

Perbandingan Tingkat Akurasi Algoritma *Naïve Bayes*, *Decision Tree* dan *Random Forest* dalam Analisis Sentimen Pengguna Aplikasi Samsat Digital Nasional

Hilmi Ammar¹, Ade Davy Wiranata^{1*}

¹Program Studi S1 Teknik Informatika, Universitas Muhammadiyah Prof.Dr.Hamka, Indonesia

*Email: adedavy@uhamka.ac.id

Info Artikel	Abstrak
<p>Kata Kunci : Analisis Sentimen, <i>Naïve Bayes</i>, <i>Decision Tree</i>, <i>Random Forest</i>, Samsat Digital Nasional</p> <p>Keywords : <i>Sentiment Analysis</i>, <i>Naïve Bayes</i>, <i>Decision Tree</i>, <i>Random Forest</i>, <i>National Digital Samsat</i></p> <p>Tanggal Artikel Dikirim : 13 November 2025 Direvisi : 25 Desember 2025 Diterima : 30 Desember 2025</p>	<p>Penelitian ini menganalisis sentimen pengguna terhadap aplikasi Signal – Samsat Digital Nasional. Dari total 2.000 ulasan yang terkumpul, setelah melalui tahapan <i>filter</i> data diperoleh 1.743 data yang digunakan untuk analisis. Data tersebut kemudian dipecah menjadi 1.394 data pelatihan (alokasi 80%) dan 349 data pengujian (alokasi 20%). Tiga model klasifikasi yang digunakan dalam penelitian ini meliputi <i>Naïve Bayes</i>, <i>Decision Tree</i>, dan <i>Random Forest</i>. Hasil penelitian menunjukkan bahwa <i>Random Forest</i> memiliki performa terbaik dengan akurasi 92,9% serta keseimbangan tinggi dalam mengenali sentimen positif (<i>f1-score</i> 95,9%) dan negatif (<i>f1-score</i> 73,1%). <i>Naïve Bayes</i> mencapai akurasi 89,4% namun kurang seimbang, sedangkan <i>Decision Tree</i> memperoleh akurasi 86,8% dengan hasil yang lebih stabil dibandingkan <i>Naïve Bayes</i>. Secara keseluruhan, visualisasi data berdasarkan analisis menggunakan kamus <i>lexicon</i> menunjukkan bahwa 82,9% ulasan bersentimen positif dan didominasi oleh rating bintang lima, yang mengindikasikan bahwa pengalaman pengguna terhadap aplikasi Signal – Samsat Digital Nasional tergolong sangat baik.</p> <p>Abstract</p> <p><i>This research analyzes user sentiment towards the Signal – National Digital Samsat application. From a total of 2,000 collected reviews, after going through the data filtering stage, 1,743 data were obtained for analysis. The data was then split into 1,394 training data (80% allocation) and 349 testing data (20% allocation). Three classification models used in this study were Naïve Bayes, Decision Tree, and Random Forest. The results showed that Random Forest had the best performance with 92.9% accuracy and high balance in recognizing positive (f1-score 95.9%) and negative (f1-score 73.1%) sentiment. Naïve Bayes achieved 89.4% accuracy but was less balanced, while Decision Tree achieved 86.8% accuracy with more stable results than Naïve Bayes. Overall, data visualization based on analysis using the lexicon dictionary shows that 82.9% of reviews are positive and dominated by five-star ratings, which indicates that the user experience of the Signal – Samsat Digital Nasional application is classified as very good.</i></p>

1. PENDAHULUAN

Era digitalisasi telah mengubah paradigma layanan publik di Indonesia, salah satunya adalah adopsi teknologi digital dalam berbagai sektor pemerintahan [1]. Pada sektor pemerintahan, transformasi digital berperan penting dalam menciptakan sistem pelayanan publik yang lebih efisien dan transparan serta membuka ruang partisipasi masyarakat dalam proses pengambilan keputusan. Sejalan dengan perkembangan teknologi yang semakin pesat, pemerintah terus berupaya meningkatkan kualitas serta efektivitas layanan publik melalui penerapan berbagai inovasi digital dalam tata kelola pemerintahan [2].

Merujuk pada Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, Sistem Administrasi Manunggal Satu Atap (SAMSAT) merupakan mekanisme administrasi terpadu yang diterapkan di Indonesia untuk mengelola administrasi perpajakan dan penerimaan negara terkait kendaraan bermotor. Tujuan dari penerapan SAMSAT adalah meningkatkan kepatuhan wajib pajak kendaraan, sekaligus menyediakan layanan administrasi yang efektif dan terintegrasi [3]. Melalui sistem ini, pemerintah dapat memantau penerimaan pajak kendaraan bermotor guna mendukung pembiayaan pembangunan infrastruktur dan penyediaan layanan publik lainnya [4].

Dalam upaya meningkatkan kualitas layanan administratif, SAMSAT mengembangkan inovasi melalui peluncuran aplikasi Samsat Digital Nasional (SIGNAL), yang dapat diunduh secara resmi melalui platform *Google Play Store*. Aplikasi ini dikembangkan melalui kerja sama antara Korlantas Polri, Kementerian Dalam Negeri dan Jasa Raharja dengan tujuan menyediakan layanan pembayaran Pajak Kendaraan Bermotor (PKB) serta pengesahan tahunan Surat Tanda Nomor Kendaraan (STNK) secara *online* [5]. Tujuan pengembangan aplikasi SIGNAL adalah untuk menyederhanakan proses birokrasi, mengurangi antrean di kantor SAMSAT, meningkatkan kepatuhan wajib pajak, serta meminimalisir potensi praktik pungutan liar melalui sistem yang lebih transparan [6].

Berdasarkan ulasan pengguna di *Google Play Store*, aplikasi Samsat Digital Nasional (SIGNAL) diapresiasi karena kemudahan pembayaran pajak kendaraan, namun masih dinilai kurang stabil. Beberapa pengguna melaporkan kendala seperti proses verifikasi yang sulit, kesalahan transaksi, dan waktu pemrosesan yang lambat. Hal ini menunjukkan bahwa keberhasilan aplikasi tidak hanya ditentukan oleh aspek teknis, tetapi juga oleh kemampuannya dalam memenuhi kebutuhan dan harapan pengguna. Oleh karena itu, evaluasi pengalaman pengguna melalui analisis sentimen ulasan diperlukan untuk mengidentifikasi faktor kepuasan dan ketidakpuasan sebagai dasar perbaikan dan pengembangan aplikasi di masa mendatang [7].

