Progresif: Jurnal Ilmiah Komputer
https://ojs.stmik-banjarbaru.ac.id/index.php/progresif/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0877

Penerapan Algoritma *Random Forest* Berbasis *Shap Feature Importance* dan *GridsearchCV* Untuk Deteksi Phishing

DOI: http://dx.doi.org/10.35889/progresif.v22i1.3345

Samuel Effendi Pratama[1*], Daniel Udjulawa[2]
Informatika, Universitas Multi Data Palembang, Palembang, Indonesia
*e-mail *Corresponding Author:* samueleffendipratama_2226250024@mhs.mdp.ac.id

Abstract

The rapid growth of internet users in Indonesia has increased the risk of cyberattacks, particularly phishing. Phishing is a digital fraud attempt that disguises links to resemble official websites in order to steal users' sensitive information. This study aims to develop a phishing link detection model using a machine learning approach. The dataset consists of 11,430 URL entries from Mendeley Data, including features such as URL length, suspicious symbols, and subdomain levels. The Random Forest algorithm was chosen for its ability to handle high-dimensional data and resist overfitting. Feature selection was performed using SHAP (Shapley Additive Explanations) to assess feature contributions, while model optimization was conducted with GridSearchCV. The best configuration, RF + GS + SHAP Threshold-P10, achieved an accuracy of 0.9650 and an F1-score of 0.9651, producing an accurate, efficient, and interpretable phishing detection model.

Keywords: *Phishing; Random Forest; GridSearchCV; SHAP; Machine Learning*

Abstrak

Pesatnya pertumbuhan pengguna internet di Indonesia meningkatkan risiko serangan siber, salah satunya *phishing*. *Phishing* merupakan upaya penipuan digital dengan menyamarkan tautan agar menyerupai situs resmi untuk mencuri informasi sensitif pengguna. Penelitian ini bertujuan membangun model deteksi tautan *phishing* menggunakan pendekatan *machine learning*. Dataset yang digunakan berisi 11.430 entri URL dari Mendeley Data, mencakup fitur seperti panjang URL, simbol mencurigakan, dan tingkat subdomain. Algoritma *random forest* dipilih karena mampu menangani data berdimensi tinggi serta tahan terhadap *overfitting*. Seleksi fitur dilakukan dengan *SHAP (Shapley Additive Explanations)* untuk menilai kontribusi fitur, sedangkan optimasi parameter model menggunakan *GridSearchCV*. Hasil penelitian menunjukkan konfigurasi *RF + GS + SHAP Threshold-P10* memberikan akurasi 0,9650 dan *F1-score* 0,9651, menghasilkan model yang akurat, efisien, dan transparan dalam mendeteksi tautan *phishing*.

Kata kunci: *Phishing*; *Random Forest*; *GridSearchCV*; *SHAP*; *Machine Learning*

1. Pendahuluan

Perkembangan teknologi informasi yang pesat telah meningkatkan aktivitas digital masyarakat, mulai dari transaksi, komunikasi, hingga layanan publik. Hingga awal 2025, Indonesia memiliki lebih dari 212 juta pengguna internet dan 143 juta pengguna media sosial, menunjukkan tingginya intensitas penggunaan *platform* digital [1][2]. Kondisi ini turut meningkatkan paparan terhadap serangan *phishing*, di mana lebih dari 80% pengguna global pernah menerima tautan mencurigakan [3]. Kaspersky juga melaporkan 893 juta upaya phishing pada tahun 2024, meningkat 26% dari tahun sebelumnya [4], sehingga menegaskan bahwa phishing merupakan ancaman keamanan siber yang semakin serius di era digital saat ini [5].

Phishing merupakan upaya penipuan dengan menyamarkan halaman atau tautan agar tampak seperti situs resmi dengan tujuan memperoleh informasi sensitif pengguna [6]. Salah satu bentuk yang paling sering ditemui adalah *link phishing*, yaitu *URL* yang dimodifikasi sehingga terlihat mirip dengan alamat asli dan kerap mengecoh pengguna yang tidak teliti dalam memeriksa struktur URL [7]. Pola penyamaran ini menimbulkan kerugian signifikan bagi pengguna dan organisasi, sebagaimana dilaporkan *IBM Cost of a Data Breach Report* 2023 bahwa kerugian rata-rata akibat serangan siber dapat mencapai *USD* 5,9 juta per insiden [8]. Permasalahan semakin kompleks karena metode deteksi tradisional seperti *blacklist* hanya mampu mengenali situs yang telah tercatat sebelumnya, sehingga banyak variasi domain baru gagal teridentifikasi [9]. Kondisi ini menunjukkan adanya masalah berupa keterbatasan sistem deteksi konvensional dalam menghadapi pola serangan *phishing* yang terus berkembang.

Untuk mengatasi keterbatasan metode deteksi phishing tradisional, penelitian ini mengusulkan pemanfaatan pendekatan *machine learning* yang mampu menganalisis pola teknis pada URL secara otomatis [10]. Sejumlah studi menunjukkan bahwa proses *feature selection* dan *hyperparameter tuning* berperan penting dalam meningkatkan kinerja dan stabilitas model, di mana *GridSearchCV* dapat membantu memperoleh konfigurasi parameter yang paling optimal [11]. *Random Forest* dipilih sebagai algoritma utama karena sifatnya sebagai metode *ensemble* berbasis *bagging* yang mampu menangani variasi fitur secara lebih efektif serta menghasilkan model yang *robust* terhadap kompleksitas data [12]. Selain itu, metode *SHAP* digunakan untuk memberikan interpretasi yang lebih komprehensif terhadap kontribusi setiap fitur, sebagaimana dibuktikan dalam penelitian Puspanagara yang menekankan pentingnya transparansi model dalam proses pengambilan keputusan berbasis data [13]. Temuan lain juga menyatakan bahwa *Random Forest* memiliki keunggulan dibandingkan metode klasik seperti *Naïve Bayes* dan *Decision Tree* dalam hal ketahanan dan konsistensi performa pada berbagai skenario data [14]. Dalam penelitian ini, digunakan kombinasi algoritma *Random Forest* yang dioptimasi dengan *hyperparameter tuning GridSearchCV* dan teknik *Feature Importance SHAP*. Pendekatan ini dipilih karena ketiga metode tersebut saling melengkapi dalam meningkatkan ketahanan model terhadap variasi pola *URL* sekaligus memberikan transparansi yang dibutuhkan dalam proses interpretasi hasil klasifikasi.

