JURNAL RESISTOR | ISSN 2598-7542 | E-ISSN 2598-9650
Vol. 3 No 1 – April 2020 | https://s.id/jurnalresistor
DOI: https://doi.org/10.31598
Publishing: Prahasta Publisher

Deteksi URL Phishing Menggunakan Algoritma Support Vector Machine Berbasis Website

Aryanti Aryanti[1*], Nabila Nabila[2]

[1,2]Program Studi Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Sriwijaya
Jl. Srijaya Negara, Bukit Besar, Ilir Barat I, Kota Palembang, Sumatera Selatan, Indonesia

e-mail: aryanti@polsri.ac.id[1], nabilabila2934@gmail.comi[2]

Received : July, 2025	Accepted : August, 2025	Published : August, 2025

Abstract

The increasing incidence of phishing attacks through web links poses a significant threat to cybersecurity. The main challenge is the lack of detection systems capable of reliably classifying phishing URLs by leveraging both URL structural features and textual representations. This study proposes a web-based phishing URL detection system using the Support Vector Machine (SVM) algorithm with URL textual representation based on Natural Language Processing using Term Frequency-Inverse Document Frequency (TF-IDF). The dataset consists of 11,430 URL entries, evenly distributed between phishing and legitimate URLs. Features are extracted from URL structures and textual representations, and several combinations of kernels and text representation methods are evaluated. Experimental results show that the SVM linear kernel combined with TF-IDF achieves the highest accuracy of 91% (classification report), an average cross-validation of 0.9055, 91.6% accuracy on fold 3, and a confusion matrix of 91%. Computation times are efficient, with 18.76 seconds for training, 0.0072 seconds for batch prediction, and 3e-06 seconds per URL. The Flask-based system allows users to test URLs directly and has been verified through black-box testing, demonstrating good performance. These findings confirm the effectiveness of the proposed approach and its potential for developing more comprehensive cybersecurity systems.

Keywords: *Natural Language Processing, Phishing, Support Vector Machine, TF-IDF, Website.*

Abstrak

Meningkatnya serangan phishing melalui tautan web menimbulkan ancaman signifikan terhadap keamanan siber. Masalah utama adalah minimnya sistem deteksi yang mampu mengklasifikasikan URL phishing secara andal dengan memanfaatkan fitur struktural URL dan representasi tekstualnya. Penelitian ini mengusulkan solusi berupa sistem deteksi phishing URL berbasis web menggunakan algoritma Support Vector Machine (SVM) dengan representasi tekstual URL berbasis Natural Language Processing menggunakan Term Frequency-Inverse Document Frequency (TF-IDF). Dataset terdiri dari 11.430 entri URL, terdistribusi merata antara phishing dan legitimate. Fitur diekstraksi dari struktur URL dan representasi tekstual, kemudian beberapa kombinasi kernel dan metode representasi teks dievaluasi. Hasil eksperimen menunjukkan bahwa SVM kernel linier dengan TF-IDF mencapai akurasi tertinggi 91% (classification report), cross-validation rata-rata 0,9055, akurasi pada fold 3 sebesar 91,6%, dan nilai confusion matrix 91%. Waktu komputasi juga efisien, dengan pelatihan 18,76 detik, prediksi batch 0,0072 detik, dan prediksi per URL 3e-06 detik. Sistem berbasis Flask memungkinkan pengguna menguji URL secara langsung dan telah diuji melalui black-box testing, menunjukkan kinerja yang baik. Temuan ini menegaskan efektivitas pendekatan yang diusulkan dan potensi pengembangannya untuk sistem keamanan siber yang lebih komprehensif.

Kata Kunci: *Natural Language Processing, Phishing, Support Vector Machine, TF-IDF, Website*

1. PENDAHULUAN

Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) memperkirakan bahwa pada tahun 2024, Indonesia akan memiliki sekitar 221.563.479 pengguna internet dari total populasi 278.696.200 jiwa pada tahun 2023. Survei Penetrasi Internet Indonesia 2024 yang diterbitkan oleh APJII menunjukkan bahwa tingkat penetrasi internet nasional naik sebesar 1,4% dari sebelumnya, sekarang mencapai 79,5%[1].Namun, pertumbuhan penggunaan internet yang pesat ini turut memunculkan kekhawatiran terkait isu keamanan siber, khususnya ancaman serangan *phishing* yang semakin sering terjadi[2].

Merujuk pada laporan IDADX untuk kuartal keempat tahun 2024, serangan *phishing* menjadi jenis insiden yang paling banyak dilaporkan, dengan total mencapai 85.414 kasus[3]. Meskipun teknologi keamanan terus berkembang, *phishing* tetap menjadi ancaman yang efektif karena memanfaatkan faktor kelemahan manusia, seperti kurangnya kewaspadaan serta rendahnya tingkat literasi digital[4].

Phishing termasuk dalam tindakan kriminal yang dilakukan dengan memanipulasi perilaku atau kepercayaan seseorang[5]. *Phishing* merupakan ancaman yang dapat menimpa semua kalangan, mulai dari pengguna individu hingga entitas bisnis besar, dengan konsekuensi yang sangat merugikan[6]. Oleh karena itu, penerapan sistem pendeteksi URL *phishing* sangat penting untuk melindungi mereka dari ancaman siber yang terus berkembang[7].

Pendekatan proaktif menjadi sangat krusial dalam mendeteksi URL *phishing* secara cepat dan akurat guna mencegah penyebaran dampak yang lebih luas. Seiring berkembangnya teknologi, konsep *Machine Learning* (ML) mulai diadopsi sebagai salah satu metode dalam bidang Kecerdasan Buatan (AI) yang dirancang untuk meniru cara manusia dalam menyelesaikan berbagai permasalahan. *Machine learning* telah terbukti efektif di berbagai bidang, terutama dalam mengidentifikasi dan menganalisis ancaman siber[8]. Melalui penerapan metode *machine learning*, dapat dikembangkan model klasifikasi yang mampu mengenali pola serta karakteristik dari situs web *phishing*, sehingga memungkinkan sistem untuk membedakan secara akurat antara situs yang sah dan situs yang berpotensi melakukan penipuan[8].

