Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei

OPEN ACCESS

https://doi.org/10.63447/jimik.v7i2.1865

Deteksi *Phishing Website* Menggunakan Algoritma *Random Forest* dengan *Hyperparameter Tuning GridSearchCV*

Gabriel Mika Angelo [1*], Evangs Mailoa [2]

[1*,2] Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Provinsi Jawa Tengah, Indonesia.

Email: 672021012@Student.uksw.edu [1*], evangs.mailoa@uksw.edu [2]

Histori Artikel:
Dikirim 19 April 2026; *Diterima dalam bentuk revisi* 15 Mei 2026; *Diterima* 25 Mei 2026; *Diterbitkan* 30 Mei 2026.

Abstrak

Phishing website merupakan salah satu ancaman keamanan siber yang menipu pengguna untuk memperoleh informasi sensitif seperti *username*, *password*, dan data keuangan. Serangan ini umumnya dilakukan dengan membuat halaman web palsu yang menyerupai situs resmi sehingga sulit dikenali oleh pengguna awam. Oleh karena itu, deteksi phishing berbasis URL menjadi penting dalam upaya meningkatkan keamanan pengguna saat mengakses website. Penelitian ini bertujuan untuk meningkatkan akurasi klasifikasi URL phishing dengan menerapkan hyperparameter tuning pada algoritma Random Forest sebagai metode pembelajaran mesin yang populer. Dataset yang digunakan berjumlah 10.000 URL, terdiri dari 5.000 data phishing dan 5.000 data legitimate, yang kemudian diekstraksi menjadi 16 fitur numerik teknis yang relevan. Empat konfigurasi model diuji, terdiri atas satu model default dan tiga model hasil tuning menggunakan GridSearchCV. Hasil pengujian menunjukkan bahwa konfigurasi Grid 2 memberikan performa terbaik dengan akurasi 82,65%, precision 82,70%, recall 82,65%, F1-score 82,62%, ROC-AUC 0,9068, dan PR-AUC 0,8976. Sebagai perbandingan, model default hanya mencapai akurasi 82,40% dan F1-score 82,38%, sedangkan Grid 1 dan Grid 3 menghasilkan performa yang sedikit lebih rendah. Temuan ini menunjukkan bahwa hyperparameter tuning dapat memberikan sedikit peningkatan performa dibandingkan konfigurasi default dalam mendeteksi phishing website, meskipun tidak signifikan.

Kata Kunci: Phishing; Machine Learning; Random Forest; Hyperparameter Tuning; Klasifikasi.

Abstract

Phishing websites are one of the cybersecurity threats that deceive users into disclosing sensitive information such as usernames, passwords, and financial data. These attacks are commonly carried out by creating fake web pages that resemble legitimate websites, making them difficult for ordinary users to identify. Therefore, URL-based phishing detection is important in improving user security when accessing websites. This study aims to improve the accuracy of phishing URL classification by applying hyperparameter tuning to the Random Forest algorithm as a popular machine learning method. The dataset used consists of 10,000 URLs, comprising 5,000 phishing URLs and 5,000 legitimate URLs, which were then extracted into 16 relevant numerical technical features. Four model configurations were evaluated, consisting of one default model and three tuned models using GridSearchCV. The experimental results show that the Grid 2 configuration achieved the best performance with an accuracy of 82.65%, precision of 82.70%, recall of 82.65%, F1-score of 82.62%, ROC-AUC of 0.9068, and PR-AUC of 0.8976. In comparison, the default model only achieved an accuracy of 82.40% and an F1-score of 82.38%, while Grid 1 and Grid 3 produced slightly lower performance. These findings indicate that hyperparameter tuning can provide a slight performance improvement compared to the default configuration in detecting phishing websites, although the difference is not statistically significant.

Keyword: Phishing; Machine Learning; Random Forest; Hyperparameter Tuning; Classification.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

1. Pendahuluan

Phishing website merupakan salah satu bentuk kejahatan siber yang umum ditemukan dalam aktivitas daring (Yang *et al.*, 2022). Serangan ini dilakukan melalui situs palsu yang meniru tampilan situs resmi untuk memperoleh data sensitif dari pengguna, dan dampaknya meliputi kerugian finansial hingga pencurian identitas digital (Hapsari & Pambayun, 2023). Di Indonesia, peningkatan akses internet tidak selalu diikuti dengan kesadaran akan keamanan informasi, sehingga memicu meningkatnya jumlah serangan *phishing* (Prayetno *et al.*, 2026). Untuk mengatasi ancaman tersebut, deteksi *phishing* secara otomatis menjadi solusi penting dalam upaya mitigasi serangan siber. Salah satu pendekatan yang digunakan dalam pengembangan sistem ini adalah algoritma *machine learning*, yang mampu mengenali pola pada struktur URL dan perilaku akses pengguna (Mahmud & Wirawan, 2024). Selain itu, *machine learning* juga dapat memanfaatkan fitur teknis seperti konten halaman web dan metadata jaringan sebagai indikator serangan *phishing* (Ferdita Nugraha *et al.*, 2022). *Random Forest* merupakan salah satu algoritma klasifikasi yang banyak digunakan dalam sistem deteksi karena kemampuannya mengolah data berdimensi tinggi serta ketahanannya terhadap *noise* dan *overfitting* (Nugroho, 2025). Namun, performa model sangat dipengaruhi oleh pemilihan *hyperparameter* yang digunakan, dan konfigurasi *default* sering kali belum optimal untuk memperoleh hasil klasifikasi yang akurat dan efisien (El-Hassani *et al.*, 2024).

