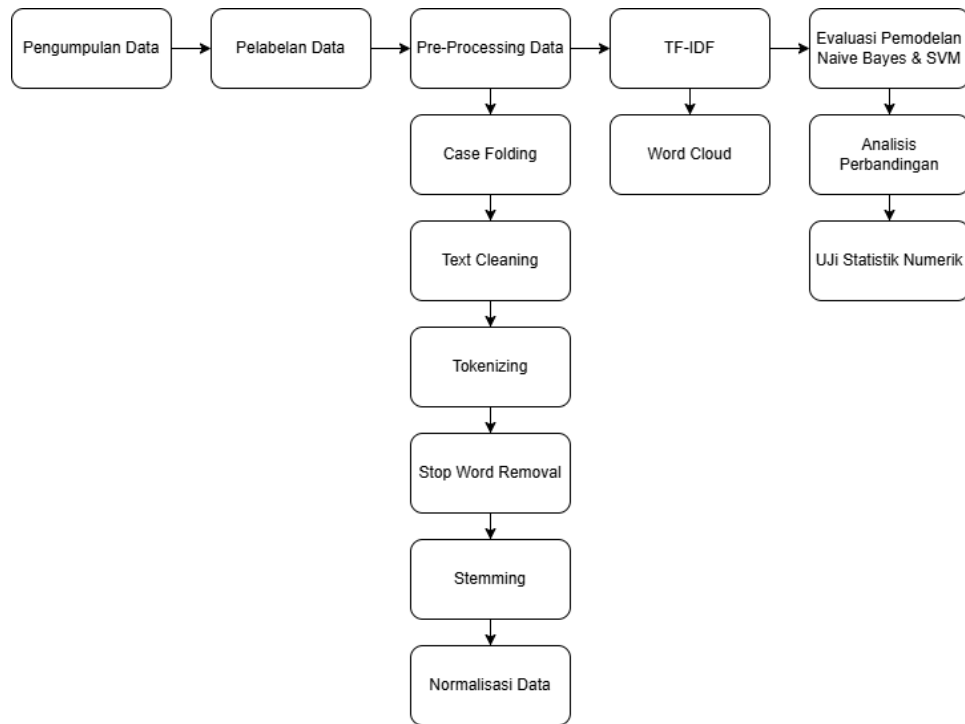
Tahap Penelitian

Ulasan pengguna aplikasi SMILE Indonesia dikumpulkan melalui metode *web scraping* dari Google Play Store. Ulasan tersebut kemudian diberi label sentimen secara otomatis berdasarkan skor rating pengguna, dengan skor 1 dan 2 dikategorikan sebagai negatif serta 3 hingga 5 sebagai positif.

Selanjutnya, data melalui proses *pre-processing* mencakup penghapusan tanda baca, simbol, dan emoji yang tidak relevan, *case folding*, tokenisasi, penghapusan *stopword*, *stemming*, serta normalisasi. Setelah itu, data dikonversi ke dalam representasi numerik menggunakan pendekatan Term Frequency–Inverse Document Frequency (TF-IDF).

Visualisasi awal dilakukan dengan Word Cloud untuk masing-masing kelas sentimen guna mengidentifikasi tema umum dalam opini pengguna. Proses pemodelan dilakukan dengan membandingkan dua algoritma klasifikasi, yaitu Multinomial Naïve Bayes dan Support Vector Machine (SVM). Evaluasi performa menggunakan metrik akurasi, presisi, *recall*, dan *f1-score*, dilengkapi dengan uji statistik McNemar untuk memastikan apakah perbedaan performa kedua model signifikan secara statistik.

Seluruh proses eksperimen dilakukan pada platform Google Colaboratory dengan bahasa pemrograman Python 3.10. Pustaka yang digunakan dalam eksperimen ini antara lain scikit-learn untuk pemodelan dan evaluasi, Sastrawi untuk stemming bahasa Indonesia, serta matplotlib, seaborn, dan wordcloud untuk visualisasi. Model Naïve Bayes menggunakan implementasi MultinomialNB dari scikit-learn tanpa parameter khusus, sedangkan model SVM menggunakan LinearSVC, yang sesuai untuk klasifikasi data teks berdimensi tinggi seperti hasil transformasi TF-IDF.



Gambar 1. Tahap Penelitian

Pengumpulan data

Data primer didapat dengan teknik *scraping* web dari ulasan aplikasi SMILE Indonesia yang ada di Google Play Store. Ulasan ini berisi 382 komen evaluasi tingkat kepuasan konsumen dari aplikasi yang akan memberikan gambaran bagaimana persepsi konsumen mengenai kualitas dari aplikasi SMILE Indonesia. analisis ini diharap dapat memberikan gambaran untuk meningkatkan kualitas dari aplikasi dan meningkatkan kepuasan para pengguna. Data yang telah diperoleh dari tahap *scraping* akan di ekspor ke dalam *file* dengan format CSV yang biasanya menjadi tipe *file* untuk data tipe tabel yang akan menjadi bahan untuk pemrosesan lebih lanjut.

Pelabelan Otomatis

Dalam penelitian ini, pelabelan otomatis dilakukan dengan membagi skor ulasan menjadi dua kategori: negatif (skor 1–2) dan positif (skor 3–5). Pendekatan ini diambil karena keterbatasan jumlah data dan untuk menjaga keseimbangan antar kelas dalam proses klasifikasi biner. Meskipun demikian, pendekatan ini memiliki potensi bias, terutama karena skor 3 sering kali dianggap sebagai sentimen netral atau ambivalen. Pengelompokan skor 3 ke dalam kategori positif dapat menyebabkan noise pada model, karena tidak semua ulasan dengan skor tersebut mengandung opini yang jelas positif. Namun, untuk menjaga kesinambungan tahapan eksperimen dan menghindari reduksi data yang signifikan, skor 3 tetap dimasukkan ke kategori positif, dengan catatan bahwa hal ini merupakan keterbatasan dalam studi dan perlu dipertimbangkan pada penelitian lanjutan dengan pendekatan multi-kelas (positif, netral, negatif) atau pelabelan manual yang lebih akurat.

Pre-Processing Data

Langkah pertama, yang dikenal sebagai *pre-processing*, mencoba mengubah masukan menjadi format yang lebih terstruktur agar sistem dapat memahami dan mempercepat analisis lebih lanjut [6]. Enam langkah meliputi *case folding*, membersihkan teks, membuat token, menghilangkan kata-kata yang tidak penting, *stemming*, dan normalisasi diselesaikan selama fase *pre-processing* data [7].

