JURNAL ILMIAH TEKNOLOGI DAN REKAYASA
Vol. 30, No. 03, December 2025, 277 – 291
p-ISSN: 1410-9093, e-ISSN: 2089-8088
https://ejournal.gunadarma.ac.id/index.php/tekno | tekrek@gunadarma.ac.id

Aplikasi Deteksi *Website Phishing* Berbasis Web Menggunakan *Random Forest* dan Ekstraksi Fitur URL

[1]Adytia Dwi Wulandari, [2*]Dyah Cita Irawati

[1,2]Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma, Jakarta, Indonesia
[1,2]Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat
[1]adytia.wulan30@gmail.com, [2]dyahcita@staff.gunadarma.ac.id

Abstract

Advancements in information technology have raised growing concerns among various stakeholders. Phishing attacks have become one of the most common cyber threats, targeting users by imitating legitimate websites to obtain sensitive information. This study aims to develop a web-based application by implementing a supervised learning approach using the Random Forest algorithm to automatically classify URLs as phishing or legitimate. The dataset used consists of 11,054 URL instances with 30 URL-based features. The research process includes data preprocessing, feature extraction, data splitting, and classification model development and evaluation using four data partition scenarios. Model performance was assessed using accuracy, precision, recall, and F1-score as evaluation metrics. The results of the experiments show that the model achieved optimal performance with an 80:20 data split, obtaining an accuracy of 97%, precision of 97%, recall of 98%, and an F1-score of 97%. Furthermore, the trained model was implemented in a web-based application, allowing users to automatically detect URLs.

Keywords: phishing website detection, random Forest, supervised learning, URL feature extraction, web-based application.

Abstrak

Seiring dengan meningkatnya kecanggihan teknologi informasi telah membawa kekhawatiran terhadap banyak pihak. Serangan *phishing* merupakan salah satu ancaman siber yang terus meningkat dan menargetkan pengguna dengan meniru situs web asli. Penelitian ini bertujuan mengembangkan aplikasi berbasis web dengan menerapkan metode *supervised learning* menggunakan algorima *Random Forest* untuk memprediksi URL secara otomatis sebagai *phishing* atau *legitimate*. *Dataset* yang digunakan terdiri dari 11.054 data URL dengan 30 fitur. . Proses penelitian meliputi tahapan pra-pemrosesan data, ekstraksi fitur, pemisahan data, serta pembangunan dan evaluasi model klasifikasi dengan empat skema pembagian data. Kinerja model dievaluasi menggunakan metrik akurasi, presisi, *recall*, dan F1-*score*. Hasil eksperimen menunjukkan bahwa model mencapai performa optimal pada skema pembagian data 80:20 dengan nilai akurasi sebesar 97%, presisi 97%, *recall* 98%, dan F1-*score* 97%. Selanjutnya, model yang telah dilatih diimplementasikan ke dalam aplikasi berbasis web sehingga memungkinkan pengguna melakukan deteksi URL secara otomatis.

Kata Kunci: aplikasi berbasis web, deteksi website phishing, ekstraksi fitur URL, random forest, supervised learning.

1. Pendahuluan

Perkembangan teknologi informasi telah memudahkan pengguna mencari dan menemukan informasi terkini melalui internet. Namun, seiring kecanggihan teknologi

* Corresponding Author

informasi, ancaman keamanan siber semakin meningkatnya sehingga membawa kekhawatiran bagi masyarakat. Salah satu bentuk ancaman siber adalah serangan *phishing*. *Phishing* merupakan serangan sosio-teknis yang menargetkan informasi berharga dengan memanfaatkan kerentanan sistem dan teknik rekayasa sosial untuk memanipulasi korban agar melakukan tindakan yang merugikan [1]. *Phishing* berupa tindakan dengan mencuri data-data pribadi pengguna melalui sebuah situs yang menyerupai situs aslinya (*website* palsu). Menurut BSSN pada tahun 2024, situs *phishing* telah menduduki posisi ke-3 dari total 10 trafik anomali yaitu sebesar 26.771.610 [2].

Perkembangan teknologi informasi yang kian pesat mendorong adopsi kecerdasan buatan, terutama *machine learning*, dalam berbagai sektor. Pendekatan *machine learning* telah banyak digunakan melalui implementasi langsung pada data nyata, seperti analisis sentiment pelayanan publik, diagnosis penyakit berdasarkan data rekam medis, deteksi fraud pada transaksi keuangan, prediksi persediaan barang pada market, serta analisis data berbasis Internet of Things [3]–[7]. Kemampuan *machine learning* dalam melakukan analisis data secara otomatis, mengenali pola tersembunyi, serta melakukan prediksi berbasis data historis menjadikannya sebagai salah satu pendekatan yang efektif dalam menyelesaikan berbagai permasalahan kompleks pada berbagai domain aplikasi. Selain digunakan pada berbagai sektor industri dan bisnis, pendekatan *machine learning* juga banyak dimanfaatkan dalam bidang keamanan siber untuk mendeteksi aktivitas berbahaya, anomali sistem, serta serangan siber secara otomatis berdasarkan pola data yang teridentifikasi. Hal ini disebabkan karena serangan siber modern memiliki pola yang kompleks dan terus berkembang sehingga sulit dideteksi menggunakan metode tradisional berbasis aturan statis.

Teknologi informasi yang semakin berkembang membuat pelaku *phishing* terus mempelajari dan memperbaharui tekniknya, sehingga metode deteksi tradisional menjadi kurang efektif. Oleh karena itu, diperlukan pendekatan *machine learning* untuk mendeteksi situs yang mengandung *phishing* [8] dan *legitimate* secara otomatis dan adaptif melalui karakteristik URL. Penelitian terdahulu menggunakan algoritma *Support Vector Machine* (SVM) yang dibandingkan dengan algoritma *Decision Tree* dan *K-Nearest Neighbor* (KNN). Penelitian tersebut menyimpulkan bahwa akurasi terbaik didapatkan pada algoritma SVM *kernel polynomial* dengan nilai akurasi 85,71% [9].

Sementara itu, penelitian lain melakukan komparasi dari ke empat algorima yaitu algoritma *Naive Bayes, Random Forest, Decision Tree,* dan SVM dalam klasifikasi *data minin*g yaitu klasifikasi *website phishing*. Hasil pengujian algoritma *Random Forest* memiliki akurasi sebesar 90,77%, nilai akurasi dari algoritma *Decision Tree* 85,77%, algoritma SVM 86,25%, dan *Naïve Bayes* yaitu 82,31% [10]. Selanjutnya penerapan algoritma J48 telah dilakukan pada identifikasi *website phishing*. Hasil pengujian yang didapatkan menunjukkan bahwa identifikasi *website phishing* menggunakan prosedur sederhana yang mudah digunakan, akan tetapi tidak dapat memberikan nilai keakuratan sehingga perlu dilakukan evaluasi, terhadap fitur melakukan kombinasi algoritma lainnya [11].

Penelitian terkait komparasi algoritma telah dilakukan untuk mendeteksi *website phishing* yaitu menggunakan algoritma *Random Forest* dan *Decision Tree.* Berdasarkan analisis percobaan dan klasifikasi algoritma komparatif yang diterapkan menghasilkan bahwa model *Random Forest* menunjukkan akurasi tertinggi sebesar 96,89%, lalu diikuti oleh Model *Decision Tree* sebesar 94,57%, dan *Extreme Gradient Boosting* (XG) [12].

