

PENGEMBANGAN SISTEM DETEKSI URL PHISHING MENGGUNAKAN METODE HYBRID RULE-BASED DAN MACHINE LEARNING

Sigit April Liawan, Imam Suharjo

Teknik Informatika, Universitas Mercu Buana Yogyakarta

Jl. Ring Road Utara, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta

221110063@student.mercubuana-yogya.ac.id

ABSTRAK

Meningkatnya penggunaan internet membuat ancaman URL phishing menjadi semakin serius dan merugikan pengguna internet. Pendekatan rule-based dan XGBoost secara mandiri masih memiliki keterbatasan dalam menangani variasi pola phishing yang terus berkembang. Penelitian ini bertujuan membangun sistem deteksi URL phishing yang lebih andal melalui pendekatan hybrid weighted ensemble. Metode yang digunakan dalam penelitian ini menggabungkan rule-based dan XGBoost. Dataset yang digunakan sebanyak 242.056 URL seimbang dari Kaggle. Sistem dibangun melalui ekstraksi 23 fitur numerik yang mencakup fitur struktur URL, entropi, indikator biner, serta fitur turunan berbasis aturan seperti rule_score. Model XGBoost dilatih menggunakan hyperparameter tuning dengan grid search dan 3-fold cross-validation. Kombinasi hybrid dilakukan dengan menggabungkan probabilitas output kedua komponen pada 40 kombinasi bobot dan threshold. Hasil penelitian menunjukkan bahwa konfigurasi terbaik berupa bobot XGBoost 0,85, bobot rule-based 0,15, dan threshold 0,45. Sistem hybrid mencapai akurasi 95,22% dan precision 96,22%, mengungguli XGBoost saja (akurasi 95,21%, precision 96,10%) dan rule-based saja (akurasi 50,69%), serta berhasil menurunkan false positive dari 925 menjadi 894 kasus.

Kata kunci : URL Phishing, Rule-Based, XGBoost, Hybrid, Machine Learning

1. PENDAHULUAN

Dalam era digital saat ini, internet telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari, dimana banyak transaksi dan komunikasi dilakukan secara online[1]. Di Indonesia, pada tahun 2019, jumlah pengguna internet mencapai 73,7% dari total populasi, menandakan peningkatan sebesar 8,9% dibandingkan tahun sebelumnya[1]. Seiring dengan peningkatan penggunaan internet tersebut muncul masalah baru mengenai keamanan internet, salah satunya yakni *Phishing*. *Phishing* adalah cara penipuan yang memanfaatkan trik untuk membuat pengguna memberikan data pribadi melalui link URL yang palsu, yang terlihat seperti situs asli[2]. Teknik *phishing* yang semakin canggih memungkinkan penjahat menipu pengguna yang kemudian dapat menyebabkan kerugian besar[3].

Salah satu pendekatan yang sering digunakan untuk mengidentifikasi URL *phishing* adalah dengan menggunakan metode yang berbasis aturan (rule-based). Rule based adalah sistem yang menggunakan aturan-aturan logika untuk memecahkan masalah atau mengambil keputusan[4]. Namun, pendekatan ini memiliki keterbatasan karena bersifat eksplisit namun cenderung kaku, sehingga rentan gagal ketika menghadapi variasi pola atau penggunaan sinonim yang tidak sesuai dengan aturan yang telah ditentukan[5]. Selain itu, kinerja sistem sangat bergantung pada kelengkapan dan kualitas aturan yang dibuat, sehingga apabila aturan yang tersedia terbatas atau tidak mampu merepresentasikan seluruh kemungkinan pola, maka sistem berpotensi tidak dapat mendeteksi kasus secara optimal[5].

Untuk mengatasi keterbatasan tersebut, pendekatan machine learning dapat digunakan sebagai pelengkap. Salah satu metode machine learning yang bisa digunakan adalah XGBoost (eXtreme Gradient Boosting). Penelitian yang dilakukan oleh [6] menunjukkan bahwa metode ini memberikan hasil akurasi tertinggi yakni mencapai 96%, menunjukkan bahwa XGBoost memiliki kemampuan yang baik dalam menangani data dalam jumlah besar serta mampu mengenali pola-pola kompleks. Metode XGBoost juga dikenal sebagai metode yang kokoh, walaupun sulit untuk menentukan nilai setiap parameter sehingga waktu yang dihabiskan tidak sedikit[7]. Namun, penggunaan model atau metode tunggal belum mampu menangkal serangan *phishing* secara efektif[8]. Oleh karena itu, diperlukan pendekatan yang menggabungkan lebih dari satu metode untuk meningkatkan keandalan deteksi.