Berbagai penelitian telah membuktikan efektivitas algoritma *machine learning* dalam mendeteksi *phishing website*. Nugraha *et al.* menggunakan metode *ensemble* seperti *Random Forest* dan *Stacking* pada *dataset* tidak seimbang dan berhasil mencapai akurasi hingga 96,4%. Mahmud dan Wirawan mengimplementasikan *Random Forest* untuk klasifikasi *phishing* dan memperoleh akurasi sebesar 83,4%. Penelitian lain oleh Suwarno dan Hardjianto mencatat akurasi 94% pada kasus serupa (Suwarno *et al.*, 2024), sedangkan Wahyudi *et al.* menerapkan *Support Vector Machine* (SVM) dengan optimasi *kernel* dan parameter yang menghasilkan akurasi 85,71% (Wahyudi *et al.*, 2022). Namun, sebagian besar studi tersebut belum secara eksplisit mengkaji pengaruh *hyperparameter tuning* terhadap performa model klasifikasi. Padahal, konfigurasi *hyperparameter* pada algoritma *machine learning* dapat memengaruhi kemampuan model dalam mengenali pola data secara optimal, dan penggunaan konfigurasi *default* belum tentu memberikan hasil terbaik pada setiap karakteristik *dataset*. Beberapa penelitian, seperti yang dilakukan Huizen *et al.*, menggunakan teknik seleksi fitur untuk meningkatkan performa SVM, namun belum membandingkan pengaruh konfigurasi parameter terhadap model *default* (Huizen *et al.*, 2025). Oleh karena itu, diperlukan penelitian yang secara khusus mengevaluasi kontribusi *hyperparameter tuning* dalam meningkatkan performa algoritma *Random Forest* pada deteksi *phishing website*.

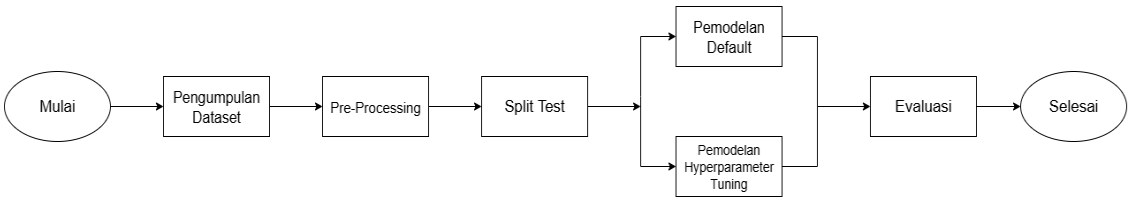
Berdasarkan permasalahan tersebut, penelitian ini bertujuan mengevaluasi efektivitas algoritma *Random Forest* dalam mendeteksi *phishing website* melalui penerapan *hyperparameter tuning* menggunakan *GridSearchCV*. Evaluasi dilakukan dengan membandingkan performa model *default* dan model hasil *tuning* berdasarkan berbagai metrik, meliputi *accuracy*, *precision*, *recall*, *F1-score*, ROC-AUC, dan PR-AUC, serta analisis *confusion matrix* untuk mengamati kesalahan *False Negative* (FN) dan *False Positive* (FP). Penelitian ini juga menyertakan analisis *feature importance* untuk mengidentifikasi fitur yang paling berpengaruh terhadap keputusan model. Hasil evaluasi diharapkan dapat menjadi acuan bagi pengembangan sistem deteksi *phishing* otomatis yang lebih akurat dan efisien dalam bidang teknologi informasi.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode *supervised learning* berbasis algoritma *Random Forest*. Algoritma ini dipilih karena mampu menangani data berdimensi tinggi, memiliki ketahanan terhadap *overfitting*, serta memberikan hasil klasifikasi yang stabil. *Random Forest* bekerja dengan membangun sejumlah pohon keputusan dari *subset* data maupun *subset* fitur secara

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

acak, kemudian menghasilkan prediksi akhir berdasarkan *voting* mayoritas dari setiap pohon (Attanasi & Coburn, 2023). Alur proses penelitian secara umum ditunjukkan pada Gambar 1.