Penelitian deteksi *website phishing* juga telah dilakukan menggunakan algoritma *Random Forest, Decision Tree* dan KNN. Algoritma *Random Forest* memiliki performa terbaik dibandingkan algoritma lainnya [13]. Selain itu, penelitian terbaru menunjukkan keberhasilan kombinasi teknik ekstraksi fitur URL dan algoritma *machine learning* modern dalam meningkatkan akurasi deteksi *phishing*. Penggunaan algoritma *Random Forest* berbasis fitur URL mampu mendeteksi *phishing* URL secara efektif dengan tingkat akurasi yang tinggi

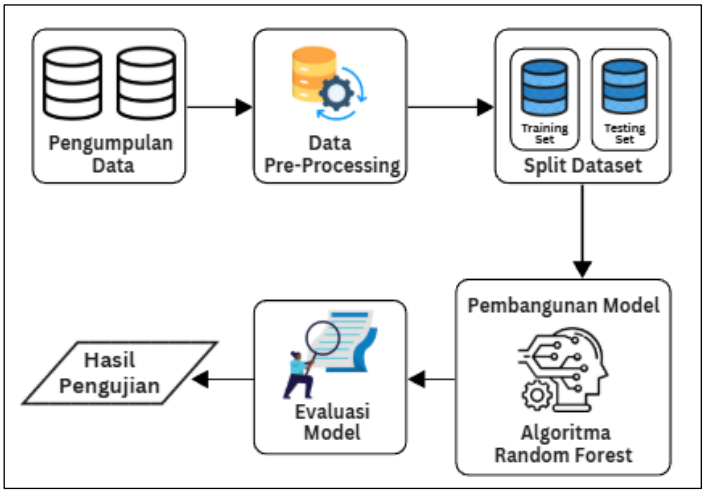
[14]–[16]. Pendekatan *tree base model* juga telah berhasil digunakan untuk mendeteksi *website phishing* dan mengimplementasikan ke dalam *web service* menggunakan Flask dan FastAPI [17]. Penelitian lain menunjukkan bahwa kombinasi beberapa algoritma seperti *Random Forest*, *Naive Bayes*, dan *LightGBM* dapat meningkatkan performa klasifikasi URL *phishing* melalui pendekatan *ensemble* [18]. Selain itu, penerapan teknik seleksi fitur yang tepat dapat meningkatkan performa model klasifikasi phishing berbasis *machine learning* [19]. Pendekatan *hybrid machine learning* juga mampu meningkatkan akurasi deteksi *phishing* dengan memanfaatkan kombinasi beberapa teknik klasifikasi [20].

Berdasarkan uraian tersebut dapat disimpulkan bahwa algoritma *Random Forest* telah banyak digunakan dan menunjukkan kinerja yang baik dalam deteksi website *phishing*. Namun, sebagian besar penelitian masih berfokus pada aspek pemodelan dan evaluasi algoritma dan masih sedikit yang mengembangkan dalam bentuk aplikasi yang dapat digunakan secara langsung oleh pengguna. Selain itu, jumlah dan variasi fitur URL yang digunakan pada penelitian sebelumnya relatif terbatas sehingga belum sepenuhnya merepresentasikan karakteristik kompleks URL *phishing*. Khususnya terkait penggunaan fitur URL yang masih terbatas dan penelitian difokuskan pada evaluasi model klasifikasi tanpa implementasi ke dalam aplikasi yang dapat diakses oleh pengguna secara langsung.

Penelitian ini mengisi celah penelitian yang ada dengan tujuan membangun model deteksi *phishing* menggunakan algoritma *Random Forest* berdasarkan fitur-fitur URL yang dilakukan dengan proses ekstraksi fitur yang bertujuan menerapkan metode *supervised learning* pada algoritma *Random Forest* menggunakan pendekatan *machine learning* pada pembuatan sistem berbasis *web*. Secara khusus, tujuan penelitian ini adalah (1) mengembangkan proses ekstraksi fitur URL menjadi 30 fitur yang mencerminkan karakteristik struktural dan leksikal URL, (2) membangun dan mengevaluasi model klasifikasi *Random Forest* dengan beberapa skema pembagian data latih dan data uji, serta (3) mengimplementasikan model deteksi *phishing* berupa aplikasi berbasis web yang memungkinkan proses deteksi URL dilakukan secara otomatis oleh pengguna.

2. Metode Penelitian

Penelitian ini dilakukan dalam dua tahapan utama, yakni pengembangan model deteksi *phishing* berbasis *Random Forest* dan implementasinya dalam aplikasi berbasis web. Tahapan dalam membangun model klasifikasi dengan algoritma *Random Forest* terdiri dari enam tahap seperti yang disajikan pada Gambar 1.

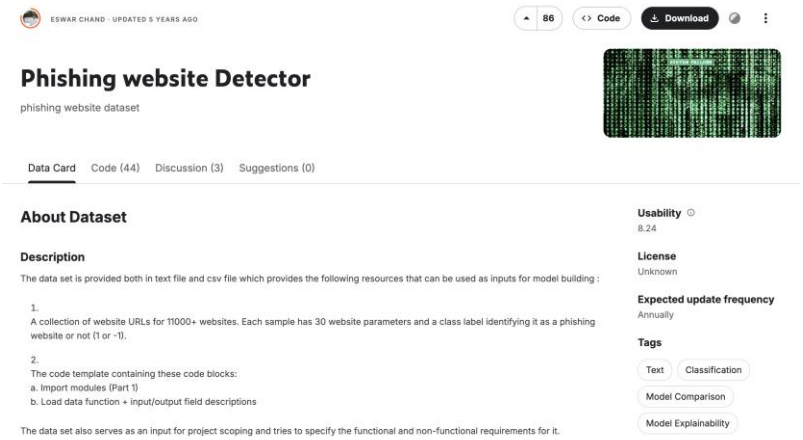


Gambar 1. Alur Pembangunan Model *Random Forest*

Pada Gambar 1 ditunjukkan alur penelitian yang meliputi pengumpulan data, pra-pemrosesan data, pembagian *dataset*, pembangunan model menggunakan algoritma *Random Forest*, dan evaluasi model sehingga diperoleh hasil pengujian model yang menunjukkan performa model klasifikasi berdasarkan data uji yang digunakan. *Random Forest* menghasilkan keputusan akhir melalui mekanisme voting dari seluruh pohon keputusan yang dibangun [21]. Tahap pertama yaitu pembangunan model *machine learning* yang dilakukan dalam 6 langkah berikut.

2.1. Pengumpulan Data

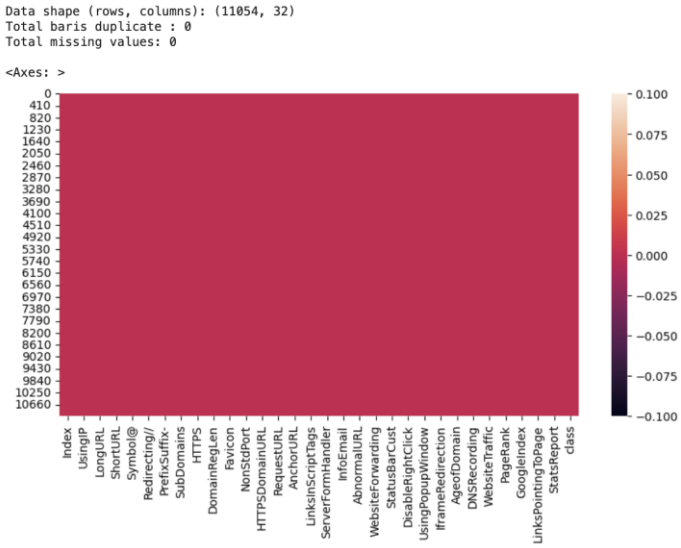
Pengumpulan *dataset* berasal dari *platform Kaggle* yang berformat csv dengan total 11.054 baris dan 32 kolom yang berisi ekstraksi data URL dari *website phishing* [22]. Gambar 2 adalah tampilan *platform* untuk mengunduh *dataset phishing*.



