

KOMPARASI KINERJA SUPPORT VECTOR MACHINE DAN RANDOM FOREST UNTUK DETEKSI SPAM EMAIL BERBASIS TF-IDF TEKS

Arum Adila Ningtias¹, Nizirwan Anwar², Arief Ichwani³, Raden Teddy Iswahyudi⁴

^{1,2,3,4} Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul

arumadila@student.esaunggul.ac.id, nizirwan.anwar@esaunggul.ac.id, arief.ichwani@esaunggul.ac.id,
raden.teddy@esaunggul.ac.id

ABSTRAK

Spam email merupakan salah satu permasalahan dalam komunikasi digital karena dapat mengganggu produktivitas, membebani sistem, serta berpotensi membawa ancaman keamanan seperti phishing, malware, dan penipuan digital. Penelitian ini bertujuan membangun dan membandingkan performa algoritma *Support Vector Machine* (SVM) dan *Random Forest* dalam mendeteksi *spam* email berbasis klasifikasi teks. Dataset yang digunakan adalah *SMS Spam Collection* yang terdiri dari 5.572 pesan dengan 2 (dua) kelas, yaitu *ham/non-spam* dan *spam*. Tahapan penelitian dilakukan secara sekuensial melalui proses *text*, *cleaning*, *case folding*, *tokenization*, *stopword removal*, ekstraksi fitur menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF), *train-test split*, pelatihan model SVM dan *Random Forest*, serta evaluasi performa. Evaluasi dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-score*, *ROC-AUC*, dan *Confusion matrix*, sebagaimana dijelaskan dalam rancangan penelitian dan evaluasi model pada dokumen. Hasil pengujian menunjukkan bahwa *Linear SVM + GridSearchCV* memperoleh performa lebih baik dibandingkan *Random Forest + GridSearchCV*, dengan *Accuracy* sebesar 0,983857, *Precision* 0,978102, *Recall* 0,899329, *F1-score* 0,937063, dan *ROC-AUC* 0,987039. *Random Forest* memperoleh *Accuracy* 0,959641, *Precision* 0,882353, *Recall* 0,805369, *F1-score* 0,842105, dan *ROC-AUC* 0,960280. Berdasarkan hasil tersebut, SVM lebih efektif dalam mendeteksi *spam* email karena mampu menghasilkan kesalahan klasifikasi yang lebih rendah dan performa evaluasi yang lebih seimbang.

Kata kunci: *Spam Email, Support Vector Machine, Random Forest, TF-IDF, Klasifikasi Teks.*

ABSTRACT

Email spam is one of the major problems in digital communication because it can reduce user productivity, increase system load, and potentially introduce security threats such as phishing, malware, and digital fraud. This study aims to build and compare the performance of Support Vector Machine (SVM) and Random Forest algorithms for spam email detection based on text classification. The dataset used in this study is the SMS Spam Collection, consisting of 5,572 messages divided into two classes: ham/non-spam and spam. The research stages were conducted sequentially through text processing, cleaning, case folding, tokenization, stopwords removal, feature extraction using Term Frequency-Inverse Document Frequency (TF-IDF), train-test split, SVM and Random Forest model training, and performance evaluation. The models were evaluated using Accuracy, Precision, Recall, F1-score, ROC-AUC, and Confusion matrix. The experimental results show that Linear SVM with GridSearchCV achieved better performance than Random Forest with GridSearchCV, with an Accuracy of 0.983857, Precision of 0.978102, Recall of 0.899329, F1-score of 0.937063, and ROC-AUC of 0.987039. Meanwhile, Random Forest achieved an Accuracy of 0.959641, Precision of 0.882353, Recall of 0.805369, F1-score of 0.842105, and ROC-AUC of 0.960280. Based on these results, SVM is more effective for spam email detection because it produces lower classification errors and more balanced evaluation performance.

