Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi, Vol. 10, No. 1, Februari 2024, Hal. 1-8
e-ISSN 2502-8995 p-ISSN 2460-8181

ANALISIS PENERAPAN ALGORITMA *DECISION TREE* DALAM KEAMANAN SIBER UNTUK KELASIFIKASI SITUS *WEBSITE PHISHING*

[1] Fitra Salam S. Nagalay , [2*] Sriyanto , [3] Zarnelly
[1,2] Fakultas Ilmu Komputer, Institut Informatikan dan Bisnis Darmajaya, Jl. Zainal Abidin PA No.93 Bandar Lampung 35142, Indonesia, [3] Sistem Informasi, Fakultas Sains dan Teknologi, UIN Suska Riau, Pekanbaru, Indonesia
Email: [1]fitra.2321211010@mail.darmajaya.ac.id, [2]sriyanto@darmajaya.ac.id, [3]zarnelly@uin-suska.ac.id

ABSTRAK

Dalam era teknologi yang semakin berkembang cepat dan ketergantungan masyrarakat pada internet, ancaman terhadap keamanan siber semakin beragam. Salah satu ancaman yang menonjol adalah kegiatan phising, dimana pihak tidak bertanggungjawab menggunakan alamat elektronik atau situs palsu untuk mendapatkan informasi peribadi pengguna. Ancaman phising tersebut tidak hanya mengancam keuangan, tetapi juga privasi pengguna. Penelitian ini menerapakan algoritma Decision Tree dalam klasifikasi. Algoritma Decision Tree terpilih karena kemudahan pemahaman dan interpretasinya serta kemampuannya dalam melakukan klasifikasi dengan baik. Fokus penelitian ini adalah mendalami potensi algoritma Decision Tree dalam mendeteksi situs website phising dengan menerapkan fitur-fitur spesifik seperti panjang URL, rangking, durasi aktif, dan karakteristik lainnya. Eksperimen dilakukan dengan menggunakan dataset yang mencakup berbagai fitur terkait URL dan dilakukan evaluasi menggunakan metrik seperti akurasi, presisi, dan recall. Hasil penelitian menunjukan bahwa model Decision Tree yang dikembangkan berhasil mencapai tingkat akuraasi sebesar 87.04%, memberikan kontribusi positif terhadap upaya mengamankan pengguna dari situs website phising.

Kata kunci: *Decison Tree, Klasifikasi, Keamanan Siber, Website Phishing*

Abstract

In an era of rapidly advancing technology and increasing societal dependence on the internet, the cybersecurity threats have become more diverse. One prominent threat is phishing, whre malicious actors use fake email addresses or websites to obtain users' personal information. This phishing threat not only jeopardizes financial security but also compromises user privacy. This research applies the Decision Tree algorithm for classification purposes. The Decision Tree algorithm was chosen for its ease of understanding and interpretation, as well as its capability to perform effective classification. The focus of this research is to explore the potential of the Decision Tree algorithm in detecting phishing websites by applying specific features such as URL length, rangking, active duration, and other characteristics. Experiments weere conducted using a dataset that encompasses various URL-related features, and the evaluation was performed using metrics such as accuracy, precision, and recall. The research findings indicate that the developed Decision Tree model successfully achived an accuracy rate of 87.04%, contributing positively to efforts in securing users from phising websites.

Keywords*: Decision Tree, Classifiaction, Cybersecurity, Phishing Websites*

A. PENDAHULUAN

Dalam era teknologi yang semakin hari semakin berkembang cepat dan ketergantungan masyarakat terhadap teknologi informasi membuat masyrakat tidak bisa lepas dari internet maupun gadget [1]. Kemajuan teknologi sangat banyak membantu masyrakat dalam aktivitas sehari-hari, terkhususnya dalam pemanfaatkan internet, masyrakat menjadi terbantu dengan pemanfaat internet ini dalam berkomunikasi maupun mencari informasi. Sekitar 175,4 juta atau 64% masyarakat Indonesia sudah mengerti dalam memanfaatkan internet. Bersamaan dengan meningkatnya pengguna internet bahaya yang timbul terhadap keamanan internet pun semakin bervariasi. Salah satunya bahaya ancaman *phishing*