Penelitian ini bertujuan membangun sistem deteksi URL phishing yang lebih andal dengan menggabungkan metode rule-based dan XGBoost dalam pendekatan hybrid. Penyelesaian masalah dilakukan melalui ekstraksi fitur numerik dari URL, pelatihan model XGBoost dengan hyperparameter tuning, serta penggabungan probabilitas kedua komponen menggunakan weighted ensemble dengan konfigurasi bobot dan threshold optimal yang diperoleh melalui grid search. Pendekatan ini diharapkan dapat saling melengkapi kelemahan masing-masing metode sehingga menghasilkan sistem deteksi yang lebih baik.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian sebelumnya telah mencoba deteksi URL phishing menggunakan berbagai pendekatan. Mahmud dan Wirawan [1] membuktikan bahwa algoritma machine learning mampu mengenali pola URL berbahaya secara efektif. Handaya dan Lukito [6] membandingkan beberapa algoritma dan menemukan bahwa XGBoost menghasilkan akurasi tertinggi sebesar 96% dibandingkan Decision Tree dan Random Forest. Mourtaji et al. [9] mengusulkan pendekatan hybrid rule-based dengan CNN yang terbukti lebih baik dibandingkan metode tunggal. Afandi et al. [10] mengimplementasikan XGBoost secara real-time dalam bentuk ekstensi browser dengan performa yang baik. Irawan et al. [2] menggunakan CNN berbasis deep learning dan berhasil mencapai tingkat akurasi yang tinggi.

2.2. Phishing

Phishing adalah cara penipuan dalam dunia digital yang dirancang untuk mengambil data atau informasi penting dari korban, seperti data pribadi, informasi keuangan, dan akun online, melalui berbagai teknik penipuan[11]. Istilah resmi *phishing* berasal dari bahasa Inggris *fishing* yaitu memancing, sedangkan pelaku kejahatan *phishing* disebut *Phisher*, dan biasanya mereka sering menyamar menjadi entitas yang sah atau terpercaya dalam sebuah komunikasi elektronik seperti perusahaan, bank dan juga situs web yang terkenal[12]. Serangan *Phishing* biasanya terdiri dari empat fase utama. yakni fase perencanaan (*planning phase*), fase persiapan (*attack preparation*), fase eksekusi (*attack conducting phase*), dan fase akuisisi (*valuable acquisition phase*) [3].

2.2. URL (Uniform Resource Locator)

URL (Uniform Resource Locator) adalah Sebuah alamat yang digunakan untuk menyimpan serta merepresentasikan data atau informasi yang terhubung langsung dengan suatu website [13]. URL memiliki tiga bagian utama yakni protokol, domain, dan path. Bagian path sering menjadi sasaran kejahatan karena menunjukkan lokasi data pada website. Selain itu, URL bisa memiliki parameter tambahan termasuk query yang digunakan untuk mengirimkan informasi ke server dan fragmen. Oleh karena itu, memahami struktur URL sangat penting dalam mendukung usaha untuk mendeteksi dan mencegah serangan berbasis web seperti penipuan berbasis *phishing*.

2.3. Metode Rule-Based

Metode *Rule-based* atau metode berbasis aturan adalah suatu metode yang merepresentasikan pengetahuan dalam bentuk aturan logika (IF-THEN) untuk menghasilkan keputusan atau kesimpulan berdasarkan fakta yang diberikan[14]. Metode *rule-based* juga memiliki sejumlah keuntungan, yakni aturan yang diterapkan bersifat jelas dan mudah dipahami serta konsisten memproses deteksi secara

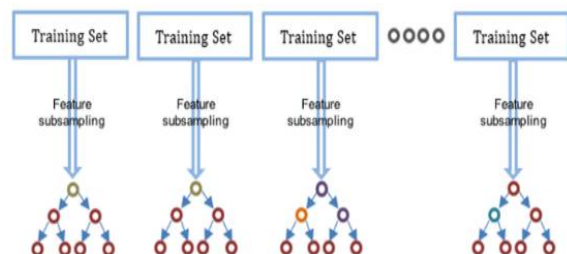
efisien. Namun, karena sifatnya yang kaku membuat metode ini kurang mampu menangani variasi pola baru[5].

2.4. Machine Learning

Machine learning atau pembelajaran mesin adalah salah satu cabang kecerdasan buatan yang memungkinkan komputer belajar secara mandiri dari data pelatihan dan akan meningkat seiring waktu tanpa diprogram secara eksplisit[15]. Algoritma ini dapat mendeteksi pola dalam sebuah data lalu dipelajari untuk membuat prediksi sendiri. Machine learning sudah terbukti secara efektif dalam berbagai bidang, salah satunya pada deteksi serangan *phishing*[1].

2.5. XGBoost (eXtreme Gradient Boosting)

XGBoost adalah singkatan dari eXtreme Gradient Boosting. XGBoost merupakan salah satu metode yang berbasis Gradient Tree Boosting canggih yang dapat bekerja secara efisien dalam menangani masalah skala besar dengan sumber daya komputasi yang sangat terbatas[7]. Selain itu, dalam penelitian yang dilakukan oleh [6], XGBoost memberikan hasil akurasi tertinggi yakni 96% dibandingkan algoritma Decision Tree dan Random Forest pada kasus deteksi URL *phishing*. XGBoost bekerja dengan membangun beberapa pohon keputusan secara bertahap, di mana setiap pohon baru dilatih untuk memperbaiki kesalahan pada pohon sebelumnya menggunakan metode gradient descent, dengan memberikan bobot lebih besar pada variabel yang sulit diprediksi sebagaimana diilustrasikan pada Gambar 1[10].



Gambar 1. Representasi klasifikasi XGBoost

Untuk mengukur performa model XGBoost, confusion matrix digunakan sebagai dasar perhitungan metrik evaluasi. Confusion matrix terdiri dari empat komponen yaitu True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN)[1]. Selain keempat metrik dalam tabel di atas, terdapat juga ROC-AUC (Receiver Operating Characteristic - Area Under Curve).

