JIPTI
JURNAL INOVASI PENDIDIKAN DAN TEKNOLOGI INFORMASI

Volume 7 No. 2, November (2026)
E-ISSN: 2720-9725
P-ISSN: 2987-8462

PERFORMA *SUPPORT VECTOR MACHINE* (SVM) DAN *RANDOM FOREST* UNTUK KLASIFIKASI SENTIMEN ULASAN APLIKASI elSantri

Supri Arrohman[1*], Zaehol Fatah[2], Ahmad Homaidi[3]
Universitas Ibrahimy[123]
E-mail: supriarrohman23@gmail.com[1*], zaeholfatah@gmail.com[2], ahmadhomaidi@ibrahimy.ac.id[3]

Abstrak

Perkembangan teknologi digital mendorong penggunaan aplikasi mobile dalam layanan administrasi pesantren, salah satunya aplikasi elSantri. Berbagai ulasan pengguna menunjukkan adanya keluhan terkait kinerja aplikasi sehingga diperlukan analisis sentimen untuk memahami opini pengguna. Penelitian ini bertujuan menganalisis sentimen ulasan pengguna aplikasi elSantri dan membandingkan performa algoritma *Support Vector Machine* (SVM) dan *Random Forest*. Penelitian menggunakan metode *Knowledge Discovery in Database* (KDD) dengan data sebanyak 383 ulasan yang diperoleh dari Google Play Store. Tahapan penelitian meliputi *preprocessing*, pembobotan TF-IDF, klasifikasi sentimen, dan evaluasi menggunakan *confusion matrix*. Hasil penelitian menunjukkan bahwa SVM memperoleh akurasi 83,12%, presisi 83%, *recall* 83%, dan *F1-score* 83%, sedangkan *Random Forest* memperoleh akurasi 77,92%, presisi 78%, *recall* 78%, dan *F1-score* 77%. Kebaruan penelitian ini terletak pada penerapan dan perbandingan kedua algoritma pada ulasan aplikasi elSantri yang merupakan aplikasi administrasi digital berbasis pesantren.

Kata Kunci: Analisis Sentimen; *Support Vector Machine*; *Random Forest*; TF-IDF; elSantri.

Abstract

The advancement of digital technology has encouraged the use of mobile applications in Islamic boarding school administrative services, one of which is the elSantri application. User reviews indicate several issues related to application performance, making sentiment analysis necessary to understand user opinions. This study aims to analyze user sentiments toward the elSantri application and compare the performance of Support Vector Machine (SVM) and Random Forest algorithms. The study employed the Knowledge Discovery in Database (KDD) method using 383 reviews collected from the Google Play Store. The research stages included preprocessing, TF-IDF weighting, sentiment classification, and evaluation using a confusion matrix. The results show that SVM achieved an accuracy of 83.12%, while Random Forest achieved 77.92%.

Arrohman, S., Fatah, Z., Homaidi, A. (2026). PERFORMA SUPPORT VECTOR MACHINE (SVM) DAN RANDOM FOREST UNTUK KLASIFIKASI SENTIMEN ULASAN APLIKASI elSantri. *Jurnal Inovasi Pendidikan dan Teknologi Informasi (JIPTI), 7(2), 541-554.* https://doi.org/10.52060/jipti.v7i2.4454

http://ejournal.ummba.ac.id/index.php/JIPTI/

The novelty of this study lies in the implementation and comparison of both algorithms on reviews of the elSantri application, a digital Islamic boarding school administration application.

Keywords: *Sentiment Analysis; Support Vector Machine; Random Forest; TF-IDF; elSantri.*

Submitted: *2026-07-01.* ***Revision:*** *2026-08-12.* ***Accepted:*** *2026-10-04.* ***Publish:*** *2026-11-02.*

PENDAHULUAN

Perkembangan teknologi informasi telah mendorong transformasi digital pada berbagai sektor, termasuk pendidikan dan administrasi pesantren. Digitalisasi layanan administrasi dapat meningkatkan efektivitas dan efisiensi pelayanan kepada pengguna, khususnya dalam pengelolaan informasi dan transaksi yang berkaitan dengan kebutuhan santri dan wali santri. Salah satu penerapannya adalah aplikasi elSantri yang menyediakan berbagai layanan administrasi pesantren, seperti informasi profil santri, pembayaran administrasi, riwayat transaksi, data kesehatan, pelanggaran, prestasi, serta pengajuan perizinan. Keberadaan aplikasi tersebut memberikan kemudahan dalam mengakses layanan administrasi secara digital. Namun, pengalaman pengguna terhadap suatu aplikasi dapat berbeda-beda dan tercermin melalui ulasan yang diberikan pada Google Play Store. Ulasan tersebut dapat berupa apresiasi, keluhan, maupun saran yang menggambarkan persepsi pengguna terhadap kualitas layanan aplikasi. Karena jumlah ulasan terus bertambah dan berbentuk data teks tidak terstruktur, diperlukan pendekatan analisis sentimen untuk mengidentifikasi opini pengguna ke dalam kategori positif, negatif, dan netral. Analisis sentimen merupakan salah satu penerapan text mining dan Natural Language Processing (NLP) yang dapat digunakan untuk mengidentifikasi kecenderungan opini pengguna terhadap suatu produk atau layanan.

