

ANALISIS SENTIMEN TERHADAP PENGGUNAAN APLIKASI MOBILE JKN DENGAN PENDEKATAN *RANDOM FOREST CLASSIFIER*

Zahra Sri Febrianti¹, Mariza Devega²

(^{1,2}Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Lancang Kuning)
Jl. Yos Sudarso KM. 8 Rumbai, Pekanbaru, Riau, telp. 0811 753 2015
e-mail: ¹zahrasrifebrianti20@gmail.com, ²marizadevega@unilak.ac.id

Abstrak

Kesehatan merupakan kebutuhan mendasar bagi seluruh masyarakat. Dalam upaya pemerataan layanan jaminan kesehatan di Indonesia, pemerintah menugaskan BPJS Kesehatan sebagai lembaga penyelenggara jaminan kesehatan nasional. Sebagai bentuk inovasi digital, BPJS Kesehatan menghadirkan aplikasi Mobile JKN guna mempermudah masyarakat dalam mengakses layanan kesehatan secara daring. Namun, implementasi aplikasi ini masih memunculkan berbagai keluhan, seperti kesulitan login, gangguan pembaruan, serta keterbatasan fitur, yang tercermin dari ulasan pengguna bernuansa negatif. Untuk menilai persepsi masyarakat secara lebih objektif, penelitian ini melakukan analisis sentimen terhadap 5.000 ulasan pengguna Mobile JKN yang diambil dari Google Play Store. Proses analisis meliputi cleaning, tokenizing, stopword removal, stemming, labeling, serta pembobotan kata dengan TF-IDF. Klasifikasi dilakukan menggunakan algoritma Random Forest Classifier. Hasil pengujian menunjukkan bahwa dari 976 data uji, 883 diprediksi benar dan 93 salah, dengan akurasi 90,47%, presisi 90,15%, recall 90,47%, dan F1-score 89,90%. Temuan ini membuktikan bahwa Random Forest efektif dalam mengidentifikasi sentimen positif maupun negatif, serta dapat menjadi dasar evaluasi pengembangan Mobile JKN di masa depan.

Kata kunci: analisis sentimen, BPJS Kesehatan, Mobile JKN, TF-IDF, Random Forest Classifier.

Abstract

Health is a fundamental need for all people. In an effort to ensure equitable access to health insurance services in Indonesia, the government has assigned BPJS Kesehatan as the national health insurance provider. As part of its digital innovation, BPJS Kesehatan introduced the Mobile JKN application to make it easier for the public to access healthcare services online. However, the implementation of this application still raises various complaints, such as login difficulties, update disruptions, and limited features, as reflected in predominantly negative user reviews. To objectively assess public perception, this study conducted sentiment analysis on 5,000 user reviews of Mobile JKN obtained from the Google Play Store. The analysis process included cleaning, tokenizing, stopword removal, stemming, labeling, and word weighting using TF-IDF. Classification was performed using the Random Forest Classifier algorithm. The results showed that out of 976 test data, 883 were correctly predicted and 93 were misclassified, with an accuracy of 90.47%, precision of 90.15%, recall of 90.47%, and an F1-score of 89.90%. These findings demonstrate that Random Forest is effective in identifying both positive and negative sentiments and can serve as a basis for the future development of the Mobile JKN application.

Keywords: sentiment analysis, BPJS Kesehatan, Mobile JKN, TF-IDF, Random Forest Classifier

1. PENDAHULUAN

Kesehatan merupakan hal yang sangat penting dalam kehidupan manusia, sehat juga merupakan keadaan dari kondisi fisik yang baik, mental yang baik, dan kesejahteraan sosial, tidak hanya merupakan ketiadaan dari penyakit atau kelemahan [1]. Pemerintah mempunyai tanggung jawab untuk menjamin setiap warga negara memperoleh pelayanan kesehatan yang berkualitas sesuai dengan kebutuhan. Kualitas kesehatan pada masyarakat di Indonesia masih termasuk pada kategori rendah. Hal tersebut dapat terjadi karena disebabkan oleh berbagai macam faktor. Faktor-faktor tersebut diantaranya berupa

tidak terjangkau pelayanannya kesehatan karena mahal biaya, penyebaran tenaga kesehatan yang tidak merata, masih rendahnya kemampuan akses dari masyarakat terhadap pelayanan kesehatan, pelayanan tidak yang memadai, serta masih rendahnya kesadaran dari masyarakat dengan perilaku hidup sehat [2].

