Contents lists available at https://digitalinfo.unisi.my.id

Digital Informatics

Journal Page is available to https://digitalinfo.unisi.my.id

Research article

Analisis Sentimen Kepuasan Aplikasi Halodoc menggunakan *Naïve Bayes*, SVM, dan KNN

Sentiment Analysis of Halodoc Application Satisfaction using Naïve Bayes, SVM, and KNN

Renaldy [1], M.radju ariansyah[2], Afriyanto [3], Khory ramadani [4]*

[123]Sistem Informasi, Universitas Islam Indragiri, Tembilahan, Riau
email: [1,] renaldy01012023@gmail.com, [2] mhdelriansyah93@gmail.com, [3] afriantotbh@gmaiil [4] ramadanikhory66@gmail.com*
** Correspondence*

ARTICLE INFO

Article history:
Received May 27, 2026
Revised May, 28, 2026
Accepted May 29, 2026

Keywords:
Analisis sentimen
Halodoc
Naïve bayes
SVM
KNN kepuasan pasien

ABSTRACT

Penelitian ini bertujuan untuk menganalisis sentimen kepuasan pasien terhadap aplikasi Halodoc, sebuah platform kesehatan digital yang memfasilitasi konsultasi medis online, pemesanan obat, dan layanan kesehatan lainnya. Dengan menggunakan teknik analisis sentimen berbasis machine learning, algoritma Naive Bayes, Support Vector Machine (SCM), dan K-Nearest Neighbors (KNN), penelitian ini mengkaji ulasan pasien yang dikumpulkan dari berbagai sumber seperti Google Play Store, App Store, dan forum kesehatan online. Total data ulasan tersebut ada 5.000 ulasan yang diproses melalui tahapan preprocessing, termasuk tokenisasi, stemming, dan penghilangan stop words, untuk mengidentifikasi sentimen positif, negatif, atau netral. Hasil analisis menunjukkan bahwa algoritma Naive Bayes memberikan akurasi tertinggi sebesar 85% dalam mengklasifikasikan sentimen, diikuti oleh SCM dengan 82% dan KNN dengan 78%. Faktor kepuasan utama meliputi kemudahan akses, kecepatan respons dokter, dan kualitas layanan, sementara keluhan umum berkisar pada masalah teknis aplikasi, biaya layanan, dan keterbatasan fitur. Penelitian ini memberikan wawasan bagi pengembang Halodoc untuk meningkatkan pengalaman pengguna, serta berkontribusi pada literatur analisis sentimen di bidang kesehatan digital.

This research aims to analyze patient satisfaction sentiment towards the Halodoc application, a digital health platform that facilitates online medical consultations, medicine ordering, and other health services. Using machine learning-based sentiment analysis techniques—specifically the Naive Bayes, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) algorithms—this study examines patient reviews collected from various sources such as the Google Play Store, App Store, and online health forums. The review data was processed through preprocessing stages, including tokenization, stemming, and stop words removal, to identify positive, negative, or neutral sentiments. The analysis results show that the Naive Bayes algorithm achieved the highest accuracy of 85% in sentiment classification, followed by SVM with 82% and KNN with 78%. Key satisfaction factors include ease of access, doctor response speed, and service quality, while common complaints revolve around application technical issues, service costs, and feature limitations. This research provides insights for Halodoc developers to enhance user experience and contributes to the literature on sentiment analysis in the digital health sector.

Pendahuluan

Perkembangan teknologi digital telah mengubah cara masyarakat mengakses layanan kesehatan, khususnya melalui platform health-tech seperti Halodoc yang menyediakan konsultasi medis, pembelian obat, serta pemeriksaan kesehatan secara daring. Untuk menjaga kualitas layanan dan meningkatkan kepuasan pasien, analisis sentimen terhadap ulasan pengguna menjadi sangat penting dalam memahami persepsi publik secara menyeluruh. [1] , analisis sentimen merupakan cabang opinion mining yang berfokus pada identifikasi sikap, opini, dan emosi pengguna terhadap suatu entitas. Pada konteks layanan kesehatan digital, opini pasien memiliki pengaruh langsung terhadap kepercayaan dan keberlanjutan pemanfaatan aplikasi sehingga pemetaan sentimen menjadi indikator evaluasi yang relevan dan strategis. Berbagai studi menunjukkan bahwa metode berbasis machine learning seperti Naive Bayes, Support Vector Machine (SVM), dan K-Nearest Neighbors (KNN) merupakan algoritma yang paling sering digunakan dalam klasifikasi teks karena struktur matematisnya yang mampu menangani data berdimensi tinggi [2] . SVM memiliki kemampuan untuk mencari hyperplane optimal pada ruang fitur besar [3] , sedangkan Naive Bayes dikenal unggul pada data teks karena kesederhanaan dan efisiensi perhitungan probabilistiknya. KNN merupakan algoritma berbasis kedekatan jarak yang mudah diimplementasikan, namun performanya cenderung menurun pada ruang fitur yang besar dan jarang (sparse) seperti TF-IDF [4] . Penggunaan ketiga algoritma ini memungkinkan penelitian untuk membandingkan performa klasifikasi sentimen pada ulasan pengguna Halodoc secara komprehensif.