Penelitian terdahulu menunjukkan bahwa metode machine learning telah banyak digunakan untuk melakukan klasifikasi sentimen pada berbagai objek. (Petiwi, Triayudi, & Sholihati, 2022) menerapkan Naïve Bayes dan *Support Vector Machine* (SVM) untuk menganalisis sentimen berdasarkan data Twitter terkait layanan GoFood. (Larasati, Ratnawati, & Hanggara, 2022) menggunakan *Random Forest* untuk menganalisis sentimen ulasan aplikasi DANA. (Ndapamuri, Manongga, & Iriani, 2023) membandingkan SVM, K-Nearest Neighbor, dan Naïve Bayes pada ulasan aplikasi Tripadvisor, sedangkan (Setiawan & Kaburuan, 2023) menerapkan SVM untuk analisis ulasan aplikasi Digital Korlantas. Penelitian lain juga menunjukkan penerapan SVM dengan pembobotan TF-IDF pada analisis sentimen, seperti penelitian (Emarapenta, Sinulingga, Cesar, & Sitorus, 2024), sementara (Prasetyo, Utami, & Yaqin, 2024) mengkaji pengaruh komposisi pembagian data terhadap performa klasifikasi sentimen. (Givari, Sulaeman, & Umaidah, 2022) juga membandingkan SVM, *Random Forest*, dan XGBoost pada permasalahan klasifikasi. Hasil penelitian-penelitian tersebut menunjukkan bahwa SVM dan *Random*

Forest memiliki potensi untuk digunakan dalam permasalahan klasifikasi, tetapi performanya dapat berbeda bergantung pada karakteristik data, objek penelitian, serta tahapan pengolahan data yang diterapkan.

Meskipun penelitian analisis sentimen telah banyak dilakukan pada aplikasi layanan digital, media sosial, perbankan, dan layanan lainnya, kajian yang secara khusus membahas ulasan pengguna aplikasi administrasi pesantren masih terbatas. Selain itu, penelitian yang secara langsung membandingkan performa SVM dan *Random Forest* pada ulasan aplikasi elSantri belum banyak ditemukan. Kondisi tersebut menunjukkan adanya kesenjangan penelitian berupa keterbatasan kajian analisis sentimen pada konteks aplikasi administrasi digital berbasis pesantren serta belum jelasnya algoritma yang memberikan performa lebih baik pada karakteristik ulasan pengguna elSantri. Penelitian ini mengisi kesenjangan tersebut dengan menerapkan dan membandingkan algoritma SVM dan *Random Forest* menggunakan pembobotan TF-IDF pada data ulasan aplikasi elSantri. Perbandingan dilakukan berdasarkan *accuracy*, *precision*, *recall*, dan *F1-score* untuk memperoleh gambaran performa kedua algoritma secara menyeluruh. Dengan demikian, penelitian ini bertujuan untuk menganalisis sentimen pengguna aplikasi elSantri dan menentukan algoritma yang memiliki performa klasifikasi lebih baik dalam mengidentifikasi sentimen positif, negatif, dan netral pada ulasan pengguna.

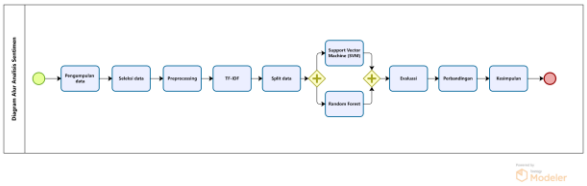
METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode *Knowledge Discovery in Database* (KDD) sebagai kerangka kerja pengolahan dan analisis data. Pendekatan kuantitatif digunakan karena penelitian mengukur performa algoritma klasifikasi berdasarkan nilai *accuracy*, *precision*, *recall*, dan *F1-score*. Metode KDD digunakan untuk mengolah data ulasan pengguna secara sistematis mulai dari pemilihan data hingga evaluasi hasil klasifikasi. Tahapan KDD dalam penelitian ini terdiri atas selection, *preprocessing*, transformation, data mining, dan evaluation.

Pada tahap selection, data yang digunakan berupa ulasan pengguna aplikasi elSantri yang diperoleh dari Google Play Store dan dipilih sesuai dengan kebutuhan penelitian. Tahap *preprocessing* dilakukan untuk membersihkan dan menyeragamkan data teks melalui proses *case folding*, *cleaning*, *tokenizing*, *normalization*, *stopword removal*, dan *stemming*. Selanjutnya, tahap transformation dilakukan dengan mengubah data teks hasil *preprocessing* menjadi representasi numerik menggunakan pembobotan Term Frequency-Inverse Document Frequency (TF-IDF). Pada tahap data mining, data yang telah ditransformasikan digunakan untuk membangun model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM) dan *Random Forest*. Kedua algoritma tersebut dipilih untuk membandingkan kemampuan model berbasis *hyperplane* dengan model *ensemble* dalam

mengklasifikasikan data teks hasil pembobotan TF-IDF. SVM digunakan karena sesuai untuk klasifikasi data berdimensi tinggi dan bersifat *sparse*, sedangkan *Random Forest* digunakan sebagai model pembanding berbasis *ensemble* untuk melihat perbedaan performa kedua pendekatan pada dataset yang sama. Tahap terakhir, yaitu evaluation, dilakukan dengan mengukur performa kedua algoritma menggunakan *confusion matrix* serta parameter *accuracy*, *precision*, *recall*, dan *F1-score*. Alur keseluruhan tahapan penelitian disajikan pada Gambar 1.