Keywords: *Spam Email, Support Vector Machine, Random Forest, TF-IDF, Text Classification.*

1. Pendahuluan

Perkembangan komunikasi digital menjadikan email dan pesan elektronik sebagai media utama dalam pertukaran informasi. Namun, peningkatan penggunaan media komunikasi digital juga diikuti oleh meningkatnya jumlah pesan tidak diinginkan atau *spam*. *Spam* email dapat mengganggu produktivitas pengguna, membebani sistem, serta membawa risiko keamanan seperti *phishing*, *malware*, tautan berbahaya, dan penipuan digital. Oleh karena itu, sistem deteksi *spam* menjadi komponen penting dalam menjaga keamanan dan kenyamanan komunikasi digital [2], [6], [10], [12].

Deteksi *spam* email dapat dipandang sebagai permasalahan klasifikasi teks, yaitu proses mengelompokkan pesan ke dalam kelas tertentu berdasarkan isi teksnya. Dalam penelitian ini, kelas yang digunakan adalah *spam* dan *ham/non-spam*. Karena teks email atau pesan bersifat tidak terstruktur, maka diperlukan proses pengolahan teks agar data dapat dipahami oleh algoritma *machine learning*. Tahapan umum yang dilakukan meliputi *cleaning*, *case folding*, *tokenization*, *stopword removal*, dan ekstraksi fitur [4], [7], [36], [54]. Salah satu metode ekstraksi fitur yang banyak digunakan dalam klasifikasi teks adalah *Term Frequency-Inverse Document Frequency* (TF-IDF).

TF-IDF mengubah teks menjadi vektor numerik dengan memberi bobot pada kata berdasarkan frekuensi kemunculannya dalam dokumen dan tingkat kelangkaannya dalam keseluruhan korpus [7], [31], [53]. Representasi numerik ini kemudian digunakan sebagai masukan bagi algoritma supervised learning.

Penelitian ini berfokus pada komparasi 2 (dua) algoritma *machine learning*, yaitu SVM dan Random Forest. SVM dikenal efektif untuk klasifikasi teks berdimensi tinggi karena mampu membentuk *hyperplane* pemisah antar kelas [8], [13], [28], [29]. Sementara itu, Random Forest merupakan algoritma *ensemble* berbasis *decision tree* yang menggabungkan banyak *decision tree* untuk menghasilkan prediksi yang stabil [30], [42], [51].

Berdasarkan penelitian ini menggunakan dataset SMS Spam Collection yang terdiri atas 5.572 pesan dengan 2 (dua) kelas, yaitu *ham/non-spam* dan *spam*. Evaluasi dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, *F1-score*, ROC-AUC, dan *Confusion matrix*. Hasil pengujian menunjukkan bahwa Linear SVM + *GridSearchCV* menghasilkan performa lebih baik dibandingkan Random Forest + *GridSearchCV*, dengan *Accuracy* sebesar 0,983857, *Precision* 0,978102, *Recall* 0,899329, *F1-score* 0,937063, dan ROC-AUC 0,987039.

Dengan tujuan penelitian ini adalah merancang model deteksi *spam* email berbasis TF-IDF, membandingkan kinerja algoritma SVM dan Random Forest, serta menentukan model yang lebih efektif berdasarkan metrik evaluasi klasifikasi.

2. Kajian Literatur

2.1 Spam Email dan Spam Filtering

Spam email merupakan pesan elektronik yang dikirim tanpa permintaan penerima dan umumnya berisi promosi, penipuan, tautan mencurigakan, phishing, atau konten berbahaya. *Spam filtering* menjadi salah satu bidang penting dalam keamanan informasi karena bertujuan menyaring pesan yang tidak relevan atau berpotensi merugikan pengguna [6], [10], [12]. Penelitian Sahami et al. menjadi salah satu dasar awal penggunaan *machine learning* untuk *junk email filtering* dengan pendekatan probabilistik [2].

