DETEKSI HALAMAN WEBSITE PHISHING MENGGUNAKAN ALGORITMA MACHINE LEARNING GRADIENT BOOSTING CLASSIFIER

Muhammad Rizal Supriadi[1]

Teknik Informatika, Program Studi Teknik Informatika
Universitas Logistik dan Bisnis Internasional
Jl. Sari Asih No.54, Sarijadi, Kec. Sukasari, Kota Bandung, Jawa Barat 40151
Email: [1]muhammadrizalsupriadi@gmail.com

ABSTRAK

Phising merupakan suatu tindakan untuk mendapatkan informasi penting milik seseorang berupa *username, password* dan informasi penting lainnya dengan menyediakan situs web palsu yang mirip dengan aslinya. *Phising* adalah bentuk tindakan pidana yang bermaksud untuk mendapatkan informasi rahasia dari seseorang, seperti nama pengguna, sandi dengan menyamar sebagai orang atau bisnis terpercaya. Seiring dengan perkembangannya penggunakan media elektronik yang diikuti dengan meningkatnya kejahatan dunia maya, seperti serangan *phising*.

Oleh karena itu, untuk meminimalkan serangan *phising*, diperlukan sistem yang dapat mendeteksi serangan tersebut. *Machine Learning* merupakan salah satu metode yang dapat digunakan untuk membuat sistem yang dapat mendeteksi *phising*. Data yang digunakan dalam penelitian ini adalah 11054 data *website* yang terbagi menjadi dua bagian yaitu "sah" dan "*phising*". Sedangkan algoritma yang digunakan adalah *Gradient Boosting Classifier* Dari hasil pengujian pada penelitian ini diperoleh akurasi sistem terbaik 97,4% dengan nilai f1_score 0,977 serta untuk aplikasi web menggunakan framework Flask bahasa pemrograman python.

ABSTRACT

Phishing is an act of obtaining important information belonging to someone in the form of usernames, passwords and other important information by providing a fake website that is similar to the original. Phishing is a form of criminal action that intends to obtain confidential information from someone, such as usernames, passwords by posing as a trusted person or business. Along with the development of the use of electronic media followed by an increase in cyber crimes, such as phishing attacks.

Therefore, to minimize phishing attacks, we need a system that can detect these attacks. Machine Learning is one method that can be used to create a system that can detect phishing. The data used in this study is 11054 website data which is divided into two parts, namely "legitimate" and "phishing". While the algorithm used is the Gradient Boosting Classifier. From the test results in this study, the best system accuracy was 97.4% with an f1_score value of 0.977 and for web applications using the Python programming language Flask framework.

Keywords: Phishing, Machine Learning, Websites, Framework, Flask

I. PENDAHULUAN

Dengan adanya kemajuan teknologi, hal ini dapat membantu masyarakat dalam menunjang aktivitas, khususnya dalam pemanfaatan internet, masyarakat dapat lebih mudah dan efektif dalam berkomunikasi maupun mencari sistem informasi Dengan meningkatnya pengguna internet dan perkembangan teknologi, ancaman keamanan semakin beragam. Salah satunya adalah *phising*.

Phising merupakan suatu kegiatan yang bersifat mengancam dan menjebak seseorang dengan cara memancing target untuk secara tidak langsung memberikan informasi kepada penjebak Selain itu *phising* bertujuan untuk mengirimkan tautan berbahaya, biasanya menyamar sebagai yang legal, melalui spam ataupun jejaring media sosial untuk mendorong pengguna mengunjungi dan memperoleh informasi pribadi mereka. Hal ini sejalan dengan pendapat bahwa *phising* merupakan skema *cyber* berdasarkan kegiatan kriminal yang menarik perhatian.

Banyak kejahatan *phising* yang terjadi, hal tersebut berpotensi menimpulkan beberapa kerugian baik kerugian *privacy* seseorang maupun lembaga atau perusahaan. APWG (*Anti- Phising Working Group*) mengemukakan bahwa dari tahun ke tahun, masyarakat di Indonesia semakin sadar tentang adanya website *Phising.* Maka salah satu upaya untuk meminimalisir dengan cara identifikasi untuk mendeteksi website yang terindikasi *phising*, dengan hal ini dibutuhkan klasifikasi dalam data mining supaya dapat mengetahui data maupun parameter yang dijadikan acuan dalam deteksi website phising

II. ALGORITMA MACHINE LEARNING

Pembelajaran mesin adalah aplikasi kecerdasan buatan (AI) yang memberikan sistem kemampuan untuk belajar dan belajar secara otomatis meningkatkan dari pengalaman tanpa diprogram secara eksplisit. Ini berfokus pada pengembangan program komputer yang bisa mengakses data dan menggunakannya untuk belajar sendiri.