3. METODE PENELITIAN

3.1. Pendekatan Penelitian

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan eksperimental. Pendekatan ini menggabungkan rule-based dan XGBoost untuk mendeteksi URL phishing. Proses ini dilakukan melalui tahapan pengolahan dataset, ekstraksi fitur,

pelatihan model, pengujian, serta evaluasi performa menggunakan metrik klasifikasi.

3.2. Bahan Penelitian

Bahan penelitian yang digunakan dalam penelitian ini adalah data URL yang diperoleh dari

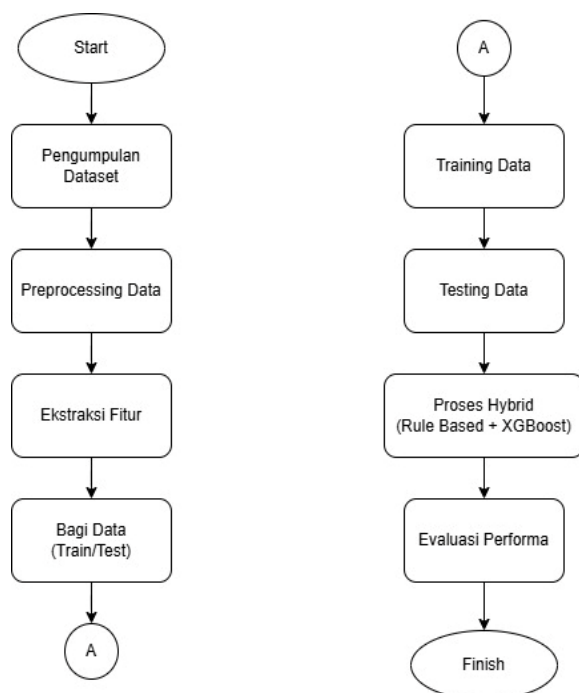
dataset Kaggle, yakni “Phishing URL Detection Dataset” [16]. Dataset ini terdiri dari 242.056 data URL dengan komposisi 121.028 URL *phishing* (label = 1) dan 121.028 URL *legitimate* (label = 0), sehingga tidak diperlukan teknik penyeimbangan data seperti *oversampling* atau *undersampling*.

Tabel 1. Contoh dataset

No	url	url_length	domain_length	...	label
1	http://emailupgradedpt.3eeweb.com/webmail-upgrades.html	55	26	...	1
2	http://www.longmeadowlibrary.org/	33	25	...	0
3	https://www.hermesllc.com/	26	17	...	0
...	...	...	...	...	...
242056	http://dislineltd.skylimited.com.ng/lp/auth/view/share/index.html	65	28	...	1

Berdasarkan Tabel 1 di atas, dataset terdiri dari 242.056 data URL dengan total 17 fitur numerik dan 1 kolom label. Dalam tabel tersebut mencakup berbagai karakteristik URL, fitur berbasis konten serta pelabelan untuk URL tersebut. Keterangan di dalam dataset tersebut cukup untuk pengolahan dalam proses berikutnya.

3.3. Alur Penelitian



Gambar 2. Diagram alur penelitian

Penelitian dilakukan melalui beberapa tahapan yang meliputi pengumpulan dataset, preprocessing data, ekstraksi fitur, pembagian data training dan

testing, training dan testing data, proses hybrid dan evaluasi performa. Alur penelitian ditunjukkan pada Gambar 2.

3.4. Preprocessing

Tahap preprocessing dilakukan untuk memastikan kualitas data sebelum digunakan dalam proses selanjutnya. Tahapan ini meliputi pemeriksaan data kosong (missing value), pengecekan duplikat, konversi tipe data, serta penyesuaian format data agar dapat diproses algoritma machine learning. Selain itu, dilakukan juga penambahan fitur tambahan berbasis rule-based untuk mendukung proses klasifikasi. Setelah proses preprocessing selesai, dilakukan pembagian dataset menjadi dua yakni data latih dan data uji dengan perbandingan 80:20.

3.5. Ekstraksi Fitur

Tahap ekstraksi fitur dilakukan untuk mengubah URL menjadi atribut numerik yang dapat diproses machine learning. Tahap ini mengelompokkan fitur-fitur URL ke dalam tiga kategori berdasarkan karakteristiknya. Karakteristik URL yang sering digunakan sebagai indikator phishing seperti panjang dan struktur URL, karakteristik domain dan entropi (tingkat keacakan url), indikator domain, indikator biner, serta fitur tambahan berbasis rule-based.

3.6. Pelatihan Model XGBoost

Pada proses ini model XGboost dilatih menggunakan dataset yang sudah dibagi menjadi data latih dan data uji sebelumnya. Proses berikutnya yakni hyperparameter tuning menggunakan Grid Search dan 3-Fold Cross Validation untuk memperoleh konfigurasi model yang optimal. Parameter yang diuji meliputi *max_depth*, *learning_rate*, *n_estimators*, *subsample*, dan *colsample_bytree*. Hasil konfigurasi ditunjukkan pada Tabel 2.

