ROUTERS: Jurnal Sistem dan Teknologi Informasi
Vol. 4, No. 1, Februari 2026, pp. 13-20
ISSN: 2962-1224, DOI: 10.25181/rt.v5i1.4330

Analisis Performa Random Forest, Decision Tree, dan Naive Bayes untuk Deteksi Link *Phishing* Berbasis Fitur URL

Silcilia[1], Nadira Parsha Salsabila[2], Tarisza Apriani[3]
[1,2,3]Teknologi Rekayasa Internet, Jurusan Teknologi Informasi, Politeknik Negeri Lampung

INFORMASI ARTIKEL

Diterima 24 Juli 2025
Direvisi 13 Februari 2026
Diterbitkan 19 Februari 2026

Kata kunci:

Phishing;
Random Forest;
Decision Tree;
Naïve Bayes;
Pembelajaran mesin

ABSTRAK

Serangan *phishing* melalui tautan palsu masih menjadi salah satu ancaman siber yang sering terjadi dan dapat mengakibatkan kebocoran data pada pengguna jaringan komputer. Deteksi manual seringkali kurang efektif karena metode *phishing* terus berkembang dengan pola tautan yang menyerupai domain asli. Eksperimen ini bertujuan untuk menganalisis performa tiga algoritma klasifikasi, yaitu Random Forest, Decision Tree, dan Naive Bayes, dalam mendeteksi tautan *phishing* berbasis fitur dasar URL. Eksperimen ini diharapkan dapat membantu mengenali URL *phishing* secara otomatis berdasarkan pola karakteristik tautan yang dianalisis dengan metode machine learning. Eksperimen dilakukan dengan mengumpulkan *dataset* berisi tautan *phishing* dan *legitimate*, kemudian dilakukan ekstraksi fitur seperti panjang URL, panjang *hostname*, jumlah simbol tertentu, deteksi IP address pada domain, penggunaan layanan penyingkat URL, serta pola *prefix-suffix* pada *hostname*. *Dataset* dibagi menjadi data latih dan data uji dengan perbandingan 80:20. Model dilatih menggunakan ketiga algoritma dan diuji untuk membandingkan akurasi, *precision* , *recall*, serta F1-*score*. Hasil pengujian menunjukkan bahwa algoritma Random Forest memiliki akurasi tertinggi sebesar 80,75% dengan *precision* dan *recall* yang seimbang, sedangkan Decision Tree memiliki akurasi 77,73% dan Naive Bayes hanya mencapai akurasi 68,15%. Temuan ini menunjukkan bahwa Random Forest paling layak digunakan untuk mendeteksi tautan *phishing* berbasis analisis fitur URL sederhana. Dengan demikian, model ini dapat diterapkan sebagai sistem deteksi dini untuk meminimalkan risiko serangan *phishing* di berbagai lingkungan.

Performance Analysis of Random Forest, Decision Tree, and Naive Bayes for Phishing Link Detection Based on URL Features

ARTICLE INFO

Received July 24, 2025
Revised Februariy13, 2026
Published February 19, 2026

Keyword:

Phishing;
Random Forest;
Decision Tree;
Naïve Bayes;
Machine Learning

ABSTRACT

Phishing attacks through fake links remain one of the most common cybersecurity threats and can lead to data breaches for computer network users. Manual detection is often ineffective because phishing methods continue to evolve, with link patterns that closely resemble legitimate domains. This experiment aims to analyze the performance of three classification algorithms – Random Forest, Decision Tree, and Naive Bayes – in detecting phishing links based on basic URL features. The experiment is expected to assist in the automatic recognition of phishing URLs based on link characteristics analyzed using machine learning methods. The process involves collecting a dataset containing both phishing and legitimate links, followed by feature extraction such as URL length, hostname length, number of specific symbols, detection of IP addresses in the domain, use of URL shortening services, and prefix-suffix patterns in the hostname. The dataset is divided into training and testing data with an 80:20 ratio. The models are trained using the three algorithms and tested to compare their accuracy, precision , recall, and F1-score. The testing results show that the Random Forest algorithm achieved the highest accuracy of 80.75%, with balanced precision and recall. Meanwhile, the Decision Tree achieved an accuracy of 77.73%, and Naive Bayes only reached 68.15%. These findings indicate that Random Forest is the most suitable for detecting phishing links based on simple URL feature analysis. Therefore, this model can be applied as an early detection system to minimize phishing attack risks in various environments.

