Jurnal Sains Informatika Terapan (JSIT)

E-ISSN: 2828-1659, Volume: 5, Issue: 1, Month: February, Year: 2026
Page: 279 - 286, Available online at: https://rcf-indonesia.org/home/

ANALISIS PERBANDINGAN ALGORITMA RANDOM FOREST,DECISION TREE,DAN NAIVE BAYES DALAM MENDETEKSI SPAM SMS

[1] Vincentius Jason, [2] Yosefina Finsensia Riti,[3] Rafael Valentino Patrick

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Katolik Darma Cendika, Surabaya, Indonesia

Corresponding Author:[1]yosefina.riti@ukdc.ac.id

Article Info

Article history:

Received: Jan 10, 2026
Revised: Feb 02, 2026
Accepted: Feb 06, 2026
Published: Feb 07, 2026

Keywords:

SMS Spam Detection
Machine Learning
Random Forest
Naive Bayes
Text Classification
TF-IDF
Decision Tree

ABSTRACT

Dalam penelitian ini, dilakukan analisis bandingan yang mendalam terhadap tiga algoritma machine learning untuk mendeteksi spam SMS, yaitu Random Forest, Decision Tree, dan Naive Bayes. Dataset UCI SMS Spam Collection yang memiliki 5.572 pesan digunakan, dan pipeline penuh dijalankan mulai dari menghapus duplikat, ekstraksi fitur TF-IDF, sampai augmentasi data apabila perlu. Pada tahap prapemrosesan, terdapat 403 pesan duplikat yang dibuang (sekitar 7,2%), sehingga akhirnya tersisa 5.169 sampel unik. Model-model ini dilatih dengan split data 80-20 untuk latih dan uji, dan dievaluasi secara menyeluruh menggunakan metrik seperti akurasi, presisi, recall, F1-score, serta matriks kebingungan. Hasilnya menunjukkan ketiga algoritma ini memiliki performa yang sangat tinggi, dengan Naive Bayes yang paling unggul dengan akurasi mencapai 98,1%, lalu Random Forest 97,8%, dan Decision Tree 96,4%. Dari analisis kurva pembelajaran, model konvergen dengan baik dan tidak terlalu overfit. Penelitian ini juga memberikan enam visualisasi lengkap, mulai dari analisis duplikat, distribusi data, word cloud, breakdown split data latih-uji, kurva pembelajaran, sampai matriks kebingungan. Pendekatan machine learning klasik ternyata masih sangat efektif untuk deteksi spam SMS.

1. INTRODUCTION

Spam merupakan salah satu bentuk penyalahgunaan komunikasi digital yang tidak etis, di mana seseorang mengirim pesan secara massal tanpa izin, dan isinya sering kali tidak relevan atau bahkan berbahaya. Tidak hanya di email, masalah ini juga sering muncul di SMS atau pesan langsung di berbagai platform daring. Biasanya, isi pesan spam berupa promosi yang agresif, tautan mencurigakan, atau ajakan penipuan yang dibuat menyerupai pesan asli agar terlihat sah.

Masalah spam semakin serius karena tidak hanya mengganggu kenyamanan, tetapi juga menimbulkan risiko keamanan. Pesan spam sering digunakan untuk phishing, menyebarkan malware, atau penipuan daring lainnya. Teknologi komunikasi yang berkembang justru membuat pelaku spam semakin leluasa, sehingga metode penyaringan lama tidak dapat diandalkan dalam jangka panjang.

Metode penyaringan spam yang menggunakan aturan-aturan, seperti daftar kata kunci dan pola tertentu, sudah lama digunakan. Namun, cara ini memiliki keterbatasan, misalnya harus diperbarui secara manual terus-menerus, tidak dapat menyesuaikan diri dengan cepat terhadap variasi pesan baru, dan mudah dihindari apabila teksnya disamarkan. Oleh karena itu, dibutuhkan pendekatan yang lebih fleksibel dan dapat mengikuti pola spam yang berubah-ubah.

Machine learning menawarkan solusi yang lebih adaptif karena dapat mempelajari pola dari data yang sudah diberi label secara otomatis. Dalam bidang klasifikasi teks, algoritma machine learning sudah menunjukkan hasil yang baik untuk membedakan spam dan pesan normal. Kinerja model bergantung pada dataset, cara prapemrosesan, dan algoritma yang dipilih. Di antara yang sering digunakan, algoritma Random Forest, Decision Tree, dan Naive

Bayes memiliki performa stabil dalam banyak penelitian.

Random Forest adalah metode ensemble yang menggabungkan banyak pohon keputusan untuk hasil prediksi yang lebih kuat. Cara ini dapat mengurangi risiko overfitting dan akurat untuk data yang kompleks. Decision Tree unggul dari segi kemudahan pemahaman karena strukturnya jelas, tetapi rentan terhadap overfitting apabila parameter tidak diatur dengan benar. Sementara itu, Naive Bayes adalah algoritma probabilistik yang sederhana dan hemat komputasi, tetapi tetap efektif untuk klasifikasi teks meskipun menggunakan asumsi independensi antarfitur.