Gambar 2. Tampilan *Platform Kaggle* Pada *Dataset Phishing* [22]

2.2. Data *Pre-processing*

Tahap *pre-processing* merupakan sebuah tahapan yang penting dalam analisis data. Tahapan *pre-processing* diawali dengan menghapus nilai kosong dan ditampilkan pada visualisasi data *isnull*.



Gambar 3. Visualisasi Data *isnull*

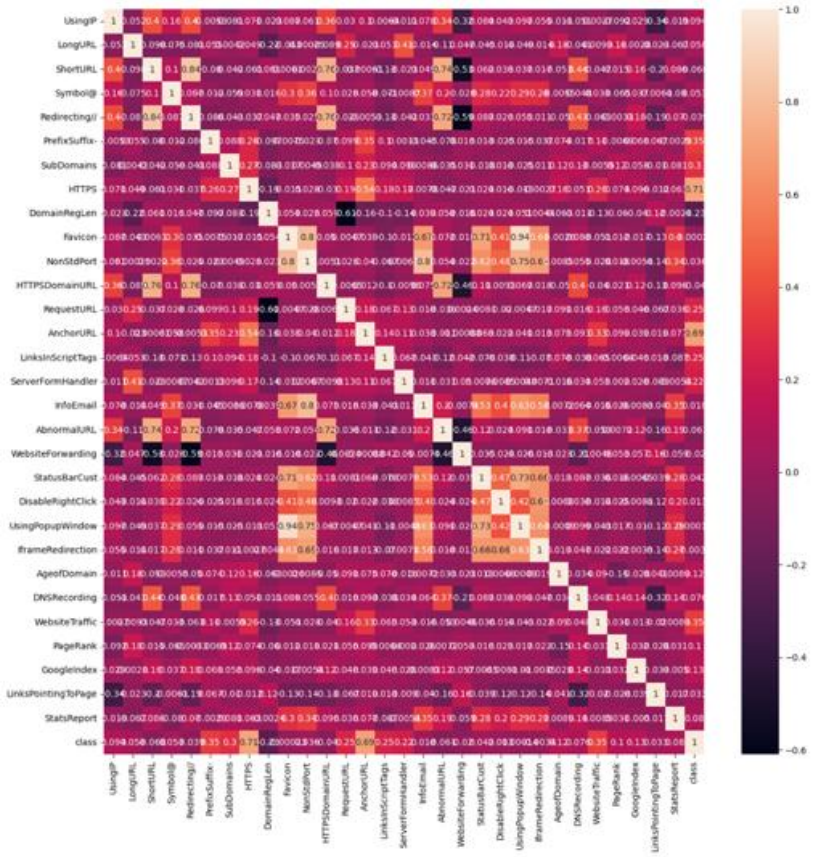
Gambar 3 menunjukkan hasil pemeriksaan kualitas *dataset* sebelum digunakan dalam proses pelatihan model. Berdasarkan hasil analisis, dataset memiliki ukuran 11.054 baris data dan 32 kolom fitur. Hasil pemeriksaan menunjukkan bahwa tidak terdapat data duplikat pada *dataset*, serta tidak ditemukan nilai kosong (*missing values*) pada seluruh atribut. Selain itu, visualisasi *heatmap* menunjukkan distribusi nilai kosong pada setiap fitur *dataset*. Warna yang seragam pada seluruh area *heatmap* menunjukkan bahwa tidak terdapat nilai kosong pada dataset, sehingga *dataset* dinilai telah memenuhi syarat untuk digunakan dalam proses pelatihan model *machine learning* tanpa memerlukan proses imputasi data.

index	count	mean	std	min	25%	50%	75%	max
Index	11054.0	5526.5	3191.1592721141324	0.0	2763.25	5526.5	8289.75	11053.0
UsingIP	11054.0	0.3139135154695133	0.9494945300528388	-1.0	-1.0	1.0	1.0	1.0
LongURL	11054.0	-0.6333453953320065	0.7659725279822142	-1.0	-1.0	-1.0	-1.0	1.0
ShortURL	11054.0	0.7387371087389181	0.6740241710750983	-1.0	1.0	1.0	1.0	1.0
Symbol@	11054.0	0.7005608829383029	0.713624915165166	-1.0	1.0	1.0	1.0	1.0
Redirecting//	11054.0	0.7416319884204813	0.670837316382356	-1.0	1.0	1.0	1.0	1.0
PrefixSuffix-	11054.0	-0.7349375791568663	0.6781654372999564	-1.0	-1.0	-1.0	-1.0	1.0
SubDomains	11054.0	0.06404921295458657	0.8174924841058729	-1.0	-1.0	0.0	1.0	1.0
HTTPS	11054.0	0.2510403473855618	0.9118559757696955	-1.0	-1.0	1.0	1.0	1.0
DomainRegLen	11054.0	-0.33671069296182377	0.9416507447594394	-1.0	-1.0	-1.0	1.0	1.0
Favicon	11054.0	0.6285507508594174	0.7778037560880788	-1.0	1.0	1.0	1.0	1.0
NonStdPort	11054.0	0.7282431698932513	0.6853498211173514	-1.0	1.0	1.0	1.0	1.0
HTTPSDomainURL	11054.0	0.6752306857246245	0.7376399827896496	-1.0	1.0	1.0	1.0	1.0
RequestURL	11054.0	0.18671973946082865	0.9824576620852524	-1.0	-1.0	1.0	1.0	1.0
AnchorURL	11054.0	-0.07644291659127918	0.7151161385413755	-1.0	-1.0	0.0	0.0	1.0
LinksInScriptTags	11054.0	-0.11823774199384839	0.7639331714596265	-1.0	-1.0	0.0	0.0	1.0
ServerFormHandler	11054.0	-0.5957119594716844	0.7591676640270358	-1.0	-1.0	-1.0	-1.0	1.0
InfoEmail	11054.0	0.6357879500633254	0.7718986877115581	-1.0	1.0	1.0	1.0	1.0
AbnormalURL	11054.0	0.7054459924009409	0.7087957397152309	-1.0	1.0	1.0	1.0	1.0
WebsiteForwarding	11054.0	0.11570472227248055	0.3198849738152504	0.0	0.0	0.0	0.0	1.0
StatusBarCust	11054.0	0.7620770761715217	0.6475156058464809	-1.0	1.0	1.0	1.0	1.0
DisableRightClick	11054.0	0.9138773294734938	0.406008792032043	-1.0	1.0	1.0	1.0	1.0
UsingPopupWindow	11054.0	0.6133526325312104	0.7898449121200884	-1.0	1.0	1.0	1.0	1.0
IframeRedirection	11054.0	0.8168988601411253	0.5768070306236605	-1.0	1.0	1.0	1.0	1.0
AgeofDomain	11054.0	0.06133526325312104	0.9981623707068409	-1.0	-1.0	1.0	1.0	1.0
DNSRecording	11054.0	0.37723900850370906	0.9261578312936537	-1.0	-1.0	1.0	1.0	1.0
WebsiteTraffic	11054.0	0.28740727338519995	0.8276801756812626	-1.0	0.0	1.0	1.0	1.0
PageRank	11054.0	-0.48362583680115795	0.8753144360885378	-1.0	-1.0	-1.0	1.0	1.0
GoogleIndex	11054.0	0.7215487606296364	0.6923949421559671	-1.0	1.0	1.0	1.0	1.0
LinksPointingToPage	11054.0	0.34394789216573185	0.5699357456617828	-1.0	0.0	0.0	1.0	1.0
StatsReport	11054.0	0.7197394608286594	0.6942756760424511	-1.0	1.0	1.0	1.0	1.0
class	11054.0	0.11398588746155237	0.9935273097083025	-1.0	-1.0	1.0	1.0	1.0

Gambar 4. Persebaran Data Statistik Deskriptif

Gambar 4 menunjukkan hasil analisis statistik deskriptif dataset yang terdiri dari 11.054 data dengan beberapa fitur yang merepresentasikan karakteristik URL. Analisis statistik meliputi nilai rata-rata (*mean*), standar deviasi (*standard deviation*), nilai minimum, kuartil, median, dan nilai maksimum untuk setiap fitur. Berdasarkan hasil analisis, sebagian besar fitur memiliki rentang nilai antara -1 dan 1, yang menunjukkan bahwa dataset telah melalui proses normalisasi atau representasi kategorikal numerik. Nilai standar deviasi yang relatif stabil pada sebagian besar fitur menunjukkan bahwa distribusi data tidak terlalu menyimpang dan memiliki variasi yang cukup untuk digunakan dalam proses pelatihan model *machine learning*.