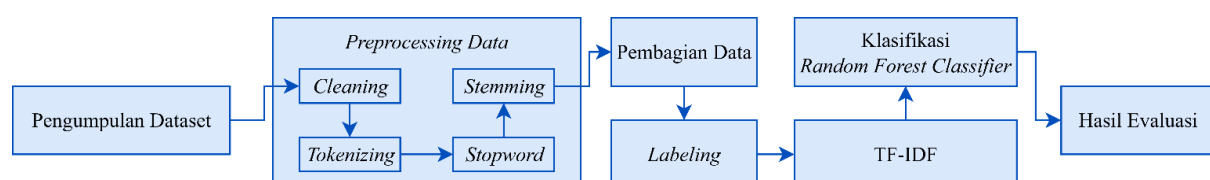
BPJS Kesehatan memberikan sebuah inovasi terbaru berupa layanan yang berbasis teknologi informasi yaitu *Mobile JKN*. *Mobile JKN* merupakan aplikasi dari BPJS Kesehatan yang memfasilitasi akses layanan kesehatan, khususnya bagi pasien rumah sakit, melalui fitur pendaftaran *online*, pengecekan ketersediaan kamar, dan lainnya [3]. Aplikasi *Mobile JKN* merupakan aplikasi yang bisa diakses melalui *smartphone*. Aplikasi ini menjadi terobosan baru. Namun dalam perjalanannya, aplikasi *Mobile JKN* masih belum sepenuhnya mendapatkan persepsi publik yang positif dan masih memiliki banyak kekurangan [4]. Pengguna saat ini menghadapi masalah dengan *Mobile JKN* seperti: aplikasi tidak dapat digunakan karena sering meminta untuk diupdate ke versi terbaru yang menyebabkan pengguna tidak dapat login [5].

Ulasan pengguna sangat efektif digunakan untuk menemukan informasi dari perspektif pengguna sehingga dapat dijadikan acuan terhadap hal-hal yang harus diperbaiki atau belum diketahui oleh pengembang aplikasi. Karena banyaknya jumlah ulasan yang diberikan pengguna, salah satu metode yang dapat digunakan adalah analisis sentimen [4]. Analisis sentimen yang dikombinasikan dengan metode komputasi, dapat dimanfaatkan untuk mengolah opini masyarakat guna mengetahui kecenderungan opini tersebut, apakah bersifat positif atau negatif. Pendekatan dua kelas dapat memudahkan proses pelabelan data, meningkatkan efisiensi pelatihan model, dan mengoptimalkan akurasi karena model tidak perlu menangani ambiguitas sentimen netral. Hasil dari analisis sentimen dapat dijadikan sebagai bahan pertimbangan untuk mengetahui respon masyarakat terhadap aplikasi *Mobile JKN* [6].

Terdapat banyak sekali metode komputasi yang digunakan dalam analisis sentiment ini seperti; *Naïve Bayes Classifier* (NBC), *K-Nearest Neighbors* (KNN), *Convolutional Neural Network* (CNN), *Decision Tree Classifier* (DTC), *Support Vector Machine* (SVM), dan *Random Forest Classifier* (RFC). Dari ke-6 metode tersebut, salah satu metode komputasi yang dipilih dalam penelitian ini adalah *Random Forest Classifier*. Penelitian sebelumnya yang menggunakan *Random Forest Classifier* berjudul “Analisis Sentimen Ulasan Aplikasi PeduliLingdungi dengan Metode *Random Forest*”. Menggunakan data kedalaman *tree* 65 dan jumlah *tree* 400 mendapatkan pembagian data rasio data latih 90% dan data uji 10%. Dengan nilai performa yang didapatkan adalah cukup dengan nilai *precision* 71%, *recall* 71%, *f1-score* 71% dan *accuracy* 72% [6].

Penelitian ini bertujuan untuk menganalisis sentimen pengguna terhadap aplikasi *Mobile JKN* berdasarkan 5.000 ulasan di *Google Play Store*. Hasil penelitian diharapkan dapat memberikan gambaran yang lebih objektif mengenai persepsi masyarakat serta menjadi dasar evaluasi dan pengembangan aplikasi *Mobile JKN* di masa mendatang.

2. METODE PENELITIAN



Gambar 1. Metodologi Penelitian

2.1. Pengumpulan Dataset

Pengumpulan data dalam penelitian ini menggunakan teknik *Crawling* yang dilakukan pada *Google Colaboratory*. Data yang diambil merupakan data ulasan aplikasi *Mobile JKN* dari *Google Play Store*.

Sebanyak 5.000 ulasan berbahasa Indonesia berhasil dikumpulkan sebagai *dataset* penelitian. Data yang sudah diambil kemudian dilakukan *preparation* yaitu pengambilan kolom yang akan digunakan dalam penelitian ini yakni kolom *content*. Setelah itu baru dilakukan tahap *preprocessing data*. Data akan diolah menggunakan bahasa pemrograman *Python* pada *Google Colaboratory*.

2.2. Preprocessing Data

Preprocessing data merupakan suatu proses membersihkan data yang dilakukan agar data menjadi berkualitas dan dapat memenuhi persyaratan untuk dieksekusi pada sebuah algoritma. [7]. Tahapan *preprocessing data* pada penelitian ini mencakup beberapa langkah yaitu *cleaning*, *tokenizing*, *stopword*, *stemming*, *labeling*, dan TF-IDF. Langkah-langkah ini memastikan data yang lebih bersih dan terstruktur, sehingga meningkatkan efisiensi dan akurasi dalam proses *machine learning*. [8]

2.2.1. Cleaning

Cleaning merupakan sebuah proses untuk menghapus tanda baca atau karakter yang tidak diperlukan dari suatu dokumen. Karakter yang terdapat pada kata dan tidak diperlukan, seperti tanda baca, emotikon, dan angka, akan dihapus. Karakter selain huruf akan dihapus, dan hanya mengambil huruf "a" sampai dengan "z" saja [9]. Dalam tahapan *cleaning* ini, terdapat juga *case folding*, *case folding* adalah proses mengubah huruf kapital menjadi huruf kecil untuk menghindari masalah *case sensitive* dalam proses klasifikasi [10].

