Jurnal Ilmiah Sistem Informasi dan Ilmu Komputer
Volume 6, Nomor 3, November 2026
e-ISSN: 2827-7953; p-ISSN: 2827-8135, Hal. 17-27
DOI: https://doi.org/10.55606/juisik.v6i3.2512
Tersedia: http://journal.sinov.id/index.php/juisik

Aplikasi Deteksi URL *Phishing* Menggunakan Algoritma Random Forest

Fiqri Dian Priyatna Sinaga[1*], Armansyah[2], Ahmad Taufik Al Afkari Siahaan[3]
[1-3]Jurusan Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara Medan, Indonesia
Email: fiqridpss@gmail.com[1], armansyah@uinsu.ac.id[2], ahmadtaufikalafkari@uinsu.ac.id[3]
**Penulis Korespondensi: fiqridpss@gmail.com*

Abstract. *Phishing is one of the cyberattacks that frequently occurs and can cause financial losses as well as personal data theft. This attack is carried out through malicious URLs that are difficult for users to distinguish from legitimate URLs. Therefore, a website-based phishing URL detection application was developed to enable users to determine the security level of a URL. This study uses the Prototype method. The requirements analysis stage was conducted through a literature review and utilized the PhiUSIIL Phishing Website dataset consisting of 235,795 records categorized as phishing and non-phishing, as well as formal and non-formal requirements. The application design stage includes the development of the detection interface, use case diagram, entity relationship diagram, and activity diagram. The model development stage applies the Random Forest algorithm with an 80:20 dataset split, resulting in an accuracy of 99.98%. The application implementation stage was carried out using Flask technology by implementing the previous stages into a website application. The testing stage was conducted using Unit Testing and Blackbox Testing. The Unit Testing results show that the application successfully extracts URL features and performs detection. Testing using the URL "https://www.instagram.com/" identified the URL as "Not Phishing" with a "Legit Probability of 0.95.." and a "Phishing Probability of 0.04..". The Blackbox Testing results show that the test cases in 3 out of 3 testing scenarios were successfully executed. It can be concluded that the website-based phishing URL detection application using the Prototype method and the Random Forest algorithm was successfully developed and implemented. This application can help users determine the security of a URL and facilitate the phishing URL detection process.*

Keywords: *Application; Phishing; Prototype; Random Forest; URL Detection.*

Abstrak. *Phishing* merupakan salah satu serangan siber yang sering terjadi dan dapat menyebabkan kerugian finansial serta pencurian data pribadi. Serangan ini dilakukan melalui URL berbahaya yang sulit dibedakan dari URL asli oleh pengguna. Oleh karena itu, dikembangkan sebuah aplikasi deteksi URL *phishing* berbasis website yang memungkinkan pengguna mengetahui tingkat keamanan URL. Penelitian ini menggunakan metode Prototype. Tahap analisis kebutuhan dilakukan melalui studi literatur serta menggunakan dataset PhiUSIIL *Phishing Website* yang berjumlah 235.795 data dengan kategori *phishing* dan tidak *phishing*, serta kebutuhan formal dan non-formal. Tahap perancangan aplikasi meliputi pembuatan antarmuka deteksi, use case diagram, *entity relationship* diagram, dan *activity* diagram. Tahap pembentukan model menerapkan algoritma Random Forest dengan pembagian dataset sebesar 80:20 yang menghasilkan akurasi sebesar 99,98%. Tahap penerapan aplikasi dilakukan menggunakan teknologi Flask dengan mengimplementasikan tahapan sebelumnya ke dalam aplikasi website. Tahap pengujian dilakukan menggunakan Unit Testing dan Blackbox Testing. Hasil Unit Testing menunjukkan bahwa aplikasi berhasil mengekstraksi fitur URL dan melakukan deteksi. Pengujian menggunakan URL "https://www.instagram.com/", sistem mengidentifikasi URL sebagai "Tidak Phishing" dengan nilai "Legit Probability sebesar 0,95.." dan "*Phishing Probability* sebesar 0,04..". Hasil Blackbox Testing menunjukkan bahwa test case pada 3 dari 3 skenario pengujian berhasil dijalankan. Dapat disimpulkan bahwa aplikasi deteksi URL phishing berbasis website menggunakan metode Prototype dan algoritma Random Forest berhasil dikembangkan dan diimplementasikan dengan baik. Aplikasi ini dapat membantu pengguna mengetahui keamanan URL serta memudahkan proses deteksi URL phishing.