[2]. *Phishing* adalah kegiatan penipuan menggunakan alamt elektronik palsu maupun website palsu untuk menjebak pengguna dengan cara menjebak target agar secara tidak langsung memasukan informasi tentang data diri pribadi sesuai nama pengguna, alamat, tanggal lahir, kata sandi, informasi kartu kredit dan informasi sensitif lainnya [1]. Seseorang yang melakukan *phishing* ini disebut dengan *pisher* [2]. Sebagian besar dari *pisher* menjalankan aksi kejahatannya dengan membuat sebuah *website* yang menyerupai situs *website* aslinya (tampilan, URL domain, atau lainnya) untuk mengelabui pengguna sehingga pengguna mengklik seolah-olah sedang mengakses halaman situs asli [3]. *Website* yang banyak dijadikan sasaran *phishing* biasanya *website* yang berhubungan dengan *online bangking* karena memiliki kesempatan untuk datanya diambil lebih tinggi dibandingkan dengan *website* pada umumnya [1].

Ancaman situs *website phishing* ini tidak hanya dapat menyebabkan kerugian finasial, tetapi juga membahayakan privasi pennggguna. Dalam rangka melawan dan mencegah situs *website phishing*, diperlukan pendekatan proaktif yang mampu mengidentifikasi dengan cepat dan tepat. Salah satu metode yang menarik perhatian adalah penerapan algoritma *Decision Tree* dalam klasifikasi situs *website phishing.*

Algoritma yang banyak digunakan dalam klasifikasi adalah *Decision Tree*. Alasan mengapa algoritma *Decision Tree* ini paling banyak digunakan dalam klasifikasi karena mudah dimengerti dan dijelaskan oleh banyak pengguna, serta cabang pohonnya dapat disimpulkan dalam bentuk klasifikasi [4].

Penelitian ini bertujuan untuk mendalami potensi algoritma *Decison Tree* dalam mendeteksi situs *website phishing.* Dengan melakukan identifikasi dan menganalisis karakteristik utama dalam membedakan suatu situs *website phising* dengan yang asli, serta dengan menerapkan algoritma *Decision Tree* untuk klasifikasi situ *website phising*, diharapkan dapat mengukur dan mengevaluasikinerja algoritma tersebut dalam mendeteksi sebuah situs *website* phising berdasarkan fitur-fitur yang spesifik yang mencakup aspek-aspek seperti panjang URL, rangking, durasi aktif, dan karakteristik lainnya sehingga dapat meningkatkan ketepatan dalam megklasifikasikan situs *website* dan mengurangi risiko terhadap serangan *phising.*

B. LANDASAN TEORI

B.1. Decision Tree

Decion Tree adalah sebuah algoritma di *meachine learning* yang mempunyai kegunaan dalam mengeksplorasi data dan mendapatkan kaitan dengan beberapa data yang ada pada *dataset.* Perhitungan dalam *decison tree* berhubungan dengan *Gain* dan *entropy*. *Gain* dan *Entropy* dirumuskan pada persamaan (1) dan (2) [1].

$$Gain(S,A) = \text{Enthropy}(S) - \sum_{i=1}^{n} \frac{|si|}{|s|} \times Enthropy(S_i) \quad (1)$$

Dimana,
S : Himpunan kejadian
A : Atribut
N : Jumlah pembatas atribut A
|Si| : Jumlah kejadian pada pembatas ke-i
|S| : Jumlah kejadian dalam S

Kemudian persamaan *entropy* sebagai berikut.

$$\text{Entrhropy(S)} = \sum_{i=1}^{n} - pi \times log_2 pi \quad (2)$$

Dimana,
S : Himpunan kejadian
A : Fitur
N : Jumlah pembatas S
Pi : Proporsi dari Si kepada S

B.2. Klasifikasi

Klasifikasi merupakan sebuah proses atau operasi pencarian model atau tugas yang mampu menjelaskan dan memilah kelas data atau konsep-konsep. Tujuan dari proses ini adalah agar model tersebut bisa dipakai dalam memprediksi kelas suatu objek yang kelasnya belum diketahui [5].