Penelitian ini bertujuan untuk mengembangkan model deteksi *link phishing* menggunakan algoritma *Random Forest* yang dioptimalkan melalui *GridSearchCV* dan seleksi fitur berbasis *SHAP*. Secara teoritis, penelitian ini berkontribusi pada pengembangan metode deteksi *phishing* yang lebih efektif melalui analisis fitur *URL*. Secara praktis, penelitian ini diharapkan dapat membantu pengguna digital dan penyedia layanan daring dalam mengurangi risiko pencurian data akibat serangan *phishing* sehingga mendukung peningkatan keamanan siber secara lebih luas.

2. Tinjauan Pustaka

Penelitian yang dilakukan oleh Saputra et al. [11] menunjukkan bahwa penerapan *feature selection* dan *hyperparameter tuning* meningkatkan performa model klasifikasi dalam mendeteksi *phishing*. Studi tersebut melaporkan peningkatan akurasi dari di bawah 95% menjadi 96,28%, dan meningkat lagi menjadi 96,77% setelah optimasi parameter menggunakan *GridSearchCV*. Penelitian tersebut juga menerapkan perhitungan *feature importance MDI-OOB* serta evaluasi menggunakan metrik *ROC-AUC*.

Selanjutnya, *Windarni et al* [14] menggunakan teknik seleksi fitur *Pearson Correlation* dalam pendeteksian *website phishing*. Studi tersebut menerapkan tiga algoritma klasifikasi, yaitu *Naïve Bayes*, *Decision Tree*, dan *Random Forest*. Hasil penelitian menunjukkan bahwa *Naïve Bayes* memperoleh akurasi 60,4%, *Decision Tree* 94,4%, dan *Random Forest* 96,3%.

Penelitian yang dilakukan oleh Lukito dan Handaya [10] menguji beberapa algoritma *machine learning* untuk deteksi phishing berbasis *URL* pada *website*. Algoritma yang dievaluasi meliputi *Decision Tree*, *Random Forest*, dan *XGBoost* dengan penerapan *hyperparameter tuning*. Hasil penelitian menunjukkan bahwa model *XGBoost* dengan optimasi parameter memberikan performa terbaik dengan tingkat akurasi mencapai 96%, lebih tinggi dibandingkan *Random Forest* dan *Decision Tree*. Temuan ini mengindikasikan bahwa proses optimasi parameter memiliki peran penting dalam meningkatkan kinerja model klasifikasi *phishing*.

Selanjutnya, Kencana et al [15] mengimplementasikan sistem deteksi link phishing berbasis *Random Forest* menggunakan dataset sebanyak 2.457 data *URL*. Hasil pengujian menunjukkan bahwa model *Random Forest* mampu mencapai akurasi sebesar 94,36% serta nilai

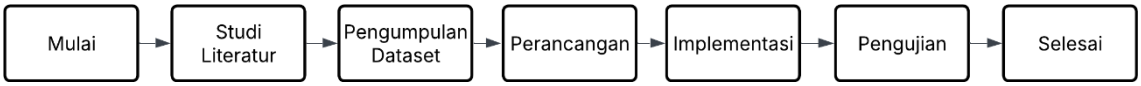
validasi sebesar 94,77%, yang menunjukkan stabilitas model dalam mengklasifikasikan situs phishing dan *non-phishing*. Meskipun demikian, penelitian tersebut belum mengintegrasikan proses *hyperparameter tuning* maupun metode interpretabilitas fitur untuk memahami kontribusi masing-masing atribut *URL* terhadap keputusan model.

Penelitian oleh Kalla dan Kuraku [16] melakukan evaluasi komparatif terhadap berbagai algoritma *machine learning*, termasuk Regresi Logistik, *K-Nearest Neighbors*, *Decision Tree*, *Random Forest*, *Support Vector Classifier (SVC), Linear SVC*, dan *Naïve Bayes* dalam konteks deteksi *phishing*. Hasil penelitian menunjukkan bahwa *Linear SVC* dan *Random Forest* memberikan performa paling unggul dibandingkan algoritma lainnya. Studi ini juga menegaskan bahwa pendekatan *machine learning* sangat efektif untuk deteksi *phishing* dan masih berpotensi dikembangkan lebih lanjut melalui penggunaan *ensemble model* serta integrasi teknik *Natural Language Processing (NLP)*.

Berbeda dari penelitian sebelumnya yang menerapkan *Random Forest*, seleksi fitur, atau optimasi parameter secara terpisah, penelitian ini mengintegrasikan *Random Forest*, *GridSearchCV*, dan *SHAP* dalam satu alur deteksi *phishing* berbasis *URL*. Integrasi ini memungkinkan pemilihan parameter optimal, pemodelan yang lebih stabil, serta interpretasi kontribusi fitur secara langsung. Pendekatan terpadu tersebut menjadi kebaruan utama yang belum diterapkan secara simultan pada studi-studi terdahulu, khususnya dalam konteks deteksi *phishing*.

3. Metodologi

Penelitian ini menerapkan pendekatan iteratif, di mana tahap pra-pemrosesan, pelatihan, evaluasi, dan perbaikan model dilakukan secara berulang. Pendekatan ini memberikan fleksibilitas dalam pengembangan model, memungkinkan penyesuaian parameter atau fitur ketika ditemukan kelemahan pada hasil evaluasi. Tahapan metodologi penelitian dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metodologi Penelitian

3.1. Studi Literatur

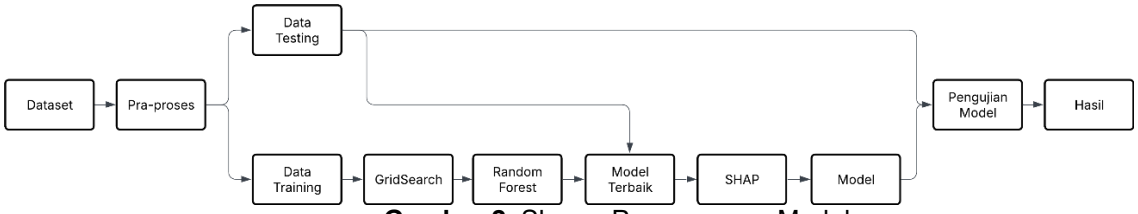
Tahapan studi literatur diawali dengan pembacaan jurnal-jurnal terdahulu yang berfokus pada topik pendeteksian link phishing dengan menggunakan *Random Forest* dengan teknik seleksi fitur *SHAP* dan optimasi *GridSearchCV*.

