

STUDI SIMULASI DETEKSI EMAIL PHISHING BERBASIS NLP MENGUNAKAN TF-IDF DENGAN SUPPORT VECTOR MACHINE DAN RANDOM FOREST

Muhamad Ali Mubarak¹, Nizirwan Anwar², Arief ichwani³, Raden Teddy Iswahyudi⁴

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

Email ;

muhamadali1203@gmail.com, nizirwan.anwar@esaunggul.ac.id, arief.ichwani@esaunggul.ac.id raden.teddy@esaunggul.ac.id

Abstrak

Phishing melalui email merupakan ancaman keamanan informasi yang memanfaatkan isi pesan untuk mendorong pengguna memberikan informasi sensitif atau melakukan tindakan tertentu. Penelitian ini bertujuan menganalisis kinerja Support Vector Machine (SVM) dan Random Forest pada klasifikasi email phishing berbasis Natural Language Processing (NLP) dengan representasi Term Frequency-Inverse Document Frequency (TF-IDF). Dataset simulasi awal berjumlah 10.420 email dan setelah pemeriksaan duplikasi, data kosong, serta label tidak valid diperoleh 10.000 email, yang terdiri atas 5.000 phishing dan 5.000 legitimate. Data diproses melalui cleaning, case folding, tokenization, dan stopword removal, kemudian ditransformasikan menjadi feature vector TF-IDF dengan unigram dan bigram. Dataset dibagi menggunakan rasio 80:20 untuk data training dan testing. Evaluasi model menggunakan confusion matrix, accuracy, precision, recall, dan F1-score. Hasil simulasi menunjukkan SVM memperoleh accuracy 96,40%, precision 95,94%, recall 96,90%, dan F1-score 96,42%, sedangkan Random Forest memperoleh accuracy 94,80%, precision 95,34%, recall 94,20%, dan F1-score 94,77%. SVM juga menghasilkan false negative yang lebih rendah, yaitu 31 dibandingkan 58 pada Random Forest. Berdasarkan skenario simulasi, SVM menunjukkan kinerja lebih konsisten terutama pada recall dan F1-score, sehingga lebih sesuai sebagai model utama pada rancangan prototype deteksi email phishing. Hasil ini tetap perlu dikonfirmasi melalui eksperimen aktual dan pengujian lintas dataset sebelum digunakan sebagai dasar implementasi operasional.

Kata kunci: email phishing; natural language processing; TF-IDF; support vector machine; random forest

Abstract

Email phishing is an information-security threat that exploits message content to induce users to disclose sensitive information or perform risky actions. This study analyzes the performance of Support Vector Machine (SVM) and Random Forest for phishing email classification using a Natural Language Processing (NLP) pipeline with Term Frequency-Inverse Document Frequency (TF-IDF) representation. The simulated dataset initially contained 10,420 emails; after duplicate, missing-text, and invalid-label checks, 10,000 emails remained, comprising 5,000 phishing and 5,000 legitimate messages. Text was processed through cleaning, case folding, tokenization, and stopword removal, then represented using unigram and bigram TF-IDF features. The dataset was split into 80% training and 20% testing data. Model evaluation used a confusion matrix, accuracy, precision, recall, and F1-score. In the simulation, SVM achieved 96.40% accuracy, 95.94% precision, 96.90% recall, and 96.42% F1-score, while Random Forest achieved 94.80% accuracy, 95.34% precision, 94.20% recall, and 94.77% F1-score. SVM also produced fewer false negatives, 31 compared with 58 for Random Forest. These results indicate that SVM is more consistent in the simulated scenario, particularly in recall and F1-score, and is therefore more appropriate as the primary model for the proposed phishing-detection prototype. Actual experiments and cross-dataset validation remain necessary before operational deployment.

Keywords: phishing email; natural language processing; TF-IDF; support vector machine; random forest.

