Perbandingan Algoritma *Random Forest* Dan *Gradient Boosting* Dalam Mendeteksi Link Berbahaya

[1st]Idfian Azhar Hidayat
S1 Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
idfianazharhidayat@student.telkomuniversity.ac.id

[2nd]Gita Fadila Fitriana
S1 Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
gitafadila@telkomuniversity.ac.id

[3nd]Yudha Islami Sulistya
S1 Rekayasa Perangkat Lunak
Universitas Telkom Purwokerto
Purwokerto, Indonesia
yudhaislami@telkomuniversity.ac.id

***Abstrak* — Serangan *phishing* berbasis *URL* masih menjadi ancaman siber yang signifikan karena pelaku dapat dengan mudah membuat alamat web palsu yang menyerupai situs resmi. Pendekatan deteksi tradisional seperti *blacklist* memiliki keterbatasan dalam mengenali *URL* baru, sehingga dibutuhkan metode yang lebih adaptif. Penelitian ini bertujuan mengembangkan sistem deteksi *URL phishing* berbasis *machine learning* serta membandingkan kinerja dua algoritma *ensemble*, yaitu *Random Forest* dan *Gradient Boosting*. Dataset yang digunakan merupakan dataset *sekunder* dari *Kaggle* yang berisi sekitar 450,000 *URL* dengan label *benign* dan *phishing*. Data diproses melalui beberapa tahapan, yaitu *preprocessing*, ekstraksi fitur berbasis struktur *URL*, pembagian data latih dan data uji, serta penyeimbangan distribusi kelas menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE). Model dievaluasi menggunakan *accuracy, precision, recall, F1-score*, waktu pelatihan, dan waktu *inference*. Hasil pengujian menunjukkan bahwa kedua algoritma memiliki performa yang baik dengan tingkat akurasi di atas 98%. *Gradient Boosting* menghasilkan *F1-score* tertinggi, yaitu sekitar 97,5%, sedangkan *Random Forest* waktu pelatihannya lebih singkat. Pada pengujian tanpa *SMOTE*, *Random Forest* justru menunjukkan hasil yang sedikit lebih baik dengan *F1-score* 97,79% dan *recall* 96,41%. Model kemudian diterapkan ke dalam *prototipe* sistem berbasis web menggunakan *Streamlit*. Sistem memungkinkan untuk mengunggah dataset, melatih model, dan melakukan prediksi *URL*. Hasil penelitian menunjukkan bahwa pendekatan *machine learning* berbasis fitur struktur *URL* mampu digunakan secara efektif untuk mendeteksi *URL phishing,* dan *Random Forest* dipilih sebagai model terbaik dalam memprediksi *URL phishing* berdasarkan akurasi dan kemampuan deteksi yang lebih optimal.*

Kata kunci*— *phishing, machine learning, Random Forest, Gradient Boosting*, deteksi *URL*, *SMOTE

I. PENDAHULUAN

Serangan *phishing* berbasis *URL* masih menjadi salah satu ancaman siber yang dominan di era digital saat ini. Pelaku kejahatan siber dapat dengan mudah membuat *URL* palsu yang menyerupai situs resmi sehingga sulit dikenali oleh pengguna awam [1], [2]. Laporan *Anti-Phishing Working Group* (APWG) periode 2023–2025 menunjukkan bahwa tren serangan phishing berbasis situs masih berada pada tingkat tinggi dan relatif konsisten, Berdasarkan data *APWG*, hingga kuartal III tahun 2025 tercatat sebanyak 3,026,811 serangan *phishing* berbasis situs unik. Pada kuartal pertama tahun 2025 jumlah serangan mencapai 1,003,924, yang termasuk salah satu angka tertinggi sejak akhir 2023. Angka tersebut meningkat pada kuartal kedua menjadi 1,130,393 serangan, lalu menurun pada kuartal ketiga menjadi 892,494 serangan.