Case Folding

Teknik pengubahan semua kapital dalam suatu data menjadi huruf kecil atau *lowercase* dikenal sebagai *case folding*. Metode ini bertujuan untuk mencegah karakter huruf besar dan huruf kecil disalahartikan sebagai entitas yang berbeda. Misalnya, “Aplikasi” dan “app” akan dikenali sebagai satu kata setelah *case folding* [8].

Text Cleaning

Tahap pembersihan teks (*text cleaning*) adalah proses penghapusan karakter atau simbol yang tidak relevan terhadap analisis sentimen, seperti karakter ASCII yang tidak terbaca, angka, mention, tautan (link/url), hashtag, tanda baca, serta spasi berlebih. Tahap ini memastikan bahwa tidak ada komponen yang tidak perlu dalam data yang digunakan [8].

Tokenizing

Memecah kalimat menjadi bagian-bagian kecil yang terdiri dari kata atau token disebut tokenisasi. Pengolahan data teks dalam tahapan analisis berikutnya menjadi lebih mudah dengan proses ini. Sebagai contoh, kata "aplikasi yang sangat bagus" akan dipisahkan menjadi kata-kata "aplikasi", "yang", "sangat", "bagus", dan sebagainya [8].

Stopword Removal

Tahap penghapusan *stopword* ialah proses menghilangkan kata-kata yang umum digunakan dalam kalimat tetapi tidak memiliki arti yang sebenarnya. Penghapusan *stopword* sangat penting untuk mencegah kata-kata yang umum digunakan tetapi tidak memiliki arti penting mengganggu kapasitas sistem untuk mengkategorikan kelas sentimen. Kata-kata seperti "dan", "di", "dari", "ke", "pada", "itu", "ini", "adalah", dan seterusnya dihilangkan [8].

Stemming

Stemming adalah proses penghilangan awalan atau akhiran dan melakukan transformasi suatu kata pada dataset menjadi kata dasar. Tahap stemming dilakukan agar kata berimbuhan dengan kata dasar yang memiliki makna sama penulisannya seragam. Kata 'memudahkan' menjadi 'mudah', dan kata 'banyaknya' akan menjadi 'banyak' [8].

Normalize

Normalisasi adalah proses mengubah istilah gaul, kesalahan penulisan, dan singkatan menjadi istilah baku berdasarkan Kamus Besar Bahasa Indonesia (KBBI). Untuk memastikan penulisan yang konsisten dan identik, teknik normalisasi harus diterapkan pada kumpulan data yang diperoleh dari media sosial, di mana pengguna sering menggunakan terminologi dan singkatan yang tidak baku. [8].

TF-IDF (Term Frequency-Inverse Document Frequency)

Teknik untuk menentukan seberapa sering suatu kata muncul dalam sebuah dokumen atau artikel disebut TF-IDF (Term Frequency-Inverse Document Frequency). Tujuan utama metode ini adalah untuk menentukan makna suatu kata dalam konteks materi yang sedang dianalisis [9]. Dua komponen utama, TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*), digabungkan untuk membentuk TF-IDF. Sementara komponen TF menunjukkan seberapa sering suatu kata muncul dalam suatu dokumen, komponen IDF mengevaluasi relevansi istilah-istilah tersebut di seluruh koleksi dokumen [10].

Sebuah rumus untuk menghitung TF-IDF adalah sebagai berikut [11]:

$$TFIDF_{t,d} = TF_{t,d} \times \log\left(\frac{N}{DF_t}\right)$$

Dimana:

$TFIDF_{t,d}$: Bobot akhir kata t pada dokumen d

$TF_{t,d}$: Frekuensi kemunculan kata t dalam suatu dokumen d

N : Total dokumen dalam korpus

DF_t : Jumlah dokumen yang mengandung kata t

Word Cloud

Word cloud ialah pemvisualissian kata-kata yang paling umum dalam dataset, di mana ukuran setiap kata sesuai dengan frekuensinya. Semakin banyak frekuensi kata, semakin besar pula dimensinya pada tampilan word cloud, menandakan dominansi kata tersebut dalam data yang dianalisis. Tampilan ini menyajikan gambaran secara intuitif mengenai kata-kata yang paling awam atau menonjol pada setiap kelas sentimen, sehingga mempermudah pemahaman terhadap pola dan karakteristik sentimen dalam data tersebut [12].

Pembagian Data *Train* dan *Test*

Data dibagi menjadi dua bagian sebelum tahap pemodelan: 80% digunakan sebagai data pelatihan (*training*) dan 20% digunakan sebagai data uji (*testing*). Pembagian dilakukan secara terstratifikasi untuk mempertahankan rasio distribusi kelas di kedua subkelompok. Fungsi `train_test_split` dari modul Scikit-learn digunakan dengan opsi `random state` untuk memastikan replikasi hasil yang konsisten.

Naïve Bayes

Algoritma klasifikasi Naïve Bayes yakni teknik klasifikasi yang menerapkan konsep Teorema Bayes, yang mengasumsikan bahwa setiap variabel atau fitur dalam data adalah independen. Berdasarkan probabilitas kemunculan kata tertentu di setiap kelas, teknik analisis sentimen ini membagi konten ulasan ke kategori sentimen, seperti negatif atau positif [13].

Perhitungan probabilitas dalam Naïve Bayes mengacu pada rumus berikut [14]:



$$P(A|G) = \frac{P(G|A) \times P(A)}{P(G)}$$

Dimana:

$P(A|G)$: Probabilitas kelas A terhadap fitur G .

$P(G|A)$: Probabilitas fitur G muncul dalam kelas A .

$P(A)$: Probabilitas awal terjadinya kelas A .

$P(G)$: Probabilitas fitur G muncul secara keseluruhan dalam data.

Support Vector Machine (SVM)

Teknik klasifikasi yang disebut SVM merupakan teknik klasifikasi yang berupaya mengidentifikasi bidang pemisah atau *hyperplane* yang ideal untuk memisahkan dua kelas data secara efektif. Untuk setiap kelas (*support vector*), *hyperplane* dengan margin terbesar terhadap titik data terdekat dipilih. Fungsi kernel digunakan untuk menerjemahkan data ke ruang berdimensi lebih tinggi guna menemukan *hyperplane* pemisah jika data tidak dapat dipisahkan secara linear [15].