Gambar 1. Diagram Alur Analisis Sentimen

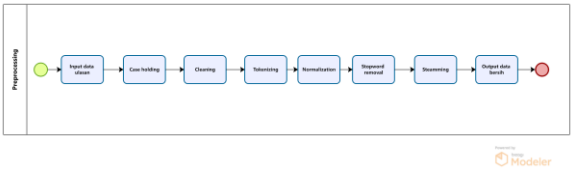
B. Teknik Pengumpulan Data

Populasi dalam penelitian ini berupa ulasan pengguna aplikasi elSantri yang tersedia pada Google Play Store. Pengumpulan data dilakukan secara manual pada halaman resmi aplikasi elSantri di Google Play Store. Data yang dikumpulkan berupa ulasan pengguna yang tersedia hingga 20 Oktober 2025. Setiap ulasan yang diperoleh dicatat sebagai satu data observasi dan kemudian dikumpulkan dalam format CSV untuk memudahkan proses pengolahan data.

Berdasarkan proses pengumpulan tersebut diperoleh sebanyak 383 ulasan pengguna yang selanjutnya digunakan sebagai dataset penelitian. Jumlah 383 data merupakan keseluruhan data ulasan yang berhasil dikumpulkan sampai dengan tanggal 20 Oktober 2025 dan sesuai dengan kebutuhan penelitian. Data tersebut kemudian diberi label sentimen secara manual ke dalam tiga kelas, yaitu positif, netral, dan negatif, sebelum memasuki tahap *preprocessing* dan klasifikasi.

C. Tahapan Penelitian

Tahap *preprocessing* dilakukan melalui beberapa tahapan, yaitu *case folding*, *cleaning*, *tokenizing*, *normalization*, *stopword removal*, dan *stemming* (Fadhlurrohman, Primajaya, & Dimyati, 2025). Tahapan *preprocessing* yang diterapkan dalam penelitian ini ditunjukkan pada Gambar 2.



Gambar 2. Diagram Alur Preprocesing

Setelah proses *preprocessing* selesai, data kemudian ditransformasikan menggunakan metode TF-IDF untuk memberikan bobot pada setiap kata (Emarapenta et al., 2024). Dataset selanjutnya dibagi menjadi data *training* dan data *testing* dengan perbandingan 80:20 (Prasetyo et al., 2024). Proses klasifikasi sentimen dilakukan menggunakan algoritma *Support Vector Machine* dan *Random Forest*. Sementara itu, evaluasi performa model dilakukan menggunakan *confusion*

matrix dengan parameter akurasi, presisi, *recall*, dan F1-skor.

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Data

Data penelitian diperoleh dari ulasan pengguna aplikasi elSantri yang tersedia pada Google Play Store. Total data yang berhasil dikumpulkan sebanyak 383 ulasan pengguna yang terdiri atas sentimen positif, negatif, dan netral. Seluruh data kemudian disimpan dalam format CSV untuk mempermudah proses pengolahan dan analisis data menggunakan bahasa pemrograman Python.

Hasil Preprocesing Data

Tahap *preprocessing* dilakukan untuk membersihkan serta mempersiapkan data teks agar dapat diproses dengan lebih optimal oleh algoritma klasifikasi. Proses *preprocessing* yang diterapkan meliputi *case folding*, *cleaning*, *tokenizing*, *normalization*, *stopword removal*, dan *stemming*.

Case folding merupakan tahap yang dilakukan dengan mengubah seluruh huruf pada teks menjadi huruf kecil (lowercase) untuk menghindari perbedaan makna antara huruf kapital dan huruf kecil. Sedangkan, *Cleaning* merupakan tahap yang dilakukan dengan menghapus berbagai karakter yang tidak diperlukan, seperti tanda baca, angka, URL, emoji, dan simbol lainnya agar data teks menjadi lebih bersih dan mudah diproses. Hasil penerapan *case folding* dan *cleaning* terhadap data ulasan ditampilkan pada Tabel 1.

Tabel 1. *Case Folding* dan *Cleaning*

Input	Output
Aplikasinya bagus dan mudah tapi untuk top up saldo sangat susah	aplikasinya bagus dan mudah tapi untuk top up saldo sangat susah
Aplikasinya bagus tapi untuk loginnya jangan terlalu sering	aplikasinya bagus tapi untuk loginnya jangan terlalu sering
Keluhan-keluhan sebelumnya sudah didengar dan diperbaiki	keluhan keluhan sebelumnya sudah didengar dan diperbaiki
Hasil update 1 september sudah bisa cek saldo	hasil update september sudah bisa cek saldo
Tidak bisa login kak, pake mode lupa pin tapi tetap gagal	tidak bisa login kak pake mode lupa pin tapi tetap gagal
aplikasinya bagus!!! 👍	aplikasinya bagus

Tahap selanjutnya adalah *tokenizing*, *tokenizing* merupakan tahap yang dilakukan dengan memecah kalimat menjadi beberapa kata atau token. Hasil dari *tokenizing* terhadap data ulasan ditampilkan pada Tabel 2.

Tabel 2. *Tokenizing*

Input	Output
Aplikasinya bagus dan mudah tapi untuk top up saldo sangat susah	[aplikasinya, bagus, dan, mudah, tapi, untuk, top, up, saldo, sangat, susah]
Aplikasinya bagus tapi untuk loginnya jangan terlalu sering	[aplikasinya, bagus, tapi, untuk, loginnya, jangan,

Input	Output
	terlalu, sering]
Keluhan-keluhan sebelumnya sudah didengar dan diperbaiki	[keluhan, keluhan, sebelumnya, sudah, didengar, dan, diperbaiki]
Hasil update 1 september sudah bisa cek saldo	[hasil, update, september, sudah, bisa, cek, saldo]
Tidak bisa login kak, pake mode lupa pin tapi tetap gagal	[tidak, bisa, login, kak, pake, mode, lupa, pin, tapi, tetap, gagal]

Tahap berikutnya adalah *normalization*, *normalization* merupakan proses untuk mengubah kata tidak baku maupun singkatan menjadi kata baku agar data teks lebih konsisten. Hasil tahap *normalization* disajikan pada Tabel 3.