2.2.2. Tokenizing

Tokenizing merupakan pemilahan kalimat menjadi perkata. Suatu proses pemotongan seluruh karakter dan sebuah set dokumen yang diberikan menjadi satu potongan kata. Potongan tersebut dikenal dengan istilah *token* [11].

2.2.3. Stopword

Stopword adalah proses penghapusan kata-kata yang tidak memiliki makna atau pengaruh signifikan dalam kalimat [10]. Contoh kata yang tidak memiliki arti penting atau tidak mempengaruhi sentimen, yaitu "dan," "yang," "bagaimana," dan lain-lain [12].

2.2.4. Stemming

Stemming adalah suatu proses pengembalian suatu kata berimbuhan ke dalam bentuk dasarnya. Proses ini akan menghilangkan awalan, akhiran, sisipan, dan *confixes* (kombinasi dari awalan dan akhiran) [11]. Contoh proses *stemming* yaitu kata "membaca" menjadi "baca," "meneruskan" menjadi "terus," dan lain sebagainya [13].

2.2.5. Labeling

Labeling adalah proses memberikan label pada setiap kalimat. Pelabelan ini dilakukan secara otomatis dengan *Python TextBlob*. *Python TextBlob* merupakan *library Python* yang digunakan untuk memproses data tekstual. *TextBlob* menyediakan API yang dapat digunakan untuk pemrosesan bahasa alami (NLP) seperti ekstraksi frase kata benda, analisis sentimen, klasifikasi, penerjemahan, dll. Hasil objek *TextBlob* digunakan untuk memproses pembelajaran bahasa alami dan *library TextBlob* hanya dapat mengenali Bahasa Inggris [14].

2.2.6. TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) adalah tahapan untuk memberikan nilai pada setiap kata yang ada, bertujuan untuk menemukan frekuensi kemunculan term dan dikalikan dengan inverse dari frekuensi dokumen [7]. Menurut [13], Pembobotan TF-IDF adalah metode untuk memberikan bobot pada hubungan antara kata dan dokumen. Metode ini mengombinasikan dua konsep penghitungan bobot: frekuensi kemunculan kata dalam sebuah dokumen tertentu dan frekuensi kebalikan dari dokumen yang mengandung kata tersebut. Rumus TF-IDF adalah sebagai berikut:

$$TF(d, t) = \frac{f(d, t)}{n(d)} \quad (1)$$

$$IDF(t) = \log\left(\frac{N}{df(t)}\right) \quad (2)$$

$$TF - IDF(d, t) = TF(d, t) \times IDF(t) \quad (3)$$

Keterangan:

- d : Dokumen
- t : Term (kata) yang sedang dievaluasi
- $f(d, t)$: Frekuensi kemunculan term t dalam dokumen d
- $n(d)$: Jumlah total term dalam dokumen d
- N : Jumlah total dokumen dalam koleksi
- $df(t)$: Jumlah dokumen di koleksi yang mengandung term t

2.3. Random Forest Classifier

Random Forest Classifier adalah evolusi dari algoritma *decision tree*, dimana setiap pohon keputusan dilatih menggunakan data individu, dan setiap atribut dipilih secara acak pada setiap pembentukan pohon [15]. Menurut [10], *Random Forest* adalah algoritma yang digunakan untuk mengklasifikasikan data dalam volume besar, melibatkan penggabungan beberapa pohon keputusan dan melatihnya menggunakan sampel data yang ada. Masing-masing pohon keputusan memiliki node-node yang digunakan untuk membagi keputusan. *Random Forest* melakukan prediksi dalam analisis sentimen dengan menggabungkan hasil dari setiap pohon keputusan. Menurut [6], Algoritma *Decision tree* yang menjadi dasar dari *Random Forest*, dimulai dengan menghitung nilai *entropy* yang menjadi acuan dalam menentukan tingkatan normatif pada node atribut. Nilai *information gain* digunakan sebagai penentu tingkat informatif suatu node atribut.

Rumus menghitung nilai *Entropy* adalah sebagai berikut:

$$Entropy(Y) = - \sum p(c|Y) \log_2 p(c|Y) \quad (4)$$

Keterangan:

- Y : Himpunan Kasus
- $p(c|Y)$: Proporsi nilai c terhadap kelas Y

Rumus *information gain* adalah sebagai berikut:

$$information\ gain(Y, a) = Entropy(Y) - \sum_{v \in Values(a)} \frac{|Y_v|}{|Y|} Entropy(Y_v) \quad (5)$$

Keterangan:

- $Values(a)$: Nilai yang mungkin dalam himpunan kasus a
- Y_v : Subkelas dari kelas Y dengan kelas v yang berhubungan dengan kelas a .
- Y_a : Semua nilai yang sesuai dengan a

2.4. Heatmap

Heatmap adalah teknik visualisasi data yang menggunakan warna untuk merepresentasikan besarnya nilai suatu data, di mana warna yang berbeda menunjukkan nilai yang berbeda pula. Warna yang lebih gelap menunjukkan intensitas data yang lebih tinggi [16].