SVM bekerja dengan mengoptimalkan fungsi keputusan sebagai berikut [14]:

$$f(x) = \sum_{i=1}^{n_s} \alpha_i y_i K(\underline{x}_i, \underline{x}_d) + b$$

Dimana:

n_s : Jumlah *support vector*

α_i : Nilai untuk setiap titik data

y_i : Label kelas data ke- i

\underline{x}_i : Vektor *support vector* ke- i

\underline{x}_d : Data yang akan diklasifikasikan

$K(\underline{x}_i, \underline{x}_d)$: Fungsi kernel antara \underline{x}_i dan \underline{x}_d

b : Bias atau konstanta pemisah

Evaluasi Pemodelan

Untuk menentukan sejauh mana model memenuhi tujuan penelitian, evaluasi model merupakan langkah krusial dalam proses pembangunan model klasifikasi [16]. Penggunaan matriks kebingungan (*confusion matrix*) merupakan teknik populer untuk menilai kinerja model klasifikasi. Dengan membandingkan label aktual dengan prediksi model, matriks ini memberikan gambaran visual tentang hasil pengelompokan [17].

Confusion matrix atau matriks kebingungan pada Tabel 1 menggunakan empat komponen utama untuk menjelaskan hubungan antara nilai yang diharapkan dan nilai aktual: Ketika sistem memprediksi kelas yang seharusnya tidak ada, hal ini disebut sebagai *False Positive* (FP); ketika sistem berhasil mengidentifikasi data yang tidak termasuk dalam kelas target, hal ini disebut sebagai *False Negative* (FN); ketika sistem gagal mendeteksi kelas yang sebenarnya, hal ini disebut sebagai *True Negative* (TN); dan ketika prediksi dan label aktual keduanya akurat, hal ini disebut sebagai *True Positive* (TP) [18].

Tabel 1. *Confusion Matrix*

Aktual	Prediksi	
	Negatif	Positif
Negatif	TN	FP
Positif	FN	TP

Dari *confusion matrix* diatas, beberapa matrik evaluasi dapat dihitung, antara lain:

1. Akurasi (*Accuracy*) : Menggambarkan proporsi total prediksi yang benar terhadap seluruh jumlah prediksi yang dilakukan [19].

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

2. Presisi (*Precision*) : Mengukur seberapa akurat model dalam memprediksi suatu kelas tertentu, yakni proporsi data yang benar-benar relevan dari seluruh prediksi untuk kelas tersebut [19].

$$Precision = \frac{TP}{TP + FP}$$

3. *Recall* : Menjelaskan seberapa baik model dapat mengidentifikasi semua data yang relevan dalam suatu kelas tertentu, atau berapa banyak data yang diklasifikasikan dengan akurat yang sebenarnya termasuk dalam kelas tersebut [19].

$$Recall = \frac{TP}{TP + FN}$$

4. *f1-score* : Rata-rata harmonik antara presisi dan *recall* [19].

$$f1\ score = 2 \times \frac{precision \times recall}{precision + recall}$$

Uji Statistik McNemar

Uji McNemar merupakan metode statistik non-parametrik yang digunakan untuk membandingkan dua algoritma klasifikasi yang diuji pada dataset yang sama. Uji ini bekerja dengan menganalisis perbedaan prediksi antara dua model terhadap label yang sebenarnya, khususnya berfokus pada instance di mana kedua model menghasilkan prediksi yang berbeda. Dengan menggunakan pendekatan *continuity correction*, pengujian menghasilkan nilai statistik chi-square dan p-value. Jika p-value < 0,05, maka perbedaan antara kedua model dianggap signifikan secara statistik pada tingkat kepercayaan 95%.

3. HASIL DAN PEMBAHASAN

Scraping Data

Langkah pertama yakni melakukan *scraping* data. Berikut merupakan output yang diperoleh melalui proses *scraping* data:

	reviewId	userName	userImage	content	score	thumbsUpCount	reviewCreatedVersion	at	replyContent	repliedAt	appVersion
0	00591b6b-1ceb-4571-bcfe-02c701d22a6	fatma maya	lh.googleusercontent.com/a/ACg8oc...	Tolong untuk SO nya dipermudah. Stok 0 kenapa ...	1	4	3.7.8	2025-02-17 01:44:40	NaN	NaN	3.7.8
1	09254acd-d231-4a60-a23f-4606b49c8aa5	Fatmawaty Sulaiman	lh.googleusercontent.com/a/ACg8oc...	Kasian Farmasi di susahkan oleh banyaknya apli...	1	8	3.7.8	2025-01-23 13:32:40	NaN	NaN	3.7.8
2	872682b8-475d-46e6-8da5-801fc7234462	penabur kasih	lh.googleusercontent.com/a/ALVU...	Mohon maaf sebelumnya menurut saya dengan meng...	1	0	3.7.8	2025-02-07 06:33:26	NaN	NaN	3.7.8
3	0d8819eb-95b4-4477-8117-86ecf2ab5b1	Anna Siti Solihat	lh.googleusercontent.com/a/ALVU...	Dari pas awal aplikasi muncul udah ribet, seka...	1	14	3.7.0	2024-06-29 04:49:10	NaN	NaN	3.7.0
4	ea613889-a281-4d29-9fc2-926d08f53cf	erry dayusman	lh.googleusercontent.com/a/ACg8oc...	Aplikasi yg aneh..tdk efektif mestinya stok yg...	1	10	3.6.3	2024-06-02 11:15:08	NaN	NaN	3.6.3
...
378	d415b473-6618-4fe5-bdeb-984784788b05	Herman Herman	lh.googleusercontent.com/a/ALVU...	Bagus	5	1	NaN	2022-04-27 05:12:21	NaN	NaN	NaN
379	f9cea15c-917d-4fab-accf-3767b030bea0	Ika Alim	lh.googleusercontent.com/a/ACg8oc...	Bagus bingit	5	1	NaN	2021-09-12 17:52:20	NaN	NaN	NaN
380	b694d3c1-7c26-4e65-b613-2e8aaf711b56	Sigit Wasito	lh.googleusercontent.com/a/ACg8oc...	Sangat praktis	5	1	NaN	2021-04-21 16:02:49	NaN	NaN	NaN
381	f9bd903-2c6b-4ac7-b40c-b8f5883308ea	Mas Rudin	lh.googleusercontent.com/a/ALVU...	Coba dulu	5	0	NaN	2022-03-10 05:58:43	NaN	NaN	NaN
382	e998bbc0-25b0-4599-b6a3-a32d5f7e30c	Martinus Mone	lh.googleusercontent.com/a/ACg8oc...	👍	5	0	NaN	2024-07-23 05:22:57	NaN	NaN	NaN

383 rows x 11 columns

Gambar 2. Hasil Scraping Data Ulasan Aplikasi SMILE Indonesia

Program Python *google-play-scrapers* digunakan untuk melakukan *web scraping* guna mengumpulkan data ulasan pengguna untuk aplikasi SMILE Indonesia. Aplikasi ini terdaftar di Google Play Store dengan ID `com.logistikimunitas.mobile`. Melalui proses *scraping*, 382 titik data ulasan pengguna berhasil dikumpulkan. Selain informasi lain seperti nama pengguna dan versi aplikasi, data tersebut juga mencakup isi ulasan, skor penilaian, dan tanggal ulasan.