Corresponding Author:

Silcilia, Politeknik Negeri Lampung
Email: silcilia1605@gmail.com

1. PENDAHULUAN

Pesatnya perkembangan teknologi internet telah membawa perubahan besar dalam cara manusia berinteraksi dan bertukar informasi. Kemudahan akses dan konektivitas global yang ditawarkan teknologi digital menjadikan berbagai aktivitas, termasuk komunikasi, transaksi, dan layanan publik, semakin praktis. Namun, di balik kemudahan tersebut, muncul berbagai risiko keamanan informasi yang dapat mengancam data pribadi pengguna [1]. Salah satu bentuk kejahatan siber yang masih sering terjadi hingga saat ini adalah *phishing*, yaitu tindakan penipuan dengan cara mengecoh pengguna melalui tautan palsu yang dibuat sangat mirip dengan tautan resmi. Serangan *phishing* umumnya digunakan untuk mencuri informasi penting seperti kredensial akun, kata sandi, atau data finansial yang bersifat sensitif [2].

Pola serangan *phishing* terus berkembang mengikuti tren teknologi, sehingga cara-cara manual untuk mendeteksi link *phishing* semakin sulit diandalkan. Penjahat siber memanfaatkan berbagai teknik rekayasa sosial dan manipulasi link agar korban sulit membedakan antara tautan asli dan palsu [3]. Oleh karena itu, deteksi *phishing* memerlukan pendekatan otomatis yang mampu

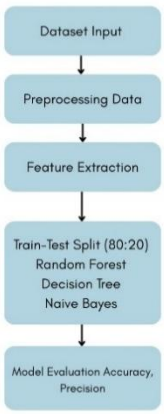
mengenali pola-pola mencurigakan pada URL secara cepat dan akurat. Salah satu solusi yang banyak digunakan adalah penerapan algoritma machine learning untuk menganalisis ciri-ciri dasar pada sebuah URL [4].

Sejumlah penelitian sebelumnya telah menunjukkan bahwa fitur dasar URL, seperti panjang tautan, panjang hostname, jumlah karakter simbol tertentu, penggunaan alamat IP pada domain, serta pemanfaatan layanan penyingkat URL, dapat digunakan untuk membedakan tautan *phishing* dengan tautan *legitimate* [5]. Berbagai algoritma klasifikasi juga telah diuji coba untuk mengolah fitur-fitur tersebut, mulai dari metode sederhana hingga model dengan kompleksitas yang lebih tinggi. Namun demikian, kajian yang membandingkan performa beberapa algoritma secara bersamaan dengan fokus pada akurasi dan efektivitas deteksi link *phishing* berbasis fitur URL masih terbatas dan perlu dieksplorasi lebih mendalam.

Eksperimen ini dilakukan untuk menjawab kebutuhan tersebut dengan menganalisis dan membandingkan performa tiga algoritma klasifikasi, yaitu Random Forest, Decision Tree, dan Naive Bayes. Ketiga algoritma dipilih karena memiliki karakteristik yang berbeda dalam hal cara kerja, interpretasi model, dan tingkat akurasi yang dihasilkan pada data tabular [6]. Melalui eksperimen ini, diharapkan dapat diperoleh informasi yang lebih komprehensif mengenai algoritma yang paling sesuai untuk diterapkan sebagai deteksi awal tautan *phishing* berbasis fitur URL. Kajian ini diharapkan dapat membantu pengembangan sistem keamanan yang mampu mendeteksi URL *phishing* secara otomatis, sehingga dapat meminimalkan risiko serangan *phishing* di berbagai sektor.

