Sistemasi: Jurnal Sistem Informasi
Volume 13, Nomor 4, 2024: 1368-1380
ISSN:2302-8149
e-ISSN:2540-9719

Deteksi *Phishing Website* menggunakan *Machine Learning* Metode Klasifikasi

Phishing Website Detection using Machine Learning Classification Method

[1]Azzam Fawwaz Mahmud*, [2] Setia Wirawan
[1]Perangkat Lunak dan Sistem Informasi, Manajemen Sistem Informasi, Universitas Gunadarma
[2]Sistem Informasi, Ilmu Komputer dan TI, Universitas Gunadarma
[*]e-mail: azzamfmahmud@gmail.com

(***received*:** 24 August 2023, ***revised*:** 10 March 2024, ***accepted*:** 19 July 2024)

Abstrak

Phishing website merupakan mekanisme kriminal yang menggunakan *social engineering* serta dalih teknis untuk mengambil data identitas personal dan kredensial akun keuangan dari pelanggan. Di Indonesia sendiri menurut laporan Pengelola Nama Domain Internet Indonesia (Pandi), tercatat jumlah phishing dalam kurun waktu 5 tahun terakhir mencapai 34.622. Jumlah serangan *phishing* unik yang dilaporkan pada Q3 2022 sebanyak 7.988. Penelitian ini bertujuan untuk mencari algoritma *machine learning* klasifikasi dengan performa terbaik untuk mendeteksi *phishing website* menggunakan fitur-fitur URL. Algoritma klasifikasi yang akan dibandingkan adalah *Decision Tree*, *Random Forest*, dan KNN. Hasil dari penelitian ini adalah model pertama yang menggunakan *Decision Tree* didapat akurasi sebesar 0.833, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83. Model kedua yang menggunakan algoritma *Random Forest* mendapat akurasi sebesar 0.834, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83. Model terakhir yang menggunakan algoritma *K-Nearest Neighbors* mendapat akurasi sebesar 0.482, presisi sebesar 0.24, *recall* sebesar 0.50, dan *F1-score* sebesar 0.48. Maka, dari ketiga algoritma tersebut *random forest* merupakan algoritma terbaik untuk mendeteksi *phishing website*

Kata kunci: *phishing website*, *machine learning*, klasifikasi, *decision tree*, *random forest*, KNN

Abstract

Phishing websites are criminal mechanisms that use social engineering and technical pretexts to extract personal identification data and financial account credentials from customers. In Indonesia alone, according to a report by the Indonesian Internet Domain Name Manager (Pandi), the number of phishing recorded in the last 5 years has reached 34,622. The number of unique phishing attacks reported in Q3 2022 was 7,988. This study aims to find a classification machine learning algorithm with the best performance for detecting phishing websites using URL features. The classification algorithms to be compared are the decision tree, random forest, and KNN. The results of this study are that the first model that uses a decision tree obtains an accuracy of 0.833, a precision of 0.86, a recall of 0.83, and an F1-score of 0.83. The second model that uses the random forest algorithm gets an accuracy of 0.834, a precision of 0.86, a recall of 0.83, and an F1-score of 0.83. The last model that uses the K-Nearest Neighbors algorithm gets an accuracy of 0.482, a precision of 0.24, a recall of 0.50, and an F1-score of 0.48. Thus, of the three algorithms random forest is the best algorithm for detecting phishing websites

Keywords: *phishing website, machine learning, classification, decision tree, random forest, KNN*

1 Pendahuluan

Dalam era digital saat ini, internet telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari, dimana banyak transaksi dan komunikasi dilakukan secara online. Pemanfaatan internet telah merasuki hampir semua sektor dan industri, seperti *e-commerce*, transportasi, pariwisata, kesehatan,

 ISSN:2302-8149
e-ISSN:2540-9719

pemerintahan (*e-government*), dan industri keuangan. Di Indonesia, pada tahun 2019, jumlah pengguna internet mencapai 73,7% dari total populasi, menandakan peningkatan sebesar 8,9% dibandingkan tahun sebelumnya [1]. Hal ini mencerminkan pertumbuhan yang pesat dalam akses internet di Indonesia. Melalui internet, masyarakat Indonesia dapat mengakses berbagai konten dan situs web, yang telah menjadi sarana utama untuk mendapatkan informasi dan berinteraksi dalam kehidupan sehari-hari.

Dengan semakin canggihnya teknologi datanglah ancaman serius dalam bentuk serangan siber, salah satunya adalah serangan phishing. APWG (Anti-Phishing Working Group) mendefinisikan *phishing* sebagai mekanisme kriminal yang menggunakan *social engineering* serta dalih teknis untuk mengambil data identitas personal dan kredensial akun keuangan dari pelanggan [2].

Serangan *phishing* ini telah menimbulkan kerugian finansial dan kebocoran informasi pribadi yang signifikan bagi individu maupun organisasi. *Website* sendiri merujuk pada sekelompok halaman web yang memiliki domain yang sama dan digunakan untuk tujuan dan topik tertentu. *Website* memungkinkan berbagai aktivitas, seperti berbelanja daring, membaca berita, belajar secara daring, dan melakukan transaksi bisnis. Namun, penggunaan *website* juga membawa risiko kejahatan, terutama dalam bentuk phishing *website*. Phishing *website* adalah situs web yang meniru tampilan dari *website* resmi untuk tujuan mencuri informasi pengguna. Dalam praktiknya, banyak pengguna yang terjebak dan menjadi korban dari serangan phishing *website*. Data yang paling sering menjadi target dari phishing *website* adalah username, password, dan nomor PIN.

