JURNAL SAINS DAN INFORMATIKA
RESEARCH OF SCIENCE AND INFORMATIC V9.11

Vol.09 No.01(2023) 8-14 http://publikasi.lldikti10.id/index.php/jsi

Vol.09 No.01(2023) 1-7 http://publikasi.lldikti10.id/index.php/jsi

Klasifikasi Malicious URL Menggunakan Algoritma Improved Random Forest dan Random Forest Berbasis Web

Octavan Adiputra[a] , Eman Setiawan[b]
[ab]Program Studi Sistem Infomrasi, Universitas Narotama, Surabaya
adiputraoctavan@gmail.com, eman.setiawan@narotama.ac.id

Submitted: 21-07-2022, Reviewed: 29-08-2022, Accepted 29-11-2022
http://doi.org/10.22216/jsi.v9i1.1378

Abstract

URLs are very much on the network of computer systems. Moreover, nowadays all activities use an online system. Starting from social media, and marketplaces to group chat applications. An early prevention system from malicious URL attacks is needed to counteract the large number of URLs circulating in the online system. Previously, malicious URL detection based on Blacklisting and Heuristic URLs could not recognize the new type of malicious URL without first being analyzed. For this reason, a technique is needed to detect malicious URLs using machine learning. The lack of machine learning in the detection of malicious URLs is that it is not 100% able to detect malicious URLs precisely. This study will use an improved random forest approach with a random forest as a classifier to detect malicious URLs. Improved Random Forest is a Random Forest that is used using evaluator features and filter instances to improve the accuracy of ordinary random forests. This study concluded that both methods of improved random forest and ordinary random forest have an accuracy value above 99%.

Keywords: *malicious URL, improved random Forest, random forest*

Abstrak

URL sangat banyak berada pada jaringan sistem komputer. Apalagi saat ini semua kegiatan menggunakan sistem online. Mulai dari media sosial, marketplace hingga aplikasi chatting grup. Untuk menangkal banyaknya *URL* yang beredar di sistem online tersebut, maka dibutuhkan sistem pencegahan dini dari serangan *URL* berbahaya. Sebelumnya deteksi *malicious URL* berbasis *Blacklisting* dan *URL Heuristic* tidak dapat mengenali malicious *URL* jenis baru tanpa di analisis terlebih dahulu. Untuk itu diperlukan teknik mendeteksi *malicious URL* menggunakan *machine leaning*. Kekurangan *machine learning* dalam pendeteksian *malicious URL* bahwa tidak 100% dapat mendeteksi *malicious URL* secara tepat. Pada penelitian ini akan digunakan pendekatan *improved random forest* dengan *random forest* sebagai *classifier* untuk mendeteksi *malicious URL*. *Improved Random Forest* merupakan *Random Forest* yang dipakai menggunakan *feature evaluator* dan *instance filter* untuk meningkatkan akurasi dari *random forest* biasa. Penelitian ini menghasilkan kesimpulan bahwa kedua metode baik *improved random forest* maupun *random forest* biasa memiliki nilai akurasi diatas 99%.

Kata kunci: *malicious URL*, *improved random Forest, random forest*

1. Pendahuluan

Internet menjadi hal yang penting dan signifikan terhadap kehidupan kita sehari-hari. Banyak layanan yang dapat dilakukan internet yang bergantung pada fungsionalitas dan keamanannya, misalnya bisnis, pembelajaran, perbankan, jejaring sosial, kesehatan dan banyak lainnya yang merupakan aplikasi berbasis web[1]. Web menjadi semakin penting, penjahat dunia maya secara ilegal dapat megeksploitasi kerentanan dan memiliki peluang untuk melakukan banyak serangan terhadap aplikasi *web[2]*.

.

Laporan keamanan *Symantec* menguraikan tentang berbagai ancaman global yang mencakup data perusahaan, pelanggaran, serangan terhadap situs web dan berbagai kegiatan lainnya[3]. Laporan tersebut juga mengungkapkan bahwa di masa *pandemic* ini, penjahat dunia maya telah memanfaatkan krisis kesehatan untuk meningkatkan serangan siber terhadap rumah sakit, fasilitas kesehatan dan penelitian medis serta terhadap personel medis dan organisasi kesehatan masyarakat internasional[3].

