Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

Natural Language Processing (NLP) and Support Vector Machine (SVM) Optimization in Detecting Phishing Website URLs

Mhd Adi Setiawan Aritonang[*1], Maradona Jonas Simanulang[2], Toras Pangidoan Batubara[3], Imanuel Zega[4], M Hafis Afrizal[5]

[1,5]Teknologi Informasi, Teknik Komputer, Institut Teknologi Batam, Indonesia
[2]Teknologi Informasi, Universitas Senior Medan, Indonesia
[3]Sistem Informasi, Universitas Murni Teguh, Indonesia
[4]Sistem Informasi, Universitas Pignatelli Triputra, Indonesia

Email: [1]adi@iteba.ac.id

Received : Sep 25, 2025; Revised : Oct 13, 2025; Accepted : Nov 7, 2025; Published : Feb 15, 2026

Abstract

Phishing remains one of the most pervasive cyber-threats, with recent reports indicating a sharp rise in both volume and sophistication of attacks. According to the Anti-Phishing Working Group, phishing incidents reached nearly 1 million in Q4 2024. To address this evolving threat, this study aims to develop an automated phishing-URL classification system based on Natural Language Processing (NLP) and Support Vector Machine (SVM). We utilised the Kaggle "PhiUSIIL Phishing URL Dataset" comprising 256,795 URL records and applied comprehensive preprocessing, feature extraction (structural URL features plus NLP-based keyword analysis), and SVM training with grid search optimisation. Evaluation was performed via confusion matrix and standard metrics of accuracy, precision, recall and F1-score. The best model achieved an accuracy of 99.99%, precision of 99.98%, recall of 100%, and F1-score of 99.99%. These results demonstrate that the combined NLP + SVM approach can effectively distinguish phishing from legitimate URLs with very high reliability. The proposed system contributes to cybersecurity by offering a feasible AI-based solution for real-time URL screening that can be integrated into browser extensions or enterprise email filters to bolster phishing defences.

Keywords : *Cybersecurity, Natural Language Processing, Phishing Detection, Support Vector Machine, URL Classification*

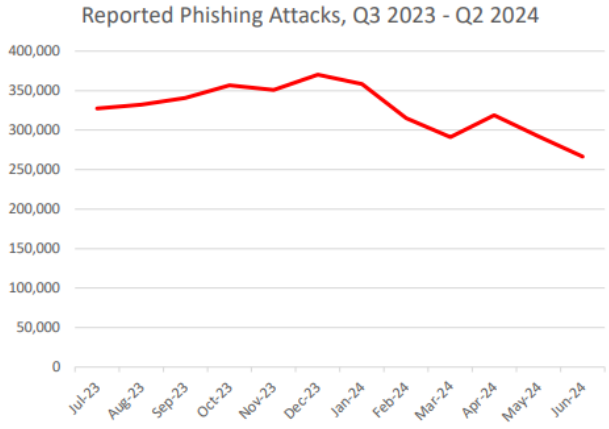
1. PENDAHULUAN

Serangan phishing merupakan salah satu bentuk kejahatan siber yang paling sering terjadi di era digital modern. Serangan ini memanfaatkan teknik rekayasa sosial dengan membuat situs palsu yang menyerupai situs resmi untuk mencuri informasi sensitif seperti kredensial login, data perbankan, dan identitas pengguna. Berdasarkan laporan Anti-Phishing Working Group (APWG) tahun 2024, jumlah serangan phishing meningkat lebih dari 35% dibandingkan tahun sebelumnya, dengan lebih dari 1,3 juta domain phishing aktif di seluruh dunia [1]. Laporan dari National Cyber Security Centre (NCSC) [2]dan FBI Internet Crime Complaint Center (IC3) [3] juga menunjukkan bahwa phishing masih menjadi bentuk kejahatan siber paling dominan dalam tiga tahun terakhir. Peningkatan signifikan ini menegaskan perlunya sistem pendeteksian otomatis berbasis kecerdasan buatan untuk mendeteksi situs phishing secara cepat dan akurat.

Berbagai penelitian telah dilakukan untuk mengidentifikasi situs phishing menggunakan pendekatan Machine Learning (ML). Zamir et al. [4] menerapkan beberapa algoritma ML seperti Support Vector Machine (SVM), Random Forest, dan Decision Tree untuk mengklasifikasikan URL berbahaya, namun performanya masih terbatas pada pola data yang linear. Tamal et al. [5] menggunakan

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

Random Forest untuk deteksi URL phishing dan memperoleh nilai recall sebesar 95%, tetapi modelnya belum mampu mengenali variasi pola baru. Anand et al.[6] mengusulkan penggunaan SVM dengan fitur berbasis struktur dan host URL, menghasilkan akurasi tinggi namun belum optimal terhadap perubahan dinamis URL phishing. Penelitian oleh Catal et al. [7] dan Balogun et al. [8] juga menunjukkan hasil yang baik namun masih bergantung pada fitur statis seperti panjang URL atau jumlah simbol tertentu, sehingga rentan menurun performanya terhadap pola baru.



Gambar 1. Grafik Report Attack pada URL

Dalam perkembangan selanjutnya, beberapa penelitian mulai memanfaatkan pendekatan Natural Language Processing (NLP) dan Deep Learning untuk menangkap pola semantik dari URL phishing. Kalla & Kuraku [9] memanfaatkan tokenisasi kata dan word embedding untuk mengekstraksi konteks linguistik dari URL dan meningkatkan akurasi klasifikasi. Uddin et al. [10] menerapkan representasi berbasis embedding pada model hibrid dan menunjukkan peningkatan performa pada data non-linear. Haq et al. [11] menggunakan arsitektur Convolutional Neural Network (CNN) untuk klasifikasi phishing URL dan memperoleh hasil yang unggul, namun membutuhkan sumber daya komputasi tinggi. Catal & Giray [12] dalam tinjauan sistematisnya menyebut bahwa model Deep Learning memiliki akurasi tinggi tetapi kurang efisien untuk implementasi waktu nyata. Oleh karena itu, integrasi antara NLP dan SVM dipandang sebagai solusi yang menjanjikan karena mampu menyeimbangkan antara akurasi dan efisiensi [13].

Walaupun berbagai pendekatan telah dikembangkan, sebagian besar penelitian sebelumnya masih memperlakukan URL sebagai deretan karakter tanpa mempertimbangkan aspek linguistik yang dapat mengindikasikan niat phishing, seperti penggunaan kata login, verify, atau secure [14]. Selain itu, SVM konvensional masih memiliki keterbatasan dalam menangani pola data non-linear apabila tidak dioptimalkan dengan representasi fitur kontekstual [15]. Beberapa penelitian terdahulu juga menunjukkan bahwa model ML klasik bergantung pada fitur statis yang tidak cukup adaptif terhadap perubahan pola serangan phishing [16][17]. Berdasarkan tinjauan pustaka sistematis [18][19][20], masih terdapat celah penelitian (*research gap*) dalam hal optimasi SVM menggunakan pendekatan NLP untuk meningkatkan kemampuan klasifikasi terhadap URL phishing yang bersifat dinamis.

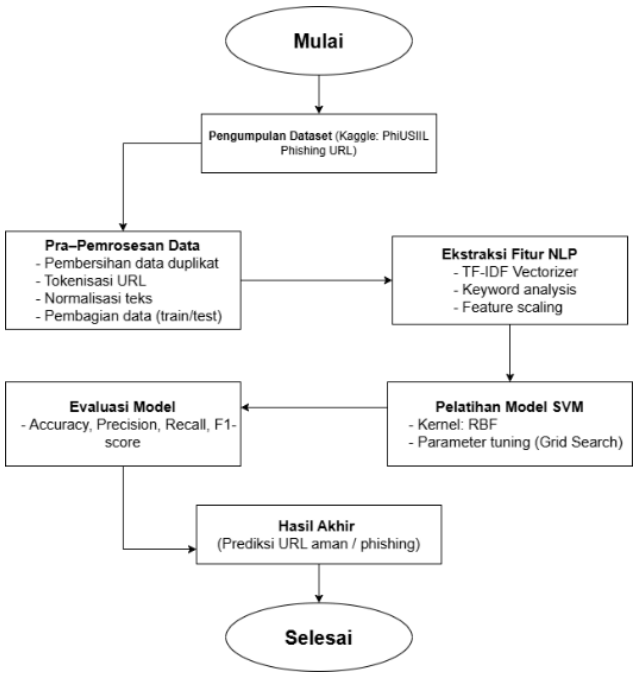
Berbeda dengan penelitian terdahulu seperti Anand et al. [6] dan Tamal et al. [5] yang masih mengandalkan fitur tradisional, penelitian ini mengusulkan pendekatan terpadu antara NLP dan SVM yang dioptimalkan melalui grid search. Pendekatan ini memanfaatkan analisis linguistik pada token URL untuk mengekstraksi fitur kontekstual yang kemudian diklasifikasikan menggunakan algoritma SVM kernel Radial Basis Function (RBF). Integrasi ini diharapkan mampu mengatasi keterbatasan SVM dalam mendeteksi pola non-linear pada data phishing serta meningkatkan nilai recall dari 95% pada penelitian sebelumnya [5] menjadi 100% pada penelitian ini.