1. PENDAHULUAN

Email merupakan salah satu media komunikasi digital yang banyak digunakan pada lingkungan pendidikan, organisasi, dan bisnis. Kemudahan distribusi pesan dan dokumen membuat email menjadi sarana komunikasi yang efisien, tetapi karakteristik tersebut juga dimanfaatkan dalam serangan phishing. Pada serangan berbasis email, pelaku berupaya membuat pesan terlihat meyakinkan agar penerima membuka tautan, memasukkan informasi akun, atau menyerahkan

data pribadi. Karena bentuk pesan dapat menyerupai komunikasi resmi, deteksi phishing tidak selalu dapat bergantung pada indikator teknis sederhana. Isi bahasa pada subject dan body email menjadi salah satu sumber informasi yang dapat dianalisis untuk membedakan pesan phishing dan legitimate [2], [6].

Natural Language Processing (NLP) menyediakan rangkaian teknik untuk membersihkan, menormalkan, dan merepresentasikan teks sehingga dapat digunakan

oleh algoritma komputasi. Pada konteks phishing email, NLP dapat digunakan untuk mengekstraksi pola kata dan frasa yang muncul pada pesan. [1] menunjukkan penerapan machine learning bersama NLP untuk deteksi phishing email, sedangkan [8] melalui systematic literature review menegaskan bahwa pemrosesan bahasa merupakan komponen penting pada berbagai pendekatan deteksi phishing berbasis teks. Dengan demikian, NLP tidak hanya berfungsi sebagai tahap pembersihan data, tetapi juga menjadi penghubung antara data bahasa yang tidak terstruktur dan model klasifikasi.

Sebelum teks dapat diproses oleh algoritma machine learning, representasi numerik perlu dibentuk. Penelitian ini menggunakan Term Frequency-Inverse Document Frequency (TF-IDF), yaitu pembobotan term yang mempertimbangkan frekuensi kata pada sebuah dokumen dan tingkat kelangkaannya pada kumpulan dokumen. TF-IDF dipilih karena sederhana, transparan, dan relatif ringan dari sisi komputasi. Representasi ini juga masih relevan sebagai baseline pada penelitian phishing email, termasuk ketika dibandingkan dengan Word2Vec dan BERT [3]. Secara metodologis, TF-IDF menghasilkan feature space berdimensi tinggi dan sparse, sehingga cocok dikombinasikan dengan algoritma klasifikasi yang mampu bekerja pada data teks berdimensi besar.

Dua algoritma yang dibandingkan adalah Support Vector Machine (SVM) dan Random Forest. SVM membentuk hyperplane pemisah antarkelas dan telah lama digunakan untuk klasifikasi berdimensi tinggi [5], sedangkan Random Forest merupakan metode ensemble yang menggabungkan banyak decision tree untuk meningkatkan stabilitas prediksi [4]. Pemilihan kedua algoritma juga memiliki relevansi langsung dengan studi [10] yang membandingkan Support Vector Classification dan Random Forest pada klasifikasi phishing email. Perbandingan kedua model pada feature representation yang sama memungkinkan perbedaan performa lebih mudah dikaitkan dengan karakteristik algoritma.

Kesenjangan yang diangkat pada penelitian ini bukan pembangunan arsitektur deep learning yang kompleks, melainkan penyusunan pipeline yang sederhana dan terkontrol untuk membandingkan dua model machine learning klasik pada representasi fitur yang sama. Fokus tersebut penting untuk penelitian tingkat sarjana karena keseluruhan tahapan, mulai dari pemeriksaan data, preprocessing, pembentukan fitur, training, testing, hingga evaluasi, dapat diamati secara eksplisit. Selain itu, model yang relatif ringan lebih mudah diintegrasikan ke dalam prototype sederhana tanpa kebutuhan infrastruktur komputasi yang besar.

Berdasarkan latar belakang tersebut, penelitian bertujuan: (1) mengolah teks email menggunakan preprocessing NLP dan TF-IDF; (2) membangun model SVM dan Random Forest pada dataset dan feature representation yang sama; serta (3) membandingkan kinerja kedua model menggunakan accuracy, precision, recall, F1-score, dan confusion matrix. Karena data yang tersedia pada dokumen sumber masih berupa hasil simulasi, seluruh nilai numerik pada artikel ini diposisikan hasil simulasi.