Gambar 1. Alur Proses Penelitian

Dataset yang digunakan dalam penelitian ini diperoleh dari platform *Kaggle* (https://www.kaggle.com/datasets/hassaanmustafavi/phishing-urls-dataset) dengan nama *Phishing URLs Dataset* yang disusun oleh Hassaan Mustafavi. *Dataset* berisi kumpulan URL *phishing* dan *legitimate* yang digunakan untuk proses klasifikasi. Setiap URL diekstraksi menjadi 16 fitur numerik yang merepresentasikan karakteristik teknis tertentu, seperti keberadaan alamat IP, panjang URL, jumlah subdirektori, penggunaan simbol khusus, *iframe*, umur domain, hingga *redirection*. Seluruh fitur dikonversi ke dalam bentuk numerik biner agar dapat diproses oleh algoritma *machine learning*. Sebelum dilakukan pemodelan, data melalui tahap *pre-processing* berupa pemeriksaan nilai kosong serta penghapusan duplikat. *Dataset* ini memiliki keterbatasan karena jumlah data yang relatif kecil dan distribusi kelas yang seimbang, yang berbeda dengan kondisi nyata di mana jumlah situs *phishing* jauh lebih sedikit dibandingkan situs *legitimate* (*imbalanced*). Namun, *dataset* dengan distribusi kelas yang seimbang digunakan untuk memastikan model memperoleh representasi data *phishing* dan *legitimate* secara proporsional selama proses pelatihan dan evaluasi. Pendekatan ini dilakukan agar pengaruh *hyperparameter tuning* terhadap performa *Random Forest* dapat diamati secara lebih terkontrol guna meminimalkan bias akibat dominasi salah satu kelas. Penulis menyadari bahwa distribusi data pada kondisi nyata cenderung bersifat *imbalanced*, sehingga pengujian pada skenario distribusi data yang lebih realistis dapat menjadi arah pengembangan pada penelitian selanjutnya.

Pembagian data dilakukan secara acak menggunakan *random state* = 42 menjadi dua *subset*, yaitu 80% data latih dan 20% data uji. Model klasifikasi dibangun dengan dua pendekatan: pertama, model *Random Forest* dengan konfigurasi *default*; kedua, model *Random Forest* hasil optimasi melalui *hyperparameter tuning*. Proses *tuning* dilakukan menggunakan teknik *GridSearchCV*, yaitu metode optimasi parameter yang menguji seluruh kombinasi nilai parameter yang ditentukan sebanyak tiga *Grid* (Subaşı, 2024). Rentang parameter pada masing-masing *Grid* ditentukan berdasarkan eksplorasi *hyperparameter Random Forest* untuk mengevaluasi pengaruh kompleksitas model terhadap performa klasifikasi *phishing website*. *Grid* 1 digunakan sebagai ruang pencarian parameter yang luas dengan berbagai kombinasi nilai untuk mengeksplorasi pengaruh parameter terhadap performa model. *Grid* 2 dirancang dengan ruang pencarian yang lebih terfokus pada kombinasi parameter yang dianggap potensial, khususnya pada jumlah *estimator* dan kedalaman pohon yang lebih tinggi, untuk mengevaluasi stabilitas serta performa klasifikasi. Sementara itu, *Grid* 3 menggunakan konfigurasi parameter yang lebih sederhana untuk mengamati pengaruh pembatasan kompleksitas model terhadap performa *Random Forest*. Evaluasi tiap kombinasi dilakukan dengan *cross-validation*, di mana data latih dibagi menjadi 5 lipatan (*fold*) dan model dilatih serta diuji secara bergantian pada tiap lipatan. Total kombinasi parameter yang diuji pada seluruh *Grid* sebanyak 522 kombinasi, sehingga hasil evaluasi lebih konsisten dan tidak bergantung pada satu pembagian data tertentu (Yates *et al.*, 2023). Parameter yang diuji meliputi jumlah pohon (*n_estimators*), kedalaman pohon (*max_depth*), jumlah minimal sampel pada percabangan (*min_samples_split*), jumlah minimal sampel pada daun (*min_samples_leaf*), pemilihan fitur (*max_features*), dan opsi *bootstrap*.

Evaluasi kinerja model dilakukan menggunakan enam metrik utama, yaitu *accuracy*, *precision*, *recall*, *F1-score*, ROC-AUC, dan PR-AUC. *Accuracy* menunjukkan proporsi prediksi yang benar dari keseluruhan data; *precision* menggambarkan ketepatan model dalam mengidentifikasi URL *phishing*;

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

recall mengukur seberapa banyak data *phishing* yang berhasil dikenali; dan *F1-score* memberikan rata-rata harmonik antara *precision* dan *recall*. Sementara itu, ROC-AUC (*Receiver Operating Characteristic – Area Under Curve*) digunakan untuk mengukur kemampuan model dalam membedakan kelas *phishing* dan *legitimate* secara keseluruhan, sedangkan PR-AUC (*Precision-Recall Area Under Curve*) dipilih karena lebih representatif pada data klasifikasi biner. Perhitungan seluruh metrik didasarkan pada *confusion matrix* dengan komponen *true positive* (TP), *true negative* (TN), *false positive* (FP), dan *false negative* (FN), yang juga digunakan untuk menilai dampak kesalahan klasifikasi model. Formula perhitungan masing-masing metrik ditunjukkan pada persamaan (1) hingga (4).

