PHISHING WEBSITE DETECTION USING A MACHINE LEARNING CLASSIFICATION APPROACH

DETEKSI WEB PHISHING MENGGUNAKAN PENDEKATAN KLASIFIKASI MACHINE LEARNING

Ibnu Arifin[1], Chairani[2]
[1,2]Magister Teknik Informatika, Institut Informatika dan Bisnis Darmajaya, Bandar Lampung, Indonesia
Email: [1]ibnu.2321211018@mail.darmajaya.ac.id, [2]chairani@mail.darmajaya.ac.id

Abstract *- Phishing is a form of cybercrime that is increasingly prevalent, with millions of attacks recorded annually. This study develops a phishing website detection model using a machine learning classification approach, employing a pipeline that includes data preprocessing, feature selection, and model validation. The dataset was obtained from the UCI Machine Learning Repository and consists of 235,795 URLs with a relatively balanced distribution between phishing (100,945) and non-phishing (134,850). After data cleaning and feature selection, 21 optimal features were retained, ensuring they were safe from potential data leakage. Two algorithms were evaluated: decision tree and random forest, using 10-fold cross-validation. The random forest algorithm achieved an average accuracy of 97.78%, while the decision tree was slightly higher at 98.02%. However, random forest outperformed in class discrimination, as measured by ROC-AUC (99.73%) and PR-AUC (99.78%), compared to decision tree values of 99.49% and 99.40%. The method also incorporated a 10-fold cross-validation procedure to minimize data leakage and ensure reliable model evaluation. The Wilcoxon test further confirmed that the performance difference between the two algorithms is statistically significant. Overall, although the decision tree demonstrates strong classification performance, random forest proves to be more consistent and reliable in detecting phishing websites, making it a superior choice in the context of cybersecurity.*

Keywords *- Phishing; Machine Learning; Random Forest; Phishing Detection; Website Classification*

Abstrak - *Phishing* merupakan bentuk kejahatan siber yang semakin meningkat, dengan jutaan serangan tercatat setiap tahunnya. Penelitian ini mengembangkan model deteksi *phishing website* menggunakan pendekatan klasifikasi *machine learning* dengan *pipeline* yang mencakup pra-pemrosesan data, seleksi fitur, dan validasi model. Dataset diperoleh dari *UCI Machine Learning Repository*, terdiri dari 235.795 URL dengan distribusi relatif seimbang antara *phishing* (100.945) dan *non-phishing* (134.850). Setelah proses pembersihan data dan seleksi fitur dipilih 21 fitur akhir yang terbaik yang aman dari potensi kebocoran data. Dua algoritma diuji, yaitu *decision tree*, dan *random forest*, menggunakan *cross-validation 10-fold*. Algoritma random forest menunjukkan akurasi rata-rata 97,78%, sedangkan *decision tree* sedikit lebih tinggi pada 98,02%. Namun, *random forest* unggul dalam hal kemampuan diskriminasi kelas yang dinilai melalui nilai *ROC-AUC* (99,73%) dan *PR-AUC* (99,78%), dibandingkan dengan *decision tree* yang mencatat nilai 99,49% dan 99,40%. Metode ini juga menyertakan prosedur *cross-validation 10-fold* untuk meminimalkan kebocoran data dan memastikan keandalan evaluasi model. Hasil uji signifikansi *wilcoxon test* menegaskan bahwa perbedaan kinerja antara kedua algoritma itu signifikan. Secara keseluruhan, meskipun *decision tree* memiliki performa klasifikasi yang baik, *random forest* terbukti lebih konsisten dan andal dalam mendeteksi *web phishing*, menjadikannya pilihan yang lebih baik dalam konteks keamanan siber.

Kata Kunci - *Phishing; Machine Learning; Random Forest;* Deteksi *Phishing;* Klasifikasi *Website*

I. PENDAHULUAN

Phishing merupakan salah satu bentuk kejahatan siber yang memanfaatkan kelalaian korban dalam mengakses sebuah tautan di sebuah situs web, sehingga korban memasukkan data-data sensitif di tautan palsu [1]. *Phishing* adalah serangan siber yang paling umum yang digunakan oleh penjahat siber untuk mendapatkan akses ke informasi pribadi pengguna internet, seperti informasi kartu kredit, nama pengguna, dan kata sandi [2]. Dalam serangan *phishing*, pengguna online ditipu oleh entitas yang tampaknya tepercaya untuk memberikan data pribadi mereka, misalnya kredensial *login* atau detail kartu kredit. Ketika informasi pribadi ini bocor ke peretas, informasi ini menjadi sumber serangan canggih lainnya [3] *Anti-Phishing Working Group (APWG)* [4] mengamati hampir lima juta serangan *phishing* selama tahun 2023, yang merupakan tahun rekor. Pada kuartal pertama tahun 2024, *APWG* mengamati 963.994 serangan *phishing*. Ini adalah total kuartalan terendah sejak 4Q 2021, dan jauh di bawah 1.624.144 serangan yang terlihat pada Q1 2023, yang merupakan rekor kuartalan tertinggi dalam pengamatan historis *APWG*. Secara keseluruhan, jumlah serangan per bulan stabil dari Juni 2023 hingga Maret 2024.