2. METODE

Eksperimen ini dilaksanakan melalui beberapa tahapan utama yang disusun secara sistematis agar tujuan eksperiman dapat tercapai. Secara umum, metode yang digunakan meliputi pengumpulan *dataset*, *preprocessing data*, *feature extraction*, pembagian data menjadi *training data* dan *testing data*, pelatihan model klasifikasi menggunakan tiga algoritma *machine learning*, serta evaluasi performa model berdasarkan metrik akurasi, *precision* , *recall*, dan f1-score [7]. Alur lengkap proses eksperimen ini ditunjukkan pada Gambar 1.



Gambar 1. Diagram blok prosedur alur eksperimen deteksi link *phishing*.

2.1. Desain Pengolahan Data

Eksperimen ini menggunakan pendekatan kuantitatif dengan menerapkan metode klasifikasi berbasis *machine learning* untuk mendeteksi link *phishing*. Desain pengolahan data diawali dengan tahap pengumpulan *dataset* yang terdiri dari kumpulan URL *phishing* dan URL legitimate. *Dataset* tersebut kemudian diproses melalui tahap *preprocessing* agar data siap digunakan sebagai input model *machine learning*. Selanjutnya dilakukan *feature extraction* untuk mendapatkan informasi dasar dari setiap URL, seperti panjang URL, panjang *hostname*, jumlah karakter simbol tertentu, keberadaan IP address pada domain, penggunaan *shortening service*, serta pola *prefix-suffix* pada hostname.

Tahap berikutnya adalah pembagian *dataset* menjadi *training data* dan *testing data* dengan perbandingan 80:20. *Training data* digunakan untuk membangun model klasifikasi, sedangkan *testing data* digunakan untuk mengukur performa model [8]. Tiga algoritma yang digunakan dalam eksperimen ini adalah Random Forest, Decision Tree, dan Naive Bayes. Pemilihan ketiga algoritma didasarkan pada popularitas, kemudahan interpretasi, dan kemampuan algoritma dalam mengolah data tabular dengan banyak fitur [9].

2.2. Prosedur Pelaksanaan

Prosedur pelaksanaan melibatkan beberapa langkah utama. Pertama, *dataset* diimpor ke dalam program Python menggunakan pustaka *pandas*, kemudian dilakukan *label encoding* untuk mengubah status URL (*phishing* atau legitimate) menjadi numerik. Setelah itu dilakukan *feature extraction* sebanyak 12 fitur dasar URL menggunakan modul *urlparse* dan *regular expression*. Fitur-fitur yang diekstraksi meliputi panjang URL, panjang hostname, jumlah titik, tanda hubung, simbol '@', tanda tanya, '&', '=', keberadaan token HTTPS, pola *prefix-suffix* pada domain, penggunaan *shortening service*, dan pola IP address.

Setelah seluruh fitur diperoleh, *dataset* dibagi menjadi *training data* dan *testing data*. Model Random Forest, Decision Tree, dan Naive Bayes dilatih menggunakan *training data*. Evaluasi model dilakukan dengan menghitung metrik akurasi, *precision* , *recall*, dan f1-score pada *testing data* [10]. Eksperimen ini menggunakan pustaka *scikit-learn* sebagai alat bantu dalam proses pelatihan dan evaluasi model.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil pengujian model deteksi link *phishing* menggunakan tiga algoritma machine learning, yaitu Random Forest, Decision Tree, dan Naive Bayes. Pengujian dilakukan dengan menggunakan data uji sebesar 20% dari total *dataset*. Evaluasi kinerja model dilakukan menggunakan metrik akurasi, *precision* , *recall*, dan f1-score. Untuk memperkuat interpretasi hasil, analisis juga dilengkapi dengan visualisasi *feature importance* dan *confusion matrix*. Ringkasan performa masing-masing algoritma disajikan pada Tabel 1.