Selain itu, distribusi nilai pada beberapa fitur menunjukkan kecenderungan dominasi nilai tertentu, yang mencerminkan karakteristik pola URL *phishing* dan *legitimate* dalam *dataset*. Analisis statistik deskriptif ini digunakan untuk memahami karakteristik awal data sebelum dilakukan proses pelatihan model klasifikasi. Pada data statistik deskriptif diatas menampilkan adanya *outlier* pada fitur indeks karena memiliki nilai *max* = 11053 dan *mix* = 0, berbeda jauh dengan fitur lainnya dan tidak relevan untuk model, sehingga fitur indeks tersebut tidak digunakan dalam proses pembangunan model.



Gambar 5. *Correlation Heatmap Fitur-Fitur Website Phishing*

Pada Gambar 5 ditampilkan *correlation* fitur pada tahapan data *pre-processing* ini. *Correlation* pada *dataset* yang bertujuan untuk mengukur seberapa kuat arah hubungan linear antara dua variabel. Korelasi tiap fitur penting dipertimbangkan dalam proses *split dataset* yaitu *selection feature* agar model yang dibuat tidak terlalu kompleks atau tumpang tindih. Gambar 5 menunjukkan visualisasi matriks korelasi antar fitur pada *dataset* menggunakan *heatmap*. Matriks korelasi digunakan untuk mengidentifikasi hubungan linear antar fitur serta mengetahui tingkat keterkaitan antar variabel dalam *dataset*. Korelasi mendekati 1 menunjukkan hubungan positif kuat, mendekati −1 menunjukkan hubungan negatif kuat, dan mendekati 0 menunjukkan tidak terdapat hubungan linier.

Berdasarkan visualisasi *heatmap*, sebagian besar fitur menunjukkan tingkat korelasi yang rendah hingga sedang. Hal ini menunjukkan bahwa fitur-fitur dalam *dataset* relatif independen. Selain itu, *dataset* memiliki tingkat redundansi fitur yang rendah sehingga seluruh fitur masih relevan untuk digunakan dalam proses pelatihan model klasifikasi. Resiko multikolinearitas pada *dataset* relatif kecil karena tidak terlihat adanya kelompok fitur dengan korelasi sangat tinggi secara dominan. Analisis korelasi ini digunakan sebagai bagian dari tahap pra-pemrosesan data untuk memahami hubungan antar fitur sebelum dilakukan proses pelatihan model *machine learning*.

Tahap akhir dari tahapan *pre-processing* adalah menentukan fitur yang digunakan dalam pembangunan model. Proses ekstraksi fitur adalah proses mengubah data yang tidak terstruktur (URL *website*) menjadi sekumpulan numerik atau kategorikal (angka) yang bisa dibaca oleh *machine learning* untuk melakukan prediksi. Fitur yang telah dipilih akan tetapkan sebagai *feature* untuk proses pemisahan antara *feature* dan target.

2.3. Pemisahan *Dataset*

Pada tahap selanjutnya, *dataset* dipisahkan menjadi dua bagian yang terdiri atas data pelatihan dan data pengujian. Skema pembagian *dataset* dibagi ke dalam empat skema yaitu:

1) Data pelatihan 60% dan data pengujian 40%
2) Data pelatihan 70% dan data pengujian 30%
3) Data pelatihan 80% dan data pengujian 20%
4) Data pelatihan 90% dan data pengujian 10%

Dataset dibagi menjadi data pelatihan dan data pengujian menggunakan fungsi *train_test_split* dari Scikit-learn untuk melatih dan mengevaluasi performa model terhadap data yang belum pernah dilihat sebelumnya.

2.4. Pembangunan Model *Machine Learning*

Model deteksi *website phishing* merupakan model klasifikasi yang dibangun menggunakan algoritma *Random Forest* dengan memanfaatkan data latih. Proses pelatihan dilakukan untuk mempelajari hubungan antara fitur URL dan label kelas sehingga model mampu mengklasifikasikan URL sebagai *phishing* atau *legitimate*.

2.5. Evaluasi Model

Evaluasi model dilakukan untuk mengukur performa model yang dibangun dalam melakukan prediksi atau klasifikasi untuk membedakan URL *phishing* dan *legitimate*. Proses evaluasi ini menggunakan *confusion matrix*. *Confusion matrix* menyajikan informasi mengenai total jumlah prediksi benar dan salah yang dilakukan oleh model terhadap masing-masing kelas, tertera pada Tabel 1.

Tabel 1. Struktur *Confusion Matrix*

	Prediksi Kelas	
Kelas Sebenarnya	TN	FP
	FN	TP

Keterangan: TP adalah *True Positive*, FP adalah *False Positives*, TN adalah *True Negatives*, dan FN adalah *False Negatives.*

Confusion matrix terdiri dari empat komponen utama seperti yang ditunjukkan pada Tabel 1 yang digunakan untuk menganalisis tingkat kesalahan klasifikasi model secara detail. Berdasarkan nilai-nilai tersebut, berbagai metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* dapat dihitung untuk mengukur performa model klasifikasi secara menyeluruh. Penggunaan *confusion matrix* memungkinkan analisis yang lebih komprehensif terhadap performa model, khususnya dalam mengidentifikasi jenis kesalahan klasifikasi yang dapat berdampak pada performa sistem secara keseluruhan [23]–[25].