Menurut APWG, kesadaran masyarakat terhadap situs phising meningkat setiap tahun, namun jumlah situs phising dan kerugian yang ditimbulkan meningkat lebih cepat. Dalam laporan APWG kuartal keempat 2016, tercatat 89.232 situs yang terdeteksi sebagai situs phising pada bulan Oktober 2016. Kemudian, jumlah situs yang terindikasi sebagai situs phising meningkat menjadi 118.928 dan 69.533 pada bulan November dan Desember 2016. Laporan tersebut juga mencatat adanya sekitar 17 juta malware baru. Situasi ini dapat menciptakan ketakutan dan mengurangi kepercayaan pengguna internet dalam melakukan transaksi online, padahal transaksi online sedang booming di Indonesia. Berdasarkan laporan Kaspersky Security Network tahun 2021, serangan phishing menjadi ancaman keamanan siber utama di Indonesia, dengan terdeteksinya sekitar 1,6 juta serangan phishing selama kuartal keempat 2020 [3]. Serangan phishing juga merupakan ancaman terbesar di seluruh Asia Tenggara, dengan Indonesia menjadi yang tertinggi dalam jumlah serangan phishing.

Kerugian yang ditimbulkan oleh phishing *website* sendiri sangat besar. Menurut laporan Grand View Research, pasar keamanan siber global bernilai US$223 miliar di tahun 2022, dan diproyeksikan tumbuh sebesar 12,3% selama periode 2023-2030 [4]. Jumlah kasus serangan siber juga tercatat telah meningkat sebesar 13%. Sementara, Federal Bureau of Investigation (FBI) pada Maret 2023 melaporkan, kerugian akibat serangan siber sepanjang tahun 2022 telah mencapai lebih dari US$10 miliar. Jika dirupiahkan, maka nilai kerugian tersebut setara Rp147 triliun. Di Indonesia sendiri menurut laporan Pengelola Nama Domain Internet Indonesia (Pandi), tercatat jumlah phishing dalam kurun waktu 5 tahun terakhir mencapai 34.622. Jumlah serangan *phishing* unik yang dilaporkan pada Q3 2022 sebanyak 7.988. Sektor bisnis yang paling menjadi sasaran serangan phishing pada Q3 2022 adalah lembaga pemerintahan, sedangkan jumlah domain unik yang digunakan untuk serangan phishing pada Q3 2022 sebanyak 181. Dampak yang dapat ditimbulkan dari kejahatan *phishing website* ini cukup besar karena dapat mengurangi tingkat kepercayaan masyarakat saat bertransaksi secara daring sehingga bisa merugikan para pelaku usaha secara keseluruhan.

Machine learning adalah cabang kecerdasan buatan yang memungkinkan sistem untuk belajar dari data dan pengalaman sebelumnya, dan membuat prediksi atau pengambilan keputusan tanpa adanya instruksi eksplisit [5]. *Machine learning* telah membuktikan keefektifannya dalam berbagai bidang, termasuk deteksi serangan siber. Dengan menggunakan teknik *machine learning*, dapat dikembangkan model klasifikasi yang dapat mempelajari pola dan karakteristik dari situs web *phishing*, sehingga dapat membedakan dengan akurat antara situs web yang sah dan situs web yang berpotensi *phishing*. Model ini akan terus diperbarui dan ditingkatkan dengan melibatkan data baru yang ditemukan.

Penelitian ini bertujuan mengimplementasikan *machine learning* model klasifikasi untuk mendeteksi phishing *website* dengan tingkat keberhasilan yang tinggi. Penelitian ini akan fokus pada pengembangan model yang dapat mempelajari fitur-fitur penting dari situs web *phishing*, seperti struktur halaman dan alamat URL. Selain itu, penelitian ini juga akan mempelajari berbagai algoritma *machine learning* yang dapat digunakan untuk mengklasifikasikan situs web, seperti *Decision Tree*, *K-*

 ISSN:2302-8149
e-ISSN:2540-9719

Nearest Neighbor (*KNN*), dan *Random Forest*. Ketiga algoritma tersebut dipilih karena merupakan algoritma dengan performa terbaik dalam mendeteksi *phishing website* dan dapat diimplementasikan ke dalam aplikasi berbasis web. Untuk menentukan algoritma dengan performa terbaik dalam mendeteksi *phishing website*, ketiga algoritma tersebut akan dibandingkan menggunakan dataset seragam yang telah disiapkan sebelumnya.

2 Tinjauan Literatur

Berdasarkan penelitian sebelumnya yang dilakukan M. Korkmaz et al [6] dimana dilakukan perbandingan antara 8 *classifier* untuk mendeteksi *phishing website* menggunakan URL didapat hasil yang menunjukan bahwa algoritma *Random Forest* yang memiliki tingkat akurasi tertinggi. Jain et al [7] mengusulkan pendekatan anti-phishing berdasarkan algoritma *machine learning* hanya menggunakan fitur berbasis hyperlink tetapi memiliki masalah dalam salah mengklasifikasikan situs web non-html. Sedangkan M. Abutaha et al [8] melakukan perbandingan 3 algoritma klasifikasi, yaitu *Gradient Boosting Classifiers (GBC)*, *Support Vector Machine (SVM)*, dan *Random Forest*. Dari ketiga algoritma tersebut, *Random Forest* keluar sebagai algoritma dengan performa terbaik untuk mendeteksi *phishing website* menggunakan URL. Penelitian yang dilakukan oleh Sahingoz et al [9] juga menunjukkan *Random Forest* menghasilkan performa tertinggi dibanding tujuh algoritma klasifikasi yang digunakan dalam penelitian tersebut. Menurut resensi literatur (*literature review*) yang dilakukan oleh A. Safi et al [10], *random forest* merupakan algoritma yang paling sering digunakan untuk mengklasifikasi *phishing website* diikuti dengan algoritma *Support Vector Machine (SVM)*.