Tabel 1. Hasil Evaluasi Model Deteksi Link *Phishing*

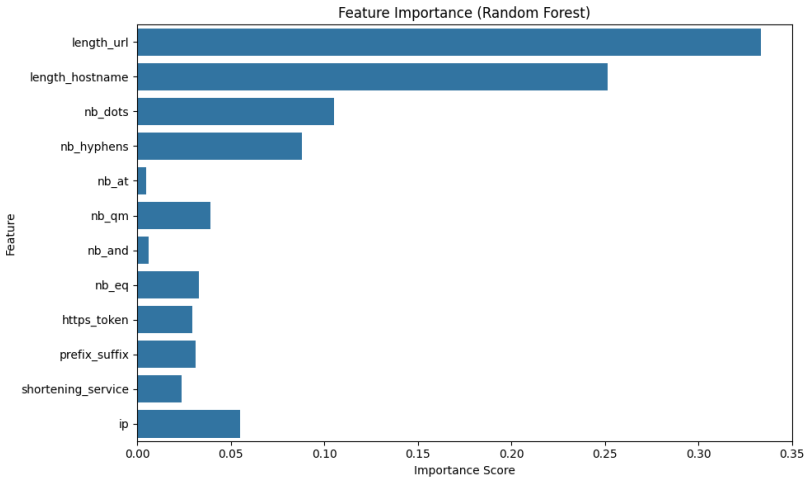
Algoritma	Akurasi (%)	*Precision* (*Phishing*)	*Recall* (*Phishing*)	F1-Score (*Phishing*)
Random Forest	80,75	82%	78%	80%
Decision Tree	77,73	80%	73%	76%
Naive Bayes	68,15	90%	40%	55%

Berdasarkan Tabel 1, Random Forest menunjukkan performa terbaik dengan akurasi tertinggi sebesar 80,75%. Nilai *precision* (82%) dan *recall* (78%) yang relatif seimbang mengindikasikan bahwa model ini tidak hanya mampu mengklasifikasikan URL *phishing* secara tepat, tetapi juga cukup sensitif dalam mendeteksi sebagian besar kasus *phishing*. Keseimbangan ini penting dalam konteks keamanan siber karena kesalahan dalam mendeteksi URL *phishing* (false negative) berpotensi menyebabkan pengguna tetap terpapar ancaman.

Decision Tree berada pada posisi kedua dengan akurasi sebesar 77,73%. Meskipun nilai *precision* dan *recall* masih tergolong baik, performanya berada di bawah Random Forest. Hal ini dapat disebabkan oleh karakteristik Decision Tree yang cenderung sensitif terhadap variasi data dan

berpotensi mengalami overfitting, sehingga kemampuan generalisasi terhadap data uji menjadi lebih rendah dibandingkan metode ensemble seperti Random Forest.

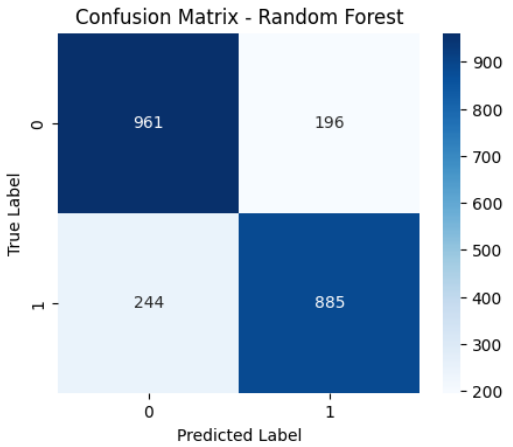
Sebaliknya, Naive Bayes memiliki akurasi terendah sebesar 68,15%. Meskipun *precision* pada kelas *phishing* cukup tinggi (90%), nilai *recall* yang rendah (40%) menunjukkan bahwa model ini gagal mendeteksi sebagian besar URL *phishing* dalam data uji. Dengan kata lain, Naive Bayes cenderung bias mengklasifikasikan URL sebagai legitimate, sehingga kurang sesuai jika diterapkan sebagai sistem deteksi dini *phishing* yang membutuhkan sensitivitas tinggi terhadap ancaman.