3.2. Pengumpulan Dataset

Pada tahap ini, dilakukan proses pengumpulan dataset berupa dataset deteksi phishing yang bersifat publik dan diperoleh dari Mendeley Data. Dataset yang digunakan berjumlah 11.430 URL, terdiri dari 50% *URL phishing* dan 50% *URL legitimate*, dengan masing-masing data telah dilengkapi label yang valid untuk keperluan klasifikasi.

3.3. Perancangan

Pada tahap ini, sistem deteksi phishing dirancang menggunakan algoritma *Random Forest* yang dioptimasi melalui *GridSearchCV* dan didukung oleh seleksi fitur berbasis *SHAP* (*SHapley Additive exPlanations*).



Gambar 2. Skema Perancangan Model

3.3.1. Pembersihan Data

Tahap pembersihan data dilakukan untuk memastikan kualitas dataset sebelum proses pelatihan model. Proses ini mencakup penghapusan atribut yang tid`ak relevan seperti kolom *URL*, pemeriksaan serta penanganan nilai kosong (*missing values*), dan penghapusan data duplikat. Setelah tahap ini, seluruh data dinyatakan bersih dan siap digunakan untuk tahap pra-pemrosesan berikutnya.

3.3.2. Normalisasi Data

Tahap normalisasi data dilakukan untuk menyeragamkan skala antar fitur numerik agar seluruh atribut berada pada rentang nilai yang sama. Proses ini bertujuan menghindari dominasi fitur dengan nilai besar terhadap fitur lain yang bernilai kecil. Normalisasi dilakukan menggunakan metode *Min-Max Scaling* yang mengubah nilai setiap fitur ke dalam rentang [0, 1] dengan rumus seperti pada persamaan 1 berikut:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

Keterangan:
X'= nilai hasil normalisasi
X= nilai asli fitur
$X_{\min}$, $X_{\max}$= nilai minimum dan maksimum dari fitur yang bersangkutan

Dengan penerapan *Min-Max Scaling*, seluruh atribut numerik memiliki skala yang seragam sehingga model dapat melakukan pembelajaran secara lebih optimal dan stabil.

3.3.3. Pembagian Dataset

Dataset dibagi menjadi dua bagian, yaitu 70% untuk data latih (*training set*) dan 30% untuk data uji (*testing set*). Pembagian dilakukan menggunakan metode *stratified split* agar proporsi kelas pada data latih dan data uji tetap seimbang.

3.3.4. *Hyperparameter Tuning*

Tahap *hyperparameter tuning* dilakukan untuk memperoleh konfigurasi parameter terbaik pada algoritma random forest agar performa model menjadi optimal. Proses optimasi dilakukan menggunakan metode *GridSearchCV* dengan skema 5-*fold cross validation*, yang menguji beberapa kombinasi parameter secara sistematis.

Parameter yang diuji meliputi:

- Jumlah pohon (*n_estimators*): 50, 100, dan 200
- Kedalaman maksimum pohon (*max_depth*): 10, 20, dan 30

Dari hasil pengujian, kombinasi parameter dengan nilai akurasi tertinggi dipilih sebagai konfigurasi model akhir yang digunakan pada tahap pelatihan dan evaluasi berikutnya.

3.3.5. Seleksi Fitur

Tahap seleksi fitur dilakukan untuk mengidentifikasi atribut yang memiliki pengaruh paling besar terhadap hasil prediksi model. Proses ini menggunakan metode *SHAP (Shapley Additive Explanations)* yang menghitung kontribusi setiap fitur terhadap keluaran model dengan pendekatan teori permainan (*game theory*). Nilai *SHAP* menunjukkan seberapa besar perubahan rata-rata pada hasil prediksi ketika suatu fitur dimasukkan ke dalam model. Fitur dengan nilai rata-rata absolut *SHAP* tertinggi dianggap memiliki pengaruh paling besar terhadap keputusan model. Fitur-fitur tersebut dipertahankan, sedangkan fitur dengan kontribusi rendah dihapus untuk mengurangi kompleksitas model dan meningkatkan efisiensi proses pelatihan tanpa menurunkan akurasi secara signifikan.

3.4. Implementasi

Pada tahap ini dilakukan implementasi sistem deteksi *phishing* berdasarkan rancangan yang telah dibuat. Proses implementasi mencakup pelatihan algoritma *Random Forest* menggunakan parameter optimal hasil *GridSearchCV* serta seleksi fitur berbasis *SHAP* untuk mengidentifikasi atribut paling berpengaruh terhadap deteksi *phishing*. Implementasi ini bertujuan agar model *Random Forest* dapat mempelajari pola dari data latih secara efisien dan melakukan klasifikasi *URL* pada data uji dengan tingkat akurasi tinggi serta interpretabilitas yang baik.

3.4.1. *Random Forest*

Random Forest merupakan pengembangan dari algoritma *Decision Tree* dengan pendekatan *ensemble* yang digunakan untuk klasifikasi dan regresi. Algoritma ini membangun banyak pohon keputusan secara independen menggunakan teknik *bagging*, yaitu pengambilan sampel acak dengan pengembalian. Hasil dari seluruh pohon digabungkan untuk menghasilkan prediksi akhir yang lebih akurat dan stabil [11].