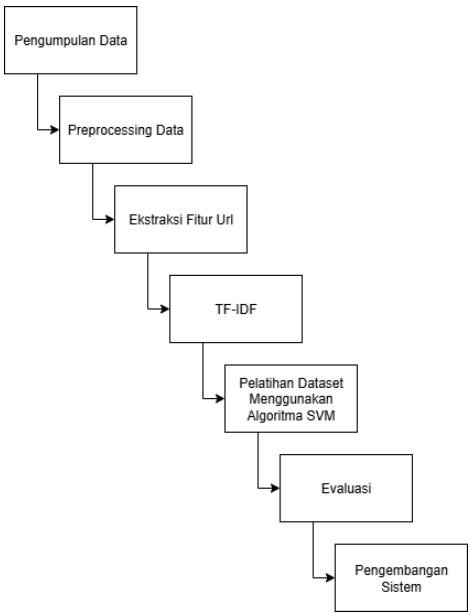
Penelitian [9] membandingkan algoritma *Naive Bayes* dan *Support Vector Machine* menunjukkan bahwa SVM unggul dalam seluruh *platform* yang diuji, dengan akurasi masing-masing sebesar 86,82% pada Tokopedia, 80,91% pada Shopee, dan 88,93% pada Lazada. Temuan ini menunjukkan bahwa SVM lebih andal dibandingkan *Naive Bayes* dalam klasifikasi sentimen berbasis teks pada data media sosial. Pada penelitian [10] menunjukkan bahwa kernel linier memberikan akurasi tertinggi, yaitu sebesar 92,68%. Hasil ini memperkuat temuan bahwa pemilihan jenis kernel pada SVM berperan penting dalam meningkatkan performa klasifikasi, khususnya dalam pengolahan data teks. Penelitian [11] menggunakan SVM dengan kernel linier untuk klasifikasi sentimen ulasan film IMDb. Kombinasi SVM dan TF-IDF menghasilkan akurasi tertinggi sebesar 89,55%, mengungguli metode fitur lain seperti BoW. Temuan ini menguatkan efektivitas TF-IDF dan SVM linier dalam analisis teks.

Penelitian oleh [12] menunjukkan bahwa algoritma *Decision Tree* mampu mengklasifikasikan situs *phishing* dengan akurasi 87,04%. Penelitian [13] menunjukkan bahwa *K-Nearest Neighbors* (KNN) berhasil mendeteksi situs *phishing* dengan akurasi 88%. Penelitian oleh [8] membandingkan *Decision Tree*, *Random Forest*, dan K-NN untuk deteksi situs *phishing*. *Decision Tree* meraih akurasi 83,3%, *Random Forest* 83,4%, sedangkan K-NN hanya 48,2%. Penelitian [14] membandingkan algoritma *Naïve Bayes* dan C4.5 untuk klasifikasi URL *phishing*. C4.5 menghasilkan akurasi lebih tinggi (87,11%) dibanding *Naïve Bayes* (78,48%).

Penelitian sebelumnya masih terbatas pada pemanfaatan fitur URL manual tanpa integrasi *Natural Language Processing* (NLP) maupun penerapan dalam *platform* berbasis web. Beberapa studi telah menggunakan algoritma seperti *Decision Tree, K-Nearest Neighbors*, dan Naïve Bayes dengan akurasi yang beragam, namun belum menggabungkan NLP berbasis TF-IDF untuk pengolahan data teks. Selain itu, sistem yang dapat diakses langsung melalui website juga belum tersedia. Penelitian ini hadir dengan pendekatan yang lebih komprehensif,

yaitu menggabungkan fitur struktural URL, NLP berbasis TF-IDF, dan algoritma SVM kernel linier yang sebelumnya terbukti memberikan performa tinggi. Seluruh metode ini diimplementasikan dalam website, sehingga menghasilkan solusi deteksi *phishing* URL yang praktis, efektif, dan mudah diakses.

2. METODE PENELITIAN



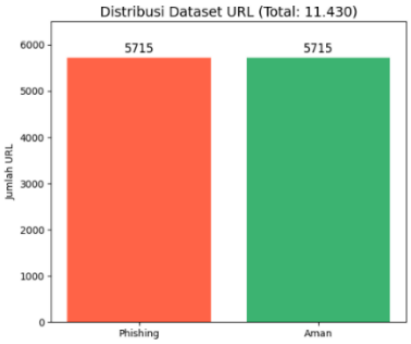
Gambar 1. Tahapan Penelitian

Penelitian ini melalui beberapa tahapan, dimulai dari pengumpulan dan *preprocessing* data URL, lalu dilanjutkan dengan ekstraksi fitur struktural dari URL. Data kemudian diubah ke bentuk numerik menggunakan TF-IDF, sebelum dilatih menggunakan algoritma SVM kernel linier. Model yang dihasilkan diintegrasikan ke dalam sistem berbasis web, lalu dievaluasi untuk mengukur tingkat akurasi dan kinerjanya dalam mendeteksi *phishing* URL.

Tujuan penelitian ini adalah mengembangkan sistem deteksi URL phishing berbasis web yang akurat, andal, dan efektif dalam melindungi data pribadi serta mengurangi risiko serangan siber khususnya bagi individu dan organisasi yang rentan.

2.1 Pengumpulan Data

Dataset yang digunakan berasal dari tahun 2021 dengan jumlah 11.430 entri yang terdistribusi seimbang antara *phishing* dan *legitimate* URL (masing-masing 5.715 data). Dataset ini dipilih karena ciri-ciri fundamental *phishing* URL, seperti panjang URL berlebihan, penggunaan karakter khusus, serta struktur domain tidak wajar, masih konsisten ditemukan hingga saat ini. Untuk menjaga kualitas data, dilakukan pemeriksaan duplikasi melalui peninjauan langsung terhadap keseluruhan entri, dan dari hasil pemeriksaan tersebut tidak ditemukan adanya data ganda (*duplicate*). Data kemudian dibagi menggunakan rasio 80:20, yaitu 80% untuk pelatihan dan 20% untuk pengujian, guna memastikan distribusi kelas tetap seimbang dan hasil evaluasi model lebih akurat serta representatif.



Gambar 2. Distribusi Data

Tabel 1. Sample Url

No	Url	Label
1.	https://www.google.co.id	*Legitimate*
2.	https://polarklimatsgserver.blogspot.com/	*Phishing*

2.2 *Natural Language Processing*

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang dirancang untuk mengembangkan sistem yang mampu memahami dan mengolah bahasa manusia secara alami[15]. Salah satu tahapan awal dalam NLP adalah *preprocessing* data, yang bertujuan untuk merapikan dan menyiapkan data dalam format yang sesuai sebelum memasuki proses pemrosesan utama[16]. Tahapan ini melibatkan beberapa proses, yaitu:

1. *Case Folding*, Yaitu Proses Mengubah Seluruh Huruf Dalam Teks Menjadi Huruf Kecil (*Lowercase*) Untuk Menyamakan Format Penulisan.
2. *Cleaning*, Yang Bertujuan Untuk Menghapus Elemen-Elemen Yang Tidak Relevan Seperti

Tanda Baca, Angka, Atau Karakter Khusus Agar Data Menjadi Lebih Bersih.