2.5. Confusion Matrix

Confusion Matrix adalah tabel yang menyatakan klasifikasi jumlah data uji yang benar dan jumlah data uji yang salah [17]. Menurut [10], *Confusion Matrix* adalah suatu alat evaluasi dalam bentuk matriks yang berguna untuk mengevaluasi kinerja model klasifikasi. *Confusion Matrix* memiliki empat kejadian

yang meliputi; TP (*True Positive*): data yang berhasil terklasifikasi sebagai positif dengan benar; TN (*True Negative*): data yang berhasil terklasifikasi sebagai negatif dengan benar; FP (*False Positive*): data yang keliru terklasifikasi sebagai positif; FN (*False Negative*): data yang keliru terklasifikasi sebagai negatif. Dengan membandingkan nilai kejadian yang terdapat dalam *Confusion Matrix*, maka dapat dihitung berbagai metrik evaluasi seperti akurasi, presisi, *recall*, dan *f1-score*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Penelitian

3.1.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini diperoleh melalui proses *crawling* yang dilakukan secara otomatis menggunakan library *google_play_scraper* pada tools *Google Colaboratory*. Proses ini mengambil data ulasan aplikasi *Mobile JKN* yang tersedia di platform *Google Play Store*, dengan parameter pencarian menggunakan bahasa Indonesia (*lang='id'*), wilayah Indonesia (*country='id'*), serta pengurutan berdasarkan ulasan terbaru. Data yang berhasil dikumpulkan terdiri dari *username* pengguna, komentar ulasan, dan tanggal pengiriman ulasan, dengan jumlah ulasan yang diambil sebanyak 5.000 entri.

3.1.2. Preprocessing Data

Pada tahap *pre-processing* terdapat beberapa tahap yang telah dilakukan sebelumnya agar data bersih dan siap digunakan kedalam tahap selanjutnya. Adapun tahapan- tahapan tersebut adalah sebagai berikut.

1. *Cleaning*

Pada tahap ini, elemen-elemen yang dianggap tidak relevan seperti angka, simbol, tanda baca, dan emotikon dihapus menggunakan perintah *re.sub()* untuk mempertahankan hanya karakter huruf. Selain itu, seluruh teks juga dikonversi ke huruf kecil (*lowercase*) dan spasi yang tidak diperlukan dihilangkan. Proses pembersihan ini bertujuan untuk memperoleh data teks yang lebih bersih, seragam, dan siap digunakan dalam tahapan *preprocessing* dan analisis selanjutnya.

Tabel 1. Hasil *Cleaning*

Komentar	<i>Cleaning</i>
Login sulit, verif sulit, daftar antrian sulit, No rujukan dipencet faskes lanjut udah daftar pun ga keluar no rujuknya, APLIKASI GA JELAS!!, KERJA WOY TIM IT! percuma aplikasi kesehatan guna memudahkan malah MEMPERSULIT! 🤔	login sulit verif sulit daftar antrian sulit no rujukan dipencet faskes lanjut udah daftar pun ga keluar no rujuknya aplikasi ga jelas kerja woy tim it percuma aplikasi kesehatan guna memudahkan malah mempersulit

2. *Tokenizing*

Pada tahap ini, setiap kalimat dipisahkan menjadi satuan kata atau *token* menggunakan fungsi *word_tokenize* dari library *Natural Language Toolkit* (NLTK). Fungsi ini secara otomatis memecah teks menjadi daftar kata berdasarkan aturan linguistik dasar, sehingga teks yang semula dalam bentuk kalimat panjang menjadi lebih terstruktur dalam bentuk unit kata.

Tabel 2. Hasil *Tokenizing*

<i>Cleaning</i>	<i>Tokenizing</i>
login sulit verif sulit daftar antrian sulit no rujukan dipencet faskes lanjut udah daftar pun ga keluar no rujuknya aplikasi ga jelas kerja woy tim it percuma aplikasi kesehatan guna memudahkan malah mempersulit	['login', 'sulit', 'verif', 'sulit', 'daftar', 'antrian', 'sulit', 'no', 'rujukan', 'dipencet', 'faskes', 'lanjut', 'udah', 'daftar', 'pun', 'ga', 'keluar', 'no', 'rujuknya', 'aplikasi', 'ga', 'jelas', 'kerja', 'woy', 'tim', 'it', 'percuma', 'aplikasi', 'kesehatan', 'guna', 'memudahkan', 'malah', 'mempersulit']

3. *Stopword*

Pada tahap ini digunakan *library* NLTK dengan daftar kata dari modul *stopword* bahasa Indonesia, yang berfungsi menghapus kata-kata umum yang dianggap tidak memiliki kontribusi signifikan terhadap makna analisis, seperti “lanjut”, “pun”, dan “keluar”. Untuk menjaga akurasi konteks sentimen, daftar *stopword* disesuaikan dengan mengecualikan beberapa kata penting seperti “tidak”, “masalah”, dan “gagal” agar tidak ikut terhapus. Penghapusan *stopword* bertujuan mengurangi noise dalam data teks, sehingga hasil analisis menjadi lebih fokus, akurat, dan bermakna.