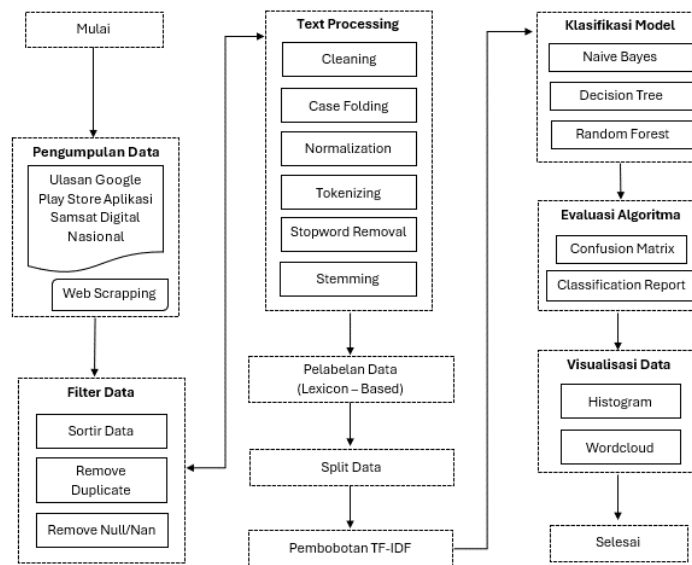
Analisis sentimen merupakan bagian dari *Natural Language Processing* (NLP) yang bertujuan untuk mengidentifikasi opini, emosi, dan tanggapan pengguna dari data tekstual [8]. Analisis ini memanfaatkan teknik pengolahan bahasa dan algoritma *machine learning* untuk mengklasifikasikan ulasan ke dalam tiga kategori, yaitu sentimen positif, negatif, dan netral [9]. Beberapa algoritma yang umum digunakan antara lain *Naïve Bayes*, *Decision Tree*, dan *Random Forest*, di mana masing-masing memiliki keunggulan tersendiri. Namun, berbagai penelitian menunjukkan bahwa tidak ada algoritma yang selalu unggul pada semua kondisi, karena kinerjanya sangat dipengaruhi oleh karakteristik dataset seperti jumlah data, fitur dan penggunaan bahasa [10].

Penelitian terkait analisis sentimen pada aplikasi Samsat Digital Nasional (SIGNAL) telah banyak dilakukan sebelumnya. Imam Fathur Rahman menggunakan algoritma *Naïve Bayes* dan memperoleh akurasi 89% dengan nilai *recall* 85% dan *f1-score* 87% dari 1.000 data ulasan [11]. Penelitian lain oleh Slamet Kacung menerapkan algoritma *Support Vector Machine* (SVM) dan menghasilkan akurasi 96,19% dengan rata-rata *K-Fold* sebesar 97,65% [12]. Selain itu, Ruby Chandra juga menggunakan algoritma *Naïve Bayes* dan mencapai akurasi 91,64% dengan *precision* sebesar 90,41% dan *recall* sebesar 74,973% [13]. Hasil-hasil tersebut menunjukkan bahwa berbagai algoritma mampu memberikan kinerja yang baik, meskipun tingkat akurasi yang diperoleh berbeda-beda.

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengisi *gap* dengan membandingkan kinerja tiga algoritma klasifikasi, yaitu *Naïve Bayes*, *Decision Tree*, dan *Random Forest* untuk menentukan metode mana yang memberikan akurasi performa terbaik dalam mengklasifikasikan sentimen pengguna aplikasi Samsat Digital Nasional. Dengan demikian, penelitian ini dapat menjadi dasar bagi pengembang dalam memperbaiki kualitas layanan dan menyempurnakan fitur aplikasi.

2. METODE PENELITIAN

Metode penelitian yang digunakan dengan memanfaatkan *Google Colab* dan pustaka pendukung yang tersedia di bahasa pemrograman *Python*. Proses analisis dilakukan dengan pendekatan algoritma *Naïve Bayes*, *Decision Tree*, dan *Random Forest* untuk mengevaluasi serta membandingkan tingkat akurasinya. Gambar 1 merupakan alur pelaksanaan penelitian ini secara menyeluruh.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Data

Pengumpulan data dilakukan dengan metode *web scraping* untuk memperoleh ulasan pengguna aplikasi Signal – Samsat Digital Nasional yang tersedia pada *Google Play Store*. Proses *web scraping* dijalankan dengan memanfaatkan layanan *Google Colab* sebagai media komputasi berbasis awan (*cloud-based*) yang mendukung bahasa pemrograman *Python* untuk mengakses dan mengekstrak data komentar pengguna aplikasi. Data yang telah terkumpul disimpan dalam format *CSV* dan kemudian diakses kembali melalui *Google Colab* untuk dilakukan analisis [14].

2.2 Filter Data

Filter data dilakukan dalam dua tahapan, yaitu pada data mentah sebelum dan sesudah proses *text processing*. Pada tahap pertama, kolom yang tidak relevan dihapus dan dipilih variabel utama penelitian, yaitu *username*, *rating*, dan *review*, serta dilakukan eliminasi data duplikat. Selanjutnya, pada tahap kedua data hasil *text processing* yang meliputi *cleaning*, *case folding*, *normalization*, *tokenizing*, *stopword removal*, dan *stemming* kembali difilter untuk memvalidasi dan memastikan tidak ada data tidak valid yang tersisa sebelum masuk ke tahap analisis utama.

2.3 Text Processing

Pra-pemrosesan teks merupakan langkah untuk mengoreksi data yang tidak konsisten atau tidak sesuai dalam dataset. Dalam proses ini, berbagai komponen yang tidak relevan akan dihapus, dan data akan diubah menjadi bentuk yang lebih terstruktur [15]. Proses ini mencakup beberapa tahapan sebagai berikut:

- Cleaning** (Pembersihan): Tahapan untuk pembersihan data yang bertujuan untuk mengeliminasi berbagai elemen. Elemen-elemen ini mencakup simbol, karakter khusus seperti `{!@#$%^&*()[,:;".',/<>?}`, angka (0-9), dan emotikon [15].
- Case Folding** (Penyeragaman Huruf): Tahapan untuk mengonversi seluruh teks dalam dataset menjadi satu format huruf yang seragam, biasanya dari huruf kapital (*uppercase*) menjadi huruf kecil (*lowercase*). Hal ini penting agar kata yang sama namun memiliki kapitalisasi berbeda tidak dianggap sama [15].
- Normalization** (Normalisasi): Tahapan untuk mengoreksi kesalahan pengetikan (*typo*) dan mengubah kata-kata tidak baku, seperti singkatan atau bahasa gaul, menjadi bentuk kata yang baku [15]. Metode ini memanfaatkan kamus *slang* [16] yang berisi pemetaan dari kata tidak baku ke bentuk kata bakunya.
- Tokenizing** (Tokenisasi): Tahapan untuk memecahkan kalimat menjadi unit-unit yang lebih kecil yang disebut "token" sehingga menjadi kata-kata individual yang berdiri sendiri [15].
- Stopword Removal** (Penghapusan Kata): Tahapan untuk menyaring dan membuang kata-kata umum yang sering muncul namun tidak memiliki kontribusi yang signifikan dan memfokuskan pada kata-kata yang penting dan mewakili isi utama dari kalimat [15].

- f. *Stemming* (Pencarian Kata Dasar): Tahapan untuk mengubah kata menjadi bentuk dasarnya dengan menghilangkan semua imbuhan, baik awalan (*prefix*) maupun akhiran (*suffix*) dari kata tersebut [15].

2.4 Pelabelan Data

Pelabelan data dilakukan untuk mengkategorikan ulasan ke dalam dua kelas sentimen, yaitu positif atau negatif, berdasarkan jumlah bobot kata yang dihitung menggunakan kamus *Lexicon InSet* [17]. Penentuan label sentimen didasarkan pada dominasi bobot kata, di mana ulasan diberi label positif apabila bobot kata positif lebih dominan, dan diberi label negatif apabila bobot kata negatif lebih dominan.

2.5 Split Data

Split data dilakukan untuk membagi dataset menjadi dua bagian, yaitu data training (data latih) dan data testing (data uji). Rasio pembagian yang digunakan adalah 80:20, di mana 80% data dialokasikan sebagai data latih untuk melatih model, sedangkan 20% sisanya digunakan sebagai data uji untuk mengevaluasi kinerja objektif [18]. Dataset yang digunakan merupakan hasil pengumpulan data yang telah melalui proses *filtering* sehingga diperoleh 1.743 data bersih. Pembagian tersebut menghasilkan 1.394 data latih dan 349 data uji. Rasio 80:20 dipilih untuk memastikan ketersediaan data yang memadai pada tahap pelatihan serta memungkinkan evaluasi performa model secara objektif.