1.1 Posisi Penelitian terhadap Studi Terdahulu

Literatur yang digunakan dalam dokumen sumber memperlihatkan perkembangan deteksi phishing dari teknik filtering menuju machine learning, NLP, deep learning, dan contextual representation. [2] memetakan teknik penyaringan phishing email, sedangkan [6] meninjau penerapan machine learning pada spam dan phishing filtering. Perkembangan berikutnya menempatkan konten linguistik sebagai sumber informasi yang penting. [1] secara langsung menggabungkan machine learning dan NLP untuk deteksi phishing email, sementara [8] merangkum berbagai teknik NLP yang digunakan pada domain tersebut.

Pada sisi representasi fitur, [3] membandingkan TF-IDF, Word2Vec, dan BERT. Studi tersebut memperlihatkan bahwa pendekatan berbasis term masih relevan sebagai baseline yang ringan sekaligus pembanding terhadap representasi yang lebih kompleks. Pada sisi classifier, [10] membandingkan Support Vector Classification dan Random Forest untuk phishing email classification. Penelitian ini mengambil irisan dari dua arah tersebut, yaitu menggunakan TF-IDF secara tetap lalu membandingkan SVM dan Random Forest dalam pipeline preprocessing, training, testing, dan evaluasi yang sama. Dengan desain ini, fokus pembahasan tidak tersebar pada banyak representasi atau banyak algoritma, tetapi diarahkan pada perbandingan terkontrol dua model yang mudah direproduksi.

Tabel 1. Posisi penelitian terhadap beberapa studi terkait

Penelitian	Fokus utama	Keterkaitan dengan penelitian ini
[2]	Survei filtering phishing email	Dasar perkembangan teknik deteksi
[6]	Review ML untuk spam/phishing	Dasar penggunaan machine learning
[1]	ML dan NLP	Dasar pipeline NLP + classifier
[8]	Systematic review NLP	Dasar pemrosesan teks

Penelitian	Fokus utama	Keterkaitan dengan penelitian ini
[3]	TF-IDF, Word2Vec, BERT	Dasar pemilihan TF-IDF
[10]	SVC dan Random Forest	Dasar model pembandingan

Posisi tersebut membuat kontribusi artikel bersifat metodologis dan aplikatif. Artikel tidak dimaksudkan untuk mengklaim kebaruan arsitektur, tetapi untuk menunjukkan bagaimana pipeline klasifikasi phishing berbasis teks dapat disusun secara transparan, dibandingkan dengan metrik yang konsisten, dan diarahkan menuju prototype. Pendekatan ini sesuai untuk penelitian sarjana serta dapat menjadi baseline sebelum menggunakan embedding atau transformer yang lebih kompleks.

2. METODE PENELITIAN

Penelitian menggunakan pendekatan eksperimen komparatif untuk membangun dan membandingkan dua model klasifikasi email phishing. Alur umum penelitian terdiri atas pemeriksaan dataset, text preprocessing, transformasi TF-IDF, pembagian data, pelatihan SVM dan Random Forest, pengujian, evaluasi performa, serta pemilihan model terbaik. Tahapan tersebut dirancang agar kedua algoritma menerima data dan feature representation yang sama sehingga perbandingan dilakukan pada kondisi yang setara.



Gambar 1. Alur penelitian deteksi email phishing berbasis NLP dan TF-IDF

2.1 Dataset dan Pemeriksaan Data

Dataset penelitian terdiri atas email yang memiliki dua label, yaitu phishing dan legitimate. Pada simulasi, dataset awal berjumlah 10.420 email. Pemeriksaan dilakukan untuk mengidentifikasi data duplikat, teks kosong, dan label tidak valid. Sebanyak 280 email duplikat, 90 email dengan teks kosong, dan 50 email dengan label tidak valid dikeluarkan, sehingga diperoleh 10.000 email yang dapat digunakan pada tahapan berikutnya. Dataset akhir memiliki distribusi seimbang, yaitu 5.000 phishing dan 5.000 legitimate.