Dalam literatur, bermacam-macam teknik diusulkan untuk mengidentifikasi situs web *phishing*, Berbasis Daftar, Kesamaan Visual, Heuristik, Pembelajaran Mesin [5] dan *Deep Learning* [6]. Pembelajaran Mesin adalah hal yang lazim pendekatan untuk mendeteksi situs *web phishing* [7].Atribut umum seperti informasi *URL*, struktur situs web, dan fitur *JavaScript* dikumpulkan untuk mewakili *URL phishing* dan situs web terkait. Kemudian, berdasarkan fitur tersebut, data *phishing set* diperoleh. Setelah itu, pengklasifikasi *Machine Learning* adalah dilatih untuk mendeteksi situs web *phishing* berdasarkan fitur-fitur tersebut [8]. Teknik ini bekerja sangat baik dengan *Big Data set* (memiliki *Velocity*, *Variety*, *Volume*, *Value*, dan *Veracity* yang tinggi). Pengklasifikasi berbasis *Machine Learning* mencapai akurasi lebih dari 99%, yang terbukti menjadi metode yang paling efektif [9].

Berbagai penelitian menggunakan algoritma *machine learning* telah digunakan untuk mendeteksi *website phishing*. Penelitian [10] menujukan bahwa algoritma *Random forest* memiliki akurasi tertinggi sebesar 99,33% dari empat algoritma yang diuji. Hal yang sama juga ditemukan oleh [11] dan [12] memperoleh hasil masing-masing 97,2%, 99,78%, untuk algoritma *random forest*.

Meskipun *random forest* menunjukan performa yang tinggi, sebagian besar studi masih menyisakan keterbatasan penting: potensi data *leakage* akibat pra-pemrosesan dan seleksi fitur yang tidak terintegrasi dalam pipeline, penggunaan seluruh fitur tanpa strategi seleksi yang sistematis, serta kurangnya perhatian pada *trade-off* antara *false positive* dan *false negative*. Kondisi ini membuat hasil evaluasi berisiko terlalu optimistis dan kurang siap diimplementasikan pada skenario nyata berskala besar.

Penelitian ini bertujuan untuk mengembangkan *pipeline* anti-*leakage* dalam deteksi *phishing* skala besar dengan memastikan setiap tahap pra-pemrosesan, seleksi fitur, validasi, dan evaluasi dilakukan di dalam kerangka *cross-validation* yang ketat sehingga terhindar dari kebocoran data. Untuk meningkatkan kualitas model, penelitian ini juga menerapkan seleksi fitur berbasis *correlasi Analysi* dan *ANOVA (Analysis of Variance)* guna mengidentifikasi fitur-fitur paling diskriminatif sekaligus mengurangi kompleksitas komputasi agar sistem lebih efisien, terutama pada skenario *real-time*. Selanjutnya, penelitian ini menganalisis dan membandingkan kinerja algoritma pembelajaran mesin, yaitu *decision tree* dan *random forest*. Kinerja model dievaluasi menggunakan metrik komprehensif, termasuk akurasi, presisi, recall, F1-score, *ROC-AUC*, dan *PR-AUC*, guna menangkap *trade-off* antara *false positive* dan *false negative* yang krusial dalam konteks keamanan siber.

Beberapa *Research* tentang Deteksi *web Phishing* sebagai berikut :

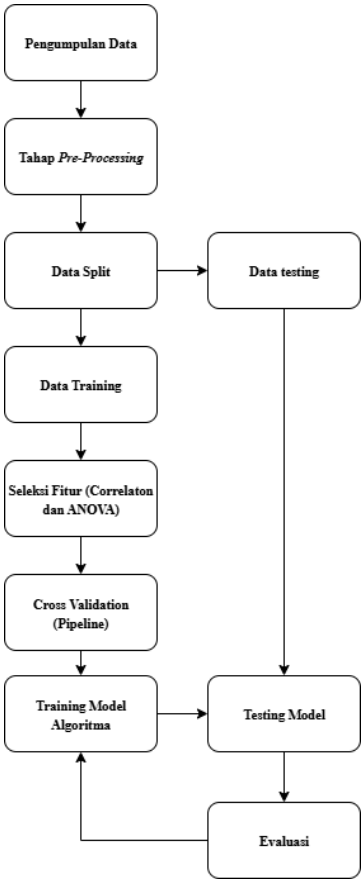
Tabel 1. *Research* terdahulu tentang deteksi *web phishing*