2.6 Pembobotan TF-IDF

Pembobotan TF-IDF dilakukan untuk menilai bobot sebuah kata dalam dokumen. Metode ini terdiri dari dua matriks perhitungan. Pertama, *Term Frequency* (TF), yaitu kalkulasi frekuensi kemunculan sebuah kata dalam dokumen. Kedua, *Inverse Document Frequency* (IDF), yang berfungsi mengidentifikasi seberapa langka sebuah kata pada seluruh dokumen (korpus) [19]. Adapun rumus perhitungan TF-IDF sebagai berikut:

$$W_{dt} = tf_{dt} \times IDF_t \quad (1)$$

Keterangan:

W_{dt} = nilai bobot dari kata (*term*) ke-t di dalam dokumen ke-d.

tf_{dt} = jumlah kemunculan kata ke-t pada dokumen ke-d.

IDF_t = perbandingan antara jumlah dokumen yang mengandung kata ke-t.

2.7 Klasifikasi Model

Klasifikasi model dilakukan untuk mengevaluasi seberapa baik kinerja model dalam memprediksi data. Proses pemodelan menggunakan tiga algoritma klasifikasi, yaitu *Naïve Bayes*, *Decision Tree*, dan *Random Forest*.

a. Naïve Bayes

Naïve Bayes adalah sebuah algoritma klasifikasi yang bekerja berdasarkan probabilitas. Algoritma ini menghitung peluang suatu data untuk menjadi bagian dari setiap kelas, kemudian memilih kelas dengan peluang terbesar sebagai hasil prediksinya [20]. Adapun rumus *Naïve Bayes* sebagai berikut:

$$P(X|Y) = \frac{P(Y|X) \cdot P(X)}{P(Y)} \quad (2)$$

Keterangan:

$P(X|Y)$ = probabilitas kejadian X terjadi jika diketahui bahwa Y terjadi.

$P(X)$ = probabilitas awal dari kejadian X tanpa mempertimbangkan kondisi.

$P(Y|X)$ = probabilitas kejadian Y terjadi dengan syarat X sudah terjadi.

$P(Y)$ = probabilitas total dari kejadian Y dalam keseluruhan data.

b. Decision Tree

Decision Tree merupakan algoritma klasifikasi yang memodelkan data dalam bentuk pohon keputusan dengan membagi dataset ke dalam cabang-cabang berdasarkan atribut tertentu. Setiap percabangan mewakili keputusan yang diambil menggunakan ukuran seperti *information gain* atau *gini index* untuk menentukan pemisahan terbaik [20]. Adapun rumus *Decision Tree* sebagai berikut:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \times Entropy(S_v) \quad (3)$$

Keterangan:

$Gain(S, A)$ = information gain atribut A pada dataset S .

$Entropy(S)$ = ukuran ketidakpastian dalam dataset S .

S_v = subset data dari S untuk nilai atribut $A = v$.

$|S_v| / |S|$ = proporsi jumlah data pada subset S_v dibanding total data.

c. Random Forest

Random Forest merupakan metode klasifikasi yang memanfaatkan prinsip *ensemble learning* dengan membangun banyak pohon keputusan. Algoritma ini secara acak memilih subset data dan fitur untuk melatih setiap pohon. Hasil dari semua pohon individu tersebut kemudian *diagregasi* (digabungkan) melalui proses *majority voting* [20]. Adapun rumus *Random Forest* sebagai berikut:

$$H(x) = \text{majority_vote} \{ h_1(x), h_2(x), \dots, h_k(x) \} \quad (4)$$

Keterangan:

$H(x)$ = hasil prediksi akhir untuk data x .

$h_1(x)$ = prediksi dari pohon keputusan ke- i .

k = jumlah pohon.

majority_vote = penggabungan hasil prediksi dengan memilih kelas terbanyak

2.8 Evaluasi Algoritma

Evaluasi algoritma dilakukan untuk menilai kinerja model klasifikasi yang telah dibangun. Proses ini memanfaatkan metode *Confusion Matrix*, yang merupakan sebuah tabel untuk membandingkan hasil prediksi model dengan label data yang aktual [21]. Adapun rumus *Confusion Matrix* sebagai berikut:

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

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

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

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

Keterangan:

TP: Prediksi positif yang tepat untuk data aktual positif.

TN: Prediksi negatif yang tepat untuk data aktual negatif.

FP: Prediksi positif yang keliru untuk data aktual negatif.

FN: Prediksi negatif yang keliru untuk data aktual positif

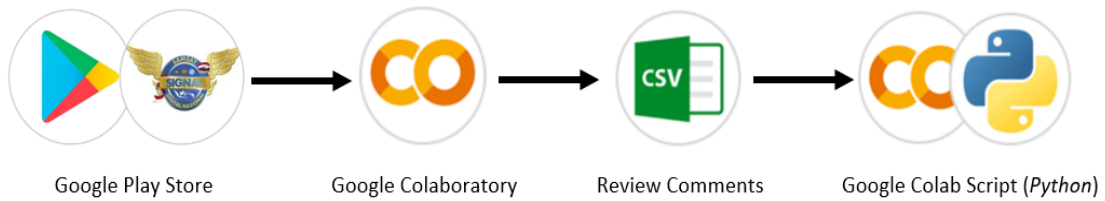
2.9 Visualisasi Data

Visualisasi data dilakukan untuk memudahkan pemahaman pola dalam penelitian. Visualisasi ini meliputi distribusi sentimen yang menampilkan kategori sentimen (positif atau negatif), distribusi rating yang menampilkan tingkat kepuasan pengguna, serta *word cloud* yang menampilkan kata-kata paling dominan [22].

3. HASIL DAN PEMBAHASAN

3.1 Pengumpulan Data

Pada tahap ini, peneliti menggunakan metode *web scraping* untuk memperoleh ulasan pengguna aplikasi Signal – Samsat Digital Nasional yang tersedia pada *Google Play Store*. Jumlah data yang di *scraping* sebanyak 2000 ulasan yang kemudian digunakan sebagai dataset dalam penelitian.



Gambar 2. Alur Pengumpulan Data

Gambar 2 menampilkan alur proses pengumpulan data menggunakan metode *web scraping* pada ulasan pengguna aplikasi SAMSAT di Google Play Store. Data dikumpulkan secara otomatis menggunakan *script Python* di *Google Colab* dan disimpan dalam *format CSV*. Dataset tersebut kemudian diproses menggunakan *script Python* di *Google Colab* untuk tahap pengolahan data. Hasil pengumpulan data ditampilkan pada Tabel 1.

Tabel 1. Ilustrasi Hasil *Web Scraping*

<i>index</i>	<i>reviewId</i>	<i>userName</i>	<i>userImage</i>	<i>content</i>	<i>score</i>	<i>thumbsUp Count</i>	<i>version</i>
1	cd33c99e-1a2a-4aa5-....	Ridwan Firmansyah	https://play-lh.googleusercontent.com/a-/	Aplikasi signal ini sangat membantu sekali...	5	0	1.7.3
2	145f0f6d-b7d1-4c8b-b4cd-....	Ria Apriana	https://play-lh.googleusercontent.com/a-/	ini aplikasi pemerintah kok gini sih?...	1	0	1.7.3

Tabel 1 menampilkan hasil pengumpulan ulasan pengguna yang berisi informasi seperti *reviewId*, *userName*, *userImage*, *content*, *score*, *thumbsUpCount* dan *version* yang menggambarkan identitas, isi, serta penilaian pengguna terhadap aplikasi. Data yang diperoleh masih bersifat mentah (*raw*) dan akan digunakan sebagai dasar untuk tahap pengolahan dan analisis selanjutnya.