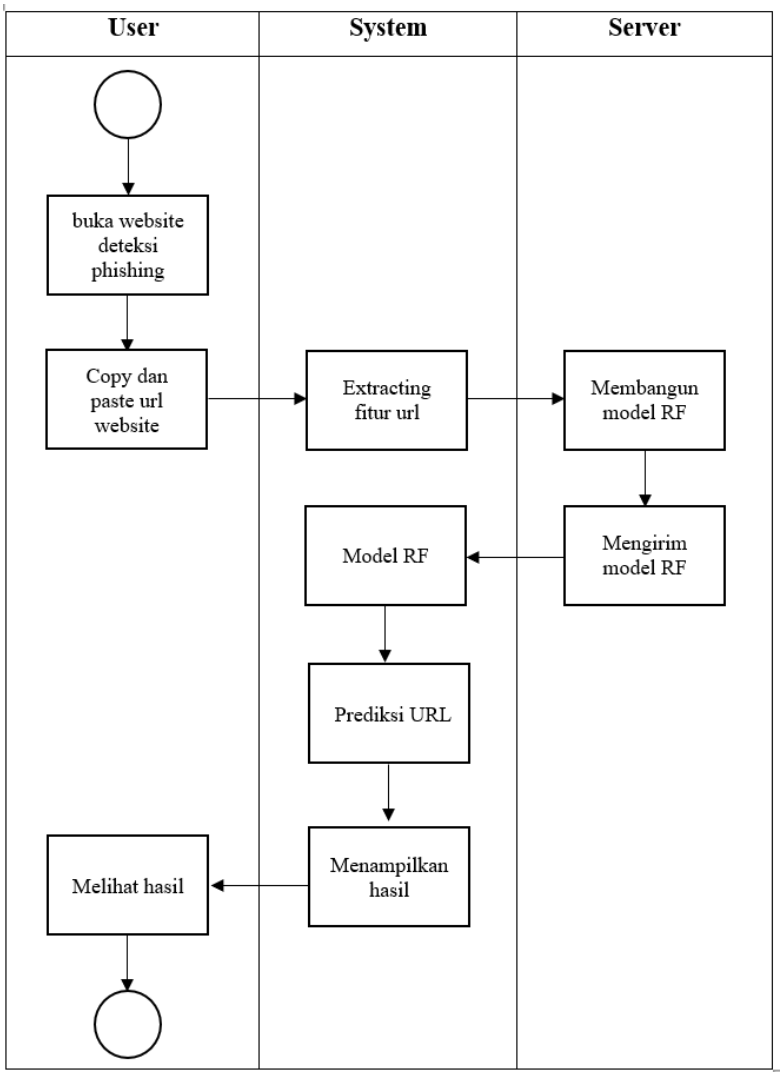
Proses evaluasi model menentukan seberapa baik model melakukan klasifikasi terhadap data yang belum pernah dilihat sebelumnya. Penelitian ini melakukan pembagian pengujian menjadi empat skema untuk menentukan proporsi pembagian data terbaik yang digunakan dalam pembangunan model.

2.6. Implementasi Model dalam Aplikasi Berbasis Web

Setelah model berhasil dibangun menggunakan algoritma *Random Forest* dan menunjukkan performa model yang baik, kemudian model diimplementasikan kedalam sebuah aplikasi *web* berbasis *Flask* yang berfungsi sebagai sistem deteksi *website phishing* secara otomatis. Aplikasi ini digambarkan dengan *activity diagram* yang menunjukkan proses alur dari *user*, sistem, dan *server*. Gambar 6 adalah alur kerja aplikasi dan Gambar 7 adalah struktur navigasi *website*, menggunakan alur navigasi hierarki tanpa adanya *login*.

Gambar 6 tersebut menunjukkan *activity diagram* alur kerja sistem deteksi *website phishing* yang melibatkan tiga komponen utama, yaitu pengguna (*user*), sistem aplikasi, dan

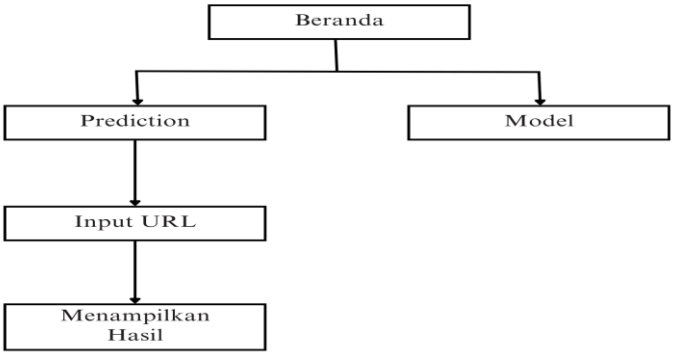
server. Proses dimulai ketika pengguna membuka halaman *website* deteksi *phishing* dan memasukkan URL *website* yang akan diperiksa melalui proses *copy* dan *paste*. URL yang dimasukkan oleh pengguna kemudian diproses oleh sistem pada tahap ekstraksi fitur URL untuk mengambil karakteristik struktural dan leksikal yang diperlukan sebagai input model klasifikasi. Selanjutnya, sistem akan mengirimkan data hasil ekstraksi fitur ke *server* untuk diproses menggunakan model klasifikasi *Random Forest* yang telah dibangun sebelumnya. *Server* melakukan proses klasifikasi menggunakan model *Random Forest* dan mengirimkan hasil prediksi kembali ke sistem. Sistem kemudian menampilkan hasil klasifikasi kepada pengguna dalam bentuk status apakah URL yang diuji termasuk kategori phishing atau legitimate. Proses berakhir setelah pengguna melihat hasil deteksi yang ditampilkan oleh sistem.



Gambar 6. Alur Kerja Aplikasi

Gambar 7 tersebut menunjukkan struktur navigasi halaman pada aplikasi deteksi *website phishing* berbasis web. Struktur navigasi dimulai dari halaman utama (*Beranda*) yang menjadi pusat akses pengguna terhadap fitur utama aplikasi. Dari halaman beranda, pengguna dapat mengakses dua menu utama yaitu menu *Prediction* dan menu *Model*. Menu *Prediction* digunakan untuk melakukan proses deteksi *phishing*. Pada menu ini, pengguna dapat memasukkan URL *website* yang ingin diuji melalui fitur *Input URL*. Setelah URL dimasukkan, sistem akan memproses data menggunakan model klasifikasi dan menampilkan

hasil deteksi pada halaman *Menampilkan Hasil*. Sementara itu, menu *Model* menyediakan informasi terkait model klasifikasi yang digunakan dalam sistem deteksi *phishing*. Struktur navigasi ini dirancang untuk memudahkan pengguna dalam mengakses fitur deteksi URL serta informasi terkait model yang digunakan dalam sistem.



Gambar 7. Struktur Navigasi *Website*

3. Hasil dan Pembahasan

3.1 Hasil Evaluasi Model

Pengujian model dilakukan menggunakan empat skema pembagian *dataset*, yaitu 60:40, 70:30, 80:20, dan 90:10 antara data latih dan data uji. Hasil evaluasi kinerja model berdasarkan masing-masing skema pembagian data disajikan pada Tabel 2.

Tabel 2. Kumpulan Hasil Pengujian Evaluasi Model

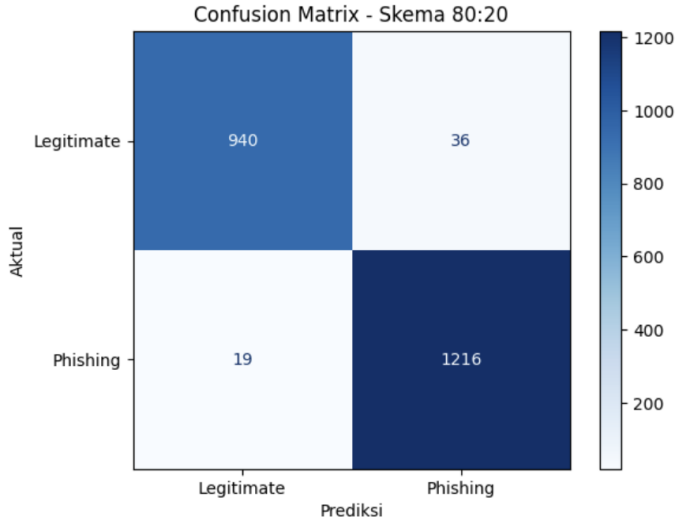
Data *Train*	Data *Test*	Akurasi	Presisi	*Recall*	*F1-score*	*Cross Validation*
60%	40%	0.968	0.966	0.978	0.974	0.966
70%	30%	0.965	0.963	0.975	0.969	0.968
80%	20%	0.975	0.971	0.984	0.974	0.970
90%	10%	0.958	0.954	0.97	0.959	0.966

Berdasarkan Tabel 2, skema pembagian data 80:20 menghasilkan kinerja terbaik dengan nilai akurasi sebesar 0,97, presisi 0,97, *recall* 0,98, dan *F1-score* 0,97. Skema 60:40 dan 70:30 juga menunjukkan kinerja yang tinggi, namun nilai *recall* pada skema 80:20 lebih unggul dalam mendeteksi URL *phishing*. Sementara itu, skema 90:10 menghasilkan kinerja yang lebih rendah dibandingkan skema lainnya, yang mengindikasikan bahwa proporsi data uji yang terlalu kecil dapat memengaruhi kemampuan model dalam melakukan generalisasi.