Pelabelan Otomatis

	content	score	sentiment
0	Tolong untuk SO nya dipermudah. Stok 0 kenapa ...	1	Negatif
1	Kasian Farmasi di susahkan oleh banyaknya apli...	1	Negatif
2	Mohon maaf sebelumnya menurut saya dengan meng...	1	Negatif
3	Dari pas awal aplikasi muncul udah ribet, seka...	1	Negatif
4	Aplikasi yg aneh..tdk efektif mestinya stok yg...	1	Negatif
...
378	Bagus	5	Positif
379	Bagus bingit	5	Positif
380	Sangat praktis	5	Positif
381	Coba dulu	5	Positif
382	👍	5	Positif

383 rows x 3 columns

Gambar 3. Hasil Pelabelan Otomatis Berdasarkan Skor Ulasan

Gambar 3 di atas menunjukkan hasil dari proses pelabelan otomatis. Pelabelan otomatis digunakan untuk membagi ulasan pengguna ke dalam kategori sentimen negatif dan positif berdasarkan skor rating, rating lebih dari sama dengan 3 dianggap positif, dan rating lebih rendah dari 3 dianggap negatif. Hasil pelabelan menghasilkan 196 ulasan pengguna yang dianggap positif dan 187 ulasan pengguna yang dianggap negatif.



Pre-Processing Case Folding

	content	case_folding
0	Tolong untuk SO nya dipermudah. Stok 0 kenapa ...	tolong untuk so nya dipermudah. stok 0 kenapa ...
1	Kasian Farmasi di susahkan oleh banyaknya apli...	kasian farmasi di susahkan oleh banyaknya apli...
2	Mohon maaf sebelumnya menurut saya dengan meng...	mohon maaf sebelumnya menurut saya dengan meng...
3	Dari pas awal aplikasi muncul udah ribet, seka...	dari pas awal aplikasi muncul udah ribet, seka...
4	Aplikasi yg aneh..tdk efektif mestinya stok yg...	aplikasi yg aneh..tdk efektif mestinya stok yg...
...
378	Bagus	bagus
379	Bagus bingit	bagus bingit
380	Sangat praktis	sangat praktis
381	Coba dulu	coba dulu
382	🙏	🙏

383 rows x 2 columns

Gambar 4. Hasil *Case Folding* pada Teks Ulasan

Untuk melakukan *case folding*, setiap huruf dalam data teks ulasan diubah menjadi *lowercase* atau huruf kecil. Hasil dari prosedur case folding ditampilkan pada kolom “case_folding” di Gambar 4. Semua huruf telah diubah menjadi huruf kecil. Kata-kata tidak dianggap berbeda satu sama lain dengan menstandarkan format teks.

Text Cleaning

	case_folding	clean_text
0	tolong untuk so nya dipermudah. stok 0 kenapa ...	tolong untuk so nya dipermudah stok kenapa mus...
1	kasian farmasi di susahkan oleh banyaknya apli...	kasian farmasi di susahkan oleh banyaknya apli...
2	mohon maaf sebelumnya menurut saya dengan meng...	mohon maaf sebelumnya menurut saya dengan meng...
3	dari pas awal aplikasi muncul udah ribet, seka...	dari pas awal aplikasi muncul udah ribet sekar...
4	aplikasi yg aneh..tdk efektif mestinya stok yg...	aplikasi yg anehtdk efektif mestinya stok yg s...
...
378	bagus	bagus
379	bagus bingit	bagus bingit
380	sangat praktis	sangat praktis
381	coba dulu	coba dulu
382	🙏	

383 rows x 2 columns

Gambar 5. Hasil Pembersihan Teks Ulasan (*Text Cleaning*)

Output dari proses membersihkan teks ditunjukkan pada Gambar 5 di atas. Teks ulasan dibersihkan dari emoji, angka, tanda baca, spasi ganda, huruf berulang, dan elemen yang tidak relevan atau mengganggu analisis. Data teks dibersihkan untuk mempersiapkan tahapan analisis berikutnya.

Tokenizing

	clean_text	tokenized
0	tolong untuk so nya dipermudah stok kenapa mus...	[tolong, untuk, so, nya, dipermudah, stok, ken...
1	kasian farmasi di susahkan oleh banyaknya apli...	[kasian, farmasi, di, susahkan, oleh, banyakny...
2	mohon maaf sebelumnya menurut saya dengan meng...	[mohon, maaf, sebelumnya, menurut, saya, denga...
3	dari pas awal aplikasi muncul udah ribet sekar...	[dari, pas, awal, aplikasi, muncul, udah, ribe...
4	aplikasi yg anehtdk efektif mestinya stok yg s...	[aplikasi, yg, anehtdk, efektif, mestinya, sto...
...
378	bagus	[bagus]
379	bagus bingit	[bagus, bingit]
380	sangat praktis	[sangat, praktis]
381	coba dulu	[coba, dulu]
382		[]

383 rows x 2 columns

Gambar 6. Hasil Tokenisasi Teks Ulasan

Hasil yang terlihat pada Gambar 6 di atas merupakan hasil dari proses tokenisasi. Tokenisasi digunakan untuk membagi konten ulasan menjadi token, yang merupakan unit kata. Tokenisasi dilakukan dengan menggunakan spasi untuk memisahkan kata-kata. Sebelum langkah analisis selanjutnya, proses ini berusaha untuk mengubah

data teks menjadi format yang lebih terstruktur. Setiap ulasan akan diubah menjadi daftar kata yang dapat dianalisis lebih lanjut pada langkah stemming atau modeling menggunakan teknik tokenisasi ini.

Stopwords Removal

	clean_text	tokenized
0	tolong untuk so nya dipermudah stok kenapa mus...	[tolong, untuk, so, nya, dipermudah, stok, ken...
1	kasian farmasi di susahkan oleh banyaknya apli...	[kasian, farmasi, di, susahkan, oleh, banyakny...
2	mohon maaf sebelumnya menurut saya dengan meng...	[mohon, maaf, sebelumnya, menurut, saya, denga...
3	dari pas awal aplikasi muncul udah ribet sekar...	[dari, pas, awal, aplikasi, muncul, udah, ribe...
4	aplikasi yg anehtdk efektif mestinya stok yg s...	[aplikasi, yg, anehtdk, efektif, mestinya, sto...
...
378	bagus	[bagus]
379	bagus bingit	[bagus, bingit]
380	sangat praktis	[sangat, praktis]
381	coba dulu	[coba, dulu]
382		[]

383 rows x 2 columns

Gambar 7. Hasil Penghapusan *Stopwords* pada Teks Ulasan

Hasil yang ditampilkan pada Gambar 7 di atas merupakan hasil dari proses penghapusan *stopwords*. Istilah umum (*stopwords*) seperti “yang,” “dan,” “di,” dan sebagainya yang tidak penting bagi proses analisis telah dihapus pada langkah ini. Dalam penelitian ini, digunakan file eksternal dan beberapa kata tambahan dalam kode perangkat lunak untuk menyesuaikan daftar *stopwords* dengan konteks bahasa Indonesia. Dalam proses analisis sentimen, penghapusan *stopwords* membantu menghilangkan istilah-istilah yang kurang relevan dan signifikan.