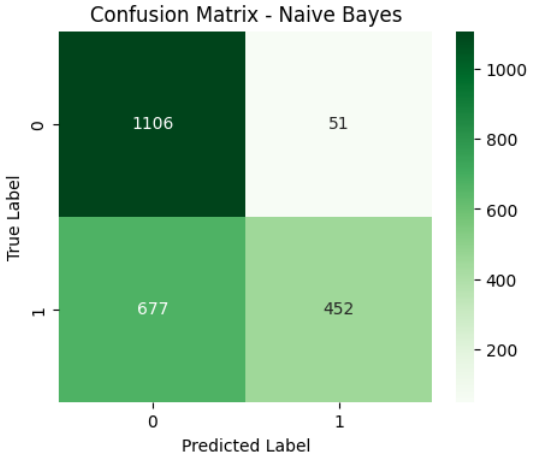
Gambar 2. *Feature importance* Random Forest untuk deteksi link *phishing*

Gambar 2 memperlihatkan hasil *feature importance* pada model Random Forest. Terlihat bahwa fitur *length_url* dan *length_hostname* memiliki kontribusi terbesar dalam proses klasifikasi. Hal ini menunjukkan bahwa panjang URL dan hostname merupakan indikator penting dalam membedakan URL *phishing* dan legitimate. URL *phishing* umumnya memiliki struktur yang lebih panjang dan kompleks sebagai upaya menyamarkan identitas domain palsu agar menyerupai domain resmi.

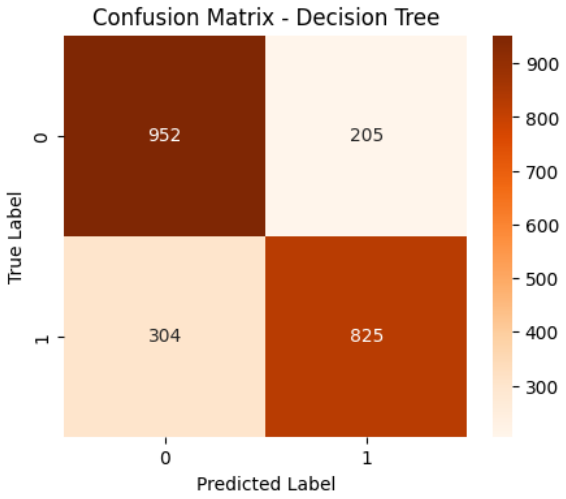
Selain itu, fitur *nb_dots* dan *nb_hyphens* juga memiliki pengaruh yang cukup signifikan. Penggunaan banyak titik dan tanda hubung sering dimanfaatkan untuk membentuk subdomain palsu yang menyerupai domain asli. Temuan ini memperkuat bahwa karakteristik struktural URL sederhana dapat menjadi indikator yang efektif dalam mendeteksi aktivitas *phishing* secara otomatis.



Gambar 3. *Confusion matrix* Random Forest



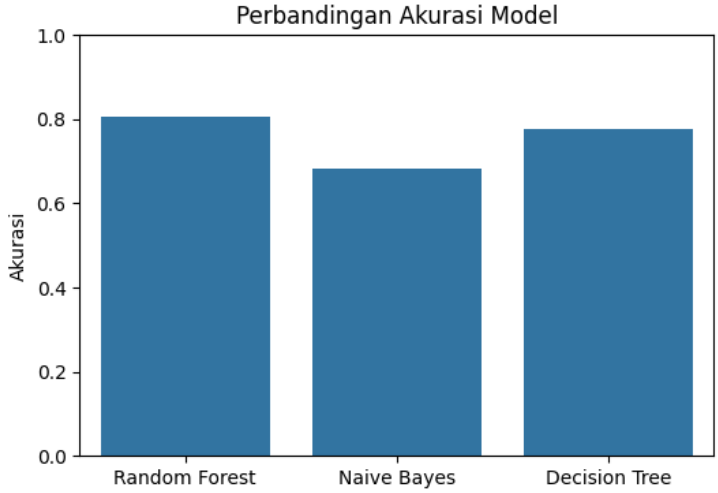
Gambar 4. *Confusion matrix* Naive Bayes