Semakin banyak jumlah pohon yang digunakan, semakin tinggi pula tingkat akurasi prediksi yang dapat dicapai. Pendekatan ini membantu mengurangi risiko *overfitting* yang sering terjadi pada model *decision tree* tunggal [14]. Namun demikian, penggunaan pohon dalam jumlah besar dapat meningkatkan beban komputasi secara signifikan, terutama pada data berdimensi tinggi atau yang memiliki banyak fitur [12]. *Random Forest* juga termasuk dalam metode *Bootstrap Aggregating* (*Bagging*) yang diperkenalkan oleh Breiman. Prosesnya dimulai dengan membuat sub-dataset acak dari data latih untuk melatih banyak pohon keputusan. Pada setiap node digunakan subset fitur acak guna mengurangi korelasi antar pohon. Prediksi akhir ditentukan dengan *majority voting* pada klasifikasi atau rata-rata pada regresi, sehingga model menjadi lebih akurat dan robust [17]. Rumus umum untuk prediksi yang diagregasi didefinisikan pada Persamaan 2.

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B} \quad \hat{f}_b(x) \tag{2}$$

3.4.2. *GridSearchCV*

Hyperparameter tuning memiliki peran penting dalam mengoptimalkan kinerja algoritma *machine learning*, karena nilai *hyperparameter* harus ditentukan sebelum model menjalani proses pembelajaran. Salah satu teknik yang banyak digunakan adalah *Grid Search*, yaitu metode yang secara sistematis menguji berbagai kombinasi *hyperparameter* untuk menemukan nilai terbaik. *Grid Search* memang memberikan hasil yang menyeluruh, namun dapat menyebabkan lonjakan waktu komputasi, khususnya saat ruang pencarian parameter menjadi luas [18].

Proses ini biasanya dikombinasikan dengan *Cross Validation* agar hasil yang diperoleh lebih andal dan konsisten. Secara umum, *GridSearch* dilakukan dengan cara menentukan ruang pencarian parameter, menguji setiap kombinasi menggunakan *Cross Validation*, memilih model dengan hasil terbaik, lalu melatih ulang model dengan parameter terbaik [19]. Evaluasi setiap kombinasi parameter pada *GridSearchCV* dihitung berdasarkan rata-rata skor *K-Fold* sebagaimana dinyatakan pada Persamaan (3).

$$Score\big(M(\theta)\big) = \frac{1}{K}\sum_{k=1}^{K} Score_k\big(M(\theta)\big) \tag{3}$$

3.4.3. *SHAP*

SHAP merupakan metode interpretabilitas model yang berasal dari konsep nilai *Shapley* dalam teori permainan. Metode ini mampu menjelaskan kontribusi setiap fitur terhadap hasil prediksi secara konsisten, adil, dan mudah dipahami, serta memiliki kelebihan dalam kompatibilitasnya dengan berbagai algoritma *machine learning* dan kemampuannya mengukur pengaruh relatif setiap fitur secara akurat [20]. *Threshold* digunakan untuk menilai relevansi fitur terhadap model. Umumnya digunakan nilai minimum 0,05. Fitur dengan nilai di atas *threshold* dipertahankan, sedangkan yang di bawahnya dihapus karena dianggap kurang berpengaruh terhadap akurasi prediksi [11].

Dalam konteks ini, *SHAP* digunakan untuk mengevaluasi sejauh mana masing-masing fitur memengaruhi output prediksi model. Nilai kontribusi tersebut dapat dihitung melalui formula matematis yang merepresentasikan distribusi pengaruh setiap fitur berdasarkan nilai marjinalnya terhadap semua kombinasi fitur yang mungkin [21]. Nilai kontribusi setiap fitur diperoleh menggunakan persamaan 4:

$$\phi i \; = \sum_{S\subseteq\{1,2,3,\ldots,p\}\{i\}} \frac{|S|!\;(p\;-\;|S|\;-\;1)!}{p!}[f\;(S\;\cup\;\{i\})\;-\;f\;(S)] \tag{4}$$

3.5. Pengujian

Hasil pengujian dievaluasi menggunakan *confusion matrix* untuk menghitung nilai *precision*, *recall*, dan *accuracy* sebagaimana telah ditunjukkan pada persamaan (5), (6), dan (7), serta nilai *F1-score* sebagai metrik tambahan pada persamaan (8). Pada metode *Random Forest*, hasil akhir prediksi diperoleh melalui mekanisme pemungutan suara mayoritas dari seluruh pohon keputusan yang terbentuk. Setiap pohon memberikan satu hasil klasifikasi, dan kelas yang paling sering muncul ditetapkan sebagai hasil akhir prediksi.

$$Precision = \frac{TP}{TP + FP} \times 100\% \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (7)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (8)$$

4. Hasil dan Pembahasan

4.1. Pra-proses Data

Pra-pemrosesan data dilakukan untuk memastikan kualitas data sebelum digunakan pada tahap pelatihan model. Tahapan ini meliputi pemeriksaan nilai kosong (*missing values*), penghapusan kolom yang tidak relevan seperti *url* dan *status* (fitur target), serta proses scaling terhadap seluruh fitur numerik agar memiliki skala yang seragam. Proses ini bertujuan untuk mengurangi bias akibat perbedaan rentang nilai antar fitur serta mempermudah algoritma dalam melakukan pembelajaran pola data.

Langkah pertama pada tahap pra-pemrosesan adalah pemeriksaan nilai kosong (*missing values*) untuk memastikan bahwa seluruh atribut data dapat digunakan tanpa kehilangan informasi. Hasil pemeriksaan ditunjukkan pada Gambar 3, yang memperlihatkan bahwa seluruh kolom pada dataset tidak memiliki nilai kosong (*null*).

```
Jumlah missing values: 0
Jumlah data duplikat: 0
```

Gambar 3. Pemeriksaan *Null Value* & Duplikasi

Selanjutnya, dilakukan proses penghapusan kolom yang tidak relevan dan proses *scaling* terhadap seluruh fitur numerik. Hasil perbandingan nilai fitur sebelum dan sesudah *scaling* ditunjukkan pada Tabel 1, di mana tampak adanya perubahan skala nilai pada setiap fitur agar memiliki distribusi yang lebih seragam. Setelah proses *scaling*, dataset kemudian dibagi menjadi dua bagian, yaitu 70% data latih (*training*) dan 30% data uji (*testing*) menggunakan metode *stratified split* untuk menjaga proporsi kelas yang seimbang antara kedua subset data. Proses ini bertujuan agar model dapat belajar dengan baik dari data latih dan mampu menguji performanya secara objektif menggunakan data uji yang belum pernah dilihat sebelumnya.