Tabel 3. *Normalization*

Input	Output
Aplikasinya bagus dan mudah tapi untuk top up saldo sangat susah	aplikasi bagus dan mudah tetapi untuk isi saldo sangat susah
Aplikasinya bagus tapi untuk loginnya jangan terlalu sering	aplikasi bagus tetapi login jangan terlalu sering
Keluhan-keluhan sebelumnya sudah didengar dan diperbaiki	keluhan sebelumnya sudah dengar dan perbaiki
Hasil update 1 september sudah bisa cek saldo	hasil pembaruan sudah bisa cek saldo
Tidak bisa login kak, pake mode lupa pin tapi tetap gagal	tidak bisa login menggunakan mode lupa pin tetapi tetap gagal

Kemudian stop removal, stop removal merupakan tahap yang dilakukan untuk menghapus kata-kata umum yang tidak memiliki makna penting dalam proses klasifikasi. Sementara itu, *stemming* dilakukan untuk mengubah kata yang memiliki imbuhan menjadi bentuk kata dasar. Tahap ini menghasilkan data yang disajikan pada Tabel 4.

Tabel 4. *Stopword removal* dan *Stemming*

Input	Output
Aplikasinya bagus dan mudah tapi untuk top up saldo sangat susah	aplikasi bagus mudah isi saldo susah
Aplikasinya bagus tapi untuk loginnya jangan terlalu sering	aplikasi bagus login sering
Keluhan-keluhan sebelumnya sudah didengar dan diperbaiki	keluhan dengar baik
Hasil update 1 september sudah bisa cek saldo	hasil pembaruan cek saldo
Tidak bisa login kak, pake mode lupa pin tapi tetap gagal	login mode lupa pin gagal

Tahapan *preprocessing* tersebut membantu meningkatkan kualitas data sehingga proses klasifikasi dapat berjalan lebih optimal.

Hasil Transformasi Data Menggunakan TF-IDF

Setelah proses *preprocessing* selesai dilakukan, data teks selanjutnya diubah ke dalam bentuk numerik menggunakan metode TF-IDF (Term Frequency–Inverse Document Frequency). Term Frequency (TF) digunakan untuk mengukur tingkat kemunculan suatu kata dalam dokumen, di mana semakin sering kata tersebut muncul maka semakin besar pula nilai TF yang diperoleh. Nilai TF dapat dihitung menggunakan rumus berikut:

$$TF = \frac{\text{Jumlah kemunculan kata dalam dokumen}}{\text{Jumlah total kata dalam dokumen}}$$

Inverse Document Frequency (IDF) mengukur seberapa penting dengan memperhitungkan seberapa sering kata tersebut muncul dalam semua dokumen. N melambangkan jumlah total dokumen dalam kumpulan teks seperti pada rumus berikut:

$$IDF = \log \frac{N}{n}$$

Dimana N merupakan jumlah seluruh dokumen dalam kumpulan data, sedangkan n adalah jumlah dokumen yang memuat kata tersebut. Setelah nilai TF dan IDF diperoleh, nilai TF-IDF dapat dihitung dengan mengkalikan nilai TF dengan IDF. Kata-kata yang dimiliki nilai TF-IDF yang tinggi dianggap penting dan memberi kontribusi yang lebih besar dalam menentukan topik dokumen atau kumpulan dokumen tersebut (Firzatullah & Nuroji, 2025).

Metode ini digunakan untuk memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya di dalam dokumen. Kata yang memiliki frekuensi kemunculan tinggi pada suatu dokumen tetapi jarang ditemukan pada dokumen lain akan memperoleh bobot yang lebih besar. Dengan menggunakan metode TF-IDF, data teks dapat diubah menjadi representasi vektor numerik sehingga dapat diproses menggunakan algoritma machine learning.

Hasil Pembagian Dataset

Dataset yang telah melalui tahap *preprocessing* dan transformasi kemudian dibagi menjadi data latih dan data uji dengan rasio 80:20. Sebanyak 80% data digunakan sebagai data latih untuk membentuk model klasifikasi, sedangkan 20% data lainnya digunakan sebagai data uji untuk mengevaluasi performa model. Pembagian data latih dan data uji berdasarkan masing-masing kelas sentimen ditampilkan pada Tabel 5.

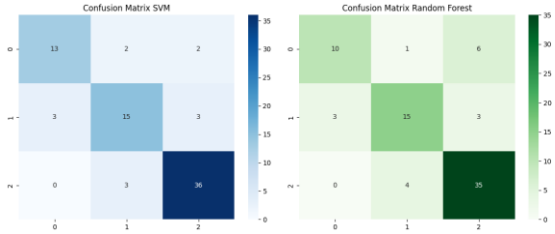
Tabel 5. Data Latih dan Data Uji

Kelas Sentimen	Jumlah Data	Data Latih 80%	Data Uji 20%
Positif	194	155	39
Negatif	84	67	17
Netral	105	84	21
Total	383	306	77

Confusion Matrix

Hasil pengujian model klasifikasi sentimen dengan menerapkan algoritma

Support Vector Machine dan *Random Forest* disajikan dalam bentuk *confusion matrix* pada Gambar 3. *Confusion matrix* digunakan untuk mengetahui jumlah prediksi yang benar maupun prediksi yang salah pada masing-masing kelas sentimen.