Stemming

	no_stopwords	stemmed
0	[tolong, so, dipermudah, stok, musti, stok, op...	[tolong, so, mudah, stok, musti, stok, opnameb...
1	[kasian, farmasi, susahkan, banyaknya, aplikas...	[kasi, farmasi, susah, banyak, aplikasi, yg, b...
2	[mohon, maaf, apk, mempermudah, pekerjaan, kar...	[mohon, maaf, apk, mudah, kerja, karna, ne, in...
3	[aplikasi, muncul, udah, ribet, ribet, udah, m...	[aplikasi, muncul, udah, ribet, ribet, udah, m...
4	[aplikasi, yg, anehtdk, efektif, mestinya, sto...	[aplikasi, yg, anehtdk, efektif, mesti, stok, ...
...
378	[bagus]	[bagus]
379	[bagus, bingit]	[bagus, bingit]
380	[praktis]	[praktis]
381	[coba]	[coba]
382	[]	[]

383 rows x 2 columns

Gambar 8. Hasil Proses *Stemming* pada Teks Ulasan

Hasil operasi *stemming* ditampilkan pada Gambar 8 di atas. Pada tahap ini, perpustakaan Sastrawi yang sering digunakan untuk teks yang ditulis dalam bahasa Indonesia digunakan untuk melaksanakan prosedur *stemming*. Untuk memungkinkan model secara andal mengidentifikasi makna dasar setiap kata, setiap token dari prosedur sebelumnya dikurangi menjadi bentuk dasarnya.

Normalisasi Data

	stemmed	normalized
0	[tolong, so, mudah, stok, musti, stok, opnameb...	[tolong, so, mudah, stok, musti, stok, opnameb...
1	[kasi, farmasi, susah, banyak, aplikasi, yg, b...	[kasi, farmasi, susah, banyak, aplikasi, yang...
2	[mohon, maaf, apk, mudah, kerja, karna, ne, in...	[mohon, maaf, aplikasi, mudah, kerja, karna, n...
3	[aplikasi, muncul, udah, ribet, ribet, udah, m...	[aplikasi, muncul, sudah, ribet, ribet, sudah,...
4	[aplikasi, yg, anehtdk, efektif, mesti, stok, ...	[aplikasi, yang, aneh tidak, efektif, mesti, s...
...
378	[bagus]	[bagus]
379	[bagus, bingit]	[bagus, banget]
380	[praktis]	[praktis]
381	[coba]	[coba]
382	[]	[]

383 rows x 2 columns

Gambar 9. Hasil Normalisasi Teks Ulasan

Setelah selesainya proses stemming, langkah berikutnya adalah normalisasi data. Kolom “normalized” pada Gambar 9 menampilkan hasil dari proses normalisasi data, di mana istilah-istilah non-standar, kesalahan, dan singkatan telah diganti dengan kata-kata standar yang sesuai dengan peraturan bahasa Indonesia. Hal ini sangat penting untuk membuat data yang digunakan dalam pemodelan lebih konsisten dan lebih mudah diinterpretasikan.

Normalisasi dilakukan dengan menggunakan kamus berbasis dictionary yang berisi pasangan kata tidak baku dan padanannya dalam bentuk baku. Proses ini diterapkan pada hasil *stemming* berupa token-token kata. Setiap token yang terdapat dalam kamus akan digantikan dengan padanan katanya. Jika token tidak ditemukan dalam kamus, maka token tersebut tetap digunakan tanpa perubahan. Hasil dari normalisasi ini disimpan dalam bentuk token yang telah diperbarui serta dalam bentuk kalimat akhir yang sudah siap untuk digunakan dalam tahap ekstraksi fitur atau vektorisasi.

TF-IDF (Term Frequency-Inverse Document Frequency)

Setelah semua prosedur *pre-processing* selesai, teks masukan harus diubah menjadi representasi numerik menggunakan pendekatan TF-IDF. Teknik ini menghitung nilai relatif setiap kata dibandingkan dengan kata-kata lain dalam sebuah dokumen. Transformasi ini dilakukan menggunakan paket *TfidfVectorizer* dari *Scikit-learn*. Sebanyak 769 karakteristik atau kata unik berhasil diidentifikasi setelah 382 dokumen yang dianalisis menjalani proses tokenisasi, stemming, dan normalisasi.

Tabel 2 di bawah ini menampilkan sepuluh kata dengan bobot TF-IDF tertinggi dalam kumpulan ulasan pengguna aplikasi SMILE Indonesia. Kata-kata ini dianggap paling bermakna secara statistik karena memiliki nilai pembobotan tertinggi dalam keseluruhan korpus.

Tabel 2. Sepuluh Kata dengan Bobot TF-IDF Tertinggi

Kata	Bobot TF-IDF
bagus	26,468
aplikasi	22,975
oke	22,326
bantu	10,758
tidak	10,727
vaksin	10,249
masuk	10,142
mantap	8,916
daftar	8,467
mudah	8,272

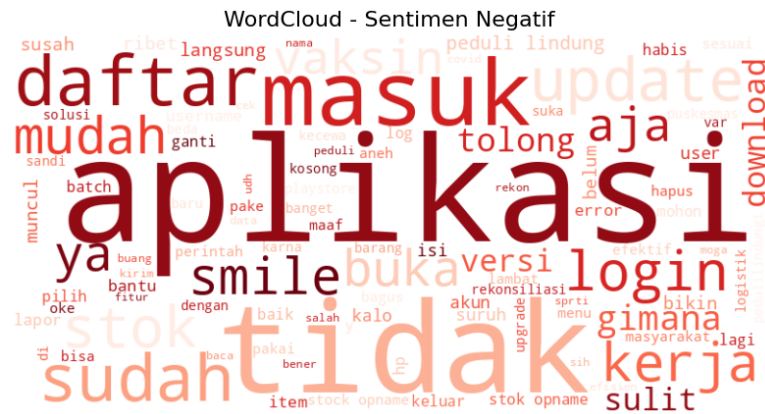
Visualisasi *Word Cloud*

Visualisasi frekuensi kata dalam ulasan pengguna terhadap aplikasi Smile Indonesia ditampilkan dalam bentuk *word cloud*. Ukuran tiap kata dalam *word cloud* merepresentasikan frekuensi kemunculannya dalam ulasan kata-kata yang lebih sering disebutkan akan muncul dengan ukuran font yang lebih besar, sehingga mudah diidentifikasi secara visual.