3.2 Filter Data

Pada tahap ini, data hasil *web scraping* yang masih mentah (*raw*) difilter agar bersih. Proses ini mencakup penghapusan duplikat, entri kosong, serta penyortiran data. Kolom utama yang digunakan adalah *username*, *rating*, dan *review*, sedangkan kolom lain seperti *reviewId*, *userImage*, *thumbsUpCount*, dan *version* seperti pada Tabel 1 diabaikan karena tidak berpengaruh terhadap analisis. Hasil pemilihan kolom ditampilkan pada Tabel 2.

Tabel 2. Ilustrasi Filter Pemilihan Kolom

<i>username</i>	<i>rating</i>	<i>review</i>
Ridwan Firmansyah	5	Aplikasi signal ini sangat membantu sekali apalagi kami yang ada didaerah kabupaten yang belum ada samsatnya
Ria Apriana	1	ini aplikasi pemerintah kok gini sih? mau daftar pengesahan STNK malah muncul notif tidak berhasil

Tabel 2 menampilkan hasil data ulasan pengguna setelah melalui tahap *filtering*. Pada tahap ini, hanya kolom *username*, *rating*, dan *review* yang dipertahankan sebagai variabel utama penelitian. Sementara itu, kolom lain seperti *reviewId*, *userImage*, *thumbsUpCount*, dan *version* dihapus. Adapun total data setelah melakukan filter data ditampilkan pada Gambar 3.

Hasil Pengumpulan Data	Filter Sebelum Text-Processing	Filter Sesudah Text-Processing
<pre>Data columns (total 5 columns): # Column Non-Null Count Dtype --- - 0 id 2000 non-null object 1 username 2000 non-null object 2 rating 2000 non-null int64 3 review 2000 non-null object 4 date 2000 non-null object dtypes: int64(1), object(4) memory usage: 78.3+ KB</pre>	<pre>Data columns (total 3 columns): # Column Non-Null Count Dtype --- - 0 username 1753 non-null object 1 rating 1753 non-null int64 2 review 1753 non-null string dtypes: int64(1), object(1), string(1) memory usage: 41.2+ KB</pre>	<pre>Data columns (total 9 columns): # Column Non-Null Count Dtype --- - 0 username 1743 non-null object 1 rating 1743 non-null int64 2 review 1743 non-null string 3 cleaning 1743 non-null object 4 case_folding 1743 non-null object 5 normalization 1743 non-null object 6 tokenize 1743 non-null object 7 stopword_removal 1743 non-null object 8 stemming 1743 non-null object dtypes: int64(1), object(7), string(1) memory usage: 135.9+ KB</pre>

Gambar 3. Hasil Penghapusan Data Duplikat/Kosong

Gambar 3 menampilkan hasil proses filter data yang mencakup penghapusan data duplikat dan data kosong. Pada tahap awal, diperoleh 2.000 data dengan lima kolom utama, kemudian difilter menjadi 1.753 data dengan tiga kolom, yaitu *username*, *rating*, dan *review*. Selanjutnya, setelah tahap text processing yang meliputi *cleaning*, *case folding*, *normalization*, *tokenizing*, *stopword removal*, dan *stemming*, sehingga diperoleh 1.743 data bersih siap digunakan untuk analisis.

3.3 Text Processing

Pada tahap ini, data hasil *filtering* dilanjutkan ke *text processing* untuk mengubah data tidak terstruktur menjadi lebih terstruktur. Proses ini dilakukan untuk menormalkan data sehingga dapat diolah oleh algoritma *machine learning*. Beberapa langkah yang dilakukan adalah sebagai berikut:

Tabel 3. Ilustrasi Hasil Text Processing

Tahapan	Ulasan
Data Awal	Sya udah bayar pajak motor lewat aplikasi ini tapi udah hampir 3 bulan blm sampe ke rumah.....gimana ini.....kok malah jadi ribet
Cleaning	Sya udah bayar pajak motor lewat aplikasi ini tapi udah hampir bulan blm sampe ke rumah gimana ini kok malah jadi ribet
Case Folding	sya udah bayar pajak motor lewat aplikasi ini tapi udah hampir bulan blm sampe ke rumah gmn ini kok malah jadi ribet
Normalization	saya sudah bayar pajak motor lewat aplikasi ini tetapi sudah hampir bulan belum sampai ke rumah bagaimana ini kok malah jadi ribet
Tokenizing	['saya', 'sudah', 'bayar', 'pajak', 'motor', 'lewat', 'aplikasi', 'ini', 'tetapi', 'sudah', 'hampir', 'bulan', 'belum', 'sampai', 'ke', 'rumah', 'bagaimana', 'ini', 'kok', 'malah', 'jadi', 'ribet']
Stopword Removal	['bayar', 'pajak', 'motor', 'aplikasi', 'rumah', 'ribet']
Stemming	bayar pajak motor aplikasi rumah ribet

Tabel 3 menampilkan tahapan *text processing* pada data ulasan, mulai dari *cleaning*, *case folding*, *normalization*, *tokenizing*, *stopword removal* dan *stemming*. Setiap tahap ditunjukkan secara berurutan sehingga teks mentah berubah menjadi data yang siap untuk analisis.

3.4 Pelabelan Data

Pada tahap ini, pelabelan data dilakukan untuk mengkategorikan kelas sentimen sebagai dasar penelitian. Proses ini menggunakan kamus *Lexicon InSet* [20] untuk memberikan bobot pada setiap kata sehingga memudahkan proses pengkategorian secara otomatis. Ulasan diberi label positif jika bobot kata positif lebih dominan dan negatif jika bobot kata negatif lebih dominan. Hasil pelabelan data ditampilkan pada Tabel 4.

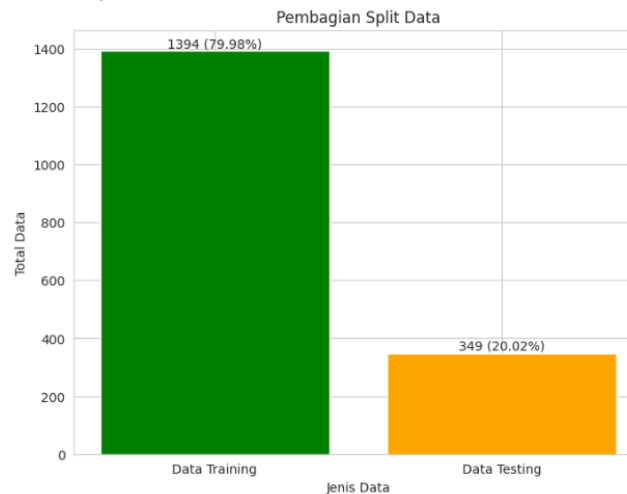
Tabel 4. Ilustrasi Hasil Transformasi Pelabelan Data

Hasil Stemming	Sentimen
gagal bayar pajak daftar gagal tolong sistem ganggu	Negatif
bayar aplikasi transfer apa tolong tindak lanjut	
aplikasi bantu bayar pajak tahun Kendara daring mudah	Positif
mana kota daerah moga depan mudah lancar	

Tabel 4 menampilkan hasil *stemming* terhadap ulasan pengguna yang telah diberi label sentimen menggunakan *Lexicon InSet*. Ulasan dengan bobot kata yang mengandung keluhan atau pengalaman kurang menyenangkan dikategorikan sebagai sentimen negatif, sedangkan ulasan dengan bobot kata yang menunjukkan apresiasi atau pujian dikategorikan sebagai sentimen positif.