Gambar 3. *Confusion Matrix* Algoritma SVM dan *Random Forest*

Pada algoritma *Support Vector Machine*, nilai prediksi benar berada pada bagian diagonal *confusion matrix*, yaitu sebanyak 13 data pada kelas 0, 15 data pada kelas 1, dan 36 data pada kelas 2. Hasil tersebut menunjukkan bahwa model SVM mampu mengenali pola sentimen dengan cukup baik pada sebagian besar ulasan pengguna aplikasi elSantri.

Sementara itu, pada algoritma *Random Forest* jumlah prediksi benar diperoleh sebanyak 10 data pada kelas 0, 15 data pada kelas 1, dan 35 data pada kelas 2. Walaupun *Random Forest* mampu mengklasifikasikan sebagian besar data dengan cukup baik, masih ditemukan beberapa kesalahan klasifikasi, khususnya pada kelas 1 yang diprediksi ke kelas lain. Hal tersebut menyebabkan performa *Random Forest* sedikit berada di bawah algoritma SVM.

Berdasarkan hasil *confusion matrix* tersebut, algoritma *Support Vector Machine* menunjukkan kemampuan klasifikasi yang lebih stabil dibandingkan *Random Forest* karena mampu menghasilkan distribusi prediksi yang lebih baik pada seluruh kelas sentimen.

Hasil Evaluasi Model

Evaluasi model dilakukan menggunakan *confusion matrix* dengan parameter *accuracy*, *precision*, *recall*, dan *F1-score*. Hasil pengujian menunjukkan bahwa algoritma *Support Vector Machine* (SVM) memperoleh nilai akurasi sebesar 83,12%, sedangkan algoritma *Random Forest* memperoleh nilai akurasi sebesar 77,92%. Hasil tersebut menunjukkan bahwa SVM memiliki kemampuan klasifikasi yang lebih baik dibandingkan *Random Forest* dalam mengklasifikasikan sentimen ulasan aplikasi elSantri. Hasil perbandingan dapat dilihat pada Tabel 6.

Tabel 6. Perbandingan Hasil Evaluasi Model

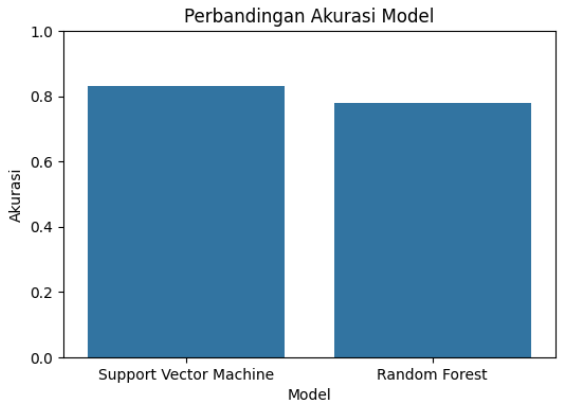
Metode	Akurasi	presisi	recall	F1-Skor
SVM	0.8312	0.83	0.83	0.83
RF	0.7792	0.78	0.78	0.77

Berdasarkan *classification report*, algoritma SVM memperoleh nilai *precision* sebesar 81% pada kelas negatif, 75% pada kelas netral, dan 88% pada kelas positif. Nilai recall masing-masing sebesar 76%, 71%, dan 92%, sedangkan nilai *F1-score* mencapai 79%, 73%, dan 90%. Hasil tersebut menunjukkan bahwa SVM mampu mengidentifikasi sentimen positif dengan sangat baik serta memiliki keseimbangan performa yang cukup baik pada seluruh kelas sentimen.

Sementara itu, algoritma *Random Forest* memperoleh nilai *precision* sebesar 77% pada kelas negatif, 75% pada kelas netral, dan 80% pada kelas positif. Nilai *recall* yang diperoleh masing-masing sebesar 59%, 71%, dan 90%, dengan *F1-score* sebesar 67%, 73%, dan 84%. Nilai *recall* yang relatif rendah pada kelas negatif menunjukkan bahwa *Random Forest* masih mengalami kesulitan dalam mengenali seluruh data sentimen negatif secara optimal.

Perbandingan Akurasi Model

Perbandingan nilai akurasi menunjukkan bahwa algoritma *Support Vector Machine* memperoleh akurasi sebesar 83,12%, lebih tinggi dibandingkan *Random Forest* yang memperoleh akurasi sebesar 77,92%. Selisih akurasi sebesar 5,20% menunjukkan bahwa SVM lebih efektif dalam mengolah data teks ulasan aplikasi elSantri yang memiliki karakteristik berdimensi tinggi dan bersifat *sparse* setelah proses pembobotan TF-IDF. Hasil perbandingan nilai *accuracy* antara algoritma SVM dan *Random Forest* dapat dilihat pada Grafik 1.

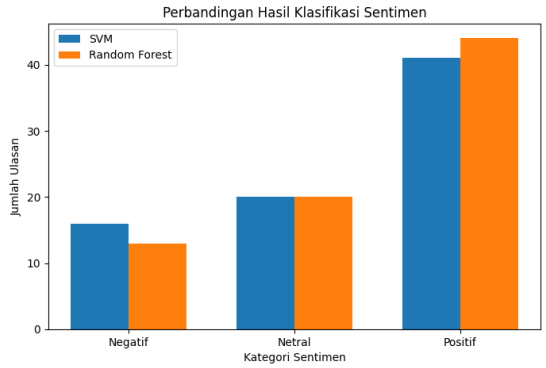


Grafik 1. Perbandingan Akurasi SVM dan *Random Forest*

Keunggulan SVM dipengaruhi oleh kemampuannya dalam membangun hyperplane optimal yang mampu memisahkan kelas sentimen positif, negatif, dan netral secara lebih akurat. Sebaliknya, *Random Forest* menghasilkan performa yang lebih rendah dibandingkan SVM. Salah satu faktor yang diduga berkontribusi adalah karakteristik representasi TF-IDF yang berdimensi tinggi dan sparse, sehingga pendekatan berbasis *decision tree* pada konfigurasi yang digunakan belum mampu memberikan performa sebaik SVM.