3.2 Analisis *Confusion Matrix*

Gambar 8 menunjukkan *confusion matrix* hasil pengujian model klasifikasi *website phishing* yang menggunakan algoritma *Random Forest* dengan skema pembagian data 80:20. *Confusion matrix* digunakan untuk menilai kemampuan model dalam mengklasifikasikan data ke dalam dua kelas, yaitu *legitimate* dan *phishing*. Hasil pengujian menunjukkan bahwa sebanyak 940 data *legitimate* berhasil diklasifikasikan dengan tepat sebagai *legitimate*, sementara 36 data *legitimate* salah diklasifikasikan sebagai *phishing*. Pada kelas *phishing*, sebanyak 1.216 data berhasil diidentifikasi dengan benar sebagai *phishing*, sedangkan 19 data *phishing* salah diklasifikasikan sebagai *legitimate*.

Berdasarkan hasil *confusion matrix*, jumlah *false negative* yang relatif kecil menunjukkan bahwa model mampu meminimalkan kesalahan dalam mendeteksi website *phishing* sebagai *legitimate* sehingga tidak merugikan pengguna. Selain itu, jumlah *false positive* yang relatif rendah menunjukkan bahwa model tidak terlalu sering mengklasifikasikan website *legitimate* sebagai *phishing.* Hal ini mengindikasikan bahwa model memiliki kinerja yang seimbang dalam mengklasifikasikan kedua kelas, sehingga dapat diterapkan pada sistem deteksi *phishing* berbasis web.



Gambar 8. *Confusion Matrix* Skema 80:20

Tabel 3 adalah nilai dari hasil pengujian dengan skema 80:20. Evaluasi dilakukan menggunakan presisi, *recall*, F1-*score*, dan *support* untuk masing-masing kelas, yaitu *legitimate* (-1) dan *phishing* (1). Nilai presisi yang diperoleh sebesar 0,97 pada kedua kelas. Sementara itu, nilai *recall* pada kelas *legitimate* mencapai 0,96 dan pada kelas *phishing* sebesar 0,98, yang mengindikasikan kemampuan model yang sangat baik dalam mendeteksi URL *phishing*. Nilai F1-*score* masing-masing sebesar 0,96 untuk kelas *legitimate* dan 0,97 untuk kelas *phishing*, yang mencerminkan keseimbangan antara presisi dan *recall* pada kedua kelas. Selain itu, nilai *macro average* dan *weighted average* yang sama-sama sebesar 0,97 menunjukkan bahwa model memiliki performa klasifikasi yang stabil dan seimbang pada kedua kelas data.

Tabel 3. Hasil Pengujian Skema 80:20

	precision	***recall***	***F1-score***	***support***
-1	0,97	0,96	0,96	976,00
1	0,97	0,98	0,97	1235,00
Accuracy	0,97	0,97	0,97	0,97
macro avg	0,97	0,97	0,97	2211,00
weighted avg	0,97	0,97	0,97	2211,00

Jumlah kesalahan klasifikasi yang relatif kecil pada kelas *phishing* menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi URL *phishing*. Hal ini penting dalam konteks keamanan siber, karena kesalahan dalam mendeteksi *phishing* dapat menyebabkan potensi kerugian bagi pengguna. Selain itu, jumlah kesalahan pada kelas *legitimate* juga relatif rendah sehingga menunjukkan bahwa model tidak terlalu sering mengklasifikasikan *website legitimate* sebagai *phishing*.

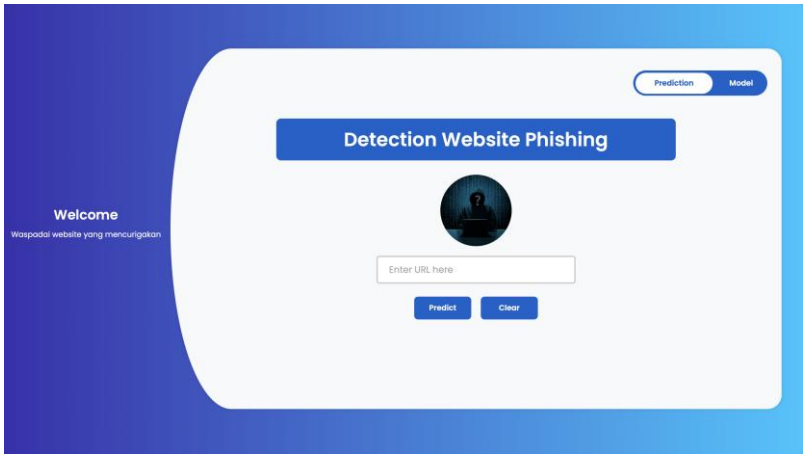
Hasil ini menunjukkan bahwa algoritma *Random Forest* mampu mempelajari pola karakteristik URL *phishing* dan *legitimate* dengan baik berdasarkan fitur URL yang digunakan dalam penelitian ini. Keseimbangan model dalam mengklasifikasikan kedua kelas menunjukkan kemampuan generalisasi yang baik terhadap data, sementara rendahnya tingkat kesalahan pada kelas *phishing* menunjukkan efektivitas model dalam mendeteksi URL *phishing*.

Hasil penelitian ini mendukung penelitian sebelumnya yang menunjukkan keunggulan algoritma *Random Forest* dalam menghasilkan akurasi klasifikasi yang tinggi pada deteksi *website phishing* [4], [6], [7]. Selain itu, penelitian lain menunjukkan bahwa penggunaan pendekatan berbasis fitur URL mampu meningkatkan kemampuan model dalam membedakan karakteristik website *phishing* dan *legitimate* [8], [10]. Hal ini mendukung hasil penelitian ini yang menggunakan ekstraksi fitur URL sebagai variabel *input* model klasifikasi.

Penelitian terbaru juga menunjukkan bahwa penggunaan algoritma *ensemble* dan pendekatan *hybrid machine learning* mampu meningkatkan performa deteksi *phishing* [11]–[14]. Hasil penelitian ini menunjukkan kecenderungan yang serupa, di mana algoritma *Random Forest* sebagai metode *ensemble* mampu memberikan performa klasifikasi yang tinggi dalam mendeteksi website *phishing*. Kemampuan model dalam mempelajari pola URL *phishing* dan *legitimate* menunjukkan bahwa fitur URL yang digunakan dalam penelitian ini mampu merepresentasikan karakteristik penting dalam membedakan kedua kelas tersebut. Pemanfaatan jumlah fitur yang lebih besar dibandingkan beberapa penelitian terdahulu berpotensi memperkuat kemampuan model dalam mengidentifikasi pola *phishing* yang bersifat lebih kompleks.