Menurut S. Shabudin et al [11] *phishing website* memiliki karakteristik atau ciri-ciri yang dapat dibedakan dengan *website* yang asli atau *legitimate website*. Karakteristik tersebut dapat dilihat pada Tabel 1 di bawah, pada Tabel 1 karakteristik *phishing website* dibagi menjadi 4 fitur besar yang kemudian dibagi kembali menjadi sub-sub fitur di bawahnya.

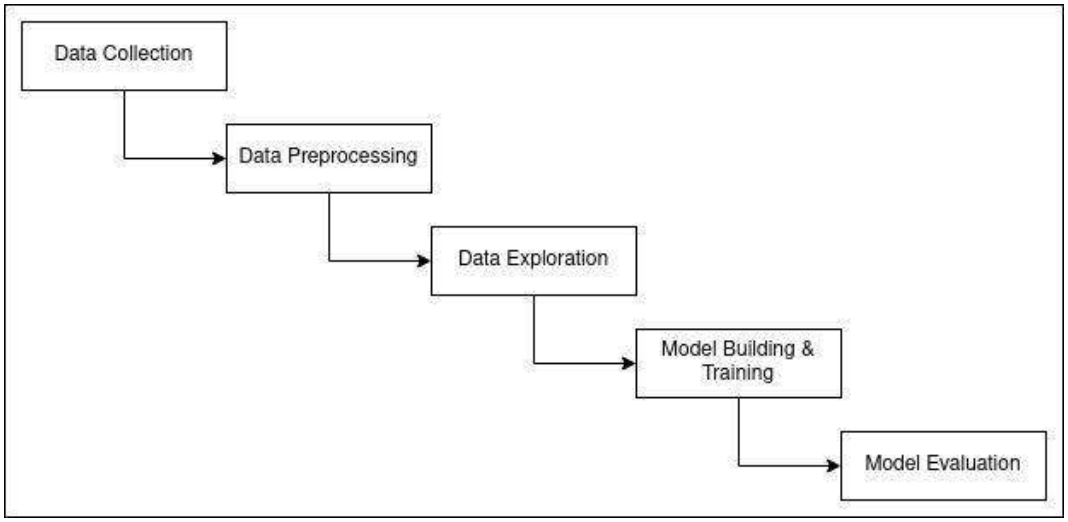
Tabel 1. Karakteristik *phishing website*

Fitur	Sub-Fitur
Fitur berdasarkan *Address Bar*	Menggunakan *IP Address* pada nama domain
	Menggunakan URL yang panjang untuk menutupi bagian yang mencurigakan pada nama domain
	Menggunakan URL *shortening services 'TinyURL'*
	Redirection menggunakan '//'
	Menambahkan Prefiks atau Sufiks yang dipisahkan dengan simbol '-' pada domain
	Terdapat simbol '@' pada nama domain
	Memiliki multi sub-domain
	Menggunakan non-standar *port*
	URL yang terlalu panjang
	Menggunakan *favicon*
	Terdapat token http/https pada nama domain
Fitur berdasarkan abnormalitas	*Request URL*
	URL of Anchor
	Tautan pada tag <Meta>, <Script>, dan <Link>
	Server Form Handler (SFH)
	Input informasi ke Email
	Abnormal URL
HTML dan fitur berdasarkan Javascript	*Website Forwarding*
	Kustomisasi *Status Bar*
	Menonaktifkan *Right Click*
	Menggunakan *pop-up window*
	Iframe Redirection
Fitur berdasarkan domain	Umur dari domain
	DNS Record

 ISSN:2302-8149
e-ISSN:2540-9719

3 Metode Penelitian

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.



Gambar 1. Tahapan membangun model machine learning

Tahapan dari metode penelitian ini dapat dilihat pada Gambar 1, yang dimana dibagi menjadi lima tahap, yaitu:

3.1. *Data Collection*

Pada proses pemilihan data, dataset *phishing website* dan *legitimate website* diambil secara terpisah. Dataset *phishing website* diambil dari situs PhishTank.org, sedangkan dataset *legitimate website* diambil dari situs data.world.

3.1.1. Data *Legitimate Website*

Proses pemilihan/akuisisi data *legitimate website* pada penelitian ini dilakukan dengan cara mengunduh dataset dari alamat *https://data.world/crowdflower/url-categorization/workspace/file?filename=URL-categorization-DFE.csv*. Dataset diterbitkan oleh data.world yang diteliti oleh Figure Eight Inc. (crowdflower) dimana dataset tersebut berisi 31.000 URL dari berbagai negara termasuk Indonesia, dataset tersebut berisi *website* dengan berbagai macam kategori seperti situs berita, hiburan, pendidikan, bisnis & industri, hingga situs pemerintahan. Atribut dari dataset tersebut berisi kategori konten dari URL *website* yang ada, namun atribut-atribut tersebut tidak dipakai pada penelitian ini karena tidak menunjukkan fitur-fitur URL yang dimiliki oleh *website* yang terdapat di dataset. Karena itu untuk dataset legitimate *website* yang didapat dari data.world, atribut yang akan digunakan hanya atribut url.