Kata kunci: Application; *Phishing*; Prototype; Random Forest; URL Detection.

1. LATAR BELAKANG

Phishing merupakan salah satu serangan siber yang paling sering terjadi dan menyebabkan kerugian finansial serta pencurian data pribadi (Alkhalil et al., 2021). Data APWG terakhir menunjukkan bahwa media sosial menjadi target paling populer dengan

Naskah Masuk: 05 Juni 2026; Revisi: 05 Juli 2026; Diterima: 05 Agustus 2026; Terbit: 01 November 2026

persentase 22,5%, dan Cyber Security News juga menunjukkan bahwa kasus *phishing* masih tergolong tinggi (Anti-Phishing Working Group, 2024; Cyber Security News, 2024).

Phishing biasanya dilakukan melalui URL atau tautan berbahaya yang dikirim melalui email dari pihak yang dikenal (Opara et al., 2024). Pengguna sering kali sulit membedakan antara URL asli dan URL palsu. Oleh karena itu, diperlukan aplikasi yang dapat membantu pengguna mendeteksi keamanan URL. Proses deteksi secara manual dinilai kurang efektif bagi pengguna umum.

Machine learning menjadi pendekatan yang digunakan untuk mendeteksi *phishing* dalam penelitian ini. Berdasarkan penelitian SLR Safi dan Singh, *machine learning* merupakan metode yang paling banyak digunakan serta Random Forest menjadi yang paling konsisten dan akurat dibandingkan dengan Decision Tree, XGBoost, dan beberapa algoritma populer lainnya (Safi & Singh, 2023). Algoritma ini mampu mengenali pola URL *phishing* secara otomatis dan menjadi alternatif yang lebih cepat dibanding analisis manual.

Berdasarkan permasalahan tersebut, solusi yang ditawarkan adalah merancang aplikasi deteksi URL *phishing* berbasis website. Website menjadi *platform* yang dipilih karena mudah diakses (Serrano & Emonet, 2022). Pengembangan aplikasi dilakukan menggunakan metode SDLC *Prototype*. Pengguna memasukkan URL, kemudian sistem melakukan klasifikasi menggunakan Random Forest. Hasil deteksi ditampilkan dalam bentuk kategori *phishing* atau tidak *phishing*. Penelitian ini bertujuan membantu masyarakat memverifikasi URL secara mudah, mengurangi risiko menjadi korban *phishing*, serta meningkatkan kesadaran terhadap keamanan siber.

2. KAJIAN TEORITIS

URL *Phishing*

Phishing merupakan kejahatan dunia maya yang bertujuan memperoleh informasi sensitif dari korban (Alkhalil et al., 2021). Dalam praktiknya, *phishing* dilakukan dengan menyamar sebagai pihak tepercaya untuk mengelabui pengguna internet agar memberikan data pribadi mereka (Safi & Singh, 2023). Serangan ini umumnya dilakukan melalui tautan atau URL berbahaya yang dikirim melalui email atau pesan dari pihak tidak dikenal untuk mencuri informasi pribadi pengguna (Opara et al., 2024).