Pendekatan machine learning dapat menjadi solusi dari permasalahan yang telah diuraikan diatas. Dan algoritma klasifikasi yang digunakan adalah *random forest* yang digabungkan dengan pemilihan fitur (*feature selection*) untuk memilih fitur yang relevan dengan *malicious URL,* dan dengan pendekatan *random sampling* untuk mengatasi *imbalance dataset*. Pendekatan ini dinamakan *improved random forest*[4].

Beberapa penelitian tentang *machine learning* untuk melakukan kasifikasi *malicious URL,* yaitu penelitian yang dilakukan oleh Tao dkk[5], tentang penggunaan machine learning untuk mengklasifikasi situs web berbahaya dengan mengumpulkan informasi berdasarkan sesi *HTTP* dan fitur berbasis domain. Dari penelitian ini akurasi mencapai 92,2%. Kemudian penelitian yang dilakukan oleh Sirageldin dkk[6], yang meenggunakan 2 fitur yaitu, fitur leksikal dan fitur berbasis konten. Dari penelitian tersebut akurasi mencapai 96%. Lalu penelitian yang dilakukan oleh Altaher[7], dimana beliau melakukan klasifikasi web *phising* dengan menggunakan metode *SVM* dan *KNN*, dan akurasi yang diperoleh sebesar 90,04%. Kemudian penelitian yang dilakukan oleh Cui dkk[8], yang menggunakan analisis statistik dan level *sigmoidal* dalam pemilihan fiturnya serta penggunaan *Naïve Bayes, Decision Tree* dan *SVM* untuk metode klasifikasinya, penelitian ini mendapatkan akurasi sebesar 98,7%. Dan yang terakhir yaitu penelitian yang dilakukan oleh Liu dkk[9], dimana beliau menggunakan 6 teknik klasifikasi dari proses *machine learning,* dan hasilnya Teknik Random Foresr yang memiliki akurasi paling tinggi.

Tujuan dari penelitian ini adalah melakukan klasifikasi malicious URL dengan pendekatan *machine learning* yang menggunakan fitur leksikal dan fitur berbasis host pada web serta metode klasifikasi yang digunakan adalah *Random Forest* yang akan dibandingkan dengan *Random Forest* yang ditambah dengan pemilihan fitur dan mengatasi imbalance data dengan teknik sampling yang bisa disebut juga sebagai teknik *Improved Random Forest* dengan harapan dapat memperbaiki akurasi dan performa model *machine learning*.

2. Tinjauan Pustaka

2.1 *Website*

Website merupakan halaman yang berisi informasi yang diakses melalui jaringan internet diseluruh dunia[10].

2.2 *Features Representation*

Ini merupakan sekumpulan informasi dari sebuah *URL* agar dapat dikenali dan memberikan informasi yang berguna untuk mendukung proses pengenalan terhadap *URL* yang dianalisis.

2.3 *Lexical Features*

Fitur Leksikal adalah fitur yang diperoleh dari nama *URL* itu sendiri. Berdasarkan tampilan *URL* memungkinkan untuk mengidentifikasi apakah *URL* berbahaya atau tidak. Fitur leksikal tidak cukup dalam menentukan bahwa web tersebut berbahaya atau tidak, tetapi harus digunakan bersama fitur lainnya seperti fitur host, fitur nama dan fitur konten[11]. Fitur Leksikal terbagi atas dua kategori yaitu, fitur leksikal tradisional dan fitur leksikal lanjutan. Fitur leksikal tradisional mencakup property umum yang dimiliki *URL* sendiri seperti, panjang *URL*, jumlah titik didalamnya, jumlah karakter khusus, panjang nama domainnya, protokol yang digunakan, *TLD* yang digunakan dll.