3. *Tokenization*, Yaitu Pemisahan Kalimat Menjadi Kata-Kata Tunggal Atau Token, Dengan Cara Mengenali Spasi Sebagai Pemisah Antar Kata, Sehingga Setiap Kata Dapat Diidentifikasi Secara Terpisah.

Tabel 2. Hasil *Preprocessing* Data

Url	*Casefolding*	*Cleaning*	Tokenization
https://www.google.co.id	https://www.google.co.id	www google co id	'www' 'google' 'co' 'id'
https://polarklimatsgserver.blogspot.com/	https://polarklimatsgserver.blogspot.com/	Polarklimatsgserver blogspot com	'Polarklimatsgserver' 'blogspot' 'com'

Selanjutnya yaitu ekstraksi fitur url, tahap ini bertujuan mengidentifikasi komponen penting dari URL untuk dijadikan fitur analisis. Penelitian ini menggunakan 21 atribut, seperti panjang URL, jumlah titik, dan karakter khusus lainnya. URL *phishing* umumnya memiliki struktur yang lebih panjang dan rumit dibandingkan dengan URL asli, sehingga dapat dijadikan indikator dalam proses deteksi. Pada tabel 3 merupakan hasil dari ekstraksi fitur url.

Tabel 3. Hasil Ekstraksi Fitur Url

Url	Panjang Url	Titik	Slash
https://www.google.co.id	24	3	2
https://polarklimatsgserver.blogspot.com/	41	2	3

Selanjutnya yaitu masuk ke metode TF-IDF yang digunakan untuk mengubah teks menjadi angka dengan menilai seberapa sering suatu kata muncul dalam dokumen dan seberapa jarang di seluruh data. Langkah ini penting agar teks bisa diproses oleh algoritma *machine learning*. Berikut rumus dari TF-IDF

$$TF = \frac{\text{frekuensi suatu kata dalam sebuah dokumen}}{\text{total seluruh kata yang terdapat pada dokumen}} \quad (1)$$

$$IDF = \text{Log}\left(\frac{1+N}{1+\text{df}(t)}\right) + 1 \quad (2)$$

$$TF\text{-}IDF = TF \times IDF \quad (3)$$

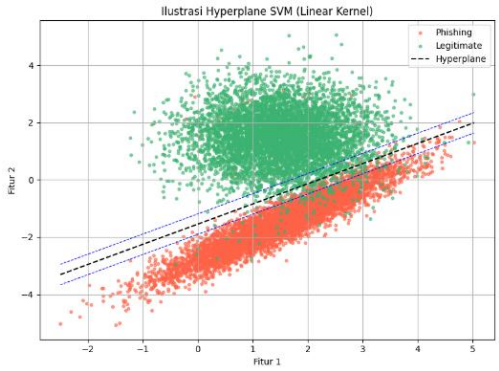
Berdasarkan persamaan (1), (2), dan (3), *Term Frequency* (TF) menunjukkan seberapa sering sebuah kata muncul dalam satu dokumen, sedangkan *Inverse Document Frequency* (IDF) menilai seberapa unik kata tersebut dengan melihat jarangnya kata itu muncul di seluruh kumpulan dokumen. Simbol N mengacu pada jumlah total dokumen, dan Df(t) menyatakan jumlah dokumen yang mengandung kata tersebut.

Tabel 4. Hasil TF-IDF

Url	www	google	co	id	polarklimatsgserver	blogspot	com
https://www.google.co.id	0,3	0,3	0,3	0,3	0	0	0
https://polarklimatsgserver.blogspot.com/	0	0	0	0	0,4	0,4	0,4

2.3 *Support Vector Machine*

Data yang telah dikonversi ke bentuk numerik selanjutnya digunakan untuk melatih model klasifikasi menggunakan algoritma *Support Vector Machine* (SVM), yang dikenal efektif dalam tugas klasifikasi. Penelitian menunjukkan bahwa kernel linier menghasilkan akurasi tertinggi dibandingkan kernel lainnya [10], dan hasil serupa juga diperoleh dalam penelitian ini. SVM bertujuan menemukan *hyperplane* terbaik yang memisahkan kelas data dengan margin maksimal[17]. Berdasarkan evaluasi ini, kernel linier dipilih sebagai konfigurasi optimal karena menunjukkan performa terbaik dalam membedakan URL phishing dan non-phishing, baik pada metrik akurasi maupun stabilitas model di berbagai subset data. Pemilihan kernel linier ini juga mendukung kemampuan generalisasi model terhadap variasi URL yangberbeda, sehingga cocok untuk aplikasi deteksi phishing berbasis web.



Gambar 3. *Hyprlane* SVM

2.4 Evaluasi

Evaluasi model dilakukan untuk mengukur kinerja sistem deteksi URL *phishing* yang dikembangkan. Penilaian dilakukan dengan menggunakan metode evaluasi berikut.

1. *Classification Report*

Classification report merupakan tabel evaluasi yang digunakan untuk menilai performa model *machine learning*. Laporan ini menyajikan nilai *precision*, *recall*, f1-*score*, dan *support* sebagai hasil dari proses pelatihan model[18].

2. *Confusion Matrix*

Untuk menilai performa model klasifikasi, metode yang umum digunakan adalah *confusion matrix* karena dapat menunjukkan seberapa tepat hasil prediksi yang dihasilkan. Dalam penghitungan tingkat akurasi, empat komponen utama yang menjadi tolak ukur adalah *true positive* (TP), *true negative* (TN), *false positive* (FP), serta *false negative* (FN)[19]. Keempat parameter tersebut digunakan sebagai dasar dalam menghitung metrik performa seperti akurasi, presisi, *recall*, dan f1-*score*[20].

a. Akurasi menunjukkan seberapa banyak URL yang diprediksi dengan benar dari seluruh data. Berikut rumus dari akurasi.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

b. Presisi menunjukkan proporsi URL yang benar-benar phishing dari seluruh prediksi phishing. Berikut rumus dari presisi.