Tabel 2. Pemeriksaan dataset simulasi

Kondisi data	Jumlah email	Keterangan
Dataset awal	10.420	Data sebelum

Kondisi data	Jumlah email	Keterangan
		pemeriksaan
Data duplikat	280	Dikeluarkan
Data kosong	90	Dikeluarkan
Label tidak valid	50	Dikeluarkan
Dataset akhir	10.000	Digunakan untuk pemodelan

Dataset akhir kemudian dibagi menjadi 8.000 email untuk training dan 2.000 email untuk testing dengan rasio 80:20. Karena kelas phishing dan legitimate seimbang, masing-masing kelas berkontribusi 4.000 data pada training dan 1.000 data pada testing. Pembagian dilakukan secara stratified untuk mempertahankan distribusi kelas. Proses fitting TF-IDF dilakukan hanya pada training data, sedangkan testing data ditransformasikan menggunakan vectorizer yang dibentuk dari training data untuk mengurangi risiko data leakage.

2.2 Text Preprocessing

Preprocessing bertujuan mengurangi noise dan menyeragamkan bentuk teks sebelum feature extraction. Tahapan terdiri atas cleaning, case folding, tokenization, dan stopword removal. Cleaning menghilangkan karakter khusus, HTML tag, serta whitespace yang tidak diperlukan. Case folding mengubah teks ke lowercase agar variasi kapitalisasi tidak membentuk fitur yang berbeda. Tokenization memecah teks menjadi unit kata, sedangkan stopword removal menghapus kata umum yang dianggap memiliki informasi diskriminatif rendah. Tahapan ini disesuaikan dengan karakteristik bahasa dataset agar informasi yang berpotensi menjadi indikator phishing tidak ikut terhapus.

Tabel 3. Contoh tahapan preprocessing

Tahap	Contoh hasil
Raw text	URGENT!!! Please Verify Your ACCOUNT Immediately!!!
Cleaning	URGENT Please Verify Your ACCOUNT Immediately
Case folding	urgent please verify your account immediately
Tokenization	urgent please verify your account immediately
Stopword removal	urgent verify account immediately

2.3 Representasi TF-IDF

Teks hasil preprocessing diubah menjadi feature vector menggunakan TF-IDF. Secara umum, bobot TF-IDF suatu term t pada dokumen d dapat ditulis sebagai $TFIDF(t,d) = TF(t,d) \times IDF(t)$.

Term yang sering muncul pada suatu dokumen namun relatif jarang pada keseluruhan dokumen akan memperoleh bobot lebih tinggi. Konsep pembobotan term semacam ini merupakan pendekatan klasik dalam information retrieval dan text representation [9]. Pada simulasi, konfigurasi menggunakan analyzer berbasis kata, kombinasi unigram dan bigram, min_df = 2, max_df = 0,95, max_features = 20.000, dan normalisasi L2.

Tabel 4. Konfigurasi TF-IDF pada simulasi

Parameter	Nilai
Analyzer	Word
N-gram	Unigram + bigram
min_df	2
max_df	0,95
max_features	20.000
Normalization	L2

Penggunaan unigram memungkinkan model mengenali kata individual seperti verify, account, password, login, dan security, sedangkan bigram dapat menangkap frasa seperti verify account, click link, account suspended, atau security alert. Kombinasi keduanya diharapkan memberikan representasi yang lebih kaya tanpa meningkatkan kompleksitas ke tingkat contextual embedding.

2.4 Pembangunan Model

SVM diuji menggunakan kernel linear dengan nilai C sebesar 0,1; 1; dan 10. Kernel linear dipilih sebagai konfigurasi awal karena feature space TF-IDF umumnya berdimensi tinggi dan sparse. Parameter C mengatur kompromi antara margin dan kesalahan klasifikasi. Sementara itu, Random Forest diuji menggunakan n_estimators sebanyak 100, 200, dan 300 tree. Parameter max_depth dibiarkan tidak dibatasi pada konfigurasi dasar. Implementasi model secara konseptual dapat dilakukan menggunakan library machine learning seperti scikit-learn [7].