Tabel 2. Hyperparameter XGBoost

Parameter	Nilai	Keterangan
max_depth	8	Kedalaman maksimum pohon keputusan
learning_rate	0.2	Kecepatan pembelajaran
n_estimators	300	Jumlah pohon yang dibangun
subsample	1.0	Proporsi data per pohon
colsample_bytree	0.8	Proporsi fitur per pohon
objective	binary:logistic	Fungsi Objektif klasifikasi biner
eval_metric	logloss	Metrik evaluasi pelatihan

Berdasarkan Tabel 2, kombinasi parameter terbaik digunakan sebagai konfigurasi akhir model XGBoost pada proses pelatihan. Model yang telah dilatih kemudian dievaluasi menggunakan metrik accuracy, precision, recall, F1-score, dan ROC-AUC untuk mengukur kemampuan model dalam membedakan URL phishing dan legitimate. Selain itu, analisis feature importance dilakukan untuk mengetahui fitur yang paling berkontribusi terhadap proses klasifikasi.

3.7. Sistem Hybrid

Proses hybrid menggabungkan pendekatan rule-based dan XGBoost untuk meningkatkan kemampuan deteksi URL phishing. Komponen rule-based melakukan penilaian terhadap URL berdasarkan 13 aturan yang menggambarkan karakteristik umum URL phishing. Setiap aturan memiliki bobot tertentu sesuai tingkat indikasinya terhadap aktivitas phishing. Daftar aturan dan bobot yang digunakan ditampilkan pada Tabel 3.

Tabel 3. Aturan dan bobot rule-based scoring

No	Aturan	Bobot	Kondisi Deteksi
1	Panjang URL	10	Panjang URL > 75 karakter
2	Penggunaan IP address	25	URL mengandung pola IP (x.x.x.x)
3	Tidak pakai HTTPS	10	URL tidak diawali https://
4	Simbol @	20	URL mengandung karakter @
5	Double slash redirect	15	Terdapat // setelah bagian protokol
6	URL shortener	15	Domain cocok dengan layanan pemendek URL
7	TLD mencurigakan	10	TLD termasuk .tk, .ml, .xyz, dll
8	Jumlah subdomain berlebih	10	Jumlah titik di hostname > 3
9	Keyword <i>phishing</i>	15	Mengandung kata login, verify, password
10	Brand di path URL	15	Nama brand ada di path bukan domain
11	Hyphen di path URL	10	Domain utama mengandung tanda -
12	Punycode	15	URL mengandung xn--
13	Port non-standar	10	Port selain 80 dan 443

Berdasarkan Tabel 3, setiap aturan dirancang untuk mendeteksi karakteristik mencurigakan yang biasanya ditemukan pada URL *phishing*. Aturan yang memiliki tingkat kepastian lebih tinggi akan diberikan bobot yang lebih besar, sedangkan yang bersifat pendukung diberikan bobot lebih kecil. Total skor yang diperoleh kemudian dinormalisasi menjadi nilai probabilitas untuk digunakan pada proses penggabungan dengan model XGBoost.

Model XGBoost yang telah dilatih pada tahap sebelumnya menghasilkan probabilitas klasifikasi berdasarkan fitur-fitur URL yang telah diekstraksi. Selanjutnya, probabilitas dari komponen rule-based dan XGBoost digabungkan menggunakan pendekatan weighted ensemble. Bobot optimal diperoleh melalui proses grid search dengan menguji berbagai kombinasi bobot dan threshold pada data validasi. Konfigurasi terbaik yang diperoleh ditunjukkan pada Tabel 4.

Tabel 4. Konfigurasi optimal sistem hybrid

Parameter	Nilai Optimal
Bobot XGBoost (w_xgb)	0,85
Bobot Rule-based (w_rule)	0,15
Threshold keputusan	0,45

Berdasarkan Tabel 4, sistem memberikan bobot yang lebih besar kepada XGBoost sebagai komponen utama klasifikasi, sementara rule-based sebagai informasi pendukung. Probabilitas akhir sistem *hybrid* dihitung melalui persamaan berikut.

$$P_{\text{hybrid}} = (0,85 \times P_{\text{XGB}}) + \left(0,15 \times \frac{\text{rule_score}}{100}\right) \quad (1)$$

Keterangan :

P_{hybrid} = probabilitas akhir gabungan (0-1)

P_{XGB} = probabilitas keluaran dari model XGBoost (0-1)

rule_score = skor mentah dari *rule-based* (0-100)

3.8. Evaluasi Performa

Performa sistem *hybrid* akan dianalisis dengan menggunakan confusion matrix sebagai dasar untuk menghitung metrik. Akurasi digunakan untuk mengukur seberapa besar proporsi prediksi yang benar secara keseluruhan :

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Presisi mengukur seberapa andal model dalam memprediksi kelas positif, sedangkan recall mengukur kemampuan model dalam menemukan seluruh sampel positif yang sebenarnya:

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

F1-Score menggabungkan presisi dan recall menjadi satu nilai tunggal melalui rata-rata harmonisnya:

$$\text{F1-Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (5)$$

Selain itu, ROC-AUC digunakan untuk mengukur kemampuan model dalam membedakan kelas positif dan negatif secara keseluruhan di berbagai nilai ambang batas:

$$\text{ROC-AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \quad (6)$$

Kelima metrik tersebut dipilih karena saling melengkapi dalam menilai kinerja model dalam mendeteksi URL *phishing*. Dengan melihat hasil dari metrik tersebut juga kita dapat mengetahui hasil dari pengujian dataset. Metriks ini nantinya akan digunakan untuk menilai kinerja sistem rule-based, XGBoost dan hybrid nantinya.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Preprocessing Dataset

Tahap preprocessing dilakukan untuk memastikan kualitas data sebelum digunakan pada proses pelatihan model. Dalam proses ini meliputi pemeriksaan data kosong, pengecekan duplikasi, pembagian data latih dan data uji, serta pembentukan fitur tambahan melalui feature engineering. Ringkasan hasil preprocessing dapat dilihat pada Tabel 5.