Dari 31.000 data yang terdapat di url, yang akan digunakan pada penelitian ini hanya 5.000 agar proses data tidak memakan banyak waktu. Untuk pemilihan data yang akan dipilih, digunakan fungsi *Random State*.

3.1.2. Data *Phishing Website*

Untuk pemilihan/akuisisi data phishing *website* dilakukan dengan cara mengunduh dataset dari alamat *http://data.phishtank.com/data/online-valid.csv*. Dataset diterbitkan oleh Organisasi PhishTank dimana dataset tersebut berisi 5.601 URL berbagai *phishing website*. Atribut dari dataset tersebut berjumlah 8 buah, namun atribut-atribut tersebut juga tidak dipakai pada penelitian

 ISSN:2302-8149
e-ISSN:2540-9719

ini karena tidak menunjukkan fitur-fitur URL yang dimiliki oleh *website* yang terdapat di dataset kecuali atribut url. Karena itu untuk dataset *phishing website* yang didapat dari phishtank.org, atribut yang akan digunakan hanya atribut url.

Setelah itu, dilakukan pemilihan 5.000 data yang akan digunakan sebagai data *phishing website* secara acak dengan fungsi Random State.

3.2. *Data Preprocessing*

Pada tahap ini, langkah pertama yaitu URL yang ada pada masing-masing dataset (*legitimate* dan *phishing website*) akan diekstrak fitur-fiturnya menggunakan untuk mengetahui apakah URL tersebut memiliki fitur-fitur yang dimiliki oleh *phishing website*. Karakteristik *website* yang digunakan pada penelitian ini untuk klasifikasi phishing *website* adalah sebagai berikut:

1. Menggunakan *IP Address* pada URL
2. Terdapat simbol '@' pada URL
3. Memiliki URL yang panjang untuk menutupi bagian yang mencurigakan (lebih dari 54 karakter)
4. Jumlah *sub-page* atau *sub-domain* yang mencurigakan yang ditandai dengan simbol '/'
5. Redirection page dengan menggunakan '//'
6. Terdapat http/https pada nama domain untuk mengelabui pengguna
7. Menggunakan URL *shortening services 'TinyURL'*
8. Terdapat prefiks atau sufiks '-' pada nama domain

Setelah melakukan ekstraksi fitur-fitur URL pada masing-masing dataset, maka didapat atribut-atribut baru bagi masing-masing dataset. Setelah dilakukan ekstraksi fitur URL, langkah selanjutnya adalah menggabungkan kedua dataset *phishing* dan *legitimate website* menjadi satu untuk dilakukan proses selanjutnya. Setelah dilakukan penggabungan, maka didapat dataset baru yang memuat dataset *phishing* dan *legitimate website*.

3.3. *Data Exploration*

Data exploration, juga dikenal sebagai eksplorasi data, adalah proses penyelidikan awal terhadap dataset dengan tujuan untuk memahami karakteristiknya, mengidentifikasi pola, mengenali anomali, serta mempersiapkan diri untuk tahapan analisis lebih lanjut. Tujuan dari *data exploration* adalah untuk mendapatkan wawasan awal tentang data sebelum membangun model prediksi. Ini adalah langkah penting dalam pemahaman data dan pengambilan keputusan yang berbasis data.

Beberapa aspek utama dari data exploration meliputi:

1. Statistik Deskriptif: Perhitungan statistik dasar seperti rata-rata, median, modus, kuartil, dan deviasi standar dari variabel-variabel dalam dataset. Informasi ini memberikan gambaran awal tentang nilai-nilai pusat dan sebaran data.
2. Visualisasi Grafis: Membuat grafik dan visualisasi untuk mewakili data dengan cara yang lebih intuitif. Ini meliputi histogram, scatter plot, line chart, box plot, dan lain-lain. Visualisasi membantu mengidentifikasi pola, relasi, dan distribusi data.
3. Penanganan Data Hilang: Mendeteksi dan memutuskan bagaimana data yang hilang (missing data) akan diatasi. Apakah data yang hilang perlu diimputasi atau apakah baris data tersebut perlu dihapus.
4. Analisis Distribusi: Melihat bagaimana data terdistribusi dengan menganalisis bentuk kurva distribusi. Hal ini dapat membantu dalam memahami apakah data mengikuti distribusi normal atau memiliki kecenderungan tertentu.
5. Korelasi dan Hubungan Antar Variabel: Menganalisis korelasi antara variabel-variabel dalam dataset. Ini membantu mengidentifikasi hubungan linier atau non-linier antara variabel.
6. Identifikasi Pola atau Tren: Mengidentifikasi pola, tren, atau perubahan seiring waktu atau kondisi tertentu dalam data.

Data exploration membantu dalam memahami sifat data yang akan diolah, serta membuka peluang untuk pertanyaan-pertanyaan lebih lanjut dan analisis mendalam. Ini adalah langkah awal yang penting sebelum melakukan pembangunan model machine learning.