Blanzieri dan Bryl menjelaskan bahwa teknik *machine learning* banyak digunakan dalam *spam filtering* karena mampu mengenali pola pesan berdasarkan data historis [6]. Guzella dan Caminhas juga menegaskan bahwa pendekatan *machine learning* dalam *spam filtering* dapat menggunakan berbagai algoritma, mulai dari Naive Bayes, SVM, *decision tree*, hingga *ensemble learning* [10]. Cormack menyatakan bahwa evaluasi *spam filtering* perlu memperhatikan tidak hanya akurasi,

tetapi juga dampak kesalahan klasifikasi terhadap pengguna [12].

2.2 Klasifikasi Teks dan Text Mining

Klasifikasi teks merupakan proses mengelompokkan dokumen teks ke dalam kategori tertentu berdasarkan fitur yang terkandung di dalamnya. Dalam konteks deteksi *spam*, setiap pesan diperlakukan sebagai dokumen teks yang akan diklasifikasikan menjadi *spam* atau *non-spam*. Sebastiani menjelaskan bahwa *machine learning* dalam text categorization sangat bergantung pada representasi dokumen, pemilihan fitur, dan algoritma klasifikasi [4].

Text mining berperan mengubah teks tidak terstruktur menjadi data yang dapat dianalisis secara komputasional. Manning et al. menjelaskan bahwa proses information retrieval dan text mining membutuhkan representasi numerik seperti term frequency dan pembobotan kata [7]. Salton dan McGill juga menekankan pentingnya representasi dokumen dalam sistem temu kembali informasi [31]. Aggarwal dan Zhai memperluas konsep ini dalam mining text data, termasuk preprocessing, *feature extraction*, dan klasifikasi teks [54].

2.3 Preprocessing dan Ekstraksi Fitur

Preprocessing teks merupakan tahapan awal untuk membersihkan data dari noise. Tahapan ini dapat mencakup *cleaning*, *case folding*, *tokenization*, *stopword removal*, dan *stemming*. Bird et al. menjelaskan bahwa pengolahan bahasa alami membutuhkan proses tokenisasi dan normalisasi agar teks dapat diproses secara sistematis [36]. Porter memperkenalkan algoritma *stemming* yang dapat digunakan untuk mengubah kata berimbuhan menjadi bentuk dasar [50].

Setelah preprocessing, teks diubah menjadi fitur numerik. TF-IDF merupakan salah satu metode populer karena mampu memberi bobot pada kata berdasarkan tingkat kepentingannya dalam dokumen dan korpus [7], [31], [53]. Selain TF-IDF, BM25 juga dikenal sebagai skema pembobotan dalam information retrieval yang mempertimbangkan relevansi kata terhadap dokumen [11]. *Feature selection* juga penting karena data teks biasanya memiliki dimensi fitur yang sangat tinggi. Forman serta Yang dan Pedersen menunjukkan bahwa seleksi fitur dapat meningkatkan efisiensi dan performa klasifikasi teks [16], [39].

2.4 Support Vector Machine (SVM)

SVM merupakan algoritma supervised learning yang bekerja dengan mencari *hyperplane* terbaik untuk memisahkan kelas data. Cortes dan Vapnik

memperkenalkan konsep support-vector networks sebagai dasar algoritma SVM [29], sedangkan Vapnik mengembangkan landasan statistical learning theory yang menjadi fondasi teoritis SVM [56]. Burges menjelaskan bahwa SVM efektif digunakan untuk pattern recognition karena mampu memaksimalkan margin antar kelas [8].

Dalam klasifikasi teks, Joachims menunjukkan bahwa SVM cocok untuk data berdimensi tinggi dengan banyak fitur relevan [28]. Drucker et al. juga menunjukkan bahwa SVM efektif digunakan untuk *spam categorization* [13]. Sculley dan Wachman menambahkan bahwa SVM dapat diterapkan dalam *spam filtering* secara online, sehingga relevan untuk lingkungan pesan digital yang terus berubah [41].