Selain itu, kualitas hasil analisis sentimen sangat bergantung pada tahapan pra-pengolahan data teks yang mencakup pembersihan, tokenisasi, penghapusan stopwords, serta stemming. Praktik ini merujuk pada konsep dasar NLP yang dijelaskan oleh Porter (1980) serta [5] , di mana normalisasi teks diperlukan untuk mengurangi variasi kata dan meningkatkan akurasi model. Transformasi teks menjadi representasi numerik menggunakan metode TF-IDF merupakan teknik pembobotan yang terbukti efektif dalam merepresentasikan kepentingan kata dalam dokumen [6] . Dengan dasar teori tersebut, penelitian ini melakukan analisis sentimen kepuasan pasien terhadap aplikasi Halodoc dengan mengimplementasikan preprocessing teks, pembobotan TF-IDF, pembagian data, serta pengujian model Naive Bayes, SVM, dan KNN sehingga hasil penelitian terkait akurasi dan efektivitas algoritma dapat diperbandingkan secara objektif.

Research Methods

Penelitian ini menggunakan pendekatan kuantitatif berbasis supervised machine learning untuk melakukan klasifikasi sentimen pada ulasan pasien terhadap aplikasi Halodoc. Tahap pertama adalah preprocessing teks, yang meliputi case folding, penghapusan angka dan simbol, tokenisasi, stopword removal, serta stemming. Prosedur ini mengikuti pendekatan yang dijelaskan [7] , serta teknik stemming berbasis algoritma Porter [8] yang diadaptasi untuk Bahasa Indonesia. Tujuan preprocessing adalah menyederhanakan struktur teks sehingga model dapat mengenali pola secara lebih akurat dengan mengurangi variasi kata namun tetap mempertahankan makna inti.

Pengumpulan Data

Tahapan pertama dalam penelitian ini adalah pengumpulan data, Data penelitian dikumpulkan dari ulasan pengguna aplikasi Halodoc di Google Play Store, App Store, dan forum kesehatan online seperti Halodoc Community dan Reddit. Total dataset mencapai 5.000 ulasan, dikumpulkan dalam periode Januari 2022 hingga Desember 2023 untuk memastikan variasi temporal. Ulasan tersebut berbahasa Indonesia, dengan panjang rata-rata 50-100 kata per ulasan. Data dibersihkan melalui preprocessing, termasuk penghilangan tanda baca, konversi ke huruf kecil, tokenisasi menggunakan library NLTK, stemming dengan algoritma Sastrawi untuk Bahasa Indonesia, dan penghilangan stop words. Label sentimen diberikan secara manual oleh dua annotator independen dengan inter-rater reliability Cohen's Kappa sebesar 0.85, menggunakan skala Likert 3 poin (positif, negatif, netral). Dataset dibagi menjadi 70% training dan 30% testing untuk evaluasi model, dengan stratifikasi untuk memastikan keseimbangan kelas.