2.4 *Host Based Features*

Fitur berbasis host dapat menginformasikan dimana situs web dihosting yaitu, negara, lokasi, waktu hosting, tidak hanya itu, kita juga bisa mengetahui siapa pemilik website, pembuat website dan bagaimana website tersebut dikelola[11]. Inilah beberapa properti dari fitur berbasis host yang diidentifikasi oleh hostname dari URL.

1. *IP Address Properties*

   Ini menjelaskan fitur alamat *IP URL*. Alamat *IP* adalah sebuah set dari set 0s dan 1s, dan itu terbuat dari 32 bit. Setiap 4(empat) set terdiri dari 8 bit. Properti alamat *IP* menginformasikan apakah alamat *IP* digunakan dalam *URL*.
2. *WHOIS Properties*

   Kata *WHOIS* menunjukkan siapa yang bertanggung jawab atas nama domain, ini menunjukkan siapa yang menciptakan domain, di Negara mana web terbuat, kapan waktu pembuatannya dll. Properti ini menunjukkan informasi mengenai nama domain.
3. *Domain Name Properties*

   Nama domain digunakan untuk mengidentifikasi alamat IP, misalnya,nama domain google.com memiliki selusin alamat IP. Nama domain digunakan dalam URL untuk mengidentifikasi halaman web tertentu.
4. *Geographic Properties*

   Ini menunjukkan lokasi alamat IP, yaitu di benua, Negara atau kota mana alamat IP tersebut berada.

2.5 *Feature Selection*

Fitur Selection merupakan sebuah metode untuk mengidentifikasi fitur yang berkaitan dan tidak berkaitan dari sebuah dataset[12]. Kegunaan dari fitur evaluator ialah untuk meningkatkan performa dari machine learning. Metode ini juga digunakan sebagai data reduction agar proses komputasi menjadi lebih cepat[4]. Peneliti disini menggunakan *Feature Importance* sebagai Teknik pemilihan fiturnya

2.6 *Machine Learning Approach*

Pendekatan ini menganalisis informasi yang berbeda mengenai URL dan halaman webnya. Teknik ini menganalisis beberapa informasi dari nama, alamat IP, nama domain, nama host dll. Informasi ini dikenal dengan nama fitur, kemudain fitur inilah yang digunakan untuk melatih model dan model tersebut akan diumpankan ke model klasifikasi, lalu model klasifikasi akan memprediksi apakah *URL* termasuk yang berbahaya atau jinak (*benign*)[13][14].

2.7 *Random Forest*

Random Forest merupakan algoritma dalam klasifikasi data supervised, menggabungkan beberapa *tree*, yang masing-masing dilatih secara terpisah dimana model dasar dilath dan dikombinasikan menggunakan skema pembobotan yang canggih, biasanya tree dilatih secara independen dan prediksi *tree* digabungkan melalui rata-rata[15].

2.8 *Improved Random Forest*

Metode *improved random forest* berdasar pada metode *random forest* biasa dengan menambahkan fitur evaluator dan mengatasi imbalance data dengan teknik sampling. Fitur evaluator digunakan untuk memilih fitur yang relevan. Sampling digunakan untuk mengatasi data agar akurasi dapat semakin meningkat. Arsitektur dari metode improved *random forest*[4].

2.9 *Evaluasi Model*

Evaluasi Model adalah rangkaian tahapan dalam melakukan proses machine learning dimana dalam proses ini akan diperoleh hasil daripada model klasifikasi yang telah dilakukan sebelumnya. Pada proses ini akan diperoleh beberapa data angka yakni nilai *Precision, Recall* dan *F-1 Score* yang dapat dilihat pada persamaan 1, 2, dan persamaan 3.