2.5 Random Forest

Random Forest merupakan algoritma *ensemble learning* yang membangun banyak *decision tree* dan menggabungkan hasil prediksinya untuk menghasilkan keputusan akhir. Breiman memperkenalkan Random Forest sebagai metode yang mampu meningkatkan stabilitas dan akurasi model dengan mengurangi risiko overfitting pada *decision tree* tunggal [30]. Ho sebelumnya memperkenalkan konsep random *decision forests* sebagai pendekatan berbasis banyak *decision tree* [42]. Random Forest dapat digunakan untuk klasifikasi teks karena setiap fitur hasil ekstraksi, seperti bobot TF-IDF, dapat diperlakukan sebagai atribut masukan dalam *decision tree*. Liaw dan Wiener menjelaskan implementasi Random Forest untuk klasifikasi dan regresi [51]. Hastie et al. juga menempatkan Random Forest sebagai bagian dari metode *ensemble* yang kuat dalam pembelajaran statistik [55].

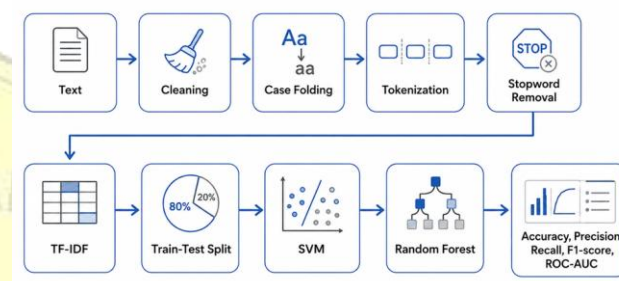
2.6 Evaluasi Model Klasifikasi

Evaluasi model klasifikasi dilakukan untuk mengetahui kemampuan model dalam memprediksi kelas secara benar. Powers menjelaskan metrik *Precision*, *Recall*, dan F-measure sebagai ukuran penting dalam evaluasi klasifikasi [18]. *Accuracy* dapat menunjukkan performa umum, tetapi tidak selalu cukup terutama pada dataset yang tidak seimbang. Fawcett menjelaskan ROC analysis sebagai pendekatan untuk mengukur kemampuan model dalam membedakan 2 (dua) kelas [37]. Pada dataset yang tidak seimbang, Saito dan Rehmsmeier menunjukkan bahwa *Precision-Recall* curve dapat lebih informatif dibandingkan ROC curve [38]. Oleh karena itu, penelitian deteksi *spam* perlu menggunakan beberapa metrik sekaligus, seperti *Accuracy*, *Precision*, *Recall*, F1-score, ROC-AUC, dan *Confusion matrix*, agar analisis performa model lebih objektif.

3. Metodologi dan Desain

3.1 Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen komputasional dengan metode supervised learning. Desain penelitian bersifat komparatif karena membandingkan performa 2 (dua) algoritma, yaitu SVM dan Random Forest, pada dataset dan skenario evaluasi yang sama. Alur penelitian disusun secara sekuensial dari data teks hingga pemilihan model terbaik. Alur penelitian adalah sebagai berikut:



Gambar 1. Alur Penelitian

3.2 Dataset Penelitian

Tabel 1. Komponen Dataset

Komponen	Keterangan
Jenis data	Teks pesan
Jumlah data	5.572 pesan
Kelas	Ham/non-spam dan spam
Metode ekstraksi fitur	TF-IDF
Model klasifikasi	SVM dan Random Forest
Evaluasi	Accuracy, Precision, Recall, F1-score, ROC-AUC, Confusion matrix