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (5)$$

c. *Recall* mengukur seberapa banyak URL phishing yang berhasil dikenali dari total URL phishing yang ada. Berikut rumus dari recall.

$$\textit{Recall} = \frac{TP}{TP+FN} \quad (6)$$

d. F1-*score* merupakan rata-rata harmonis dari nilai presisi dan recall. Berikut rumus dari F1-Score.

$$\text{F1-}\textit{Score} = 2\ x\ \frac{precision\ x\ recall}{Precision+\ recall} \quad (7)$$

3. *Cross Validation*

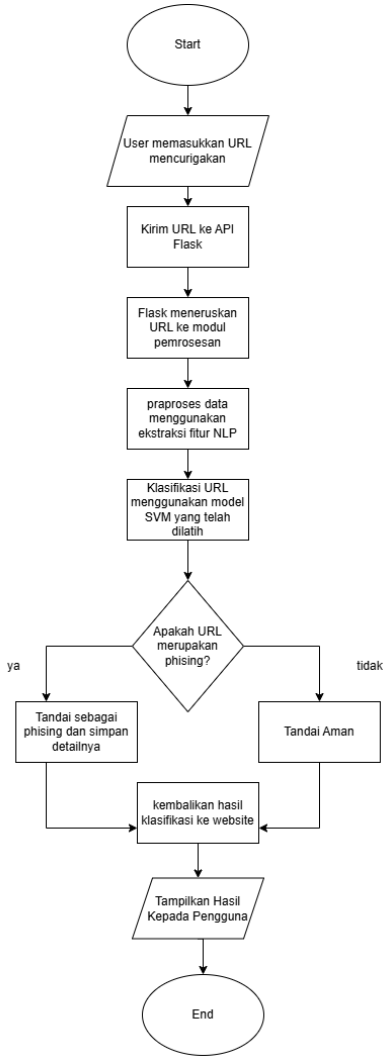
Untuk memperoleh hasil evaluasi yang lebih reliabel dan mengurangi potensi bias akibat pembagian data tunggal, digunakan metode validasi silang (k-*fold cross-validation*). Pada penelitian ini digunakan nilai k = 10, di mana dataset dibagi menjadi sepuluh lipatan (*fold*) dan setiap lipatan secara bergantian digunakan sebagai data uji, sedangkan sisanya sebagai data latih. Hasil dari kelima percobaan tersebut kemudian dirata-ratakan untuk mendapatkan nilai akurasi, presisi, *recall*, dan f1-*score* yang lebih stabil.

4. Kompleksitas Waktu Komputasi

Analisis kompleksitas waktu komputasi digunakan untuk menilai efisiensi model dari sisi pelatihan (*training time*) dan prediksi (*prediction time*). *Training time* menunjukkan waktu yang diperlukan model untuk mempelajari data, sedangkan *prediction time* menggambarkan kecepatan model dalam mengklasifikasikan data baru. Dalam sistem berbasis web, aspek ini penting agar deteksi URL dapat dilakukan secara cepat dan *real-time*.

2.5 Pengembangan Sistem

Pengembangan sistem deteksi URL *phishing* ini dilakukan menggunakan framework Flask, yang bersifat ringan dan cocok untuk aplikasi berbasis web. Sistem dirancang tanpa menggunakan *database*, sehingga seluruh proses deteksi dilakukan secara langsung (*real-time*) saat pengguna memasukkan URL ke dalam form yang tersedia. Antarmuka web dibangun secara sederhana namun fungsional, memungkinkan pengguna mengakses fitur deteksi URL *phishing* dengan mudah. Setelah URL dimasukkan, sistem akan mengekstraksi fitur, melakukan transformasi dengan TF-IDF, dan memprosesnya menggunakan model SVM yang telah dilatih sebelumnya. Hasil klasifikasi langsung ditampilkan kepada pengguna tanpa perlu penyimpanan data. Berikut *flowchart* sistem dapat dilihat pada gambar 4.



Gambar 4. *Flowchart* Sistem

3. HASIL DAN PEMBAHASAN

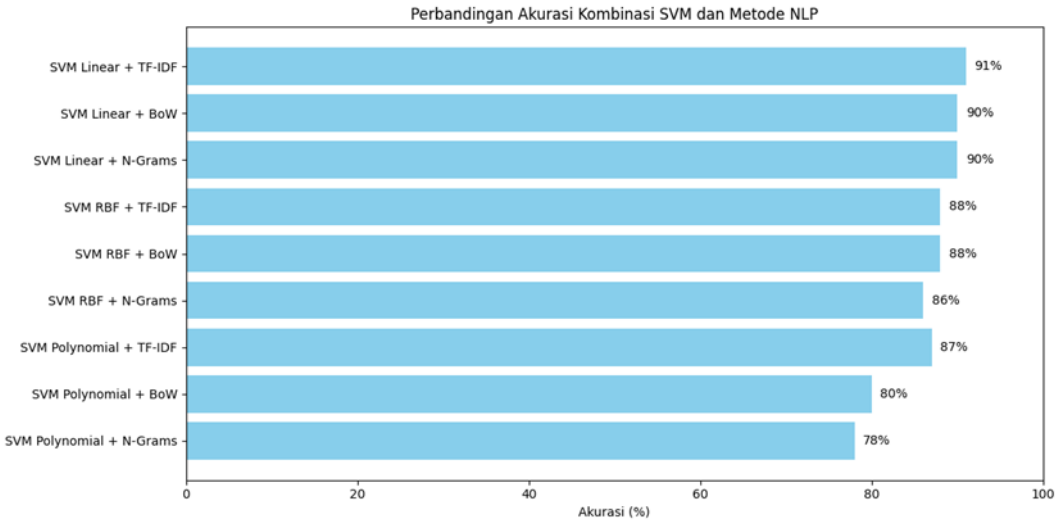
3.1 Hasil Eksperimen

Pada tahap ini dilakukan serangkaian eksperimen guna mencari dan menentukan kombinasi terbaik antara metode ekstraksi fitur teks dan jenis kernel pada algoritma *Support Vector Machine* (SVM) dalam mendeteksi URL *phishing*. Pengujian dilakukan dengan membandingkan tiga jenis kernel, yaitu Linear, *Radial Basis Function* (RBF), dan *Polynomial*, serta tiga teknik representasi teks, yaitu TF-IDF, *Bag of Words*, dan N-Grams.

Tabel 5. Hasil Perbandingan *Classification Report*

Parameter	Akurasi	Presisi	Recall	F1Score
SVM Linear + TF-IDF	91%	91%	91%	91%
SVM Linear + *Bag Of Words*	90%	90%	90%	90%
SVM Linear + N-Grams	90%	90%	90%	90%
SVM RBF + TF-IDF	88%	92%	84%	88%
SVM RBF +*Bag Of Words*	88%	87%	90%	89%
SVM RBF + N-Grams	86%	92%	80%	85%
SVM *Polynomial* + TF-IDF	87%	91%	83%	87%

SVM *Polynomial* + *Bag Of Words*	80%	94%	64%	76%
SVM *Polynomial* + N-Grams	78%	94%	61%	74%