Tabel 1. Proses *Scaling* Beberapa Fitur

Sampel Fitur	Sebelum *Scaling*	Sesudah *Scaling*
ratio_intMedia	100.0000	1.2353
links_in_tags	80.0000	0.6749
domain_registration_length	45.0000	-0.5493
length_url	37.0000	-0.4363

Sampel Fitur	Sebelum *Scaling*	Sesudah *Scaling*
length_hostname	19.0000	-0.1940
nb_hyperlinks	17.0000	-0.4209
longest_words_raw	11.0000	-0.1990
longest_word_host	11.0000	0.1079
avg_word_host	7.0000	-0.1895
longest_word_path	6.0000	-0.1977

4.2. Optimalisasi *GridSeachCV*

Proses optimasi hyperparameter dilakukan untuk memperoleh kombinasi parameter terbaik pada algoritma Random Forest. Optimasi ini menggunakan metode GridSearchCV dengan skema 5-Fold Cross Validation agar hasil evaluasi model tidak bergantung pada satu subset data saja. Parameter yang diuji meliputi jumlah pohon (*n_estimators*) dengan nilai {50, 100, 200} dan kedalaman maksimum pohon (*max_depth*) dengan nilai {10, 20, 30}.

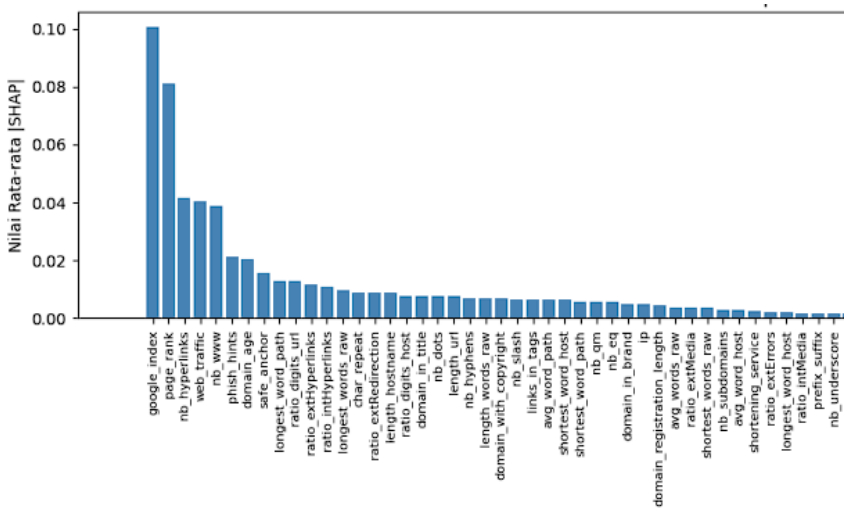
Hasil pengujian setiap kombinasi parameter ditunjukkan pada Tabel 2, yang menampilkan nilai akurasi pada lima lipatan validasi untuk masing-masing konfigurasi model. Berdasarkan hasil pengujian tersebut, kombinasi parameter terbaik diperoleh pada *n_estimators* = 200 dan *max_depth* = 30, dengan rata-rata akurasi tertinggi sebesar 96,51%. Kombinasi ini menghasilkan performa yang paling konsisten di seluruh lipatan validasi dibandingkan konfigurasi lainnya.

Tabel 2. Parameter Terbaik *GridSearchCV*

n_estimators	max_depth	split0_test_score	split1_test_score	split2_test_score	split3_test_score	split4_test_score
200	30	96.8100	96.6900	96.3800	96.1900	96.5000
200	20	96.7500	96.6200	96.3100	96.1900	96.3100
100	30	96.7500	96.4400	96.0600	96.1200	96.2500
100	20	96.5600	96.0600	96.1200	96.0600	96.0600
50	30	96.5000	96.0600	95.8100	95.8800	96.2500
50	20	96.2500	96.0600	95.6900	95.6900	96.0000
200	10	96.4400	95.8800	95.8100	95.8100	95.8100
100	10	96.2500	95.9000	95.7500	95.4400	95.8800
50	10	96.0000	95.7500	95.5600	95.3100	95.5600

4.3. *Feature Importance*

Analisis feature importance dilakukan menggunakan metode *SHAP* untuk mengetahui kontribusi setiap fitur terhadap hasil prediksi model Random Forest. Hasil perhitungan nilai *SHAP* divisualisasikan pada Gambar 4, yang menunjukkan tingkat kepentingan dari 89 fitur yang digunakan dalam proses pelatihan. Grafik tersebut memperlihatkan bahwa sebagian besar fitur memiliki pengaruh yang relatif kecil, sementara beberapa fitur seperti *google_index*, *page_rank*, dan *nb_hyperlinks* menonjol dengan nilai *SHAP* yang lebih tinggi. Hal ini menunjukkan bahwa meskipun banyak fitur digunakan, hanya sebagian kecil yang berkontribusi signifikan terhadap klasifikasi antara tautan phishing dan legitimate.