No	Penelitian	Tahun	Objek Penelitian	Kategori feature	optimasi pemilihan feature	Prediksi	Modeling	Validasi	Database	jumlah data	hasil akurasi
1	J. Stobbs dkk [10]	2020	*Phishing Web*	*URL, DOM structrure, page Range, dan Page Informasi*	*Genetic Algoritm (GA), Moth flame optimisation(MFO) dan (GA), Particle swarm optimisation (PSO)*	*phishing Or Legitimate URL*	*LR, SVM, RF dan NN*	*cross validation*	*Phistank dan Alexa*	20.000 dan 10.000	*RF(with PSO)* =99,33% *SVM (with MFO)*=90,9% *Linear Regresion (with MFO)* =91,2% *NN(with MFO)* =91,87%
2	Wei dkk [13]	2020	*Phishing Web*	*URL*	.-	*phishing Or Legitimate URL*	*CNN*	*cross validation*	*Phistank dan Commoncrawl*	21.208	99,98%
3	Wulan Lestari dkk[14]	2024	*Phishing Web*	*URL, DOM structrure, page Range, dan Page Informasi*	*ANOVA*	*phishing Or Legitimate URL*	*DNN*	*Random, (80-20)% dan (30-70)%*	*Kaggle*	11.430	95,29%
4	Onyiagha dkk [15]	2024	*Phishing Web*	.-	*EDA*	*phishing Or No Phishing URL*	*XGBoost, Multilayer Perceptrons (MP), DT, RF dan SVM*	*80-20 %*	*Phistank dan Universitas New Brunswick*	5.000 dan 5.000	*XGBoost*=86,4% *MP*=86,4%, *DT*=81,2%, RF=81,1% dan *SVM*=79,4%
5	Ahammad dkk [16]	2022	*Phishing Web*	*Address Bar dan domain*	.-	*phishing Or Legitimate URL*	*Light GBM, RF, DT, LR,dan SVM*	*cross validation*	*Phistank dan Universitas New Brunswick*	3.000 dan 3.5300	*Light GBM*=86%, *RF*=85,3%, *DT*=85%, *LR*=84,2%,dan *SVM*=83,5%

No	Penelitian	Tahun	Objek Penelitian	Kategori feature	optimasi pemilihan feature	Prediksi	Modeling	Validasi	Database	jumlah data	hasil akurasi
6	Yadav dkk [11]	2021	*Phishing Web*	*Address Bar based Features ,Abnormal Based Features, HTML*	.-	*phishing Or Legitimate URL*	*RF,DT*	.-	*Phistank, Kaggle dan Alexa*	-	*RF*=97,2%,*DT*=95,9%
7	Kathiravan dkk [17]	2024	*Phishing Web*	.-	-	*phishing Or Legitimate URL*	*DT,RF,dan Gradien Boosting*	80-20 %	*kaggle*	11.055	*DT*=96,*RF*=96,9%,dan *Gradien Boosting*=98,9%
8	Adeyemi Onih [12]	2024	*Phishing Web*	.-	.-	*phishing Or No Phishing URL*	*RF,KNN,ANN*	90-10 %	*Jupyter Notebook dan Universitas New Brunswick*	.-	*RF*=99,78%,*KNN*=99,67 %,*ANN*=99,11%
9	Das Guptta dkk [18]	2022	*Phishing Web*	*URL Based Feature dan Hyperlink Based Features*	.-	*phishing Or Legitimate URL*	*LR, DT, SVM, RF, XG Boost*	80-20 %	*Universitas New Brunswick dan Alexa*	3.000 dan 3.000	*LR*=96,2%, *DT*=97,67%, *SVM*=96,9%, *RF*=98,75%, XG *Boost*=99,17%
10	Penulis	2025	*Phishing Web*	*URL, DOM structure, page Range, dan Page Informasi*	Correlation dan ANOVA	*phishing Or Legitimate URL*	*DT, RF,*	*cross validation*	*UCI Machine Learning Repository*	134.850 dan 100.945	

II. SIGNIFIKASI STUDI

Untuk membangun model sebuah model *machine learning* yang handal terdapat beberapa tahapan yang perlu dilakukan sehingga hasil dari model tersebut maksimal dan sesuai dengan harapan. Untuk mempermudah proses penelitian, metode yang digunakan dalam penelitian ini adalah metode *waterfall*. Penelitian dengan metode *waterfall* proses pengembangannya menggunakan model *fase one by one*, sehingga meminimalisir kesalahan yang mungkin akan terjadi. Berikut tahapan dalam membangun model *machine learning* dalam penelitian ini :



Gambar 1. Tahapan membangun model *machine learning*

Tahapan dari metode penelitian ini dapat dilihat pada Gambar 1, yang dimana dibagi menjadi beberapa proses sebagai berikut :

A. Pengumpulan Data

Pengumpulan data yaitu tahapan mengumpulkan data, pada penelitian ini menggunakan data sekunder, diperoleh dari *database UCI Repository*. Pada *dataset phishing website* terdiri dari 134.850 *record URL* Bukan *Phishing* dan 100.945 *URL Phishing* dengan atribut 57 (56 atribut dan 1 target atribut). Target atibut mempunyai dua kategori yaitu bukan *phishing* dan *phishing*[15].