Tabel 5. Skenario parameter model

Model	Parameter	Nilai uji
SVM	Kernel	Linear
SVM	C	0,1; 1; 10
Random Forest	n_estimators	100; 200; 300
Random Forest	max_depth	None / disesuaikan

2.5 Evaluasi Model

Kinerja model dievaluasi menggunakan confusion matrix yang terdiri atas True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Pada penelitian ini, phishing diperlakukan sebagai kelas positif. Accuracy dihitung sebagai $(TP + TN)/(TP + TN + FP + FN)$,

precision sebagai $TP/(TP + FP)$, recall sebagai $TP/(TP + FN)$, dan F1-score sebagai $2 \times (\text{precision} \times \text{recall})/(\text{precision} + \text{recall})$. Model terbaik tidak hanya ditentukan melalui accuracy, tetapi juga mempertimbangkan recall dan F1-score. Recall diberi perhatian khusus karena false negative berarti email phishing salah diklasifikasikan sebagai legitimate dan berpotensi lolos dari sistem deteksi.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Preprocessing dan Pembentukan Fitur

Setelah pemeriksaan data, dataset simulasi yang digunakan berjumlah 10.000 email dengan distribusi seimbang. Preprocessing menghasilkan teks yang lebih seragam, misalnya pesan “URGENT!!! Please Verify Your ACCOUNT Immediately!!!” menjadi “urgent verify account immediately” setelah cleaning, case folding, tokenization, dan stopword removal. Hasil tersebut kemudian direpresentasikan ke dalam ruang fitur TF-IDF. Kombinasi unigram dan bigram memungkinkan model memanfaatkan informasi tingkat kata dan pasangan kata. Dalam contoh simulasi, frasa seperti verify account, account suspended, update password, login account, dan click link memperoleh bobot relatif tinggi karena berkaitan dengan pola permintaan atau peringatan yang sering muncul pada phishing.

Meskipun demikian, bobot tinggi suatu term tidak secara otomatis menunjukkan bahwa email tersebut phishing. Kata seperti account, password, dan security juga dapat muncul pada komunikasi legitimate. Kondisi ini menjelaskan mengapa representasi berbasis term masih dapat menghasilkan false positive. TF-IDF membantu membedakan distribusi kata, tetapi tidak memahami konteks semantik secara mendalam. Keterbatasan tersebut menjadi salah satu alasan evaluasi model perlu memperhatikan kesalahan klasifikasi, bukan hanya nilai accuracy.

3.2 Hasil Pengujian SVM

Pada tahap pemilihan parameter, SVM dengan $C = 1$ memberikan accuracy validasi simulasi tertinggi sebesar 96,20%, lebih tinggi dibandingkan $C = 0,1$ sebesar 94,85% dan $C = 10$ sebesar 95,75%. Konfigurasi tersebut kemudian digunakan pada 2.000 testing data. Confusion matrix menghasilkan 969 true positive, 959 true negative, 41 false positive, dan 31 false negative. Berdasarkan nilai tersebut, SVM memperoleh accuracy 96,40%, precision 95,94%, recall 96,90%, dan F1-score 96,42%.

Tabel 6. Hasil simulasi parameter SVM

Konfigurasi	C	Accuracy validasi
SVM-1	0,1	94,85%
SVM-2	1	96,20%
SVM-3	10	95,75%

Recall sebesar 96,90% menunjukkan bahwa dari 1.000 email phishing pada testing data, 969 berhasil dideteksi dan 31 email phishing salah diklasifikasikan sebagai legitimate. Pada konteks keamanan, jumlah false negative menjadi indikator penting karena kesalahan tersebut merepresentasikan ancaman yang tidak berhasil dikenali. SVM pada simulasi menghasilkan false negative rate sebesar 3,10%.

3.3 Hasil Pengujian Random Forest

Random Forest diuji menggunakan 100, 200, dan 300 tree. Konfigurasi 200 tree menghasilkan accuracy validasi simulasi tertinggi sebesar 94,65%, sedangkan 100 tree menghasilkan 93,70% dan 300 tree menghasilkan 94,40%. Pada testing data, confusion matrix menghasilkan 942 true positive, 954 true negative, 46 false positive, dan 58 false negative. Berdasarkan nilai tersebut, Random Forest memperoleh accuracy 94,80%, precision 95,34%, recall 94,20%, dan F1-score 94,77%.