Gambar 10. *Word Cloud* Opini Positif

Berdasarkan *word cloud* pada opini positif (Gambar 10), terlihat bahwa kata-kata seperti "bagus", "mantap", "mudah", dan "bantu" mendominasi. Hal ini mengindikasikan bahwa pengguna merasa aplikasi membantu proses distribusi logistik secara praktis dan efisien, serta mudah digunakan. Kata “vaksin” yang cukup menonjol menunjukkan bahwa fitur distribusi vaksin merupakan aspek penting yang diperhatikan pengguna dalam penilaian mereka.



Gambar 11. *Word Cloud* Opini Negatif

Sementara itu, pada opini negatif (Gambar 11), kata-kata seperti "tidak", "masuk", "tolong", dan “sulit” muncul sebagai kata dominan. Ini mencerminkan adanya keluhan terkait masalah teknis, khususnya dalam proses login atau pendaftaran. Penggunaan kata “tidak” secara berulang juga mempertegas intensitas keluhan. Temuan ini memberikan wawasan bahwa kendala teknis menjadi faktor utama yang perlu diperbaiki pengembang aplikasi.

Dengan demikian, visualisasi *word cloud* tidak hanya menunjukkan frekuensi kata, tetapi juga memberikan gambaran umum mengenai tema utama dalam persepsi pengguna, baik dalam bentuk pujian maupun kritik. Temuan ini mendukung pentingnya analisis sentimen sebagai alat evaluasi layanan publik digital secara langsung dari pengalaman pengguna.

Hasil Evaluasi

Hasil Evaluasi Naïve Bayes

MultinomialNB Accuracy: 0.7922077922077922
 MultinomialNB Precision: 0.80563355633557
 MultinomialNB Recall: 0.7922077922077922
 MultinomialNB f1_score: 0.7941265301046422

```
confusion_matrix:  
[[35 11]  
 [ 5 26]]
```

	precision	recall	f1-score	support
Negatif	0.88	0.76	0.81	46
Positif	0.70	0.84	0.76	31
accuracy			0.79	77
macro avg	0.79	0.80	0.79	77
weighted avg	0.81	0.79	0.79	77

Gambar 12. Hasil Evaluasi Metode Naïve Bayes

Untuk mengklasifikasikan sentimen ulasan pengguna aplikasi SMILE Indonesia, hasil evaluasi model Multinomial Naïve Bayes ditunjukkan pada Gambar 12. Model ini mencapai akurasi 0,79 (79%), nilai presisi 0,81, nilai *recall* 0,79, dan nilai *f1-score* 0,79. Nilai-nilai ini menunjukkan bahwa model tidak hanya memiliki kecenderungan dominan terhadap satu kelas, tetapi juga bekerja dengan cukup seimbang dan stabil.

Metode ini berhasil mendeteksi 35 data negatif dan 26 data positif, seperti yang ditunjukkan dalam Tabel 3. Namun, 11 data negatif dan 5 data positif salah diklasifikasikan sebagai positif dan negatif, masing-masing. Akibatnya, model kami menunjukkan kemampuan yang memadai untuk mengidentifikasi ulasan positif dan negatif (*recall* 76%).

Tabel 3. Hasil *Confusion Matrix* Naïve Bayes

Aktual	Prediksi	
	Negatif	Positif
Negatif	35	11
Positif	5	26

Hasil Evaluasi Support Vector Machine (SVM)

```

SVM Accuracy: 0.7532467532467533
SVM Precision: 0.7547077922077923
SVM Recall: 0.7532467532467533
SVM f1_score: 0.7538493252778966
confusion_matrix:
[[36 10]
 [ 9 22]]
=====
              precision    recall  f1-score   support

   Negatif      0.80      0.78      0.79        46
   Positif      0.69      0.71      0.70        31

 accuracy              0.75        77
 macro avg      0.74      0.75      0.74        77
 weighted avg   0.75      0.75      0.75        77

```

Gambar 13. Hasil Evaluasi Metode SVM

Gambar 13 menampilkan hasil evaluasi model klasifikasi berbasis SVM. Model ini memperoleh *f1-score* sebesar 0,75, akurasi 0,75 (75%), presisi 0,75, dan *recall* 0,75. Model ini tetap mempertahankan kinerja yang kompetitif dalam klasifikasi dua kelas, meskipun nilai akurasinya sedikit lebih rendah daripada Naive Bayes.

Tabel 4 menunjukkan bahwa pendekatan SVM secara akurat mengidentifikasi 22 titik data positif dan 36 titik data negatif. Sepuluh item data negatif secara keliru dikategorikan sebagai positif, dan sembilan titik data positif secara keliru diklasifikasikan sebagai negatif. Hal ini menunjukkan bahwa model SVM sedikit lebih unggul daripada Naive Bayes dalam mengidentifikasi titik data positif, tetapi lebih unggul dalam mengidentifikasi titik data negatif (*recall* 78%).

Tabel 4. Hasil *Confusion Matrix* SVM

Aktual	Prediksi	
	Negatif	Positif
Negatif	36	10
Positif	9	22

Analisis Perbandingan

Pada Tabel 5 membandingkan matriks evaluasi antara pendekatan Naive Bayes dan SVM. Pada penelitian ditemukan bahwa pendekatan Naive Bayes unggul sedikit dengan nilai akurasi 0,79 dibandingkan dengan SVM dengan nilai akurasi 0,75.

Dari sisi presisi, Naive Bayes mencatat nilai 0,81, sedikit lebih tinggi dibandingkan dengan SVM yang memiliki nilai presisi sebesar 0,75. Ini mengindikasikan bahwa prediksi sentimen negatif atau positif dari Naive Bayes lebih akurat dibandingkan SVM dalam hal menghindari kesalahan prediksi positif (*false positive*).

Untuk *recall*, yang mengukur sejauh mana model mampu mengenali seluruh *instance* dari masing-masing kelas, Naive Bayes juga mencatat nilai yang lebih baik sebesar 0,79, dibandingkan dengan SVM yang memiliki *recall* sebesar 0,75. Artinya, Naive Bayes lebih andal dalam menangkap sentimen yang sebenarnya, baik pada kelas negatif maupun positif.