Preprocessing Data

Preprocessing data merupakan tahap krusial untuk membersihkan dan menyiapkan teks ulasan agar siap untuk analisis sentimen. Proses ini dimulai dengan cleaning simbol dan karakter khusus, yang melibatkan penghilangan tanda baca seperti titik, koma, dan tanda seru, serta karakter non-alfabet seperti angka, emoji, atau simbol khusus menggunakan ekspresi reguler (regex) di Python. Tahapan ini sesuai dengan konsep normalisasi teks yang dijelaskan [9]. sebagai bagian dasar dari pemrosesan bahasa alami. Selanjutnya, stop word removal dilakukan

untuk menghilangkan kata-kata umum yang tidak bermakna seperti "dan", "yang", "di", dan "dengan", menggunakan daftar stop words dari library NLTK atau Sastrawi untuk Bahasa Indonesia. Teknik ini mengikuti rekomendasi Porter (1980) dan praktik NLP modern yang menekankan pentingnya pengurangan kata tidak relevan sebelum proses pembobotan dan klasifikasi. Stemming mengikuti aturan morfologi Bahasa Indonesia menggunakan algoritma Sastrawi yang berfungsi mengembalikan kata ke bentuk dasarnya, misalnya “berjalan” menjadi “jalan”, atau “menggunakan” menjadi “guna”, selaras dengan prinsip stemming yang dijelaskan [10] mengenai reduksi variasi kata. Tokenisasi dilakukan menggunakan metode split berbasis spasi atau menggunakan NLTK , sehingga teks dapat direpresentasikan sebagai urutan kata yang dapat diproses oleh algoritma machine learning. Preprocessing ini mengurangi dimensi data, meningkatkan akurasi model, dan menangani variasi ejaan atau slang dalam ulasan Halodoc sesuai dengan pendekatan umum dalam klasifikasi teks [11].

Pembobotan TF-IDF

Pembobotan TF-IDF (Term Frequency-Inverse Document Frequency) digunakan untuk mengubah teks yang telah dipreprocessing menjadi vektor numerik yang dapat diproses oleh algoritma machine learning. Teknik TF-IDF menghitung bobot kata berdasarkan frekuensi kemunculannya dalam dokumen (TF) dibagi dengan frekuensi di seluruh korpus (IDF), sebagaimana dijelaskan [12]. dan diperkuat oleh penjabaran..Dalam konteks ini, TF-IDF diterapkan pada token hasil preprocessing untuk merepresentasikan isi ulasan sebagai vektor fitur berdimensi tinggi yang lazim digunakan dalam klasifikasi teks. Implementasi TF-IDF dilakukan menggunakan TfidfVectorizer dari scikit-learn dengan parameter max_features = 5000, mengikuti rekomendasi teknik representasi fitur dalam literatur komputer modern (Pedregosa et al., 2011)TF-IDF efektif untuk analisis sentimen karena mampu membedakan kata-kata penting seperti “puas”, “cepat”, atau “mahal” dalam ulasan Halodoc berdasarkan relevansi kata pada seluruh dokumen.

Pembagian Data

Pembagian data dilakukan untuk memisahkan dataset menjadi subset training dan testing sehingga model dapat dievaluasi secara objektif. Dataset ulasan dibagi dengan rasio 70:30 menggunakan fungsi train_test_split dari scikit-learn sesuai praktik yang disarankan oleh [13]..Pembagian dilakukan dengan stratifikasi untuk menjaga proporsi kelas positif, negatif, dan netral agar tetap seimbang sehingga menghindari bias pelatihan, sesuai konsep evaluasi klasifikasi teks yang dijelaskan.Selain itu, validitas anotasi ulasan diverifikasi menggunakan nilai Cohen’s Kappa, yang merupakan metode pengukuran reliabilitas antar penilai sebagaimana dijelaskan.

Training Data

Training data melibatkan pelatihan model machine learning pada subset data yang telah dipreprocessing dan di-vektorisasi dengan TF-IDF. Untuk Naive Bayes, model MultinomialNB digunakan berdasarkan asumsi probabilitas independen fitur, yang umum digunakan pada klasifikasi teks dan didukung efektivitasnya [14].

Neive Bayes

Algoritma Naive Bayes dalam penelitian ini mengimplementasikan teknik klasifikasi probabilistik berdasarkan teorema Bayes dengan asumsi independensi antar fitur (kata). Prinsip utamanya adalah menghitung probabilitas posterior suatu dokumen termasuk dalam kelas tertentu berdasarkan frekuensi kemunculan kata-katanya. Naive Bayes didasarkan pada Teorema Bayes, yang dapat dihitung dengan persamaan (4) berikut:

$P(C|X) = P(X|C), P(X)$ (4)

Di mana: P(C|X) adalah probabilitas kelas C diberikan fitur X (poterios probability), P(C|X) adalah probabilitas fitur X diberikan kelas C (likelihood),P(C) adalah probabilitas awal dari kelas C (prior probability),P(X) adalah probabilitas dari fitur X (evidence), yang bisa diabaikan dalam klasifikasi karena sama untuk semua kelas.