Tabel 7. Hasil simulasi parameter Random Forest

Konfigurasi	n_estimators	Accuracy validasi
RF-1	100	93,70%
RF-2	200	94,65%
RF-3	300	94,40%

Random Forest masih menghasilkan performa yang relatif tinggi pada seluruh metrik, namun jumlah false negative lebih besar dibandingkan SVM. Dari 1.000 email phishing, 58 email tidak berhasil dikenali dan diklasifikasikan sebagai legitimate. Nilai tersebut setara dengan false negative rate 5,80%. Perbedaan ini menjadi penting ketika tujuan sistem adalah meminimalkan ancaman phishing yang lolos.

3.4 Perbandingan Kinerja Model

Tabel 8. Perbandingan hasil simulasi SVM dan Random Forest

Metrik	SVM	Random Forest	Selisih
Accuracy	96,40%	94,80%	1,60 pp
Precision	95,94%	95,34%	0,60 pp
Recall	96,90%	94,20%	2,70 pp
F1-score	96,42%	94,77%	1,65 pp
False positive	41	46	5 email

Metrik	SVM	Random Forest	Selisih
False negative	31	58	27 email
Total error	72	104	32 email

Perbandingan menunjukkan bahwa SVM memberikan nilai lebih tinggi pada seluruh metrik utama. Selisih accuracy sebesar 1,60 poin persentase terlihat moderat, tetapi perbedaan tersebut menjadi lebih bermakna ketika diterjemahkan ke jumlah kesalahan. SVM menghasilkan 72 kesalahan dari 2.000 email, sedangkan Random Forest menghasilkan 104 kesalahan. Dengan demikian, SVM mengurangi 32 kesalahan prediksi pada skenario simulasi yang sama. Error rate SVM sebesar 3,60%, sedangkan Random Forest sebesar 5,20%.

Perbedaan paling besar terdapat pada recall, yaitu 2,70 poin persentase. SVM mendeteksi 27 email phishing lebih banyak dibandingkan Random Forest. Secara relatif, false negative pada SVM juga jauh lebih rendah. Temuan ini mendukung pemilihan SVM sebagai model utama karena sistem deteksi phishing perlu mengutamakan kemampuan menangkap kelas ancaman. Precision SVM juga sedikit lebih tinggi dan false positive lebih rendah, sehingga peningkatan recall tidak diikuti lonjakan kesalahan terhadap email legitimate.

Dari sudut pandang karakteristik algoritma, hasil tersebut dapat dijelaskan melalui sifat feature space TF-IDF. Representasi TF-IDF menghasilkan matriks berdimensi tinggi dan sparse. SVM linear bekerja dengan mencari hyperplane yang memisahkan dua kelas pada ruang tersebut, sedangkan Random Forest membentuk keputusan melalui pemisahan berulang pada subset fitur di sejumlah decision tree. Pada data teks yang memiliki banyak fitur sparse, pemisahan linear dapat memberikan batas keputusan yang lebih efektif jika pola phishing dan legitimate cukup terpisah. Namun, interpretasi ini bersifat kontekstual dan tidak membuktikan bahwa SVM selalu lebih baik dari Random Forest pada semua dataset.

Hasil penelitian sejalan dengan posisi metodologis dalam studi terdahulu bahwa performa deteksi phishing dipengaruhi oleh representasi teks dan algoritma klasifikasi. [10] menempatkan Support Vector Classification dan Random Forest sebagai model yang dapat dibandingkan pada phishing email classification. [3] juga menunjukkan bahwa TF-IDF masih relevan sebagai salah satu bentuk representasi teks ketika dibandingkan dengan Word2Vec dan BERT. Pada artikel ini, kontribusi utamanya terletak pada pipeline yang sederhana, eksplisit, dan terkontrol untuk

membandingkan kedua model menggunakan preprocessing, feature representation, dan testing data yang sama.