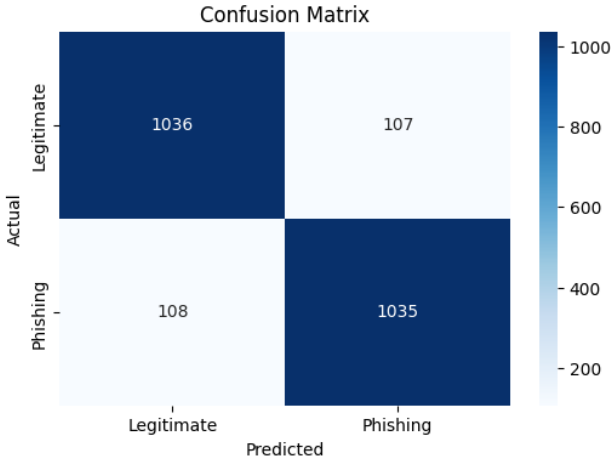
Gambar 5. Perbandingan Performa Kombinasi SVM Dan Metode NLP

Berdasarkan hasil pengujian pada tabel 5 terhadap kombinasi berbagai kernel algoritma *Support Vector Machine* (SVM) dan metode *Natural Language Processing* (NLP), diperoleh bahwa kombinasi SVM Linear dengan TF-IDF menghasilkan performa terbaik dengan akurasi sebesar 91%. Kombinasi lain dengan kernel SVM Linear, yaitu BoW dan N-Grams, juga menunjukkan hasil yang tinggi, masing-masing dengan akurasi 90%. Hal ini menunjukkan bahwa kernel linear secara konsisten unggul dalam menangani fitur teks pada kasus deteksi URL *phishing*. Sementara itu, kombinasi kernel SVM RBF dengan metode TF-IDF dan BoW menghasilkan akurasi yang sama yaitu 88%, sedangkan kombinasi dengan N-Grams sedikit lebih rendah dengan akurasi 86%. Adapun kernel SVM *Polynomial* menunjukkan performa yang lebih rendah dibandingkan kernel lainnya, di mana kombinasi terbaiknya yaitu dengan TF-IDF hanya mencapai akurasi 87%, dan kombinasi dengan BoW serta N-Grams masing-masing memperoleh 80% dan 78%. Hasil ini menunjukkan bahwa metode TF-IDF secara umum lebih unggul dalam mengekstraksi informasi dari data URL dibandingkan BoW dan N-Grams, serta kernel SVM Linear lebih efektif dalam memisahkan data *phishing* dan non-*phishing* dibandingkan kernel RBF dan *Polynomial*.

Tabel 6 memperlihatkan bahwa *classification report* menghasilkan performa yang seimbang dari model *SVM* dalam membedakan antara URL *phishing* dan *legitimate*. Untuk kedua kategori, yakni *phishing* (label 1) dan *legitimate* (label 0), diperoleh skor *precision*, *recall*, serta *f1-score* masing-masing sebesar 0,91. Nilai tersebut mengindikasikan bahwa model mampu mengenali URL dengan akurasi tinggi (*precision*) dan juga konsisten dalam menjangkau seluruh URL yang seharusnya termasuk ke dalam kategori yang sesuai (*recall*). Nilai f1-*score* yang tinggi mengindikasikan keseimbangan antara *precision* dan *recall*. Selain itu, nilai *accuracy* model secara keseluruhan juga mencapai 91%, sedangkan *macro average* dan *weighted average* untuk ketiga metrik utama tersebut masing-masing berada pada angka 0.91. Dengan demikian, model terbukti memiliki kinerja klasifikasi yang stabil dan andal untuk kedua jenis URL.

Tabel 6. *Classification Report*

	Precision	***Recall***	**F1 *score***	***Support***
0	0.91	0.91	0.91	1143
1	0.91	0.91	0.91	1143
Accuracy			0.91	2286
Macro avg	0.91	0.91	0.91	2286
weighted avg	0.91	0.91	0.91	2286



Gambar 6. *Confusion Matrix*

Berdasarkan *confusion matrix* yang ditampilkan pada gambar 6, model *Support Vector Machine* (SVM) berhasil mengklasifikasikan URL *phishing* dan *legitimate* dengan cukup baik. Dari total 2.286 data uji, sebanyak 1.035 URL *phishing* berhasil diprediksi dengan benar sebagai *phishing* (*True Positive*), dan 1.036 URL *legitimate* juga diprediksi dengan benar sebagai *legitimate* (*True Negative*). Namun, terdapat kesalahan prediksi sebanyak 107 URL *legitimate* yang salah diklasifikasikan sebagai *phishing* (*False Positive*), serta 108 URL *phishing* yang salah diklasifikasikan sebagai *legitimate* (*False Negative*).

$$Accuracy = \frac{1035 + 1036}{1035 + 1036 + 107 + 108} = 91\ \%$$

$$Precision = \frac{1035}{1035 + 107} = 91\%$$

$$Recall = \frac{1035}{1035+108} = 91\%$$

$$\text{F1- } score = 2 \times \frac{0{,}91\ x\ 0{,}91}{0.91+0{,}91} = 91\%$$

Dari data tersebut, diperoleh nilai akurasi sebesar 91%, yang dihitung dari rasio jumlah prediksi benar terhadap total data uji. Selain itu, model juga menunjukkan nilai *precision* sebesar 91%, yang menunjukkan bahwa sebagian besar prediksi *phishing* memang benar *phishing*. Kemampuan model dalam mengidentifikasi *URL phishing* secara tepat tercermin dari nilai *recall* sebesar 91%. Selain itu, skor *F1-score* yang juga mencapai 91% menunjukkan adanya keseimbangan yang baik antara *precision* dan *recall*. Hasil ini secara menyeluruh mengindikasikan bahwa performa klasifikasi model tergolong unggul dan layak diandalkan untuk mendeteksi *URL phishing*.

Tabel 7. *Cross Validation*

Parameter	**K1**	**K2**	**K3**	**K4**	**K5**	**K6**	**K7**	**K8**	**K9**	**K10**	***Mean***
SVM Linear + TF-IDF	0,909	0,908	0,916	0,888	0,908	0,911	0,909	0,909	0,894	0,900	0,9055
SVM Linear + *Bag Of Words*	0,905	0,894	0,905	0,894	0,900	0,906	0,904	0,908	0,893	0,885	0,8997

SVM Linear + N-Grams	0,908	0,897	0,909	0,892	0,906	0,902	0,909	0,911	0,893	0,889	0,9021
SVM RBF + TF-IDF	0,885	0,877	0,901	0,887	0,887	0,888	0,885	0,878	0,884	0,870	0,8846
SVM RBF +*Bag Of Words*	0,887	0,879	0,878	0,865	0,881	0,881	0,887	0,888	0,882	0,881	0,8813
SVM RBF + N-Grams	0,863	0,844	0,874	0,853	0,855	0,869	0,863	0,849	0,867	0,853	0,8595
SVM *Polynomial* + TF-IDF	0,875	0,867	0,882	0,872	0,881	0,880	0,870	0,864	0,867	0,849	0,8711
SVM *Polynomial* + *Bag Of Words*	0,802	0,783	0,802	0,797	0,797	0,797	0,787	0,783	0,789	0,768	0,7909
SVM *Polynomial* + N-*Grams*	0,790	0,766	0,783	0,781	0,718	0,784	0,782	0,771	0,771	0,753	0,7766