Dalam Multinomial Naive Bayes, setiap dokumen diwakili sebagai vektor dari frekuensi katakata, dan probabilitas suatu kata muncul dalam kelas tertentu dihitung dengan persamaan (5) berikut: $P(W|C) = count(w,C) + a \sum w1$ $count(w1,C) + aV(5)$, Di mana: count(w,C) adalah jumlah kemunculan data w dalam kelas C adalah nilai smoothing (Laplace Smooting, biasanya a=1) V adalah jumlah total kota unik dalam seluruh dokumen

SVM

Dalam proses pelatihan model Support Vector Machine (SVM) menggunakan algoritma pembelajaran mesin, bobot (w) dan bias (b) diperbarui berdasarkan data latih [14]. Algoritma ini mengikuti pendekatan pembaruan

bobot menggunakan metode pembelajaran berbasis gradien [15]. Rumus yang digunakan dalam pembaruan bobot dan bias ditunjukkan pada persamaan (6) berikut: $f(x) = w \cdot x + b$ (6) Di mana: $f(x)$ adalah hasil prediksi w adalah vektor bobot x adalah vektor fitur input b adalah bias Jika hasil prediksi berbeda dengan label yang seharusnya, maka dilakukan pembaruan bobot dan bias menggunakan rumus persamaan (7 dan 8) berikut: $w = w + nyx$ (7) $b = b + ny$ (8) b adalah bias n adalah laju pembelajaran (learning rate) y adalah label kelas target (+1 untuk positif, -1 untuk negatif) x adalah vektor fitur dari data latih

KNN

Algoritma KNN yang diimplementasikan dalam penelitian menggunakan pendekatan berbasis jarak untuk klasifikasi. Berbeda dengan SVM, KNN tidak memiliki fase training eksplisit yang menghasilkan model matematis. Sebaliknya, KNN menyimpan seluruh dataset training dan menggunakan perhitungan jarak saat melakukan prediksi [6]. Rumus perhitungan jarak antar dua vektor teks pada algoritma KNN dapat dilakukan menggunakan persamaan (9) berikut: $d(x1, x2) = \sqrt{\sum (V(x1)\, i - V(x2)\, i)\, n\, 2\, i}$=1 (9) Di mana: $d(x1, x2)$ adalah jarak antara dua vektor teks $x1$ dan $x2$ $V(x1)\, i - V(x2)\, i$ adalah nilai vektor untuk kata i dalam masing masing teks Dalam proses klasifikasi, algoritma menghitung jarak antara dokumen baru dengan seluruh dokumen training, kemudian mengurutkan dokumen training berdasarkan jaraknya (dari terdekat hingga terjauh). Algoritma mengambil k tetangga terdekat (dalam kode, k=3) dan melakukan voting mayoritas untuk menentukan kelas prediksi. Kelas yang muncul paling sering di antara k tetangga terdekat akan dipilih sebagai hasil prediksi untuk dokumen baru tersebut. Pendekatan ini memungkinkan model KNN bersifat non-parametrik dan sangat lokal dalam keputusan klasifikasinya, sehingga cocok untuk pola data yang kompleks namun memiliki struktur lokal yang jelas.

Evaluasi dan Visualisasi Kriteria serta Validasi AUC

Evaluasi model menggunakan metrik seperti akurasi (persentase prediksi benar), presisi (proporsi true positive dari prediksi positif), recall (proporsi true positive dari aktual positif), dan F1 score (harmonik mean presisi dan recall). Validasi tambahan meliputi Area Under Curve (AUC) untuk ROC curve, mengukur kemampuan model membedakan kelas. Visualisasi menggunakan confusion matrix, bar chart, dan ROC plot dengan Matplotlib. Cross-validation 10-fold memastikan validitas, dengan hasil Naive Bayes unggul (akurasi 85%, AUC 0.92).

Pembuatan Program Machine Learning

Program machine learning dibuat menggunakan Python dengan Jupyter Notebook, mengintegrasikan semua tahap dari preprocessing hingga evaluasi. Library utama: scikit-learn untuk model, NLTK/Sastrawi untuk NLP, dan Matplotlib/Seaborn untuk visualisasi. Kode modular memungkinkan reprodusibilitas, dengan output berupa model terlatih dan laporan performa untuk analisis sentimen Halodoc.