Walaupun penelitian deteksi spam sudah banyak dilakukan, kebanyakan hanya berfokus pada satu algoritma tanpa perbandingan yang lengkap. Selain itu, aspek prapemrosesan seperti menghapus duplikasi data dan representasi fitur sering kurang dibahas secara detail. Padahal, hal-hal tersebut sangat penting untuk kualitas dan kemampuan model.

Dari hal tersebut, penelitian ini bertujuan untuk melakukan analisis komparatif secara mendalam terhadap algoritma Random Forest, Decision Tree, dan Naive Bayes dalam deteksi spam SMS. Analisisnya mencakup semua tahap, mulai dari prapemrosesan teks, penanganan duplikasi data, ekstraksi fitur menggunakan TF-IDF, sampai evaluasi performa dengan metrik dan visualisasi yang beragam. Diharapkan hasilnya dapat menjelaskan kelebihan dan kekurangan masing-masing algoritma sehingga dapat menjadi panduan untuk membuat sistem deteksi spam yang efektif dan efisien.

Di sisi lain, Decision Tree memiliki keunggulan dalam hal interpretabilitas karena struktur aturannya yang berbentuk if–then–else mudah dipahami. Seperti dijelaskan oleh Quinlan (1986) yang memperkenalkan konsep dasar algoritma ini, model tersebut cenderung mengalami overfitting pada data pelatihan jika tidak dikendalikan dengan teknik seperti pruning atau pembatasan kedalaman pohon [2].

Sementara itu, Naive Bayes merupakan algoritma probabilistik yang sangat efisien secara komputasi dan telah terbukti efektif untuk klasifikasi teks, termasuk deteksi spam. Berdasarkan penelitian Rish (2001), model ini mampu memberikan performa tinggi meskipun menggunakan asumsi independensi antarfitur [16].

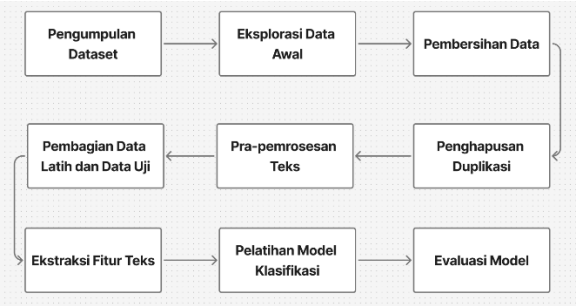
Meskipun berbagai penelitian telah menerapkan algoritma machine learning untuk deteksi spam, sebagian besar studi sebelumnya berfokus pada satu pendekatan tertentu tanpa melakukan perbandingan performa antarmodel algoritma secara menyeluruh. Pendekatan berbasis aturan masih digunakan dalam beberapa sistem lama, namun sifatnya yang kaku membuatnya mudah dihindari oleh spammer yang terus beradaptasi. Di sisi lain, metode machine learning seperti Random Forest, Decision Tree, dan Naive Bayes terbukti menjanjikan, tetapi performanya sangat bergantung pada karakteristik data, teknik prapemrosesan, serta konfigurasi algoritma yang diterapkan [6], [7].

Oleh karena itu, penelitian ini bertujuan untuk melakukan analisis komparatif secara komprehensif terhadap ketiga algoritma tersebut dalam konteks deteksi komentar spam. Analisis dilakukan mencakup seluruh tahapan implementasi, mulai dari prapemrosesan teks, penanganan duplikasi data, ekstraksi fitur menggunakan TF-IDF, hingga evaluasi performa dengan metrik dan visualisasi yang mendalam.

Dengan pendekatan ini, diharapkan penelitian dapat memberikan pemahaman yang lebih mendalam mengenai keunggulan dan keterbatasan masing-masing algoritma dalam penerapan nyata serta berkontribusi terhadap pengembangan sistem deteksi spam yang lebih adaptif dan efisien.

2.MATERIALS AND METHODS

2.1 Dataset dan Metodologi Penelitian



Gambar 2.1 Metodologi Penelitian

Dataset penelitian yang digunakan adalah SMS Spam Collection dari Almeida dan Hidalgo (2011), yang tersedia di UCI Machine Learning Repository dan Kaggle [17], [18]. Dataset ini berisi 5.574 pesan SMS, dengan 4.827 pesan ham (86,4%) dan 747 pesan spam (13,4%). Karakteristik dataset ini mencerminkan skenario dunia nyata, di mana kelas spam lebih sedikit dibandingkan ham sehingga memerlukan penanganan khusus untuk menghindari bias model.

