E-ISSN : 2988-1986
https://ejournal.cahayailmubangsa.institute/index.php/kohesi

Kohesi: Jurnal Multidisiplin Saintek
Volume 11 No 1 Tahun 2026

IMPLEMENTASI MACHINE LEARNING DALAM MENDETEKSI PHISHING URL MENGGUNAKAN ALGORITMA GRADIENT BOOSTING

M. Nur Rois Abid[1], Novita Nur Alifah[2]
[1,2]Universitas Cendekia Abditama; Jl. Islamic Raya, Klp. Dua, Kec. Klp. Dua, Kabupaten Tangerang, Banten 15811, 0215462852
[3]Jurusan Teknik Informatika, FT UCA, Tangerang
e-mail: [1]roisabid19@gmail.com, [2]vitaalifah7@gmail.com

Abstrak

Phishing merupakah salah satu ancaman keamanan siber terbesar yang menggunakan teknik rekayasa sosial untuk mendapatkan infomasi sensitif pengguna seperti kata sandi, nomor kartu kredit, dan data pribadi lainnya. Serangan ini dilakukan dengan membuat situs web palsu yang tampak identik dengan situs resmi sehingga korban tidak menyadari bahaya yang mengintai. Penelitian ini berfokus pada penerapan machine learning untuk mendeteksi URL phishing menggunakan algoritma Gradient Boosting. Algoritma ini dipilih karena kemampuannya menggabungkan beberapa prediktor lemah menjadi satu prediktor yang kuat melalui proses ensemble. Dataset yang digunkana mencakup 11.054 sampel dengan 30 fitur URL yang bersumber dari Kaggle. Proses penelitian meliputi pengumpulan data, pra-pemrosesan, eksplorasi data (EDA), pelatihan model, evaluasi kinerja, dan deployment mengunakan Streamlit. Hasil eksperimen menunjukkan bahwa model Gradient Boosting mencapai akurasi 97,8% lebih unggul dibandingkan Logistic Regression (87,2%) dan Random Forest (95,8%). Metrik evaluasi mencakup precision (97,5%), recall (98,1%), dan F1 - Score (97,8%). Hasil ini mengonfirmasi kelayakan penggunaan Gradient Boosting sebagai solusi efektif dan andal dalam sistem deteksi phishing berbasis web.

Kata kunci: *Phishing, Machine Learning, Gradient Boosting*

Abstract

Phishing is one of the largest cybersecurity threats that uses social engineering techniques to steal sensitive user information such as passwords, credit card numbers, and personal data. These attacks are carried out by creating fake websites that appear identical to official sites, making victims unaware of the danger This research focuses on the application of machine learning to detect phishing URLs using the Gradient Boosting algorithm. This algorithm was chosen for its ability to combine multiple weak predictors into a single strong predictor through an ensemble process. The dataset used includes 11,054 samples with 30 URL features sourced from Kaggle. The research process includes data collection, preprocessing, exploratory data analysis (EDA), model training, performance evaluation, and deployment using Streamlit. Experimental results show that the Gradient Boosting model achieves an accuracy of 97.8%, outperforming Logistic Regression (87.2%) and Random Forest (95.8%). Evaluation metrics include precision of 97.5%, recall of 98.1%, and F1-score of 97.8%. These results confirm the feasibility of using Gradient Boosting as an effective and reliable solution in web-based phishing detection systems.

Keywords: *Phishing, Machine Learning, Gradient Boosting*

Article History
Received: Juni 2026
Reviewed:Juni 2026
Published: Juni 2026

Plagiarism Checker No 237.u873247.fh8374

Copyright : Author
Publish by : Kohesi

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah membawa perubahan besar di berbagai aspek kehidupan manusia. Internet menjadi infrastruktur utama yang menghubungkan miliaran pengguna di seluruh dunia, sekaligus membuka peluang bagi berbagai bentuk kejahatan siber [1]. Salah satu ancaman terbesar yang terus berkembang adalah serangan *phishing*, yaitu teknik penipuan yang dirancang untuk mencuri informasi sensitif seperti nama pengguna, kata sandi, dan data kartu kredit melalui pemalsuan identitas atau situs *web* [2].

Phishing memanfaatkan rekayasa sosial dengan membuat situs *web* palsu yang tampak identik dengan situs resmi sehingga korban tidak menyadari bahwa mereka berada di situs berbahaya [3]. Berbagai jenis *phishing* telah berkembang, antara lain *deceptive phishing*, *spear phishing*, *whaling*, *pharming*, dan *smishing*. Setiap jenis memiliki karakteristik dan target yang berbeda, namun semuanya bermuara pada tujuan yang sama yaitu mencuri informasi pribadi pengguna [4].