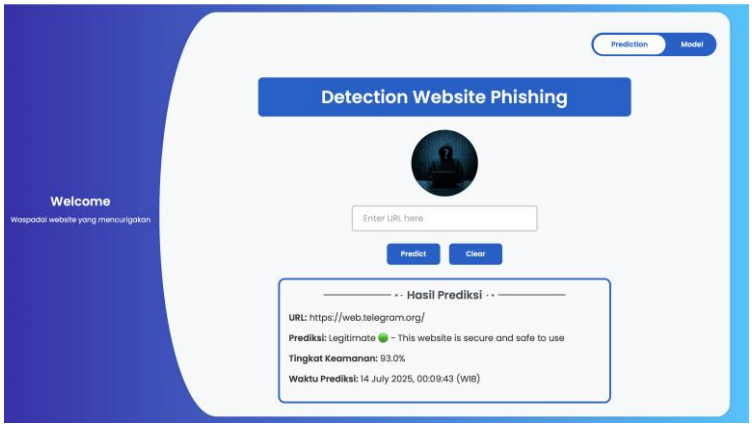
3.3 Implementasi Aplikasi Berbasis Web

Setelah proses pembangunan model *machine learning* maka dilakukan implementasi model ke dalam aplikasi web. Hasil penelitian yang telah dilakukan berhasil membuat aplikasi deteksi *website phishing* menggunakan bahasa pemrograman Python dan *framework Flask*. Implementasi web pada aplikasi deteksi *website phishing*, terdiri dari 2 halaman, yakni halaman *prediction* untuk melakukan prediksi dan model untuk menyajikan hasil evaluasi model berdasarkan data uji, terlihat pada Gambar 9. Dalam aplikasi deteksi *website phishing*, *user* memasukkan URL untuk dilakukan prediksi. Saat *user* memasukkan URL dan melakukan *predict* aplikasi dijalankan dengan melakukan ekstraksi fitur dengan menggolongkan situs tersebut aman (*legitimate*), mencurigakan, atau *phishing*. Fitur-fitur yang diekstrak berjumlah 30 fitur.



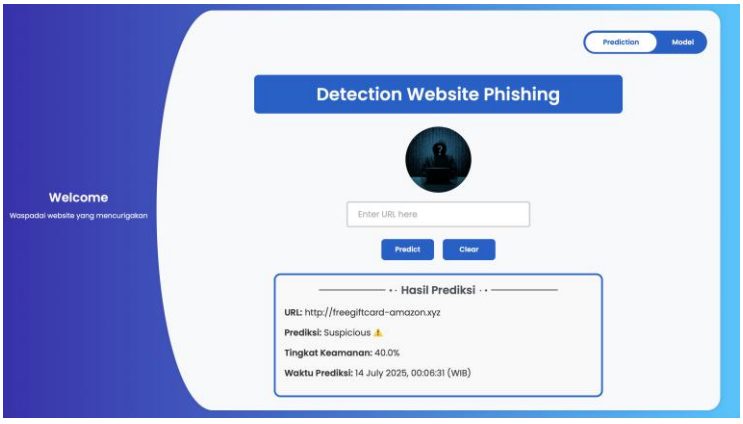
Gambar 9. Tampilan Aplikasi Melakukan Prediksi

User melakukan prediksi dengan menekan *button* "*predict*". Kemudian, jika hasil prediksi *user* berhasil ditampilkan, *user* dapat melakukan prediksi kembali dengan menekan *button clear* untuk menghapus semua *history* dan melakukan prediksi kembali dengan URL yang berbeda. Gambar 10, Gambar 11, dan Gambar 12 adalah gambar hasil prediksi dari beberapa *website*.



Gambar 10. Tampilan Aplikasi Prediksi *Website* Aman

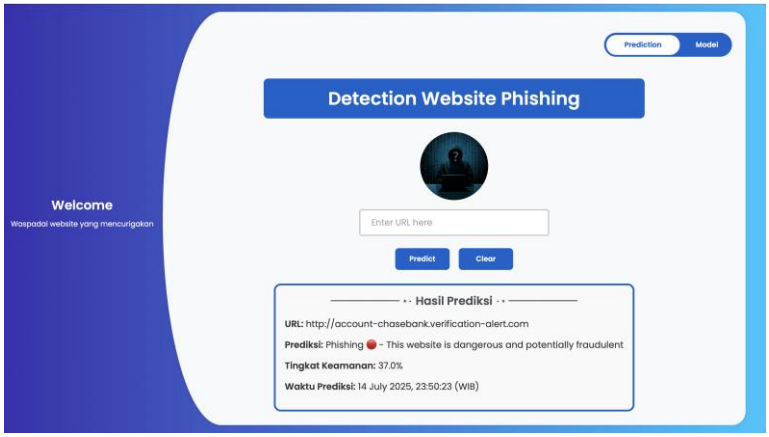
Gambar 10 memperlihatkan aplikasi berhasil melakukan prediksi pada situs *telegram web* dengan hasil tergolong aman/*legitimate* dengan tingkat keamanan 93% aman dan 7% *phishing*.



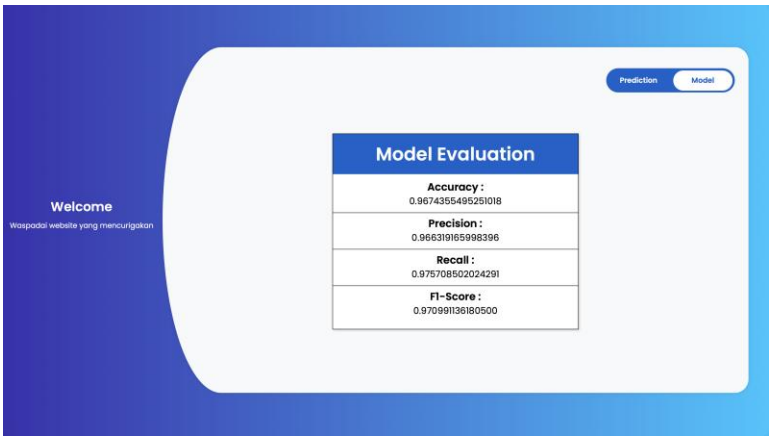
Gambar 11. Tampilan Aplikasi Prediksi *Website* Mencurigakan

Kemudian Pada Gambar 11 adalah tampilan aplikasi memprediksi *website* mencurigakan pada URL "http://freegiftcard-amazon.xyz" dengan tingkat keamanannya sebesar 40% aman dan 60% adanya indikasi *phishing*. Prediksi yang dilakukan bahwa aplikasi tersebut mencurigakan karena meniru hadiah amazon, dengan tampilan dari *link* tersebut mirip dengan tampilan resmi hadiah amazon sehingga *user* akan diminta untuk memasukkan data pribadi.

Gambar 12 adalah tampilan aplikasi memprediksi *website* yang berbahaya karena berpotensial mengandung *phishing* yang menyatakan bahwa tingkat keamanan yang dimiliki *website* tersebut adalah 37% dinyatakan aman dan 63% adalah *phishing*. Hasil tersebut menunjukkan *website* tergolong *phishing* dan berbahaya.



Gambar 12. Tampilan Aplikasi Prediksi *Website* Potensial *Phishing*



Gambar 13. Tampilan Model Evaluasi dari Data Uji

Beralih ke *button* Model yang berfungsi untuk menampilkan evaluasi model. Nilai evaluasi model digunakan sebagai indikator untuk menilai tingkat kinerja model yang telah dikembangkan. Pada Gambar 13 terlihat bahwa rata-rata perolehan nilai metrik dari hasil evaluasi model diperoleh nilai diatas 95%. Hal ini mengindikasi bahwa model memiliki kemampuan klasifikasi yang baik.

Hasil penelitian ini menunjukkan potensi penerapan model klasifikasi pada sistem deteksi *phishing* berbasis web. Dengan tingkat kesalahan klasifikasi yang relatif rendah, model dapat digunakan sebagai sistem pendukung dalam mendeteksi URL phishing secara otomatis. Implementasi model pada aplikasi berbasis web menunjukkan bahwa pendekatan *machine learning* dapat diterapkan secara efektif dalam sistem keamanan siber berbasis aplikasi.