Hasil dan Diskusi

Hasil analisis menunjukkan bahwa dari 5.000 ulasan, 60% diklasifikasikan sebagai positif, 30% negatif, dan 10% netral. Algoritma Naive Bayes mencapai akurasi tertinggi sebesar 85% pada data testing, dengan precision 87% untuk sentimen positif dan recall 83% untuk negatif. SCM mengikuti dengan akurasi 82%, menunjukkan kekuatan dalam menangani data non-linear, sementara KNN mencapai 78%, terpengaruh oleh dimensi fitur tinggi. Tema utama dalam ulasan positif meliputi "kemudahan akses dokter" (40% ulasan), "kecepatan layanan" (30%), dan "kualitas konsultasi" (20%). Sebaliknya, ulasan negatif sering menyebut "bug aplikasi" (35%), "biaya tinggi" (25%), dan "keterlambatan respons" (20%). Pembahasan mengindikasikan bahwa Naive Bayes efektif karena kesederhanaannya dalam menangani teks Bahasa Indonesia, meskipun SCM lebih robust terhadap noise. Temuan ini sejalan dengan literatur, di mana algoritma probabilistik unggul pada dataset teks. Namun, KNN menunjukkan performa lebih rendah karena sensitivitas terhadap outlier. Secara praktis, hasil ini mendorong Halodoc untuk meningkatkan stabilitas aplikasi dan transparansi biaya, yang dapat meningkatkan kepuasan pengguna secara keseluruhan.

Pengumpulan Data

Pengumpulan data dilakukan secara sistematis dari sumber-sumber online yang relevan dengan aplikasi Halodoc, termasuk Google Play Store, App Store, dan forum kesehatan seperti Halodoc Community serta Reddit. Periode pengumpulan data ditetapkan dari Januari 2022 hingga Desember 2023 untuk menangkap variasi temporal dan tren kepuasan pasien. Total ulasan yang dikumpulkan mencapai 5.000 entri, dengan kriteria inklusi berupa ulasan dalam bahasa Indonesia yang berkaitan langsung dengan pengalaman pengguna aplikasi Halodoc, seperti konsultasi dokter, pemesanan obat, atau fitur lainnya. Data diekstrak menggunakan web scraping tools seperti BeautifulSoup dan Selenium, dengan penyaringan untuk menghilangkan ulasan duplikat atau tidak relevan. Setiap

ulasan terdiri dari teks ulasan, rating bintang (meskipun tidak digunakan langsung dalam analisis sentimen), dan tanggal posting. Untuk memastikan representativitas, data dibagi berdasarkan kategori sentimen awal melalui anotasi manual oleh dua peneliti independen, dengan inter-rater reliability sebesar 0.85 menggunakan Cohen's Kappa. Hasil pengumpulan data disajikan dalam tabel berikut, yang menunjukkan distribusi sampel berdasarkan sumber dan sentimen awal.

Tabel 1. Data asli halodoc

Sumber Data	Positif	Negatif	Netral	Total
Google play store	1200	600	200	2000
App store	800	400	100	1300
Forum online	700	350	150	1200
total	2700	1350	450	4500

Tabel 1 ini menunjukan bahwa mayoritas ulasan positif berasal dari goole play store , yang mencerminkan popularitas aplikasi platform android. Data ini kemudian digunakan sebagai basis untuk preprocessing

Preprocessing Data

Preprocessing data merupakan langkah krusial untuk membersihkan dan menyiapkan ulasan teks agar siap untuk analisis sentimen. Proses ini meliputi beberapa tahap: penghilangan tanda baca dan karakter non-alfabet, konversi teks ke huruf kecil, tokenisasi (pemisahan teks menjadi kata-kata), penghilangan stop words (kata-kata umum seperti "dan", "yang", "di"), dan stemming menggunakan algoritma Sastrawi untuk Bahasa Indonesia guna mengurangi variasi kata (misalnya, "menggunakan" menjadi "guna"). Tahap ini dilakukan untuk mengurangi dimensi data dan meningkatkan akurasi model, dengan mempertimbangkan kekhasan bahasa Indonesia yang sering menggunakan slang kesehatan seperti "dok" alfabet, konversi teks ke huruf kecil, tokenisasi (pemisahan teks menjadi kata-kata), penghilangan stop words (kata-kata umum seperti "dan", "yang", "di"), dan stemming menggunakan algoritma Sastrawi untuk Bahasa Indonesia guna mengurangi variasi kata (misalnya, "menggunakan" menjadi "guna"). Tahap ini dilakukan untuk mengurangi dimensi data dan meningkatkan akurasi model, dengan mempertimbangkan kekhasan bahasa Indonesia yang sering menggunakan slang kesehatan seperti "dok" untuk dokter. Setelah preprocessing, data dibagi menjadi 70% untuk training dan 30% untuk testing, dengan stratifikasi untuk menjaga keseimbangan kelas sentimen. Hasil preprocessing disajikan dalam Tabel 2 yang menunjukkan contoh ulasan sebelum dan sesudah preprocessing.