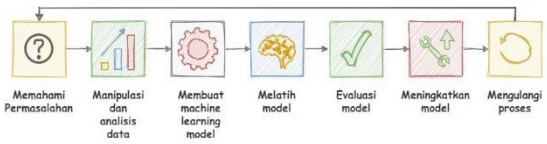
$$Precision(t) = 100x\frac{|S(t) \cap G|}{|S(t)|} \quad (1)$$

$$Recall(t) = 100x\frac{|S(t) \cap G|}{|G|} \quad (2)$$

$$F1(t) = \frac{2\ x\ precision(t)\ x\ recall(t)}{precision(t) + recall(t)} \quad (3)$$

3. Metodologi Penelitian

Metodologi penelitian pada penelitian ini akan mencakup beberapa hal seperti gambar dibawah ini.



Gambar 1. Tahapan dalam proses *machine learning*

Tahapan proses *machine learning* adalah sebagai berikut.

1) Memahami permasalahan
2) Manipulasi dan analisis data
3) Membuat model machine learning
4) Evaluasi model
5) Meningkatkan model
6) Mengulangi proses

Langkah langkah dalam proses machine learning adalah sebagai berikut.

1. Memahami permasalahan
   Memahami permasalahan adalah tahap awal dimana kita melakukan proses pengumpulan dataset yang akan kita olah. Dataset yang digunakan adalah dataset URL jinak (benign) dan URL berbahaya.
2. Manipulasi dan analisis data
   Pada tahap ini dilakukan proses manipulasi data seperti penghilangan duplikasi data, menghilangkan data yang berisi *null*. Sehingga menjadi data yang bersih dari segala *noise.*
3. Membuat model *machine learning*
   Pada tahap ini akan dilakukan proses menyusun model machine learning yang diinginkan, mulai dari mengekstrak fitur data URL, melatih data menjadi sebuah data training dan mengatur data menjadi data test. Data training yang digunakan adalah sebesar 80% dari jumlah keseluruhan data, sedangkan data test sebesar 20% dari jumlah keseluruhan data.
4. Evaluasi Model
   Pada tahap ini akan dilakukan proses perhitungan performa dari suatu model yang telah kita atur sebelumnya, mulai dari menghitung *confusion matrix* (*recall, precision, f1 score*), menghitung akurasi dengan model *kfold-cross validation* dengan ketentuan *fold* sebanyak 5 *fold.*

Dari beberapa tahapan diatas jika hasil dari cross validation sebanyak 5 fold belum stabil, maka akan dilakukan proses ulang dengan cara mengatur kembali model yang dibuat agar menghasilkan nilai fold yang stabil.

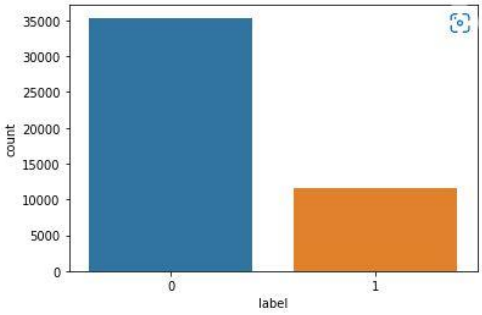
4. Hasil dan Pembahasan

Hasil dari penelitian ini menghasilkan beberapa pembahasan sebagai berikut.

4.1. Perancangan model

4.1.1. Dataset

Dataset yang digunakan pada penelitian ini adalah dataset *URL* jenis *malware* dan *URL benign* (jinak) yang diperoleh dari dari website *UNB* (*University of New Braunswick*) yang telah dikumpulkan menjadi suatu kumpulan *URL* yang berisi *URL* benign sebanyak 35.378 *URL* dan *URL malware* 11.566[16].



Gambar 2. Jumlah Dataset *URL Malware* dan *URL Benign*

Pada gambar 2 ditunjukkan jumlah dataset kedua class yakni URL jinak (benign) yang diberi label 0 dan URL malware yang diberi label 1.

4.1.2. Data *Preprocessing*

Dari dataset diatas, dengan proses data *preprocessing* yang dilakukan dengan menghilangkan *duplicate* data dan data yang benilai *null*.