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada pengembangan optimasi model terhadap analisis URL *phishing* menggunakan pendekatan *Natural Language Processing* dan *Support Vector Machine*. Tujuan khusus dari penelitian ini adalah untuk mengembangkan sistem optimasi dalam mengidentifikasi analisis URL dalam deteksi *phishing*, serta meningkatkan efisiensi, dan konsistensi dibandingkan metode *blacklist*. Oleh karena itu, hasil dari penelitian ini diharapkan dapat berkontribusi dalam pengembangan optimasi analisis deteksi phishing, sehingga dapat dilakukan secara responsif dan terstandar. Dengan demikian, tujuan utama penelitian ini adalah mengembangkan sistem deteksi phishing berbasis analisis URL menggunakan pendekatan NLP dan SVM yang akurat, efisien, serta dapat diterapkan pada lingkungan dunia nyata untuk memperkuat keamanan siber berbasis kecerdasan buatan.

2. METODE

Penelitian ini dilakukan dengan langkah-langkah yang terstruktur secara sistematis. Proses dimulai dengan mengumpulkan data URL dari *Kaggle*, kemudian dilanjutkan dengan proses *pra-pemrosesan* data untuk mempersiapkan input yang bersih dan terorganisir. Setelah itu, dirancang model klasifikasi yang kemudian dilatih menggunakan data tersebut. Setelah pelatihan selesai, dilakukan evaluasi untuk mengetahui seberapa baik akurasi dan efektivitas model. Tahap terakhir adalah analisis hasil untuk menarik kesimpulan mengenai kinerja model yang telah dibuat. Seluruh alur proses telah ditunjukkan dalam *flowchart* berikut.



Gambar 2. Diagram Alir Tahapan Penelitian

Diagram alir pada Gambar 2 menjelaskan alur penelitian yang dilakukan secara bertahap mulai dari pengumpulan dataset hingga evaluasi model. Tahapan pertama adalah pengumpulan data, di mana dataset *PhiUSIIL Phishing URL* diperoleh dari platform Kaggle dan berisi data URL *phishing* serta legitimate. Selanjutnya dilakukan pra–pemrosesan data, meliputi pembersihan data duplikat, normalisasi teks URL, tokenisasi, serta pembagian dataset menjadi data latih dan data uji. Tahap berikutnya adalah ekstraksi fitur berbasis NLP, yaitu proses konversi teks URL menjadi representasi numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) dan analisis kata kunci (*keyword analysis*). Hasil ekstraksi fitur ini digunakan pada tahap pelatihan model SVM

(Support Vector Machine) dengan kernel *Radial Basis Function* (RBF) dan optimasi parameter menggunakan *Grid Search*.Terakhir, dilakukan evaluasi model untuk mengukur performa sistem menggunakan empat metrik utama, yaitu akurasi, presisi, recall, dan F1-score. Proses ini menghasilkan sistem yang mampu mengklasifikasikan URL secara otomatis menjadi kategori phishing atau legitimate.

2.1. Pengumpulan Dataset

Dataset yang digunakan dalam penelitian ini adalah *PhiUSIIL Phishing URL Dataset* yang diunduh dari UCI Machine Learning Repository pada tanggal 25 Agustus 2025 melalui tautan: https://archive.ics.uci.edu/dataset/967/phiusiil+phishing+url+dataset. Dataset ini awalnya dikembangkan sebagai bagian dari penelitian oleh A. Prasad. (2024) dan juga tersedia di platform Kaggle dengan struktur yang identik. Dataset tersebut terdiri dari 256.795 baris dan 56 fitur (atribut) yang merepresentasikan berbagai karakteristik dari sebuah *Uniform Resource Locator* (URL) [21].

Setiap entri dalam dataset diklasifikasikan ke dalam dua kategori, yaitu *phishing* (label = 1) dan *legitimate* (label = 0). Fitur-fitur di dalam dataset mencakup berbagai aspek numerik dan kategorikal seperti *URLLength*, *NumDots*, *NumSubdomain*, *PrefixSuffix*, *DomainAge*, *HTTPS*, *WebTraffic*, dan *DomainReputation*. Fitur-fitur numerik tersebut diolah melalui proses standardisasi menggunakan metode *StandardScaler*, agar setiap atribut memiliki skala yang sebanding. Rumus yang digunakan untuk proses standardisasi ditunjukkan pada Persamaan.

$$z = \frac{x-\mu}{\sigma} \quad (1)$$

Keterangan: x adalah nilai asli fitur, adalah rata-rata, dan σ adalah simpangan baku dari fitur tersebut. Standardisasi ini dilakukan untuk menghindari dominasi fitur dengan rentang nilai yang besar terhadap hasil klasifikasi.

Selain fitur numerik, penelitian ini juga melakukan analisis berbasis *Natural Language Processing* (NLP) pada kolom URL. Proses ini mencakup tokenisasi URL menggunakan *library* NLTK, di mana setiap URL dipecah berdasarkan delimiter seperti tanda garis miring (/), tanda hubung (-), titik (.), dan angka. Token-token yang dihasilkan kemudian diubah menjadi representasi numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) dengan bantuan *scikit-learn vectorizer*. Pendekatan ini memungkinkan sistem untuk mengenali pola linguistik seperti kemunculan kata "*login*", "*secure*", "*account*", atau "*verify*" yang sering digunakan dalam situs *phishing*.

Selanjutnya, dilakukan pemeriksaan distribusi data terhadap kolom label. Berdasarkan hasil analisis, dataset memiliki komposisi *phishing* sebesar 53,4% dan *legitimate* sebesar 46,6%, sehingga masih berada dalam kategori relatif seimbang. Oleh karena itu, teknik penyeimbangan data seperti *Synthetic Minority Oversampling Technique* (SMOTE) tidak diterapkan pada penelitian ini. Namun, metode tersebut dapat dipertimbangkan pada penelitian lanjutan apabila distribusi data menjadi tidak seimbang.

Dengan struktur yang kaya akan fitur numerik dan linguistik, serta ukuran data yang besar, dataset ini dinilai representatif untuk digunakan sebagai dasar pelatihan model deteksi *phishing* berbasis NLP dan algoritma *Support Vector Machine* (SVM).

2.2. Pra – Pemrosesan Data

Pada tahapan ini, dilakukan proses data preprocessing, yang merupakan serangkaian teknik untuk mempersiapkan dataset agar bisa digunakan dalam ekstraksi fitur secara lebih baik. *Preprocessing* adalah langkah penting dalam analisis data yang bertujuan untuk membersihkan data dari kesalahan dan membuat skala variabel-variabel tersebut menjadi sama. Tahap ini bertujuan mengubah data mentah menjadi bentuk yang rapi, bersih, seragam, dan terstruktur, sehingga lebih mudah diolah dengan

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

komputasi. Proses ini juga memungkinkan setiap teks yang menjadi label diubah menjadi angka unik, yang nantinya bisa digunakan oleh algoritma klasifikasi dalam proses pelatihan dan evaluasi model. Seluruh tahapan pra-pemrosesan ini dirancang agar data bisa dipersiapkan secara optimal, sehingga lebih efektif dalam pelatihan model dengan input yang sudah terstruktur dan bersih [22].

	id	url	label	url_length	num_subdomains	has_ip	has_at_symbol	hostname_length	hostname_has_dash	num_slashes	has_question_mark	has_equals_sign	has_dot
0	1	http://www.crestonwood.com/router.php	1	37	1	0	0	19	0	3	0	0	1
1	2	https://www.uni-mainz.de	1	24	1	0	0	16	1	2	0	0	1
2	3	https://www.voicefmradio.co.uk	1	30	2	0	0	22	0	2	0	0	1
3	4	https://www.sfnmjournal.com	1	27	1	0	0	19	0	2	0	0	1
4	5	https://www.rewildingargentina.org	1	34	1	0	0	26	0	2	0	0	1

Gambar 3. *Pre-Processing* Data

Gambar 9 menampilkan sebagian dari dataset *PhiUSIIL Phishing URL* yang digunakan pada penelitian ini. Dataset ini memuat berbagai atribut yang merepresentasikan karakteristik struktural dari sebuah URL. Setiap baris pada tabel menunjukkan satu sampel data yang terdiri atas beberapa fitur seperti *url_length*, *num_subdomains*, *has_ip*, *hostname_length*, dan sebagainya. Kolom label menunjukkan kategori dari URL tersebut, di mana nilai 1 menandakan situs *phishing* dan 0 menandakan situs *legitimate* (aman).