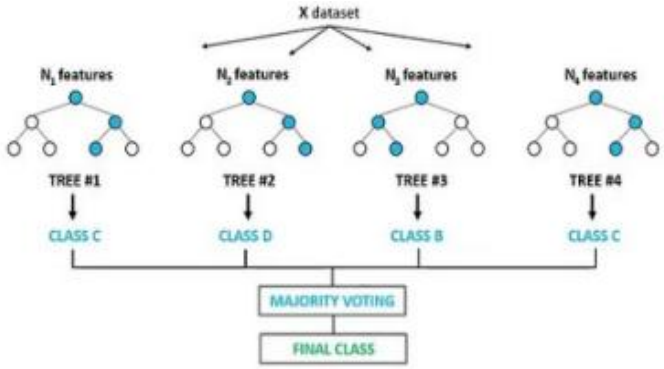
Aplikasi

Aplikasi merupakan perangkat lunak yang digunakan untuk mengolah data secara lebih optimal melalui sistem komputer (Wahyuni, 2023). Berdasarkan *platform*nya, aplikasi dibedakan menjadi desktop, web, dan mobile (Larasati et al., 2021). Aplikasi web adalah jenis

aplikasi yang diakses melalui *browser*, seperti Mozilla Firefox dan Google Chrome (Kadir, 2023). Aplikasi web tidak memerlukan proses instalasi karena pengguna hanya perlu mengakses situs melalui *browser* pada perangkat mereka (Putra et al., 2023).

Random Forest

Random Forest merupakan algoritma *ensemble learning* dalam kategori *supervised learning* yang digunakan untuk tugas klasifikasi. Algoritma ini membangun beberapa pohon keputusan dan menggabungkan hasil prediksinya untuk meningkatkan akurasi serta mengurangi *overfitting*. Setiap pohon dilatih menggunakan *subset* data dan fitur secara acak sehingga mampu menghasilkan generalisasi yang lebih baik dalam menangani klasifikasi yang kompleks (Rahmaddeni et al., 2025).



Gambar 1. Random forest *classifier*

Keterangan: Tahapan dari Random Forest classifier.

Sumber: Website Medium (2021).

Dalam menentukan pohon keputusan, digunakan rumus *Gini* untuk mengukur tingkat *impurity* data, yaitu:

$$Gini = 1 - \sum_{i=1}^{c} p_i^2 \quad \text{.....(i)}$$

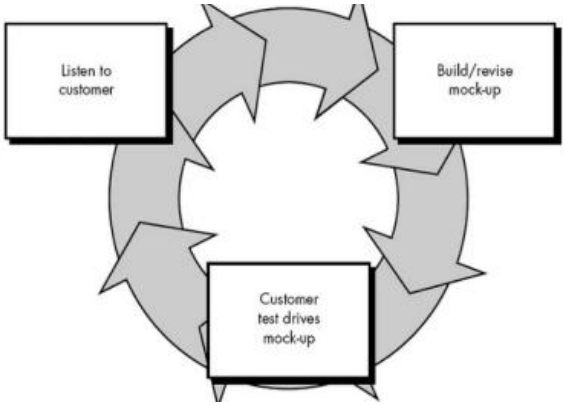
Dengan (c) sebagai jumlah kelas dan (p_i) sebagai probabilitas setiap kelas. Jika jumlah kelas kurang dari atau sama dengan dua, maka rumus *Gini* dapat dituliskan sebagai:

$$Gini = 1 - (p_1)^2 - (p_2)^2 \quad \text{.....(ii)}$$

(Andika, 2024).

Prototype

Prototype merupakan rancangan awal untuk menguji konsep dan proses kerja suatu produk agar kekurangan dan kesalahan dapat diketahui sebelum produk dikembangkan dan dirilis (Pantiwati et al., 2024).



Gambar 2. Metode *prototype*

Keterangan: Tahapan-tahapan SDLC *protoype*.

Sumber: Website Medium.

3. METODE PENELITIAN

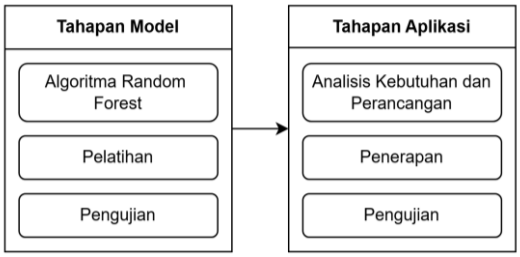
Penelitian ini menggunakan *dataset* sekunder PhiUSIIL Phishing URL (Website) yang diperoleh dari UCI Machine Learning Repository. *Dataset* tersebut terdiri dari URL *phishing* dan tidak *phishing* dengan 8 fitur pendukung, 1 fitur target, dan total 235.795 data. *Dataset* dapat dilihat pada Tabel 1.