Meskipun terjadi fluktuasi antar kuartal, jumlah serangan tetap berada pada level yang sangat tinggi, menunjukkan bahwa phishing masih merupakan ancaman yang berkelanjutan dan terus berkembang [1]. Metode deteksi konvensional seperti *blacklist* dan *whitelist* semakin kurang efektif karena *URL* berbahaya dapat muncul dengan cepat dan terus berubah mengikuti pola serangan baru. Pendekatan berbasis daftar statis umumnya lambat dalam merespons ancaman terbaru, sehingga tidak mampu memberikan perlindungan optimal terhadap serangan phishing yang dinamis. Dengan berkembangnya teknologi, berbagai pendekatan berbasis *machine learning* dan *deep learning* telah dikembangkan untuk mendeteksi *URL phishing* secara otomatis. Namun demikian, sejumlah penelitian menunjukkan bahwa performa model dapat menurun seiring waktu akibat sifat *phishing* yang adaptif. Perubahan pola serangan menyebabkan model yang dilatih menggunakan data lama menjadi kurang relevan ketika dihadapkan pada data baru [3], [4].

Hal ini menegaskan perlunya metode deteksi yang tidak hanya memiliki tingkat akurasi tinggi, tetapi juga adaptif terhadap perubahan pola serangan serta mampu bekerja secara *real-time* guna meminimalkan risiko bagi pengguna maupun organisasi [3]. Pendekatan deteksi yang ada saat ini masih memiliki keterbatasan, sehingga diperlukan penelitian lanjutan untuk mengembangkan metode yang lebih efektif. Oleh karena itu, penelitian ini mengusulkan penggunaan algoritma machine learning berbasis *ensemble*, yaitu *Random Forest* dan *Gradient Boosting*, untuk mendeteksi *URL phishing. Random Forest* dipilih karena menggabungkan banyak pohon keputusan dalam satu model, sehingga memiliki stabilitas tinggi, tahan terhadap noise, dan mampu

mengurangi risiko *overfitting* pada data berbasis fitur *URL* [5], [6]. Di sisi lain, *Gradient Boosting* memiliki keunggulan dalam meningkatkan akurasi melalui proses pembelajaran bertahap, di mana setiap model baru dirancang untuk memperbaiki kesalahan model sebelumnya. Pendekatan ini efektif dalam menangkap pola *phishing* yang kompleks dan terus berkembang [2], [7].

Penelitian terdahulu menunjukkan bahwa *Random Forest* mampu mencapai akurasi hingga 92% dalam klasifikasi *URL phishing* [2], sementara *Gradient Boosting Classifier* dilaporkan dapat mencapai akurasi hingga 98% pada dataset tertentu [3]. Berdasarkan latar belakang tersebut, penelitian ini bertujuan mengembangkan sistem deteksi *URL phishing* berbasis *machine learning* dengan membandingkan performa algoritma *Random Forest* dan *Gradient Boosting*. Evaluasi dilakukan menggunakan metrik *accuracy, precision, recall, F1-score* untuk menentukan model yang paling efektif dalam mendeteksi *URL phishing* secara akurat dan andal.

II. KAJIAN TEORI

A. Keamanan Informasi dan Serangan *Phishing*

Keamanan informasi merupakan upaya melindungi data dari ancaman yang dapat mengganggu kerahasiaan, keutuhan, dan ketersediaan informasi. Tujuannya adalah memastikan data hanya dapat diakses oleh pihak yang berwenang, menjaga data dari manipulasi yang tidak sah, serta tetap tersedia saat dibutuhkan [10]. Salah satu ancaman yang sering terjadi adalah *phishing. Phishing* adalah bentuk serangan *social engineering* yang bertujuan memperoleh informasi sensitif seperti *username*, kata sandi, data perbankan, dan data pribadi lainnya dengan cara menyamar sebagai situs resmi [11].