Sebagai contoh, pada baris pertama terlihat URL http://www.crestonwood.com/router.php memiliki panjang URL *(url_length*) sebesar 37 karakter, jumlah subdomain sebanyak satu, serta tidak mengandung alamat IP (has_ip = 0). Sementara kolom *num_slashes* menunjukkan jumlah tanda garis miring (“/”) pada struktur URL yang dapat menjadi indikator pola *phishing* tertentu.

Berdasarkan pengamatan awal terhadap dataset, fitur-fitur seperti panjang URL (*url_length*), jumlah subdomain (*num_subdomains*), dan keberadaan karakter khusus seperti tanda tanya (*has_question_mark*) dan tanda sama dengan (*has_equals_sign*) dapat menjadi variabel penting dalam proses klasifikasi. Informasi ini kemudian digunakan pada tahap pra–pemrosesan data untuk standarisasi dan ekstraksi fitur berbasis NLP sebelum dilakukan pelatihan model SVM.

2.3. Perancangan Model dan Pelatihan Model

Dalam tahap perancangan model ini, dataset dibagi menjadi dua bagian, yaitu 80% digunakan sebagai data latihan dan 20% sebagai data uji. Pembagian ini dilakukan agar model bisa dilatih dengan data yang cukup banyak, sedangkan data uji digunakan untuk mengetahui sejauh mana kemampuan model bekerja secara objektif.

```
# Split data: 80% train, 20% test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

Gambar 4. Pembagian Dataset Menjadi Dua Bagian

Algoritma *Support Vector Machine* (SVM) dipilih karena kemampuannya dalam mengklasifikasikan data menjadi dua kategori. *Support Vector Machine* bekerja dengan mencari garis pemisah terbaik yang mampu membedakan URL *phishing* dari URL yang sah berdasarkan hasil ekstraksi dari metode *Natural Languange Processing*.

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

```
# Inisialisasi model SVM
svm_model = SVC(kernel='linear', random_state=42)

# Training model
svm_model.fit(X_train_scaled, y_train)
```

```
SVC
SVC(kernel='linear', random_state=42)
```

Gambar 5. Pelatihan Model

2.4. Evaluasi Model

Evaluasi model dilakukan untuk mengevaluasi kemampuan prediksi terhadap dua jenis label keluaran, yaitu kategori *phishing* dan *legitimate*. Setelah proses pelatihan selesai, model diuji menggunakan data uji yang telah dipisahkan sebelumnya. Evaluasi mencakup perhitungan berbagai metrik seperti akurasi, *precision*, *recall*, *F1-Score*, dan *Confusion Matrix*. Penggunaan metrik tersebut bertujuan untuk mengetahui seberapa baik model dalam mengenali setiap kelas, terutama dalam kondisi distribusi kelas yang tidak seimbang. Hasil evaluasi memberikan gambaran mengenai tingkat akurasi model dalam mendeteksi URL *phishing* dengan pendekatan *Natural Language Processing* dan *Support Vector Machine*, serta menjadi dasar dalam menentukan seberapa efektif model yang dikembangkan.

Untuk mengevaluasi kinerja model secara kuantitatif pada tugas ekstraksi entitas, digunakan empat kategori dasar yang umum dipakai dalam bidang *Natural Language Processing* (NLP) dan *Support Vector Machine* (SVM). Evaluasi dilakukan pada tingkat entitas dengan membandingkan hasil prediksi terhadap ground truth. Empat kategori tersebut adalah:

a. *True Positive* (TP): *phishing* terdeteksi dengan benar
b. *False Positive* (FP): *legitimate* terdeteksi sebagai *phishing*
c. *False Negative* (FN): *phishing* diklasifikasikan sebagai *legitimate*
d. *True Negative* (TN): *legitimate* terdeteksi benar

Berdasarkan klasifikasi tersebut, metrik-metrik berikut dihitung untuk setiap skenario:

a. Precision: Metrik ini secara spesifik mengukur tingkat keandalan atau kepercayaan dari entitas yang berhasil diekstrak oleh model. Presisi yang tinggi mengindikasikan bahwa model jarang membuat kesalahan dengan "menciptakan" data yang tidak ada atau salah mengklasifikasikan entitas, sehingga memiliki tingkat *False Positive* (FP) yang rendah.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2)$$

b. Recall: Metrik ini mengukur tingkat kelengkapan atau kesensitifan model dalam menemukan semua entitas yang relevan dari teks. Recallyang tinggi menunjukkan bahwa model mampu mengidentifikasi sebagian besar entitas yang seharusnya ada dan tidak banyak yang terlewat, sehingga memiliki tingkat *False Negative* (FN) yang rendah.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

c. F1-Score: Merupakan rata-rata harmonik dari Presisi dan *Recall*, memberikan sebuah skor tunggal yang menyeimbangkan kedua metrik tersebut. F1-Score sangat berguna terutama jika terdapat ketidakseimbangan antara jumlah FP dan FN.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

d. Accuration: Menunjukkan tingkat ketepatan model dalam memprediksi URL dengan benar.

$$f_{baud} = \frac{2^{SMOD}}{64} x f_{osc} \quad (5)$$

e. Confusion Matrix : Digunakan untuk melihat distribusi hasil prediksi secara visual antara kelas *phishing* dan *legitimate*.

Secara kolektif, penggunaan ketiga metrik ini memberikan pandangan diagnostik yang komprehensif terhadap perilaku model. Presisi mengukur kualitas dari apa yang ditemukan, *Recall* mengukur kuantitas dari apa yang seharusnya ditemukan, dan *F1-Score* merangkum efektivitas keseluruhan dalam satu angka.

Pada tahap ini dilakukan uji apakah *Natural Language Processing* dan *Support Vector Machine* dapat digunakan sebagai algoritma untuk memprediksi penelitian ini. Untuk mendapatkan jawabannya, diperlukan beberapa fungsi klasifikasi. Dalam pemodelan *Support Vector Machine*, terlebih dahulu dilakukan prediksi, kemudian dilihat *confusion matrix*, *classification report*, nilai akurasi, nilai AUC, serta AUC *Curve* [23]. Semua proses dilakukan menggunakan bahasa pemrograman Python, dan hasil dari pengujian ini adalah sebagai berikut:

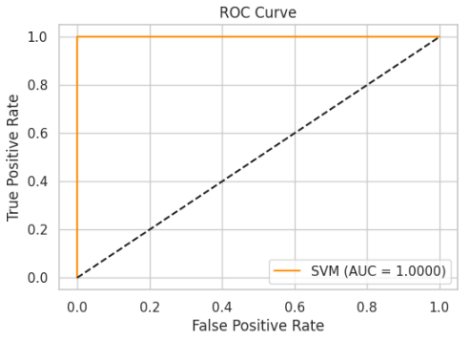
```
# Prediksi pada data uji
y_pred = svm_model.predict(X_test_scaled)

# Evaluasi Akurasi dan Laporan Klasifikasi
print("Akurasi Model:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

```
Akurasi Model: 0.9999151805593842
```

Gambar 6. Hasil Akurasi dari Model

Akurasi model *Support Vector Machine* mencapai 99%,itu menunjukkan bahwa model memiliki kinerja yang sangat baik dalam mengklasifikasikan data pengujian dengan benar. Namun, akurasi saja tidak selalu memberikan gambaran yang lengkap tentang kinerja model, terutama dalam kasus dataset yang tidak seimbang atau jika model cenderung mengklasifikasikan satu kelas lebih sering daripada kelas lainnya. Oleh karena itu, meskipun akurasi tinggi, penting untuk juga mengevaluasi metrik lain seperti confusion matrix, ROC curve,dan AUC score untuk memahami lebih dalam seberapa baik model dalam menangani kedua kelas (kelas "0 = *Phishing*" dan "1 = Legitimate"). Dalam beberapa kasus, meskipun akurasi tinggi, model masih melakukan kesalahan signifikan dalam mengklasifikasikan kelas minoritas, yang bisa mengurangi keandalan keputusan yang dihasilkan.