B.3. Uji Normalitas

Tahap awal dalam pengolahan data adalah dengan dilakukannya uji normalitas. Uji normalitas dilakukan dengan untukk menilai sejauh mana data dari variabel yang diobservasi memiliki distribusi normal, yang dapat dikenali dengan bentuk kurva menyerupai lonceng.

Uji normalitas data beserta distribusi normal memakai nilai rata-rata dan standar deviasi. Data dianggap normal ketika kurva distribusinya sama baiknya dengan sebelah kiri maupun kanan. Kurva

distribusi normal dapat dihitung menggunakan persamaan berikut [5].

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, -\infty \leq x \leq \infty \quad (3)$$

Dimana f(x) merupakan fungsi distribusi normal, σ merupakan standar deviasi, dan μ merupakan data rata-rata. Kemudian untuk meraih nilai rata-rata agar bisa digunakan dapat dilihat rumus persamaan 4. Dan untuk mencari nilai standar deviasi bisa menggunakan rumus persamaan 5 [6].

$$X = \frac{\sum x_i}{n} \quad (4)$$

Dimana X adalah nilai rata-rata, sementara xi merupaak data ke-i dan n merupakan total dari datanya.

$$S = \sqrt{\frac{\sum(xi - x)^2}{n-1}} \quad (5)$$

Dimana S adalah standar deviasi, yang didapat dengan nilai akar dari total jumlah bagi kuadrat selisih data ke-i dana nilai rata-rata dikurang 1.

B.4. Normalisasi

Normalisasi adalah sebuah proses untuk mengubah skala data menjadi 0-1. Proses normalisasi dirumuskan pada persamaan 6 [6].

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (6)$$

Dimana x_{norm} merupakan data yang telah di normalisasi, x_i merupakan data ke-i, x_{min} merupakan data yang paling kecil dan x_{max} adalah data yang paling besar.

C. METODE PENELITIAN

Terdapat 4 tahapan penelitian yang dilakukan, yaitu (1) Identifikasi masalah yang sebenarnya dalam situs *website phishing*, dimana dalam penelitian ini juga mencoba menjawab pertanyaan seputar sejauh mana algoritma *Decision Tree* efektif dalam mengidentifikasi situs *website phishing*, (2) Tahap Persiapan, (3) Preprocessing, uji normalitas dan training data (4) Evaluasi. Tahapan penelitian tersebut digambarkan pada gambar 1.

Gambar 1. Tahapan Penelitian

C.1 Tahap Persiapan

Pada tahap persiapan ini dilakukan dengan menyiapkan *dataset.* Data yang dipakai dalam penelitian ini bersumber dari *website kaggle, dataset* ini berjumlah 10.306 data dan memiliki 11 atribut termasuk 1 kelas yang digunakan sebagai label. Adapun keterangan mengenai atribut data terdapat pada tabel 1.

Tabel 1. Atribut Dataset

No	Nama Atribute	Keterangan
1	Rangking	Peringkat Halaman
2	islp	Apakah terdapat alamat IP di tautan web
3	valid	Data ini diambil dari *google whois API* yang memberikan informasi lebih banyak tentang arus saat ini dari pendaftaran status URL
4	activeDuration	Juga dari *whois API* untuk memberikan durasi waktu sejak pendaftaran sampai sekarang
5	urlLen	Panjang URL
6	is@	Jika tautan link mempunyai karakter '@' maka nilainya = 1
7	isredirect	Jika tautan memiliki tanda hubung ganda, ada kemungkinan tautan tersebut dialihkan. 1-> banyak tanda hubung hadir bersama.
8	haveDash	Jika ada tanda hubung pada domain
9	domainLen	Panjang domain
10	noOfSubdomain	Jumlah subdomain yang telah ditetapkan sebelumnya di URL
11	Labels	0 = website yan sah , 1 = Phishing Link/ Spam Link

C.2 Preprocessing, Uji Normalitas, dan Training Data

Pada tahap ini data akan dilakukan proses *preprocessing* yang melibatkan pengelolaan data awal, seperti pembersihan data *(cleaning data).* Langkah ini bertujuan untuk menilai bahwasannya data yang dipakai pada pelatihan memiliki kualitas yang bagus [7]. Pembersihan data ini mencakup nilai-nilai yang hilang atau data yang duplikat.