Tabel 5. Hasil preprocessing dataset

No	Keterangan	Hasil
1	Jumlah data awal	242.056 baris, 18 kolom
2	Data kosong	240 data (194 train + 46 test)
3	Data duplikat	0
4	Jumlah label <i>phishing</i> (1)	121.028 data
5	Jumlah label <i>legitimate</i> (0)	121.028 data
6	Jumlah fitur yang digunakan	23 fitur
7	Data untuk pelatihan (80%)	193.644 data
8	Data untuk pengujian (20%)	48.412 data

Jumlah data *phishing* dan *legitimate* yang sama persis menunjukkan bahwa dataset tersebut sudah seimbang, sehingga tidak diperlukan teknik tambahan untuk menyeimbangkan data. Data selanjutnya dibagi menjadi dua yakni 80% untuk data latih dan 20% untuk data uji. Proses *feature engineering* menghasilkan 23 fitur yang digunakan sebagai input model, terdiri dari 16 fitur hasil ekstraksi awal dan tujuh fitur turunan berbasis rule seperti *rule_score*, *rule_binary* dan *rule_boundary_distance*.

4.2. Hasil Pelatihan dan Pengujian XGBoost

Pelatihan model dilakukan menggunakan 193.644 data latih yang sudah dibagi pada proses sebelumnya. Dari jumlah tersebut, selanjutnya 20% (38.729 data) disisihkan sebagai data validasi. Dengan demikian, model hanya dilatih menggunakan 154.915 data secara efektif. Hasil evaluasi model tersebut dapat dilihat pada Tabel 6.

Tabel 6. Hasil evaluasi XGBoost pada data validasi

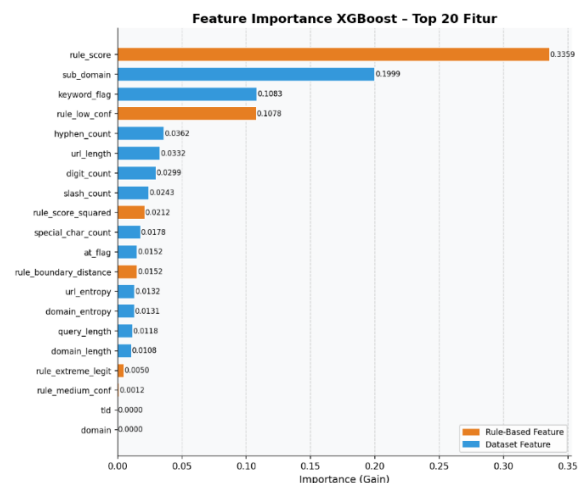
Metrik Evaluasi	Nilai
Accuracy (Akurasi)	95,34%
Precision	96,17%
Recall	94,44%
F1-Score	95,30%
ROC-AUC	0,9901

Proses *hyperparameter tuning* menghasilkan parameter terbaik yaitu *max_depth*=8, *learning_rate*=0,2, *n_estimators*=300, *subsample*=1,0 dan *colsample_bytree*=0,8 dengan nilai *cross-validation score* sebesar 95,25%. Model ini menunjukkan stabilitas yang baik dengan rata-rata F1-Score 0,9525 dan standar deviasi hanya $\pm 0,0004$, yang mengindikasikan konsistensi model terhadap berbagai pembagian data.

Tabel 7. Peringkat fitur terpenting XGBoost

No	Nama Fitur	Nilai Kepentingan
1	<i>rule_score</i>	0,3359
2	<i>sub_domain</i>	0,1999
3	<i>keyword_flag</i>	0,1083
4	<i>rule_low_conf</i>	0,1078
5	<i>hyphen_count</i>	0,0362
6	<i>url_length</i>	0,0332
7	<i>digit_count</i>	0,0299
8	<i>slash_count</i>	0,0243
9	<i>rule_score_squared</i>	0,0212
10	<i>special_char_count</i>	0,0178

Berdasarkan Tabel 7, fitur *rule_score* menjadi fitur yang memiliki pengaruh paling tinggi dengan nilai 0,3359. Fitur kedua *sub_domain* (0,1999) dan *keyword_flag* (0,1083) dan *rule-low-conf* (0,1078) mengikuti di belakang.



Gambar 3. Fitur terpenting XGBoost

Berdasarkan Tabel 7 dan Gambar 3 di atas, informasi yang dihasilkan dari komponen rule-based memberikan kontribusi yang signifikan terhadap keputusan klasifikasi model XGBoost.