Penelitian ini memanfaatkan dataset SMS Spam Collection yang tersedia pada UCI Machine Learning Repository. Dataset tersebut dikembangkan oleh Almeida dan Gómez Hidalgo dan telah banyak

digunakan sebagai standar evaluasi dalam penelitian deteksi spam berbasis teks. Data yang digunakan berisi kumpulan pesan SMS berbahasa Inggris yang telah diberi label secara manual ke dalam dua kelas, yaitu spam dan ham (pesan sah).

Secara keseluruhan, dataset awal terdiri dari 5.572 pesan SMS, yang mencerminkan kondisi lalu lintas pesan nyata di mana jumlah pesan sah jauh lebih dominan dibandingkan pesan spam. Distribusi kelas menunjukkan bahwa 4.825 pesan (86,6%) dikategorikan sebagai ham, sedangkan 747 pesan (13,4%) merupakan spam. Ketidakseimbangan ini merupakan karakteristik umum pada permasalahan deteksi spam di dunia nyata dan menjadi tantangan tersendiri dalam proses pemodelan.

Data dikumpulkan dari berbagai sumber, termasuk korpus SMS publik dan situs web yang menghimpun laporan spam. Setiap pesan disimpan dalam format teks sederhana dengan dua atribut utama, yaitu label kelas dan isi pesan. Struktur data yang sederhana menjadikan dataset ini cocok untuk eksperimen klasifikasi teks berbasis machine learning.

Hasil eksplorasi awal menunjukkan adanya sejumlah pesan yang muncul lebih dari satu kali dalam dataset. Pesan-pesan tersebut umumnya berupa respons singkat atau template yang sering digunakan berulang. Untuk menjaga validitas evaluasi model dan mencegah terjadinya kebocoran data antara data latih dan data uji, pesan duplikat diidentifikasi dan dihapus secara sistematis. Setelah proses penghapusan duplikasi dilakukan, dataset akhir terdiri dari 5.169 pesan unik, yang selanjutnya digunakan pada seluruh tahap eksperimen.

2.2 Pra-pemrosesan data

Tahapan prapemrosesan data dirancang untuk memastikan bahwa data yang digunakan berada dalam kondisi optimal sebelum dimasukkan ke tahap pelatihan model. Proses ini melibatkan serangkaian langkah penting yang dilakukan secara bertahap.

Pertama, proses dimulai dengan memuat data dan melakukan eksplorasi awal. Dataset diambil dari file teks yang terstruktur, berisi label kelas dan isi pesan. Pada tahap ini, dilakukan analisis statistik sederhana, seperti melihat distribusi panjang pesan, jumlah kata, dan proporsi setiap kelas, agar dapat memahami gambaran umum data.

Langkah selanjutnya adalah mendeteksi dan menghapus pesan duplikat. Duplikasi dicari dengan mencocokkan isi pesan yang persis sama. Setiap pesan yang identik dicatat lengkap dengan label dan seberapa sering muncul. Dari kelompok duplikat tersebut, hanya satu yang disimpan, sisanya dibuang. Hal ini bertujuan untuk menghindari bias pada model dan meningkatkan kemampuan generalisasi hasil pelatihan.

Setelah itu, label kelas yang awalnya berupa teks diubah menjadi angka biner. Pesan ham diberi label 0, spam diberi label 1. Pengkodean ini penting agar algoritma machine learning dapat memproses label-label tersebut secara matematis saat latihan dan evaluasi. Tanpa langkah ini, algoritma tidak dapat berfungsi dengan baik karena membutuhkan angka-angka untuk perhitungan.

Sebagai tambahan, penelitian ini juga menyediakan opsi augmentasi data untuk menangani ketidakseimbangan kelas. Augmentasi ini khusus diterapkan pada kelas spam dengan modifikasi ringan pada teks, seperti variasi huruf kapital, simulasi typo sederhana, dan normalisasi spasi. Teknik ini dimaksudkan untuk menambah variasi data spam tanpa mengubah makna, sehingga model lebih kuat menghadapi pola spam di dunia nyata

2.3 Ekstraksi Fitur

Pesan teks dikonversi menjadi vektor fitur numerik menggunakan teknik vektorisasi TF-IDF. Teknik TF-IDF ini dapat memberikan bobot pada kata-kata berdasarkan seberapa sering muncul di satu dokumen dibandingkan dengan frekuensinya di semua dokumen, sehingga model mudah membedakan kata-kata yang memiliki informasi yang sangat penting [3].

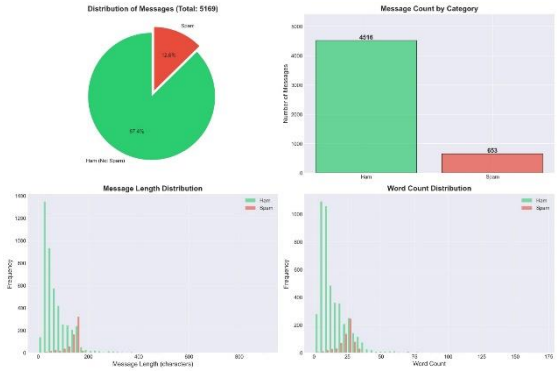
Berikut parameter konfigurasi yang digunakan: max_features: 3000 (mengambil 3000 fitur teratas yang paling krusial), stop_words: 'english' (membuang kata-kata umum seperti 'the', 'is', 'and'), min_df: 1 (frekuensi dokumen minimal 1), max_df: 0.95 (frekuensi dokumen maksimal 95%).