Setelah tahap pembersihan data, selanjutnya uji normalitas, uji ini bertujuan untuk menguji sejauh mana distribusi data mendekati distribusi normal yang ditandai dengan bentuk kurva bergelombang seperti lonceng [6]. Pada uji normalitas ini memiliki dua metode yang sering dipakai, yaitu uji *Kolmogorov-Smirnov* dan uji *Shapiro-Wilk*. Dalam konteks uji normalitas, *Kolmogorov-Smirnov* dapat dipakai dalam memastikan sejauh mana sampel cocok atau tidak dengan distribusi normal. Jika nilai p-value dari uji ini lebih besar dari tingkat signifikansi yang ditemukan nilainya $\alpha = 5\% = 0{,}05$, maka bisa disimpulkan bahwasannya data yang dipakai itu berasal dari distribusi normal. Sementara uji *Shapiro-Wilk* ini nilai p-value ini dapat dipakai dalam memilih apakah suatu *dataset* memliki distribusi normal, kalau nilai p-value lebih besar dari tingkat signifikansi yang ditentukan maka bisa dibilang data tersebut berasal dari distribusi normal [8].

Adapun jika data yang digunakan bersumber dari distribusi tidak normal maka dilakukan proses normalisasi. Proses normalisasi ini adalah proses mengubah skala data memiliki nilai rata-rata mendakti 0 dan standar deviasi mendakati 1. Proses normalisasi yang digunakan dalam penelitian ini menggunakan metode *min-max.* Metode ini dilakukan dengan hasil pengurangan data ke-i itu dikurang data terkecil kemudian dibagi data terbesar dikurang data terkecil kemudian hasil tersebut dibagi [6].

Setelah tahap uji normalitas atau normalisasi data, dataset akan dilatih dengan menggunakan algoritma *Decision Tree*. Dataset akan dibagi menjadi set pelatihan dan pengujian. Data pelatihan ini adalah data yang ada sebelumnya yang sesuai dengan fakta sementara data uji adalah data yang digunakan untuk menilai sejauh mana pengklasifikasian berhasil mengkelompokkan dengan tepat [9]. Proses pelatihan ini bertujuan untuk membangun model *Decision Tree* berdasarkan pola-pola yang terdapat dalam data pelatihan.

C.3 Evaluasi

Pada tahap evaluasi ini dilakukan dengan cara menganalisa hasil yang diperoleh dari model *Decision Tree* guna memastikan kesesuaian dengan tujuan penelitian. Validasi dijalankan untuk mengukur kinerja klasifikasi dengan fokus pada akurasi *(accuracy)*, presisi *(precision)*, dan *recall* atau bisa disebut dengan metric akurasi [10].

a) Akurasi

Akurasi mengukur sejauh mana model dapat mengklasifikasikan situs *website* dengan benar, baik *phishing* maupun *non- phishing [11]*. Formula akurasi dapat ditulis menggunakan persamaan 1.

$$\frac{TP+TN}{Total} \times 100\% \quad (1)$$

Formula tersebut didefinisiakan dengan rasio % dimana total data *true positive (TP)* dijumlahkan dengan total data *true negative (TN)* dibagi total jumlah data uji [10].

b) Presisi

Presisi menggambarkan seberapa baik model dalam mengkalsifikasikan sebuah situs *website* sebagai *phishing* yang sebenarnya [10]. Formula presisi dapat ditulis menggunakan persamaan 7.

$$\frac{TP}{TP+FP} \times 100\% \quad (7)$$

Formula tersebut didefinisiakan dengan rasio % dimana total data *true positive (TP)* dibagi dengan total data *true positive (TP)* dijumlahkan *false positive (FP)* jumlah situs *website* yang diklasifikasikan sebagai *phishing.*

c) *Recall*

Recall mengukur kemampuan model untuk mengidentifikasi dan mengklasifikasikan situs *website phishing* dengan benar [10]. Formula *recall* dapat ditulis menggunakan persamaan 8.