Tabel 1. *Dataset*

No	URL	URL Length	Domain Length	...	URL Similarity Index	Label
1	..southba..	31	24	…	100	1
2	..uni-mai..	23	16	…	100	1
…	…	…	…	…	…	…
235.795	..cell-40f..	55	47	…	28	0

"URLLength" menunjukkan panjang URL, "DomainLength" panjang *domain*, "IsDomainIP" penggunaan alamat IP, "TLDLength" panjang *top-level domain*, "NoOfSubDomain" jumlah *subdomain*, "NoOfOtherSpecialCharsInURL" jumlah karakter khusus pada URL, "IsHTTPS" penggunaan protokol HTTPS, "URLSimilarityIndex" tingkat kemiripan URL, dan "Label" sebagai kategori *phishing* (0) atau tidak *phishing* (1).

Dari segi tahapan penelitian, penelitian ini menggunakan algoritma Random Forest dalam pembuatan model serta menerapkan metode pengembangan *Prototype* dalam pengembangan aplikasi.



Gambar 3. Tahapan penelitian

Keterangan: Tahapan model dengan random forest dan tahapan aplikasi dengan *prototype*.

Sumber: Buatan sendiri.

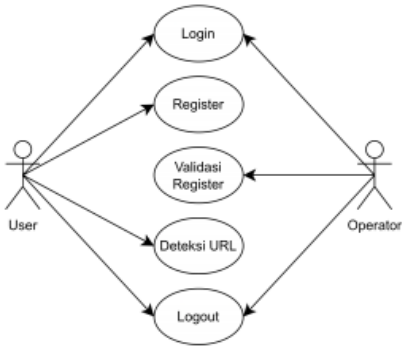
Tahapan penelitian pada Gambar 3 dimulai dari pembuatan model menggunakan algoritma Random Forest dengan perbandingan data 80:20 sehingga diperoleh hasil pelatihan dan pengujian akurasi.

Selanjutnya, tahapan pengembangan aplikasi menggunakan metode *Prototype* yang dimulai dari analisis kebutuhan dan perancangan. Analisis kebutuhan dibagi menjadi dua bagian, yaitu kebutuhan formal dan *non*-formal yang dapat dilihat pada Tabel 2.

Tabel 2. Kebutuhan sistem aplikasi

No	Deskripsi Kebutuhan Sistem	Bagian
1	Sistem menerima *input* berupa URL dari pengguna	Formal
2	Sistem memproses dan menganalisis URL menggunakan model Random Forest	Formal
3	Sistem menampilkan hasil deteksi berupa kategori *phishing* atau tidak *phishing*	Formal
4	Antarmuka aplikasi sederhana, informatif, dan mudah digunakan	*Non*-formal
5	Sistem mampu memberikan hasil deteksi secara cepat dan akurat	*Non*-formal
6	Sistem responsif dalam menerima *input* dan menampilkan hasil deteksi	*Non*-formal

Berdasarkan hasil analisis kebutuhan, disusun rancangan *use case diagram*, *entity relationship diagram*, *activity diagram*, dan antarmuka deteksi.

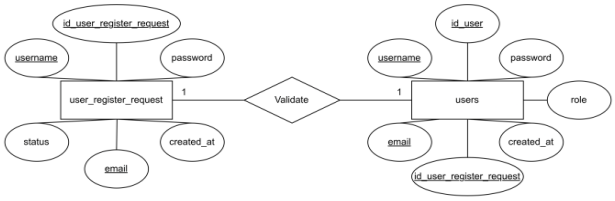


Gambar 4. *Use case diagram*

Keterangan: *Use case diagram* dari aplikasi.

Sumber: Buatan sendiri.

Use case diagram pada Gambar 4 terdiri atas dua aktor, yaitu "user" dan "operator". "User" dapat melakukan login, registrasi, deteksi URL, dan logout, sedangkan "operator" dapat melakukan login, validasi registrasi *user*, dan *logout*.