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

Selain itu, dilakukan analisis *feature importance* untuk mengidentifikasi kontribusi relatif setiap fitur terhadap keputusan model, sehingga dapat memberikan pemahaman lebih lanjut mengenai perilaku model (Berrar, 2021; Pensa *et al.*, 2025). Evaluasi ini juga dilengkapi dengan visualisasi *learning curve* untuk memeriksa kecenderungan model mengalami *overfitting* atau *underfitting* selama proses pelatihan. Dengan alur dan tahapan penelitian tersebut, dapat dinilai sejauh mana penerapan *hyperparameter tuning* berkontribusi dalam meningkatkan efektivitas algoritma *Random Forest* untuk mendeteksi *phishing website* berbasis URL.

3. Hasil dan Pembahasan

3.1 Hasil

Setiap URL diekstraksi menjadi enam belas fitur numerik yang merepresentasikan karakteristik teknis tertentu, seperti keberadaan alamat IP dalam URL, panjang URL, jumlah subdirektori, penggunaan simbol khusus, *iframe*, umur domain, hingga *redirection*. Seluruh fitur kemudian dikonversi ke dalam bentuk numerik biner agar dapat diproses oleh algoritma *machine learning*. Tabel 1 menyajikan uraian masing-masing fitur yang digunakan dalam penelitian ini. *Dataset* terdiri dari sejumlah URL dengan distribusi seimbang antara kategori *phishing* (1) dan *legitimate* (0).

Tabel 1. Ekstraksi Fitur URL

Nama Fitur	Deskripsi
Have_Ip	1 jika URL mengandung IP address langsung, 0 jika tidak.
Have_At	1 jika URL mengandung simbol '@', yang sering digunakan untuk mengelabui pengguna.
URL_Length	Panjang karakter dari URL. URL yang sangat panjang biasanya mencurigakan.
URL_Depth	Jumlah direktori dalam URL setelah domain utama. Semakin dalam, semakin mencurigakan.
Redirection	1 jika terdapat penggunaan '//' secara berlebihan di luar protokol.
https_Domain	1 jika domain URL mengandung kata "http" atau "https".
TinyURL	1 jika URL menggunakan layanan pemendek URL seperti bit.ly, tinyurl, dll.
Prefix/Suffix	1 jika domain mengandung tanda minus (-), yang umum pada domain palsu.
DNS_Record	1 jika domain tidak memiliki DNS record (mencurigakan).

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

Web_Traffic	1 jika *traffic* web sangat rendah atau tidak tersedia.
Domain_Age	1 jika domain baru dibuat (biasanya < 6 bulan).
Domain_End	1 jika masa aktif domain akan segera habis.
iFrame	1 jika halaman menggunakan tag *iframe* tersembunyi.
Mouse_Over	1 jika terdapat aksi mencurigakan saat *mouse* diarahkan ke elemen tertentu.
Right_Click	1 jika fungsi klik kanan dinonaktifkan di halaman.
Web_Forwards	Jumlah redireksi otomatis dalam halaman sebelum mencapai tujuan.

Setelah proses ekstraksi fitur, penelitian dilanjutkan dengan eksperimen *hyperparameter tuning* menggunakan *GridSearchCV*. Tabel 2 menunjukkan bahwa pengujian dilakukan dengan membandingkan model *Random Forest* default dengan tiga model hasil *tuning*. Parameter yang diuji meliputi *n_estimators*, *max_depth*, *min_samples_split*, *min_samples_leaf*, *max_features*, serta *bootstrap*.