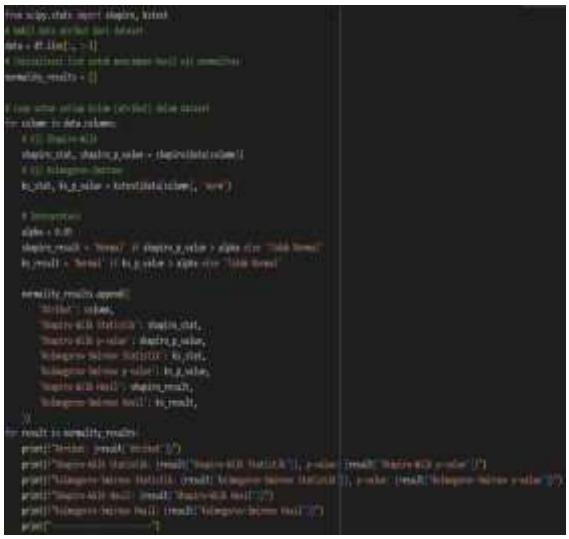
$$\frac{TP}{TP+FN} \times 100\% \quad (8)$$

Formula tersebut didefinisiakan dengan rasio % dimana total data *true positive (TP)* dibagi dengan total data *true positive (TP)* ditambah *false negative (FN)* jumlah situs *website phishing [11]*.

Evaluasi ini sangat penting dalam memastikan bahwa model *Decision Tree* dapat memberikan hasil yang efektif dalam mendeteksi situs *website phishing*.

D. HASIL DAN PEMBAHASAN

Pada penelitian ini *dataset* yang sudah di lakukan pembersihan data *(cleaning data).* Maka data tersebut akan dijalan proses uji normalitas menggunakan uji normalitas *kolmogorov-smirnov dan shapiro-wilk.* Berikut gambar dari penulisan *code:*



Gambar 2. *Code Uji Normalisasi*

Pada gambar *code* tersebut bisa dilihat untuk nilai pengambilan keputusan ditentukan α = 5% = 0,05. Kemudian kriteria yang diambil menggunakan sig.α atau p-value, apabila sig.α < α maka B tidak normal. berikut hasil dari uji normalitas *kolmogorov-smirnov dan shapiro-wilk* untuk setiap atribut dalam dataset.

Tabel 2. Hasil Uji Normalitas

Atribute	Shapiro-Wilk Statistik	K-S statistik	Shapiro-Wilk & K-S p-value
Rangking	0.945	0.095	0.000
islp	0.606	0.161	0.000
valid	0.845	0.091	0.000
activeDuration	0.885	0.080	0.000
urlLen	0.928	0.053	0.000
is@	0.282	0.280	0.000
isredirect	0.882	0.081	0.000
haveDash	0.734	0.112	0.000
domainLen	0.878	0.083	0.000
noOfSubdomain	0.782	0.122	0.000

Hipotesis:

H0: Data berdistribusi Normal

H1: Data tidak berdistribusi Normal

Dengan α = 5% = 0,05, hasil analisis data tersebut didapatkan bahwasannya nilai p-value pada uji *kolmogorov-smirnov dan shapiro-wilk* nilai dari masing-masing atribut lebih kecil dari α maka tolak H0 atau data tersebut tidak normal.

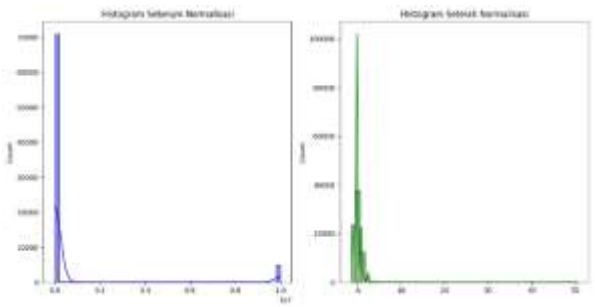
Karena data yang didapatkan tidak normal, proses selanjutnya akan dilakukan normalisasi dengan menggunakan metode *min-max.* Berikut gambar penulisan *code:*

Gambar 3. *Code Normalisasi*

Pada *code* tersebut dilakukan proses normalisasi, dan beberapa baris data hasil normalisasi bisa dilihat pada gambar berikut:

Gambar 4. Hasil Normalisasi

Agar lebih jelas lagi hasil dari data yang sudah di normalisasi maka divisualisasikan dengan histogram perbandingan data sebelum normalisasi dan setelah normalisasi. Bisa dilihat pada Gambar 5.