3.5 Split Data

Pada tahap ini, dataset dibagi menjadi dua bagian, yaitu data *training* (alokasi 80%) dan data *testing* (alokasi 20%) melalui proses *split* data. Hasil *split* data ditampilkan pada Gambar 4.



Gambar 4. Hasil Pembagian Split Data

Gambar 4 menampilkan hasil pembagian dataset menjadi 1.394 data latih (80%) dan 349 data uji (20%). Rasio 80:20 dipilih agar sebagian besar data dapat digunakan untuk proses pelatihan model, sementara sisanya dimanfaatkan untuk menguji dan mengevaluasi performa model secara objektif.

3.6 Pembobotan TF-IDF

Pada tahap ini, dilakukan pembobotan TF-IDF untuk menilai tingkat kepentingan setiap kata dalam dokumen. Metode ini menghitung bobot kata berdasarkan frekuensi kemunculan (*Term Frequency*) dan tingkat kekhususannya dalam seluruh dokumen (*Inverse Document Frequency*), sehingga dapat mengidentifikasi kata-kata yang paling relevan dan berpengaruh terhadap analisis. Hasil transformasi pembobotan TF-IDF ditampilkan pada Tabel 5.

Tabel 5. Ilustrasi Hasil TF-IDF

<i>index</i>	<i>acab</i>	<i>account</i>	<i>app</i>	<i>adalagi</i>	<i>apdet</i>	<i>calo</i>	...	<i>jansip</i>	<i>admin</i>
0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0

Tabel 5 menampilkan hasil pembobotan TF-IDF, yang menunjukkan bobot setiap kata berdasarkan seberapa sering kata itu muncul. Apabila sebuah kata memiliki nilai bobot 0, ini mengindikasikan bahwa kata tersebut tidak ditemukan atau tidak memberi kontribusi pada dokumen.

3.7 Klasifikasi Model

Pada tahap ini, dilakukan klasifikasi untuk mengevaluasi kinerja model dalam memprediksi data berdasarkan pembobotan TF-IDF. Proses pemodelan menggunakan tiga algoritma, yaitu *Naïve Bayes*, *Decision Tree*, dan *Random Forest*. Hasil klasifikasi model ditampilkan pada Gambar 5.

<pre>from sklearn.naive_bayes import MultinomialNB # Latih Model Naive Bayes Dengan Data Training nb = MultinomialNB() nb.fit(tfidf_training, y_training) # Prediksi Data Testing y_pred = nb.predict(tfidf_testing)</pre>	<pre>from sklearn.tree import DecisionTreeClassifier # Latih Model Decision Tree Dengan Data Training tree = DecisionTreeClassifier(random_state=42) tree.fit(tfidf_training, y_training) # Prediksi Data Testing y_pred_tree = tree.predict(tfidf_testing)</pre>	<pre>from sklearn.ensemble import RandomForestClassifier # Latih Model Random Forest Dengan Data Training forest = RandomForestClassifier(n_estimators=100, random_state=42) forest.fit(tfidf_training, y_training) # Prediksi Data Testing y_pred_forest = forest.predict(tfidf_testing)</pre>
<p>MultinomialNB</p>	<p>DecisionTreeClassifier</p>	<p>RandomForestClassifier</p>

Gambar 5. Modelling Algoritma

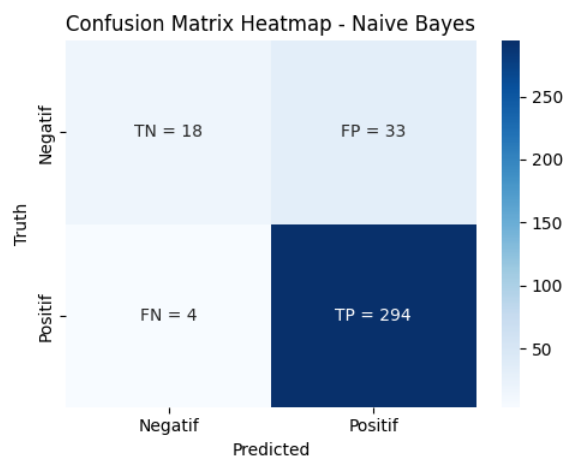
Gambar 5 menampilkan proses modelling menggunakan tiga algoritma klasifikasi dari *library scikit-learn*, yaitu *Naïve Bayes* dengan parameter *MultinomialNB()*, *Decision Tree* dengan parameter *DecisionTreeClassifier()*, dan *Random Forest* dengan parameter *RandomForestClassifier()*. Model dilatih menggunakan perintah *fit(tfidf_training, y_training)* pada data *training* dan diuji dengan perintah *predict(tfidf_testing)* pada data *testing*. Hasil dari ketiga model ini nantinya dibandingkan untuk mengevaluasi performa dan akurasi algoritma terbaik.

3.8 Evaluasi Algoritma

Pada tahap ini, dilakukan pengukuran kinerja terhadap model klasifikasi yang sudah dibangun. Model-model ini menggunakan tiga algoritma berbeda: *Naïve Bayes*, *Decision Tree*, dan *Random Forest*. Rincian dari hasil evaluasi ketiga algoritma itu adalah sebagai berikut.

3.8.1 Evaluasi Algoritma *Naïve Bayes*

Evaluasi algoritma *Naïve Bayes* divisualisasikan melalui *confusion matrix* pada Gambar 6, yang menunjukkan kemampuan model dalam memprediksi kelas positif dan negatif. Berdasarkan hasil tersebut, model menghasilkan 294 *True Positive* (TP), 4 *False Negative* (FN), 18 *True Negative* (TN), dan 33 *False Positive* (FP).



Gambar 6. Hasil Evaluasi *Naïve Bayes*

Gambar 6 menampilkan *confusion matrix heatmap* dari algoritma *Naïve Bayes* yang menunjukkan bahwa model memiliki kemampuan lebih baik dalam memprediksi kelas positif dibandingkan kelas negatif. Adapun hasil perhitungan evaluasi setiap kelas disajikan sebagai berikut:

1. Evaluasi Kelas Positif

$$Precision = \frac{TP}{TP + FP} = \frac{294}{294 + 33} = \frac{294}{327} = 0.899 \text{ (89,9\%)}$$

$$Recall = \frac{TP}{TP + FN} = \frac{294}{294 + 4} = \frac{294}{298} = 0.987 \text{ (98.7\%)}$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.899 \times 0.987)}{0.899 + 0.987} = 0.939 (93.9\%)$$

2. Evaluasi Kelas Negatif

$$Precision = \frac{TN}{TN + FN} = \frac{18}{18 + 4} = \frac{18}{22} = 0.818 (81.8\%)$$

$$Recall = \frac{TN}{TN + FP} = \frac{18}{18 + 33} = \frac{18}{51} = 0.353 (35.3\%)$$