Kode untuk melakukan penghilangan *duplicate* data dapat dilihap pada gambar 3 dibawah ini.

```
In [10]: df.duplicated().sum()
Out[10]: 0

In [9]: # remove duplicated values
        df = df.drop_duplicates(keep='first')
```

Gambar 3. Teknik data Preprocessing

4.2. Implementasi Model

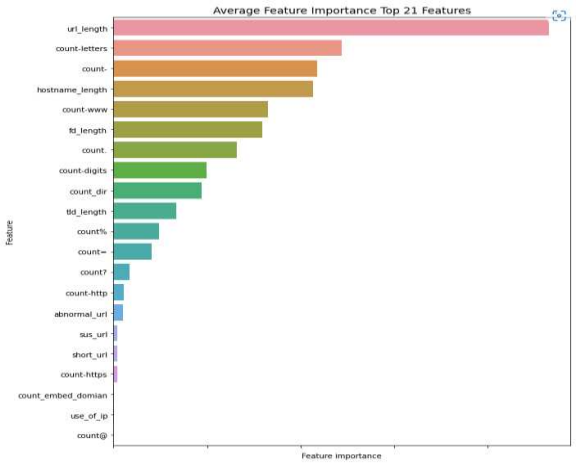
4.2.1. Seleksi fitur (*Feature Selection*)

Berikut ini fitur yang akan dijadikan acuan untuk mengklasifikasikan URL benign dan malware.

Tabel 1 Nama Fitur untuk klasifikasi *URL*

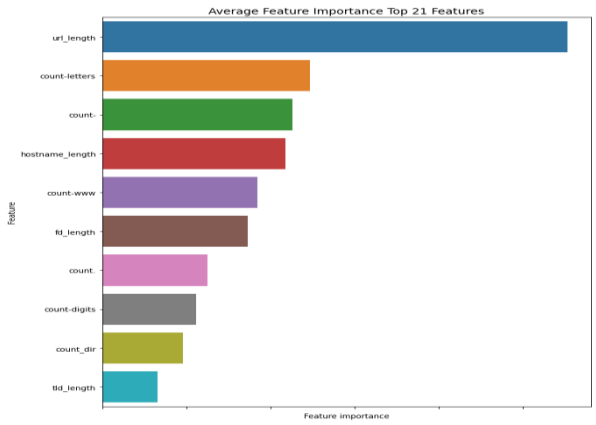
Nama Fitur	RF Feature Importance
Use_of_ip	0.000119
Abnormal_url	0.004984
count	0.064599
Count-www	0.085412
count@	0.000074
Count_dir	0.048153
Count embed domain	0.000562
Short url	0.001938
Count https	0.002191
Count-http	0.005585
Count%	0.021294
Count?	0.006889
Count-	0.102850
Count=	0.023673
url_length	0.234340
Hostname_length	0.107529
Sus url	0.001852
Fd_length	0.082994
Tld_length	0.034822
Count-digits	0.051400
Count-letter	0.118739

Tabel 1 diatas merupakan nama-nama fitur yang akan digunakan untuk mngklasifikasikan *class URL*. Yang mana telah dighitung nilai *feature importance* nya.



Gambar 4. Nama fitur untuk algoritma random forest

Pada gambar 4 diatas merupakan fitur yang akan digunakan saat menggunakan algoritma *Random Forest* yang berjumlah 21 fitur.
Dari fitur-fitur *URL* diatas nantinya akan dipilih berdasarkan nilai *feature importance* yang paling tinggi ke yang rendah. Dari fitur diatas akan dipilih jumlah fitur sebanyak 10 fitur. Berikut 10 fitur yang memiliki nilai *feature importance*.



Gambar 5. Nama fitur untuk algoritma *improved random forest*

pada gambar 5 diatas merupakan fitur yang telah dilakukan proses *feature selection*. Jumlah fitur yang telah disaring diambil sebanyak 10 fitur.