Dataset yang digunakan adalah SMS *Spam Collection* yang terdiri atas 5.572 pesan dengan 2 (dua) kelas, yaitu *ham/non-spam* dan *spam*. Dataset ini relevan untuk penelitian *spam filtering* karena menyediakan data teks mentah dan label kelas, sehingga mendukung proses preprocessing, ekstraksi fitur, dan klasifikasi supervised learning. Informasi dataset ini juga disebutkan secara eksplisit dalam abstrak terlampir. Dataset SMS *Spam Collection* sendiri dirujuk dalam daftar pustaka melalui penelitian Almeida et al. [46]. Struktur dataset dapat dijelaskan sebagai berikut:

3.3 Preprocessing Teks

Preprocessing dilakukan untuk mengubah teks mentah menjadi bentuk yang lebih *clean* dan konsisten. Tahapan preprocessing meliputi:

1. *Cleaning*, yaitu menghapus URL, angka, simbol, tanda baca, dan karakter tidak relevan.
2. *Case folding*, yaitu mengubah seluruh huruf menjadi huruf kecil.
3. *Tokenization*, yaitu memecah kalimat menjadi token kata.
4. *Stopword removal*, yaitu menghapus kata umum yang kurang bermakna.
5. Normalisasi teks, yaitu memastikan teks siap diproses pada tahap ekstraksi fitur.

Preprocessing diperlukan karena algoritma *machine learning* tidak dapat memproses teks mentah secara langsung. Hasil preprocessing menjadi input untuk proses TF-IDF [7], [36], [50].

3.4 Ekstraksi Fitur TF-IDF

Ekstraksi fitur dilakukan menggunakan *Term Frequency-Inverse Document Frequency* (TF-IDF). TF-IDF menghasilkan bobot numerik untuk setiap kata berdasarkan frekuensi kemunculan kata dalam dokumen dan tingkat kelangkaannya dalam keseluruhan korpus. Kata yang sering muncul dalam satu dokumen tetapi jarang muncul pada dokumen lain akan memiliki bobot lebih tinggi [7], [31], [53]. Secara konseptual, TF-IDF dapat dinyatakan sebagai:

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

dengan $TF(t, d)$ menunjukkan frekuensi term t dalam dokumen d , sedangkan $IDF(t)$ menunjukkan tingkat kelangkaan term dalam korpus.

3.5 Train-Test Split

Dataset dibagi menjadi data latih dan data uji. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk mengukur kemampuan model pada data baru. Pembagian data dilakukan dengan stratifikasi agar proporsi kelas *spam* dan *ham* tetap terjaga. Validasi parameter dilakukan menggunakan *GridSearchCV* dengan cross-validation. Kohavi menjelaskan bahwa cross-validation dapat digunakan untuk estimasi akurasi dan pemilihan model secara lebih stabil [40].

3.6 Model Support Vector Machine

Model SVM digunakan untuk membangun *hyperplane* pemisah antara kelas *spam* dan *non-spam*. Pada penelitian ini, SVM diuji menggunakan optimasi parameter melalui *GridSearchCV*. Parameter yang dapat diuji meliputi:

Parameter SVM	Fungsi
Kernel	Menentukan fungsi pemisah kelas
C parameter	Mengatur regularisasi model
Gamma	Mengatur pengaruh data latih pada kernel non-linear
Class weight	Menangani ketidakseimbangan kelas

SVM relevan untuk klasifikasi teks karena fitur TF-IDF biasanya menghasilkan data berdimensi tinggi dan *sparse* [13], [28].

3.7 Model Random Forest

Random Forest digunakan sebagai model pembandingan berbasis *ensemble learning*. Model ini membangun banyak *decision tree* dan menggabungkan hasil prediksi melalui *voting* [30], [51]. Parameter yang dapat diuji meliputi:

Tabel 3. Parameter Model Random Forest

Parameter Random Forest	Fungsi
Jumlah pohon	Menentukan banyaknya <i>decision tree</i>
Maksimum kedalaman pohon	Mengatur kompleksitas <i>tree</i>
Minimum <i>sample split</i>	Menentukan jumlah minimum sampel untuk membagi node
Minimum <i>sample leaf</i>	Menentukan jumlah minimum sampel pada daun
<i>Feature selection per tree</i>	Menentukan jumlah fitur yang dipertimbangkan tiap percabangan