3.4. *Model Building & Training*

 ISSN:2302-8149
e-ISSN:2540-9719

Model building & training adalah tahap kunci dalam proses pengembangan aplikasi atau solusi berbasis *machine learning*. Pada tahap ini, langkah yang dilakukan adalah merancang, memilih, dan melatih model *machine learning* untuk memahami pola-pola dalam data dan membuat prediksi atau pengambilan keputusan. Berikut adalah penjelasan lebih detail tentang tahap *model building & training*:

1. Pemilihan Algoritma atau Model:
   Berdasarkan studi literatur yang telah dilakukan sebelumnya, akan dipilih algoritma atau model *machine learning* yang paling sesuai.
2. Pemilihan Fitur (*Feature Selection*):
   Identifikasi fitur-fitur yang paling relevan dan berpengaruh dalam memprediksi atau mengklasifikasikan data. Pemilihan fitur membantu mengurangi dimensi data dan meningkatkan efisiensi model.
3. Inisialisasi Model:
   Inisialisasi model dengan parameter awal yang sesuai untuk masing-masing model.
4. Pelatihan Model:
   Gunakan data pelatihan untuk melatih model. Ini melibatkan menyajikan data ke model, menghitung prediksi, mengukur kesalahan (error), dan mengoptimalkan parameter model untuk mengurangi kesalahan.

3.5. *Model Evaluation*

Untuk evaluasi dan perbandingan dari hasil algoritma-algoritma tersebut dilakukan perhitungan nilai akurasi, presisi, *F1 score*, dan *recall* dari masing-masing algoritma. Perhitungan nilai-nilai tersebut dapat dilakukan dengan menggunakan hasil dari *confusion matrix*. Tabel 2 dibawah ini menunjukan tabel *confusion matrix. Confusion matrix* sendiri adalah ringkasan hasil prediksi pada kasus klasifikasi. *Confusion matrix* digunakan untuk mengevaluasi performa pengklasifikasi pada dataset [12]. 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) [13].

Tabel 2. Tabel *confusion matrix phishing website*

		Predicted	
		Legitimate	Phishing
Actual	Legitimate	True Negative	False Positive
	Phishing	False Negative	True Positive

1. Akurasi
   Akurasi adalah nilai rasio prediksi yang benar (positif dan negatif) dengan keseluruhan data [14]. Akurasi akan menjawab pertanyaan "Berapa persen *website* yang benar diprediksi sebagai *phishing website* dan *legitimate website*".

2. *Recall (Sensitivity)*
   Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif [14]. Recall menjawab pertanyaan "Berapa persen *website* yang diprediksi sebagai *phishing* dibandingkan keseluruhan *website* yang sebenarnya *phishing website*".

3. Presisi
   Presisi atau *Precision* adalah rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif [14]. Presisi menjawab pertanyaan "Berapa persen *website* yang benar *phishing website* dari keseluruhan *website* yang diprediksi sebagai *phishing website*?".

 ISSN:2302-8149
e-ISSN:2540-9719

4 Hasil dan Pembahasan

4.1. *Data Collection*

Langkah pertama yang dilakukan adalah mengunduh data URL dari situs yang telah disebutkan sebelumnya. Untuk data URL *phishing website* diunduh dari phishtank.org dan data *legitimate website* diunduh dari situs data.world dimana perintah tersebut ditampilkan pada Gambar 2 dan Gambar 3.

```
#Downloading the phishing URLs file
!wget http://data.phishtank.com/data/online-valid.csv
```

Gambar 2. Proses download data URL phishing

```
#Loading legitimate files
data1 = pd.read_csv('https://query.data.world/s/iyzsdocjzfhtvgfe4yicf2lbanhpwq')
data1.drop(data1.columns.difference(['url']), 1, inplace=True)
data1.head()
```

Gambar 3. Proses download data URL ;egitimate

Setelah diunduh, dataset yang berisi banyak url tersebut masing-masing hanya akan diambil 5,000 URL untuk *phishing website* dan *legitimate website* sehingga total keseluruhan terdapat 10,000 URL. Untuk mendapatkan 5,000 data URL dari dataset yang telah diunduh, digunakan fungsi *random state* untuk mendapatkan data URL secara acak. Pada Gambar 4 digunakan fungsi *random state* untuk dataset *phishing* URL dan pada Gambar 5 digunakan *random state* pada dataset *legitimate* URL.

```
#Collecting 5,000 Phishing URLs randomly
phishurl = data0.sample(n = 5000, random_state = 12).copy()
phishurl = phishurl.reset_index(drop=True)
phishurl.head()
```

Gambar 4. Fungsi random state pada phishing URL

```
#Collecting 5,000 Legitimate URLs randomly
legiurl = data1.sample(n = 5000, random_state = 12).copy()
legiurl = legiurl.reset_index(drop=True)
legiurl.head()
```

Gambar 5. Fungsi random state pada legitimate URL

4.2. *Data Preprocessing*

Pada tahap *preprocessing* dilakukan ekstraksi fitur-fitur URL dari dataset yang telah dibuat sebelumnya. Fitur-fitur yang diekstraksi antara lain: menggunakan *IP Address* pada URL, terdapat simbol '@' pada URL, memiliki URL yang panjang untuk menutupi bagian yang mencurigakan (lebih dari 54 karakter), jumlah *sub-page* atau *sub-domain* yang mencurigakan yang ditandai dengan simbol '/', *redirection page* dengan menggunakan '//', terdapat http/https pada nama domain untuk mengelabui pengguna, menggunakan URL *shortening services 'TinyURL'*, dan terdapat prefiks atau sufiks '-' pada nama domain.

Setelah dilakukan ekstraksi pada URL, pada Gambar 6 data yang didapat diubah kedalam bentuk biner dan disimpan dalam bentuk list. Setelah disimpan dalam bentuk list, pada Gambar 7 data URL tersebut diubah kedalam bentuk *dataframe* agar dapat digunakan dalam model.