Gambar 7. Diagram ROC Curve

Untuk menghitung dan menampilkan kurva ROC (*Receiver Operating Characteristic*) serta skor AUC (*Area Under the Curve*) sebagai cara mengevaluasi performa model klasifikasi, kita menggunakan

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

fungsi predict_proba untuk memperoleh probabilitas kelas positif. Selanjutnya, fungsi roc_curve digunakan untuk menghitung tingkat *False Positive Rate* (FPR) dan *True Positive Rate* (TPR) [24]. Skor AUC dihitung dengan fungsi auc, yang mengukur area di bawah kurva ROC [25]. Grafik ROC menunjukkan trade-off antara FPR dan TPR, dengan skor AUC sebagai indikator performa model: semakin mendekati angka 1, semakin baik kemampuan model dalam membedakan antara kedua kelas.

3. HASIL

Alat yang digunakan dalam penelitian ini adalah *Google Colab* dan bahasa pemrograman Python. Untuk membuat *association rules* dengan Python, digunakan beberapa *library* tertentu. *Library pandas* digunakan untuk mengolah data yang digunakan. *Library seaborn* digunakan untuk menampilkan visualisasi data yang digunakan. Jadi, semua *library* tersebut harus diimport terlebih dahulu sebelum menjalankan proses perhitungan algoritma. Penelitian ini juga berhasil mengembangkan dan melatih model klasifikasi berbasis *Support Vector Machine* untuk mengidentifikasi apakah URL tersebut *phishing* atau legitem. Evaluasi kinerja dilakukan terhadap data uji sebesar 20% dan data latih 80% dari total data, serta model terbaik disimpan berdasarkan nilai *F1-Score* yang tinggi dalam proses klasifikasi. Berikut gambar *listing* program *import library*.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import re
import urllib.request
import urllib
import math
from urllib.parse import urlparse
from collections import Counter

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, roc_curve, auc
from bs4 import BeautifulSoup
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

import joblib
import warnings
```

Gambar 8. Program Import Library

3.1. Gambaran Umum Dataset

Dataset yang digunakan pada penelitian ini adalah *PhiUSIIL_Phishing_URL* Dataset, yang diperoleh dari platform Kaggle [26]. Dataset ini terdiri dari 256.795 baris data dengan 56 kolom fitur, di mana salah satunya adalah kolom target dengan label 0 (*legitimate*) dan 1 (*phishing*). Dataset ini dipilih karena memiliki jumlah data yang besar, representatif, serta mencakup berbagai karakteristik URL yang umum digunakan dalam penelitian deteksi *phishing*.

	id	URLLength	DomainLength	IsDomainIP	URLSimilarityIndex	CharContinuationRate	TLDLegitimateProb	URLCharProb	TLDLength	NoOfSubDomain	...	Crypto	HasCopyrightInfo	NoOfImage	NoOfCSS	NoOfJS	NoOfSelfRef
count	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	...	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000	235795.000000
mean	117898.000000	34.573121	21.470396	0.002706	78.430778	0.845508	0.260423	0.055747	2.764456	1.164758	...	0.023474	0.486775	26.075889	6.333111	10.522305	65.071113
std	68068.297699	41.314152	9.150793	0.051946	28.976055	0.216632	0.261628	0.010587	0.599739	0.600969	...	0.151403	0.499826	79.411815	74.886298	22.312192	176.687539
min	1.000000	13.000000	4.000000	0.000000	0.155574	0.000000	0.000000	0.001083	2.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	58949.500000	23.000000	16.000000	0.000000	57.024793	0.680000	0.005977	0.050747	2.000000	1.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	117898.000000	27.000000	20.000000	0.000000	100.000000	1.000000	0.079963	0.057970	3.000000	1.000000	...	0.000000	0.000000	8.000000	2.000000	6.000000	12.000000
75%	176846.500000	34.000000	24.000000	0.000000	100.000000	1.000000	0.522907	0.062875	3.000000	1.000000	...	0.000000	1.000000	29.000000	8.000000	15.000000	88.000000
max	235795.000000	6097.000000	110.000000	1.000000	100.000000	1.000000	0.522907	0.090824	13.000000	10.000000	...	1.000000	1.000000	8956.000000	35820.000000	6957.000000	27397.000000

8 rows × 53 columns

Gambar 9. Statistik Deskriptif Tabel Dataset

Gambar 10 menampilkan hasil analisis statistik deskriptif dari dataset *PhiUSIIL Phishing* URL yang digunakan dalam penelitian ini. Analisis ini mencakup ukuran-ukuran statistik seperti jumlah data

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

(*count*), nilai rata-rata (*mean*), simpangan baku (*standard deviation*), nilai minimum (*min*), maksimum (*max*), serta nilai kuartil (25%, 50%, dan 75%) untuk setiap fitur numerik yang ada dalam dataset.

Berdasarkan hasil pada Gambar 10, terlihat bahwa panjang URL (*URLLength*) memiliki nilai rata-rata sebesar 54,57 karakter, dengan panjang maksimum mencapai 110 karakter. Hal ini menunjukkan bahwa situs *phishing* cenderung memiliki URL yang lebih panjang dibandingkan situs *legitimate*. Fitur *DomainLength* juga menunjukkan variasi yang cukup tinggi dengan rata-rata 21,47, yang menandakan bahwa banyak domain dalam dataset memiliki nama yang panjang dan kompleks.

Selain itu, nilai *standard deviation* yang relatif besar pada beberapa fitur seperti *NoOfImage* dan *NoOfCSS* menunjukkan adanya variasi signifikan antar situs, yang dapat memengaruhi performa model dalam membedakan pola URL *phishing* dan *legitimate*. Nilai fitur *IsDomainIP* dan *Crypto* yang mendekati nol menunjukkan bahwa sebagian besar URL tidak menggunakan alamat IP langsung maupun metode enkripsi pada domainnya.

Secara keseluruhan, hasil analisis statistik ini menggambarkan bahwa dataset memiliki karakteristik yang beragam dan cukup kaya fitur, sehingga layak digunakan untuk membangun model klasifikasi berbasis *Support Vector Machine* (SVM) dengan pendekatan *Natural Language Processing* (NLP) untuk deteksi *phishing*.

Gambar 10. Distirbusi Label

Berdasarkan distribusi label, diketahui bahwa dataset ini terdiri atas 134.850 entri *phishing* (57,2%) dan 100.945 entri *legitimate* (42,8%). Komposisi ini menunjukkan bahwa dataset relatif seimbang, sehingga mendukung proses pelatihan model klasifikasi secara optimal dan mengurangi risiko bias terhadap salah satu kelas.

Keseimbangan jumlah data pada kedua kelas ini penting untuk menjaga kinerja model agar tidak bias terhadap salah satu kategori. Jika dataset terlalu didominasi oleh salah satu kelas, model cenderung akan menghasilkan prediksi yang berat sebelah. Dengan distribusi yang proporsional seperti pada Gambar 10, proses pelatihan model *Support Vector Machine* (SVM) diharapkan dapat berjalan lebih stabil dan menghasilkan kemampuan klasifikasi yang akurat untuk mendeteksi URL *phishing* maupun *legitimate*.

Dataset ini dipilih karena ukurannya besar, memiliki berbagai macam fitur yang lengkap, dan sesuai dengan topik penelitian deteksi *phishing* URL. Dengan adanya banyak atribut yang tersedia, model bisa dilatih untuk memahami pola-pola yang membedakan URL *phishing* dari URL yang aman secara lebih mendalam. Pendekatan yang digunakan mencakup *Natural Language Processing* (NLP) dan algoritma *Support Vector Machine* (SVM).

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

3.2. Hasil Pra – Pemrosesan Data

Pra-pemrosesan data dilakukan agar data bisa digunakan oleh algoritma *Support Vector Machine* (SVM). Tahap ini bertujuan membersihkan data, membuat format data sama, serta mengubah variabel yang berbentuk kategori menjadi angka agar model bisa belajar menemukan pola dengan baik.

Berdasarkan pemeriksaan terhadap PhiUSIIL_PhishingURL Dataset, tidak ada nilai yang kosong di semua kolom, sehingga tidak diperlukan proses pengisiannya. Pra-pemrosesan dilakukan dengan langkah-langkah berikut:

1. Pemisahan Fitur dan Label Target

Kolom label dipisahkan dari variabel prediktor. Nilai 0 menunjukkan URL *legitimate*, sedangkan nilai 1 menunjukkan URL *phishing*. Proses ini diperlukan agar model dapat belajar memetakan fitur URL terhadap kategori target.