Tabel 7 menunjukkan bahwa SVM kernel linear secara konsisten unggul dibanding kernel lain. Kombinasi SVM Linear + TF-IDF mencapai akurasi tertinggi 90,55%, diikuti SVM Linear + N-Grams 90,21% dan SVM Linear + *Bag of Words* 89,97%. Kernel RBF memberikan hasil lebih rendah, dengan akurasi tertinggi 88,46% (TF-IDF), sedangkan *Polynomial* menunjukkan performa paling rendah, terutama SVM *Polynomial* + N-*Grams* hanya 77,66%. Stabilitas akurasi kernel linear di setiap fold menandakan kemampuan generalisasi yang baik. Hasil ini menegaskan bahwa SVM Linear + TF-IDF merupakan kombinasi paling optimal untuk deteksi URL *phishing*.

Temuan ini sejalan dengan literatur sebelumnya [11] menyatakan bahwa representasi TF-IDF mampu memproyeksikan data teks ke ruang fitur berdimensi tinggi sehingga pola data lebih mudah dipisahkan secara linear. Hal ini memperkuat pemilihan kernel linear dalam penelitian ini, mengingat data URL *phishing* yang direpresentasikan dengan TF-IDF memiliki karakteristik serupa dengan data teks, sehingga cenderung lebih mudah dipisahkan secara linear dibandingkan dengan pendekatan kernel non-linear. Hasil penelitian [21] juga menunjukkan bahwa SVM dengan kernel linier mencapai akurasi rata-rata sebesar 95,43%, menandakan kemampuannya dalam menangani hubungan linier antara fitur-fitur dalam data teks. Data Twitter untuk analisis sentimen Pilpres 2024 lebih condong ke linear, karena kernel linier memberikan akurasi tertinggi dibandingkan kernel lain. Meskipun teks pada dasarnya kompleks dan sering dianggap non-linear, setelah direpresentasikan dalam bentuk vektor TF-IDF, struktur fiturnya menjadi relatif lebih mudah dipisahkan oleh *hyperplane* linear.

Tabel 8. *Kompleksitas waktu komputasi*

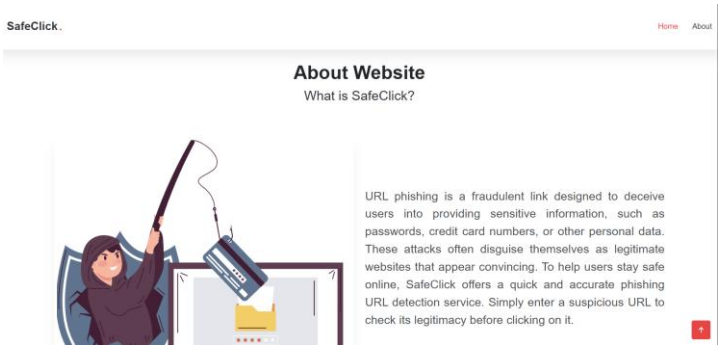
Parameter	Training time	Prediction Time (batch)	Prediction Time (per URL)
SVM Linear + TF-IDF	18.76 detik	0.0072 detik	3e-06 detik
SVM Linear + Bag Of Words	1.36 detik	0.005 detik	2e-06 detik
SVM Linear + N-Grams	10.43 detik	0.0061 detik	3e-06 detik
SVM RBF + TF-IDF	23.41 detik	0.5029 detik	0.00022 detik
SVM RBF +Bag Of Words	688.92 detik	4.4335 detik	0.001939 detik
SVM RBF + N-Grams	28.02 detik	0.5989 detik	0.000262 detik
SVM Polynomial + TF-IDF	26.28 detik	0.6023 detik	0.000263 detik
SVM Polynomial + Bag Of Words	26.59 detik	0.5027 detik	0.00022 detik
SVM Polynomial + N-Grams	25.67 detik	0.7037 detik	0.000308 detik

Hasil pengujian pada Tabel 8 menunjukkan bahwa kernel linear secara konsisten lebih efisien dibandingkan kernel non-linear. Sebagai contoh, SVM Linear + TF-IDF hanya memerlukan 18,76 detik untuk pelatihan dan 0,0072 detik untuk prediksi batch, sementara kernel RBF dengan *Bag of Words* membutuhkan waktu hingga 688,92 detik untuk pelatihan dengan prediksi batch lebih dari 4 detik. Perbedaan yang signifikan ini menunjukkan bahwa kernel non-linear menambah beban komputasi tanpa memberikan peningkatan akurasi yang sepadan. Di sisi lain, meskipun SVM Linear dengan Bag of Words dan SVM Linear dengan N-Grams memiliki waktu komputasi lebih singkat dibanding TF-IDF, tingkat akurasinya lebih rendah. Hal ini menegaskan adanya *trade-off* antara kecepatan dan akurasi. Dengan mempertimbangkan kedua aspek tersebut, SVM Linear + TF-IDF dipilih sebagai model terbaik karena mampu memberikan keseimbangan optimal, akurasi tinggi di atas 90% sekaligus efisiensi komputasi yang memadai untuk mendukung implementasi deteksi *phishing* URL secara *real-time.*

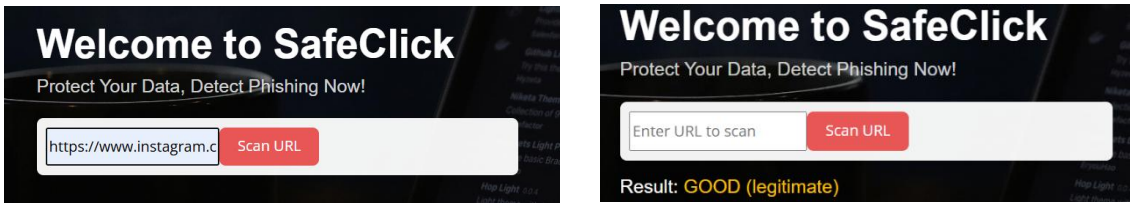
3.2 Website Deteksi Url *Phishing*

Sistem deteksi URL *phishing* yang dikembangkan pada penelitian ini diimplementasikan dalam bentuk website berbasis *framework* Flask, dengan antarmuka sederhana dan ramah pengguna. Website ini ditujukan agar pengguna dapat mengakses fitur deteksi secara langsung melalui browser tanpa perlu instalasi tambahan. Pada gambar 7 merupakan tampilan halaman *home* pada website yang berfungsi sebagai antarmuka awal dari aplikasi bernama *SafeClick*, yang dirancang untuk mendeteksi URL *phishing*.