Tabel 2. Kombinasi Parameter

Hyperparameter	Grid 1	Grid 2	Grid 3
n_estimators	100, 200, 300	300, 400, 500	50, 100, 150
max_depth	None, 10, 20, 30	30, 40, None	5, 10, 15
min_samples_split	2, 5, 10	2, 5	10, 20
min_samples_leaf	1, 2, 4	1	5, 10
max_features	sqrt, log2	sqrt	log2
bootstrap	True, False	True	True, False

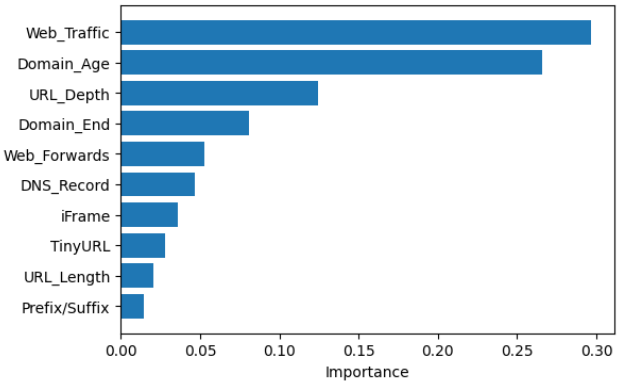
Evaluasi dilakukan terhadap empat model, yaitu *Random Forest* default dan tiga model hasil *hyperparameter tuning* menggunakan *GridSearchCV* (Grid 1, Grid 2, dan Grid 3). Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, ROC-AUC, dan PR-AUC. Hasil pengujian disajikan pada Tabel 3.

Tabel 3. Hasil Evaluasi Performa Model

Model	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC
Default	0,8240	0,8243	0,8240	0,8238	0,9044	0,8953
Grid 1	0,8175	0,8185	0,8175	0,8176	0,9065	0,8910
Grid 2	0,8265	0,8270	0,8265	0,8262	0,9068	0,8976
Grid 3	0,8215	0,8219	0,8215	0,8216	0,9062	0,8926

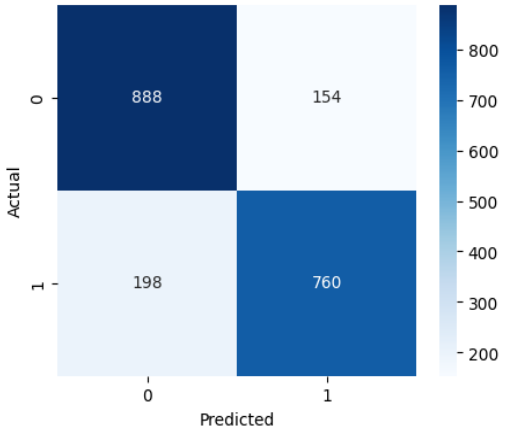
Hasil pengujian menunjukkan bahwa model dengan konfigurasi Grid 2 memiliki performa tertinggi dibandingkan model lainnya, dengan nilai *F1-score* sebesar 0,8262, ROC-AUC sebesar 0,9068, dan PR-AUC sebesar 0,8976. Sementara itu, model *default* menghasilkan nilai *F1-score* 0,8238 dan ROC-AUC 0,9044, sedangkan Grid 1 dan Grid 3 justru menunjukkan penurunan kinerja dibandingkan model *default*. Selain evaluasi performa, dilakukan pula analisis *feature importance* untuk mengetahui kontribusi masing-masing fitur terhadap hasil klasifikasi. Hasil menunjukkan bahwa terdapat beberapa fitur yang memiliki tingkat kepentingan jauh lebih tinggi dibandingkan fitur lainnya, di antaranya adalah Web_Traffic, Domain_Age, URL_Depth, Domain_End, Web_Forwards, DNS_Record, iFrame, TinyURL, URL_Length, dan Prefix/Suffix, sebagaimana ditunjukkan pada Gambar 2.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865



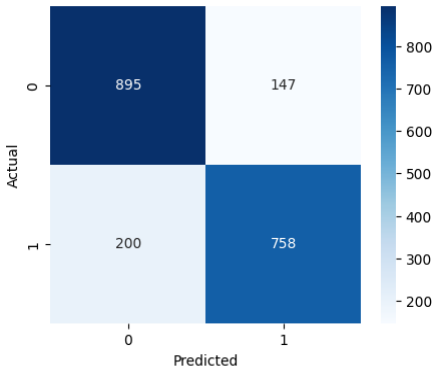
Gambar 2. *Feature Importance*

Analisis *confusion matrix* juga dilakukan untuk memahami pola kesalahan prediksi pada masing-masing model. Pada model *Random Forest* default, sebagaimana ditunjukkan pada Gambar 4, sebanyak 888 data *phishing* berhasil diklasifikasikan dengan benar (*True Positive*) dan 154 data *phishing* salah diklasifikasikan sebagai *legitimate* (*False Negative*). Untuk data *legitimate*, sebanyak 760 berhasil diklasifikasikan dengan benar (*True Negative*) dan 198 salah diklasifikasikan sebagai *phishing* (*False Positive*).