3.5 Analisis False Positive dan False Negative

Analisis confusion matrix menunjukkan bahwa kesalahan kedua model tidak hanya berbeda dari sisi jumlah, tetapi juga dari sisi konsekuensi. SVM menghasilkan 41 false positive dan 31 false negative, sedangkan Random Forest menghasilkan 46 false positive dan 58 false negative. False positive terjadi ketika email legitimate diprediksi sebagai phishing. Contoh yang berpotensi memicu kondisi tersebut adalah komunikasi resmi yang mengandung kata update, account, password, atau security. Karena TF-IDF memberikan bobot berdasarkan distribusi term, kemunculan kata-kata tersebut dapat meningkatkan kemiripan dengan pola phishing walaupun konteks sebenarnya legitimate.

False negative lebih kritis karena terjadi ketika email phishing diprediksi sebagai legitimate. Pesan phishing yang menggunakan bahasa sopan, tidak menampilkan kata-kata mencolok seperti urgent atau suspended, dan meniru pola komunikasi resmi dapat lebih sulit dibedakan oleh representasi berbasis term. Pada simulasi, SVM melewati 31 dari 1.000 email phishing, sedangkan Random Forest melewati 58. Perbedaan 27 email menunjukkan bahwa keunggulan recall SVM memiliki implikasi langsung terhadap tujuan keamanan, yaitu mengurangi ancaman yang lolos tanpa peringatan.

Tabel 9. Analisis kesalahan klasifikasi pada data simulasi

Jenis kesalahan	SVM	Random Forest	Implikasi
False positive	41	46	Email legitimate salah ditandai phishing
False negative	31	58	Email phishing lolos sebagai legitimate
Total error	72	104	Jumlah seluruh prediksi salah

Meskipun SVM lebih baik pada kedua jenis kesalahan, konfigurasi model pada implementasi nyata tetap perlu mempertimbangkan trade-off antara sensitivitas dan gangguan terhadap pengguna. Sistem yang terlalu agresif dapat meningkatkan false positive, sedangkan sistem yang terlalu konservatif dapat meningkatkan false

negative. Oleh karena itu, threshold, class weighting, atau strategi evaluasi tambahan perlu dipertimbangkan apabila model dikembangkan di luar skenario binary classification yang seimbang.

3.6 Implikasi untuk Prototype Sistem

Model terbaik pada simulasi digunakan sebagai dasar rancangan prototype deteksi email phishing. Alur prototype terdiri atas input subject dan body email, preprocessing menggunakan prosedur yang sama dengan training data, transformasi menggunakan TF-IDF vectorizer yang telah disimpan, prediksi menggunakan model SVM terlatih, dan keluaran berupa label phishing atau legitimate. Pada tahap deployment, vectorizer tidak di-fit ulang terhadap email baru karena vocabulary dan bobot IDF harus konsisten dengan training data. Pemisahan antara training phase dan prediction phase juga mencegah sistem melakukan pelatihan ulang setiap kali menerima email baru.

Prototype sederhana dapat digunakan sebagai demonstrasi bahwa model hasil training dapat menerima teks baru dan menghasilkan prediksi. Namun, sistem tersebut belum dapat dianggap sebagai solusi keamanan email produksi. Implementasi operasional memerlukan pengujian terhadap data aktual, distribusi kelas yang tidak seimbang, perubahan pola phishing dari waktu ke waktu, serta integrasi fitur tambahan seperti URL, email header, domain pengirim, alamat IP, dan attachment. Selain itu, pengujian lintas dataset diperlukan untuk mengetahui kemampuan generalisasi model.