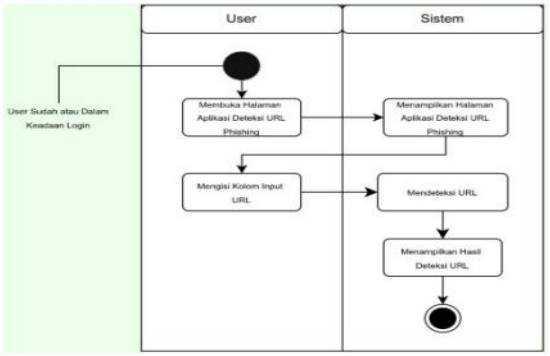


Gambar 5. *Entity relationship diagram*

Keterangan: *Entity relationship diagram* dari aplikasi.

Sumber: Buatan sendiri.

Pada ERD Gambar 5, "user_register_request" berhubungan dengan "users" dengan kardinalitas *one-to-one* (1:1), yang menunjukkan bahwa setiap permintaan registrasi pengguna hanya menghasilkan satu akun "user" setelah divalidasi oleh "operator".

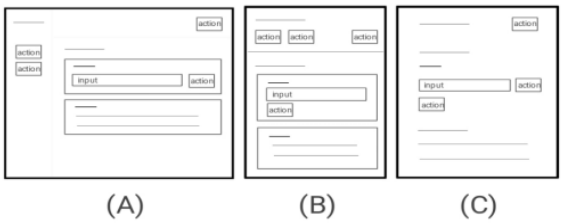


Gambar 6. *Activity diagram*

Keterangan: *Activity diagram* dari proses deteksi pada aplikasi.

Sumber: Buatan sendiri.

Gambar 6 menunjukkan aktivitas deteksi URL oleh "user". Proses dimulai setelah "user" login, memasukkan URL pada halaman deteksi, kemudian aplikasi memproses dan menampilkan hasil deteksi URL yang diperiksa.



Gambar 7. Desain antarmuka aplikasi deteksi

Keterangan: (A) tampilan *browser* desktop, (B) *browser* mobile, dan (C) *extension browser*.

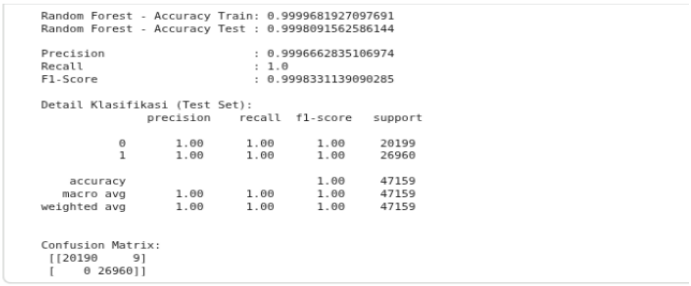
Sumber: Buatan sendiri.

Gambar 7 menunjukkan desain antarmuka aplikasi deteksi berbasis website pada *browser* desktop, *browser* mobile, dan *browser extension*. Ketiganya memiliki fungsi yang sama, yaitu memasukkan URL dan menampilkan hasil deteksi, dengan tampilan yang sederhana dan mudah digunakan.

Penerapan dilakukan dengan membangun aplikasi deteksi URL *phishing* berbasis website menggunakan Flask dan mengintegrasikan model Random Forest. Diakhiri dengan pengujian menggunakan *Unit Testing* dan *Blackbox Testing*.