```
# Pisahkan fitur dan label
X = df.drop(['label', 'url', 'Domain', 'TLD', 'Title', 'domain_age'], axis=1)
y = df['label']
```

Gambar 11. Pemisahan Fitur dan Label

2. Encoding Variabel Kategorikal

Kolom TLD dan Domain yang sebelumnya bertipe kategorikal diubah menjadi representasi numerik menggunakan teknik label encoding. Contoh hasil *encoding* dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Contoh Encoding Kolom TLD

TLD Asli	TLD_Encode
.com	1
.org	2
.co.id	3

Tabel 1 memperlihatkan hasil proses label encoding pada variabel kategorikal Top Level Domain (TLD). Pada tahap ini, setiap nilai unik pada kolom TLD yang semula berupa data teks, seperti .com, .org, dan .co.id, dikonversi menjadi representasi numerik agar dapat dikenali oleh model pembelajaran mesin.

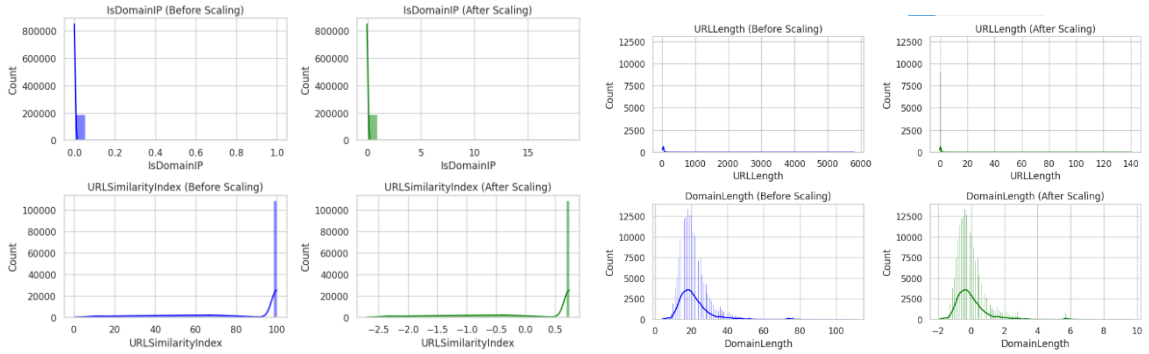
Berdasarkan Tabel 1, domain dengan ekstensi .com direpresentasikan dengan nilai 1, .org dengan nilai 2, dan .co.id dengan nilai 3. Proses ini dilakukan menggunakan metode LabelEncoder dari pustaka scikit-learn. Dengan demikian, seluruh variabel kategorikal di dalam dataset dapat diproses dalam format numerik tanpa kehilangan makna semantik aslinya.

Teknik *encoding* ini penting karena algoritma Support Vector Machine (SVM) hanya dapat menerima input numerik. Oleh sebab itu, konversi semacam ini memastikan bahwa model mampu mengenali pola antar domain berdasarkan nilai representatif yang telah disesuaikan.

3. Normalisasi Fitur Numerik

Fitur-fitur numerik seperti *URLLength*, *DomainLength*, *IsDomainIP*, *URLSimilarity* dan *ChartContinuationRate* dinormalisasi menggunakan metode StandardScaler agar berada pada rentang yang seragam. Normalisasi ini penting bagi SVM karena membantu membentuk *hyperplane* pemisah yang optimal. Berikut hasil dari Normalisasi fitur numerik dibawah ini.

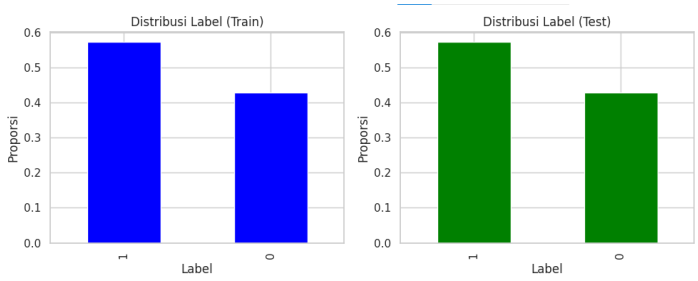
Jurnal Teknik Informatika (JUTIF)
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334



Gambar 12. Normalisasi Fitur Numerik

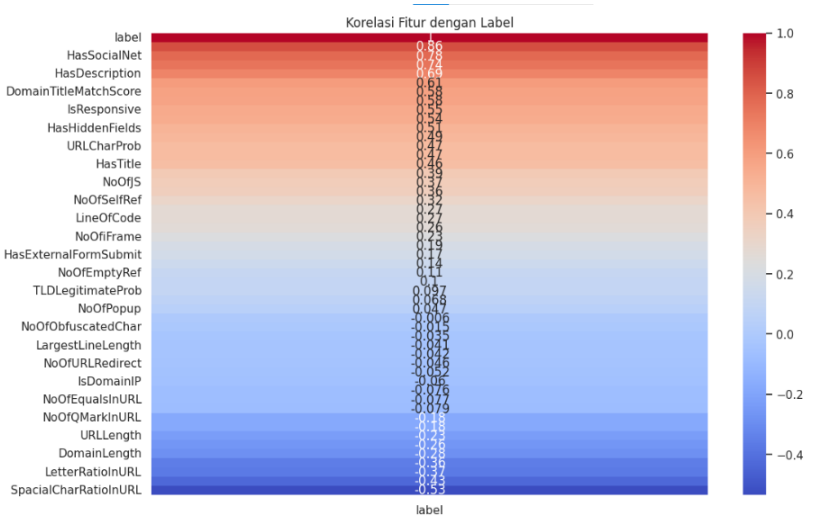
4. Pembagian Data Latih dan Data Uji

Dataset dibagi menjadi 80% data latih (*train data*) dan 20% data uji (*test data*). Data latih digunakan untuk melatih model *Support Vector Machine* (SVM), sedangkan data uji digunakan untuk mengukur kinerja model secara objektif. Berikut hasil visual data pembagian data latih dan data uji.



Gambar 13. Distribusi Label untuk Data Train dan Test

Selain keempat langkah utama tersebut, dilakukan pula analisis korelasi antar fitur dengan label target sebagai langkah penting untuk mengidentifikasi variabel-variabel yang paling berpengaruh dalam proses klasifikasi URL *phishing*.



Gambar 14. Korelasi Fitur dengan Label

Analisis korelasi ini tidak hanya berfungsi sebagai eksplorasi awal, tetapi juga sebagai dasar ilmiah untuk memahami kontribusi setiap fitur terhadap kinerja model. Berdasarkan hasil perhitungan matriks korelasi, fitur-fitur seperti *URLLength*, *DomainLength*, *TLDLegitimateProb*, dan *NoOfSubDomain* menunjukkan tingkat korelasi yang relatif lebih kuat dengan label target dibandingkan fitur lainnya. Temuan ini memberikan indikasi jelas bahwa atribut-atribut tersebut berperan signifikan dalam membedakan URL *phishing* dan URL *legitimate*, serta menjadi landasan dalam pemilihan dan optimasi fitur pada model *Support Vector Machine* yang digunakan.

Melalui proses pra-pemrosesan ini, data yang awalnya berisi berbagai jenis variabel, baik yang berupa angka maupun kategori, diubah menjadi bentuk angka yang rapi dan seragam. Dengan demikian, model Su*pport Vector Machine* bisa menggunakan semua fitur dari URL secara efektif untuk mengenali pola yang membedakan antara URL *phishing* dan URL yang *legitimate*.

3.3. Hasil Perancangan dan Pelatihan Model

Tahap perancangan dan pelatihan model dilakukan setelah proses pra-pemrosesan data selesai. Pada tahap ini, algoritma *Support Vector Machine* (SVM) diimplementasikan untuk melakukan klasifikasi URL *phishing* dan URL *legitimate*. Pemilihan algoritma SVM didasarkan pada kemampuannya dalam menangani data berdimensi tinggi dan menghasilkan batas pemisah (*hyperplane*) yang optimal antara kelas yang berbeda.

Proses pelatihan dilakukan dengan membagi dataset menjadi 80% data latih dan 20% data uji sesuai hasil pra-pemrosesan. Data latih digunakan untuk mengoptimalkan parameter model, sedangkan data uji digunakan untuk mengevaluasi kinerja model secara objektif.

Selama proses pelatihan, dilakukan juga pencarian parameter terbaik (hyperparameter tuning) dengan pendekatan *grid search* untuk memperoleh kombinasi parameter kernel, C, dan gamma yang menghasilkan performa optimal. Dari hasil pencarian parameter ini diperoleh model SVM dengan kombinasi parameter yang memberikan akurasi dan F1-*score* tertinggi.