Korban umumnya tertipu karena tampilan situs dibuat menyerupai aslinya. Pada layanan berbasis *web, phishing* sering dilakukan melalui *link* yang meniru alamat resmi. Pelaku membuat *URL* palsu yang mirip dengan domain asli sehingga sulit dikenali pengguna. Kondisi ini dimanfaatkan untuk mencuri data melalui halaman login palsu. Salah satu metode deteksi yang umum digunakan adalah *blacklist* dan *whitelist*, yaitu membandingkan *URL* yang diakses dengan daftar *URL* berbahaya yang sudah diketahui sebelumnya [2], [12]. Metode ini cukup efektif untuk ancaman yang sudah terdaftar, namun kurang mampu dalam mendeteksi *URL* baru yang terus bermunculan. Karena keterbatasan itu, pendekatan machine learning digunakan. Dengan memanfaatkan ekstraksi fitur dari *URL*, model dapat mengklasifikasikan apakah suatu link termasuk aman atau berpotensi *phishing*.

B. Ekstraksi Fitur dari *URL*

Ekstraksi fitur untuk mengubah data mentah menjadi numerik agar dapat diproses oleh algoritma *machine learning*. Pada deteksi *phishing*, proses ini dilakukan dengan mengambil karakteristik dari struktur *URL* agar dapat membedakan antara *URL* aman dan *URL* berbahaya. *URL* dianalisis melalui proses parsing, memisahkan bagian penting seperti *protokol, domain,* dan *path* yang digunakan sebagai input model klasifikasi [13]. Beberapa fitur yang digunakan seperti panjang *URL* (*url_length*), jumlah titik (*nb_dots*), jumlah tanda hubung (*nb_hyphens*), indikator *HTTPS* (*isHttps*), kata sensitif (*sensitive_words*), dan panjang path (*path_length*). Setelah itu, setiap *URL* diubah menjadi numerik yang kemudian digunakan sebagai input model machine learning [11], [14].

C. SMOTE

Ketidakseimbangan kelas (*class imbalance*) terjadi ketika jumlah data pada satu kelas jauh lebih banyak dibanding kelas lainnya. Untuk mengatasi masalah, penelitian ini menggunakan metode *Synthetic Minority Over-sampling Technique* (SMOTE). SMOTE adalah teknik *oversampling* yaitu menambah data pada kelas yang lebih sedikit atau minoritas dengan membuat data *sintetis* berdasarkan data yang sudah ada [15]. Proses dilakukan dengan memilih satu data minoritas, lalu mencari tetangga terdekatnya (*k-nearest neighbors*). Dengan cara ini, jumlah data minoritas menjadi lebih seimbang dan model dapat mempelajari pola dari kedua data kelas dengan lebih baik [16].

D. *Machine Learning* dalam Klasifikasi *URL*

Machine learning adalah metode yang memungkinkan sistem mempelajari pola dari data untuk menghasilkan prediksi secara otomatis. Pendekatan ini banyak digunakan, termasuk dalam keamanan informasi. Dalam klasifikasi *URL, machine learning* digunakan untuk menentukan apakah suatu *URL* termasuk aman (*benign*) atau berbahaya (*malicious*). Model mempelajari fitur *URL* sebagai *input* dan label kelas sebagai *output*. Proses klasifikasi *URL* meliputi, *preprocessing* data, ekstraksi fitur dari struktur *URL*, pembagian data menjadi data latih dan data uji, pelatihan model, dan evaluasi performa menggunakan dataset [11], [9].

E. Random Forest

Random Forest adalah algoritma *ensemble learning* yang menggabungkan banyak *decision tree*. Metode ini mampu memberikan hasil yang lebih stabil dibandingkan satu pohon tunggal. Cara kerja membuat beberapa pohon keputusan dari *dataset* yang dipilih secara acak (*bootstrap sampling*). Pada setiap percabangan, model memilih secara acak, teknik ini membantu mengurangi *overfitting* untuk meningkatkan kemampuan [17], [8]. Pada prediksi, hasil dari seluruh *tree* digabungkan menggunakan metode *majority voting*, kelas yang paling banyak dipilih menjadi hasil akhir. *Random Forest* memiliki kelebihan, seperti dapat menangani data dengan banyak fitur, dapat tahan terhadap *noise*, dan memiliki performa yang baik diklasifikasi.