Tabel 3. Hasil *Stopword*

<i>Tokenizing</i>	<i>Stopword</i>
['login', 'sulit', 'verif', 'sulit', 'daftar', 'antrian', 'sulit', 'no', 'rujukan', 'dipencet', 'faskes', 'lanjut', 'udah', 'daftar', 'pun', 'ga', 'keluar', 'no', 'rujuknya', 'aplikasi', 'ga', 'jelas', 'kerja', 'woy', 'tim', 'it', 'percuma', 'aplikasi', 'kesehatan', 'guna', 'memudahkan', 'malah', 'mempersulit']	['login', 'sulit', 'verif', 'sulit', 'daftar', 'antrian', 'sulit', 'no', 'rujukan', 'dipencet', 'faskes', 'udah', 'daftar', 'ga', 'no', 'rujuknya', 'aplikasi', 'ga', 'kerja', 'woy', 'tim', 'it', 'aplikasi', 'kesehatan', 'memudahkan', 'mempersulit']

4. *Stemming*

Dalam tahap ini digunakan *library* Sastrawi melalui metode *StemmerFactory*, yang secara otomatis menghapus imbuhan awalan, sisipan, maupun akhiran dari kata-kata dalam bahasa Indonesia. Contohnya, kata “antrian” diubah menjadi “antri”, “rujukan” menjadi “rujuk”, “memudahkan” menjadi “mudah”, dan kata dengan dua imbuhan seperti “mempersulit” menjadi “sulit”. Proses ini bertujuan menyederhanakan bentuk kata agar analisis dapat dilakukan secara lebih konsisten dan akurat. Data yang telah melalui tahap *stemming* ini kemudian siap digunakan dalam proses analisis selanjutnya.

Tabel 4. Hasil *Stemming*

<i>Stopword</i>	<i>Stemming</i>
['login', 'sulit', 'verif', 'sulit', 'daftar', 'antrian', 'sulit', 'no', 'rujukan', 'dipencet', 'faskes', 'udah', 'daftar', 'ga', 'no', 'rujuknya', 'aplikasi', 'ga', 'kerja', 'woy', 'tim', 'it', 'aplikasi', 'kesehatan', 'memudahkan', 'mempersulit']	['login', 'sulit', 'verif', 'sulit', 'daftar', 'antri', 'sulit', 'no', 'rujuk', 'pencet', 'faskes', 'udah', 'daftar', 'ga', 'no', 'rujuk', 'aplikasi', 'ga', 'kerja', 'woy', 'tim', 'it', 'aplikasi', 'sehat', 'mudah', 'sulit']

5. *Join*

Proses ini dilakukan menggunakan fungsi *join()* yang menggabungkan setiap elemen dalam daftar kata hasil *stemming* menjadi kalimat utuh. Dengan mengembalikan struktur teks ke format semula, data menjadi lebih siap digunakan dalam tahap *labeling* sentimen. Tahap ini penting untuk memastikan analisis sentimen dapat dilakukan secara lebih akurat pada proses berikutnya

Tabel 5. Hasil *Join*

<i>Stemming</i>	<i>Join</i>
['login', 'sulit', 'verif', 'sulit', 'daftar', 'antri', 'sulit', 'no', 'rujuk', 'pencet', 'faskes', 'udah', 'daftar', 'ga', 'no', 'rujuk', 'aplikasi', 'ga', 'kerja', 'woy', 'tim', 'it', 'aplikasi', 'sehat', 'mudah', 'sulit']	login sulit verif sulit daftar antri sulit no rujuk pencet faskes udah daftar ga no rujuk aplikasi ga kerja woy tim it aplikasi sehat mudah sulit

6. *Labeling*

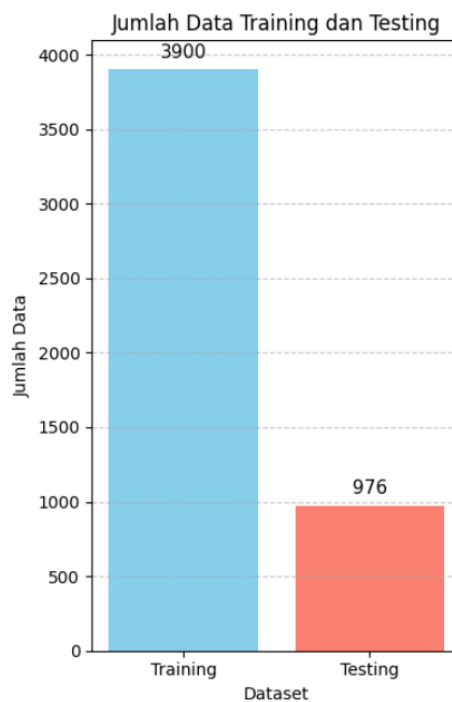
Pada tahap ini, setiap kalimat diterjemahkan terlebih dahulu menggunakan *GoogleTranslator*, kemudian dianalisis oleh *TextBlob* untuk menghitung nilai *polarity* dalam rentang -1 hingga +1. Berdasarkan nilai tersebut, komentar dikategorikan ke dalam dua label sentimen, yaitu positif jika nilai ≥ 0 dan negatif jika nilai < 0 . Hasil dari proses *labeling* ini digunakan sebagai dasar dalam analisis lanjutan, seperti visualisasi sentimen atau pelatihan model klasifikasi.