3.8 Evaluasi Model

Evaluasi dilakukan menggunakan metrik:

Tabel 4. Parameter Evaluasi Metriks

Metrik	Fungsi
<i>Accuracy</i>	Mengukur proporsi prediksi benar
<i>Precision</i>	Mengukur ketepatan prediksi <i>spam</i>
<i>Recall</i>	Mengukur kemampuan model menemukan <i>spam</i>
F1-score	Mengukur keseimbangan <i>Precision</i> dan <i>Recall</i>
ROC-AUC	Mengukur kemampuan membedakan 2 (dua) kelas
<i>Confusion matrix</i>	Menampilkan TP, TN, FP, dan FN

Penggunaan multi-metrik diperlukan karena dataset *spam* sering memiliki distribusi kelas yang tidak seimbang [18], [37], [38].

4. Hasil dan Analisis Pembahasan

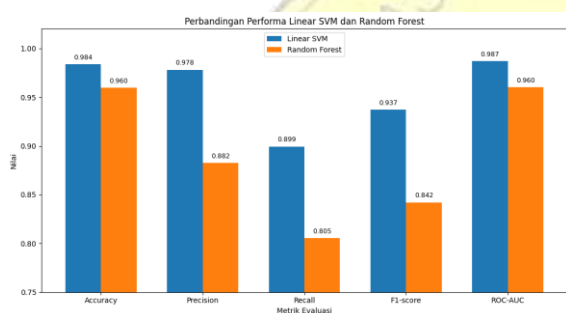
Tabel 2. Parameter Model SVM

4.1 Hasil Pengujian Model

Berdasarkan abstrak terlampir, hasil pengujian menunjukkan bahwa Linear SVM + *GridSearchCV* memperoleh performa lebih baik dibandingkan Random Forest + *GridSearchCV*. Nilai evaluasi model adalah sebagai berikut:

Tabel 5. Hasil Pengujian Model SVM dan Random Forest

Model	Accuracy	Precision	Recall	F1-score	ROC-AUC
Linear SVM	0,9838	0,9781	0,8993	0,9370	0,9870
<i>GridSearchCV</i>	57	02	29	63	39
Random Forest	0,9596	0,8823	0,8053	0,8421	0,9602
<i>GridSearchCV</i>	41	53	69	05	80



Gambar 2. Grafik Hasil Pengujian Model SVM dan Random Forest (tabel 5.)

4.2 Analisis Performa SVM

Linear SVM memperoleh nilai *Accuracy* sebesar 0,983857, *Precision* 0,978102, *Recall* 0,899329, *F1-score* 0,937063, dan ROC-AUC 0,987039. Nilai *Precision* yang tinggi menunjukkan bahwa sebagian besar pesan yang diprediksi sebagai *spam* benar-benar merupakan *spam*. Nilai *Recall* yang cukup tinggi menunjukkan bahwa SVM mampu mengenali sebagian besar pesan *spam* dalam data uji. Keunggulan SVM dapat dijelaskan melalui karakteristiknya yang sesuai untuk data teks berdimensi tinggi. TF-IDF menghasilkan *matrix* fitur dengan banyak kolom, sehingga SVM mampu membentuk *hyperplane* yang efektif untuk memisahkan *spam* dan *non-spam* [8], [13], [28]. Hasil ini juga sejalan dengan penelitian Drucker et al. yang menunjukkan efektivitas SVM dalam *spam categorization* [13].