4. HASIL DAN PEMBAHASAN

Hasil pelatihan dan pengujian model menggunakan algoritma Random Forest dalam mendeteksi URL *phishing* dan tidak *phishing* menunjukkan kinerja yang sangat baik.



```
Random Forest - Accuracy Train: 0.9999681927097691
Random Forest - Accuracy Test : 0.9998091562586144

Precision                     : 0.9996662835106974
Recall                        : 1.0
F1-Score                      : 0.9998331139090285

Detail Klasifikasi (Test Set):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     20199
           1       1.00      1.00      1.00     26960

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


Confusion Matrix:
 [[20190     9]
 [    0 26960]]
```

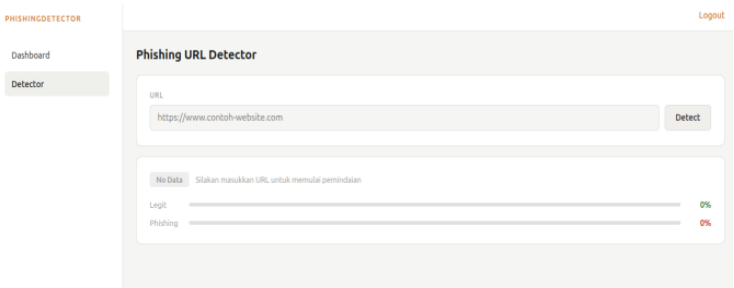
Gambar 8. Pengujian model random forest

Keterangan: Tampilan dari pengujian model dengan Random Forest.

Sumber: Google Colab.

Gambar 8 menjelaskan hasil pelatihan dan pengujian berdasarkan akurasi, di mana model Random Forest memperoleh akurasi sebesar 99,98%, yang menunjukkan kinerja klasifikasi yang sangat baik.

Gambar 9 sampai dengan Gambar 11 menampilkan penerapan aplikasi deteksi menggunakan teknologi Flask yang dikembangkan berdasarkan rancangan pada tahap perancangan dengan metode *Prototype*.

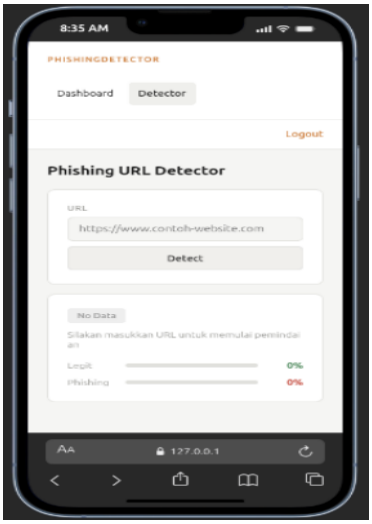


Gambar 9. Tampilan aplikasi pada *browser* desktop

Keterangan: Tampilan aplikasi deteksi URL *phishing* pada *browser* desktop.

Sumber: Buatan sendiri.

Gambar 9 menunjukkan tampilan aplikasi deteksi URL *phishing* pada *browser* desktop yang dirancang dengan antarmuka sederhana dan mudah digunakan.

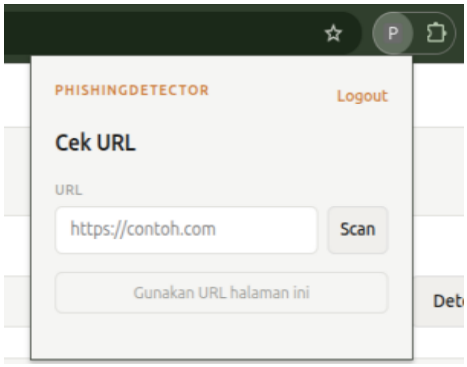


Gambar 10. Tampilan aplikasi pada *browser* mobile

Keterangan: Tampilan aplikasi deteksi URL *phishing* pada *browser* mobile.

Sumber: Buatan sendiri.

Gambar 10 menunjukkan tampilan aplikasi deteksi URL *phishing* pada *browser* mobile yang telah disesuaikan dengan perangkat seluler tanpa mengubah fungsi deteksi URL.