F. Gradient Boosting

Gradient Boosting merupakan algoritma *ensemble learning* yang menggunakan teknik *boosting*. Model dibangun secara bertahap, di mana setiap model baru memperbaiki kesalahan model sebelumnya. Jika *Random Forest* secara *paralel*, *Gradient Boosting* bekerja secara berurutan. Setiap *iterasi* menambahkan model lemah (*weak learner*) [18], [19]. Pendekatan *boosting* ini dapat menghasilkan akurasi yang baik karena mampu mempelajari pola dengan memperbaiki kesalahan model sebelumnya.

G. Metrik Evaluasi

Metrik evaluasi digunakan untuk menilai seberapa baik model klasifikasi dalam memprediksi *dataset*. Pada penelitian ini digunakan empat metrik [20], [21] yaitu, *accuracy* persentase prediksi yang benar dari seluruh data,

precision mengukur seberapa tepat model saat memprediksi kelas *positif* (*phishing*), *recall* mengukur kemampuan model menemukan data *phishing*, dan *F1-score* merupakan gabungan antara *precision* dan *recall*. Dengan metrik evaluasi performa model dapat dianalisis secara lebih menyeluruh tidak hanya bergantung pada satu nilai.

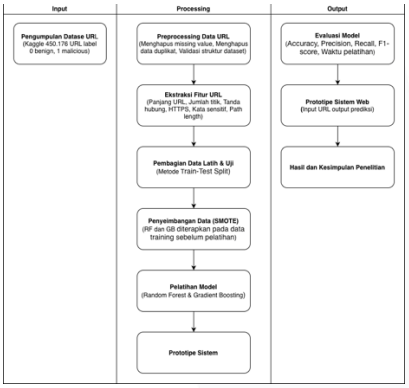
III. METODE

A. Subjek dan Objek Penelitian

Subjek Penelitian ini berupa *dataset URL* yang diperoleh dari *Kaggle* dengan judul *Malicious and Benign URLs* yang dipublikasikan oleh *Siddharth Kumar*. Dataset ini berisi sekitar *450,176 URL* yang telah diberi label *benign* (aman) dan *malicious* (berbahaya). Data memiliki struktur *URL*, *protokol, domain, subdomain, path,* dan *parameter query*. *Dataset* digunakan sebagai sumber data utama pada tahap *preprocessing*, ekstraksi fitur, pelatihan model, dan metrik evaluasi untuk melihat performa model dalam mendeteksi *URL phishing*. Objek penelitian ini adalah model klasifikasi *URL phishing* berbasis *machine learning* yang dibangun menggunakan dua algoritma *ensemble*, yaitu *Random Forest* dan *Gradient Boosting.*

B. Diagram Alir Penelitian

Secara umum, tahapan penelitian yang dilakukan meliputi beberapa proses seperti diagram alir penelitian berikut secara umum ditunjukkan pada Gambar 1.



GAMBAR 1
(Diagram Alir Penelitian)

C. Pengumpulan Dataset URL

Penelitian ini menggunakan data sekunder dari *Kaggle* berjudul *Malicious and Benign URLs*. *Dataset* berisi sekitar *450,176 URL* yang telah diberi label *benign* (aman) dan *malicious* (berbahaya).

URL	Target	Teknik Penyamaran	Label	Result
https://www.google.com/	Google	Aman	*Benign*	0
https://www.youtube.com/	Youtube	Aman	*Benign*	0
hxxp://lovejessy[.]com/whatSapp/	WhatsApp	**Brand impersonation** huruf kapital acak agar lolos filter	*Malicious/ phishing*	1
hxxp://alianzafrancesamty[.]com/.../paypal	PayPal	**Path spoofing** **PayPal** bukan domain asli	*Malicious/ phishing*	1

GAMBAR 2
(Contoh Dataset URL)

Format data *CSV* yang diproses menggunakan *Python* dan *Pandas*. *Dataset* ini dipilih karena jumlah datanya besar, memiliki variasi struktur *URL*, serta sudah memiliki label. Data tersebut digunakan sebagai input utama dalam proses *preprocessing*, ekstraksi fitur, pelatihan model, dan hasil performa metrik evaluasi.