Dari sisi *f1-score*, yaitu metrik yang menggabungkan presisi dan *recall* untuk menyajikan gambaran yang seimbang tentang performa model, Naive Bayes menunjukkan keunggulan dengan nilai 0,79, sedangkan SVM mencatat nilai 0,75. Dalam Naive Bayes, nilai *f1-score* yang lebih tinggi berarti model tersebut kurang mungkin melewatkan data penting (*recall*) dan lebih akurat dalam prediksinya (*precision*).

Tabel 5. Perbandingan Metode Naive Bayes dan SVM

Matrik Evaluasi	Naive Bayes	SVM
Akurasi	0,79	0,75
Presisi	0,81	0,75
<i>Recall</i>	0,79	0,75
<i>f1-score</i>	0,79	0,75

Secara algoritmis, keunggulan Naive Bayes dalam penelitian ini juga dapat dijelaskan dari sifat dasar algoritmanya yang mengasumsikan independensi antar fitur. Dalam konteks data teks yang telah diubah menjadi representasi TF-IDF, kata-kata dalam ulasan pengguna dianggap sebagai fitur yang relatif independen. Meskipun asumsi ini tergolong sederhana, justru sangat cocok untuk menangani data berbasis teks yang memiliki distribusi kata yang tersebar dan tidak saling tergantung secara eksplisit.

Selain itu, Naive Bayes dikenal sangat efisien pada dataset kecil hingga menengah seperti yang digunakan dalam penelitian ini (382 data). Algoritma ini tidak memerlukan parameter tuning yang kompleks, sehingga mampu mencapai performa optimal tanpa banyak konfigurasi. Sebaliknya, SVM membutuhkan pemilihan kernel

dan parameter regulasi yang lebih sensitif terhadap ukuran dataset dan kompleksitas data. Hal ini memungkinkan Naïve Bayes mencapai hasil klasifikasi yang relatif lebih stabil dan akurat dalam kasus ini.

Validasi Statistik

Untuk memperkuat klaim perbedaan performa antara algoritma Naïve Bayes dan Support Vector Machine (SVM), dilakukan uji statistik menggunakan uji McNemar. Uji ini bertujuan untuk mengetahui apakah terdapat perbedaan yang signifikan secara statistik antara prediksi kedua model terhadap data uji yang sama.

Berdasarkan Tabel 6, diperoleh nilai statistik *chi-square* sebesar 1,125 dengan p-value sebesar 0,289. Karena nilai $p > 0,05$, maka perbedaan performa kedua model tidak signifikan secara statistik pada tingkat kepercayaan 95%. Artinya, meskipun model Naïve Bayes memiliki metrik evaluasi (akurasi, presisi, *recall*, dan *f1-score*) yang lebih tinggi dibandingkan SVM, keunggulan tersebut belum dapat disimpulkan sebagai signifikan secara statistik.

Tabel 6. Perbandingan Metode Naïve Bayes dan SVM

Uji	<i>Chi-Square</i>	<i>p - values</i>
<i>McNemar</i>	1.125	0.289

Interpretasi Hasil

Hasil analisis sentimen terhadap aplikasi SMILE Indonesia menunjukkan bahwa persepsi positif dari pengguna banyak terkait dengan kemudahan penggunaan dan fungsi distribusi logistik yang efisien. Temuan ini selaras dengan teori kepuasan layanan publik, di mana kualitas layanan teknologi informasi yang baik dalam hal kemudahan akses, kecepatan, dan reliabilitas merupakan determinan utama kepuasan pengguna layanan pemerintah.

Sebaliknya, sentimen negatif yang didominasi oleh keluhan teknis seperti kesulitan *login* dan *error* aplikasi menunjukkan bahwa aspek ketersediaan layanan dan dukungan teknis masih perlu diperbaiki. Dalam konteks *e-government*, keberhasilan implementasi sistem digital seperti SMILE sangat bergantung pada kepercayaan dan persepsi masyarakat terhadap kualitas layanan berbasis teknologi.

Oleh karena itu, hasil penelitian ini tidak hanya menjadi masukan teknis untuk pengembang, tetapi juga menjadi refleksi penting bagi pemangku kebijakan bahwa digitalisasi layanan publik memerlukan pemeliharaan berkelanjutan terhadap kualitas teknis dan *user experience* agar sejalan dengan prinsip akuntabilitas, aksesibilitas, dan partisipasi publik dalam *e-government*.

4. SIMPULAN

Penelitian ini telah mengevaluasi sentimen pengguna terhadap aplikasi SMILE Indonesia, sebuah *platform* layanan publik digital yang berfungsi memantau distribusi logistik kesehatan secara *real-time*. Sebanyak 382 ulasan pengguna dari Google Play Store dikumpulkan dan dianalisis melalui tahapan pra-pemrosesan teks, transformasi fitur menggunakan TF-IDF, serta klasifikasi sentimen menggunakan dua model: Naïve Bayes dan Support Vector Machine (SVM).

Berdasarkan hasil evaluasi, model Naïve Bayes memperoleh nilai akurasi 79%, presisi 81%, *recall* 79%, dan *f1-score* 79%, sementara SVM mencatat angka 75% pada semua metrik tersebut. Meskipun Naïve Bayes menunjukkan performa metrik yang lebih tinggi, uji McNemar menghasilkan p-value sebesar 0,289 yang menunjukkan bahwa perbedaan tersebut tidak signifikan secara statistik.

Analisis word cloud menunjukkan bahwa sentimen positif terutama ditandai oleh kemudahan penggunaan dan kebermanfaatan fitur, sedangkan sentimen negatif banyak mengandung keluhan teknis seperti kesulitan login dan pesan kesalahan sistem. Temuan ini mengindikasikan bahwa persepsi pengguna terhadap layanan digital pemerintah dipengaruhi oleh pengalaman teknis yang langsung mereka alami.

Dalam konteks layanan publik dan *e-government*, hasil penelitian ini memperkuat pentingnya kualitas teknis, aksesibilitas, dan keandalan sistem dalam meningkatkan kepuasan masyarakat terhadap layanan digital pemerintah. Oleh karena itu, hasil analisis sentimen ini dapat menjadi dasar bagi pengembang dan pembuat kebijakan dalam merancang perbaikan teknis maupun peningkatan strategi komunikasi layanan, guna mendukung penerimaan dan keberlanjutan implementasi aplikasi SMILE Indonesia sebagai bagian dari sistem logistik kesehatan nasional.