Gambar 5. Histogram Sebelum dan Sesudah Normalisasi

Bisa dilihat pada gambar histogram tersebut sebelum normalisasi dan setelah normalisasi memiliki perbedaan dimana data setelah dinormalisasikan bentuk histogramnya menyerupai lonceng.

Setelah data tersebut sudah dinormalisasikan maka data yang sudah dipersiapkan selanjutnya dilatih. Secara khusus, ini mencakup langkah-langkah berikut:

a) Penskalaan Fitur

Pada langkah ini, peneliti menormalkan (mengubah skala) fitur menggunakan *StandarScaler*. Ini adalah langkah pra-pemrosesan umum untuk memastikan bahwa semua fitur memiliki skala yang beragam. Berikut gambar penulisan *code:*

```
scaler = StandardScaler()
scaler.fit(X)
X = scaler.transform(X)
```

Gambar 6. *Code* pra-pemrosesan

b) Pemisahan Dataset

Langkah selanjutnya dari penelitian ini, membagi *dataset* menjadi set pelatihan dan pengujian menggunakan *train_test_split* [5]. Ini penting untuk menilai kinerja model pada data yang belum pernah dilihat sebelumnya. Berikut gambar penulisan *code:*

Gambar 7. *Code* Set Pelatihan dan Set Pengujian

Dapat dilihat pada potongan *code* tersebut dataset dipisahkan menjadi 20% data pengujian dan 80% data pelatihan yang bisa dilihat pada potohan *code test_size=0.2*. Data pelatihan tersebut digunakan untuk pelatihan model sedangkan data uji untuk menguji peforma model.

D.1 Implementasi Model Decision Tree

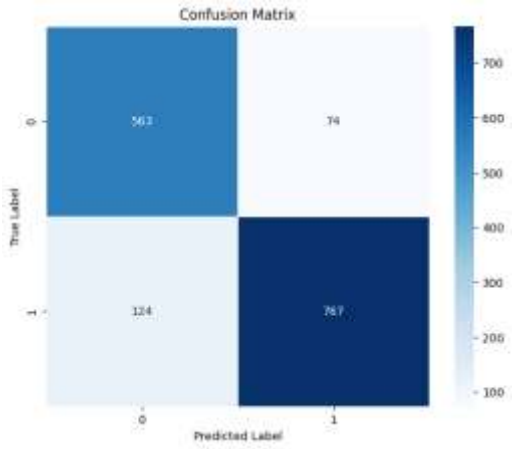
Model *Decision Tre* diimplementasikan menggunakan *library scikit-learn*pada bahasa pemrograman Python. Berikut gambar penlisan *code:*

```
from sklearn.metrics import confusion_matrix

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Prediksi pada data uji
y_pred = clf.predict(X_test)
```

Gambar 8. *Code* Implementasi Model *Decsion Tree*

Pada proses tersebut dilakukan penginisiasian model, dimana membuat objek *DecisionTreeClassifier*, kemudian model tersebut dilatih dengan data pelatihan (X_train, y_train), dan setelah itu melakukan prediksi pada data uji (X_test) menggunakan model yang telah dilatih. Hasil prediksi tersebut disimpan dalam y_pred.