Gambar 5. *Confusion matrix* Decision Tree

Berdasarkan *confusion matrix* pada Gambar 3, model Random Forest mampu mengklasifikasikan 961 URL legitimate dan 885 URL *phishing* secara benar, dengan jumlah kesalahan yang relatif seimbang antara *false positive* (196) dan *false negative* (244). Hasil ini menunjukkan bahwa Random Forest memiliki kemampuan yang stabil dalam membedakan kedua kelas, baik legitimate maupun *phishing*.

Gambar 5 memperlihatkan *confusion matrix* Decision Tree, di mana model berhasil mendeteksi 952 URL legitimate dan 825 URL *phishing* dengan benar. Meskipun performanya lebih baik dibandingkan Naive Bayes (Gambar 4), jumlah *false negative* pada Decision Tree masih lebih tinggi dibandingkan Random Forest. Dengan demikian, Random Forest dapat disimpulkan sebagai model dengan keseimbangan terbaik antara kemampuan deteksi dan stabilitas performa pada eksperimen ini.



Gambar 6. Perbandingan akurasi model Random Forest, Naive Bayes, dan Decision Tree

Gambar 6 menampilkan perbandingan akurasi ketiga model dalam bentuk diagram batang. Random Forest menunjukkan nilai akurasi tertinggi, diikuti Decision Tree, sedangkan Naive Bayes memiliki akurasi terendah. Visualisasi ini memperkuat kesimpulan bahwa Random Forest adalah pilihan paling stabil untuk mendeteksi link *phishing* dengan *feature extraction* URL sederhana. Model

ini dapat diimplementasikan untuk mendeteksi link *phishing* secara otomatis dan dapat dikembangkan lebih lanjut dengan penambahan fitur lanjutan, seperti validasi *SSL certificate* atau integrasi *blacklist database*, sehingga sistem deteksi dapat beradaptasi dengan pola serangan *phishing* yang semakin variatif.

4. KESIMPULAN

Berdasarkan hasil eksperimen, dapat disimpulkan bahwa tujuan eksperimen untuk menganalisis dan membandingkan performa tiga algoritma *machine learning*, yaitu Random Forest, Decision Tree, dan Naive Bayes dalam mendeteksi link *phishing* berbasis fitur URL sederhana telah tercapai. Hasil pengujian menunjukkan bahwa Random Forest memberikan akurasi tertinggi sebesar 80,75% dengan nilai *precision* dan *recall* yang seimbang, sedangkan Decision Tree memiliki akurasi 77,73% dan Naive Bayes hanya mencapai 68,15%. Temuan ini sesuai dengan harapan pada tahap awal eksperimen yang menginginkan algoritma dengan kemampuan deteksi *phishing* yang efektif, akurat, dan mudah diimplementasikan.

Hasil ini menunjukkan bahwa analisis fitur dasar URL dapat digunakan sebagai pendekatan praktis untuk mendeteksi tautan *phishing* secara otomatis. Ke depan, model yang dihasilkan dapat dikembangkan lebih lanjut dengan menambahkan fitur lanjutan seperti validasi *SSL certificate*, data *WHOIS*, pola *blacklist database*, serta penerapan pada data real-time untuk meningkatkan keakuratan dan adaptivitas deteksi *phishing*. Diharapkan eksperimen ini dapat menjadi referensi dan inspirasi bagi pengembangan metode deteksi *phishing* yang lebih komprehensif dan adaptif di masa mendatang.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada pihak yang telah memberikan dukungan fasilitas, pembimbingan, serta pendanaan dalam pelaksanaan eksperimen ini. Penulis juga berterima kasih kepada institusi dan semua pihak yang telah membantu dalam pengumpulan data, diskusi teknis, serta penyusunan artikel ini hingga selesai.