Gambar 3. *Confusion Matrix* (*Default*)

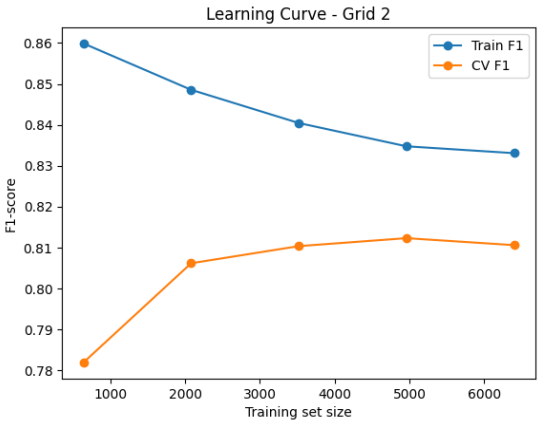
Pada model hasil *tuning* terbaik (Grid 2), sebagaimana ditunjukkan pada Gambar 4, jumlah *True Positive* meningkat menjadi 895 dan *False Negative* menurun menjadi 147, sedangkan *False Positive* sedikit meningkat menjadi 200 dan *True Negative* sedikit menurun menjadi 758.



Gambar 4. *Confusion Matrix* (Grid 2)

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

Visualisasi *learning curve* juga dilakukan untuk mengevaluasi kecenderungan model terhadap gejala *overfitting* atau *underfitting*. Pada model terbaik (Grid 2), sebagaimana ditunjukkan pada Gambar , terlihat bahwa nilai F1 pada data pelatihan awalnya tinggi kemudian menurun secara bertahap seiring bertambahnya data pelatihan, sedangkan nilai F1 pada data validasi awalnya rendah kemudian meningkat dan mendekati nilai pada data pelatihan.



Gambar 5. *Learning Curve* (Grid 2)

3.2 Pembahasan

Perbedaan performa antar model relatif kecil, yaitu berada pada rentang 0,26% hingga 0,33% atau kurang dari 1%, sehingga dapat disimpulkan bahwa *hyperparameter tuning* tidak selalu menjamin peningkatan hasil, terutama bila ruang pencarian parameter yang digunakan terlalu sempit. Nilai ROC-AUC dan PR-AUC yang tinggi pada seluruh model juga menunjukkan kemampuan diskriminatif yang baik dari algoritma *Random Forest* pada data yang digunakan. Untuk mengevaluasi apakah perbedaan performa antar model bersifat signifikan secara statistik, dilakukan pengujian menggunakan *paired t-test* terhadap hasil *5-fold cross-validation* antara model *default* dan model hasil *tuning* terbaik. Hasil pengujian menunjukkan bahwa perbedaan performa yang diperoleh tidak signifikan secara statistik ($p > 0,05$). Hal ini menunjukkan bahwa peningkatan performa yang dihasilkan oleh proses *hyperparameter tuning* kemungkinan masih dipengaruhi oleh variasi acak.

Hasil pengujian juga menunjukkan bahwa setiap *Grid* memiliki performa dan waktu pelatihan yang berbeda. Grid 1 memerlukan waktu pelatihan paling lama, yaitu sekitar 25 menit, karena memiliki ruang pencarian parameter yang luas dengan total 432 kombinasi parameter. Meskipun mengevaluasi lebih banyak konfigurasi, performa yang dihasilkan belum optimal sehingga menunjukkan bahwa ruang pencarian yang terlalu luas tidak selalu menghasilkan kombinasi parameter terbaik. Grid 2 menghasilkan performa terbaik dengan waktu pelatihan sekitar 2 menit karena menggunakan ruang pencarian parameter yang lebih terarah dengan total 18 kombinasi parameter, sehingga proses *tuning* menjadi lebih efisien. Meskipun peningkatan performa yang dihasilkan tidak signifikan secara statistik, Grid 2 menunjukkan kecenderungan memberikan keseimbangan yang lebih baik antara performa klasifikasi dan efisiensi waktu pelatihan dibandingkan *Grid* lainnya. Sementara itu, Grid 3 memerlukan waktu pelatihan paling singkat, yaitu sekitar 1 menit, karena menggunakan konfigurasi parameter yang lebih sederhana dengan total 72 kombinasi parameter. Namun, pembatasan parameter pada Grid 3 menyebabkan kemampuan model dalam mempelajari pola data menjadi lebih terbatas dan berpotensi menyebabkan *underfitting*. Berdasarkan hasil tersebut, ruang pencarian parameter yang terlalu luas maupun terlalu sederhana dapat memengaruhi performa dan efisiensi *tuning Random Forest*.

Dominasi fitur Web_Traffic dan Domain_Age menunjukkan bahwa tingkat popularitas *website* dan usia domain memiliki pengaruh penting dalam proses deteksi *phishing website*. *Website phishing* umumnya memiliki *traffic* yang rendah dan menggunakan domain yang relatif baru karena bersifat sementara untuk menghindari deteksi, sedangkan *website legitimate* cenderung memiliki jumlah

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

pengunjung yang lebih stabil serta usia domain yang lebih lama. Temuan ini menunjukkan bahwa fitur terkait reputasi dan usia domain berperan penting dalam membantu model *Random Forest* membedakan *website phishing* dan *legitimate*.