D. *Preprocessing* Data *URL*

Preprocessing dilakukan untuk membersihkan dan menyiapkan data agar dapat digunakan dalam proses *machine learning.* Tahapan yang dilakukan, menghapus kolom yang tidak relevan, menghapus data kosong (*missing value*), menghapus data duplikat agar tidak menimbulkan bias, memastikan kolom *URL* dan label tersedia, dan mengubah *teks* menjadi data *numerik*, yaitu 0 untuk URL aman dan 1 untuk URL berbahaya. Setelah tahapan ini dilakukan, data siap diproses.

E. Subjek dan Objek Penelitian

Tahap ini mengubah *URL* yang masih berupa teks menjadi data numerik agar dapat diproses model *machine learning.* Fitur diambil dari struktur *URL* menggunakan proses parsing. Fitur yang digunakan yaitu:

- Panjang URL

$$L = |U| \qquad (1)$$

Menunjukkan jumlah karakter dalam *URL*. *URL phishing* dapat lebih panjang karena berisi tambahan kata, parameter, atau manipulasi alamat.

- Jumlah titik (.)

$$D = \sum_{i=1}^{n} I\,(c_i = '.') \qquad (2)$$

Menghitung jumlah tanda titik (.) pada *URL*. Banyak titik menunjukkan penggunaan *subdomain* berlebihan untuk menyamarkan alamat asli.

- Jumlah tanda hubung (-)

$$H = \sum_{i=1}^{n} I\,(c_i = '-') \qquad (3)$$

Menghitung jumlah tanda hubung (-). Tanda ini dipakai untuk membuat *domain* palsu terlihat menyerupai situs asli.

- Penggunaan HTTPS

$$HTTPS(U) = \begin{cases} 1, & jika\ scheme(u)=https \\ 0, & lainnya \end{cases} \qquad (4)$$

Menunjukkan apakah *URL* menggunakan *protokol HTTPS.*

- Kata sensitif seperti *login, verify, password*

$$S(U) = \begin{cases} 1, & jika\ ada\ kata\ sensitif\ dalam\ U \\ 0, & lainnya \end{cases} \qquad (5)$$

Memeriksa kandungan kata seperti *login, account, verify,* atau *password* dalam struktur *URL.*

- Panjang *path URL*

$$P = |path(U)| \qquad (6)$$

Mengukur panjang bagian path setelah domain. *URL phishing* cenderung memiliki *path* yang panjang untuk menyembunyikan *URL phishing.*

F. Pembagian Data Latih dan Data Uji

Dataset dibagi menjadi data latih dan data uji menggunakan metode *train-test split*. Pembagian dilakukan dengan rasio 80:20, yaitu 80% untuk pelatihan dan 20% untuk pengujian. Metode *stratified* sampling digunakan agar

proporsi kelas aman dan berbahaya tetap seimbang pada kedua data. Parameter *random_state* = 42 digunakan agar hasil pembagian konsisten.

G. Penyeimbangan Data dengan *SMOTE*

Jumlah *URL* aman pada *dataset* lebih banyak dibanding *URL* berbahaya, sehingga terjadi ketidakseimbangan kelas. Kondisi ini dapat membuat model lebih fokus pada kelas aman yang lebih banyak dibandingkan dengan kelas berbahaya. Metode *SMOTE digunakan* pada data latih.

$$x_{baru} = x_i + \delta \times (x_{zi} - x_i) \quad (7)$$

SMOTE menambah data sintetis pada kelas minoritas yaitu kelas *URL* berbahaya berdasarkan sampel yang mirip. Hal ini membantu model mempelajari kedua kelas dengan seimbang.

H. Pelatihan Model Machine Learning

Setelah data diseimbangkan dengan menggunakan *SMOTE*, dilakukan pelatihan model menggunakan dua algoritma *ensemble,* yaitu *Random Forest* dan *Gradient Boosting. Random Forest* dengan banyak pohon keputusan dan menggunakan *voting* untuk menentukan hasil akhir.

Gradient Boosting membangun model secara bertahap dimana model selanjutnya akan memperbaiki kesalahan sebelumnya untuk meningkatkan akurasi. Parameter yang digunakan *n_estimators = 200, learning_rate = 0.1 (Gradient Boosting), random_state = 42*, waktu pelatihan, dan waktu *inference* untuk melihat efisiensi model dalam melakukan prediksi.