Algoritma pembelajaran mesin sering dikategorikan sebagai diawasi atau tidak diawasi. Algoritma yang diawasi membutuhkan ilmuwan data atau analis data dengan keterampilan pembelajaran mesin untuk memberikan input dan yang diinginkan output, selain memberikan umpan balik tentang keakuratan prediksi selama pelatihan algoritma [2].

Algoritma pembelajaran mesin *Supervised Learning* dapat menerapkan apa yang telah dipelajari di masa lalu ke data baru menggunakan contoh berlabel memprediksi peristiwa masa depan. Dimulai dari analisis dataset pelatihan yang diketahui, algoritma pembelajaran menghasilkan kesimpulan berfungsi untuk membuat prediksi tentang nilai output. Sistem mampu memberikan target untuk setiap masukan baru setelah cukup pelatihan. Algoritme pembelajaran juga dapat membandingkan keluarannya dengan keluaran yang benar dan diinginkan serta menemukan kesalahan untuk dimodifikasi sesuai dengan modelnya.[3]

1. Logistic Regression

Regresi logistik adalah model prediktif yang digunakan untuk mengevaluasi hubungan antara variabel dependen (target) yang merupakan data kategorikal dengan skala nominal atau ordinal dan variabel independen (prediktor) yang merupakan data kategorikal dengan skala interval atau rasio. Algoritma ini juga dapat digunakan untuk pemodelan deret waktu untuk menemukan hubungan antar

variabel yang terlibat.[4]

2. K-Nearest Neighbors

K-Nearest Neighbor adalah metode klasifikasi dengan mencari jarak terdekat antara data yang akan dievaluasi dengan *K-Nearest Neighbors* terdekatnya dalam data pelatihan. Model ini dapat digunakan dalam klasifikasi yang akan dilakukan data training didalam proses pelatihan tersebut.[5]

3. Support Vector Machine

SVM adalah teknik pembelajaran mesin berdasarkan Supervised Learning dan sesuai untuk kedua regresi dan klasifikasi. SVM dianggap sebagai pencapaian teknik modern penerimaan cepat karena hasil yang baik dicapai dalam banyak bidang masalah data mining, berdasarkan fondasi yang kuat dalam teori belajar statistik. SVM adalah teknik klasifikasi berdasarkan pembelajaran statistik, yang berhasil dimanfaatkan dalam banyak aplikasi klasifikasi nonlinier dan besar dataset dan masalah. [6].

4. Naïve Bayes

Pengklasifikasi Naïve Bayes adalah salah satu deteksi tinggi pendekatan untuk mempelajari klasifikasi dokumen teks. Diberikan satu set sampel pelatihan rahasia, sebuah aplikasi dapat belajar dari sampel tersebut, sehingga dapat memprediksi kelas sampel yang tidak terpenuhi. [7].

5. Decision Tree

Pohon keputusan (DT). DT mengklasifikasikan barang berdasarkan pembuatan keputusan pada setiap cabang untuk mendapatkan sebanyak-banyaknya keuntungan entropi sebanyak mungkin. Sebuah pohon keputusan terdiri dari a simpul akar, beberapa simpul internal, dan simpul daun. Daun node menunjukkan hasil dari classifier, dan lainnya node menunjukkan setiap atribut. Setiap rute dari simpul akar ke simpul daun sesuai dengan penentuan urutan pengujian. Ini mengikuti aturan dari memecah dan menaklukkan [8].

6. Random Forest

Algoritma *Random Forest* dapat mencapai akurasi tertinggi sebelum dan sesudah pemilihan fitur dan peningkatan bangunan secara dramatis. Hasil percobaan menunjukkan bahwa menggunakan pendekatan seleksi dengan mesin algoritma pembelajaran dapat meningkatkan efektivitas model klasifikasi untuk deteksi phishing tanpa mengurangi kinerja mereka. [9].