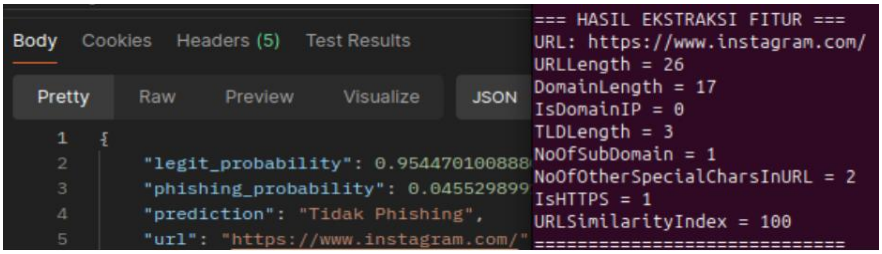
Gambar 11. Tampilan aplikasi pada *extension browser*

Keterangan: Tampilan aplikasi deteksi URL *phishing* pada *extension browser*.

Sumber: Buatan sendiri.

Gambar 11 menunjukkan tampilan aplikasi deteksi URL *phishing* pada *extension browser* yang memungkinkan pengguna melakukan deteksi URL secara lebih praktis.

Setelah penerapan aplikasi selesai, pengujian dilakukan menggunakan *Unit Testing* dan *Blackbox Testing*. Pengujian menggunakan URL “https://www.instagram.com/”.



Gambar 12. Pengujian *unit testing*

Keterangan: Tampilan dari *debugging unit testing* pada suatu URL.

Sumber: Buatan sendiri.

Hasil pengujian pada Gambar 12 menunjukkan bahwa proses deteksi berjalan dengan baik dan menghasilkan prediksi "Tidak *Phishing*". Detail pengujian *Unit Testing* disajikan pada Tabel 3.

Tabel 3. *Unit testing*

No	Komponen yang Diuji	Hasil
1	URLLength	26
2	DomainLength	17
3	IsDomainIP	0
4	TLDLength	3
5	NoOfSubDomain	1
6	NoOfOtherSpecialCharsInURL	2
7	IsHTTPS	1
8	URLSimilarityIndex	100
9	Legit Probability	0.95..
10	Phishing Probability	0.04..
11	Prediction	Tidak *Phishing*
12	Status Pengujian	Berhasil

Tabel 3 menunjukkan hasil *Unit Testing* pada URL "https://www.instagram.com/". Hasil pengujian menunjukkan bahwa proses ekstraksi fitur berhasil menghasilkan delapan fitur yang dibutuhkan model Random Forest, yaitu "URLLength 26", "DomainLength 17", "IsDomainIP 0", "TLDLength 3", "NoOfSubDomain 1", "NoOfOtherSpecialCharsInURL 2", "IsHTTPS 1", dan "URLSimilarityIndex 100". Berdasarkan fitur tersebut, model menghasilkan nilai "Legit Probability sebesar 0,95.." dan "Phishing Probability sebesar 0,04.." sehingga URL dideteksi sebagai "Tidak *Phishing*". Dengan demikian, pengujian dinyatakan berhasil karena proses ekstraksi fitur dan klasifikasi berjalan sesuai harapan.

Selanjutnya, pengujian *Blackbox Testing* dilakukan dengan 3 *test case* yang berfokus pada fungsionalitas deteksi URL. Hasil pengujian disajikan pada Tabel 4.

Tabel 4. *Blackbox testing*

No	Skenario Pengujian	*Test Case*	Hasil yang diharapkan	Hasil Pengujian	Ket.
1	Tidak meng*input*kan data (URL)		Sistem akan memberitahu bahwasanya data kosong		Berhasil
2	Meng*input*kan data (URL kategori aman)		Sistem akan klasifikasi data (URL kategori tidak *phishing*)		Berhasil
3	Meng*input*kan data (URL berpotensi berbahaya)		Sistem akan klasifikasi data (URL kategori *phishing*)		Berhasil