Tabel 2. Database Berdasarkan Class

No	*Klasifikasi*	*Jumlah Record Dataset*
1	Bukan *Phishing*	134.850
2	*Phishing*	100.945
	Jumlah	**235.795**

B. Tahap *Pre-processing (Cleaning, Encoding)*

Tahap ini dilakukan untuk memastikan data yang digunakan dalam pelatihan model berada dalam kondisi optimal dan konsisten. Proses ini mencakup beberapa langkah utama, yaitu data *cleaning* untuk menghapus nilai yang hilang, duplikasi, atau anomali yang berpotensi menurunkan kualitas model; *encoding* untuk mengubah fitur kategorikal menjadi representasi numerik yang dapat diproses oleh algoritma pembelajaran mesin. Pada tahap ini merubah fitur TLD menggunkanan *label encoder* untuk menormalkan skala antar fitur numerik sehingga memiliki rentang nilai yang sebanding.

C. *Split Data* (Pembagian data)

Pada tahap ini data dibagi menjadi 20% data testing dan 80% data training. Langkah ini sangat penting untuk mengantisipasi adanya kebocoran data ketika pemodelan algoritma *machine learning*.

D. Seleksi fitur *(correlation* dan *ANOVA)*

Seleksi fitur dilakukan untuk mendapatkan atribut yang paling relevan sekaligus mengurangi dimensi data agar model lebih efisien. Seleksi fitur ini hanya diterapkan di data training saja. Metode *correlation analysis* digunakan untuk mengidentifikasi fitur yang sangat berkorelasi dengan target sekaligus menghapus fitur yang redundan. Ada 2 fitur yang dihilangkan karena memiliki nilai korelasi tinggi (>0,9) terhadap fitur lainnya yaitu : *'NoOfLettersInURL', 'URLTitleMatchScore'*. Sementara itu, *ANOVA (f_classif)* dipakai untuk memilih fitur yang berpotensi menimbulkan kebocoran data (data *leakage*). Ada 26 fitur yang memiliki potensi data *leakage* karena memiliki *f-value* yang terlalu besar. Dalam tahap ini sehingga dipilih 21 fitur yang terbaik yang digunakan untuk pemodelan *machine learning.*

E. *Cross Validation (Pipeline)*

Tahap ini menggunakan *k-fold cross validation* untuk memastikan evaluasi model lebih andal dan tidak bias terhadap data tertentu. Data dibagi ke dalam 10 kelompok, di mana setiap kelompok bergantian menjadi data uji sementara kelompok lainnya digunakan untuk pelatihan. Proses ini dibungkus dalam *pipeline* yang mencakup pra-pemrosesan, seleksi fitur, dan pelatihan model di setiap *fold*. Dengan cara ini, setiap langkah hanya menggunakan data latih sehingga mencegah data *leakage*, sekaligus memberikan estimasi performa yang lebih konsisten dan representatif untuk implementasi nyata.

F. Pemodelan Klasifikasi *(Random Forest (RF), Decision Tree (DT),*

DT adalah algoritma pembelajaran mesin yang membuat keputusan berdasarkan model berbentuk pohon keputusan. Prosesnya dimulai dari sebuah simpul tunggal yang merepresentasikan keseluruhan dataset, kemudian data dipartisi berdasarkan kondisi tertentu. Proses ini berlanjut secara rekursif sehingga terbentuk struktur menyerupai pohon dengan cabang dan daun. Setiap simpul internal pada pohon merepresentasikan sebuah fitur, setiap cabang menunjukkan aturan keputusan, dan setiap simpul daun merepresentasikan sebuah hasil, yaitu kelas yang diprediksi. DT banyak digunakan karena interpretabilitas dan kesederhanaannya, sebab cara kerjanya menyerupai proses pengambilan keputusan manusia serta tidak memerlukan perhitungan yang kompleks. Pendekatan ini juga memudahkan dalam mengidentifikasi faktor-faktor keputusan yang paling penting serta bagaimana pengaruhnya terhadap hasil prediksi [19].

RF adalah algoritma pembelajaran mesin yang termasuk dalam kelas metode ensemble learning. Algoritma ini menggabungkan banyak pohon keputusan untuk membentuk model prediksi yang lebih kuat dan lebih andal. Setiap pohon dalam hutan dibangun menggunakan subset fitur dan data yang dipilih secara acak, sehingga mencegah overfitting dan meningkatkan kemampuan generalisasi model. Prediksi akhir pada RF diperoleh dengan menggabungkan hasil prediksi dari masing-masing pohon, umumnya melalui mekanisme majority voting. Berbeda dengan pohon keputusan tunggal yang rentan mengalami overfitting, agregasi banyak pohon keputusan dalam RF secara signifikan mengurangi risiko tersebut [20].