7. Gradient Boosting

Tujuan utama dari Boosting adalah menggabungkan semua train yang lemah bersama-sama untuk membentuk model yang kuat.

- Gradient boosting adalah suatu teknik yang sangat kuat untuk mengembangkan model prediktif. Ini berlaku untuk beberapa fungsi risiko dan mengoptimalkan akurasi prediksi model. Ini juga menyelesaikan masalah multikolinearitas di mana korelasi antar variabel prediktor tinggi.
- Gradient Boosting adalah algoritma pembelajaran mesin ansambel dan biasanya digunakan untuk menyelesaikan klasifikasi dan regresi [10].

8. CatBoost

Algoritma CatBoost membutuhkan lebih banyak waktu untuk pelatihan dan menguji kumpulan data yang menggunakan lebih banyak komputasi sumber daya. Efektivitas algoritma CatBoost memiliki telah ditunjukkan melalui kinerjanya yang lebih tinggi algoritma yang bersaing. Dalam hal pekerjaan masa depan, Apache Kerangka kerja Spark dapat digunakan untuk meningkatkan *Sickit-learn library* yang disebut Sk-dist. Sk-dist telah mengatasi batasan tersebut perpustakaan Sickit-belajar seperti memakan waktu dan lagging pelatihan model [11].

9. Python

Python adalah bahasa yang dirancang dengan baik yang dapat digunakan secara nyata pemrograman dunia. Python adalah tingkat yang sangat tinggi, dinamis, berorientasi objek, bahasa pemrograman tujuan umum yang menggunakan juru bahasa dan dapat digunakan dalam domain yang luas aplikasi [12].

III. TOOLS YANG DIGUNAKAN

1. Framework Flask

Gambar III.1 Framework Flask

Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework . Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur behaviour suatu web dengan lebih mudah.[13]

2. Jupyter Notebook

Gambar III.2 Jupyter Notebook

Jupyter Notebook (file yang berekstensi ipynb) adalah dokumen yang dihasilkan oleh Jupyter Notebook App yang berisikan kode komputer dan rich text element seperti paragraf, persamaan matematik, gambar dan tautan (links) [14].

3. Visual Studio Code

Gambar III.3 VS Code

VS Code adalah sebuah teks editor yang sangat populer dan banyak digunakan oleh para pengembang software di seluruh dunia. Dibuat oleh Microsoft, VS Code merupakan pilihan tepat bagi para pengguna sistem operasi multiplatform, seperti Linux, Mac, dan Windows. Tidak hanya itu, editor ini juga memiliki berbagai fitur yang handal dan efisien, serta dukungan langsung untuk bahasa pemrograman populer seperti JavaScript, Typescript, dan Node.js. [15]

4. Pickle

Modul pickle mengimplementasikan protokol biner untuk serialisasi dan de-serialisasi

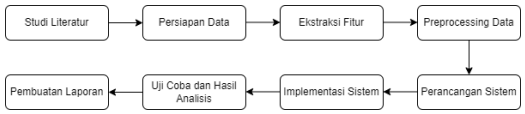
struktur objek Python. "Pickling" adalah proses di mana hierarki objek Python diubah menjadi aliran byte, dan "unpickling" adalah operasi kebalikannya, di mana aliran byte (dari file biner atau objek mirip byte) diubah kembali menjadi hierarki objek. [16]

5. Web Browser

Web browser disebut juga sebagai perambah, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web. Browser pada umumnya juga mendukung berbagai jenis URL dan protokol, misalnya ftp: untuk file transfer protocol (FTP), rtsp: untuk real-time streaming protocol (RTSP), and https: untuk versi http yang terenkripsi (SSL). [17]

IV. METODOLOGI PENELITIAN

Tahapan yang akan dilakukan dalam penelitian ini dapat dilihat pada gambar berikut:



Gambar IV.1 Metodologi Penelitian

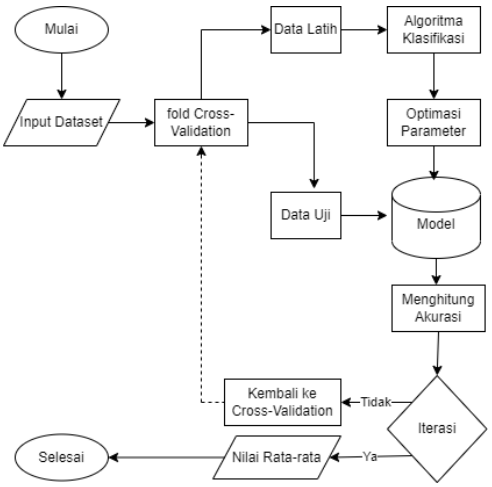
Berdasarkan diagram pada diatas, secara umum penelitian dapat digambarkan sebagai berikut:

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahapan ini dilakukan untuk mengumpulkan penelitian yang berkaitan dengan metode yang digunakan kepada ektraksi fitur dan klasifikasi.
2. Persiapan data merupakan langkah yang dilakukan untuk mendapatkan dataset yang akan digunakan dalam mengklasifikasikan website. Dataset yang digunakan bersumber dari Kaggle.
3. Ektraksi fitur dilakukan untuk mengekstraksi fiur terdapat di situs web berdasarkan dataset yang diperoleh dari Kaggle. Hasil ektraksi fitur ini kemudian akan digunakan untuk mendeteksi situs web phising.
4. Preprocessing Data dilakukan dalam bentuk menganalisis data yang akan digunakan untuk memilih data berdasarkan hasil ekstraksi ciri yang telah dilakukan sebelumnya.
5. Perancangan sistem pada penelitian ini dilakukan untuk menentukan algoritma yang akan dinguakan dalam klasifikasi situs web phising. Algoritma-algoritma yang digunakan dalam mengklasifikasi adalah Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting dan Catboost sebagi perbandingan kinerja sistem. Pada tahapan ini dibuat flowchat yang berkaitan dengan alur kerja sistem.
6. Implementasi sistem dilakukan sesuai dengan flowchart yang telah dibuat ditahap sebelumnya. Pada penelitian ini, sistem dibuat dengan menggunakan bahasa pemrograman python dan framework flask.
7. Pengujian sistem dilakukan untuk mengetahui keakuratan sistem yang dibuat, sistem akan diuji dengan beberapa percobaan dan bentuk URL yang berbeda.

Tahapan akhir dalam penelitian ini adalah menulis laporan penelitian dalam bentuk laporan.

1. Sistem Evaluasi

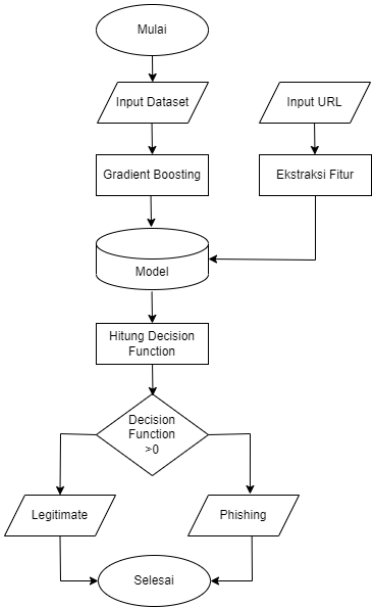
Sistem evaluasi merupakan sistem yang dirancang sebagai untuk evaluasi sistem produksi. Evaluasi terhadap sistem ini dilakukan dengan cara mengevaluasi dataset yang digunakan dalam implementasi sistem. Flowchart sistem evalusi dapat dilihat pada gambar berikut:



Gambar IV.2 Sistem Evaluasi

2. Sistem Implementasi

Sistem implementasi merupakan sistem yang dirancang untuk digunakan dan diimplementasikan dalam melakukan deteksi website phising. Hal ini yang membedakan sistem implementasi dengan sistem evaluasi adalah sistem ini dirancang untuk digunakan oleh pengguna dengan input berupa URL. Selain ini perbedaannya terdapat pada proses ekstraksi yang sangat penting dalam menentukan kinerja sistem. Untuk lebih jelasnya dapat dilihat pada flowchart sistem berikut:



Gambar IV.3 Sistem Evaluasi

V. HASIL DAN PEMBAHASAN

Pada penelitian ini melalukan perbandingan model dari ke 9 model yang dilakukan train dan test dan didapat hasil seperti berikut:

1. Perbandingan Model

```
#creating dataframe
result = pd.DataFrame({ 'ML Model' : ML_Model,
                        'Accuracy' : accuracy,
                        'f1_score' : f1_score,
                        'Recall'   : recall,
                        'Precision': precision,
                      })

# dispalying total result
result
```