Tabel 6. Hasil *Labeling*

<i>Join</i>	<i>Labeling</i>
login sulit verif sulit daftar antri sulit no rujuk pencet faskes udah daftar ga no rujuk aplikasi ga kerja woy tim it aplikasi sehat mudah sulit	negatif

3.1.3. Pembagian Data

Setelah data dibersihkan pada tahap *preprocessing data*, kemudian data yang dapat digunakan adalah sebanyak 4876 data. Pada tahap ini dilakukan pembagian data menggunakan fungsi *train_test_split* dari *library scikit-learn*, dengan proporsi 80% data untuk pelatihan (*training*) berjumlah 3900 data dan 20% untuk pengujian (*testing*) berjumlah 976 data, serta pengacakan data dikontrol dengan parameter *random_state=42* untuk memastikan reproduisibilitas. Hasil dari tahap ini ditampilkan dalam bentuk diagram menggunakan *library matplotlib.pyplot* untuk melihat perbandingan total data training dan data testing.



Gambar 2. Hasil Pembagian Data

3.1.4. TF-IDF

Proses ini dilakukan dengan memanfaatkan fungsi *TfidfVectorizer()* dari *library scikit-learn* untuk mengubah setiap kata pada kolom *teks* menjadi representasi numerik berbasis bobot kata. Selanjutnya, label sentimen pada kolom *label* dikonversi ke dalam bentuk numerik menggunakan *LabelEncoder()* untuk mempersiapkan data dalam format yang dapat diproses oleh algoritma pembelajaran mesin. Hasil dari proses ini adalah data input (X) berupa matriks nilai TF-IDF dan data target (y) berupa label numerik, yang keduanya siap digunakan dalam tahap pelatihan model klasifikasi sentimen.

Tabel 7. Hasil TF-IDF

Teks	TF-IDF
sulit	0.577350
aplikasi	0.288675
rujuk	0.288675
daftar	0.288675
no	0.288675
ga	0.288675
antri	0.144338
kerja	0.144338
it	0.144338
faskes	0.144338
mudah	0.144338
login	0.144338
pencet	0.144338
sehat	0.144338
tim	0.144338
udah	0.144338
verif	0.144338
woy	0.144338

3.1.5. Klasifikasi *Random Forest Classifier*

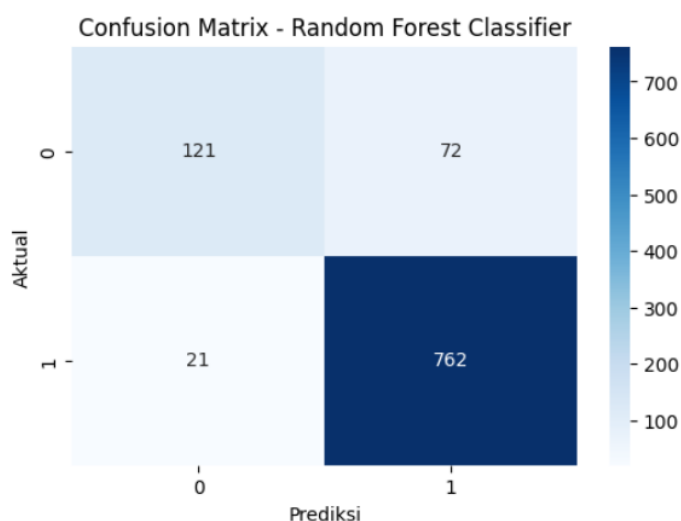
Pada tahap ini, model dibangun dengan menggunakan fungsi *Random Forest Classifier* dari library *scikit-learn*, dengan jumlah *estimators* sebanyak 100 pohon keputusan (*decision trees*) serta parameter *class_weight='balanced'* untuk mengatasi ketidak seimbangan kelas. Model dilatih pada data *training* yang telah melalui ekstraksi fitur, lalu digunakan untuk memprediksi data *testing*. Hasil prediksi dibandingkan dengan label aktual untuk menilai performa model. Data klasifikasi disajikan dalam bentuk *DataFrame* berisi kolom *Teks*, *Aktual*, *Prediksi*, dan Hasil dengan Hasil menunjukkan kecocokan antara prediksi dan nilai sebenarnya.

	Teks	Aktual	Prediksi	Hasil
1643	error desak aplikasi tidak bisa daftar antre o...	Positif	Positif	True
4047	proses mudah	Positif	Positif	True
2931	daftar wajah sulit susah deteksi wajah	Negatif	Negatif	True
4728	tolong instansi korupsi manusia	Positif	Positif	True
3845	tinggal daerah susah nakes antri tidak bagi gi...	Negatif	Negatif	True
3981	bantu	Positif	Positif	True
519	mudah	Positif	Positif	True
2752	klo bpjs sllu sulit	Negatif	Negatif	True
3287	tolong verikasi muka betul	Positif	Positif	True
2408	susah login nik password salah	Negatif	Negatif	True

Gambar 3. Hasil Klasifikasi *Random Forest Classifier*

3.1.6. Hasil Evaluasi

Evaluasi dilakukan dengan menghitung metrik performa menggunakan *confusion matrix*. *Confusion matrix* divisualisasikan menggunakan *heatmap* dari library *Seaborn*, yang membandingkan prediksi benar dan salah berdasarkan kelas aktual dan prediksi. Selain itu, hasil evaluasi disajikan dalam bentuk *DataFrame*, yang mencakup jumlah data uji, jumlah prediksi benar dan salah, serta nilai *accuracy*, *precision*, *recall*, dan *f1-score* dalam satuan persentase. Evaluasi ini memberikan gambaran menyeluruh terhadap kinerja model dalam mengklasifikasikan sentimen secara tepat dan seimbang.