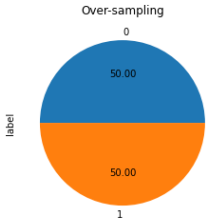
4.2.2. *Class Balancing*

Jika kita melihat jumlah dataset yang ada, jumlahnya sangat tidak seimbang (*imbalance*), maka dari itu akan dilakukan proses balance dataset agar kedua class tersebut seimbang. Teknik yang digunakan untuk menyeimbangkan jumlah class adalah *Random Oversampling*.

```
In [21]: from imblearn.over_sampling import RandomOverSampler

         #ros = RandomOverSampler(sampling_strategy=1) # Float
         ros = RandomOverSampler(sampling_strategy="not majority") # String
         X_train_ros, y_train_ros = ros.fit_resample(X_train, y_train)

         ax = y_train_ros.value_counts().plot.pie(autopct='%.2f')
         _ = ax.set_title("Over-sampling")
```



Gambar 6. Teknik *class balancing Random Over Sampling*

Pada gambar 6 merupakan proses *class balancing* yang dilakukan untuk menyeimbangkan jumlah *class* .

4.3. Uji Coba dan Evaluasi Model

Pada tahap ini akan ditampilkan performa dari suatu metode klasifikasi yang telah dilakukan uji coba dengan data training 80% dan data testing 20%.

4.3.1. Accuracy

Tabel 2. Hasil uji coba dengan *Kfold Cross Validation*

Metode	**Fold1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**
	Acc	**Acc**	**Acc**	**Acc**	**Acc**
Random Forest	100%	100%	99,96 %	99,96 %	99,96 %
Improved Random Forest	99,35 %	99,39 %	99,92 %	99,92 %	99,92 %

Pada tabel 2 menunjukkan hasil testing yang menggunakan *kfold cross validation* dengan 2 metode klasifikasi yang dilakukan. Terlihat dari ketiga metode klasifikasi bahwa dengan percobaan sebanyak 5 fold ketiga metode ini sama-sama memiliki nilai akurasi yang tinggi.

Tabel 3. Hasil rata-rata akurasi dari 5 kali percobaan

Metode	Rata-rata akurasi
Random Forest	99,99%
Improved Random Forest	99.93%

Pada tabel 3 merupakan rata-rata akurasi dari kedua metode klasifikasi *malicious URL*.

4.3.2. *Confusion Matrix*

Tabel 4. *Confusion Matrix* data testing model

Metode	Confusion Matrix	
Random Forest	7047	6
	18	543
Improved Random Forest	7048	5
	17	544

Pada tabel 4 dapat dilihat terdapat dua kolom yang berwarna pink yang menandakan berapa banyak URL yang salah dideteksi oleh model *machine learning* dan kolom dua kolom yang berwarna biru menandakan bahwa berapa banyak *URL* yang benar dideteksi oleh model *machine learning*. Dimana pada tabel diatas untuk model *Random Forest* jumlah yang benar mendeteksi sebagai URL *benign* (jinak) sebanyak 7047 record, sedangkan benar mendeteksi *URL malware* sebeanyak 543 *record*. Dan sebanyak 18 *URL* yang salah

mendeteksi *URL* jinak dan sebanyak 6 *URL* yang salah mendeteksi sebagai *URL malware*.

Selanjutnya untuk model klasifikasi *Improved Random Forest* model berhasil mendeteksi sebanyak 7048 *URL* jinak dan 544 sebagai *URL malware*. Sementara sebanyak 5 *URL* yang salah deteksi sebagai *URL malware* dan sebanyak 17 *URL* salah deteksi sebagai URL jinak.

4.3.3. Classification Report

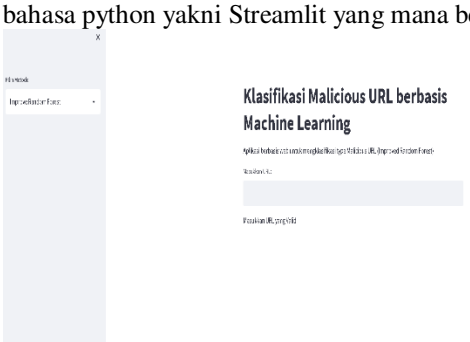
Tabel 5. *Classification report* kedua metode

Metode	Precision	Recall	F1 Score
Random Forest	99%	98%	99%
Improved Random Forest	99%	98%	99%

Pada tabel 5 dapat dilihat bahwa dari kedua model klasifikasi diatas, menghasilkan nilai precision, recall dan F1 Score yang sama.