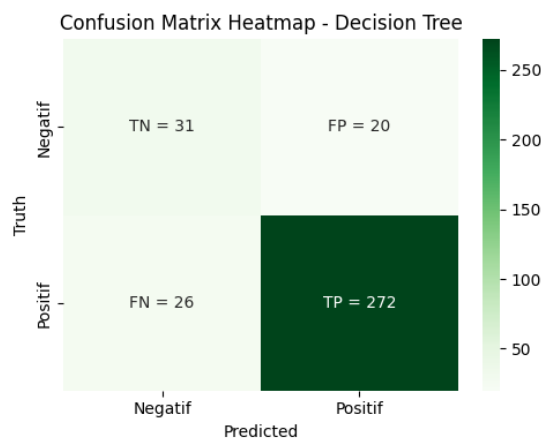
$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.818 \times 0.353)}{0.818 + 0.353} = 0.491 (49.1\%)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{294 + 18}{294 + 18 + 33 + 4} = \frac{312}{349} = 0.894 (89.4\%)$$

Dengan akurasi keseluruhan 89,4%, performa *Naïve Bayes* terbilang cukup baik. Model ini menunjukkan keandalan tinggi dalam mendeteksi sentimen positif, yang dibuktikan dengan nilai *precision* 89,9%, *recall* 98,7%, dan *f1-score* 93,9%. Akan tetapi, kinerjanya menurun drastis saat mendeteksi sentimen negatif, di mana nilai *recall*-nya hanya 35,3% (dengan *precision* 81,8% dan *f1-score* 49,1%). Dapat disimpulkan bahwa algoritma ini sangat baik untuk prediksi positif, tetapi lemah untuk prediksi negatif.

3.8.2 Evaluasi Algoritma Decision Tree

Evaluasi algoritma Decision Tree divisualisasikan melalui confusion matrix pada Gambar 7, yang menunjukkan kemampuan model dalam memprediksi kelas positif dan negatif. Berdasarkan hasil tersebut, model menghasilkan 272 *True Positive* (TP), 26 *False Negative* (FN), 31 *True Negative* (TN), dan 20 *False Positive* (FP).



Gambar 7. Hasil Evaluasi Decision Tree

Gambar 7 menampilkan *confusion matrix* heatmap dari algoritma *Decision Tree* yang menunjukkan bahwa model memiliki kemampuan lebih baik dalam memprediksi kelas positif dibandingkan kelas negatif. Adapun hasil perhitungan evaluasi setiap kelas disajikan sebagai berikut:

1. Evaluasi Kelas Positif

$$Precision = \frac{TP}{TP + FP} = \frac{272}{272 + 20} = \frac{272}{292} = 0.932 (93.2\%)$$

$$Recall = \frac{TP}{TP + FN} = \frac{272}{272 + 26} = \frac{272}{298} = 0.913 (91.3\%)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.932 \times 0.913)}{0.932 + 0.913} = 0.922 (92.2\%)$$

2. Evaluasi Kelas Negatif

$$Precision = \frac{TN}{TN + FN} = \frac{31}{31 + 26} = \frac{31}{57} = 0.544 (54.4\%)$$

$$Recall = \frac{TN}{TN + FP} = \frac{31}{31 + 20} = \frac{31}{51} = 0.608 (60.8\%)$$

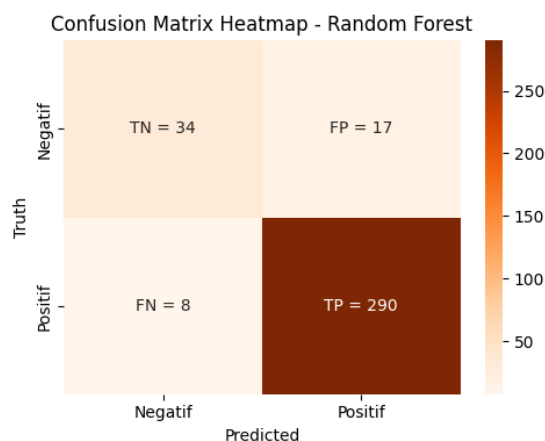
$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.544 \times 0.608)}{0.544 + 0.608} = 0.574 (57.5\%)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{272 + 31}{272 + 31 + 20 + 26} = \frac{303}{349} = 0.868 (86.8\%)$$

Dengan akurasi keseluruhan 86,8%, performa *Decision Tree* terbilang cukup baik. Model ini menunjukkan keandalan tinggi dalam mendeteksi sentimen positif, yang dibuktikan dengan nilai *precision* 93,2%, *recall* 91,3%, dan *f1-score* 92,2%. Akan tetapi, kinerjanya menurun saat mendeteksi sentimen negatif, di mana nilai *precision* hanya 54,4% (dengan *recall* 60,8% dan *f1-score* 57,5%). Dapat disimpulkan bahwa algoritma ini sangat baik untuk prediksi positif, namun masih kurang konsisten dalam mengidentifikasi sentimen negatif.

3.8.3 Evaluasi Algoritma Random Forest

Evaluasi algoritma *Random Forest* divisualisasikan melalui confusion matrix pada Gambar 8, yang menunjukkan kemampuan model dalam memprediksi kelas positif dan negatif. Berdasarkan hasil tersebut, model menghasilkan 290 *True Positive* (TP), 8 *False Negative* (FN), 34 *True Negative* (TN), dan 17 *False Positive* (FP).



Gambar 8. Hasil Evaluasi Random Forest

Gambar 8 menampilkan *confusion matrix heatmap* dari algoritma *Random Forest* yang menunjukkan bahwa model memiliki kemampuan lebih baik dalam memprediksi kelas positif dibandingkan kelas negatif. Adapun hasil perhitungan evaluasi setiap kelas disajikan sebagai berikut:

1. Evaluasi Kelas Positif

$$Precision = \frac{TP}{TP + FP} = \frac{290}{290 + 17} = \frac{290}{307} = 0.945 (94.5\%)$$

$$Recall = \frac{TP}{TP + FN} = \frac{290}{290 + 8} = \frac{290}{298} = 0.973 (97.3\%)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.945 \times 0.973)}{0.945 + 0.973} = 0.959 (95.9\%)$$

2. Evaluasi Kelas Negatif

$$Precision = \frac{TN}{TN + FN} = \frac{34}{34 + 8} = \frac{34}{42} = 0.810 (81\%)$$

$$Recall = \frac{TN}{TN + FP} = \frac{34}{34 + 17} = \frac{34}{51} = 0.667 (66.7\%)$$

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{(0.810 \times 0.667)}{0.810 + 0.667} = 0.731 (73.1\%)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{290 + 34}{290 + 34 + 17 + 8} = \frac{324}{349} = 0.929 (92.9\%)$$

Dengan akurasi keseluruhan 92,9%, performa *Random Forest* terbukti menjadi yang terbaik dibandingkan algoritma lainnya. Model ini menunjukkan keandalan sangat tinggi dalam mendeteksi sentimen positif, yang dibuktikan dengan nilai *precision* 94,5%, *recall* 97,3%, dan *f1-score* 95,9%. Selain itu, kinerjanya saat mendeteksi sentimen negatif juga jauh lebih baik, dengan *precision* 81,0%, *recall* 66,7%, dan *f1-score* 73,1%. Dapat disimpulkan bahwa algoritma ini memiliki kemampuan terbaik dalam mengklasifikasikan sentimen negatif di antara algoritma yang diuji.

3.9 Hasil Perbandingan Algoritma

Evaluasi terhadap algoritma *Naïve Bayes*, *Decision Tree*, dan *Random Forest* menunjukkan perbedaan performa dalam mengklasifikasikan sentimen. Setiap model memiliki efektivitas yang berbeda dalam membedakan sentimen positif dan negatif, dengan rincian perbandingan akurasi yang ditampilkan pada Tabel 6.