 ISSN:2302-8149
e-ISSN:2540-9719

```
#Extracting the feautres & storing them in a list
legi_features = []
label = 0

for i in range(0, 5000):
  url = legiurl['url'][i]
  legi_features.append(featureExtraction(url,label))
```

Gambar 6. Ekstraksi fitur URL dan menyimpan dalam *List*

```
#converting the list to dataframe
feature_names = ['Domain', 'Have_IP', 'Have_At', 'URL_Length', 'URL_Depth','Redirection',
                 'https_Domain', 'TinyURL', 'Prefix/Suffix', 'Label']

legitimate = pd.DataFrame(legi_features, columns= feature_names)
legitimate.head()
```

Gambar 7. Mengubah list ke dalam bentuk *dataframe*

Langkah selanjutnya, setelah diubah kedalam bentuk *dataframe*, kedua dataset *phishing* dan *legitimate* URL digabungkan menjadi satu dataset untuk digunakan pada langkah selanjutnya. Langkah penggabungan tersebut dapat dilihat pada Gambar 8

```
#Concatenating the dataframes into one
urldata = pd.concat([legitimate, phishing]).reset_index(drop=True)
urldata.head()
```

	Domain	Have_IP	Have_At	URL_Length	URL_Depth	Redirection	https_Domain	TinyURL	Prefix/Suffix	Label
0	tanisuzivo.com	0	0	0	1	0	0	0	0	0
1	cafesydney.com	0	0	0	1	0	0	0	0	0
2	nursesanywhere.com	0	0	0	1	0	0	0	0	0
3	dragees.fr	0	0	0	1	0	0	0	0	0
4	tulliehouse.co.uk	0	0	0	1	0	0	0	0	0

Gambar 8. Menggabungkan dataset *phishing* dan *legitimate*

4.3. *Data Exploration*

Pada tahap selanjutnya, dilakukan eksplorasi data untuk mengetahui karakteristik dataset yang telah dibuat. Langkah pertama adalah untuk mengetahui nama-nama kolom yang terdapat di dataset dan tipe data dari masing-masing kolom. Dari Gambar 9, dapat dilihat bahwa semua kolom kecuali kolom domain memiliki tipe data integer dikarenakan semua value berbentuk biner.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Domain         10000 non-null  object
 1   Have_IP        10000 non-null  int64
 2   Have_At        10000 non-null  int64
 3   URL_Length     10000 non-null  int64
 4   URL_Depth      10000 non-null  int64
 5   Redirection    10000 non-null  int64
 6   https_Domain   10000 non-null  int64
 7   TinyURL        10000 non-null  int64
 8   Prefix/Suffix  10000 non-null  int64
 9   Label          10000 non-null  int64
dtypes: int64(9), object(1)
memory usage: 781.4+ KB
```

Gambar 9. Tipe Data kolom-kolom dataset

Langkah selanjutnya adalah mengecek apakah terdapat *null values* di dataset yang akan digunakan. Dari Gambar 10, dapat dilihat bahwa dataset yang digunakan tidak terdapat *null values* atau semua baris dan kolom memiliki *value*.

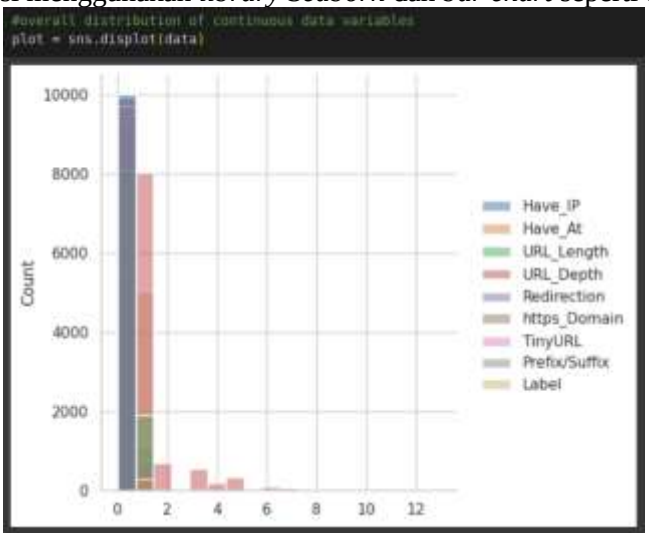
 ISSN:2302-8149
e-ISSN:2540-9719

```
#Counting and checking for null values
data.isnull().sum()

Have_IP          0
Have_At          0
URL_Length       0
URL_Depth        0
Redirection      0
https_Domain     0
TinyURL          0
Prefix/Suffix    0
Label            0
dtype: int64
```

Gambar 10. Cek *null values*

Pada langkah terakhir di eksplorasi data, dilakukan visualisasi dari distribusi kolom yang ada di dataset. Visualisasi menggunakan *library Seaborn* dan *bar chart* seperti di Gambar 11.



Gambar 11. Visualisasi distribusi kolom

4.4. *Model Building & Training*

Tahap berikutnya adalah tahap membangun model dan melatihnya menggunakan dataset yang telah disiapkan sebelumnya. Langkah pertama yang dilakukan adalah memisahkan kolom fitur dan target dan menyimpannya di variabel y dan X. Di Gambar 12, kolom target disimpan dalam variabel y sedangkan variabel x menyimpan kolom fitur dari dataset.