	ML Model	Accuracy	f1_score	Recall	Precision
0	Logistic Regression	0.934	0.941	0.943	0.927
1	K-Nearest Neighbors	0.956	0.961	0.991	0.989
2	Support Vector Machine	0.964	0.968	0.980	0.965
3	Naive Bayes Classifier	0.605	0.454	0.292	0.997
4	Decision Tree	0.957	0.962	0.991	0.993
5	Random Forest	0.965	0.969	0.993	0.990
6	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
7	CatBoost Classifier	0.972	0.975	0.994	0.989
8	Multi-layer Perceptron	0.968	0.972	0.996	0.977

Gambar V.1 Sistem Evaluasi

Pada *script* di atas, dibuat sebuah *object result* yang merupakan sebuah *DataFrame* dengan menggunakan *library pandas*. *DataFrame* dibuat dengan menggunakan argumen yang berisi *dictionary* dengan *key* sebagai nama kolom dan *value* sebagai isi dari kolom tersebut.

Selanjut kita akan melakukan *sorting* dari model-model diatas berdasarkan hasil accuracy yang paling tinggi dan didapat hasil seperti berikut.

```
#Sorting the dataframe on accuracy
sorted_result=result.sort_values(by=['Accuracy', 'f1_score'],ascending=False).reset_index(drop=True)

# dispalying total result
sorted_result
```

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Multi-layer Perceptron	0.968	0.972	0.996	0.977
3	Random Forest	0.965	0.969	0.993	0.990
4	Support Vector Machine	0.964	0.968	0.980	0.965
5	Decision Tree	0.957	0.962	0.991	0.993
6	K-Nearest Neighbors	0.956	0.961	0.991	0.989
7	Logistic Regression	0.934	0.941	0.943	0.927
8	Naive Bayes Classifier	0.605	0.454	0.292	0.997

Gambar V.2 Sistem Evaluasi

2. Menyimpan Model Terbaik

Setelah mendapatkan model terbaik selanjutnya kita simpan model tersebut untuk kita rubah menjadi format Pickle. Dimana model terbaik yaitu *Gradient Boosting Classifier* dan untuk menyimpannya menggunakan script berikut

```
#  XGBoost Classifier Model
from xgboost import XGBClassifier

# instantiate the model
gbc = GradientBoostingClassifier(max_depth=4,learning_rate=0.7)

# fit the model
gbc.fit(X_train,y_train)
```

```
GradientBoostingClassifier(learning_rate=0.7, max_depth=4)
```

Gambar V.3 Sistem Evaluasi

3. Ubah Menjadi Format Pickle

Tahapan selanjutnya yaitu mengubah model yang sudah dipilih menjadi format pickle berikut penerapannya.

```
import pickle

# dump information to that file
pickle.dump(gbc, open('model1.pkl', 'wb'))
```

Gambar V.4 Sistem Evaluasi

Pada script di atas, dilakukan ***import library pickle***. Kemudian, dilakukan proses serialisasi dengan menggunakan *method* ***dump()*** dari ***library pickle***. *Method* ini membutuhkan dua argumen yaitu object yang akan diserialisasi (dalam hal ini *object gbc* yang merupakan model yang telah dilatih) dan file yang akan digunakan untuk menyimpan hasil serialisasi (dalam hal ini file **model1.pkl**).

4. Import Library Pickle

Import terlebih dahulu untuk library yang digunakan untuk keperluan penggunaan model.

integration.py 5 ×

integration.py > ...

```
#importing required libraries

from flask import Flask, request, render_template
import numpy as np
import pandas as pd
from sklearn import metrics
import warnings
import pickle
import requests
warnings.filterwarnings('ignore')
from feature import FeatureExtraction
```

Gambar V.5 Sistem Evaluasi

Lalu panggil model1.pkl dan lakukan load untuk filenya.