G. Evaluasi *(*Akurasi, *ROC-AUC, PR-AUC,* Uji signifikansi)

Dalam evaluasi dan pembandingan hasil algoritma-algoritma tersebut, dilakukan perhitungan akurasi, presisi, *F1 score*, dan *recall* untuk masing-masing algoritma. Penghitungan nilai-nilai itu dapat dilakukan dengan memanfaatkan hasil dari *confusion matrix*. *confusion matrix* merupakan ringkasan dari hasil prediksi dalam kasus klasifikasi. Matriks kebingungan digunakan untuk menilai kinerja pengklasifikasi dalam *dataset* [23]. Terdapat 4 (empat) istilah di dalam *confusion matrix* yang menggambarkan hasil perhitungan klasifikasi, yaitu *True Positive (TP), True Negative (TN), False Positive (FP),* dan *False Negative (FN)* [24].

Beberapa metrik yang digunakan untuk mengevaluasi selain akurasi yaitu metrik berbasis probabilistik seperti *ROC-AUC (Receiver Operating Characteristic – Area Under Curve*) dan *PR-AUC (Precision-Recall – Area Under Curve*). *ROC-AUC* digunakan untuk mengevaluasi kemampuan model dalam membedakan kelas *phishing* dan *non-phishing* secara umum, sedangkan *PR-AUC* lebih menekankan keseimbangan antara presisi dan recall, yang penting pada kasus keamanan siber dengan potensi *class imbalance.* Selain evaluasi dengan metrik, dilakukan pula uji signifikansi statistik *Wilcoxon* untuk membandingkan performa antar algoritma. Uji ini memastikan bahwa perbedaan kinerja model bukan hanya akibat variasi data semata, tetapi benar-benar signifikan secara statistik. Dengan kombinasi metrik performa dan uji signifikansi, hasil evaluasi menjadi lebih robust, valid, dan dapat dipertanggungjawabkan secara ilmiah.

III. HASIL DAN PEMBAHSAN

Berdasarkan evaluasi menggunakan *cross-validation 10-fold*, performa model *decision tree* pada dataset diperoleh sebagai berikut:

1. *Accuracy*: Nilai *Accuracy* pada tiap *fold* berkisar antara 97,74% hingga 98,11%, dengan rata-rata *Accuracy* sebesar 98,02%, menunjukkan model memiliki kemampuan klasifikasi yang tinggi dan stabil di seluruh *fold*.
2. *ROC-AUC*: Model mencapai nilai *ROC-AUC* rata-rata 99,49%, dengan variasi antar *fold* berada di kisaran 99,41% hingga 99,61%, menandakan model sangat baik dalam membedakan kelas positif dan negatif.
3. *PR-AUC*: Rata-rata *PR-AUC* sebesar 99,40%, dengan nilai per *fold* berkisar antara 99,28% hingga 99,57%, menunjukkan model mampu mempertahankan presisi tinggi terutama pada kelas positif *(phishing).*

```
=== Hasil Cross-Validation ===
Accuracy CV per fold: [0.97778838 0.9774173  0.97768236 0.97847752 0.97959075 0.97821247
 0.97784022 0.97561364 0.97789323 0.97757515]
ROC-AUC CV per fold: [0.99728806 0.99730717 0.99710632 0.9974019  0.99740571 0.99711013
 0.99753875 0.9972115  0.99706851 0.99716139]
PR-AUC CV per fold: [0.99785563 0.9978521  0.99765329 0.99788363 0.9978873  0.997598
 0.99805261 0.99779289 0.99754873 0.99774223]
Rata-rata Accuracy CV: 0.9778
Rata-rata ROC-AUC CV: 0.9973
Rata-rata PR-AUC CV: 0.9978

=== Hasil Evaluasi Test Set ===
Accuracy: 0.9779
ROC-AUC : 0.9973
PR-AUC  : 0.9978
```

Gambar 2. Performa model *Random forest*

Model *random forest* juga dievaluasi menggunakan *cross-validation 10-fold*, dengan hasil sebagai berikut:

1. *Accuracy*: Nilai *Accuracy* per *fold* berada di kisaran 97,56% hingga 97,96%, dengan rata-rata 97,78%, sedikit lebih rendah dibanding *decision tree*, tetapi tetap menunjukkan performa klasifikasi yang baik.
2. *ROC-AUC*: Model mencapai rata-rata *ROC-AUC* 99,73%, dengan nilai per *fold* berkisar antara 99,71% hingga 99,75%, menunjukkan kemampuan diskriminasi kelas yang sangat baik.
3. *PR-AUC*: Rata-rata *PR-AUC* sebesar 99,78%, dengan nilai per *fold* antara 99,75% hingga 99,81%, menandakan presisi tinggi dan konsistensi performa pada kelas *phishing*.