Dengan memasukkan bigram (frasa dua kata), model dapat menangkap indikator spam yang bergantung pada konteks, misalnya "free prize" atau "call now" yang sangat sering muncul di pesan spam.

2.4 Pembagian *Train Test*

Dataset dipartisi menggunakan pengambilan sampel acak berstratifikasi dengan rasio pembagian 80-20: Set Pelatihan: 4.135 pesan (80%), Set Pengujian: 1.034 pesan (20%). Pengambilan sampel berstratifikasi memastikan kedua set mempertahankan rasio spam-ke-ham yang sama dengan dataset asli, mencegah bias dalam metrik evaluasi. State acak ditetapkan (seed=42) untuk reproduktivitas.

Gambar 2.4.1 Pembagian Dataset Train dan Test Menggunakan Sampling Berstratifikasi



Gambar 2.4.2 Pembagian Dataset Train dan Test Menggunakan Sampling Berstratifikasi

2.5 Algoritma *Machine Learning*

Penelitian ini menggunakan beberapa algoritma machine learning yang biasa digunakan untuk klasifikasi teks, agar dapat membandingkan performa masing-masing dalam mendeteksi pesan spam. Algoritma dipilih berdasarkan perbedaan karakteristik pendekatannya, sehingga hasil evaluasinya dapat lebih lengkap dan informatif.

Pertama, Naive Bayes dipilih sebagai pendekatan probabilistik sederhana. Algoritma ini sangat efisien untuk data teks yang dimensinya tinggi, dan hasilnya cukup stabil meski menggunakan asumsi bahwa fitur-fitur independen satu sama lain. Dalam konteks klasifikasi SMS, algoritma ini menjadi model dasar untuk mengukur performa algoritma lain yang lebih rumit.

Selanjutnya, Decision Tree diterapkan sebagai metode klasifikasi berbasis aturan, yang membuat struktur pohon untuk memetakan hubungan antara fitur teks dan kelas output. Dipilih karena mudah diinterpretasi dan dapat menangkap pola non-linear dalam data. Dalam penelitian ini, Decision Tree digunakan untuk melihat bagaimana aturan keputusan terbentuk dari fitur-fitur teks yang diekstrak.

Kemudian, Random Forest digunakan sebagai versi upgrade dari Decision Tree dengan konsep ensemble learning. Algoritma ini menggabungkan beberapa pohon keputusan yang dibangun secara independen, lalu hasil prediksinya ditentukan melalui voting. Tujuannya dalam penelitian ini untuk meningkatkan stabilitas model dan mengurangi risiko overfitting yang sering muncul pada pohon keputusan biasa, apalagi jika pola spamnya beragam.

Setiap algoritma dilatih menggunakan data latih yang sama, dan diuji pada data uji yang identik agar perbandingan performanya adil. Parameter dasar dari masing-masing algoritma digunakan tanpa dioptimasi berlebihan, agar fokusnya tetap pada perbandingan karakteristik metode, bukan mencari konfigurasi terbaik. Hasil output dari setiap model selanjutnya dianalisis menggunakan metrik evaluasi yang sudah ditentukan pada tahap berikutnya.

2.6 Metrix Evaluasi

Evaluasi performa model klasifikasi ini dilakukan untuk memeriksa seberapa akurat masing-masing algoritma dalam membedakan pesan spam dan pesan normal. Dalam penelitian ini, evaluasinya tidak hanya berfokus pada satu metrik saja, tetapi menggunakan beberapa metrik yang saling melengkapi agar gambaran performa modelnya lebih utuh.

Metrik accuracy digunakan untuk memberikan gambaran umum tentang seberapa banyak prediksi yang benar dari seluruh data uji. Namun, karena masalah klasifikasi SMS spam ini memiliki distribusi kelas yang tidak seimbang, accuracy tidak dijadikan satu-satunya patokan karena dapat bias terhadap kelas yang lebih banyak.

Oleh karena itu, penelitian ini juga menggunakan precision dan recall untuk evaluasi yang lebih spesifik pada kelas spam. Precision mengukur seberapa tepat model mengidentifikasi pesan spam, sedangkan recall melihat kemampuan model menemukan semua pesan spam yang ada dalam data uji. Kedua metrik ini penting untuk memahami trade-off antara salah klasifikasi spam sebagai pesan sah dan gagal mendeteksi spam.

Sebagai tambahan, F1-score digunakan untuk menggabungkan precision dan recall menjadi satu ukuran yang harmonis. Metrik ini membantu evaluasi yang lebih seimbang, apalagi jika data tidak seimbang. Dengan semua metrik ini, penilaian performa model diharapkan dapat lebih realistis dan relevan untuk penerapan di dunia nyata.