Gambar 4. Confusion Matrix – Random Forest Classifier

Tabel 8. Hasil Evaluasi

Data Uji	Prediksi Benar	Prediksi Salah	Accuracy	Precision	Recall	F1 Score
976	883	93	90.47%	90.15%	90.47%	89.90%

Berdasarkan tabel 8 diatas ini, dapat dilihat bahwa pada metode *Random Forest Classifier* diperoleh total data uji sebanyak 976 data. Dari jumlah tersebut, sebanyak 883 data berhasil diprediksi dengan benar, sementara 93 data mengalami kesalahan prediksi. Model menghasilkan nilai akurasi sebesar 90,47%, yang menunjukkan bahwa sebagian besar prediksi telah sesuai dengan label aktual. Selain itu, nilai presisi sebesar 90,15%, *recall* sebesar 90,47%, dan *f1-score* sebesar 89,90% menunjukkan bahwa model memiliki performa yang baik dan cukup seimbang dalam mengenali masing-masing kelas sentimen.

3.2 Pembahasan

Penelitian ini membuktikan bahwa algoritma *Random Forest Classifier* mampu memberikan hasil yang cukup baik dengan *accuracy* mencapai 90,47%, *precision* 90,15%, *recall* 90,47%, dan *f1-score* 89,90% dalam mengklasifikasikan ulasan pengguna *Mobile JKN*. Jika dibandingkan dengan penelitian-penelitian sebelumnya yang menggunakan metode seperti *Naive Bayes* atau *K-Nearest Neighbor*, tingkat akurasi yang diperoleh dalam penelitian ini lebih tinggi, karena sebagian besar penelitian terdahulu masih berada di bawah angka 85%. Hal ini menunjukkan bahwa *Random Forest* lebih unggul dalam menangani data ulasan yang jumlahnya besar dan beragam, serta lebih stabil ketika menghadapi perbedaan antar kelas data. Meski begitu, penelitian ini tetap memiliki keterbatasan, misalnya hanya menggunakan data dari *Google Play Store* dan belum membandingkan hasilnya langsung dengan algoritma lain pada dataset yang sama. Walaupun ada keterbatasan tersebut, penelitian ini tetap memberikan manfaat nyata. Dari sisi akademis, penelitian ini menambah bukti bahwa *Random Forest* efektif untuk analisis sentimen di aplikasi layanan publik, sedangkan dari sisi praktis, hasil penelitian dapat menjadi masukan berharga bagi BPJS Kesehatan untuk mengevaluasi dan meningkatkan kualitas aplikasi *Mobile JKN*.

4. KESIMPULAN

Dari hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma *Random Forest Classifier* mampu digunakan secara efektif dalam melakukan analisis sentimen terhadap 5.000 ulasan pengguna aplikasi *Mobile JKN* yang diperoleh dari *Google Play Store*. Proses analisis diawali dengan pengambilan data melalui metode *crawling*, kemudian dilakukan tahapan *preprocessing* meliputi

cleaning, *tokenizing*, *stopword removal*, *stemming*, dan *join*. Lalu dilakukan pelabelan data sentimen menggunakan *textblob*, tahapan pembagian data, serta pembobotan kata menggunakan TF-IDF. Kemudian data yang telah bersih diuji menggunakan algoritma *Random Forest Classifier*. Hasil pengujian menunjukkan bahwa model *Random Forest* berhasil mengidentifikasi 883 data dengan benar dari total 976 data uji, menghasilkan *accuracy* 90,47%, *precision* 90,15%, *recall* 90,47%, dan *f1-score* 89,90%. Secara lebih rinci, model mampu mengenali sentimen positif sebanyak 762 data dan sentimen negatif sebanyak 121 data sesuai dengan label aktual. Temuan ini membuktikan bahwa *Random Forest* cukup andal dalam membedakan sentimen positif maupun negatif, serta dapat menjadi acuan untuk evaluasi dan pengembangan lebih lanjut terhadap aplikasi *Mobile JKN*.