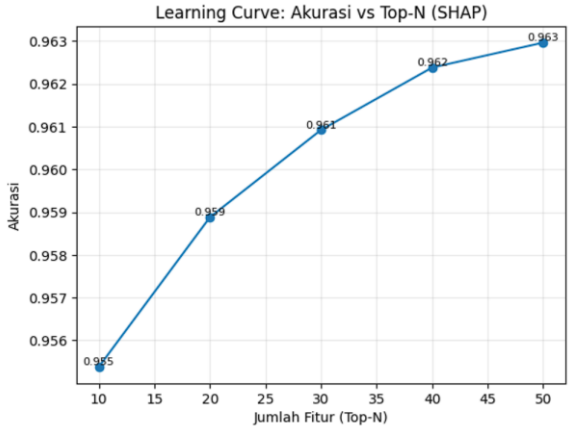


Gambar 4. *Feature Importance SHAP*

4.4. Seleksi Fitur

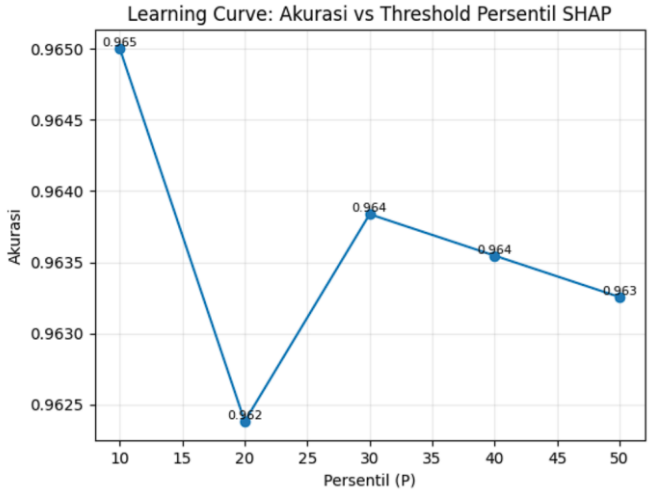
Tahap seleksi fitur bertujuan untuk mengidentifikasi atribut yang paling berpengaruh terhadap hasil klasifikasi serta mengurangi kompleksitas model tanpa menurunkan performa secara signifikan. Proses ini menggunakan nilai *SHAP (Shapley Additive Explanations)* yang dihasilkan dari model *Random Forest* terbaik hasil optimasi *GridSearchCV* sebagai dasar penentuan tingkat kepentingan fitur.

Dalam penelitian ini diterapkan dua pendekatan, yaitu berdasarkan jumlah fitur teratas (*Top-N*) dan berdasarkan ambang batas nilai *SHAP* (*threshold persentil*). Pada pendekatan *Top-N*, model dievaluasi ulang menggunakan fitur dengan nilai SHAP tertinggi sebanyak 10, 20, 30, 40, dan 50 fitur. Hasil pengujian yang ditampilkan pada Gambar 5 menunjukkan tren peningkatan akurasi seiring bertambahnya jumlah fitur yang digunakan. Nilai akurasi meningkat dari 0,955 pada Top-10 fitur menjadi 0,963 pada Top-50 fitur, dengan peningkatan yang mulai stabil setelah Top-40 fitur. Hal ini menunjukkan bahwa sebagian besar informasi penting telah tercakup pada sekitar 30–40 fitur utama, sementara penambahan fitur berikutnya hanya memberikan peningkatan kecil pada performa model.



Gambar 5. *Learning* Curve : Akurasi vs *Top-N* (*SHAP*)

Selanjutnya, pada pendekatan berbasis *threshold persentil SHAP*, fitur dipilih berdasarkan batas nilai kontribusi tertentu (10%, 20%, 30%, 40%, dan 50%) untuk menilai seberapa besar pengaruh fitur dengan nilai *SHAP* rendah terhadap hasil klasifikasi. Visualisasi pada Gambar 6 menunjukkan bahwa akurasi tertinggi sebesar 0,965 dicapai pada ambang 10%, kemudian stabil di kisaran 0,963–0,964 hingga persentil 50. Temuan ini mengindikasikan bahwa sebagian fitur dengan nilai *SHAP* rendah dapat dihilangkan tanpa menyebabkan penurunan performa yang signifikan, sehingga model menjadi lebih efisien dan tetap akurat.



Gambar 6. *Learning Curve* : Akurasi vs Persentil *SHAP*

4.5. Pelatihan Data

Tahap pelatihan dilakukan menggunakan 70% data latih melalui tiga langkah utama, yaitu pelatihan awal *Random Forest*, optimasi *hyperparameter* menggunakan *GridSearchCV*, serta perhitungan nilai *SHAP* berdasarkan model terbaik hasil optimasi. *GridSearchCV* dengan 5-*Fold Cross Validation* menghasilkan kombinasi parameter yang paling optimal, yang kemudian digunakan sebagai model utama. Nilai *SHAP* dari model tersebut selanjutnya dimanfaatkan untuk melatih ulang model dalam skenario seleksi fitur *Top-N* dan *Threshold SHAP*.

Untuk menunjukkan hasil deteksi phishing pada proses pelatihan, Tabel 3 menampilkan sampel lima baris prediksi model terbaik pada data latih, meliputi label asli, prediksi, dan probabilitas *phishing*. Hasil ini menunjukkan bahwa model mampu mempelajari pola *URL phishing* dengan baik, ditandai oleh tingkat kecocokan prediksi yang tinggi pada sebagian besar sampel data.

Tabel 3. Sampel Data Hasil Deteksi Phishing (*Training*)

Url Asli	Label Asli	Prediksi Model	Probabilitas Phishing
http://sanlikala.com/paypal/	1	1	0.8150
http://vxdse.myfreesites.net/	1	1	0.9900
https://www.skirmshop.nl/en/	0	0	0.0050
http://stardewvalleywiki.com/Fiber	0	0	0.0500
http://community.linksys.com/t5..	0	0	0.0050

4.6. Evaluasi Model

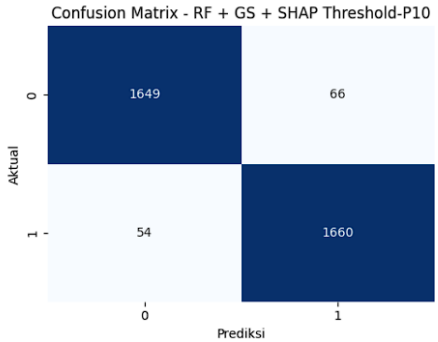
Berdasarkan hasil pengujian pada berbagai skenario seperti yang ditunjukkan pada Tabel 4, model *RF + GridSearchCV + SHAP Threshold-P10* memperoleh performa terbaik dengan nilai akurasi sebesar 0,9650 dan *F1-score* sebesar 0,9651. Nilai tersebut sedikit lebih tinggi dibandingkan model *Baseline RF (default, semua fitur)* dengan akurasi 0,9647 dan *F1-score* 0,9648, serta model *RF + GridSearchCV (semua fitur)* dengan akurasi 0,9644. Hasil ini menunjukkan bahwa penerapan optimasi *hyperparameter* dan seleksi fitur berbasis *SHAP* tidak hanya mempertahankan, tetapi juga meningkatkan performa model meskipun jumlah fitur yang digunakan lebih sedikit.