```
# Sepratating & assigning features and target columns to X & y
y = data['Label']
X = data.drop('Label',axis=1)
```

Gambar 12. Memisahkan kolom fitur dan target

Langkah berikutnya yang ditampilkan pada Gambar 13 adalah membagi dataset menjadi *train set* dan *test set*. Berdasarkan penelitian Muraina et al [15], untuk dataset dengan ukuran 100 sampai dengan 1.000.000 pembagian rasio *training* dan *test* set yang paling sesuai adalah adalah 80:20. Maka dari itu pada penelitian ini dataset yang digunakan dibagi dengan persentase 80:20, dimana 80% untuk *train set* dan 20% untuk *test set* serta menggunakan *random state* sebesar 12.

```
# Splitting the dataset into train and test sets: 80-20 split

X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.2, random_state = 12)
```

Gambar 13. Membagi *train* dan *test set*

Setelah dataset dibagi menjadi *train* dan *test set*, berikutnya dibangun model untuk masing-masing algoritma. Model pertama menggunakan algoritma *Decision Tree* dengan *max depth* dari

 ISSN:2302-8149
e-ISSN:2540-9719

pohon yang digunakan adalah 5 pohon. Kode yang digunakan dapat dilihat pada Gambar 14 di bawah.

```
from sklearn.tree import DecisionTreeClassifier

# instantiate the model
tree = DecisionTreeClassifier(max_depth = 5)
# fit the model
tree.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_tree = tree.predict(X_test)
y_train_tree = tree.predict(X_train)

tree_score=tree.score(X_test, y_test)
```

Gambar 14. Membangun model decision tree

Model berikutnya menggunakan algoritma *Random Forest* dimana sama seperti model *Decision Tree* yang menggunakan *max depth* sebesar 5, model *Random Forest* juga menggunakan nilai *max depth* yang sama, sedangkan untuk jumlah pohon yang digunakan dalam *Random Forest* adalah sebesar 100 pohon. Hal tersebut digambarkan pada Gambar 15 di bawah ini.

```
# instantiate the model
forest = RandomForestClassifier(max_depth=5, n_estimators=100, random_state=0)
# fit the model
forest.fit(X_train, y_train)
#predicting the target value from the model for the samples
y_test_forest = forest.predict(X_test)
y_train_forest = forest.predict(X_train)
model_score=forest.score(X_test, y_test)
```

Gambar 15. Membangun model random forest

Model terakhir yang dibangun adalah model pada Gambar 16 yang menggunakan algoritma KNN, dimana model ini menggunakan nilai *neighbors* sebesar 5. Setelah semua model dibangun dan dilatih, selanjutnya masing-masing performa model akan dibandingkan.

```
# instantiate the model
knn = KNeighborsClassifier(n_neighbors =5)
# fit the model
knn.fit(X_train,np.ravel(y_train,order='C'))

#predicting the target value from the model for the samples
y_test_knn = knn.predict(X_test)
y_train_knn = knn.predict(X_train)
model_score=knn.score(X_test, y_test)
```

Gambar 16 Membangun model KNN

4.5. *Model Evaluation*

Tahap terakhir adalah evaluasi dari performa masing-masing model. Untuk model pertama yang menggunakan *Decision Tree* didapat akurasi sebesar 0.833, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83, hasil tersebut dapat dilihat pada Gambar 17.

 ISSN:2302-8149
e-ISSN:2540-9719

```
Decision Tree: Accuracy on the Model:  0.833
Decision Tree: Accuracy on training Data: 0.822
Decision Tree: Accuracy on test Data: 0.833
              precision    recall  f1-score   support

           0       0.77      0.96      0.86      1036
           1       0.95      0.69      0.80       964

    accuracy                           0.83      2000
   macro avg       0.86      0.83      0.83      2000
weighted avg       0.86      0.83      0.83      2000
```

Gambar 17. Performa model *decision tree*

Sedangkan model kedua yang menggunakan algoritma *Random Forest* mendapat akurasi sebesar 0.834, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83. Hasil tersebut dapat dilihat pada Gambar 18 di bawah ini.

```
Random forest: Accuracy on the Model:  0.8335
Random forest: Accuracy on training Data: 0.823
Random forest: Accuracy on test Data: 0.834
              precision    recall  f1-score   support

           0       0.77      0.96      0.86      1036
           1       0.95      0.69      0.80       964

    accuracy                           0.83      2000
   macro avg       0.86      0.83      0.83      2000
weighted avg       0.86      0.83      0.83      2000
```

Gambar 18. Performa model *random forest*

Gambar 19 memperlihatkan model terakhir yang menggunakan algoritma *K-Nearest Neighbors* mendapat akurasi sebesar 0.482, presisi sebesar 0.24, *recall* sebesar 0.50, dan *F1-score* sebesar 0.48.