DAFTAR PUSTAKA

[1] K. S. Y. A. Lucky Nurhadiyanto, "Ancaman Risiko Keamanan Theft Identity Pada Aplikasi Berbasis Artificial Intelligence Dalam Perspektif Lifestyle Exposure Theory," *Ikra-Ith Hum. J. Sos. Dan Hum.*, Vol. 8, No. 2, Juli. 2024.
[2] "A Literature Review On Classification Of *Phishing* Attacks," *Int. J. Adv. Technol. Eng. Explor.*, Vol. 9, No. 89, Apr. 2022, Doi: 10.19101/Ijatee.2021.875031.
[3] Lukito And W. B. T. Handaya, "Deteksi Website *Phishing* Menggunakan Teknik Machine Learning," *J. Inform. Atma Jogja*, Vol. 6, No. 1, Art. No. 1, May 2025, Doi: 10.24002/Jiaj.V6i1.11538.
[4] A. F. Mahmud And S. Wirawan, "*Phishing* Website Detection Using Machine Learning Classification Method," *Sistemasi*, Vol. 13, No. 4, P. 1368, Jul. 2024, Doi: 10.32520/Stmsi.V13i4.3456.
[5] R. Fauzan, A. V. Vitianingsih, D. Cahyono, A. L. Maukar, And Y. A. B. Suprio, "Penerapan Algoritma Klasifikasi Pada Machine Learning Untuk Deteksi *Phishing*: Application Of Classification Algorithms In Machine Learning For *Phishing* Detection," *Malcom Indones. J. Mach. Learn. Comput. Sci.*, Vol. 5, No. 2, Pp. 531–540, Mar. 2025, Doi: 10.57152/Malcom.V5i2.1968.
[6] A. F. Mahmud And S. Wirawan, "*Phishing* Website Detection Using Machine Learning Classification Method," *Sistemasi*, Vol. 13, No. 4, P. 1368, Jul. 2024, Doi: 10.32520/Stmsi.V13i4.3456.
[7] D. Lusiyanti, S. Musdalifah, A. Sahari, and I. A. Fajri, "Evaluasi Kinerja Algoritma Machine learning pada Dataset Skala Besar," *MathVision J. Mat.*, vol. 7, no. 1, pp. 84–92, Mar. 2025, doi: 10.55719/mv.v7i1.1661
[8] A. Raihan, M. Fadhli, And L. Lindawati, "Implementation Of Deep Learning For Detecting *Phishing* Attacks On Websites With Combination Of Cnn And Lstm," *J. Tek. Inform. Jutif*, Vol. 5, No. 5, Pp. 1451–1459, Oct. 2024, Doi: 10.52436/1.Jutif.2024.5.5.2446.
[9] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, And A. Setiawan, "Deteksi Website *Phishing* Menggunakan Teknik Filter Pada Model Machine Learning," *Inf. Syst. J.*, Vol. 6, No. 01, Art. No. 01, Aug. 2023, Doi: 10.24076/Infosjournal.2023v6i01.1268.
[10] M. A. Sembiring, H. Saputra, R. A. Yusda, S. Sutarman, And E. B. Nababan, "Performance Of Robust Support Vector Machine Classification Model On Balanced, Imbalanced And Outliers *Dataset*s," *Jitk J. Ilmu Pengetah. Dan Teknol. Komput.*, Vol. 10, No. 1, Pp. 208–215, Aug. 2024, Doi: 10.33480/Jitk.V10i1.5272.