Namun demikian, perlu dicatat bahwa pelabelan sentimen dalam studi ini dilakukan secara otomatis dengan mengelompokkan skor 1–2 sebagai negatif dan skor 3–5 sebagai positif. Pendekatan ini dipilih untuk menyederhanakan proses klasifikasi dan menjaga jumlah data tetap mencukupi. Meskipun demikian, skor 3 sering kali dianggap sebagai opini netral yang tidak sepenuhnya positif maupun negatif. Oleh karena itu, keputusan untuk memasukkan skor 3 ke dalam kelas positif dapat menimbulkan potensi bias dalam hasil klasifikasi. Keterbatasan ini menjadi catatan penting untuk penelitian selanjutnya agar mempertimbangkan pendekatan pelabelan yang lebih presisi, misalnya dengan menyertakan kelas netral atau pelabelan manual.

DAFTAR PUSTAKA

- [1] A. Aksenta, I. Irmawati, A. Ridwan, N. Hayati, S. Sepriano, H. Herlinah, A.T. Silalah, S.J. Pipin, I. Abdurrohman, Y. Boari, and S. Mardiana, "Literasi Digital: Pengetahuan & Transformasi Terkini Teknologi Digital Era Industri 4.0 dan Society 5.0," *PT. Sonpedia Publishing Indonesia*, vol. 1, no. 2. 2023.
- [2] "Dukung Transformasi Digital Kesehatan, Kemenkes Beri Penghargaan Bagi Para Inovator Kesehatan," Kementerian Kesehatan. Accessed: Jun. 05, 2025. [Online]. Available: <https://kemkes.go.id/id/dukung-transformasi-digital-kesehatan-kemenkes-beri-penghargaan-bagi-para-juara-inovasi-kesehatan>
- [3] N. Nurwanda, N. Suarna, and W. Prihartono, "Penerapan NLP (Natural Language Processing) dalam Analisis Sentimen Pengguna Telegram di Playstore," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 2, pp. 1841–1846, 2024, doi: 10.36040/jati.v8i2.8469.
- [4] S. A. S. Mola, D. L. B. Baun, I. O. Nunes, and M. M. A. R. Sani, "Analisis Sentimen Aplikasi Halo BCA Di Google Play Store Menggunakan Metode Naive Bayes, Support Vector Machine dan Random Forest," vol. 15, no. 1, pp. 69–79, 2024. doi: <https://doi.org/10.52972/hoaq.vol15no2.p69-79>.
- [5] I. F. Rahman, A. N. Hasanah, and N. Heryana, "Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digital Nasional (Signal) dengan Menggunakan Metode Naive Bayes Classifier," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, pp. 963–969, 2024, doi: 10.23960/jitet.v12i2.4073.
- [6] K. Ganesan, "All you need to know about text preprocessing for NLP and Machine Learning," *KDnuggets*. Accessed: Jun. 05, 2025. [Online]. Available: <https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html>.
- [7] M. K. Khoirul Insan, U. Hayati, and O. Nurdiawan, "Analisis Sentimen Aplikasi Brimo pada Ulasan Pengguna di Google Play Menggunakan Algoritma Naive Bayes," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 1, pp. 478–483, 2023, doi: 10.36040/jati.v7i1.6373.
- [8] A. Herdhianto, *Sentiment Analysis Menggunakan Naive Bayes Classifier (NBC) pada Tweet Tentang Zakat*. 2020.
- [9] P. H. Saputro, M. Aristin, and Dy. L. Tyas, "Klasifikasi Lagu Daerah Indonesia Berdasarkan Lirik Menggunakan Metode Tfidf dan Naive Bayes," *Jurnal Teknologi Informatika dan Terapan*, vol. 4, no. 1, pp. 45–50, 2017.
- [10] Z. Zhu, J. Liang, D. Li, H. Yu, and G. Liu, "Hot Topic Detection Based on a Refined TF-IDF Algorithm," *IEEE Access*, vol. 7, no. 6, pp. 26996–27007, 2019, doi: 10.1109/ACCESS.2019.2893980.
- [11] E. R. N. Mustaqim, U. Pagalay, and C. Crysdiyan, "Prediksi Tingkat Kepercayaan Masyarakat Terhadap Pilpres 2024 Menggunakan Tf-Idf dan Bow Menggunakan Metode Svm," *Jurnal Cahaya Mandalika*, vol. 5, no. 1, pp. 515-530, 2024.
- [12] J. A. Wibowo, V. C. Mawardi, and T. Sutrisno, "Visualisasi Word Cloud Hasil Analisis Sentimen Berbasis Fitur Layanan Aplikasi Gojek dengan Support Vector Machine," *Jurnal Serina Sains, Teknik dan Kedokteran*, vol. 2, no. 1, pp. 61–70, 2024, doi: 10.24912/jsstk.v2i1.32058.
- [13] Budi Harijanto, Yuri Ariyanto, and Luthfia Miftahurroifa, "Penerapan Algoritma Naive Bayes Untuk Klasifikasi Retensi Arsip," *J. Inform. Polinema*, vol. 4, no. 2, p. 155, 2018, doi: 10.33795/jip.v4i2.159.
- [14] H. Apriyani and K. Kurniati, "Perbandingan Metode Naive Bayes Dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes Melitus," *Jurnal Information Technology Ampera*, vol. 1, no. 3, pp. 133–143, 2020, doi: 10.51519/journalita.volume1.issue3.year2020.page133-143.
- [15] R. W. Pratiwi, S. F. H, D. Dairoh, D. I. Af'idah, Q. R. A, and A. G. F, "Analisis Sentimen Pada Review Skincare Female Daily Menggunakan Metode Support Vector Machine (SVM)," *Journal of Informatics Information System Software Engineering and Application*, vol. 4, no. 1, pp. 40–46, 2021, doi: 10.20895/inista.v4i1.387.
- [16] D. Muriyatmoko, A. Musthafa, and M. H. Wijaya, "Klasifikasi Profil Kelulusan Nilai AKPAM dengan Metode Decision Tree," *Seminar Nasional Sains dan Teknologi 2024 Fakultas Teknik Universitas Muhammadiyah Jakarta*, no. 3, pp. 1-9, April, 2024.
- [17] M. E. Ibrahim, "Analisis Sentimen Komentar Youtube Social Commerce Tiktok dengan Metode Lexicon Based dan Long Short-Term Memory," 2024.
- [18] M. K. Suryadewiansyah, T. Endra, and E. Tju, "Naive Bayes dan Confusion Matrix untuk Efisiensi Analisa Intrusion Detection System Alert," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 8, no. 2, pp. 81–88, 2022. doi: <https://orcid.org/0000-0001-5148-4891>.
- [19] V. Oktaviana Yamin, A. Tenriawaru, L. Ode Saidi, and G. Arviana Rahman, "Penerapan Naive Bayes Classifier dengan Algoritma Nazief dan Adriani Untuk Deteksi Hoaks," *Prosiding Seminar. Nasional Pemanfaatan Sains dan Teknologi Informasi*, vol. 1, no. 1, pp. 335–344, 2023.