Daftar Pustaka

- [1] C. K. R. Lasso, "FAKTOR HAMBATAN DALAM AKSES PELAYANAN KESEHATAN PADA PUSKESMAS DI INDONESIA: SCOPING REVIEW," *J. Ilm. Permas J. Ilm. STIKES Kendal*, vol. 14, no. 4, pp. 1337–1344, 2023, doi: 10.32767/jusikom.v7i1.1538.
- [2] R. Syaputri and B. Hartono, "Implementasi Mutu Pelayanan Jaminan Kesehatan Nasional (JKN) Di Dinas Kesehatan Kabupaten Rokan Hilir (Systematic Literature Riview)," *J. Ilm. Kesehat. Indones.*, vol. 1, no. 1, pp. 108–114, 2023, [Online]. Available: <https://publikasi.abidan.org/index.php/jiki/article/view/159>
- [3] I. E. Putra, Asriyanik, and F. F. Azzahra, "PEMANFAATAN MULTINOMIAL NAIVE BAYES UNTUK ANALISIS," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 2, pp. 3283–3291, 2025.
- [4] S. Roiqoh, B. Zaman, and Kartono, "Analisis Sentimen Berbasis Aspek Ulasan Aplikasi Mobile JKN dengan Lexicon Based dan Naïve Bayes," *J. Media Inform. Budidarma*, vol. 7, no. 3, pp. 1582–1592, 2023, doi: 10.30865/mib.v7i3.6194.
- [5] N. Maulida, N. Suarna, and W. Prihartono, "Analisis Ulasan Sentimen Aplikasi Mobile Jkn Dengan Algoritma Support Vector Machine Berbasis Particle Swarm Optimization," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 2, pp. 1651–1658, 2024, doi: 10.36040/jati.v8i2.9105.
- [6] M. R. U. Pulungan, D. E. Ratnawati, and B. Rahayudi, "Analisis Sentimen Ulasan Aplikasi PeduliLindungi dengan Metode Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komun.*, vol. 6, no. 9, pp. 4378–4385, 2022, [Online]. Available: <http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/download/11582/5142>
- [7] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 9, pp. 4305–4313, 2022.
- [8] A. Priofani, M. Devega, Yuhelmi, and Walhidayat, "ANALISIS SENTIMEN TERHADAP ULASAN PADA WHIZ PRIME HOTEL SUDIRMAN PEKANBARU MENGGUNAKAN SUPPORT VECTOR MACHINE DAN NAIVE BAYES," *Zo. J. Sist. Inf.*, vol. 7, no. 2, pp. 490–501, 2025.
- [9] M. M. Rohman, Indriati, and S. Adinugroho, "Analisis Sentimen pada Ulasan Aplikasi Mobile JKN Menggunakan Metode Maximum Entropy dan Seleksi Fitur Gini Index Text," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, pp. 2646–2654, 2021, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [10] A. Sagita, A. Faqih, G. Dwilestari, B. Siswoyo, and D. Pratama, "Penerapan Metode Random Forest Dalam Menganalisis Sentimen Pengguna Aplikasi Capcut Di Google Play Store," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 6, pp. 3307–3313, 2023, doi: 10.36040/jati.v7i6.8205.
- [11] S. Nanda, D. Mualfah, and D. A. Fitri, "Analisis Sentimen Kepuasan Pengguna Terhadap Layanan Streaming Mola Menggunakan Algoritma Random Forest," *J. Apl. Teknol. Inf. dan Manaj.*, vol. 3, no. 2, pp. 210–219, 2022, doi: 10.31102/jatim.v3i2.1592.
- [12] N. A. Hapsari and A. D. Indriyanti, "Analisis Sentimen pada Aplikasi Dompot Digital Menggunakan Algoritma Random Forest," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 04, no. 03, pp. 186–192, 2023.
- [13] A. Fahrezi and A. Solichin, "ANALISIS SENTIMEN ULASAN APLIKASI MOBILE JKN PADA PLAY STORE MENGGUNAKAN METODE MULTINOMIAL NAÏVE BAYES," *Semin. Nas. Mhs. Fak. Teknol. Inf.*, vol. 3, no. 2, pp. 1–10, 2024.
- [14] R. Azhar, A. Surahman, and C. Juliane, "Analisis Sentimen Terhadap Cryptocurrency Berbasis

- Python TextBlob Menggunakan Algoritma Naïve Bayes,” *J. Sains Komput. Inform.*, vol. 6, no. 1, pp. 267–281, 2022.
- [15] M. F. Y. Herjanto and Carudin, “ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI SIREKAP PADA PLAY STORE MENGGUNAKAN ALGORITMA RANDOM FOREST CLASSIFER,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, pp. 1204–1210, 2024, doi: 10.23960/jitet.v12i2.4192.
- [16] Y. Darmayunata, M. Devega, and Y. Yuhelmi, “Application of SMART Method and Dashboard Visualization for Student Code of Conduct Violations,” *Sistemasi*, vol. 13, no. 5, pp. 2237–2253, 2024, doi: 10.32520/stmsi.v13i5.4593.
- [17] Z. Yunizar, Rusnani, Z. Ardian, H. A. Aidilof, and O. K. M. M. Maulana, “ANALISIS SENTIMEN PADA TWITTER TERHADAP APLIKASI MOBILE JKN MENGGUNAKAN METODE NAÏVE BAYES CLASSIFIER,” *J. Informatics Comput. Sci.*, vol. 9, no. 2, pp. 103–111, 2023, doi: 10.36596/jitu.v6i1.781.



ZONAsi: Jurnal Sistem Informasi

Is licensed under a [Creative Commons Attribution International \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)