I. Evaluasi dan Pengujian Model

Evaluasi dilakukan menggunakan *dataset* yang hasilnya statis, tidak ikut dalam proses pelatihan. Kedua model digunakan untuk memprediksi apakah *URL* termasuk aman atau phishing. Metrik evaluasi yang digunakan diantaranya *accuracy, precision, recall, F1-score*. Hasil kedua model kemudian dibandingkan untuk menentukan mana model terbaik. Pada penelitian ini, *F1-score* digunakan sebagai acuan.

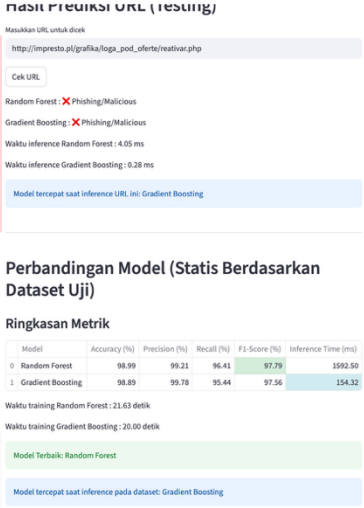
IV. HASIL DAN PEMBAHASAN

Hasil penelitian diperoleh melalui serangkaian eksperimen menggunakan dua algoritma *ensemble*, yaitu *Random Forest* dan *Gradient Boosting.* Pengujian dilakukan dengan variasi jumlah estimator (*n_estimators*) sebesar 50, 100, 150, dan 200, serta variasi pembagian data uji (*test size*) dari 0,20 hingga 0,55. Seluruh percobaan menggunakan *learning_rate* 0,1 dan pada tahap awal menerapkan *SMOTE* pada data latih untuk mengatasi ketidakseimbangan kelas. Evaluasi dilakukan menggunakan metrik *accuracy, precision, recall, F1-score*, serta waktu pelatihan sebagai ukuran efisiensi.

Model	*n_estimators*	*Test Size*	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1-Score (%)*	*Time Training (s)*
RF	200	0,2	98,82	98,01	96,90	97,45	30,54
GB	200	0,25	**98,87**	99,11	96,00	**97,53**	32,08
RF	150	0,25	98,81	97,84	97,01	97,43	20,23
GB	150	0,25	98,86	99,22	95,83	97,50	24,07
RF	100	0,25	98,81	97,84	97,02	97,43	13,46
GB	100	0,25	98,83	99,26	95,67	97,43	15,95
RF	50	0,25	98,81	97,82	97,02	97,42	6,93
GB	50	0,25	98,82	**99,48**	95,42	97,41	8,09

GAMBAR 3
(Ringkasan Hasil Terbaik Model)

Hasil menunjukkan bahwa kedua algoritma memiliki performa yang baik, dengan akurasi yang konsisten di atas 98%. Hal ini menunjukkan bahwa fitur berbasis struktur *URL* mampu mendeteksi *URL phishing* secara efektif. Berdasarkan hasil terbaik, *Gradient Boosting* dengan *n_estimators* = 200 dan *test size* = 0,25 menghasilkan performa tertinggi dengan akurasi 98,87% dan *F1-score* 97,53%. Model ini memiliki nilai *precision* yang tinggi, hingga lebih baik dalam mengidentifikasi *URL phishing*, dan *Random Forest* memiliki performa yang stabil pada berbagai konfigurasi dan unggul dari sisi efisiensi waktu pelatihan. Secara umum, kedua model memiliki keunggulan. *Gradient Boosting* lebih unggul pada ketepatan prediksi, sedangkan *Random Forest* lebih baik dari sisi kestabilan dan efisiensi.



GAMBAR 4
(Prototipe Web *Streamlit)*

Pada pengujian lanjutan tanpa menggunakan *SMOTE*, kedua model tetap menghasilkan performa tinggi. *Random Forest* sedikit lebih unggul pada *recall* dan *F1-score*, sedangkan *Gradient Boosting* lebih cepat pada waktu prediksi (*inference time*). Model kemudian diimplementasikan ke dalam prototipe sistem berbasis web menggunakan Streamlit, dan dapat melakukan prediksi cek status keamanan *URL*.