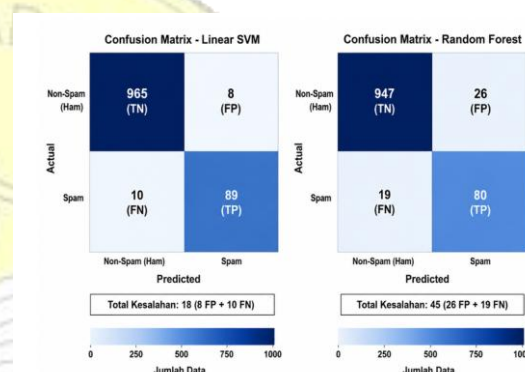
4.3 Analisis Performa Random Forest

Random Forest memperoleh *Accuracy* sebesar 0,959641, *Precision* 0,882353, *Recall* 0,805369, *F1-score* 0,842105, dan ROC-AUC 0,960280. Nilai ini menunjukkan bahwa Random Forest tetap

mampu melakukan klasifikasi *spam* dan *non-spam* dengan baik, tetapi performanya masih berada di bawah SVM. Random Forest memiliki keunggulan dalam stabilitas prediksi karena menggunakan banyak *decision tree* [30], [51]. Namun, pada data teks berbasis TF-IDF yang berdimensi tinggi, Random Forest dapat menghadapi tantangan karena banyaknya fitur *sparse*. Hal ini dapat menyebabkan model tidak seefisien SVM dalam menemukan pola pemisah antar kelas, terutama jika fitur tidak diseleksi secara optimal [16], [39].

4.4 Analisis Confusion Matrix SVM dan Random Forest

Berdasarkan hasil *run* sebelumnya, *Confusion matrix* model adalah sebagai berikut:



Gambar 3. Confusion Matrix SVM dan Random Forest

Tabel 6. Analisis Confusion Matrix SVM dan Random Forest

Model	TN	FP	FN	TP
Linear SVM	963	3	15	134
<i>GridSearchCV</i>				
Random Forest	950	16	29	120
<i>GridSearchCV</i>				

Pada SVM, nilai FP hanya 3, artinya sangat sedikit pesan *non-spam* yang salah diklasifikasikan sebagai *spam*. Nilai FN sebesar 15, artinya terdapat 15 pesan *spam* yang tidak terdeteksi. Pada Random Forest, FP sebesar 16 dan FN sebesar 29, lebih tinggi dibandingkan SVM. Hal ini menunjukkan bahwa SVM lebih baik dalam meminimalkan kesalahan klasifikasi. Dalam konteks deteksi *spam*, FP dapat menyebabkan pesan penting masuk ke *folder spam*, sedangkan FN dapat membuat pesan *spam* lolos ke kotak masuk. Oleh karena itu, model dengan nilai FP dan FN lebih rendah lebih layak dipilih.

4.5 Pembahasan Komparatif

Secara keseluruhan, Linear SVM + *GridSearchCV* unggul pada seluruh metrik evaluasi. Keunggulan ini menunjukkan bahwa kombinasi TF-IDF dan SVM efektif untuk klasifikasi teks *spam*. SVM mampu memanfaatkan fitur *sparse* berdimensi tinggi untuk membentuk batas keputusan yang kuat. Random Forest tetap memberikan performa baik, tetapi lebih sensitif terhadap tingginya dimensi fitur pada data teks. *F1-score* menjadi indikator penting dalam penelitian ini karena menggabungkan *Precision* dan *Recall*. Nilai *F1-score* SVM sebesar 0,937063 menunjukkan keseimbangan yang lebih baik dibandingkan Random Forest yang memperoleh 0,842105. Selain itu, ROC-AUC SVM sebesar 0,987039 menunjukkan kemampuan yang sangat baik dalam membedakan kelas *spam* dan *non-spam*.

Hasil penelitian menunjukkan bahwa SVM lebih efektif untuk deteksi *spam* email berbasis TF-IDF pada dataset SMS *Spam Collection*. Temuan ini sejalan dengan literatur yang menyatakan bahwa SVM unggul pada klasifikasi teks berdimensi tinggi [13], [28], sedangkan Random Forest lebih kuat pada data tabular atau fitur yang memiliki struktur percabangan lebih jelas [30], [51].