Setelah dilakukan uji, kemudian selanjutnya adalah membuat *confusion matriks. Confusion matrix* ini dibuat dengan tambahan *library Seaborn* untuk divisualisasikan. Dengan *confusion matriks* ini dapat menunjukan jumlah prediksi yang benar dan salah dari masing-masing kelas positif dan negatif. Berikut gambar *confusion matrix:*



Gambar 9. *Confusion Matrix*

Pada gambar tersebut pada *confusion matrix* dapat dilihat bahwa 767 total sampel yang betul-betul terhitung dalam kelas positif *(phishing)* dan juga diprediksi dengan benar oleh model. 74 total sampel yang sebenarnya terhitung dalam kelas negatif *(legitimate)*, tetapi salah diprediksi sebagai kelas positif *(phishing)* oleh model. 124 total sampel yang

sebenarnya terhitung dalam kelas positif *(phishing)*,tetapi salah diprediksi sebagai kelas negatif *(legitimate)* oleh model. 563 total sampel yang betul-betul terhitung dalam kelas negatif *(legitimate)* dan juga diprediksi dengan benar sama model.

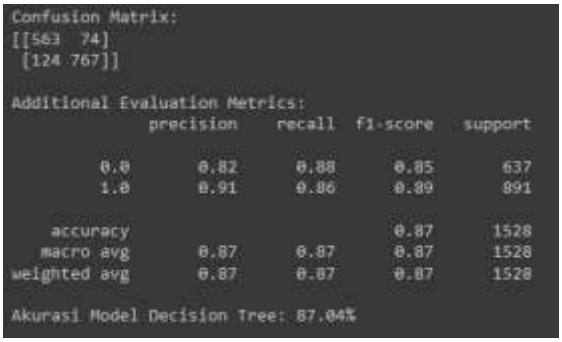
D.2 Evaluasi Model

Evaluasi model dilakukan dengan menganilisis hasil dari prediksi terhadap data uji. Berikut gambar penulisan *code*:

```
# matriks kebingungan
conf_matrix = confusion_matrix(y_test, y_pred)

# tampilkan matriks kebingungan
print("Confusion Matrix:")
print(conf_matrix)

# metrik evaluasi tambahan
print("\nAdditional Evaluation Metrics:")
print(classification_report(y_test, y_pred))

# akurasi model
decision_tree_acc = accuracy_score(y_test, y_pred)
print("Akurasi Model Decision Tree: {:.2f}%".format(decision_tree_acc * 100))
```

Gambar 10. *Code* Menampilkan Hasil Evaluasi Model

Pada proses *code* tersebut menampilkan matriks kebingungan yang akan menunjukan *true positive, true negative, false positive,* dan *false negative,* kemudian menampilkan *precesion, recall,* dan *f1-score* untuk setiap kelas, dan tidak kalah penting untuk menunjukan akurasi model, seberapa baik model tersebut dapat mengklasifikasikan data dengan benar. Berikut gambar dari output yang dihasilkan dari *code* tersebut:



```
Confusion Matrix:
[[563  74]
 [124 767]]

Additional Evaluation Metrics:
              precision    recall  f1-score   support

         0.0       0.82      0.88      0.85       637
         1.0       0.91      0.86      0.89       891

    accuracy                           0.87      1528
   macro avg       0.87      0.87      0.87      1528
weighted avg       0.87      0.87      0.87      1528

Akurasi Model Decision Tree: 87.04%
```

Gambar 11. *Output*

Pada tahap evaluasi, model *Decision Tree* menghasilkan peforma yang memuaskan dengan akurasi sebesar 87.04%. Hasil tersebut menunjukan bahwa model *Decision Tree* mampu membedakan situs *website phishing* dengan baik . *Precesion* dan *recall* yang tinggi pada kelas *phishing* menunjukan kemampuan model dalam mengidentifikasi situs *website* yang berpotensi berbahaya.

E. KESIMPULAN

Dalam penelitian ini, penerapan algoritma *Decision Tree* telah dilakukan dalam melakukan klasifikasi situs *website* sebagai *phishing* atau tidak. Hasil eksperimen menunjukan bahwa model *Decision Tree* yang dikembangkan memiliki tingkat akurasi sebesar 87.04%, dengan mampu mengidentifikasi dengan baik antara situs *phishing* dan situs *non-phishing (legitimate).*

Penerapan fitur-fitur spesifik seperti panjang URL rangkin, durasi aktif, dan karakteristik lainnya memberikan kontribusi yang signifikan terhadap kemampuan model dalam mengklasifikasi situs *website.*