V. KESIMPULAN

Random Forest dan *Gradient Boosting* mampu memberikan performa yang baik dalam mendeteksi *URL phishing.* Kedua model menghasilkan akurasi 98% dengan *F1-score* 97% pada pengujian. *Gradient Boosting* menunjukkan performa dengan *F1-score* tertinggi sebesar 97,53% unggul pada kecepatan prediksi (*inference time*), pada pengujian *dataset* dan simulasi *real-time*. *Random Forest* hasilnya lebih baik sedikit pada pengujian yang tidak menggunakan *SMOTE*, Nilai *F1-score* dan *recall*nya. Menunjukkan *Random Forest* lebih efektif dalam mendeteksi *URL phishing,* dan perubahan *estimator* tidak memberikan pengaruh signifikan terhadap performa. Terdapat *trade-off* antara prediksi dan efisiensi. *Random Forest* unggul dalam kemampuan deteksi, dan *Gradient Boosting* lebih cepat dalam prediksi. Hasil evaluasi, *Random Forest* dipilih sebagai model terbaik karena memiliki performa klasifikasi yang lebih unggul dalam mendeteksi *URL phishing*. Penelitian berhasil mengembangkan prototipe sistem deteksi *URL phishing* berbasis web menggunakan *Python* dan

Streamlit. Mengintegrasikan tahapan, mulai dari *preprocessing* data, pelatihan model, dan prediksi *URL*.

REFERENSI

[1] Anti-Phishing Working Group (APWG), "Anti-Phishing Working Group (APWG) trends report 2023–2025," [Online]. Available: https://apwg.org/trendsreports. [Accessed: Mar. 21, 2026].

[2] A. R. Kamila, F. Adikara, and C. Herdian, "Pengaruh penambahan fitur dengan perbandingan algoritma berbasis bagging dan boosting pada deteksi phishing link," JEPIN (Jurnal Edukasi dan Penelitian Informatika), vol. 10, no. 3, pp. 2460–0741, 2024.

[3] A. Pradana and S. Susanto, "Implementasi model machine learning untuk deteksi phishing dengan pendekatan ekstraksi fitur yang dioptimalkan," Jurnal Teknologi Informasi dan Multimedia, vol. 8, no. 1, pp. 27–40, 2026.

[4] M. Subiyantoro, A. Setiawan, "Desain dan Analisis Metode Pengkondisian Clear and Less Malicious Internet Traffic" Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika (JITEKI), vol. 11, no. 2, pp 322-345, 2025.

[5] N. Rahmeisi, E. Gani, and A. Arfriandi, "Tinjauan literatur: pendekatan machine learning dalam deteksi serangan web," Jurnal Ilmiah Sistem Informasi, vol. 4, no. 3, pp. 772–791, 2025, doi: 10.51903/3w0vwc80.

[6] F. A. K. Hidayat, N. Sulistianingsih, and R. Hammad, "Perbandingan algoritma XGBoost dan random forest dalam klasifikasi surat masuk pemerintahan," RIGGS: Journal of Artificial Intelligence and Digital Business, vol. 4, no. 4, pp. 10390–10397, 2026, doi: 10.31004/riggs.v4i4.5044.

[7] M. I. Mazdadi, T. H. Saragih, I. Budiman, and M. N. Anshory, "Peningkatan akurasi model boosting pada prediksi kesehatan tidur menggunakan Optuna," JIP (Jurnal Informatika Polinema), vol. 12, no. 2, pp. 341–349, 2026.

[8] I. G. P. Wiratama and A. A. I. N. E. Karyawati, "Klasifikasi URL berbahaya menggunakan algoritma random forest berbasis fitur struktural," Jurnal Nasional Teknologi Informasi dan Aplikasinya (JNATIA), vol. 4, no. 1, pp. 39–46, 2025.

[9] W. B. T. Handaya dan Lukito, "Deteksi Website Phishing Menggunakan Teknik Machine Learning," Jurnal Informatika Atma Jogja, vol. 6, no. 1, pp. 69–80, 2025.