Setelah model selesai dilatih, tahap selanjutnya adalah menguji model menggunakan data testing atau

data uji. Data uji ini terdiri dari 48.412 data yang belum pernah sama sekali dilihat oleh model selama pelatihan.

Tabel 8. Hasil evaluasi XGBoost pada data uji

Metrik Evaluasi	Nilai
Accuracy (Akurasi)	95,21%
Precision	96,10%
Recall	94,23%
F1-Score	95,16%
ROC-AUC	0,9892

Berdasarkan Tabel 8, model XGBoost memperoleh akurasi sebesar 95,21%, precision 96,10%, recall 94,23%, F1-Score 95,16%, dan ROC-AUC 0,9892. Perbedaan akurasi antara data validasi dan data uji hanya sebesar 0,13%, yang menunjukkan bahwa model memiliki kemampuan generalisasi yang baik dan tidak mengalami overfitting. Untuk mengetahui hasil prediksi secara lebih detail, digunakan *confusion matrix*.

Tabel 9. Confusion matrix XGBoost pada data uji

	Prediksi Legitimate (Aman)	Prediksi Phishing (Berbahaya)
Sebenarnya Legitimate (Aman)	TN = 23.281	FP = 925
Sebenarnya Phishing (Berbahaya)	FN = 1.396	TP = 22.810

Tabel 9 menunjukkan bahwa model berhasil mengklasifikasikan 23.281 URL legitimate dan 22.810 URL phishing dengan benar. Kesalahan klasifikasi terdiri atas 925 false positive dan 1.396 false negative. Secara keseluruhan, jumlah prediksi yang benar jauh lebih dominan dibandingkan kesalahan prediksi.

4.3. Hasil Evaluasi Sistem Hybrid

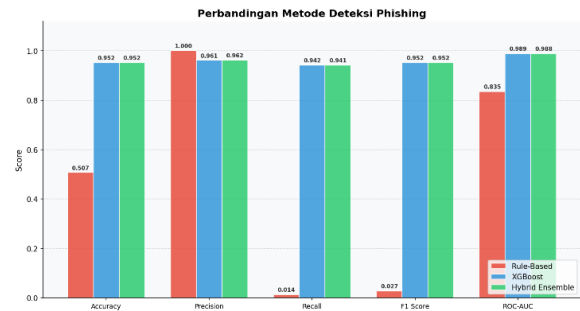
Evaluasi sistem hybrid dilakukan dengan menggabungkan hasil rule-based dan XGBoost menggunakan konfigurasi bobot optimal yang telah diperoleh sebelumnya. Performa sistem kemudian dibandingkan dengan metode rule-based dan XGBoost secara terpisah. Hasil perbandingan ditunjukkan pada Tabel 10.

Tabel 10. Perbandingan performa metode

Metode	Akurasi	F1-Score	ROC-AUC
Rule-based Saja	50,69%	0,0271	-
XGBoost Saja	95,21%	0,9516	0,9892
Hybrid (Rule-based + XGBoost)	95,22%	0,9516	0,9884

Berdasarkan hasil Tabel 10, sistem *hybrid* mampu mencapai akurasi 95,22%, yang artinya lebih baik dibanding XGBoost saja (95,21%). Meskipun peningkatannya kecil (sekitar 0,01%), hasil ini menunjukkan bahwa pendekatan *hybrid* mampu

memberikan performa yang lebih baik dibandingkan pendekatan tunggal.



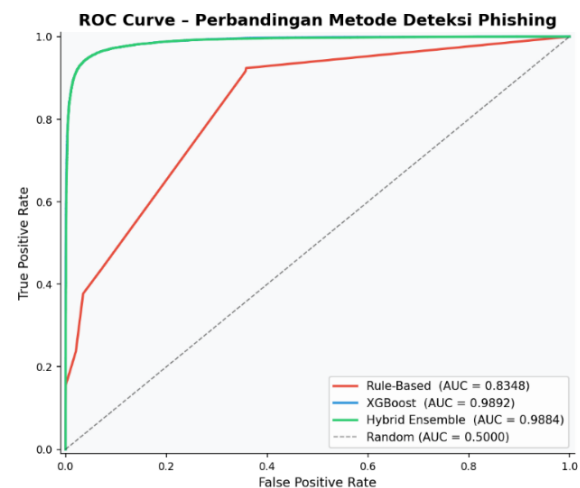
Gambar 4. Perbandingan performa model

Gambar 4 memperlihatkan perbandingan performa ketiga metode deteksi phishing yang digunakan dalam penelitian. Terlihat bahwa metode rule-based memiliki performa yang jauh lebih rendah dibandingkan pendekatan berbasis machine learning, sedangkan sistem hybrid mampu menghasilkan performa terbaik meskipun selisihnya tidak terlalu besar dibandingkan XGBoost. Untuk memperoleh gambaran mengenai performa sistem hybrid, dilakukan evaluasi menggunakan beberapa metrik klasifikasi sebagai berikut.

Tabel 11. Metrik evaluasi sistem hybrid

Metrik Evaluasi	Nilai
Accuracy (Akurasi)	95,22%
Precision	96,22%
Recall	94,13%
F1-Score	95,16%
ROC-AUC	0,9884

Berdasarkan Tabel 11, sistem hybrid menghasilkan precision sebesar 96,22%, recall 94,13%, F1-Score 95,16%, dan ROC-AUC 0,9884. Nilai tersebut menunjukkan bahwa sistem memiliki kemampuan yang sangat baik dalam membedakan URL phishing dan legitimate serta mampu mempertahankan keseimbangan antara precision dan recall.