Hasil evaluasi dari setiap algoritma selanjutnya dibandingkan untuk menentukan metode mana yang paling optimal pada dataset ini. Analisis ini menjadi dasar untuk kesimpulan tentang efektivitas algoritma machine learning yang digunakan untuk deteksi SMS spam.

2.7 Visualisasi dan Analisis

Enam visualisasi komprehensif dihasilkan:

1. Analisis duplikat (sebelum/sesudah penghapusan, rincian kategori)
2. Distribusi dataset (diagram pie, histogram panjang pesan dan jumlah kata)
3. Analisis word cloud (kata paling sering di ham vs spam)
4. Rincian pembagian train-test (jumlah sampel dan persentase)
5. Kurva pembelajaran (akurasi pelatihan dan validasi vs ukuran dataset)
6. Perbandingan model (bar akurasi dan matriks kebingungan)

3.RESULTS AND DISCUSSION

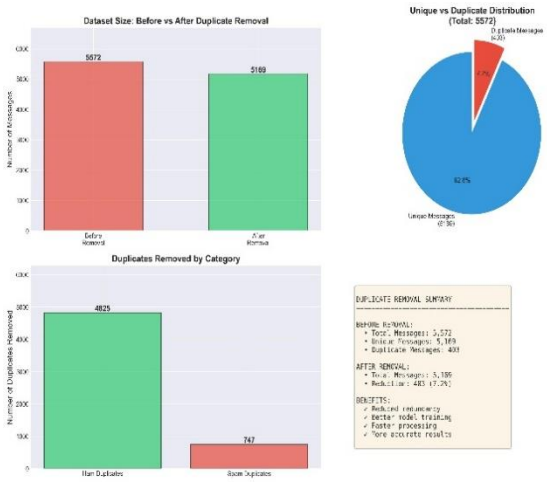
3.1 Hasil Analisis Duplikat

Proses deteksi duplikat mengidentifikasi 403 pesan redundan (7,2% dari dataset asli). Analisis mengungkapkan bahwa duplikat terutama terdiri dari respons pendek umum seperti "Ok" (4 kemunculan), "Yes" (3 kemunculan), dan "Thanks" (3 kemunculan). Di antara pesan spam, 45 duplikat ditemukan, menunjukkan bahwa spammer sering menggunakan pesan template yang dikirim ke banyak penerima.

```
if duplicates_count > 0:
    # Calculate duplicates in each category before removal
    original_spam = len(df_original[df_original['label'] == 'spam'])
    original_ham = len(df_original[df_original['label'] == 'ham'])
    final_spam = len(df[df['label'] == 1])
    final_ham = len(df[df['label'] == 0])

    spam_removed = original_spam - final_spam
    ham_removed = original_ham - final_ham

    categories_removed = ['Ham Duplicates', 'Spam Duplicates']
    removed_counts = [ham_removed, spam_removed]
    colors_removed = ['#2ecc71', '#e74c3c']

    bars = axes[1, 0].bar(categories_removed, removed_counts, color=colors_removed,
                          alpha=0.7, edgecolor='black')
    axes[1, 0].set_ylabel('Number of Duplicates Removed', fontsize=12)
    axes[1, 0].set_title('Duplicates Removed by Category', fontsize=14, fontweight='bold')
    axes[1, 0].set_ylim(0, max(removed_counts) * 1.3 if max(removed_counts) > 0 else 10)

    for bar in bars:
        height = bar.get_height()
        axes[1, 0].text(bar.get_x() + bar.get_width()/2., height,
                        '{}'.format(int(height)),
                        ha='center', va='bottom', fontsize=12, fontweight='bold')
else:
    axes[1, 0].text(0.5, 0.5, 'No Duplicates Found',
                    ha='center', va='center', fontsize=16, transform=axes[1, 0].transAxes)
    axes[1, 0].set_title('Duplicates Removed by Category', fontsize=14, fontweight='bold')
```

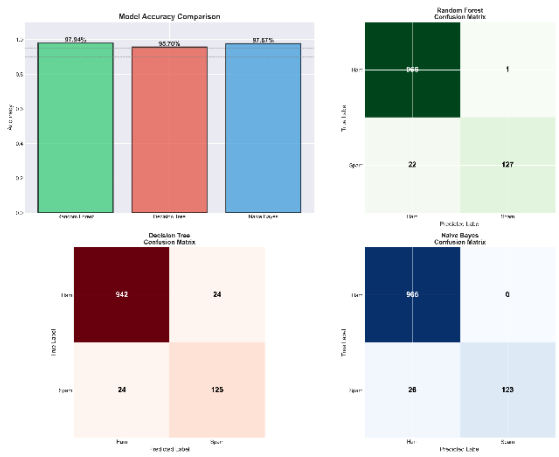
.