Tabel 4 menunjukkan bahwa seluruh skenario pengujian berjalan sesuai harapan. Pada skenario pertama, sistem memberikan notifikasi ketika URL tidak di*input*kan. Pada skenario kedua, sistem mendeteksi URL aman sebagai "Tidak *Phishing*". Pada skenario ketiga, sistem mendeteksi URL berbahaya sebagai "*Phishing*". Berdasarkan hasil tersebut, fungsi deteksi URL pada aplikasi telah berjalan sesuai kebutuhan yang dirancang.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa pembuatan aplikasi deteksi URL *phishing* berbasis website menggunakan metode *Prototype* dan algoritma Random Forest dapat berjalan serta diimplementasikan dengan baik. Hal ini didukung oleh hasil model algoritma Random Forest dalam mendeteksi URL yang memperoleh akurasi sebesar 99,98%. Selain itu, pengujian aplikasi menggunakan *Unit Testing* dan *Black Box Testing* juga berhasil diimplementasikan. Pada pengujian *Unit Testing* dengan URL "https://www.instagram.com/", aplikasi mampu mengekstraksi fitur-fitur URL dengan baik serta sistem menghasilkan nilai "Legit Probability" sebesar "0,95.." dan "Phishing Probability" sebesar "0,04.." sehingga URL diklasifikasikan sebagai "Tidak *Phishing*". Selanjutnya, pengujian *Black Box Testing* yang terdiri dari 3 *test case* dengan fokus pada fungsionalitas deteksi menunjukkan bahwa seluruh *test case* berhasil dijalankan serta

menghasilkan keluaran yang sesuai dengan hasil yang diharapkan. Secara keseluruhan, aplikasi yang dikembangkan telah berjalan sesuai dengan hasil yang diharapkan.

Sebagai saran untuk penelitian selanjutnya, aplikasi deteksi URL *phishing* dapat dikembangkan dalam bentuk aplikasi mobile agar lebih mudah diakses dan digunakan secara langsung. Selain itu, aplikasi juga dapat diintegrasikan dengan sistem keamanan jaringan guna meningkatkan pemanfaatan hasil penelitian.

DAFTAR REFERENSI

Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science, 3*, Article 563060. https://doi.org/10.3389/fcomp.2021.563060

Andika, I. (2024). *Perhitungan pohon keputusan* [Video]. YouTube. https://www.youtube.com/watch?v=Ao_0FUSgqpo

Anti-Phishing Working Group. (2024). *Laporan statistik web phishing tahun 2024*. https://apwg.org/

Cyber Security News. (2024). *Top 10 cyber security news 2024*. https://cybersecuritynews.com/

Effendy, V., Martha, A., Sabariah, M., & Afgani, F. (2025). *Pengantar rekayasa perangkat lunak*. Percetakan Diandra.

Fahri, M. (2025). Penerapan algoritma Random Forest untuk deteksi phishing pada website. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 6*(2), 186–194.

Kadir, A. (2023). *Membuat aplikasi web dengan PHP dan database MySQL*. Andi.

Larasati, I., Yusril, A. N., & Zukri, P. A. (2021). Systematic literature review analisis metode Agile dalam pengembangan aplikasi mobile. *SISTEMASI, 10*(2), 369. https://doi.org/10.32520/stmsi.v10i2.1237

Opara, C., Chen, Y., & Wei, B. (2024). Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Systems with Applications, 236*, 121183. https://doi.org/10.1016/j.eswa.2023.121183

Pantiwati, Y., Permana, F., Aminudin, & Sari, T. (2024). *Prototype e-modul model pembelajaran LI-PRO-GP*. UMM Press.

Putra, Y., Dawis, A., Novi, Natsir, F., Fitria, Widhiyanti, A., Hasan, F., Somantri, & Maniah. (2023). *Pengantar aplikasi mobile*. Widina.

Rahmaddeni, Wulandari, D., Aziz, S., Adrianto, S., & Pratiwi, F. (2025). *Teori dan implementasi machine learning menggunakan Python*. Serasi Media Teknologi.

Safi, A., & Singh, S. (2023). A systematic literature review on phishing website detection techniques. *Journal of King Saud University – Computer and Information Sciences, 35*(2), 590–611. https://doi.org/10.1016/j.jksuci.2023.01.004

Serrano, P., & Emonet, V. (2022). The FAIR extension: A web browser extension to evaluate digital object FAIRness. *Research Ideas and Outcomes, 8*, e95006. https://doi.org/10.3897/rio.8.e95006

Wahyuni, T. (2023). *Aplikasi "Si Pintar" untuk pencegahan COVID-19*. NEM.