Gambar 7. Tampilan *Home*



Gambar 8. Tampilan *About*



Gambar 9. Uji Coba Url

Selanjutnya pada gambar 8 menampilkan halaman *About Website* dari aplikasi *SafeClick*, yang berfungsi menjelaskan tujuan dan manfaat dari aplikasi ini kepada pengguna. Gambar 9 menunjukkan proses uji coba URL pada website *SafeClick*. Pengguna memasukkan URL *"https://www.instagram.com"* lalu menekan tombol *Scan* URL. Sistem kemudian menganalisis URL menggunakan model berbasis TF-IDF dan SVM. Hasil deteksi ditampilkan sebagai *"Result: GOOD (legitimate)"*, dengan warna kuning untuk menekankan bahwa Url tersebut aman dan warna merah untuk menekankan bahwa Url tersebut tidak aman (*Phishing*).

3.3 Pengujian Website Dengan Metode *Blackox*

Pengujian dilakukan menggunakan metode *blackbox*, yaitu dengan menguji fungsionalitas sistem tanpa memperhatikan struktur internal program. Pengujian difokuskan pada empat aspek utama sesuai tabel *blackbox testing* yaitu, pengujian URL aman, URL *phishing*, input kosong, dan respons tombol *Scan* URL. Setiap skenario dirancang untuk memastikan bahwa sistem dapat memberikan output yang sesuai, seperti mendeteksi URL aman sebagai "*Good*", phishing sebagai "*Bad*", memberikan peringatan saat input kosong, serta menjaga stabilitas sistem saat tombol ditekan. Pengujian ini memastikan bahwa fitur utama website bekerja sesuai harapan dari sudut pandang pengguna.

Tabel 9. *Blackbox Testing*

Pengujian	*Input*	Hasil	Status
Uji url *good*	https://www.google.co.id	*Good*	Sesuai
Uji url *phishing*	https://polarklimatsgserver.blogspot.com/	*Bad*	Sesuai
Input kosong	Tidak ada input	Muncul peringatan	Sesuai
Uji respons tombol *scan*	Klik tombol *scan* url	Tidak *crash*	Sesuai

Berdasarkan hasil pada tabel 9, dapat disimpulkan bahwa sistem berjalan sesuai fungsinya dan berhasil menangani berbagai jenis input secara tepat.

3.4 Perbandingan Penelitian

Tabel 10. Perbandingan Penelitian Terdahulu

Penelitian	*Algoritma*	Fitur/Representasi	Akurasi	Web
Aryanti dan Nabila	*Support Vector Machine*	Fitur Url +TF-IDF	91% (Classification Report)	Website
Jauhar, dkk	K-Nearest Neighbors	Fitur Url	88%(Classification Report)	Tidak
Mahmud, dkk	*Decision Tree*, *Random Forest*, dan K-NN	Fitur Url	83,3% (DT), 83,4% (RF), 48,2% (K-NN) (Classification Report)	Tidak
Jonathan, dkk	*Naïve Bayes* dan C4.5	Fitur Url	87,11% (C4.5), 78,48% (NB) (Classification Report)	Tidak
Salam Nagalay	*Decision Tree*	Fitur Url	87,94% (Classification Report)	Tidak

Hasil penelitian ini menunjukkan bahwa kombinasi SVM kernel linier dengan TF-IDF mencapai akurasi tertinggi sebesar 91% (classification report), dengan nilai *cross-validation* 90,55% dan confusion matrix 91. Jika dibandingkan dengan penelitian sebelumnya, metode konvensional seperti *Decision Tree, Random Forest, K-Nearest Neighbors*, C4.5, dan Naïve Bayes hanya mencapai akurasi antara 48,2% hingga 88%, serta umumnya hanya menggunakan fitur URL manual tanpa integrasi NLP. Selain itu, sebagian besar penelitian terdahulu belum menyediakan implementasi berbasis web, sehingga pengguna tidak dapat mengakses sistem secara langsung. Penelitian ini menggabungkan fitur struktural URL dengan

representasi teks berbasis TF-IDF dan algoritma SVM kernel linier, serta diimplementasikan dalam platform web yang dapat diakses secara praktis. Hasil ini menegaskan bahwa pendekatan yang digunakan tidak hanya lebih akurat dibandingkan metode sebelumnya, tetapi juga lebih komprehensif dan aplikatif.

Hal ini menegaskan *trade-off* antara kecepatan dan akurasi, sehingga SVM Linear + TF-IDF dipilih sebagai model optimal karena memberikan kombinasi akurasi tinggi di atas 90% dan efisiensi komputasi yang memadai untuk prediksi URL phishing secara real-time.

Meskipun demikian, penelitian ini tetap memiliki keterbatasan. Dataset yang digunakan masih terbatas, sehingga sistem perlu diperbarui secara berkala untuk menangani variasi URL *phishing* terbaru. Namun, sistem masih dapat mendeteksi *phishing* baru selama terdapat pola atau link yang mirip dengan yang sudah ada di dataset. Sebagai rekomendasi praktis, sistem ini dapat digunakan sebagai alat bantu deteksi *phishing*, namun untuk penggunaan publik yang lebih luas disarankan pengembangan tambahan, seperti integrasi *database* untuk menyimpan riwayat deteksi, perluasan variasi dataset URL, serta pengujian *real-time*. Pendekatan *deep learning* dapat dipertimbangkan pada studi lanjutan untuk meningkatkan performa klasifikasi dan kemampuan generalisasi sistem.

4. KESIMPULAN

Sebuah sistem deteksi phishing URL berbasis web telah berhasil dikembangkan dalam penelitian ini dengan memanfaatkan algoritma *Support Vector Machine* (SVM) dan representasi teks *Term Frequency-Inverse Document Frequency* (TF-IDF). Kombinasi SVM kernel linier dengan TF-IDF menghasilkan akurasi tertinggi sebesar 91% (*classification report*), *cross-validation* rata-rata (*mean*) 0,9055, akurasi pada fold 3 sebesar 91,6%, serta nilai *confusion matrix* 91%, menunjukkan konsistensi performa model. Sistem berbasis Flask yang dibangun telah diuji langsung dalam lingkungan nyata dan menunjukkan kinerja yang baik. Analisis komputasi waktu menunjukkan bahwa kernel linear lebih efisien dibandingkan kernel non-linear. SVM Linear + TF-IDF membutuhkan 18,76 detik untuk pelatihan, 0,0072 detik untuk prediksi batch, dan 3e-06 detik untuk prediksi per URL, sementara kernel RBF dengan Bag of Words memerlukan waktu jauh lebih lama. Meskipun SVM Linear dengan Bag of Words dan N-Grams lebih cepat, akurasinya lebih rendah.