Tabel 4. Ringkasan Hasil Evaluasi Model pada Berbagai Skenario

No.	Algoritma	*n_features*	*Accuracy*	*Precision*	*Recall*	*F1*
1.	*RF + GS + SHAP Threshold-P10*	78	0.9650	0.9618	0.9685	0.9651
2.	*Baseline RF (default)*	87	0.9647	0.9623	0.9673	0.9648
3.	*RF + GridSearchCV* (semua fitur)	87	0.9644	0.9612	0.9679	0.9645
4.	*RF + GS + SHAP Threshold-P30*	61	0.9638	0.9611	0.9667	0.9639
5.	*RF + GS + SHAP Threshold-P40*	52	0.9635	0.9579	0.9697	0.9638
6.	*RF + GS + SHAP Threshold-P50*	44	0.9633	0.9584	0.9685	0.9634
7.	*RF + GS + SHAP Top-50*	50	0.9630	0.9579	0.9685	0.9632
8.	*RF + GS + SHAP Top-40*	40	0.9624	0.9568	0.9685	0.9626
9.	*RF + GS + SHAP Threshold-P20*	69	0.9624	0.9573	0.9685	0.9626
10.	*RF + GS + SHAP Top-30*	30	0.9609	0.9545	0.9679	0.9612

Pada hasil evaluasi tersebut terlihat bahwa nilai *recall* cenderung sedikit lebih tinggi dibandingkan presisi. Pola ini menunjukkan bahwa model lebih sensitif dalam mengenali tautan *phishing* dibandingkan dalam menghindari kesalahan klasifikasi pada tautan *legitimate*. Kondisi ini dianggap menguntungkan dalam konteks deteksi *phishing*, karena lebih baik sistem mendeteksi secara berlebihan (terjadi *false positive*) daripada gagal mengenali tautan berbahaya (*false negative*).

Visualisasi *confusion matrix* pada Gambar 7 memperlihatkan performa model terbaik, yaitu *RF + GS + SHAP Threshold-P10*, terhadap data uji. Model berhasil mengklasifikasikan dengan benar sebanyak 1.649 data *legitimate* dan 1.660 data *phishing*, sementara kesalahan prediksi hanya terjadi pada 66 data *legitimate* yang terdeteksi sebagai *phishing* serta 54 data *phishing* yang tidak terdeteksi. Pola distribusi hasil ini menunjukkan bahwa model memiliki tingkat ketepatan dan kepekaan yang seimbang antara kedua kelas, sehingga mampu melakukan klasifikasi secara konsisten dengan tingkat kesalahan yang rendah.



Gambar 7. *Confusion Matrix* Model Terbaik

Selain evaluasi metrik, proses pengujian juga diperlihatkan melalui lima sampel hasil deteksi *phishing* pada data uji sebagaimana ditampilkan pada Tabel 5. Sampel tersebut menunjukkan bahwa *URL phishing* memperoleh probabilitas prediksi yang tinggi, sedangkan *URL legitimate* cenderung memiliki probabilitas rendah. Pola ini mengonfirmasi bahwa model mampu melakukan generalisasi dengan baik pada data yang tidak digunakan dalam proses pelatihan.

Tabel 5. Sampel Data Hasil Deteksi Phishing (*Testing*)

Url Asli	Label Asli	Prediksi Model	Probabilitas Phishing
http://www.pollsbee.com/wp-content/Doc/9600588...	1	1	1.0000
http://gbw3d.pl/wp-admin/css/trustpass.html	1	1	0.9350
http://redeabreu.com.br/wp-includes/painel-de-...	1	1	0.6500
http://www.petitcitron.com/	0	0	0.0000
https://en.wikipedia.org/wiki..	0	0	0.0000

4.7. Pembahasan

Hasil pengujian menunjukkan bahwa integrasi algoritma *Random Forest* dengan optimasi *hyperparameter* menggunakan *GridSearchCV* serta seleksi fitur berbasis *SHAP (Shapley Additive Explanations)* memberikan penguatan nyata terhadap temuan penelitian terdahulu dalam deteksi *phishing*. Temuan ini memperkuat hasil Saputra et al. [11] dan Windarni et al. [14] yang menyatakan bahwa *Random Forest* dan optimasi *hyperparameter* mampu mencapai akurasi di atas 96%, namun penelitian ini melangkah lebih lanjut dengan menunjukkan bahwa seleksi fitur berbasis *SHAP* mampu mempertahankan bahkan meningkatkan performa model dengan jumlah fitur yang lebih sedikit serta interpretabilitas yang lebih baik. Hasil ini juga sejalan dengan Lukito dan Handaya [10] yang menegaskan pentingnya optimasi *hyperparameter*, serta melengkapi penelitian Kencana et al. [15] yang belum menerapkan optimasi parameter dan interpretasi fitur. Selain itu, kecenderungan nilai *recall* yang lebih tinggi mendukung temuan Kalla dan Kuraku [16] bahwa sensitivitas model lebih krusial dalam konteks keamanan siber. Dengan demikian, penelitian ini mengintegrasikan dan mempertegas temuan-temuan sebelumnya melalui pendekatan terpadu yang meningkatkan akurasi, efisiensi, dan transparansi model deteksi phishing.

5. Simpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa penerapan algoritma *random forest* yang dioptimasi menggunakan *GridSearchCV* dan seleksi fitur berbasis *SHAP feature importance* menghasilkan performa terbaik pada konfigurasi *RF + GS + SHAP Threshold*-P10 dengan nilai *accuracy* 0,9650 dan *F1-score* 0,9651. Kombinasi optimasi parameter dan seleksi fitur terbukti meningkatkan performa serta efisiensi model, di mana fitur seperti *google_index*, *page_rank*, dan *nb_hyperlinks* memiliki kontribusi paling signifikan terhadap hasil klasifikasi. Proses *hyperparameter tuning* menghasilkan konfigurasi optimal pada *n_estimators* = 200 dan *max_depth* = 30, serta evaluasi menunjukkan keseimbangan yang baik antara *precision* dan *recall*. Ke depan, penelitian ini dapat dikembangkan melalui implementasi sistem deteksi phishing berbasis web atau mobile serta pengujian algoritma lain dan metode optimasi yang lebih efisien untuk meningkatkan skalabilitas dan kinerja model.