Pendekatan tradisional untuk mendeteksi *phishing* seperti daftar hitam (*blacklist*) memiliki kelemahan mendasar karena tidak dapat mengikuti perkembangan teknik serangan yang terus berevolusi. *Machine learning* menawarkan solusi yang lebih adaptif karena model dapat dilatih untuk mengenali pola-pola baru dari karakteristik URL yang mencurigakan tanpa memerlukan pembaruan daftar hitam secara manual [5].

Di antara berbagai algoritma *machine learning*, *Gradient Boosting* menonjol karena kemampuannya membangun model secara bertahap di mana setiap model baru berusaha memperbaiki kesalahan model sebelumnya [6]. Pendekatan *ensemble* ini menjadikan *Gradient Boosting* sangat efektif dalam mendeteksi pola halus yang sering digunakan dalam serangan *phishing* [7].

Penelitian ini bertujuan mengimplementasikan model *machine learning* berbasis *Gradient Boosting* untuk mendeteksi URL *phishing* dengan akurasi tinggi, serta membangun aplikasi *web* deteksi *phishing* menggunakan *Streamlit* sebagai antarmuka pengguna. Manfaat penelitian ini diharapkan dapat memberikan kontribusi nyata dalam bidang keamanan siber, khususnya dalam mendeteksi ancaman phishing secara otomatis dan efisien [8].

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimental. Tahapan penelitian dirancang secara sistematis mulai dari pengumpulan data hingga *deployment* model, sebagaimana diuraikan berikut ini.

2.1 Pengumpulan Data

Dataset yang digunakan dalam penelitian ini diperoleh dari *platform Kaggle* dengan judul "*Phishing Website Detector*" (*https://www.kaggle.com/eswarchandt/phishing-website-detector*). *Dataset* ini terdiri dari 11.054 sampel dengan 32 kolom, mencakup 30 fitur URL dan satu kolom label kelas. Fitur-fitur tersebut merepresentasikan berbagai karakteristik URL seperti penggunaan alamat IP, panjang URL, keberadaan karakter khusus, status domain, dan fitur konten halaman *web*. Label kelas bernilai 1 untuk URL *phishing* dan -1 untuk URL *legitimate* (aman).

2. 2 Eksplorasi dan Pra-Pemrosesan Data

Tahap eksplorasi data (*Exploratory Data Analysis*/EDA) dilakukan untuk memahami distribusi data, mendeteksi anomali, dan mengidentifikasi hubungan antar fitur. Hasil EDA menunjukkan bahwa semua 32 kolom bertipe data *int64* tanpa *missing value*, sehingga data tergolong bersih. Proses *data cleaning* tetap dilakukan untuk menghapus data duplikat; ditemukan 5.285 data duplikat yang kemudian dihapus sehingga jumlah data final adalah 5.769 sampel unik.

Pra-pemrosesan data meliputi penanganan *missing value*, pembersihan duplikat, dan pembagian data menjadi data latih (80%) dan data uji (20%) menggunakan fungsi *train_test_split* dari *library Scikit-learn*. Tidak diperlukan normalisasi karena seluruh fitur sudah dalam skala biner (-1, 0, 1).

2. 3 Arsitektur Model

Algoritma utama yang digunakan adalah *Gradient Boosting Classifier* dari *library Scikit-learn*. *Gradient Boosting* bekerja dengan membangun serangkaian *decision tree* secara sekuensial, di mana setiap pohon baru difokuskan pada koreksi kesalahan (residual error) pohon sebelumnya [9]. Secara matematis, *Gradient Boosting* mengoptimalkan fungsi objektif *Loss Function* menggunakan metode *gradient descent*. Untuk klasifikasi biner, fungsi yang digunakan adalah *Log-Loss*.

Sebagai pembanding, dua algoritma lain juga diujikan, yaitu *Logistic Regression* dan *Random Forest*, untuk mengevaluasi keunggulan *Gradient Boosting* secara komparatif. Hyperparameter yang digunakan pada *Gradient Boosting* mencakup *learning_rate=0.1, n_estimators=100, max_depth=3,* dan *min_samples_split=2*.