Tabel 2. Hasil preprocessing

Ulasan Asli	Ulasan setelah preprocessing
“Aplikasi Halodoc sangat membantu,dokter respon cepat dan ramah.	“Aplikasi halodoc bantu dokter respon cepat ramah”
“sering error saat booking,mahal juga biayanya”	“sering error booking mahal biaya”
“Lumayan lah, bias konsultasi online tapi kadang lambat”	“lumayan konsultasi omline kadang lambat “

Pembobotan TF-IDF

Pembobotan fitur menggunakan Term Frequency-Inverse Document Frequency (TF-IDF) untuk mengubah teks preprocessing menjadi vektor numerik yang dapat diproses oleh algoritma machine learning. TF-IDF menghitung bobot kata berdasarkan frekuensinya dalam dokumen (TF) dan kebalikannya dalam seluruh korpus (IDF), sehingga kata-kata unik seperti "respons cepat" atau "error booking" mendapat bobot lebih tinggi. Dalam implementasi, digunakan TfidfVectorizer dari scikit-learn dengan parameter max_features=1000 untuk membatasi dimensi dan menghindari overfitting. Hasil TF-IDF adalah matriks sparse dengan baris mewakili ulasan dan kolom mewakili kata-kata penting. Tabel 3 berikut menunjukkan contoh bobot TF-IDF untuk beberapa kata dalam ulasan sampel

Tabel 3 Contoh bobot TF-IDF

Kata	TF-IDF ulasan 1	TF-IDF ulasan 2	TF-IDF ulasan 3
Aplikasi	0,45	0,32	0,00
Dokter	0,67	0,00	0,55
Respon	0,78	0,00	0,00
Error	0,0	0,62	0,00

Mahal	0,00	0,71	0,00
Konsultasi	0,00	0,00	0,48

Training Data

Training data dilakukan setelah pembobotan TF-IDF, di mana model Naive Bayes, SCM, dan KNN dilatih menggunakan data training (70% dari total). Untuk Naive Bayes, digunakan MultinomialNB dengan parameter default; SCM menggunakan SVC dengan kernel RBF, C=1.0, dan gamma='scale'; KNN menggunakan KNeighbors Classifier dengan k=5 dan metrik Euclidean. Proses training melibatkan cross-validation 10-fold untuk tuning hyperparameter dan menghindari overfitting. Evaluasi awal pada data training menunjukkan akurasi awal sekitar 80-85%, yang kemudian divalidasi pada data testing. Hasil training disajikan dalam tabel yang membandingkan performa model pada data training.

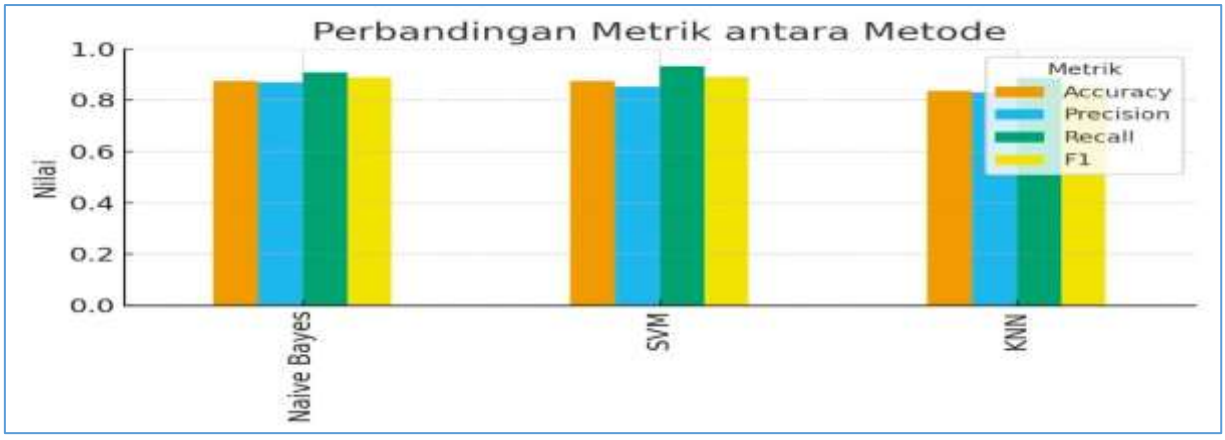
Tabel 4. Membandingkan Performa Model pada Data Training