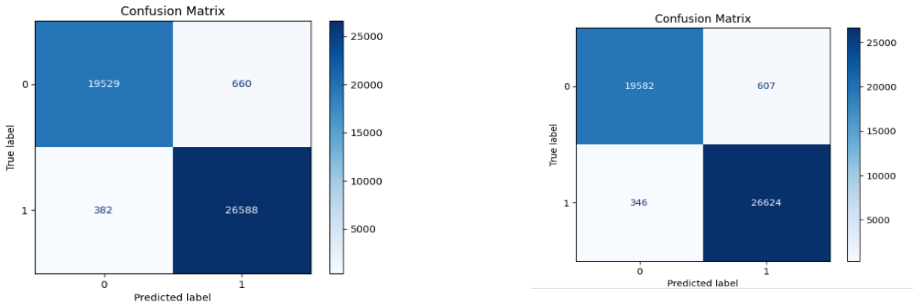
```
=== Hasil Cross-Validation ===
Accuracy CV per fold: [0.98059796 0.98065098 0.98038592 0.98091603 0.98107506 0.98001484
 0.98022584 0.97741611 0.98070296 0.97996077]
ROC-AUC CV per fold: [0.99609181 0.99578997 0.99429977 0.99467612 0.99476968 0.99437053
 0.99531737 0.99435871 0.9951577  0.9940586 ]
PR-AUC CV per fold: [0.99573313 0.99510663 0.99360491 0.9936229  0.99386433 0.99314215
 0.99435775 0.99324245 0.99417671 0.99277476]
Rata-rata Accuracy CV: 0.9802
Rata-rata ROC-AUC CV: 0.9949
Rata-rata PR-AUC CV: 0.994

=== Hasil Evaluasi Test Set ===
Accuracy: 0.9798
ROC-AUC : 0.9947
PR-AUC  : 0.9932
```

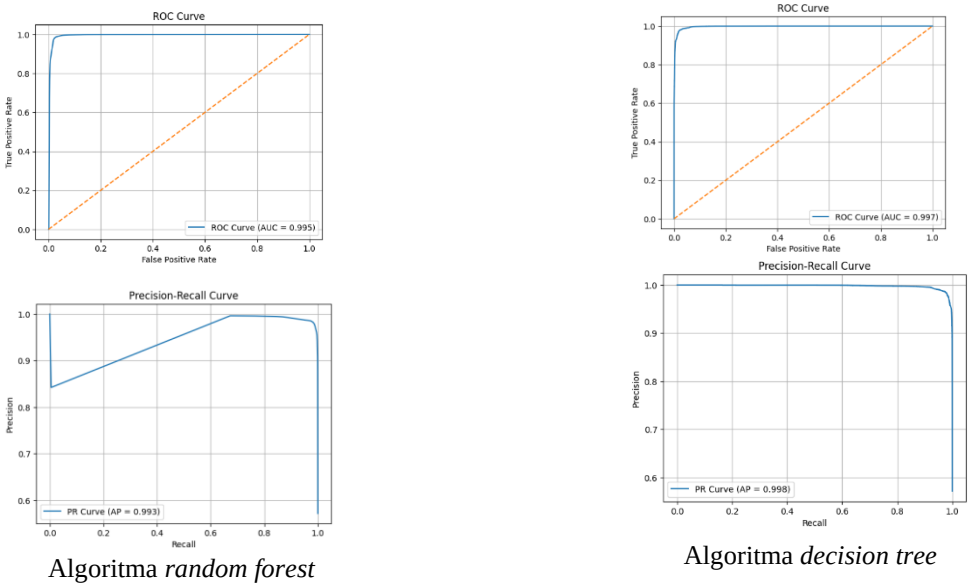
Gambar 3. Performa model *Decision Tree*

Berikut Perbandingan *confusion matrix* antara kedua algoritma

Algoritma *random forest* Algoritma *decision tree*



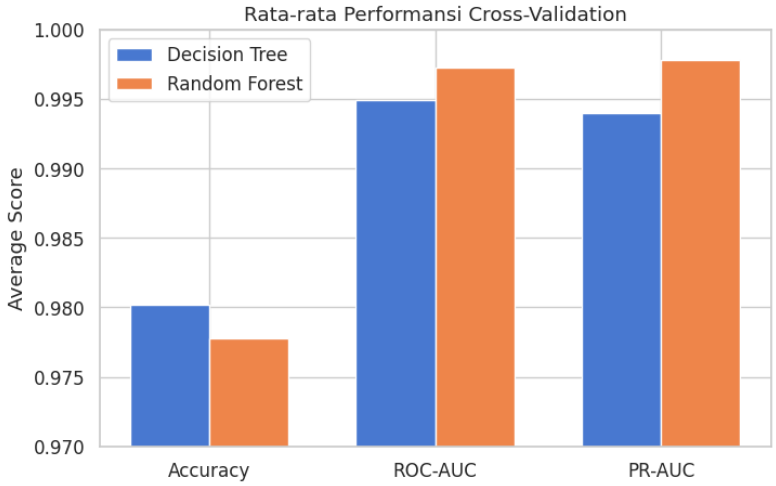
Gambar 4. Perbedaan *confusion matrix* kedua algoritma