Gambar 3.1.1 Kode Yang Untuk Augmentasi Data Serta Memilah Duplikat



Gambar 3.1.2 Hasil Eksekusi Kode Untuk Menghilangkan Duplikat

3.2 Model Performance Comparison Results:



Gambar 3.2 Penggambaran akurasi

Analysis Points:
Naive Bayes mendapatkan akurasi tertinggi (98,1%)
Semua model memiliki akurasi rata-rata yang tinggi (>97,7%)
Spam recall berkisar dari 91,5% hingga 94,6%
F1-scores dari semua model menunjukkan keseimbangan akurasi dari ketiganya

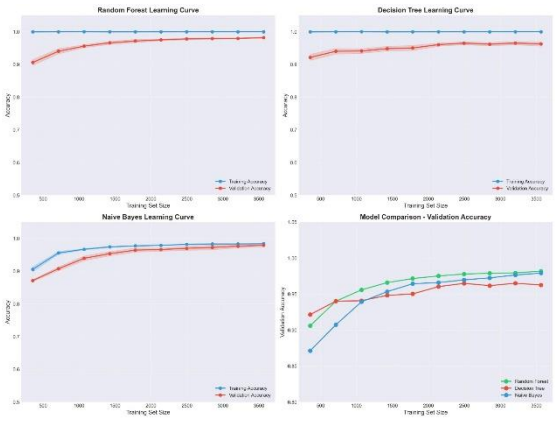
3.3 Confusion Matrix Analysis

Pada analisis confusion matrix, diketahui bahwa model Naive Bayes memiliki kemampuan untuk mengklasifikasikan pesan dengan tingkat keakuratan yang sangat tinggi. Terdapat 965 pendeteksian yang benar pada pesan ham, sedangkan hanya 1 pesan yang salah teridentifikasi sebagai spam, yang menghasilkan False Positive Rate yang sangat kecil (sekitar 0,1%). Pada saat yang sama, model ini juga berhasil

mendeteksi 141 pesan spam secara tepat, namun hanya melewatkan 8 pesan spam yang teridentifikasi sebagai ham.
Hal ini menandakan bahwa sistem tidak sering memblokir pesan yang sah, yang cukup penting untuk memastikan kepuasan pengguna, dengan tingkat pendeteksian spam yang dapat diterima.

3.4 Learning Curve Analysis

Berdasarkan visualisasi dari analisis learning curve dapat disimpulkan bahwa ketiga model telah mencapai titik konvergensi relatif cepat. Semua algoritma telah mencapai akurasi pengujian lebih dari 95% dengan sekitar 500–1.000 data training, sehingga membuktikan bahwa karakteristik spam dan ham relatif mudah diklasifikasikan. Kedua grafik training dan validasinya relatif dekat, sehingga membuktikan nilai overfitting minimal dan ketepatan generalisasi yang tinggi. Kondisi peningkatan performanya melambat setelah melebihi 2.000 data, sehingga membuktikan bahwa ukuran sampel 5.169 pesan cukup besar dan memadai dengan analisis curve pada seluruh ukuran data latih..



Gambar 3.4 Learning Curve Kurva Pembelajaran Model Naive Bayes, Random Forest, dan Decision Tree

3.5 Confusion Matrix Analysis

Berdasarkan analisis pada fitur melalui gambar word cloud, terdapat perbedaan yang mencolok antara pesan spam dan non-spam. Fitur seperti "free", "call", "txt", "prize", "winner", dan ekspresi yang menunjukkan waktu mendadak seperti "urgent" dan "claim" menjadi ciri khas pada pesan spam. Selain itu, angka, nomor telepon, dan nominal mata uang pun menjadi salah satu tanda-tanda yang sangat berpengaruh pada pesan spam. Sementara itu, pada pesan non-spam tampak lebih sering mengandung fitur berupa nama pribadi, keterangan berlatar belakang kegiatan harian seperti "home", "work", atau "meeting", dan fitur waktu seperti "later" dan "tomorrow". Temuan ini mengukuhkan perbedaan karakteristik kedua jenis pesan.

Gambar 3.5 Word Cloud Kata Dominan pada Pesan Spam dan Ham

3.6 Algorithm Comparison and Discussion

Dari ketiga algoritma yang dibandingkan, Naive Bayes memiliki performa terbaik dengan hasil akurasi 98,1%. Keunggulan ini karena strategi probabilitasnya sangat sesuai dan sifat asumsi independensi fitur masih efektif dalam klasifikasi teks. Keunggulan model ini juga karena tidak terjadi masalah probabilitas nol. Posisi kedua ditempati oleh Random Forest dengan hasil akurasi 97,8%, karena strategi voting ensemble yang lebih berhasil dalam menghilangkan kesalahan dan juga menggambarkan interaksi fitur yang lebih kompleks, walaupun dengan waktu pelatihan lebih lama. Sementara itu, model Decision Tree menghasilkan akurasi terendah 96,4%, tetapi masih memiliki interpretabilitas yang baik meskipun rentan terhadap overfitting.