Tabel 6. Hasil Perbandingan Algoritma

Metrik	Algoritma					
	<i>Naïve Bayes</i>		<i>Decision Tree</i>		<i>Random Forest</i>	
	Sentimen		Sentimen		Sentimen	
	Positif	Negatif	Positif	Negatif	Positif	Negatif
Precision	0.899	0.820	0.932	0.544	0.945	0.810
Recall	0.987	0.353	0.913	0.608	0.973	0.667
F1-score	0.941	0.491	0.922	0.574	0.959	0.731
Accuracy	0.894		0.868		0.929	

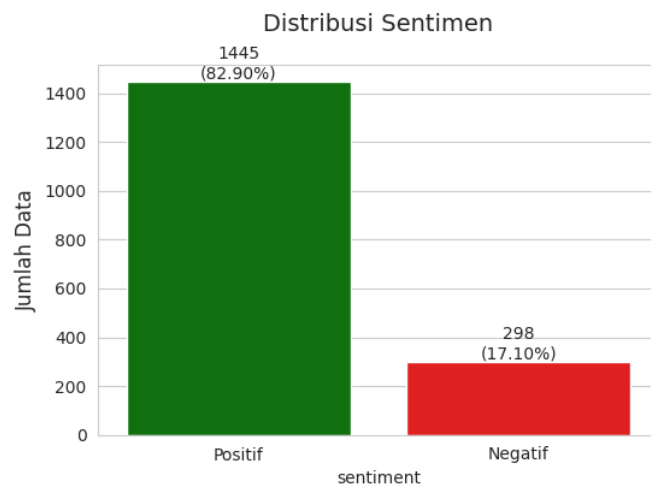
Tabel 6 menampilkan perbandingan hasil evaluasi tiga algoritma klasifikasi. *Random Forest* menunjukkan performa terbaik dengan akurasi tertinggi sebesar 92,9%, mengungguli *Naïve Bayes* (89,4%) dan *Decision Tree* (86,8%). Algoritma *Random Forest* lebih seimbang dalam mengenali sentimen positif dan negatif, sehingga menjadi yang paling efektif dalam penelitian ini.

3.10 Visualisasi Data

Pada tahap ini, visualisasi data ditampilkan untuk memudahkan pemahaman pola yang terkandung dalam dataset. Dibawah ini ditampilkan visualisasi yang meliputi distribusi sentimen, distribusi rating dan *word cloud* yang menampilkan kata-kata paling dominan yang dijabarkan sebagai berikut.

3.10.1 Distribusi Sentimen

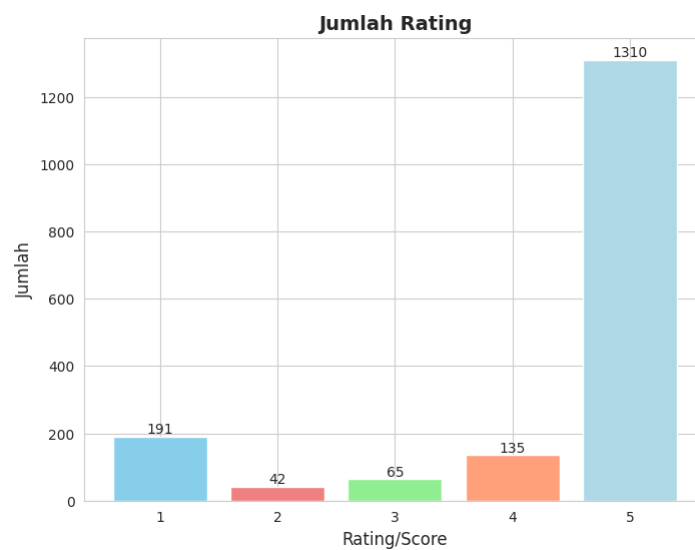
Distribusi sentimen menunjukkan ketidakseimbangan data, dari total 1743 data di mana sentimen positif (1.445 data atau 82,90%) jauh lebih dominan dibandingkan sentimen negatif (298 data atau 17,10%). Klasifikasi ini dilakukan menggunakan kamus *lexicon*. Hasil distribusi sentimen tersebut ditampilkan pada Gambar 9.



Gambar 9. Hasil Distribusi Sentimen

3.10.2 Distribusi Rating

Distribusi rating menampilkan rating dari 1 hingga 5, beserta jumlah pengguna yang memberikan ulasan. Dari total 1.743 data, terdapat rating 1 (191 data), rating 4 (135 data), rating 3 (65 data), dan rating 2 (42 data) dan rating 5 (1.310 data). Hasil distribusi rating tersebut ditampilkan pada Gambar 10.



Gambar 10. Hasil Distribusi Rating

3.10.3 Word Cloud

Dalam wordcloud ini, terlihat kata-kata yang sering digunakan oleh pengguna saat memberikan ulasan, seperti kata "bantu," "pajak," "mudah," dan "cepat. Kata-kata ini mencerminkan aspek yang dirasakan pengguna, seperti kemudahan, kenyamanan serta kepuasan terhadap aplikasi. Hasil visualisasi *word cloud* ditampilkan pada Gambar 11.



Gambar 11. Hasil *Word Cloud*

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan, dari total data sebanyak 2.000 ulasan yang diperoleh melalui proses *web scraping*, setelah melalui tahap *filter* dan *preprocessing* tersisa 1.743 data yang digunakan untuk analisis. Data ini dibagi menjadi dua bagian, yaitu data latih dengan 1.394 data dan data uji dengan 349 data. Hasil penelitian menunjukkan bahwa algoritma *Naïve Bayes* mendapatkan *accuracy* sebesar 89%, dengan *precision* sebesar 90% (positif) dan 82% (negatif), *recall* sebesar 99% (positif) dan 35% (negatif), serta *f1-score* sebesar 94% (positif) dan 49% (negatif). Sementara itu, algoritma *Decision Tree* mencapai *accuracy* sebesar 87%, dengan *precision* sebesar 93% (positif) dan 54% (negatif), *recall* sebesar 91% (positif) dan 61% (negatif), serta *f1-score* sebesar 92% (positif) dan 57% (negatif). Algoritma *Random Forest* menunjukkan hasil terbaik dengan *accuracy* sebesar 93%, *precision* sebesar 94% (positif) dan 81% (negatif), *recall* sebesar 97% (positif) dan 67% (negatif), serta *f1-score* sebesar 96% (positif) dan 73% (negatif). Pada visualisasi data yang ditampilkan pada distribusi rating, dari total 1743 data ulasan terdapat distribusi rating dengan dominasi rating bintang 5 sebanyak 1.310 ulasan, diikuti rating bintang 1 sebanyak 191 ulasan, rating bintang 4 sebanyak 135 ulasan, rating bintang 3 sebanyak 65 ulasan, dan rating bintang 2 sebanyak 42 ulasan. Sehingga terbentuk distribusi sentimen, di mana 82,9% ulasan dikategorikan sebagai sentimen positif dan 17,1% sebagai sentimen negatif. Berdasarkan distribusi ini, pengalaman pengguna aplikasi Signal – Samsat Digital Nasional dapat dikategorikan sangat positif, dengan mayoritas pengguna memberikan ulasan yang menunjukkan kepuasan terhadap layanan.