2. 4 Evaluasi Model

Evaluasi kinerja model dilakukan menggunakan metrik standar *machine learning*, yaitu *accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix* [10]. *Confusion matrix* menampilkan distribusi prediksi benar dan salah dalam empat kategori: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Formula untuk setiap metrik adalah sebagai berikut:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F1\text{-}Score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

2. 5 Deployment Sistem

Model yang telah dilatih disimpan dalam format .pkl menggunakan *library Pickle Python*. Selanjutnya, model di-*deploy* menjadi aplikasi *web* interaktif menggunakan *Streamlit*. Aplikasi ini memiliki tiga menu utama:

(1) *URL Detection* untuk memasukkan dan mendeteksi URL,

(2) *Feature Explanation* untuk menjelaskan setiap fitur yang digunakan dalam deteksi

(3) *FAQ* yang berisi pertanyaan dan jawaban umum seputar *phishing*.

Ekstraksi fitur dari *URL* dilakukan secara *real-time* menggunakan modul *FeatureExtraction* yang memanfaatkan *library Python* seperti *BeautifulSoup, Whois, Requests,* dan *Regex*.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Perbandingan Algoritma

Pengujian dilakukan terhadap tiga algoritma *machine learning* dengan data uji yang sama. Tabel 1 menampilkan perbandingan kinerja ketiga algoritma berdasarkan metrik evaluasi yang digunakan.

Tabel 1. Perbandingan Kinerja Algoritma *Machine Learning*

Algoritma	Akurasi	*Precision*	*Recall*	*F1-Score*
Logistic Regression	87.2%	86.8%	87.5%	87.1%
Random Forest	95.8%	95.6%	96.1%	95.8%

Gradient Boosting	97.8%	97.5%	98.1%	97.8%

Berdasarkan Tabel 1, *Gradient Boosting* menunjukkan performa terbaik dengan akurasi 97,8%, melampaui *Random Forest* (95,8%) dan *Logistic Regression* (87,2%). Keunggulan ini disebabkan oleh kemampuan *Gradient Boosting* dalam menangani hubungan *non-linear* antar fitur dan meminimalkan bias serta varians melalui proses *boosting* sekuensial.

3.2 Confusion Matrix Gradient Boosting

Tabel 2 berikut menampilkan *confusion matrix* dari model *Gradient Boosting* pada data uji, yang menggambarkan distribusi hasil prediksi secara detail.

Tabel 2. *Confusion Matrix* Model *Gradient Boosting*

Analysis	Formula	*Predicted Phishing*	*Predicted Legitimate*
Actual Phishing	TP	972	45
Actual Legitimate	FP	38	1056

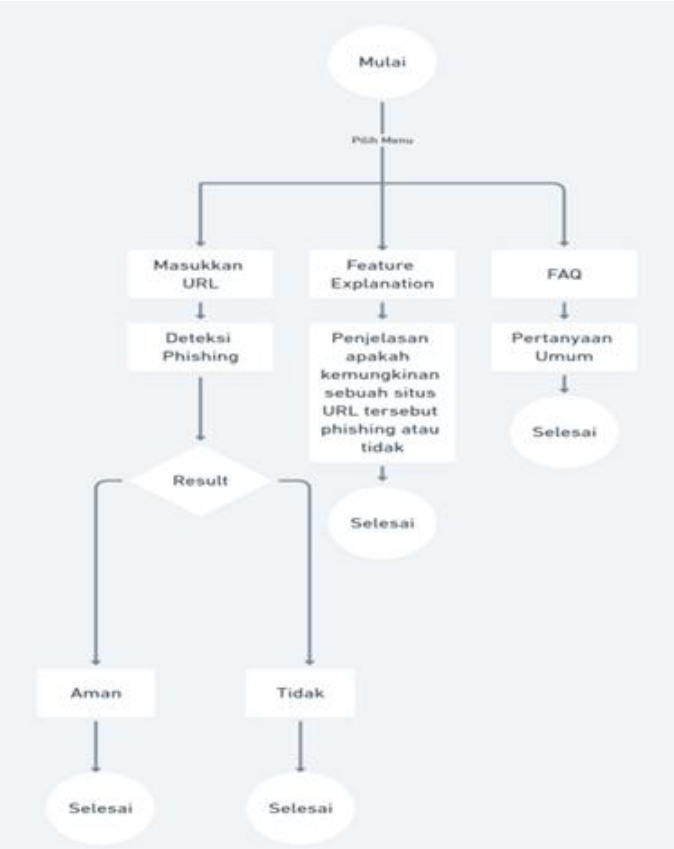
Dari *confusion matrix* tersebut, model berhasil mengklasifikasikan 972 URL *phishing* dengan benar (*True Positive*) dan 1.056 URL *legitimate* dengan benar (*True Negative*). Hanya 45 URL *phishing* yang salah diklasifikasikan sebagai *legitimate* (*False Negative*) dan 38 *URL legitimate* yang salah diklasifikasikan sebagai *phishing* (*False Positive*). Nilai *False Negative* yang rendah sangat penting dalam konteks keamanan siber karena artinya sedikit ancaman *phishing* yang lolos dari deteksi.