Perbandingan Jumlah Sentimen

Hasil klasifikasi menggunakan algoritma SVM menghasilkan 16 data sentimen negatif, 20 data sentimen netral, dan 41 data sentimen positif. Sementara itu, *Random Forest* menghasilkan 13 data sentimen negatif, 20 data sentimen netral, dan 44 data sentimen positif. Hasil klasifikasi sentimen disajikan pada Grafik 2.



Grafik 2. Perbandingan Hasil Klasifikasi Sentimen

Distribusi tersebut menunjukkan bahwa kedua algoritma cenderung mengklasifikasikan sebagian besar ulasan ke dalam kategori sentimen positif. Namun, SVM mampu mendeteksi sentimen negatif lebih banyak dibandingkan *Random Forest*. Hal ini menunjukkan bahwa SVM memiliki sensitivitas yang lebih baik dalam mengenali ulasan yang mengandung keluhan atau ketidakpuasan pengguna terhadap aplikasi elSantri.

Secara keseluruhan, hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine* merupakan metode yang lebih efektif dibandingkan *Random Forest* untuk klasifikasi sentimen ulasan aplikasi elSantri, dengan nilai akurasi, *precision*, *recall*, dan *F1-score* yang lebih tinggi pada sebagian besar kelas sentimen.

Pembahasan

Penelitian ini dilakukan untuk membandingkan performa algoritma *Support Vector Machine* dan *Random Forest* dalam klasifikasi sentimen ulasan aplikasi elSantri. Proses penelitian dimulai dengan pengumpulan data ulasan pengguna aplikasi elSantri yang diperoleh dari Google Play Store, kemudian dilanjutkan dengan tahap *preprocessing*, transformasi data menggunakan TF-IDF, proses klasifikasi, hingga evaluasi model menggunakan *confusion matrix*.

Hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine* memperoleh nilai *accuracy* sebesar 83,12%, *precision* sebesar 83%, *recall* sebesar 83%, dan *F1-score* sebesar 83%. Sementara itu, algoritma *Random Forest* memperoleh nilai akurasi sebesar 77,92%, presisi sebesar 78%, *recall* sebesar 78%, dan F1-skor sebesar 77%. Berdasarkan hasil evaluasi tersebut, algoritma SVM menunjukkan kinerja yang lebih unggul dibandingkan *Random Forest* dalam proses klasifikasi sentimen ulasan aplikasi elSantri.

Performa *Support Vector Machine* yang lebih unggul dipengaruhi oleh karakteristik data teks yang berdimensi tinggi dan bersifat *sparse* (Junaidi, Anggela, & Kariman, 2024). Dalam klasifikasi teks, jumlah fitur yang dihasilkan dari proses ekstraksi TF-IDF umumnya sangat banyak karena setiap kata direpresentasikan sebagai fitur numerik. Algoritma SVM mampu menangani kondisi tersebut dengan baik karena bekerja menggunakan hyperplane optimal untuk memisahkan kelas sentimen positif, negatif, dan netral secara maksimal (Jollyta, Hajjah, Haerani, & Siddik, 2023). Selain itu, kemampuan generalisasi yang dimiliki SVM membuat model lebih stabil dalam melakukan klasifikasi terhadap data baru.

Selain itu, penggunaan kernel pada algoritma *Support Vector Machine* memungkinkan model untuk memisahkan data yang kompleks sehingga kemampuan klasifikasi sentimen menjadi lebih baik. Kondisi tersebut membuat SVM lebih efektif dalam mengenali pola opini pengguna pada ulasan aplikasi elSantri. Oleh karena itu, algoritma SVM banyak digunakan dalam penelitian analisis

sentimen karena memiliki tingkat akurasi yang tinggi dalam pengolahan data teks.

Di sisi lain, algoritma *Random Forest* juga menunjukkan kinerja yang cukup baik dalam klasifikasi sentimen. *Random Forest* bekerja dengan menggabungkan sejumlah *decision tree* untuk menghasilkan keputusan akhir melalui metode *majority voting* (Ghozali, Pratiwi, & Handajani, 2023). Algoritma ini memiliki keunggulan dalam mengurangi overfitting serta mampu menangani noise pada data (Rachbini, 2025). Namun, pada penelitian ini performa *Random Forest* masih berada di bawah *Support Vector Machine* karena data teks hasil TF-IDF memiliki karakteristik *sparse* dan berdimensi tinggi yang kurang optimal apabila diproses menggunakan banyak pohon keputusan.

Tahapan *preprocessing* pada penelitian ini juga memberikan pengaruh yang sangat penting terhadap hasil klasifikasi. Proses *cleaning* berfungsi untuk menghapus karakter yang tidak relevan seperti simbol, angka, dan tanda baca sehingga data menjadi lebih bersih. Tahap *normalization* digunakan untuk mengubah kata tidak baku menjadi kata baku agar variasi kata dapat diminimalkan. Selain itu, proses *stopword removal* dan *stemming* membantu mengurangi kata-kata yang tidak memiliki makna penting sehingga fitur yang digunakan dalam proses klasifikasi menjadi lebih efektif.