Algoritma *random forest* Algoritma *decision tree*

Gambar 5. Perbedaan curva *ROC-AUC* dan *PR-AUC*

Meskipun *Accuracy* sedikit lebih rendah dibandingkan *decision tree*, namun *random forest* menunjukkan keunggulan pada *ROC-AUC* dan *PR-AUC*, yang menandakan kemampuan model dalam membedakan kelas dan mempertahankan presisi lebih baik.



Gambar 6. Perbandingan Performa dari ke-dua model

Dan hasil uji signifikansi menggunakan *wilcoxon test* menunjukan bahwa ada perbedaan yang signifikan performa akurasi, *ROC-AUC* dan *PR-AUC* antara Algoritma *decision tree* dan *random forest*

```
=== Wilcoxon Test: Accuracy ===
Statistic W = 0.0000
P-value = 0.0020
Kesimpulan: Perbedaan SIGNIFIKAN

=== Wilcoxon Test: ROC-AUC ===
Statistic W = 0.0000
P-value = 0.0020
Kesimpulan: Perbedaan SIGNIFIKAN

=== Wilcoxon Test: PR-AUC ===
Statistic W = 0.0000
P-value = 0.0020
Kesimpulan: Perbedaan SIGNIFIKAN
```

Gambar 7. Hasil uji *wilcoxon*

IV. KESIMPULAN

Penelitian ini menunjukkan bahwa pendekatan *machine learning* sangat efektif dalam mendeteksi situs web *phishing.* Dengan menggunakan dataset dari *UCI Repository* yang terdiri dari 235.795 data (134.850 *URL non-phishing* dan 100.945 *URL phishing*) dengan 21 fitur terbaik yang dipilih menggunakan algoritma pembelajaran mesin *decision tree dan random forest.* Berdasarkan evaluasi menggunakan *cross-validation 10-fold* dan test set, model *decision tree* menunjukkan akurasi yang sedikit lebih tinggi (rata-rata 98,02%) dibanding *random forest* (rata-rata 97,78%), namun *random forest* unggul pada kemampuan diskriminasi dan presisi, dengan *ROC-AUC* dan *PR-AUC* masing-masing 99,73%) dan 99,78% dibanding *decision tree* yang sebesar 99,49% dan 99,40%. Hal ini menunjukkan bahwa meskipun *decision tree* memiliki performa klasifikasi yang baik, namun *random forest* lebih konsisten dalam membedakan kelas *phishing* dan mempertahankan presisi tinggi, sehingga menjadi pilihan yang lebih handal untuk deteksi *web phishing.*

REFERENSI

[1] I. Saha, D. Sarma, R. J. Chakma, M. N. Alam, A. Sultana, and S. Hossain, "Phishing attacks detection using deep learning approach," *Proc. 3rd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2020*, no. Icssit, pp. 1180–1185, 2020, doi: 10.1109/ICSSIT48917.2020.9214132.

[2] A. V Ramana, K. L. Rao, and R. S. Rao, "Stop-Phish: an intelligent phishing detection method using feature selection ensemble," *Soc. Netw. Anal. Min.*, vol. 11, no. 1, p. 110, 2021, doi: 10.1007/s13278-021-00829-w.

[3] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, no. April, pp. 47–57, 2021, doi: 10.1016/j.comcom.2021.04.023.

[4] P. E. Reports, P. S. Trends, B. P. Measurement, E. P. Attacks, M. Targeted, and I. Sectors, "1 Quarter," no. March, pp. 1–7, 2023.

[5] Tedyyana, A., Ghazali, O., & W. Purbo, O. (2024). Enhancing intrusion detection system using rectified linear unit function in pigeon inspired optimization algorithm. IAES International Journal of Artificial Intelligence (IJ-AI), 13(2), 1526-1534. doi:http://doi.org/10.11591/ijai.v13.i2.pp1526-1534.

[6] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommun. Syst.*, vol. 76, no. 1, pp. 139–154, 2021, doi: 10.1007/s11235-020-00733-2.

[7] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and A. N. Saritha, "Phishing detection using random forest, SVM and neural network with backpropagation," *Proc. Int. Conf. Smart Technol. Comput. Electr. Electron. ICSTCEE 2020*, pp. 391–394, 2020, doi: 10.1109/ICSTCEE49637.2020.9277256.