Adapun implikasi dari penelitian ini mencakup potensi penggunaan *algoritma Decision Tree* sebagai salah satu metode yang efektif dalam mengamankan situs *website* dari aktivitas *phishing*. Dari hasil yang diperolah dalam penelitian ini bahwasannya dapat dijadikan suatu dasar dalam pengembangan deteksi *phising* yang lebih baik dan canggih lagi. Sebagai saran untuk penelitian selanjutnya, dapat dilakukan eksplorasi lebih lanjut terhadap penggunaan algoritma *machine learning* lainnya dan penambahan fitur-fitur baru yang mungkin dapat meningkatkan tingkat akurasi model yang lebih bagus lagi dalam mengklasfikasikan situs *website phising*.

F. REFERENSI

[1] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, dan A. Setiawan, “Deteksi Website Phishing Menggunakan Teknik Filter Pada Model Machine Learning,” *Inf. Syst. J.*, vol. 6, no. 01, hal. 39–43, 2023, doi: 10.24076/infosjournal.2023v6i01.1268.

[2] A. S. Y. Irawan, N. Heryana, H. S. Hopipah, dan D. Rahma, “Identifikasi Website Phishing dengan Perbandingan Algoritma Klasifikasi,” *Syntax J. Inform.*, vol. 10, no. 01, hal. 57–67, 2021, doi: 10.35706/syji.v10i01.5292.

[3] R. P. Ramadhan dan T. Desyani, “Implementasi Algoritma J48 Untuk Identifikasi Website Phising,” *BINER J. Ilmu Komput. …*, vol. 1, no. 2, hal. 46–54, 2023, [Daring]. Tersedia pada: https://journal.mediapublikasi.id/index.php/Biner/article/view/2557%0Ahttps://journal.mediapublikasi.id/index.php/Biner/article/download/2557/1331.

[4] A. S. Sunge, “Optimasi Algoritma C4.5 Dalam

Prediksi Web Phishing Menggunakan Seleksi Fitur Genetic Algoritma," *Paradigma*, vol. 10, no. 2, hal. 27–32, 2018, doi: 10.31294/p.v%vi%i.4021.

[5] A. H. Nasrullah, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Produk Laris," *J. Ilm. Ilmu Komput.*, vol. 7, no. 2, hal. 45–51, 2021, doi: 10.35329/jiik.v7i2.203.

[6] J. Cahyani, S. Mujahidin, dan T. P. Fiqar, "Implementasi Metode Long Short Term Memory (LSTM) untuk Memprediksi Harga Bahan Pokok Nasional," *J. Sist. dan Teknol. Inf.*, vol. 11, no. 2, hal. 346, 2023, doi: 10.26418/justin.v11i2.57395.

[7] M. Adipa, Ahmad Turmudi Zy, dan M. Makmun Effendi, "Klasifikasi Email Phishing Menggunakan Algoritma K-Nearest Neighbor," *J. RESTIKOM Ris. Tek. Inform. dan Komput.*, vol. 5, no. 2, hal. 148–157, 2023, doi: 10.52005/restikom.v5i2.152.

[8] A. Quraisy, "Normalitas Data Menggunakan Uji Kolmogorov-Smirnov dan Saphiro-Wilk," *J-HEST J. Heal. Educ. Econ. Sci. Technol.*, vol. 3, no. 1, hal. 7–11, 2022, doi: 10.36339/jhest.v3i1.42.

[9] Baiq Nurul Azmi, Arief Hermawan, dan Donny Avianto, "Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 4, no. 4, hal. 281–290, 2023, doi: 10.35746/jtim.v4i4.298.

[10] I Made Suartana, "Analisis Penerapan Deep Learning untuk Klasifikasi Serangan Terhadap Keamanan Jaringan," *Klik-Kumpulan J. Ilmu Komput.*, vol. 9, no. 1, hal. 100–109, 2022.

[11] I. W. Saputro dan B. W. Sari, "Uji Performa Algoritma Naïve Bayes untuk Prediksi Masa Studi Mahasiswa," *Creat. Inf. Technol. J.*, vol. 6, no. 1, hal. 1, 2020, doi: 10.24076/citec.2019v6i1.178.