Penggunaan metode TF-IDF pada tahap transformasi data juga memberikan kontribusi yang cukup besar terhadap performa model klasifikasi. TF-IDF mampu memberikan bobot pada setiap kata berdasarkan tingkat kepentingannya di dalam dokumen sehingga kata-kata yang lebih merepresentasikan sentimen akan memiliki bobot yang lebih tinggi. Dengan demikian, model machine learning dapat mengenali pola sentimen secara lebih baik dan menghasilkan proses klasifikasi yang lebih akurat.

Berdasarkan hasil klasifikasi, algoritma *Support Vector Machine* menghasilkan 16 ulasan dengan sentimen negatif, 20 ulasan dengan sentimen netral, dan 41 ulasan dengan sentimen positif. Sementara itu, algoritma *Random Forest* menghasilkan 13 ulasan dengan sentimen negatif, 20 ulasan dengan sentimen netral, dan 44 ulasan dengan sentimen positif.

Dominasi sentimen positif pada kedua algoritma menunjukkan bahwa sebagian besar pengguna memberikan tanggapan yang baik terhadap aplikasi elSantri. Hal ini mengindikasikan bahwa aplikasi telah mampu memenuhi kebutuhan pengguna dalam mengakses layanan administrasi pesantren secara digital, seperti pembayaran administrasi, pemantauan informasi santri, dan layanan lainnya.

Meskipun demikian, masih ditemukan sejumlah ulasan dengan sentimen negatif yang menunjukkan adanya permasalahan yang dirasakan pengguna, seperti kesulitan login, terjadinya error pada sistem, ketidaksesuaian data, maupun kendala teknis lainnya. Kehadiran sentimen negatif tersebut dapat menjadi bahan evaluasi bagi

pengembang untuk meningkatkan kualitas layanan dan stabilitas aplikasi.

Perbedaan jumlah sentimen yang dihasilkan kedua algoritma menunjukkan adanya perbedaan kemampuan dalam mengenali pola sentimen pada data ulasan. Algoritma *Support Vector Machine* mengidentifikasi lebih banyak sentimen negatif dibandingkan Random Forest, yaitu sebanyak 16 ulasan berbanding 13 ulasan. Hal ini menunjukkan bahwa SVM memiliki sensitivitas yang lebih baik dalam mendeteksi ulasan yang mengandung unsur keluhan atau ketidakpuasan pengguna. Sebaliknya, *Random Forest* cenderung mengklasifikasikan lebih banyak data ke dalam kategori sentimen positif sehingga jumlah sentimen positif yang dihasilkan lebih tinggi.

Hasil tersebut sejalan dengan nilai evaluasi model yang menunjukkan bahwa *Support Vector Machine* memiliki nilai *accuracy*, *precision*, *recall*, dan *F1-score* yang lebih tinggi dibandingkan *Random Forest*. Dengan demikian, distribusi sentimen yang dihasilkan oleh SVM dinilai lebih representatif dalam menggambarkan opini pengguna aplikasi elSantri karena mampu mengidentifikasi sentimen positif, netral, dan negatif secara lebih seimbang.

Hasil penelitian ini menunjukkan bahwa Support Vector Machine memperoleh akurasi sebesar 83,12%, sedangkan Random Forest memperoleh akurasi sebesar 77,92%. Hasil tersebut menunjukkan bahwa SVM memberikan performa yang lebih baik pada dataset ulasan aplikasi elSantri setelah melalui tahapan preprocessing dan pembobotan TF-IDF. (Afkar Nabila & Wahyu Pamungkas, 2025) yang mengatakan bahwa performa *Random Forest* lebih baik dari pada SVM dan *XGBoost* dengan tinggkat akurasi sebesar 98,61% AUC-ROC 98,96%. Temuan ini tidak sejalan karena menunjukkan bahwa Random Forest memiliki performa lebih baik dibandingkan SVM dan XGBoost. Perbedaan hasil tersebut dapat dipengaruhi oleh karakteristik data dan objek penelitian yang digunakan. Penelitian Afkar Nabila dan Wahyu Pamungkas berfokus pada prediksi stroke, sedangkan penelitian ini menggunakan data teks berupa ulasan pengguna aplikasi elSantri yang memiliki karakteristik tidak terstruktur dan memerlukan tahapan preprocessing serta pembobotan TF-IDF. Perbedaan karakteristik data tersebut dapat menyebabkan algoritma memberikan performa yang berbeda pada masing-masing penelitian. Pada data ulasan elSantri, hasil evaluasi menunjukkan bahwa SVM memperoleh akurasi 83,12%, lebih tinggi dibandingkan Random Forest sebesar 77,92%. (Petiwi et al., 2022) yang menerapkan SVM untuk analisis sentimen sejalan dengan temuan ini dan menunjukkan bahwa SVM mampu memberikan performa yang baik dalam klasifikasi data teks dengan tingkat akurasi sebesar 83% dan 98,5%, sedangkan algoritma pembandingnya *Naïve Bayes* memiliki tingkat akurasi sebesar 74,6% dan 91,5%. Hasil yang serupa juga ditunjukkan oleh (Ndapamuri et al., 2023) dalam analisis sentimen ulasan aplikasi Tripadvisor yang menggunakan SVM, K-Nearest Neighbor, dan Naive Bayes dengan

tingkat akurasi SVM sebesar 89,8% lebih tinggi dibanding algoritma lainnya.