5. Pembuatan Script pada Flask

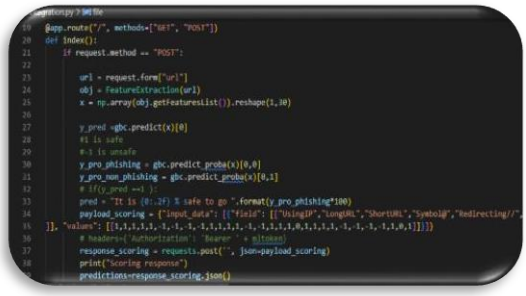
```
file = open("model1.pkl","rb")
gbc = pickle.load(file)
file.close()

app = Flask(__name__)
```

Gambar V.6 Sistem Evaluasi

Script di atas membuka file bernama "**model1.pkl**" dengan mode 'rb' (read binary). File ini diasumsikan berisi objek yang telah disimpan menggunakan pickle. Kemudian, objek tersebut dimuat menggunakan method **pickle.load()** dan disimpan ke dalam variabel gbc. Setelah itu, file ditutup dengan menggunakan method close().

Setelah objek dimuat, sebuah objek dari kelas Flask juga dibuat dengan menggunakan perintah "**app = Flask(name)**". Objek ini akan digunakan untuk membuat aplikasi web dengan Flask.

Selanjutnya buatlah script untuk dapat mengekstraksi hasil yang telah dihasilkan saat membuat model.



Gambar V.7 Sistem Evaluasi

Kirim Hasilnya kedalam template index.html dengan menggunakan render, dan akses via web browser.

Gambar V.7 Sistem Evaluasi

6. Hasil Aplikasi

Pada bagian hasil ini kita akan melakukan pengecekan url pada system phishing detection langkah pertama yaitu jalankan aplikasi flask, jika sudah muncul halaman berikut, klik tombol merah maka akan diarahkan kepada bagian cek url.

Gambar V.8 Sistem Evaluasi

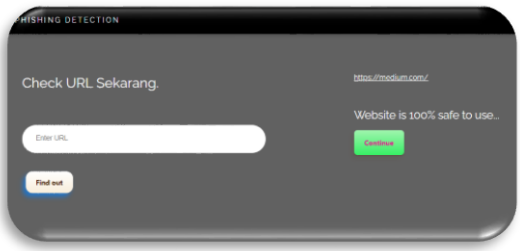
Selanjut silahkan paste atau ketikan url yang akan dilakukan pengecekan pada kolom tersebut lalu klik tombol **find out.**

Gambar V.8 Sistem Evaluasi

Sebagai contoh pertama kita coba inputkan https://atacado-barato-magazine.myshopify.com/ link berikut pada kolom pengecekan dan dipat hasilnya yaitu “system mendeteksi bahwa situs tersut merupakan phishing”

Gambar V.9 Sistem Evaluasi

Selanjutkan kita coba pengecekan kedua dengan contoh website dari medium yaitu: https://medium.com. Maka dari hasil pengecekan, terdapat hasil bahwa website medium baik untuk diakses.



Gambar V.10 Sistem Evaluasi

VI. KESIMPULAN DAN SARAN

1. Kesimpulan

kemajuan teknologi dapat membantu masyarakat dalam beraktivitas, namun juga

dapat menimbulkan ancaman keamanan seperti phishing. Phishing adalah kegiatan yang bertujuan untuk menipu pengguna internet dengan cara memancing mereka untuk memberikan informasi pribadi. Hal ini dapat menyebabkan kerugian baik dari segi privasi maupun keuangan bagi individu atau lembaga/perusahaan.

Upaya untuk meminimalisir risiko phishing adalah dengan melakukan identifikasi dan deteksi website phishing menggunakan metode klasifikasi data mining. Algoritma machine learning yang dapat digunakan dalam hal ini adalah algoritma supervised learning. Algoritma ini mampu membuat prediksi tentang nilai output berdasarkan dataset pelatihan yang diketahui dan dapat membandingkan hasil prediksi dengan hasil yang diinginkan untuk menemukan kesalahan dan memodifikasi model sesuai dengan hasilnya.

2. Saran

Saran yang dapat diberikan adalah untuk tetap waspada terhadap tautan yang didapat dari spam atau jejaring media sosial dan jangan pernah memberikan informasi pribadi kepada pihak yang tidak dikenal. Selain itu, pastikan untuk selalu menggunakan alat-alat deteksi phishing yang tersedia dan terus belajar tentang cara mengidentifikasi dan mencegah phishing untuk meningkatkan keamanan online.

DAFTAR PUSTAKA

[1] Wahyudi Diki, Niswar Muhammad. 2022. "*Website Phising Detection Application Using Vektor Machine (SVM)*". Journal of Information Technology and Its Utilization, Volume 5 : 18.