Perbedaan pada *confusion matrix* antara model *default* dan Grid 2 menunjukkan bahwa *tuning* pada Grid 2 sedikit meningkatkan kemampuan model dalam mengenali situs *phishing* (penurunan *False Negative*), tetapi di sisi lain sedikit menurunkan ketepatan dalam mengenali situs *legitimate* (kenaikan *False Positive*). Dalam konteks deteksi *phishing*, penurunan *False Negative* lebih diutamakan karena kesalahan jenis ini berpotensi membiarkan serangan *phishing* lolos tanpa terdeteksi, sedangkan kesalahan *False Positive* hanya menimbulkan ketidaknyamanan karena situs yang sebenarnya aman dapat ditandai sebagai *phishing*. Dengan demikian, meskipun peningkatan kinerja Grid 2 tidak terlalu besar secara keseluruhan, model ini menunjukkan distribusi kesalahan yang lebih seimbang untuk deteksi *phishing*. *Accuracy* sebesar 82,65% menunjukkan bahwa *hyperparameter tuning* pada model *Random Forest* dapat meningkatkan hasil klasifikasi *phishing website*. Namun, untuk implementasi pada sistem deteksi *phishing* di lingkungan nyata, performa model masih perlu ditingkatkan karena kesalahan klasifikasi pada sistem keamanan dapat berdampak pada risiko keamanan pengguna. Hasil penelitian ini juga masih berada di bawah penelitian Mahmud & Wirawan sebesar 83,4% serta Nugraha *et al.* sebesar 96,4%. Perbedaan performa tersebut dapat dipengaruhi oleh variasi *dataset*, jumlah fitur, metode *preprocessing*, distribusi data, serta konfigurasi model yang digunakan pada masing-masing penelitian. Penelitian Nugraha *et al.*, misalnya, menggunakan *dataset phishing* dengan karakteristik *imbalanced* dan menerapkan pendekatan *ensemble learning* yang berbeda, sehingga menghasilkan performa yang lebih tinggi. Oleh karena itu, hasil performa antar penelitian tidak dapat dibandingkan secara langsung tanpa mempertimbangkan perbedaan kondisi eksperimen dan metode yang digunakan. Meskipun demikian, hasil penelitian ini menunjukkan bahwa *hyperparameter tuning* tetap memberikan pengaruh terhadap performa dan perilaku model *Random Forest* dalam proses klasifikasi *phishing website*.

4. Kesimpulan

Berdasarkan hasil penelitian, penerapan *hyperparameter tuning* menggunakan *GridSearchCV* pada algoritma *Random Forest* menunjukkan sedikit peningkatan performa dalam mendeteksi *phishing website*, meskipun perbedaannya tidak signifikan secara statistik berdasarkan hasil *paired t-test*. Model terbaik (Grid 2) memperoleh nilai *accuracy* 0,8265, *precision* 0,8270, *recall* 0,8265, *F1-score* 0,8262, ROC-AUC 0,9068, dan PR-AUC 0,8976, sedikit lebih tinggi dibandingkan model *default* serta mampu menurunkan jumlah kesalahan *False Negative* (FN) yang lebih berisiko dalam konteks keamanan.

Hasil penelitian juga menunjukkan bahwa pemilihan ruang pencarian parameter memiliki pengaruh penting terhadap performa dan efisiensi proses *tuning*. Grid 1 dengan ruang pencarian parameter yang terlalu luas memerlukan waktu pelatihan lebih lama tanpa menghasilkan performa optimal, sedangkan Grid 3 dengan konfigurasi parameter yang lebih sederhana menyebabkan performa model menjadi lebih rendah dan berpotensi mengalami *underfitting*. Analisis *feature importance* menunjukkan bahwa fitur berbasis reputasi seperti Web_Traffic dan Domain_Age memiliki pengaruh dominan terhadap hasil prediksi, sementara *learning curve* pada Grid 2 memperlihatkan bahwa model tidak mengalami *overfitting* maupun *underfitting*.

Temuan ini menunjukkan bahwa proses *hyperparameter tuning* tidak selalu menghasilkan performa yang lebih baik dibandingkan konfigurasi *default* apabila parameter yang digunakan kurang sesuai dengan karakteristik data. Oleh karena itu, *GridSearchCV* masih layak digunakan dalam proses optimasi model, namun pemilihan kombinasi parameter yang tepat tetap menjadi faktor penting dalam menghasilkan performa model yang optimal. Meskipun demikian, hasil penelitian ini masih dipengaruhi oleh keterbatasan *dataset* yang berukuran kecil dan seimbang, sehingga penelitian selanjutnya disarankan menggunakan *dataset* yang lebih besar dan mencerminkan distribusi data dunia

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

nyata yang tidak seimbang (*imbalanced*), serta mengeksplorasi pendekatan optimasi *hyperparameter* lain seperti *RandomizedSearchCV* atau *Bayesian Optimization*.