[8] E. Zhu, Y. Ju, Z. Chen, F. Liu, and X. Fang, "DTOF-ANN: An Artificial Neural Network phishing detection model based on Decision Tree and Optimal Features," *Appl. Soft Comput. J.*, vol. 95, p. 106505, 2020, doi: 10.1016/j.asoc.2020.106505.

[9] M. H. Alkawaz, S. J. Steven, A. I. Hajamydeen, and R. Ramli, "A comprehensive survey on identification and analysis of phishing website based on machine learning methods," *ISCAIE 2021 - IEEE 11th Symp. Comput. Appl. Ind. Electron.*, pp. 82–87, 2021, doi: 10.1109/ISCAIE51753.2021.9431794.

[10] J. Stobbs, B. Issac, and S. M. Jacob, "Phishing web page detection using optimised machine learning," *Proc. - 2020 IEEE 19th Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2020*, pp. 483–490, 2020, doi: 10.1109/TrustCom50675.2020.00072.

[11] R. Yadav, R. Goyal, and A. Mittal, "Detection of Phishing Websites using Machine Learning Algorithm," *15th Int. Conf. Adv. Comput. Control. Telecommun. Technol. ACT 2024*, vol. 2, no. 05, pp. 962–966, 2024.

[12] V. Adeyemi Onih, "Phishing Detection Using Machine Learning: A Model Development and Integration," *Int. J. Sci. Manag. Res.*, vol. 07, no. 04, pp. 27–63, 2024, doi: 10.37502/ijsmr.2024.7403.
[13] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Networks*, vol. 178, no. January, 2020, doi: 10.1016/j.comnet.2020.107275.
[14] W. S. Lestari and M. Ulina, "Optimizing Deep Neural Networks Using ANOVA for Web Phishing Detection," *Teknika*, vol. 13, no. 1, pp. 71–76, 2024, doi: 10.34148/teknika.v13i1.758.
[15] A. B. Machine and L. Approach, "THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE Phishing URL Detection :," vol. 12, no. 3, pp. 8–14, 2024.
[16] S. H. Ahammad *et al.*, "Phishing URL detection using machine learning methods," *Adv. Eng. Softw.*, vol. 173, no. 8, pp. 198–202, 2022, doi: 10.1016/j.advengsoft.2022.103288.
[17] M. Kathiravan, V. Rajasekar, S. J. Parvez, V. Sathya Durga, M. Meenakshi, and S. Gowsalya, "Detecting Phishing Websites using Machine Learning Algorithm," *Proc. - 7th Int. Conf. Comput. Methodol. Commun. ICCMC 2023*, vol. 13, no. 4, pp. 270–275, 2023, doi: 10.1109/ICCMC56507.2023.10083999.
[18] S. Das Guptta, K. T. Shahriar, H. Alqahtani, D. Alsalman, and I. H. Sarker, "Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques," *Ann. Data Sci.*, vol. 11, no. 1, pp. 217–242, 2024, doi: 10.1007/s40745-022-00379-8.
[19] S. Abad, H. Gholamy, and M. Aslani, "Classification of Malicious URLs Using Machine Learning," *Sensors*, vol. 23, no. 18, 2023, doi: 10.3390/s23187760.
[20] Riganda Saputra and Ery Hartati, "Deteksi Website Phising Menggunakan Algoritma Random Forest Dengan Optimalisasi Gridsearch," *JUTIM (Jurnal Tek. Inform. Musirawas)*, vol. 10, no. 1, pp. 55–67, 2025, [Online]. Available: https://jurnal.univbinainsan.ac.id/index.php/jutim/article/view/2674
[21] S. Sudriyanto, M. A. Hafid, and M. A. Kurniawan, "Deteksi Akun Kaggle Bot Menggunakan Linear Regression," *J. Electr. Eng. Comput.*, vol. 6, no. 2, pp. 449–459, 2024, doi: 10.33650/jeecom.v6i2.9251.
[22] V. Indriani and H. Listiyono, "Deteksi Phishing Website Menggunakan Support," vol. 9, no. 2, pp. 3107–3113, 2025.
[23] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Syst. Appl.*, vol. 117, pp. 345–357, 2019, doi: 10.1016/j.eswa.2018.09.029.
[24] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," *Proc. - IEEE INFOCOM*, pp. 191–195, 2011, doi: 10.1109/INFCOM.2011.5934995.
[25] I Made Suartana, "Analisis Penerapan Deep Learning untuk Klasifikasi Serangan Terhadap Keamanan Jaringan," *Klik-Kumpulan J. Ilmu Komput.*, vol. 9, no. 1, pp. 100–109, 2022.
[26] I. W. Saputro and B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, p. 1, 2020, doi: 10.24076/citec.2019v6i1.178.