DAFTAR PUSTAKA

[1] A. Sudiro, M. D. Ilmawan, and N. V. Puspita, "Pendampingan Terpadu Untuk Maksimalkan Pemasaran Digital Umkm Soto Kudus Kedai Taman Cabang Mojokerto," *Dedikasimu: Journal of Community Service*, vol. 6, no. 4, 2024, doi: https://doi.org/10.30587/dedikasimu.v6i4.8556.

[2] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, and A. Setiawan, "Deteksi Website Phishing Menggunakan Teknik Filter Pada Model Machine Learning," *Information System Journal (INFOS)* , vol. 6, no. 1, 2023, doi: https://doi.org/10.24076/infosjournal.2023v6i01.1268.

[3] Indonesia Anti-Phishing Data Exchange (IDADX), "Laporan Aktivitas Abuse Domain .Id Indonesia Domain Abuse Data Exchange," 2024. Accessed: May 26, 2025. [Online]. Available: https://surli.cc/oojpve

[4] A. Erikha and Z. Arifin Hoesein, "Strategi Pencegahan Kebocoran Data Pribadi melalui Peran Kominfo dan Gerakan Siberkreasi dalam Edukasi Digital," *Jurnal Retentum*, vol. 7, no. 1, 2025, doi: 10.46930.

[5] Budiono, F. R. Fadillah, and N. Arinudin, "The Dangers of Phishing to Personal Data Security," *Formosa Journal of Applied Sciences (FJAS)*, vol. 4, no. 3, 2025, doi: https://doi.org/10.55927/fjas.v4i3.61.

[6] Y. Yuliana, "The Importance Of Cybersecurity Awareness For Children," *Lampung Journal of International Law*, vol. 4, no. 1, pp. 41–48, Jun. 2022, doi: 10.25041/lajil.v4i1.2526.

[7] M. R. Fatiha, I. Setiawan, A. N. Ikhsan, and I. R. Yunita, "Optimisasi Sistem Deteksi Phishing Berbasis Web Menggunakan Algoritma Decision Tree," *Jurnal Ilmiah IT CIDA: Diseminasi Teknologi Informasi*, vol. 10, no. 2, 2024,

doi: https://doi.org/10.55635/jic.v10i2.212.

[8] A. F. Mahmud and S. Wirawan, "Deteksi Phishing Website menggunakan Machine Learning Metode Klasifikasi," *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no. 4, 2024, doi: https://doi.org/10.32520/stmsi.v13i4.

[9] I. Kurniawan, A. L. Hananto, S. S. Hilabi, A. Hananto, B. Priyatna, and A. Y. Rahman, "Perbandingan Algoritma Naive Bayes Dan SVM Dalam Sentimen Analisis Marketplace Pada Twitter," *JATISI*, vol. 10, no. 1, 2023, doi: https://doi.org/10.35957/jatisi.v10i1.3582.

[10] C. M. Putri, M. Afdal, R. Novita, and M. Mustakim, "Perbandingan Evaluasi Kernel Support Vector Machine dalam Analisis Sentimen Chatbot AI pada Ulasan Google Play Store," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 7, no. 3, 2024, doi: https://doi.org/10.32493/jtsi.v7i3.41354.

[11] M. Z. Naeem, F. Rustam, A. Mehmood, Mui-zzud-din, I. Ashraf, and G. S. Choi, "Classification of movie reviews using term frequency-inverse document frequency and optimized machine learning algorithms," *PeerJ Comput Sci*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.914.

[12] F. S. Salam Nagalay, "Analisis Penerapan Algoritma Decision Tree Dalam Keamanan Siber Untuk Kelasifikasi Situs Website Phishing," *Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi*, vol. 10, no. 1, pp. 1–8, 2024, doi: http://dx.doi.org/10.24014/rmsi.v10i1.28401.

[13] A. Jauhar Himawan, A. Meyla Kartika Sari, N. Agatha Parsa, K. Sabilah Putri Hermansyah, and E. Sabrina Dea Rizki, "Penerapan Metode K-Nearest Neighbors dalam Mendeteksi Website Phishing," *COREAI*, vol. 5, no. 2, 2024, doi: 10.33650/coreai.v5i2.10484.

[14] K. M. Jonathan, B. Mulyawan, and N. J. Perdana, "Perbandingan Kinerja Algoritma Naïve Bayes Dan C4.5 Untuk Mendeteksi Pengelabuan Uniform Resource Locator (Phishing Url)," *JIKSI*, vol. 8, no. 1, 2020, doi: https://doi.org/10.24912/jiksi.v8i1.11479.

[15] R. Danar Dana, Mulyawan, A. Bahtiar, and I. Ali, *Dasar Dasar Natural Language Processing (NLP)* . Minhaj Pustaka, 2024. [Online]. Available: https://surl.lu/gpfpax

[16] A. N. Putri, A. Aryanti, and S. Soim, "Implementasi Algoritma SVM Non-Linear Pada Klasifikasi Analisis Sentimen Perkembangan AI di Sektor Pendidikan," *Technology and Science (BITS)*, vol. 6, no. 2, 2024, doi: 10.47065/bits.v6i2.5522.

[17] panji bintoro, ratnasari, edy wihardjo, pratiwi putri indah, and andi asari, *Pengantar Machine Learning*. Pt Mafy Media Literasi Indonesia, 2024. [Online]. Available: https://surli.cc/qqlrwg

[18] M. L. B. Permadi and R. Gumilang, "Penerapan Algoritma CNN (Convolutional Neural Network) Untuk Deteksi Dan Klasifikasi Target Militer Berdasarkan Citra Satelit," *SOSTECH Jurnal sosial dan teknologi*, vol. 4, no. 2, 2024, doi: https://doi.org/10.59188/jurnalsostech.v4i2.1138.

[19] I. wulandari, H. Yasin, and T. Widiharih, "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (Cnn)," *Jurnal Gaussian*, vol. 9, no. 3, 2020, doi: https://doi.org/10.14710/j.gauss.9.3.273-282.

[20] H. I. Islam, M. K. Mulyadien, and U. Enri, "Penerapan Algoritma C4.5 dalam Klasifikasi Status Gizi Balita ," *Jurnal Ilmiah Wahana Pendidikan* , vol. 8, no. 10, 2022, doi: https://doi.org/10.5281/zenodo.6791722.

[21] J. Anggraini and D. Alita, "Implementasi Metode SVM Pada Sentimen Analisis Terhadap Pemilihan Presiden (Pilpres) 2024 Di Twitter," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 9, no. 2, pp. 102–111, Aug. 2024, doi: 10.30591/jpit.v9i2.6560.