[2] Deekshitha B., Aswitha Ch, Dkk, 2022. "*URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms*". ***International Journal for Research in Applied Science & Engineering Technology (IJRASET*),** Volume 10.

[3] Deekshitha B., Aswitha Ch, Dkk, 2022. "*URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms*". ***International Journal for Research in Applied Science & Engineering Technology (IJRASET*),** Volume 10.

[4] Halim Z. 2017. "*Prediksi Website Pemancing Informasi Penting Phising Menggunakan Support Vector Machine (SVM)*". Information System for Educators and Professionals. 2 (1): 71 – 82.

[5] Sadli Muhammad, Fajriana, Fuadi Wahyu, 2018, Dkk. "*Penerapan Model K-Nearest Neighbors Dalam Klasifikasi Kebutuhan Daya Listrik Untuk masing-masing Daerah di Kota Lhokseumawe*". Jurnal ECOTIPE, Volume 5, No.2.

[6] Altaher Taha Altyeb, 2017. "*Phishing Websites Classification using Hybrid SVM and KNN Approach*". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol.8, No.6.

[7] Kumar Narander, Chaudhary Priyanka, 2017. "*Mobile Phishing Detection using Naive Bayesian Algorithm".* IJCSNS International Journal of Computer Science and Network Security, VOL.17 No.7

[8] Mao Jian, Bian Jingdong Bian, Tian Wenqian, Dkk, 2019. "*Phishing page detection via learning classifiers from page layout feature"* EURASIP Journal on Wileress Communication and Networking. No.43.

[9] Kumar Dutta Ashit, 2021. "*Detecting phishing websites using machine learning technique*". Detecting phishing websites using machine learning technique. PLoS ONE 16(10): e0258361.

[10] Deekshitha B., Aswitha Ch, Dkk, 2022. "*URL Based Phishing Website Detection by Using Gradient and Catboost Algorithms*". ***International Journal for Research in Applied Science &***

Engineering Technology (IJRASET), Volume 10.

[11] Chian Fang Lim, Ayop Zakiah, Anawar Syarulnaziah, Dkk, 2021. "*URL Phishing Detection System Utilizing Catboost Machine Learning Approach".* IJCSNS International Journal of Computer Science and Network Security, VOL.21 No.9.

[12] Srinath, K. R, 2017. "*Python – The Fastest Growing Programming Language*". International Research Journal of Engineering and Technology (IRJET). Volume 4.

[13] Irsyad Rahadian, "*Penggunaan Python Web Framework Flask Untuk Pemula*". Laboratorium Telematika, Sekolah Teknik Elektro & Informatika.

[14] Setiabudidaya Dedi. "*PENGGUNAAN PIRANTI LUNAK JUPYTER NOTEBOOK DALAM UPAYA MENSOSIALISASIKAN OPEN SCIENCE*". Universitas Sriwijaya, Jl. Raya Palembang Prabumulih KM 32, Indralaya 30662.

[15] Permana Yudi, Romadlon Puji, 2019. "*PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE". Volume 10 Nomor 2 Desember 2019 ISSN : 2407-3903.*

[16] C. Price Danny, Van der Velden Ellert, Dkk, 2018. "*Hickle: A HDF5-based python pickle replacement*". Journal of Open Source Software, 3(32), 1115.

[17] Arif Alfis, Isro Yogi, Dkk, 2017. "*RANCANG BANGUN WEBSITE SEKOLAH MENENGAH PERTAMA (SMP) NEGERI 8 KOTA PAGAR ALAM*". Jurnal Ilmiah Betrik, Vol. 08, No.03.

[18] Subarkah Pungkas, Nur Ikhsan Ali, 2021. "*IDENTIFIKASI WEBSITE PHISHING MENGGUNAKAN ALGORITMA CLASSIFICATION AND REGRESSION TREES (CART*)". Jurnal Ilmiah Informatika (Scientific Informatics Journal) with CC BY NC licence, P. Subarkah dkk/ JIMI 6 (2) pp. 127-136.

[19] Wahyudi Diki, Niswar Muhammad. 2022. "*Website Phising Detection Application Using Vektor Machine (SVM)*". Journal of Information Technology and Its Utilization, Volume 5 : 18.