DAFTAR PUSTAKA

- [1] M. S. J. Sangaji and J. Irianto, "Transformasi Inovasi Pelayanan Publik menuju Pemerintahan Digital," *Jejaring Administrasi Publik*, vol. 17, no. 1, pp. 54–70, Jun. 2025, doi: 10.20473/jap.v17i1.72708.
- [2] S. N. Ulfa and A. Frinaldi, "Inovasi Pemerintah Daerah Dalam Penerapan Kebijakan Publik Dan Pelayanan Publik," *SOCIAL : Jurnal Inovasi Pendidikan IPS*, vol. 5, no. 2, pp. 610–621, Jun. 2025, doi: 10.51878/social.v5i2.5965.
- [3] D. Wijaya, R. A. Saputra, and F. Irwiensyah, "Analisis Sentimen Ulasan Aplikasi Samsat Digital Nasional Pada Google Playstore Menggunakan Algoritma Naïve Bayes," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 4, pp. 2369–2380, Feb. 2024, doi: 10.30865/klik.v4i4.1738.
- [4] A. Afifudin, A. Kadir, and I. Isnaini, "Implementasi Kebijakan Sistem Administrasi Manunggal Satu Atap di Kabupaten Mandailing Natal," *Strukturasi: Jurnal Ilmiah Magister Administrasi Publik*, vol. 2, no. 1, pp. 59–69, Apr. 2020, doi: 10.31289/strukturasi.v2i1.43.
- [5] Suriyani BB, La Ode Agus Said, and Astriana Syaqira, "Digitalisasi Layanan Samsat: Peningkatan Kepatuhan Pajak Kendaraan Bermotor Melalui e-Samsat di Kabupaten Muna," *Journal Publicuho*, vol. 7, no. 4, pp. 2111–2126, Dec. 2024, doi: 10.35817/publicuho.v7i4.573.
- [6] Afandi Afandi, Zainal fatah, Sri Roekminiati, and Ika Devy Pramudiana, "Inovasi Layanan Publik dalam Era Digital: Tinjauan Persepsi Generasi Muda pada Samsat Digital di Lumajang," *SOSIAL : Jurnal Ilmiah Pendidikan IPS*, vol. 3, no. 3, pp. 265–289, Jun. 2025, doi: 10.62383/sosial.v3i3.1064.
- [7] C. Triyosanti, D. F. Mufarokhah, S. N. Angelina, A. S. Cahyono, and D. I. Margayaningsih, "Dampak Penerapan Aplikasi Layanan Signal (Samsat Digital Nasional) Terhadap Pelayanan Pajak Kendaraan

- Bermotor di Tulungagung,” *JSPH : Jurnal Sosial Politik Humaniora*, vol. 1, no. 3, pp. 42–49, Dec. 2024, doi: 10.59966/jsph.v1i3.1426.
- [8] F. F. Irfani, “Analisis Sentimen Review Aplikasi Ruangguru Menggunakan Algoritma Support Vector Machine,” *JBMI (Jurnal Bisnis, Manajemen, dan Informatika)*, vol. 16, no. 3, pp. 258–266, Feb. 2020, doi: 10.26487/jbmi.v16i3.8607.
- [9] R. Chandra and E. M. Sipayung, “Analisis Sentimen Ulasan Aplikasi Samsat Digital Nasional Menggunakan Algoritma Naive Bayes Classifier,” *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 10, no. 3, pp. 156–164, Jan. 2025, doi: 10.25077/TEKNOSI.v10i3.2024.156-164.
- [10] A. Musthafa, T. Harmini, A. Rafiq, and N. Marantika, “Pemanfaatan Machine Learning dalam Menganalisis Sentimen Terhadap Program TAPERA di Platform Digital X,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 587–597, Mar. 2025, doi: 10.57152/malcom.v5i2.1801.
- [11] I. F. Rahman, A. N. Hasanah, and N. Heryana, “Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digiital Nasional (Signal) Dengan Menggunakan Metode Naïve Bayes Classifier,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4073.
- [12] I. F. Rahman, A. N. Hasanah, and N. Heryana, “Analisis Sentimen Ulasan Pengguna Aplikasi Samsat Digiital Nasional (Signal) Dengan Menggunakan Metode Naïve Bayes Classifier,” *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4073.
- [13] S. Kacung, C. Pamungkas Putra Bagyana, and D. Cahyono, “Analisis Sentimen Terhadap Layanan Samsat Digital Nasional (Signal) Menggunakan Metode SVM,” *Jurnal Mnemonic*, vol. 7, no. 1, pp. 118–122, Apr. 2024, doi: 10.36040/mnemonic.v7i1.9557.
- [14] S. A. R. Rizaldi, S. Alam, and I. Kurniawan, “Analisis Sentimen Pengguna Aplikasi JMO (Jamsostek Mobile) Pada Google Play Store Menggunakan Metode Naive Bayes,” *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 3, pp. 109–117, Aug. 2023, doi: 10.55123/storage.v2i3.2334.
- [15] Trisnaeni Faradaningsih and Anisa Lutfiyani, “Perbandingan Analisis Sentimen Pengguna Aplikasi Shopee dan Lazada pada Situs Google Play Store Menggunakan Algoritma K-Nearest Neighbor dan Naive Bayes,” *INSOLOGI: Jurnal Sains dan Teknologi*, vol. 4, no. 3, pp. 563–578, Jun. 2025, doi: 10.55123/insologi.v4i3.5646.
- [16] Fornigulo, “Kamus Slang.” Accessed: Mar. 05, 2025. [Online]. Available: <https://kaggle.com/datasets/fornigulo/kamus-slag>
- [17] F. Koto and G. Y. Rahmanningtyas, “Inset lexicon: Evaluation of a word list for Indonesian sentiment analysis in microblogs,” in *2017 International Conference on Asian Language Processing (IALP)*, IEEE, Dec. 2017, pp. 391–394. doi: 10.1109/IALP.2017.8300625.
- [18] A. S. Pamungkas and N. Cahyono, “Analisis Sentimen Review ChatGPT di Play Store menggunakan Support Vector Machine dan K-Nearest Neighbor,” *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 1, pp. 1–10, Jun. 2024, doi: 10.29408/edumatic.v8i1.24114.
- [19] Gilbert, Syariful Alam, and M. Imam Sulisty, “Analisis Sentimen Berdasarkan Ulasan Pengguna Aplikasi Mypertamina Pada Google Playstore Menggunakan Metode Naïve Bayes,” *STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer*, vol. 2, no. 3, pp. 100–108, Aug. 2023, doi: 10.55123/storage.v2i3.2333.
- [20] I. N. Sari, M. K. L. F. Lidimilah, and M. K. A. Lutfi, “Analisis Perbandingan Algoritma Decision Tree, Random Forest, dan Naive Bayes dalam Klasifikasi Gangguan Tidur,” *Proceedings National Conference Sinesia*, vol. 1, no. 1, pp. 192–209, Jun. 2025, doi: 10.69836/ncrcs-sinesia.v1i1.41.
- [21] M. Tirta Nugraha, N. Nina Sulistiyowati, and U. Ultach Enri, “Analisis Sentimen Ulasan Aplikasi Satu Sehat Pada Google Play Store Menggunakan Naïve Bayes Classifier,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 5, pp. 3593–3601, Jan. 2024, doi: 10.36040/jati.v7i5.7753.
- [22] N. Cahyono and Anggista Oktavia Praneswara, “Analisis Sentimen Ulasan Aplikasi TikTok Shop Seller Center di Google Playstore Menggunakan Algoritma Naive Bayes,” *Indonesian Journal of Computer Science*, vol. 12, no. 6, Dec. 2023, doi: 10.33022/ijcs.v12i6.3473.