5. Kesimpulan

Penelitian ini merancang dan mem-komparasi model deteksi *spam* email menggunakan SVM dan Random Forest berbasis ekstraksi fitur TF-IDF. Dataset yang digunakan adalah SMS *Spam Collection* yang terdiri atas 5.572 pesan dengan 2 (dua) kelas, yaitu *ham/non-spam* dan *spam*. Tahapan penelitian meliputi *text processing*, *cleaning*, *case folding*, *tokenization*, *stopword removal*, ekstraksi fitur TF-IDF, *train-test split*, pelatihan model, optimasi parameter menggunakan *GridSearchCV*, dan evaluasi performa.

Hasil pengujian menunjukkan bahwa Linear SVM + *GridSearchCV* menghasilkan performa terbaik dengan *Accuracy* 0,983857, *Precision* 0,978102, *Recall* 0,899329, *F1-score* 0,937063, dan ROC-AUC 0,987039. Sementara itu, Random Forest + *GridSearchCV* menghasilkan *Accuracy* 0,959641, *Precision* 0,882353, *Recall* 0,805369, *F1-score* 0,842105, dan ROC-AUC 0,960280. Berdasarkan hasil tersebut, dapat disimpulkan bahwa SVM lebih efektif dibandingkan Random Forest dalam mendeteksi *spam* email berbasis TF-IDF. SVM mampu menghasilkan performa evaluasi yang lebih seimbang dan kesalahan klasifikasi yang lebih rendah. Dengan demikian, model Linear SVM dapat direkomendasikan sebagai model utama untuk deteksi *spam* email berbasis klasifikasi teks.

Daftar Pustaka

[1] D. Dua and C. Graff, "UCI Machine Learning Repository." University of

California, Irvine, School of Information and Computer Sciences, 2019.

- [2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *AAAI Workshop on Learning for Text Categorization*, 1998, pp. 55–62.
- [3] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *AAAI Workshop on Learning for Text Categorization*, 1998, pp. 41–48.
- [4] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, 2002.
- [5] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
- [6] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email *spam* filtering," *Artif. Intell. Rev.*, vol. 29, no. 1, pp. 63–92, 2008.
- [7] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008.
- [8] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.
- [9] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to *spam* filtering," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10206–10222, 2009.
- [10] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
- [11] G. V. Cormack, "Email *spam* filtering: A systematic review," *Found. Trends Inf. Retr.*, vol. 1, no. 4, pp. 335–455, 2008.
- [12] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for *spam* categorization," *IEEE Trans. Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
- [13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [14] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, 2003.
- [15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.
- [16] D. M. W. Powers, "Evaluation: From *Precision*, *Recall* and F-measure to ROC,

- informedness, markedness and correlation,” *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.
- [17] T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features,” in *Lecture Notes in Computer Science*, Springer Science+Business Media, 1998, pp. 137–142.
- [18] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [19] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [20] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill, 1983.
- [21] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, 2011.
- [22] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA: O’Reilly Media, 2009.
- [23] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
- [24] T. Saito and M. Rehmsmeier, “The Precision-Recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PLoS One*, vol. 10, no. 3, 2015.
- [25] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 412–420.
- [26] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1137–1143.
- [27] D. Sculley and G. M. Wachman, “Relaxed online SVMs for spam filtering,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 415–422.
- [28] T. K. Ho, “Random decision forests,” in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, 1995, pp. 278–282.
- [29] T. A. Almeida, A. Yamakami, and J. M. G. Hidalgo, “Contributions to the study of SMS spam filtering: New collection and results,” in *Proceedings of the ACM Symposium on Document Engineering*, 2011, pp. 259–262.
- [30] A. Liaw and M. Wiener, “Classification and regression by randomForest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.