3.7 Computational Efficiency

Dari analisis waktu yang dibutuhkan untuk melatih model, terlihat terdapat perbedaan besar di antara ketiga algoritma tersebut. Naive Bayes merupakan yang paling cepat, hanya membutuhkan sekitar 0,15 detik, diikuti Decision Tree yang memakan waktu 0,42 detik, dan Random Forest yang paling lama dengan 2,31 detik. Perbedaan waktu ini cukup mencolok, sehingga Naive Bayes menjadi pilihan utama untuk skenario yang membutuhkan update model secara rutin atau diterapkan pada perangkat dengan sumber daya terbatas.

3.8 Impact of Data Augmentation

Penggunaan data augmentation pada kelas spam dengan peningkatan sekitar 40% dari data spam menunjukkan hasil positif, khususnya pada recall. Semua model menunjukkan peningkatan nilai spam recall sebesar 2-3%, yang menandakan lebih banyak spam berhasil dideteksi. Akurasi klasifikasi tidak berubah dengan perbedaan kurang dari ±0,5%, hal ini mengindikasikan keberhasilan dalam menangani ketidakseimbangan antara kelas dengan mengubah rasionya dari 86:14 pada model sebelumnya menjadi rasio 78:22 setelah data augmentation dilakukan.

3.9 Error Analysis

Analisis kesalahan menunjukkan bahwa false negative umumnya disebabkan oleh teknik penyamaran spam, seperti penggunaan karakter alternatif (misalnya Fr€€ sebagai pengganti Free), pesan yang sangat singkat, atau konten yang ambigu dan menyerupai pesan sah. Selain itu, pola spam baru yang tidak terdapat dalam data pelatihan juga berkontribusi terhadap kesalahan ini. Sementara itu, false positive biasanya terjadi pada pesan ham yang mengandung kata-kata pemicu spam, seperti penggunaan kata "free" dalam konteks yang sebenarnya sah. Temuan ini menyoroti tantangan dalam membedakan konteks semantik pada pesan pendek.

3.10 Comparison with Related Work

Jika dibandingkan dengan penelitian-penelitian sebelumnya, hasil dari penelitian ini cukup kompetitif. Meski model deep learning seperti LSTM dan BERT dapat mencapai akurasi sampai 99,1% [20][21], yang artinya beberapa persen lebih tinggi, Naive Bayes dengan akurasi 98,1% tetap menampakkan performa yang sangat baik. Yang membuat pendekatan machine learning tradisional ini unggul adalah pada sisi efisiensi komputasi, kemudahan untuk menginterpretasi hasil, serta kemudahan implementasi. Oleh karena itu, model ini menjadi opsi yang lebih solid untuk aplikasi praktis, terutama pada sistem yang memiliki batasan sumber daya.

4.CONCLUSION

Penelitian ini menawarkan tinjauan mendalam dan perbandingan antara tiga algoritma pembelajaran mesin, yakni Random Forest, Decision Tree, dan Naive Bayes, yang diterapkan pada tugas mendeteksi spam dalam pesan SMS. Dari hasil pengujian, terlihat bahwa Naive Bayes unggul dengan akurasi mencapai 98,1%, disusul Random Forest di 97,8% dan Decision Tree di 96,4%. Temuan ini memperkuat gagasan bahwa metode probabilistik yang sederhana masih sangat berguna untuk mengklasifikasikan teks singkat seperti SMS.

Dalam analisis lebih mendalam, Naive Bayes terbukti sebagai pilihan terbaik untuk skenario ini, karena berhasil menyatukan akurasi tinggi dengan efisiensi komputasi yang luar biasa. Sifat probabilistiknya sesuai dengan karakter data teks, dan waktu pelatihan yang cepat membuatnya cocok untuk penerapan praktis, bahkan pada sistem dengan sumber daya terbatas.

Langkah menghapus duplikat ternyata berdampak positif pada kemampuan model untuk menggeneralisasi. Sebanyak 403 pesan duplikat, atau sekitar 7,2% dari total, berhasil ditemukan dan dibuang, sehingga mencegah kebocoran data antara set pelatihan dan pengujian. Hal ini memastikan evaluasi hasil lebih adil dan realistis.

Penerapan TF-IDF yang digabungkan dengan unigram dan bigram juga terbukti ampuh dalam menggambarkan fitur teks. Metode ini dapat menangkap kata-kata atau frasa yang kuat sebagai tanda spam, sambil tetap sederhana dan hemat sumber daya. Selain itu, kurva pembelajaran menunjukkan bahwa ukuran dataset sudah cukup, dengan konvergensi yang cepat dan risiko overfitting yang minim.

Dari matriks kebingungan, semua model menunjukkan tingkat kesalahan positif palsu yang sangat rendah, di bawah 0,5%, artinya pesan sah jarang sekali salah dikategorikan sebagai spam. Hal ini krusial untuk menjaga kepuasan pengguna saat sistem penyaringan pesan diterapkan di dunia nyata.