```
Akurasi Model: 0.9999151805593842

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     20189
           1       1.00      1.00      1.00     26970

    accuracy                           1.00     47159
   macro avg       1.00      1.00      1.00     47159
weighted avg       1.00      1.00      1.00     47159
```

Gambar 15. Laporan Klasifikasi Model

Gambar 15 menunjukkan hasil evaluasi performa model Support Vector Machine (SVM) dalam mendeteksi URL *phishing* dan *legitimate*. Berdasarkan hasil pengujian, model menghasilkan nilai akurasi sebesar 0.9999 (99,99%), dengan nilai *precision*, *recall*, dan *f1-score* yang masing-masing mencapai 1.00 pada kedua kelas (label 0 dan 1).

Nilai *precision* sebesar 1.00 menunjukkan bahwa seluruh prediksi positif yang dilakukan oleh model benar-benar tergolong *phishing*, tanpa adanya kesalahan klasifikasi terhadap URL legitimate (false positive = 0). Demikian pula, nilai recall sebesar 1.00 menandakan bahwa seluruh URL phishing berhasil terdeteksi secara benar oleh model (*false negative* = 0). Dengan demikian, nilai *f1-score* yang juga mencapai 1.00 memperlihatkan keseimbangan sempurna antara presisi dan sensitivitas model.

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

Secara keseluruhan, hasil ini menunjukkan bahwa model SVM yang diimplementasikan memiliki kinerja yang sangat baik dalam membedakan situs *phishing* dan *legitimate*. Akurasi yang mendekati sempurna mengindikasikan bahwa kombinasi antara fitur numerik, fitur linguistik hasil ekstraksi *Natural Language Processing* (NLP), serta optimasi parameter SVM melalui *Grid Search* berhasil meningkatkan kemampuan klasifikasi model secara signifikan.

Tabel 2. Parameter Model SVM

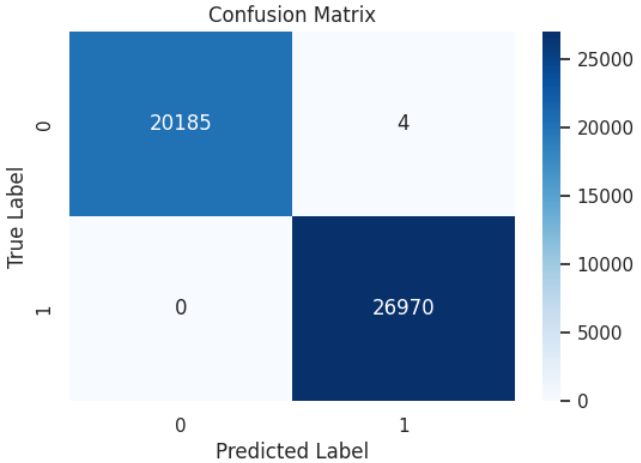
Parameter	Nilai	Keterangan
Kernel	RBF	Menangkap pola non-linear
C	1.0	Regularisasi margin
Gamma	scale	Penyesuaian otomatis

Tabel tersebut merangkum parameter utama yang digunakan dalam model pembelajaran mesin, khususnya untuk metode *Support Vector Machine* (SVM) dengan kernel RBF. Parameter "Kernel" diatur sebagai RBF, yang memungkinkan model untuk menangkap pola non-linear dalam data. Parameter "C" ditetapkan pada nilai 1.0, yang mengontrol tingkat regularisasi margin, menjaga keseimbangan antara margin maksimum dan klasifikasi yang benar. Sementara itu, parameter "*Gamma*" menggunakan nilai "*scale*", yang memungkinkan penyesuaian otomatis berdasarkan data untuk mengoptimalkan performa model. Kombinasi parameter ini dirancang untuk memastikan fleksibilitas dan akurasi dalam menangani dataset yang kompleks.

Secara umum, hasil pelatihan menunjukkan bahwa model SVM mampu mempelajari pola karakteristik URL phishing dengan baik. Model ini memanfaatkan fitur struktural, probabilistik, dan reputasi domain yang telah dinormalisasi dan di-*encode* pada tahap sebelumnya, sehingga menghasilkan pembeda yang jelas antara URL *phishing* dan URL *legitimate*.

3.4. Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan untuk menilai seberapa baik algoritma *Support Vector Machine* (SVM) yang telah dilatih dalam mendeteksi URL *phishing* dan URL *legitimate*. Evaluasi dilakukan menggunakan data uji sebesar 20% dari total dataset, sehingga hasil yang diperoleh merefleksikan kemampuan model pada data yang belum pernah dilihat sebelumnya.



Gambar 16. *Confusion Matrix*

Proses evaluasi dilakukan berdasarkan confusion matrix yang terdiri dari empat komponen utama yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Hasil

prediksi model pada data uji disajikan dalam bentuk *confusion matrix* seperti pada Gambar 16 *Confusion matri*x terdiri atas empat komponen:

a. True Positive (TP): URL phishing yang terdeteksi dengan benar sebagai phishing.
b. True Negative (TN): URL legitimate yang terdeteksi dengan benar sebagai legitimate.
c. False Positive (FP): URL legitimate yang salah terdeteksi sebagai phishing.
d. False Negative (FN): URL phishing yang salah terdeteksi sebagai legitimate.

Dengan memasukkan nilai-nilai ini ke dalam rumus evaluasi yang telah dijelaskan pada 2.4, diperoleh metrik kinerja sebagai berikut:

a. Akurasi (Accuracy) mengukur persentase prediksi yang benar terhadap keseluruhan data uji.
b. Presisi (Precision) mengukur proporsi prediksi phishing yang benar-benar phishing.
c. Recall (Sensitivity) mengukur seberapa banyak URL phishing yang berhasil terdeteksi.
d. F1-score menggabungkan presisi dan recall menjadi satu metrik harmonis.

Berdasarkan hasil pengujian terhadap data uji diperoleh nilai confusion matrix sebagai berikut.

Tabel 3. Hasil Perhitungan Matrix

Komponen	Nilai
TP	26.970
TN	20.185
FP	4
FN	0

Tabel 3 menunjukkan hasil perhitungan confusion matrix dari model Support Vector Machine (SVM) pada tahap evaluasi. Berdasarkan hasil pengujian terhadap data uji, diperoleh nilai True Positive (TP) sebesar 26.970, yang menunjukkan bahwa seluruh URL phishing berhasil terklasifikasi dengan benar sebagai phishing oleh model. Nilai True Negative (TN) sebesar 20.185 menandakan bahwa sebagian besar URL legitimate juga berhasil dikenali dengan benar.

Sementara itu, terdapat 4 kasus False Positive (FP), yakni URL legitimate yang secara keliru diklasifikasikan sebagai phishing, dan tidak ditemukan kasus False Negative (FN = 0), yang berarti tidak ada URL phishing yang salah dikategorikan sebagai legitimate. Dengan demikian, model menunjukkan kemampuan deteksi phishing yang sangat tinggi, dengan tingkat kesalahan klasifikasi yang dapat diabaikan.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa model SVM mampu memberikan performa klasifikasi yang optimal. Nilai TP yang tinggi serta FN yang bernilai nol membuktikan bahwa sistem memiliki tingkat sensitivitas (recall) yang sempurna dalam mengenali situs phishing, sedangkan jumlah FP yang sangat kecil menunjukkan bahwa presisi model juga sangat baik. Kombinasi kedua faktor ini memperkuat validitas hasil evaluasi yang sebelumnya ditunjukkan pada classification report dengan akurasi keseluruhan mencapai 99,99%.

Berdasarkan *confusion matrix* pada Gambar 16 diperoleh nilai seperti Tabel 3 seperti di atas, perhitungan metrik evaluasi dilakukan secara manual menggunakan rumus pada 2.4 sebagai berikut:

a. Akurasi (accuracy)

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (6)$$

$$\text{Accuracy} = \frac{26.970 + 20.185}{26.970 + 20.185 + 4 + 0} = 99{,}99\%$$

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

b. Presisi (Precision)

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (7)$$

$$\text{Precision} = \frac{26.970}{26.970 + 4}$$

$$\text{Precision} = \frac{26.970}{26.974} = 0.99985$$

c. Recall (Sensivity)

$$Recall = \frac{TP}{TP+FN} \quad (8)$$

$$Recall = \frac{26.970}{26.970 + 0}$$

$$Recall = \frac{26.970}{26.970} = 1{,}000$$

d. F1- Score

$$F1 - Score = \frac{2 \times 0.99985 \times 1.000}{0.99985 + 1.000}$$

$$F1 - Score = \frac{1.9997}{1.99985} = 0.99992 \ (\approx 99.99\%)$$