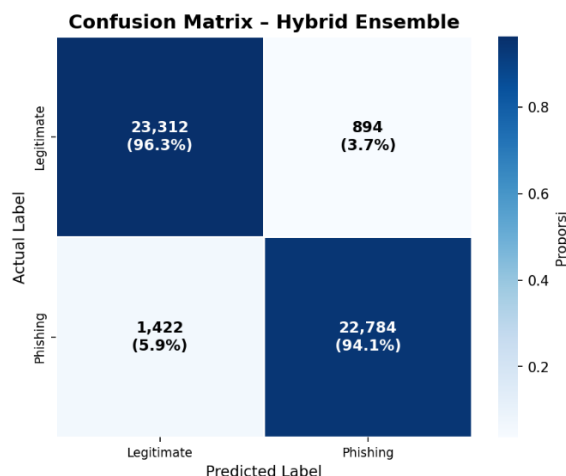
Gambar 4. Perbandingan metode deteksi phishing

Gambar 5 menunjukkan kurva ROC dari model yang digunakan. Kurva yang mendekati sudut kiri atas mengindikasikan kemampuan klasifikasi yang sangat baik. Nilai ROC-AUC sebesar 0,9884 menunjukkan bahwa sistem hybrid mampu membedakan URL phishing dan legitimate dengan tingkat akurasi yang tinggi. Selain menggunakan metrik evaluasi, analisis juga dilakukan melalui confusion matrix untuk mengetahui distribusi prediksi benar dan salah yang dihasilkan oleh sistem hybrid. Hasil confusion matrix ditunjukkan pada Tabel 12.

Tabel 12. Confusion matrix sistem hybrid

	Prediksi Legitimate (Aman)	Prediksi Phishing (Berbahaya)
Sebenarnya Legitimate (Aman)	TN = 23.312	FP = 894
Sebenarnya Phishing (Berbahaya)	FN = 1.422	TP = 22.784

Berdasarkan Tabel 12, sistem hybrid berhasil mengklasifikasikan 23.312 URL legitimate dan 22.784 URL phishing dengan benar. Kesalahan klasifikasi terdiri atas 894 false positive dan 1.422 false negative. Hasil ini menunjukkan bahwa jumlah prediksi yang benar jauh lebih dominan dibandingkan kesalahan prediksi.



Gambar 5. Confusion Matrix Heatmap

Gambar 6 memperlihatkan visualisasi confusion matrix sistem hybrid. Dominasi nilai pada kategori true negative dan true positive menunjukkan bahwa sistem mampu melakukan klasifikasi dengan baik, sedangkan jumlah false positive dan false negative relatif kecil dibandingkan total data yang diuji.

4.4. Pembahasan

Penelitian menunjukkan bahwa model XGBoost dapat mendeteksi URL phishing dengan baik, yang terlihat dari akurasi 95,21% dan ROC-AUC 0,9892. Menggabungkan komponen rule-based ke dalam sistem hybrid meningkatkan akurasi menjadi 95,22% dan precision meningkat dari 96,10% menjadi 96,22%.

Walaupun peningkatan yang didapat tidak terlalu besar, sistem hybrid berhasil menurunkan jumlah false positive, sehingga lebih sedikit URL yang sah yang salah dikenali sebagai phishing. Temuan ini menunjukkan bahwa menggabungkan metode rule-based dan XGBoost dapat saling mendukung dan memberikan manfaat tambahan untuk meningkatkan keandalan dalam mendeteksi URL phishing.

5. KESIMPULAN DAN SARAN

Metode *rule-based* yang digunakan secara mandiri terbukti kurang efektif dengan akurasi 50,69% dan F1-Score 0,0271, sedangkan model XGBoost menunjukkan kinerja unggul dengan akurasi 95,21%, F1-Score 0,9516, dan ROC-AUC 0,9892 pada konfigurasi `max\_depth=8`, `learning\_rate=0,2`, dan `n\_estimators=300`. Sistem hibrid (XGBoost 85%, *rule-based* 15%, ambang batas 0,45) mencapai akurasi 95,22%, menurunkan *false positive* dari 925 menjadi 894 kasus, dan meningkatkan *precision* dari 96,10% menjadi 96,22%, membuktikan bahwa integrasi kedua metode saling melengkapi. Penelitian lanjutan disarankan untuk menambahkan fitur usia domain dan reputasi SSL, menguji sistem secara real-time dalam bentuk ekstensi browser, serta memvalidasi pada dataset yang lebih baru mengingat teknik phishing modern yang semakin banyak menggunakan HTTPS.