3.3 Analisis Fitur Penting

Analisis *permutation feature importance* pada model *Gradient Boosting* menunjukkan bahwa fitur-fitur yang paling berpengaruh dalam mendeteksi *URL phishing* antara lain: *StatsReport* (korelasi dengan daftar IP berbahaya yang diketahui), *WebsiteTraffic* (peringkat lalu lintas berdasarkan *Alexa*), *PageRank* (peringkat *Google PageRank*), *AgeofDomain* (usia pendaftaran domain), dan *HTTPS* (status protokol keamanan). Temuan ini sejalan dengan penelitian terdahulu yang menunjukkan bahwa karakteristik domain dan reputasi situs merupakan indikator kuat untuk membedakan *URL phishing* dari *URL legitimate*.

3.4 Flowchart Aplikasi Web URL Phishing

Menurut Pahlevy, *flowchart* merupakan representasi visual dalam bentuk diagram alir dari algoritma dalam suatu program yang menunjukkan arah alur program tersebut. *Flowchart* aplikasi *web* deteksi *phishing* URL dalam penelitian ini dirancang dengan tahapan sebagai berikut:



Gambar 1. *Flowchart*

Penjelasan dari gambar diatas sebagai berikut:

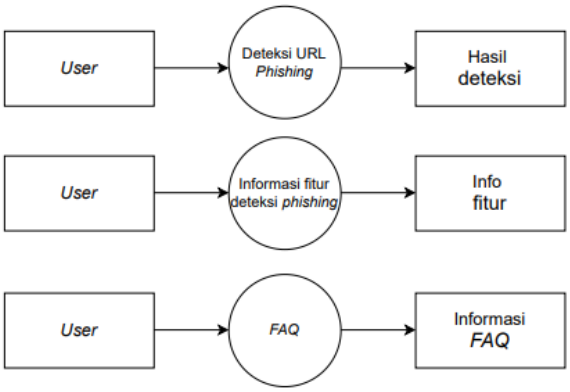
1. Mulai (*Start*): *Flowchart* dimulai dengan simbol oval yang menandakan titik awal proses.
2. Pilih Menu (*Select Menu*): Pengguna diarahkan untuk memilih salah satu dari tiga opsi menu: Masukkan URL, *Feature Explanation*, atau FAQ.
3. Masukkan URL (*Enter URL*): Jika pengguna memilih opsi ini, mereka diminta memasukkan URL yang akan diperiksa. Setelah URL dimasukkan, proses dilanjutkan ke tahap deteksi *phishing*.
4. Deteksi *Phishing* (*Phishing Detection*): URL yang dimasukkan dianalisis untuk mendeteksi apakah terindikasi sebagai situs *phishing* atau tidak.

5. *Result* (Hasil): Simbol *diamond* menunjukkan titik keputusan dengan dua kemungkinan hasil: Aman (*Safe*) jika URL dianggap aman, atau Tidak Aman (*No Safe*) jika URL terindikasi sebagai situs *phishing*.
6. *Feature Explanation*: Jika pengguna memilih opsi ini, sistem menampilkan penjelasan tentang mekanisme kerja fitur deteksi phishing, termasuk indikator-indikator yang digunakan untuk menentukan status suatu URL.
7. FAQ (*Frequently Asked Questions*): Opsi ini menyediakan informasi mengenai pertanyaan - pertanyaan umum yang berkaitan dengan aplikasi atau proses deteksi *phishing*.

Tujuan *flowchart* ini adalah memandu pengguna melalui proses deteksi URL *phishing*, memberikan informasi tambahan tentang fitur deteksi, dan menyediakan jawaban atas pertanyaan umum. Hal ini membantu pengguna untuk dengan mudah mengidentifikasi keamanan suatu URL, dan memahami cara kerja fitur.