5. Ucapan Terima Kasih

Artikel disusun sebagai tugas akhir dalam rangka penyelesaian Program Studi S1 Teknik Informatika di Universitas Kristen Satya Wacana. Penulis menyampaikan ucapan terima kasih kepada dosen pembimbing yang telah memberikan bimbingan, saran, kritik, serta motivasi selama proses penyusunan hingga artikel ini dapat diselesaikan dengan baik. Selain itu, penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman yang telah memberikan dukungan moral dan semangat kepada penulis. Penulis berharap agar karya ilmiah ini dapat memberikan manfaat bagi masyarakat luas serta menjadi sumber pengetahuan yang berguna di bidang terkait.

6. Daftar Pustaka

Attanasi, E. D., & Coburn, T. C. (2023). Random forest. Dalam *Encyclopedia of mathematical geosciences* (hlm. 1182–1185). Springer. https://doi.org/10.1007/978-3-030-85040-1_265

Berrar, D. (2021). Cross-validation. Dalam *Encyclopedia of bioinformatics and computational biology* (Vol. 1, hlm. 542–545). Elsevier.

El-Hassani, F. Z., Amri, M., Joudar, N. E., & Haddouch, K. (2024). A new optimization model for MLP hyperparameter tuning: Modeling and resolution by real-coded genetic algorithm. *Neural Processing Letters*, *56*(2), 1–31. https://doi.org/10.1007/s11063-024-11578-0

Hapsari, R. D., & Pambayun, K. G. (2023). Ancaman cybercrime di Indonesia: Sebuah tinjauan pustaka sistematis. *Jurnal Konstituen*, *5*(1), 1–17. https://doi.org/10.33701/jk.v5i1.3208

Huizen, L. M., Ardima, M. B., & Idris, M. (2025). Meningkatkan kinerja SVM: Dampak berbagai teknik seleksi fitur pada akurasi prediksi. *Aiti*, *22*(1), 1–14. https://doi.org/10.24246/aiti.v22i1.1-14

Mahmud, A. F., & Wirawan, S. (2024). Deteksi phishing website menggunakan machine learning metode klasifikasi. *Sistemasi: Jurnal Sistem Informasi*, *13*(4). http://sistemasi.ftik.unisi.ac.id

Nugraha, A. F., Aziza, R. F. A., & Pristyanto, Y. (2022). Penerapan metode stacking dan random forest untuk meningkatkan kinerja klasifikasi pada proses deteksi web phishing. *Jurnal Infomedia*, *7*(1), 39. https://doi.org/10.30811/jim.v7i1.2959

Nugroho, M. W. (2025). Analisis performa algoritma random forest dalam mengatasi overfitting pada model prediksi. *Jurnal Teknologi Informasi dan Komputer*, *9*(4). https://doi.org/10.35870/jtik.v9i4.4236

Pensa, R. G., Crombach, A., Peignier, S., & Rigotti, C. (2025). Explaining random forest and XGBoost with shallow decision trees by co-clustering feature importance. *Machine Learning*, *114*(12). https://doi.org/10.1007/s10994-025-06932-9

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 7 No. 2 (2026) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v7i2.1865

Prayetno, F. M., Riski, F., & Safitri, D. L. A. (2026). Analisis Keamanan Siber Pada Sistem Elektronik Berbasis Perspektif Jaringan Komputer Dan Ketentuan Bssn: Studi Pada Imbauan Phishing dan Pencurian Kredensial: Studi pada Imbauan Phishing dan Pencurian Kredensial. *Jurnal Teknologi Informasi: Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, *20*(1), 95-100 https://doi.org/10.47111/jti.v20i1.23404

Subaşı, N. (2024). Comprehensive analysis of grid and randomized search on dataset performance. *European Journal of Engineering and Applied Sciences*, *7*(2), 77–83. https://doi.org/10.55581/ejeas.1581494

Suwarno, D. B., & Hardjianto, M. (2024). Deteksi website phishing dari analisis URL menggunakan algoritma random forest. *Jurnal Teknik Informatika*, *21*(2), 145–152. https://doi.org/10.36080/bit.v21i2.3603

Wahyudi, D., Niswar, M., & Alimuddin, A. A. P. (2022). Website phishing detection application using support vector machine (SVM). *Journal of Information Technology and Its Utilization*, *5*(1), 18–24. https://doi.org/10.56873/jitu.5.1.4836

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, *116*, Article 102675. https://doi.org/10.1016/j.cose.2022.102675

Yates, L. A., Aandahl, Z., Richards, S. A., & Brook, B. W. (2023). Cross validation for model selection: A review with examples from ecology. *Ecological Monographs*, *93*(1). https://doi.org/10.1002/ecm.1557.