Daftar Referensi

[1] F. Haikal and R. J. Anward, “Dampak Penggunaan Teknologi Informasi dan Komunikasi terhadap Produk Domestik Regional Bruto (PDRB) Per Kapita Tingkat Provinsi di Indonesia,” *JIEP: Jurnal Ilmu Ekonomi dan Pembangunan*, vol. 6, no. 1, pp. 45–60, 2025, Accessed: Sep. 15, 2025.

[2] A. N. Nursabilah, N. V. Khurulani, A. Prasanti, D. A. Zuhra, A. A. Hg, and N. Nabanurohmah, “Perlindungan Hukum Bagi KorbanTindak Pidana Cyber Scam Serta Dampaknya Bagi Korban Sebagai Bentuk Viktimisasi Sekunder,” *Hukum Inovatif*, vol. 2, pp. 168–187, Jul. 2025.

[3] P. Wijiastuti, H. Azahro, and A. Edward, “Analisis Kesadaran Ancaman Phishing di Social Media terhadap Gen Z di Indonesia Rentang Umur 12–27 Tahun Menggunakan Metode Likert,” *JITU: Jurnal Informatika Utama*, vol. 3, no. 1, pp. 82–93, 2025.
[4] Kaspersky, “Kaspersky reports nearly 900 million phishing attempts in 2024 as cyber threats increase,” Kaspersky.com.
[5] T. F. Ramadhan, I. Ramadhan, and A. A. Pangestu, “Analisis Keamanan Teknologi Dalam Menghadapi Ancaman Phising,” in *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, Surakarta, Jul. 2024, pp. 568–573.
[6] K. 4a4 and O. Iskandar, “Analisis Kejahatan Online Phishing Pada Masyarakat,” *Leuser: Jurnal Hukum Nusantara*, vol. 1, no. 2, pp. 32–36, Jun. 2024.
[7] A. Nofiyan and M. Mushlihudin, “Analisis Forensik pada Web Phishing Menggunakan Metode National Institute Of Standards And Technology (NIST),” *JSTIE: Jurnal Sarjana Teknik Informatika*, vol. 8, no. 2, pp. 11–23, May 2020.
[8] Komdigi, “Tangguhnya Keamanan Siber LPS Dalam Menangkal Serangan Hacker,” komdigi.go.id.
[9] P. Vaitkevicius and V. Marcinkevicius, “Comparison of Classification Algorithms for Detection of Phishing Websites,” *Informatica (Netherlands)*, vol. 31, no. 1, pp. 143–160, 2020.
[10] Lukito and W. B. T. Handaya, “Deteksi Website Phishing Menggunakan Teknik Machine Learning,” *Jurnal Informatika Atma Jogja*, vol. 6, pp. 69–80, May 2025.
[11] R. Saputra and E. Hartati, “Deteksi Website Phishing Menggunakan Algoritma Random Forest dengan Optimalisasi GridSearch,” *JUTIM: Jurnal Teknologi Musi Rawas*, vol. 10, no. 1, pp. 55–67, Jun. 2025.
[12] K. D. Tzimourta, M. G. Tsipouras, P. Angelidis, D. G. Tsalikakis, and E. Orovou, “Maternal Health Risk Detection: Advancing Midwifery with Artificial Intelligence,” *Healthcare (Switzerland)*, vol. 13, no. 7, pp. 1–21, Apr. 2025.
[13] A. L. Puspanagara, “Penerapan Explainable AI untuk Prediksi Performa Akademik Mahasiswa Menggunakan Random Forest dan SHAP,” *Infoman’s: Jurnal Ilmu-ilmu Informatika dan Manajemen*, vol. 19, no. 1, pp. 1–7, May 2025.
[14] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, and A. Setiawan, “Deteksi Website Phishing Menggunakan Teknik Filter pada Model Machine Learning,” *Information System Journal (INFOS)*, vol. 6, no. 1, pp. 39–43, May 2023.
[15] A. K. Kencana, F. D. Ananda, and A. D. Hartanto, “Implementasi Metode Random Forest Klasifikasi untuk Phishing Link Detection,” *Information Technology Journal*, vol. 4, no. 2, pp. 55-59, Dec. 2022.
[16] D. Kalla and S. Kuraku, “Phishing Website URL’s Detection Using NLP and Machine Learning Techniques,” *Journal on Artificial Intelligence*, vol. 5, no. 0, pp. 145–162, 2023.
[17] K. Cao-Van, T. C. Minh, L. G. Minh, T. T. B. Quyen, and H. M. Tan, “Soft-Voting Ensemble Model: An Efficient Learning Approach for Predictive Prostate Cancer Risk,” *Vietnam Journal of Computer Science*, vol. 11, no. 4, pp. 531–552, Nov. 2024.
[18] A. R. Kamila and V. Budiyanto, “Optimasi Model dengan Algoritma Support Vector Regressor Menggunakan Grid Search pada Penilaian Essai Otomatis,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4622–4627, Jun. 2025.
[19] W. Nugraha and A. Sasongko, “Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search,” *SISTEMASI: Jurnal Sistem Informasi*, vol. 11, no. 2, pp. 2540–9719, May 2022.
[20] S. Alfadia Shauqie, M. Nurkamal Fauzan, and C. Prianto, “Analisis Pengaruh Fitur Terhadap Tinggi Badan Anak menggunakan SHAP,” *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 11, pp. 271–276, Aug. 2025.
[21] Y. Gong, Q. Du, F. Wang, and L. Zhang, “Predicting road adhesion coefficient with a fusion strategy of SHAP dynamic parameters,” *Sci Rep*, vol. 15, no. 1, p. 35603, Oct. 2025.x`