```
KNeighborsClassifier: Accuracy on the Model:  0.482
KNeighborsClassifier: Accuracy on training Data: 0.504
KNeighborsClassifier: Accuracy on test Data: 0.482
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1036
           1       0.48      1.00      0.65       964

    accuracy                           0.48      2000
   macro avg       0.24      0.50      0.33      2000
weighted avg       0.23      0.48      0.31      2000
```

Gambar 19. Performa model KNN

Setelah semua performa dari model didapat, langkah selanjutnya adalah membandingkan performa ketiga model tersebut. Dari data performa di Gambar 20, didapat model dengan nilai performa tertinggi adalah model yang menggunakan algoritma *Random Forest*. Untuk itu, model dengan algoritma *Random Forest* yang akan digunakan dalam implementasi aplikasi *browser extension* untuk mendeteksi *phishing website*.

	ML Model	Train Accuracy	Test Accuracy
1	Random forest	0.822	0.834
0	Decision Tree	0.822	0.833
2	KNeighborsClassifier	0.504	0.482

Gambar 20. Perbandingan akurasi model

 ISSN:2302-8149
e-ISSN:2540-9719

5 Kesimpulan

Penelitian ini berfokus pada tindak kriminal siber yaitu *phishing website*. Pada penelitian ini dilakukan perbandingan performa dari tiga algoritma klasifikasi untuk mencari algoritma terbaik untuk mendeteksi *phishing website*. Klasifikasi untuk menentukan *website* termasuk *phishing* atau tidak menggunakan fitur-fitur URL *website*. Hasil dari penelitian ini adalah model pertama yang menggunakan *Decision Tree* didapat akurasi sebesar 0.833, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83. Model kedua yang menggunakan algoritma *Random Forest* mendapat akurasi sebesar 0.834, presisi sebesar 0.86, *recall* sebesar 0.83, dan *F1-score* sebesar 0.83. Model terakhir yang menggunakan algoritma *K-Nearest Neighbors* mendapat akurasi sebesar 0.482, presisi sebesar 0.24, *recall* sebesar 0.50, dan *F1-score* sebesar 0.48. Maka, dari ketiga algoritma tersebut *Random Forest* merupakan algoritma terbaik untuk mendeteksi *phishing website*.

Referensi

[1] APJII, "APJII (*Indonesia Association Internet Services Organizer*) report on 2019-2020 [Q2]", Jul. 2020. https://www.infotek.id/licenses/survey_apjii_2020/Survei_APJII_2019-2020_Q2.pdf [Diakses pada 9 Juli 2023].

[2] APWG, "APWG (*Anti-Phishing Working Group) Phishing Activity Trends Report*", 1st Quarter 2021. https://docs.apwg.org/reports/apwg_trends_report_q1_2021.pdf [Diakses pada 9 Juli 2023].

[3] R. Putra Ramadhan and T. Desyani, "Implementasi Algoritma J48 untuk Identifikasi Website Phishing", *Biner*, vol. 1, no. 2, pp. 46–54, Jun. 2023.

[4] Grand View Research, *"Cyber Security Market Size and Share | Industry Report, 2019-2025,"* *Grandviewresearch.com*, 2019. https://www.grandviewresearch.com/industry-analysis/cyber-security-market [Diakses pada 10 Juli 2023].

[5] S. Badillo *et al.*, "*An Introduction to Machine Learning,*" *Clinical Pharmacology & Therapeutics*, vol. 107, no. 4, pp. 871–885, Mar. 2020, doi: https://doi.org/10.1002/cpt.1796.

[6] A. K. Jain and B. B. Gupta, "*A machine learning based approach for phishing detection using hyperlinks information," Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 2015–2028, Apr. 2018, doi: https://doi.org/10.1007/s12652-018-0798-z.

[7] M. Korkmaz, O. K. Sahingoz, and B. Diri, "*Detection of Phishing Websites by Using Machine Learning-Based URL Analysis," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Jul. 2020, doi: https://doi.org/10.1109/icccnt49239.2020.9225561*.

[8] M. Abutaha, M. Ababneh, K. A. Mahmoud, and Sherenaz W. Al-Haj Baddar, "URL *Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis," 2021 12th International Conference on Information and Communication Systems (ICICS)*, May 2021, doi: https://doi.org/10.1109/icics52457.2021.9464539.

[9] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "*Machine learning based phishing detection from URLs," Expert Systems with Applications*, vol. 117, pp. 345–357, Mar. 2019, doi: https://doi.org/10.1016/j.eswa.2018.09.029.

[10] A. Safi and S. Singh, "*A systematic literature review on phishing website detection techniques," Journal of King Saud University - Computer and Information Sciences, Jan. 2023, doi: https://doi.org/10.1016/j.jksuci.2023.01.004*.

[11] S. Shabudin, N. Samsiah, K. Akram, and M. Aliff, "*Feature Selection for Phishing Website Classification," International Journal of Advanced Computer Science and Applications, vol. 11, no. 4, 2020, doi: https://doi.org/10.14569/ijacsa.2020.0110477*.

[12] M. Hasnain, M. F. Pasha, I. Ghani, M. Imran, M. Y. Alzahrani, and R. Budiarto,

"*Evaluating Trust Prediction and Confusion Matrix Measures for Web Services Ranking," IEEE Access, vol. 8, pp. 90847–90861, 2020, doi: https://doi.org/10.1109/access.2020.2994222.*

[13] *F. Rahmad, Y. Suryanto, and K. Ramli, "Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification,*" *IOP Conference Series: Materials Science and Engineering*, vol. 879, p. 012076, Aug. 2020, doi: https://doi.org/10.1088/1757-899x/879/1/012076.

[14] J. A. Mat Jizat, A. P.P. Abdul Majeed, A. F. Ab. Nasir, Z. Taha, and E. Yuen, "*Evaluation of the machine learning classifier in wafer defects classification,*" *ICT Express*, May 2021, doi: https://doi.org/10.1016/j.icte.2021.04.007.

[15] I. Muraina, "*Ideal dataset splitting ratios in machine learning algorithms: general concerns for data scientists and data analysts*", *7th International Mardin Artuklu Scientific Research Conference*, pp. 496-504, Feb. 2022.