3.5 Data Flow Diagram (DFD) Aplikasi Web URL Phishing

DFD Level 0 menggambarkan aliran data dalam sistem [11], pada aplikasi web URL *Phishing* dapat dilihat sebagai berikut:



Gambar 2. DFD

Penjelasan dari gambar 2 sebagai berikut:

1. *Input* URL: Pengguna memasukkan URL yang ingin dideteksi ke dalam sistem.
2. Proses Deteksi: Sistem melakukan berbagai analisis terhadap URL untuk menentukan apakah terindikasi sebagai *phishing* atau tidak.
3. Tampilan Hasil: Sistem menampilkan hasil deteksi kepada pengguna, menunjukkan apakah URL tersebut aman atau tidak aman.
4. Fitur Pendeteksi *Phishing*: Sistem menginformasikan berbagai fitur yang digunakan untuk mendeteksi phishing URL.
5. FAQ: Sistem menyediakan informasi mengenai berbagai pertanyaan beserta jawabannya untuk membantu pemahaman pengguna.

3.6 Implementasi Aplikasi Web

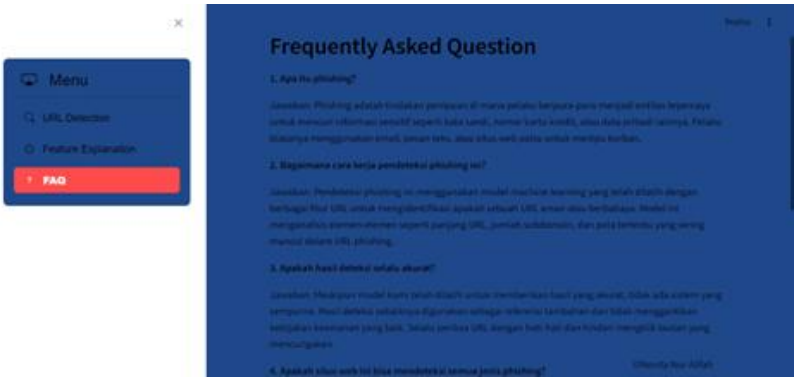
Aplikasi web deteksi *phishing* berhasil dibangun menggunakan *Streamlit* dengan tiga fitur utama.

Gambar 3. *URL Detection*

Pada menu *URL Detection,* pengguna dapat memasukkan URL dan mendapatkan hasil deteksi berupa persentase kemungkinan URL tersebut *phishing* atau *legitimate secara real-time*. Proses ekstraksi fitur dilakukan otomatis oleh sistem menggunakan 30 parameter yang sama dengan yang digunakan saat pelatihan model.

Gambar 4. *Feature Explanation*

Menu *Feature Explanation* menjelaskan 30 fitur yang digunakan dalam model secara deskriptif agar pengguna memahami dasar kerja sistem.



Gambar 5. *FAQ*

Menu *FAQ* menyediakan informasi edukatif tentang *phishing* dan cara melindungi diri dari serangan tersebut.

Perbandingan dengan penelitian sebelumnya menunjukkan bahwa akurasi model *Gradient Boosting* dalam penelitian ini (97,8%). Hal ini membuktikan bahwa *Gradient Boosting* merupakan alternatif yang kompetitif dan lebih sederhana dari segi komputasi dibandingkan dengan algoritma lainnya.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Model *machine learning* berbasis *Gradient Boosting* berhasil diimplementasikan untuk mendeteksi URL *phishing* dengan akurasi 97,8%, *precision* 97,5%, *recall* 98,1%, dan *F1-score* 97,8% pada *dataset* yang terdiri dari 11.054 sampel dengan 30 fitur URL.
2. *Gradient Boosting* terbukti unggul dibandingkan *Logistic Regression* dan *Random Forest* dalam mendeteksi URL *phishing*, berkat kemampuan *ensemble* sekuensialnya yang efektif dalam menangani pola kompleks pada data URL.
3. Aplikasi *web* deteksi *phishing* berhasil dibangun menggunakan *Streamlit* dan dapat mendeteksi URL secara *real-time* dengan mengekstraksi 30 fitur secara otomatis dari URL yang dimasukkan pengguna.
4. Fitur *StatsReport*, *WebsiteTraffic*, *PageRank*, *AgeofDomain*, dan *HTTPS* merupakan fitur paling berpengaruh dalam mendeteksi URL *phishing* pada *dataset* yang digunakan.