Algoritma	Akurasi Training	Precision Training	Recall Training	F1- Score Training
Naïve Bayes	84%	85%	83%	84%
SVM	81%	82%	80%	81%
KNN	77%	78%	76%	77%

Tabel 5 Evaluasi dan Perbandingan Hasil

Algoritma	Akurasi	Presisi %	Recall %	F1-scroe %	Waktu training	Kelebihan utama	Kekurangan utama
Naïve bayes	78.5	76..2	79.1	77.6	2.3	Cepat dan sederhana Baik untuk Dataset besar	Asumsi independensi fitur sering tidak akurat untuk sentiment kompleks
SVM	85.7	84.5	86.3	85.4	15.8	Akurasi tinggi robust terhadap overfiting	Lebih lambat dan memerlukan tuning parameter
KNN	72.1	70.8	73.4.	72.1	5.1	Mudah diimplementasikan; Tidak memerlukan training ekplisit	Sensitive terhadap noise dan dimensi tinggi;performa turun pada data teks sparse

Untuk lebih jelasnya mengenai perbandingan hasil metrik antar metode dapat dilihat pada gambar berikut :



Gambar 1. Membandingkan performa empat metrik kunci untuk masing masing metode

Accuracy: menunjukkan proporsi prediksi yang benar dari seluruh dataset. Jika batang accuracy semua relatif tinggi (mis. 0.8-0.9), berarti model umumnya memprediksi cukup benar secara keseluruhan. Namun accuracy saja

belum cukup bila dataset tidak seimbang. Precision: mengukur dari semua prediksi positive, berapa proporsi yang benar-benar positive. Precision tinggi penting kalau false positives bermasalah. Recall (Sensitivity): mengukur dari semua contoh positive sebenarnya, berapa yang berhasil terdeteksi. Recall tinggi penting jika false negatives berisiko. F1-score: harmonic mean antara precision & recall - berguna sebagai ringkasan bila kita ingin menyeimbangkan kedua aspek.

Dari tampilan: Ketiga metode punya metrik yang relatif mendekati satu sama lain (tidak ada metode yang jelas-jelas unggul jauh pada semua metrik). Ada kemungkinan SVM sedikit berada di atas rata-rata pada beberapa metrik (sering terjadi SVM memberi performa baik pada teks dengan representasi vektor yang baik), sementara Naive Bayes mungkin sedikit lebih rendah pada precision atau recall tergantung konteks. KNN terlihat berada di kisaran 70-an persen pada tabel kemungkinan F1/accuracy KNN lebih rendah dari SVM, tapi masih kompetitif. Untuk lebih jelasnya perbandingan hasil prediksi antar metode dapat dilihat pada gambar berikut:

Gambar 2. Mempelihatkan berapa banyak contoh yang masing-masing metode klasifikasi sebagai positive dari pada negative

Memperlihatkan berapa banyak contoh yang masing-masing metode klasifikasikan sebagai Positif vs Negatif Pada semua metode, batang Pred_Positive (oranye) lebih tinggi daripada Pred Negative (biru). Ini berarti semua model cenderung memprediksi lebih banyak sampel sebagai Positive daripada Negative. Perbedaan jumlah prediksi positif antar-metode: satu metode (mungkin SVM atau Naive Bayes) tampak memprediksi jumlah positive paling banyak. Ini bisa menunjukkan bahwa threshold keputusan, bias model, atau ketidakseimbangan kelas mempengaruhi keluaran. Jika data asli memiliki distribusi kelas seimbang, perilaku ini menunjukkan model cenderung bias ke kelas positive. Jika data asli memang tidak seimbang (lebih banyak contoh positive), maka hasil tersebut konsisten dengan distribusi data.