Berikut hasil perhitungan manual dari *confusion matrix* :

Table 4. Hasil Perhitungan Confusion Matrix

Metrik Evaluasi	Nilai (%)
Accuracy	99,99 %
Precision	99.98 %
Recall	100 %
F1- *Score*	99,99 %

4. DISKUSI

Hasil evaluasi model yang telah dipaparkan pada Bab 3 menunjukkan bahwa algoritma Support Vector Machine (SVM) berbasis Natural Language Processing (NLP) memiliki performa yang sangat baik dalam mendeteksi URL *phishing*. Berdasarkan hasil *confusion matrix*, diperoleh nilai *True Positive* (TP) = 26.970, *True Negative* (TN) = 20.185, False Positive (FP) = 4, dan False Negative (FN) = 0. Dengan menggunakan rumus evaluasi performa, diperoleh metrik *Accuracy* = 99,99%, *Precision* = 99,98%, *Recall* = 100%, dan *F1-score* = 99,99%. Nilai akurasi yang sangat tinggi tersebut menandakan bahwa model memiliki tingkat kesalahan yang sangat rendah dalam mengklasifikasikan URL *phishing* dan *legitimate*, sementara nilai *recall* yang mencapai 100% menunjukkan kemampuan model dalam mendeteksi seluruh URL *phishing* tanpa ada yang terlewat.

Nilai akurasi yang sangat tinggi menunjukkan bahwa model mampu mengklasifikasikan URL *phishing* dan URL *legitimate* dengan tingkat kesalahan yang sangat rendah. Nilai *recall* 100% menandakan bahwa tidak ada URL *phishing* yang lolos sebagai URL *legitimate*, sehingga model sangat andal dalam mendeteksi *phishing*. Sementara itu, nilai *precision* 99,98% menunjukkan bahwa model

jarang salah mendeteksi URL *legitimate* sebagai *phishing*, yang berarti potensi *false alarm* sangat kecil. Kombinasi presisi dan *recall* yang seimbang juga tercermin pada nilai F1-*score* 99,99%, yang mengindikasikan stabilitas model dalam mengatasi kedua metrik tersebut.

Jika dibandingkan dengan beberapa penelitian sebelumnya, hasil ini menunjukkan peningkatan yang signifikan. Penelitian oleh Anand et al. [6] menggunakan SVM dengan fitur statis dan menghasilkan akurasi sebesar 96,8%. Selanjutnya, Tamal et al. [5] melaporkan nilai *recall* 95% pada pendekatan berbasis feature engineering. Penelitian Zamir et al. [4] yang menggunakan Deep Neural Network hanya mencapai akurasi 98,4%, sedangkan Catal et al. [7] dengan metode Random Forest memperoleh 97,3%. Dalam penelitian ini, integrasi Natural Language Processing (NLP) pada tahap ekstraksi fitur serta optimasi parameter SVM melalui *Grid Search* berhasil meningkatkan performa secara signifikan hingga mencapai 99,99%. Hal ini membuktikan bahwa pendekatan NLP mampu menangkap pola linguistik yang lebih kompleks pada URL, seperti tokenisasi kata “login”, “secure”, atau “verify”, yang sering muncul pada situs phishing.

Selain itu, hasil ini sejalan dengan temuan dari Li et al. [27] yang menunjukkan bahwa pendekatan NLP–ML menghasilkan peningkatan recall dibandingkan model berbasis numerik murni, serta Aljofey et al. [28] yang melaporkan akurasi 99,1% melalui integrasi NLP dan ensemble learning. Namun, penelitian ini mampu melampaui kedua studi tersebut berkat kombinasi proses normalisasi data, label encoding variabel kategorikal, serta parameter tuning kernel RBF pada SVM yang secara efektif memisahkan kelas phishing dan legitimate secara non-linear.

Dari sisi ukuran dan karakteristik data, penelitian ini juga diuntungkan oleh penggunaan dataset besar dan relatif seimbang yang terdiri dari 256.795 URL. Kondisi ini memungkinkan model untuk belajar pola distribusi fitur dengan lebih representatif dibandingkan studi sebelumnya yang hanya menggunakan kurang dari 50.000 sampel [5][6]. Proses pra-pemrosesan seperti feature scaling dan handling imbalance turut membantu SVM membentuk *hyperplane* optimal yang memaksimalkan margin antar kelas, sebagaimana dijelaskan dalam teori SVM klasik oleh Cortes dan Vapnik.

Dari hasil perbandingan kuantitatif dan analisis performa tersebut, dapat disimpulkan bahwa model SVM berbasis NLP memiliki tingkat efektivitas tinggi dalam mendeteksi URL *phishing* dibandingkan metode klasik maupun *deep learning* dengan kompleksitas tinggi. Keunggulan ini tidak hanya terlihat dari nilai metrik yang mendekati sempurna, tetapi juga dari efisiensi waktu komputasi yang relatif ringan dibandingkan model *neural network* yang memerlukan sumber daya besar.

Penelitian ini memberikan kontribusi penting terhadap pengembangan sistem keamanan siber modern, khususnya dalam bidang deteksi phishing berbasis URL. Pendekatan yang diusulkan dapat diimplementasikan secara langsung pada ekstensi peramban (*browser extension*) atau sistem keamanan jaringan untuk melakukan deteksi phishing secara *real-time*. Dari sisi akademik, hasil penelitian ini memperkuat teori bahwa kombinasi NLP dan SVM mampu meningkatkan performa klasifikasi teks non-linier secara signifikan. Temuan ini juga membuka peluang penelitian lanjutan, seperti integrasi fitur reputasi domain (umur domain, SSL validity, *web traffic*) dan eksplorasi *model transformer-based* untuk meningkatkan ketahanan sistem terhadap pola serangan phishing yang terus berkembang.

5. KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan sistem deteksi URL *phishing* berbasis *Natural Language Processing* (NLP) dan *Support Vector Machine* (SVM). Dengan memanfaatkan dataset besar berisi 256.795 baris data URL *phishing* dan URL *legitimate* yang diperoleh dari Kaggle, sistem ini melalui tahapan lengkap mulai dari pra-pemrosesan data, perancangan model, pelatihan, hingga evaluasi kinerja. Proses pra-pemrosesan yang komprehensif (pemilahan fitur dan label, encoding variabel kategorikal, normalisasi fitur numerik, pembagian data latih-uji, dan analisis korelasi fitur)

terbukti meningkatkan kualitas input yang diterima oleh model SVM sehingga model mampu mengenali pola URL secara lebih presisi.

Hasil pelatihan menunjukkan bahwa algoritma SVM dengan kernel RBF mampu memisahkan URL *phishing* dari URL *legitimate* dengan *hyperplane* yang optimal. Model ini memanfaatkan kombinasi fitur struktural, probabilistik, dan reputasi domain sehingga menghasilkan pembeda yang jelas antara kedua kelas. Berdasarkan evaluasi pada data uji (20% dari total dataset), model mencapai performa yang sangat tinggi dengan accuracy 99,99%, precision 99,98%, recall 100%, dan F1-score 99,99%. Nilai recall yang sempurna menegaskan bahwa seluruh URL *phishing* berhasil terdeteksi dengan benar tanpa terlewat, sedangkan nilai precision yang hampir sempurna menunjukkan tingkat false alarm yang sangat rendah.

Temuan ini mengindikasikan bahwa pendekatan berbasis NLP dan SVM efektif dalam mendeteksi URL phishing secara otomatis dan dapat dijadikan dasar untuk pengembangan sistem keamanan berbasis browser extension atau aplikasi keamanan siber. Model ini juga memperlihatkan bahwa pemilihan fitur yang tepat, normalisasi, serta optimasi parameter SVM berperan besar dalam meningkatkan kinerja model. Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi Natural Language Processing (NLP) dan Support Vector Machine (SVM) mampu menjadi alternatif yang efektif, adaptif, dan akurat untuk deteksi phishing berbasis URL, sehingga dapat mendukung upaya pencegahan kejahatan siber dan meningkatkan keamanan pengguna di dunia maya.