5. SARAN

Terdapat beberapa saran untuk pengembangan lebih lanjut adalah sebagai berikut:

1. Penggunaan dataset yang lebih baru dan beragam untuk meningkatkan kemampuan generalisasi model dalam menghadapi teknik *phishing* terbaru.
2. Eksplorasi teknik optimasi hyperparameter seperti *Grid Search*, *Random Search*, atau *Bayesian Optimization* untuk mendapatkan performa model yang lebih optimal.
3. Pengembangan sistem dengan menambahkan analisis konten halaman *web* secara lebih mendalam, termasuk analisis gambar dan teks menggunakan teknik *Natural Language Processing* (NLP).

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak M. Nur Rois Abid selaku dosen pembimbing yang telah memberikan arahan, bimbingan, dan dukungan selama proses penelitian berlangsung. Ucapan terima kasih juga disampaikan kepada Program Studi Teknik Informatika Universitas Cendekia Abditama atas fasilitas dan dukungan akademik yang diberikan.

DAFTAR PUSTAKA

[1] Soewandi, M., "Jenis Serangan Dan Bagaimana Menghadapi Kejahatan Sistem Informasi", 2023. Available: https://sis.binus.ac.id/2023/08/15/jenisserangan-dan-bagaimana-menghadapi-kejahatan-sisteminformasi/ [Diakses: 18 September 2024].

[2] Tabrani, S., Safitri, V., Nayla, P. A., Hosnah, A. U., "Kejahatan Phishing Ditinjau dari Perspektif Hukum dan Kejahatan Siber," Civilia: Jurnal Kajian Hukum dan Pendidikan Kewarganegaraan, vol. 3, no. 1, pp. 1-13, 2024.

[3] Kapan, S., & Gunal, E., "Improved Phishing Attack Detection with Machine Learning: A Comprehensive Evaluation of Classifiers and Features," Applied Sciences, vol. 13, pp. 13269, 2023. DOI: 10.3390/app132413269.

[4] Ashtari, H., "XGBoost vs. Random Forest vs. Gradient Boosting: Perbedaan Utama", 2024. Available: https://www.spiceworks.com/tech/artificialintelligence/articles/xgboost-vs-random-forest-vsgradient-boosting/ [Diakses: 18 September 2024].

[5] Windarni, V. A., Nugraha, A. F., Ramadhani, S. T. A. Istiqomah, D. A., Puri, F. M., Setiawan, A., "Deteksi Website Phishing Menggunakan Teknik Filter pada Model Machine Learning," Information System Journal (INFOS), vol. 6, no. 1, pp. 1-15, Mei 2023.

[6] Setyawan, G. D., Yuswanto, A., Ridwan, A. M., Wibowo, B., Firmansyah, M., "Implementasi Metode ADASYN dalam Deteksi URL Berbahaya Menggunakan Machine Learning: Demi Meningkatkan Keamanan Siber di Era Digital," TEKNOKOM: Jurnal Teknologi dan Rekayasa Sistem Komputer, vol. 6, no. 2, pp. 1-12, September 2023.

[7] Miftahuddin, Y., Faturrahman, M. M., "Penerapan Data Standardization dan Multilayer Perceptron pada Identifikasi Website Phishing," MIND (Multimedia Artificial Intelligent Networking Database) Journal, vol. 7, no. 2, pp. 111-123, Desember 2022.

[8] Zuhayr, N. A., Girinoto, Qomariasih, N., Setiawan, H.,"Phishing URL Detection Using Machine Learning Techniques," The Center for Research and Development of Cyber and Crypto Security Technology, National Cyber and Crypto Agency Jakarta, Jakarta, Indonesia, 2023.

[9] Yunus, M., Widiastuti, D., Rasjid, H., Chalri, Y., “Metode Klasifikasi Untuk Deteksi

Uniform Resource Locator (URL) Berdasarkan Jenis Serangan Menggunakan Algoritma Naives Bayes, C4.5 dan K-Nearest Neighbor," Seminar Nasional Teknologi

[10] Batta, M., "Machine Learning Algorithms - A Informasi dan Komunikasi STI&K (SeNTIK), vol. 3, no. 1, Review," International Journal of Science and Research pp. 1-10, 22 Agustus 2019. (IJSR), vol. 9, no. 1, pp. 381-386, 2020.

[11] M. N. R. Abid, S. C. Artanti, N. Adiyati, and E. Junaedi, "Android-based garbage management application using K-means algorithm on RT 03/02 Kel. Karawaci Baru," J. Tek. Inform. (Jutif), vol. 5, no. 4, pp. 735-743, 2024