[10] E. Novianto, E. H. Heri, Rianto, "Keamanan Informasi (Information Security) pada Aplikasi Sistem Informasi Manajemen Sumber Daya Manusia," Jurnal Rabit, vol. 8, no. 1, pp. 10–15, 2023.

[11] A. F. Mahmud and S. Wirawan, "Deteksi Phishing Website menggunakan Machine Learning Metode Klasifikasi," Sistemasi: Jurnal Sistem Informasi, vol. 8, no. 1, pp. 1368–1380, 2023.

[12] F. D. Prasetyo, M. Maulana, F. Ramadhan, A. L. Setiabudi, and I. Budiawan, "Analisa Komparasi Kinerja Algoritma K-Nearest Neighbor (K-NN) dan Decision Tree dalam Klasifikasi Situs Web Phising," Jurnal Pengabdian Masyarakat dan Riset Pendidikan, vol. 4, no. 3, pp. 16587–16591, 2026.

[13] R. Riansyah, D. A. Aptanta, H. Aryanda, M. Farhan, and I. Rusydi, "Implementasi Website Deteksi Phishing Link Menggunakan SSL Validation dan URL Scoring," Neptunus: Jurnal Ilmu Komputer dan Teknologi Informasi, vol. 4, no. 1, pp. 63–75, 2026.

[14] M. A. B. Dewanto, M. Fathurrahman, D. R. Firdaus, and A. Setiawan, "Penipuan Penambah Followers Instagram: Analisis Serangan Phising dan Dampaknya pada Keamanan Data," Journal of Internet and Software Engineering, vol. 1, no. 4, pp. 1–11, 2024.

[15] W. A. Kurniawan and A. Salam, "Penggunaan Feature Space SMOTE untuk Mengurangi Overfitting Akibat Imbalance Dataset," Jurnal Transformatika, vol. 22, no. 2, pp. 140–149, 2025.

[16] C. Paramita, C. S. Simbolon, A. S. Pamungkas, J. M. Triono, E. P. W. Utomo, and E. R. Subhiyakto, "Analisis Pengaruh SMOTE terhadap Kinerja Model KNN untuk Prediksi Risiko Stroke," Jurnal Informatika: Jurnal Pengembangan IT, vol. 10, no. 4, pp. 978–988, 2025.

[17] A. Ernawati, Z. Sitorus, M. Iqbal, and D. Nasution, "Penerapan data mining untuk klasifikasi penduduk miskin di Kabupaten Labuhanbatu menggunakan random forest dan k-nearest neighbors," Bulletin of Information Technology (BIT), vol. 6, no. 2, pp. 23–35, 2025.

[18] H. Mulyo and A. K. Zyen, "Pengaruh hyperparameter tuning gradient boosting terhadap prediksi pemilihan program studi mahasiswa baru," Bulletin of Computer Science Research, vol. 5, no. 2, pp. 131–137, 2025.

[19] F. A. Rangkuti, Khairunnisa, and S. Sundari, "Implementasi gradient boosting machines untuk prediksi harga rumah pada Jakarta Selatan," Jurnal Kecerdasan Buatan dan Teknologi Informasi, vol. 4, no. 2, pp. 164–172, 2025.

[20] R. Ridwan, H. H. Handayani, S. A. P. Lestari, and Y. Cahyana, "Evaluasi kinerja algoritma random forest dan gradient boosting untuk klasifikasi penyakit jantung," Jurnal Komtika (Komputasi dan Informatika), vol. 9, no. 1, pp. 112–124, 2025.

[21] S. Helmiyah, R. Pramestiawan, "Analisis komparatif algoritma machine learning dengan metrik akurasi, presisi, recall, dan F1-score pada dataset kacang kering," Jurnal IKOMTI, vol. 6, no. 3, pp. 152–159, 2025.

[22] Kaggle, "Malicious And Benign URLs" Siddharth Kumar [Online] Available: https://www.kaggle.com/datasets/siddharthkumar25/malicious-and-benign-urls. [Accessed: Feb. 11, 2026].