3.7 Keterbatasan Penelitian

Terdapat beberapa keterbatasan yang perlu diperhatikan. Pertama, seluruh nilai performa pada artikel ini berasal dari data simulasi yang disusun pada dokumen sumber dan belum merupakan hasil eksekusi eksperimen aktual. Kedua, pendekatan hanya menggunakan teks email dan belum memasukkan fitur struktural maupun reputasi sumber. Ketiga, TF-IDF merepresentasikan frekuensi term dan belum memahami konteks semantik secara mendalam. Keempat, distribusi data simulasi seimbang 50:50, sedangkan lingkungan email operasional dapat didominasi kelas legitimate. Kelima, perbandingan hanya mencakup SVM dan Random Forest. Oleh karena itu, kesimpulan artikel dibatasi pada skenario simulasi dan tidak dapat digeneralisasikan ke seluruh kondisi phishing email detection.

Sebelum naskah diajukan ke jurnal, hasil simulasi perlu diganti atau dikonfirmasi dengan eksperimen aktual yang terdokumentasi. Minimal,

penelitian perlu menjalankan preprocessing pada dataset asli, menyimpan random seed, melakukan pembagian data secara reproducible, melaporkan hasil confusion matrix aktual, serta menambahkan cross-validation. Jika memungkinkan, pengujian pada dataset kedua akan memperkuat klaim generalisasi. Langkah ini penting agar artikel memenuhi standar evidensi empiris yang lazim pada jurnal terindeks SINTA.

Aspek reproducibility juga perlu diperkuat dengan mencatat versi Python dan library, parameter TF-IDF, parameter model, random_state, serta prosedur penyimpanan vectorizer dan model. Dokumentasi tersebut memungkinkan peneliti lain mengulangi eksperimen pada kondisi yang sama. Pada artikel berbasis machine learning, transparansi konfigurasi sering sama pentingnya dengan pelaporan nilai metrik karena perubahan kecil pada pembagian data atau parameter dapat mengubah hasil evaluasi.

4. KESIMPULAN

Penelitian menyusun pipeline deteksi email phishing berbasis NLP yang terdiri atas pemeriksaan dataset, preprocessing teks, representasi TF-IDF, pembangunan SVM dan Random Forest, serta evaluasi menggunakan confusion matrix, accuracy, precision, recall, dan F1-score. Berdasarkan skenario simulasi terhadap 10.000 email, SVM memberikan performa lebih tinggi dibandingkan Random Forest. SVM memperoleh accuracy 96,40%, precision 95,94%, recall 96,90%, dan F1-score 96,42%, sedangkan Random Forest memperoleh accuracy 94,80%, precision 95,34%, recall 94,20%, dan F1-score 94,77%. SVM juga menghasilkan false negative lebih rendah, yaitu 31 dibandingkan 58. Dengan demikian, TF-IDF + SVM menjadi kandidat model utama pada rancangan prototype karena lebih konsisten dalam menangkap email phishing dan menghasilkan jumlah kesalahan lebih rendah. Meskipun demikian, seluruh hasil tersebut masih bersifat simulasi. Pengembangan berikutnya perlu menggunakan hasil eksperimen aktual, cross-validation, pengujian lintas dataset, fitur struktural

email, serta representasi kontekstual untuk menilai generalisasi model secara lebih kuat.

DAFTAR PUSTAKA

- [1] Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers & Security*, 110, 102414.
- [2] Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys & Tutorials*, 15(4), 2070–2090.
- [3] Al Tawil, A., Almazaydeh, L., Qawasmeh, D., Qawasmeh, B., Alshinwan, M., & Elleithy, K. (2024). Comparative analysis of machine learning algorithms for email phishing detection using TF-IDF, Word2Vec, and BERT. *Computers, Materials & Continua*, 81(2), 3395–3412.
- [4] Breiman, L. (2001). Random forests. *Machine Learning*, 45, 5–32.
- [5] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297.
- [6] Gangavarapu, T., Jaidhar, C. D., & Chanduka, B. (2020). Applicability of machine learning in spam and phishing email filtering: Review and approaches. *Artificial Intelligence Review*, 53(7), 5019–5081.
- [7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [8] Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10, 65703–65727.
- [9] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523.
- [10] Umam, C., Handoko, L. B., & Isinkaye, F. O. (2024). Performance analysis of Support Vector Classification and Random Forest in phishing email classification. *Scientific Journal of Informatics*, 11(2), 367–374.