4. Kesimpulan

Penelitian ini berhasil merancang dan membangun model pendeteksian situs web *phishing* dengan memanfaatkan algoritma *Random Forest* melalui ekstraksi fitur URL, serta mengimplementasikannya dalam sebuah aplikasi berbasis web. Proses ekstraksi menghasilkan 30 fitur URL yang merepresentasikan karakteristik struktural dan leksikal URL sebagai masukan bagi model klasifikasi. Hasil evaluasi menggunakan empat skema pembagian data menunjukkan bahwa rasio 80:20 memberikan performa paling optimal, dengan capaian akurasi sebesar 97%, presisi 97%, *recall* 98%, serta F1-*score* 97%. Nilai *recall* tinggi pada

kelas *phishing* menunjukkan kemampuan model dalam mengidentifikasi URL *phishing* secara tepat.

Penerapan model ke dalam aplikasi berbasis web menunjukkan bahwa model klasifikasi yang dikembangkan mampu diintegrasikan secara efektif ke dalam sistem deteksi *phishing* otomatis. Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan sistem deteksi phishing berbasis *Random Forest* melalui penggunaan representasi fitur URL yang lebih beragam serta implementasi model dalam bentuk aplikasi nyata.

Penelitian selanjutnya dapat ditingkatkan dengan menggunakan algoritma lain seperti *XGBoost, LightGBM*, atau metode *deep learning*. Selain itu, perlu dilakukan pengujian terhadap data *real-time* dan dilengkapi dengan database sistem yang terintegrasi sehingga mampu menyimpan data hasil pengujian. Selain itu, dapat ditambahkan proses *hosting* ke *platform* aplikasi agar dapat diakses dan disebarluaskan ke masyarakat umum, sehingga masyarakat memiliki edukasi mengenai keamanan siber dan dapat melindungi diri dari serangan *phishing*.

Daftar Referensi

[1] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "Phishing attacks: A recent comprehensive study and a new anatomy," *Frontiers in Computer Science*, vol. 3, Art. no. 563060, Mar. 2021, doi: 10.3389/fcomp.2021.563060.

[2] Direktorat Operasi Keamanan Siber, Badan Siber dan Sandi Negara, *Lanskap keamanan siber Indonesia 2024*, Jakarta, Indonesia: BSSN, 2024.

[3] A. Nugraha and D. Riminarsih, "Evaluasi performa algoritma supervised learning untuk prediksi risiko serangan jantung: Pendekatan rekayasa sistem cerdas," *Jurnal Profesi Insinyur (JPI)*, vol. 6, pp. 83-88, 2025, doi: 10.23960/jpi.v6n1.169.

[4] N. Awan *et al.*, "Machine learning-enabled power scheduling in IoT-Based Smart Cities," *Computers, Materials and Continua*, vol. 67, no. 2, 2021, doi: 10.32604/cmc.2021.014386.

[5] A. Yasmin, S. Kamalakkannan, and P. Kavitha, "Stock market prediction using machine learning models," in *International Conference on Edge Computing and Applications, ICECAA 2022 - Proceedings*, 2022, doi: 10.1109/ICECAA55415.2022.9936188.

[6] A. Saxena, "Credit card fraud detection using machine learning and data science," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 12, 2022, doi: 10.22214/ijraset.2022.48293.

[7] M. Saraswati and D. Riminarsih, "Analisis sentimen terhadap pelayanan KRL Commuterline berdasarkan data Twitter menggunakan algortima bernoulli naive bayes," *Jurnal Ilmiah Informatika Komputer*, vol. 25, no. 3, 2020, doi: 10.35760/ik.2020.v25i3.3256.

[8] I. Arifin and Chairani, "Phishing website detection using a machine learning classification approach," *INOVTEK Polbeng - Seri Informatika*, vol. 10, no. 3, 2025, doi: 10.35314/yja1d830.

[9] D. Wahyudi, Aplikasi pendeteksi website phishing menggunakan machine learning, Skripsi, Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin, Gowa, Indonesia, 2020. [Online] Available: https://repository.unhas.ac.id/id/eprint/3061/4/20_D42115518(FILEminimizer)%20...%20ok.pdf. [Accessed: 20 Februari 2025].

[10] N. B. Putri and A. W. Wijayanto, "Analisis komparasi algoritma klasifikasi data mining dalam klasifikasi website phishing," *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, 2022, doi: 10.34010/komputika.v11i1.4350.
[11] R. P. Ramadhan and T. Desyani, "Implementasi algoritma J48 untuk identifikasi website phising," *Teknik dan Multimedia*, vol. 1, no. 2, 2023.
[12] M. A. Taha, H. D. A. Jabar, and W. K. Mohammed, "A machine learning algorithms for detecting phishing websites: A comparative study," *Iraqi Journal for Computer Science and Mathematics*, vol. 5, no. 3, 2024, doi: 10.52866/ijcsm.2024.05.03.015.
[13] A. F. Mahmud and S. Wirawan, "Deteksi phishing website menggunakan machine learning metode klasifikasi," *Sistemasi: Jurnal Sistem Informasi*, vol. 13, no. 4, 2024.
[14] A. Kautsar, M. Aida, and A. Yulistia, "Applying random forest algorithm for phishing URL identification," *Journal of Computers and Digital Business*, vol. 4, no. 3, 2025, doi: 10.56427/jcbd.v4i3.782.
[15] I G. P. Wiratama and A. A. I. N. E. Karyawati, "Klasifikasi URL berbahaya menggunakan algoritma random forest berbasis fitur struktural", *JNATIA*, vol. 4, no. 1, pp. 39–46, Nov. 2025, doi: 10.24843/JNATIA.2025.v04.i01.p05.
[16] A. K. Kencana, F. D. Ananda, A. D. Hartanto, and H. Hartatik, "Implementasi metode random forest klasifikasi untuk phishing link detection," *Intechno Journal (Information Technology Journal)*, vol. 4, no. 2, 2022, doi: 10.24076/intechnojournal.2022v4i2.1562.
[17] Lukito and W. B. T. Handaya, "Deteksi website phishing menggunakan teknik machine learning," *Jurnal Informatika Atma Jogja*, vol. 6, no. 1, 2025, doi: 10.24002/jiaj.v6i1.11538.
[18] C. F. M. Foozy, M. A. I. Anuar, A. Maslan, H. A. M. Adam, and H. Mahdin, "Phishing URLs detection using naives baiyes, random forest and lightgbm algorithms," *International Journal of Data Science*, vol. 5, no. 1, 2024, doi: 10.18517/ijods.5.1.56-63.2024.
[19] D. R. Patil, R. B. Wagh, V. D. Punjabi, and S. M. Pardeshi, "Enhanced phishing URLs detection using feature selection and machine learning approaches," *International Journal of Wireless and Microwave Technologies*, vol. 14, no. 6, 2024, doi: 10.5815/ijwmt.2024.06.04.
[20] M. U. Javeed, S. M. Aslam, H. A. Sadiqa, A. Raza, M. M. Iqbal, and M. Akram, "Phishing website URL detection using a hybrid machine learning approach," *Journal of Computing and Biomedical Informatics*, vol. 9, no. 1, 2025.
[21] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[22] E. Chandt, "Phishing website detector," [Online] Available: https://www.kaggle.com/datasets/eswarchandt/phishing-website-detector. [Accessed: 20 Februari 2025].
[23] J. Han, M. Kamber, and J. Pei, *Data Mining Concept and Techniques, 3rd ed.* 2012.
[24] C. M. Bishop, *Bishop - Pattern Recognition and Machine Learning - Springer 2006*, vol. 58, no. 12. 2014.
[25] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Second Edition*, vol. 27, no. 2. 2009.