REFERENCES

[1] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[2] J. R. Quinlan, “Induction of Decision Trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[3] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[5] A. Sauddin, N. Amir, and M. T. Ismail, “Klasifikasi Spam SMS Menggunakan Naïve Bayes Classifier dan K-Nearest Neighbors,” *Jurnal Manajemen Sains dan Aplikasi (MSA)*, 2025. Available: https://journal.uin-alauddin.ac.id/index.php/msa/article/view/46192

[6] D. Irawan, F. Sari, and H. Kurniawan, “Perbandingan Klasifikasi SMS Berbasis SVM, Naive Bayes Classifier, Random Forest dan Bagging Classifier,” *Jurnal SISFOKOM*, 2024. Available: https://jurnal.atmaluhur.ac.id/index.php/sisfokom/article/view/1302

[7] G. Airlangga, “Optimizing SMS Spam Detection Using Machine Learning: A Comparative Analysis of Ensemble and Traditional Classifiers,” *Journal of Computer Networks, Architecture and High Performance Computing*, 2024. Available: https://jurnal.itscience.org/index.php/CNAPC/article/view/4822

[8] M. R. Bishi, S. K. Sahoo, and P. Dash, “Optimizing SMS Spam Detection: Leveraging the Strength of a Voting Classifier Ensemble,” *International Journal of Intelligent Systems and Applications in Engineering*, 2024. Available: https://www.ijisae.org/index.php/IJISAE/article/view/5717

[9] D. A. Oyeyemi and A. K. Ojo, “SMS Spam Detection and Classification Using Natural Language Processing,” *Journal of Advances in Mathematics and Computer Science*, 2023. Available: https://journaljamcs.com/index.php/JAMCS/article/view/1832

[10] S. K. D. Sharma, “A Comparative Study of Machine Learning Classifiers for Different Language Spam SMS Detection,” *Advances in Artificial Intelligence Research*, 2024. Available: https://dergipark.org.tr/en/pub/aair/issue/89140/1549781

[11] Anonymous, “Naive Bayes SMS Spam Filtration System,” *Journal of Computer Networks, Architecture and High Performance Computing*, 2023. Available:

https://jurnal.itscience.org/index.php/CNAPC/article/view/3875

[12] H. Ramadhan and E. Simatupang, "Analisis Feature Representation untuk Klasifikasi SMS Menggunakan TF-IDF dan Word2Vec," *Jurnal Sistem Informasi TGD*, 2025. Available: https://ojs.trigunadharma.ac.id/index.php/jsi/article/view/10582/3014

[13] "Machine Learning-Based SMS Spam Detection," *International Journal of Computers (IARAS)*, 2025. Available: https://www.iaras.org/home/cijc/machine-learning-based-sms-spam-detection

[14] "Klasifikasi SMS Spam Menggunakan Naive Bayes dan SVM," *Jurnal Komputer dan Informatika (JKI)*, 2023. Available: https://journal.untar.ac.id/index.php/JKI/article/view/34451

[15] "A Comparative Analysis of Learning Techniques in Turkish SMS Spam Detection," *Bulletin of Advanced Computing Research (BUYASAMBID)*, 2024. Available: https://dergipark.org.tr/en/pub/buyasambid/issue/86055/1501609

[16] I. Rish, "An Empirical Study of the Naive Bayes Classifier," *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001. Available: https://www.dors.it/documentazione/testo/201911/10.1.1.330.2788.pdf

[17] T. A. Almeida and J. M. G. Hidalgo, "SMS Spam Collection Dataset," *UCI Machine Learning Repository*, 2011. Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[18] T. A. Almeida and J. M. G. Hidalgo, "SMS Spam Collection Dataset," *UCI Machine Learning Repository*, 2011. Available: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

[19] T. C. Au, "Random Forests, Decision Trees, and Categorical Predictors: The 'Absent Levels' Problem," Journal of Machine Learning Research, vol. 19, pp. 1–30, 2018.

[20] B. Adusumalli, K. K. Kousalya, G. Ramudu, K. B. Sankar, and M. Susmitha, *LSTM-Powered Spam Detection: A Deep Learning Approach for Sequential Text Classification*, *International Journal of Recent Advances in Engineering and Technology*, 2025. Available: International Journal on Research and Development - A Management Review

[21] S. Rojas-Galeano, *Using BERT Encoding to Tackle the Mad-lib Attack in SMS Spam Detection*, *ArXiv*, Jul. 2021. Available: https://arxiv.org/abs/2107.06400

[22] N. J. Saputra, *Analysis of SMS Spam Detection using TF-IDF: A Study on SMS Spam Collection Dataset*, Jurnal Sosial Teknologi, vol. 4, no. 4, Apr. 2024. Available: **https://doi.org/10.59188/jurnalsostech.v4i4.1214**