DAFTAR PUSTAKA

- [1] A. F. Mahmud and S. Wirawan, "Deteksi Phishing Website menggunakan Machine Learning Metode Klasifikasi," *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no. 4, pp. 1368–1380, Jul. 2024, [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [2] W. Irawan, F. Alifian Fadhlurrahman, E. Haikal Purboyo, and dan Renaldi Cahya Kusuma, "Pencegahan Serangan Siber Melalui Deteksi Url Phishing Menggunakan Model Cnn Berbasis Deep Learning Dengan Tingkat Akurasi Optimal," *SENTIMETER (Seminar Nasional Teknologi Informasi, Mekatronika dan Ilmu Komputer)*, vol. 5, pp. 181–186, Jan. 2026, Accessed: Apr. 21, 2026. [Online]. Available: <http://prosiding.sentimeter.nusaputra.ac.id/index.php/prosiding/article/view/115>
- [3] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing Attacks: A Recent Comprehensive Study and a New Anatomy," *Front. Comput. Sci.*, vol. 3, Mar. 2021, doi: 10.3389/fcomp.2021.563060.
- [4] Rifqi Mulyawan, "Rule Based System," <https://rifqimulyawan.com/kamus/rule-based-system/>. Accessed: Apr. 21, 2026. [Online]. Available: <https://rifqimulyawan.com/kamus/rule-based-system/>
- [5] B. A. Haryono and A. Prasetya, "Identifikasi Kuantitas Berbasis Rule Pada Masalah Text-to-SQL," *Jurnal Borneo Informatika dan Teknik*

- Komputer*, vol. 5, no. 1, pp. 26–39, Sep. 2025, Accessed: Jun. 13, 2026. [Online]. Available: <http://jurnal.borneo.ac.id/index.php/jbit/article/view/6819>
- [6] W. Bambang Triadi Handaya, “Deteksi Website Phishing Menggunakan Teknik Machine Learning,” *Jurnal Informatika Atma Jogja*, vol. 6, no. 1, pp. 69–80, May 2025, doi: <https://doi.org/10.24002/jiaj.v6i1.11538>.
- [7] P. Septiana Rizky, R. Haiban Hirzi, and U. Hidayatullohman, “Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang,” *J Statistika: Jurnal Ilmiah Teori dan Aplikasi Statistika*, vol. 15, no. 2, pp. 228–236, Dec. 2022, doi: <https://doi.org/10.36456/jstat.vol15.no2.a5548>.
- [8] Y. yusuf, M. Tejha Ananda, F. Defrison, J. Erianto Silitonga, and A. Trilidia Devega, “Teknik Phishing dan Strategi Pencegahannya dalam Sistem Informasi,” *Journal of Innovative and Creativity*, vol. 5, no. 2, pp. 13672–13677, Jul. 2025, doi: <https://doi.org/10.31004/joecy.v5i2.2141>.
- [9] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, “Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network,” *Wirel. Commun. Mob. Comput.*, vol. 2021, 2021, doi: 10.1155/2021/8241104.
- [10] Hanif Abdul Karim Afandi, M. Lazaro Fa. Al-Dzaki, Nurul Qomariasih, and Reza Aulia Wildana, “GuardSurfing: Ekstensi Browser sebagai Alat Bantu Deteksi Website Phishing dengan Metode Klasifikasi XGBoost untuk Deteksi URL Phishing Berbasis Flask Framework,” *Info Kripto*, vol. 19, no. 2, pp. 73–85, Sep. 2025, doi: 10.56706/ik.v19i2.124.
- [11] F. Lailiyah, “Analisis Pengaturan Hukum Tindak Pidana Cyber Crime Phising di Indonesia Berdasarkan Perspektif Hukum Responsif,” *Jurnal Riset Multidisiplin Edukasi*, vol. 3, no. 1, pp. 1216–1230, Jan. 2026, doi: <https://doi.org/10.71282/jurmie.v3i1.1634>.
- [12] Fadli Sutarli Ananta and Kurniawan Shelly, “Peranan Pemerintah Melalui Undang-Undang Perlindungan Data Pribadi dalam Menanggulangi Phising di Indonesia,” *Innovative: Journal of Social Science Research*, vol. 3, no. 2, pp. 4208–4221, May 2023, Accessed: Apr. 25, 2026. [Online]. Available: <https://j-innovative.org/index.php/Innovative/issue/view/6>
- [13] T. Hidayatullah, “Implementasi Algoritma Base-64 Dalam Mengamankan Url (Uniform Resource Locator) Website Layanan Pengaduan Masyarakat Desa Bojongraharja,” *Jurnal Media Infotama*, vol. 18, no. 2, pp. 337–343, Oct. 2022, doi: <https://doi.org/10.37676/jmi.v18i2.2937>.
- [14] N. Fadila and R. Tanamal, “Penerapan Rule-Based Expert System (RBES) Dalam Perancangan Aplikasi Sistem Pakar Untuk Mendiagnosa Penyakit Infeksi Saluran Pernapasan Akut (ISPA) Berbasis Android,” *Jurnal Ilmiah Teknologi Informasi Asia*, vol. 15, no. 2, 2021.
- [15] P. Chyan *et al.*, “Pengantar Machine Learning,” *Penerbit Mifandi Mandiri Digital*, vol. 1, Nov. 2024, Accessed: Apr. 26, 2026. [Online]. Available: <https://jurnal.mifandimandiri.com/index.php/penerbitmmd/issue/view/3>
- [16] Prakash Pugazhandhi, “Phishing URL Detection Dataset,” Kaggle. Accessed: May 07, 2026. [Online]. Available: <https://www.kaggle.com/datasets/prakashpugazhandhi/phishing-url-detection-dataset>