Sebagai saran untuk penelitian selanjutnya, model ini dapat dikembangkan lebih lanjut dengan memperluas sumber data agar mencakup URL *phishing* terbaru dan multibahasa untuk meningkatkan generalisasi model. Penelitian lanjutan juga dapat mempertimbangkan penggunaan algoritma berbasis deep learning seperti BERT atau Transformer untuk mengekstraksi konteks linguistik URL secara lebih mendalam. Selain itu, penggabungan fitur reputasi domain seperti umur domain, validitas sertifikat SSL, dan peringkat lalu lintas situs (*web traffic ranking*) diharapkan mampu memperkaya representasi fitur yang digunakan. Implementasi model dalam bentuk ekstensi browser atau sistem deteksi real-time di lingkungan jaringan juga menjadi arah pengembangan penting untuk menguji keandalan model dalam kondisi dunia nyata.

REFERENSI

[1] "PHISHING ACTIVITY TRENDS REPORT," 2025. [Online]. Available: http://www.apwg.org,

[2] "Phishing E-mail Reports and Phishing Site Trends 4 Brand-Domain Pairs Measurement 5 Brands & Legitimate Entities Hijacked by E-mail Phishing Attacks 6 Use of Domain Names for Phishing 7-9 Phishing and Identity Theft in Brazil 10-11 Most Targeted Industry Sectors 12 APWG Phishing Trends Report Contributors 13," 2024. [Online]. Available: http://www.apwg.org,

[3] NCSC, "Federal Department of Defense, Civil Protection and Sport DDPS National Cyber Security Centre NCSC Anti-Phishing Report 2024," 2025.

[4] A. Zamir *et al.*, "Phishing web site detection using diverse machine learning algorithms," *Electronic Library*, vol. 38, no. 1, pp. 65–80, Mar. 2020, doi: 10.1108/EL-05-2019-0118.

[5] S. Saeed, L. Fotia, M. Jabed, M. Chowdhury, and M. A. Tamal, "Dataset of suspicious phishing URL detection," 2024, doi: 10.17632/6tm2d6sz7p.1.

[6] A. Anand, A. Gupta, A. Dubey, and C. Naidu, "Phishing Site Detection Using ML Algorithms," 2024. [Online]. Available: www.ijfmr.com

[7] Catal & Giray, "Applications of deep learning for phishing detection: a systematic literature review ," 2022.

[8] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

[9] D. Kalla and S. Kuraku, "Phishing Website URL's Detection Using NLP and Machine Learning Techniques," *Journal on Artificial Intelligence*, vol. 5, no. 0, pp. 145–162, 2023, doi: 10.32604/jai.2023.043366.

[10] M. A. Uddin, M. Mahiuddin, and I. H. Sarker, "An Explainable Transformer-based Model for Phishing Email Detection: A Large Language Model Approach," Aug. 2025, [Online]. Available: http://arxiv.org/abs/2402.13871

[11] Q. E. ul Haq, M. H. Faheem, and I. Ahmad, "Detecting Phishing URLs Based on a Deep Learning Approach to Prevent Cyber-Attacks," *Applied Sciences (Switzerland)*, vol. 14, no. 22, Nov. 2024, doi: 10.3390/app142210086.

[12] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, "Applications of deep learning for phishing detection: a systematic literature review," *Knowl Inf Syst*, vol. 64, no. 6, pp. 1457–1500, Jun. 2022, doi: 10.1007/s10115-022-01672-x.

[13] R. N. F. Tanjung and S. Rahman, "Meningkatkan Deteksi Email Phising Melalui Pendekatan SVM yang Dioptimalkan NLP," *INCODING: Journal of Informatics and Computer Science Engineering*, vol. 5, no. 1, pp. 38–50, Apr. 2025, doi: 10.34007/incoding.v5i1.831.

[14] E. S. Aung and H. Yamana, "PhiSN: Phishing URL Detection Using Segmentation and NLP Features," *Journal of Information Processing*, vol. 32, pp. 973–989, 2024, doi: 10.2197/ipsjjip.32.973.

[15] A. Kumar, J. M. Chatterjee, and V. G. Díaz, "A novel hybrid approach of SVM combined with NLP and probabilistic neural network for email phishing," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 486–493, 2020, doi: 10.11591/ijece.v10i1.pp486-493.

[16] R. Luthfiansyah and B. Wasito, "Penerapan Teknik Deep Learning (Long Short Term Memory) dan Pendekatan Klasik (Regresi Linier) dalam Prediksi Pergerakan Saham BRI," 2023.

[17] Nailah Azzahra, Merry Dwi Handayani, and Awwaliyah Aliyah, "Evaluasi Kinerja AI berbasis Recurrent Neural Network (RNN) dalam Mengidentifikasi Ancaman Phising pada URL Website," *Bridge : Jurnal Publikasi Sistem Informasi dan Telekomunikasi*, vol. 3, no. 3, pp. 15–37, Jun. 2025, doi: 10.62951/bridge.v3i3.485.

[18] E. S. Shombot, G. Dusserre, R. Bestak, and N. B. Ahmed, "An application for predicting phishing attacks: A case of implementing a support vector machine learning model," *Cyber Security and Applications*, vol. 2, Jan. 2024, doi: 10.1016/j.csa.2024.100036.

[19] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 2, pp. 590–611, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.004.

[20] M. Abdolrazzagh-Nezhad and N. Langarib, "Phishing Detection Techniques: A review," *Data Science: Journal of Computing and Applied Informatics*, vol. 9, no. 1, pp. 32–46, Jan. 2025, doi: 10.32734/jocai.v9.i1-19904.

[21] A. Prasad and S. Chandra., "PhiUSIIL Phishing URL," UCI Machine Learning Repository.

[22] A. D. Prastiko and A. Davy Wiranata, "Analisis Sentimen Publik terhadap Fenomena Judi Online di Media Sosial X dengan SVM," *Andika Dwi Prastiko*, vol. 1, no. 2, pp. 306–315, 2025, doi: 10.55382/jurnalpustakaai.v5i2.1180.

[23] D. Fitriono, S. A. Wardani, M. Nizar, B. Al Varuq, A. Ristyawan, and E. Daniati, "Perbandingan Metode Algoritma Decission Tree dan K-Nearest Neighbors untuk Memprediksi Kualitas Air yang dapat dikonsumsi," Online, 2024.

[24] D. Sangaji and T. Sutabri, "Analisis XGBoost dan Random Forest untuk Prediksi Curah Hujan dalam Mendukung Mitigasi Karhutla," *Jurnal Pustaka AI (Pusat Akses Kajian Teknologi Artificial Intelligence)*, vol. 5, no. 1, pp. 13–18, Apr. 2025, doi: 10.55382/jurnalpustakaai.v5i1.905.

[25] V. Pramaningsih, R. Yuliawati, S. Sukisman, H. Hansen, R. Suhelmi, and A. Daramusseng, "Indek Kualitas Air dan Dampak terhadap Kesehatan Masyarakat Sekitar Sungai Karang Mumus, Samarinda," *Jurnal Kesehatan Lingkungan Indonesia*, vol. 22, no. 3, pp. 313–319, Oct. 2023, doi: 10.14710/jkli.22.3.313-319.

Vol. 7, No. 1, February 2026, Page. 552-570
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2026.7.1.5334

[26] Rakesh Jampa, "PhiUSIIL_Phishing_URL_Dataset," Kaggle. Accessed: Sep. 15, 2025. [Online]. Available: https://www.kaggle.com/datasets/rakeshjampa/phiusiil-phishing-url-dataset/data

[27] S. Asiri, Y. Xiao, and T. Li, "PhishTransformer: A Novel Approach to Detect Phishing Attacks Using URL Collection and Transformer," *Electronics (Switzerland)*, vol. 13, no. 1, Jan. 2024, doi: 10.3390/electronics13010030.

[28] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J. P. Niyigena, "An effective phishing detection model based on character level convolutional neural network from URL," *Electronics (Switzerland)*, vol. 9, no. 9, pp. 1–24, Sep. 2020, doi: 10.3390/electronics9091514