KESIMPULAN

Hasil penelitian menunjukkan bahwa algoritma *Support Vector Machine* (SVM) memiliki kinerja lebih baik dibandingkan *Random Forest* dalam klasifikasi sentimen ulasan aplikasi elSantri. SVM memperoleh akurasi sebesar 83,12%, sedangkan *Random Forest* sebesar 77,92%, dengan nilai *precision*, *recall*, dan *F1-score* SVM yang juga lebih tinggi pada sebagian besar kelas sentimen. Sentimen positif menjadi sentimen yang dominan pada ulasan pengguna, sementara sentimen negatif menunjukkan adanya keluhan terkait kendala teknis dan kualitas layanan aplikasi. Temuan ini memberikan bukti empiris bahwa SVM lebih efektif untuk klasifikasi sentimen pada ulasan aplikasi administrasi digital berbasis pesantren dan menjadi kontribusi penelitian dalam penerapan perbandingan kedua algoritma pada konteks aplikasi elSantri.

DAFTAR PUSTAKA

Afkar Nabila, H., & Wahyu Pamungkas, E. (2025). PERBANDINGAN ALGORITMA MACHINE LEARNING: SVM, RANDOM FOREST, DAN XGBOOST UNTUK PREDIKSI STROKE. *RABIT: Jurnal Teknologi Dan Sistem Informasi Univrab*, *10*(2), 1098–1110. https://doi.org/10.36341/rabit.v10i2.64

Emarapenta, J., Sinulingga, B., Cesar, H., & Sitorus, K. (2024). Analisis Sentimen Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF Sentiment Analysis of Public towards Indonesian Horror Films Using SVM and TF- IDF Methods. *Jurnal Manajemen Informatika (JAMIKA)*, *14*(April), 42–53. https://doi.org/10.34010/jamika.v14i1.11946

Fadhlurrohman, N. A., Primajaya, A., & Dimyati, A. N. (2025). Analisis Sentimen Terhadap Skema Student Loan Untuk Biaya Perguruan Tinggi Pada Twitter. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *9*(2), 2115–2123. https://doi.org/10.36040/jati.v9i2.13001

Firzatullah, & Nuroji. (2025). Analisis Sentimen Pengguna Aplikasi Byond BSI Pada Google Play Store Menggunakan Algoritma SVM Dan Random Forest. *METIK JURNAL Media Teknologi Informasi Dan Komputer Jurnal*, *9*(2), 356–363. https://doi.org/10.47002/6vtc4567

Ghozali, A., Pratiwi, H., & Handajani, S. S. (2023). Implementasi Data Mining Menggunakan Metode Random Forest Dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes. *Delta: Jurnal Ilmiah Pendidikan Matematika*, *11*(2), 147–162. https://doi.org/10.31941/delta.v11i2.2686

Givari, M. R., Sulaeman, M. R., & Umaidah, Y. (2022). Perbandingan Algoritma SVM, Random Forest, Dan XGBoost Untuk Penentuan Persetujuan Pengajuan Kredit. *JURNAL NUANSA INFORMATIKA*, *16*(1), 141–149. https://doi.org/10.25134/nuansa.v16i1.5406

Jollyta, D., Hajjah, A., Haerani, E., & Siddik, M. (2023). *Algoritma Klasifikasi untuk Pemula Solusi Python dan RapidMiner*. Deepublish. Retrieved from https://books.google.co.id/books?id=y8

4TEQAAQBAJ

Junaidi, S., Anggela, R. V., & Kariman, D. (2024). Journal Of Applied Computer Science And Technology (JACOST) Klasifikasi Metode Data Mining untuk Prediksi Kelulusan Tepat Waktu Mahasiswa dengan Algoritma Naïve Bayes , Random Forest , Support Vector Machine (SVM) dan Artificial Neural Network (ANN), *5*(1), 109–119. https://doi.org/10.52158/jacost.v5i1.489

Larasati, F. A., Ratnawati, D. E., & Hanggara, B. T. (2022). Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *6*(9), 4305–4313. Retrieved from https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/11562

Ndapamuri, A. M., Manongga, D., & Iriani, A. (2023). Analisis Sentimen Ulasan Aplikasi Tripadvisor Dengan Metode Support Vector Machine, K-Nearest Neighbor, Dan Naive Bayes. *JURNAL INOVTEK POLBENG - SERI INFORMATIKA*, 127–140. https://doi.org/10.35314/isi.v8i1.3260

Petiwi, M. I., Triayudi, A., & Sholihati, I. D. (2022). Analisis Sentimen Gofood Berdasarkan Twitter Menggunakan Metode Naïve Bayes dan Support Vector Machine. *MIB Jurnal Media Informatika Budidarma*, *6*, 542–550. https://doi.org/10.30865/mib.v6i1.3530

Prasetyo, Y. A., Utami, E., & Yaqin, A. (2024). Pengaruh Komposisi Split Data Terhadap Performa Akurasi Analisis Sentimen Algoritma Naïve Bayes dan SVM. *Journal of Electrical Engineering and Computer (JEECOM)*, *6*(2), 382–390. https://doi.org/10.33650/jeecom.v6i2.9

188

Rachbini, W. (2025). *DATA SCIENCE untuk Pemula dengan RapidMiner*. CV. AA. Rizky. Retrieved from https://books.google.co.id/books?id=95lMEQAAQBAJ

Setiawan, N. R., & Kaburuan, E. R. (2023). Sentimen Analisis Review Aplikasi Digital Korlantas Pada Google Play Store Menggunakan Metode SVM. *JURNAL SISFOKOM (Sistem Informasi Dan Komputer)*, *12*, 105–116. https://doi.org/10.32736/sisfokom.v12i1.1614