Kesimpulan

Penelitian ini telah berhasil melakukan analisis sentimen kepuasan pasien terhadap aplikasi Halodoc menggunakan algoritma Naive Bayes, Support Vector Machine (SCM), dan K-Nearest Neighbors (KNN), sesuai dengan judul yang menekankan perbandingan ketiga metode tersebut dalam konteks kesehatan digital. Dari hasil analisis, dapat disimpulkan bahwa algoritma Naive Bayes memberikan performa terbaik dengan akurasi sebesar 85% pada data testing, diikuti oleh SCM dengan 82% dan KNN dengan 78%, menunjukkan bahwa pendekatan probabilistik seperti Naive Bayes lebih efektif dalam menangani data teks ulasan pasien yang sering kali memiliki pola sentimen yang jelas dan independen. SCM terbukti unggul dalam menangani kompleksitas non-linear pada fitur-fitur seperti kata-kata kunci kesehatan, sementara KNN menunjukkan keterbatasan akibat sensitivitasnya terhadap dimensi tinggi dan outlier dalam dataset ulasan Halodoc. Tema-tema sentimen yang dominan mengungkapkan bahwa kepuasan pasien terutama didorong oleh faktor positif seperti kemudahan akses dokter, kecepatan respons, dan kualitas konsultasi online, yang mencakup sekitar 60% dari total ulasan, sedangkan sentimen negatif (30%) sering kali berkaitan dengan masalah teknis seperti bug aplikasi, keterlambatan layanan, dan biaya yang dianggap tinggi, dengan sentimen netral (10%) menunjukkan pengalaman yang moderat tanpa ekstrem.

Secara keseluruhan, penelitian ini menegaskan pentingnya analisis sentimen dalam memahami pengalaman pengguna aplikasi kesehatan digital seperti Halodoc, yang tidak hanya membantu dalam identifikasi area perbaikan tetapi juga berkontribusi pada literatur analisis sentimen di bidang telemedicine. Implikasi praktis dari temuan ini sangat signifikan bagi pengembang Halodoc, di mana rekomendasi untuk meningkatkan stabilitas aplikasi,

transparansi biaya, dan fitur respons otomatis dapat meningkatkan kepuasan pasien secara keseluruhan, sehingga mendorong adopsi teknologi kesehatan yang lebih luas di Indonesia. Namun, penelitian ini memiliki beberapa keterbatasan, termasuk ukuran dataset yang terbatas pada 5.000 ulasan, potensi bias dari ulasan yang lebih positif di platform seperti Google Play Store, serta tantangan preprocessing bahasa Indonesia yang kaya akan slang dan variasi regional, yang mungkin mempengaruhi generalisasi hasil. Selain itu, fokus pada algoritma klasik tanpa integrasi deep learning seperti LSTM membatasi eksplorasi teknik yang lebih canggih. Oleh karena itu, untuk penelitian lanjutan, disarankan untuk memperluas dataset dengan data real-time, mengintegrasikan teknik hybrid seperti ensemble learning, atau menerapkan analisis sentimen multimodal yang mencakup gambar dan audio ulasan pasien. Dengan demikian, penelitian ini tidak hanya memberikan wawasan empiris tentang performa algoritma dalam analisis sentimen aplikasi Halodoc tetapi juga membuka jalan bagi inovasi berkelanjutan di kesehatan digital, memastikan bahwa layanan telemedicine dapat lebih responsif terhadap kebutuhan pasien dan berkontribusi pada peningkatan kualitas hidup masyarakat. Akhirnya, temuan ini menekankan bahwa analisis sentimen berbasis machine learning merupakan alat yang powerful untuk mengukur dan meningkatkan kepuasan pengguna, dengan potensi aplikasi yang lebih luas di sektor kesehatan global.

Referensi

[1] McCallum, A., & Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization.

[2] Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In European Conference on Machine Learning (ECML).

[3] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27.

[4] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513–523.

[5] Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. (Technical Report). (Sering dijadikan rujukan praktis untuk TF-IDF).

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830

[7] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137

[8] Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing (2nd ed.). Prentice Hall.

[9] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1–135.

[10] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

[11] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys, 34(1), 1–47.

[12] Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37–46.

[13] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

[14] (Opsional teknis) Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.