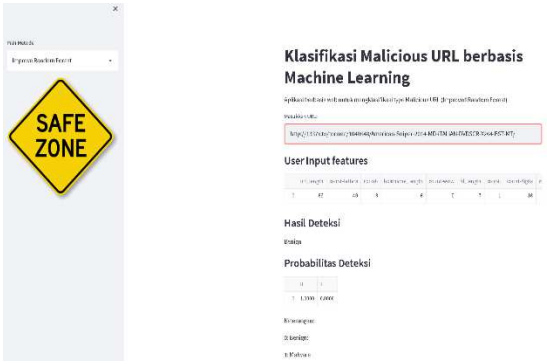
4.4. *Graphical User Interface* (GUI)

Graphical User Interface (GUI) dari penerapan improved random forest untuk mendeteksi malware URL yang dilakukan pada penelitian ini. Web GUI dari aplikasi ini dihasilkan dengan web framework untuk bahasa python yakni Streamlit yang mana berbasis web.



Gambar 5. Tampilan web aplikasi *malicious URL*

Pada gambar 6 dapat dijelaskan bahwa pada tampilan awal web klasifikasi malicious URL terdiri dari sidebar disisi kiri untuk memilih metode klasifikasinya. Dan di bagian tengah (container) terdapat input URL yang ingin di masukkan.



Gambar 6. Tampilan web saat menjalankan klasifikasi *URL*

Pada gambar 7 dapat dijelaskan bahwa pada tampilan saat web dijalankan maka akan muncul beberapa informasi diantaranya yakni, user input features adalah informasi fitur yang diekstrak dari *URL* yang dimasukkan. Selanjutnya ada hasil deteksi yang menunjukkan *URL* diatas adalah *URL* jinak. Selain itu ada informasi hasil deteksi dan probabilitas deteksi yang merupakan besarnya probabilitas deteksi.

5. Kesimpulan

Kesimpulan yang diperoleh dari hasil pengujian diatas adalah pertama hasil akurasi dari kedua algoritma diatas kedua algoritma sama-sama memiliki akurasi diatas 99%.

Penelitian ini masih bisa dikembangkan lagi dengan menambahkan fitur berbasis konten sebagai informasi pendukung dalam mengklasifikasi *malicious URL*.

6. Daftar Pustaka

[1] F. Alkhudair, M. Alassaf, R. Ullah Khan, and S. Alfarraj, "Detecting Malicious URL," *2020 International Conference on Computing and Information Technology, ICCIT 2020*, pp. 0–4, 2020, doi: 10.1109/ICCIT-144147971.2020.9213792.

[2] D. Stevanovic, N. Vlajic, and A. An, "Unsupervised Clustering of Web Sessions to Detect Malicious and Non-malicious Website Users," *Procedia Comput Sci*, vol. 5, pp. 123–131, 2011, doi: 10.1016/j.procs.2011.07.018.

[3] B. Stackpole, "Red Cross to World Governments: Do More to Stop attacks on Healthcare Orgs," 2020.

[4] A. Chaudhary, S. Kolhe, and R. Kamal, "An improved random forest classifier for multi-class classification," *Information Processing in Agriculture*, vol. 3, no. 4, pp. 215–222, 2016, doi: 10.1016/j.inpa.2016.08.002.

[5] T. Wang, S. Yu, and B. Xie, "A novel framework for learning to detect malicious web pages," in *Proceedings - 2010 International Forum on Information Technology and Applications, IFITA 2010*, 2010, vol. 2, pp. 353–357. doi: 10.1109/IFITA.2010.173.

[6] A. Sirageldin, B. B. Baharudin, and L. T. Jung, "Malicious web page detection: A machine learning approach," in *Lecture Notes in Electrical Engineering*, 2014, vol. 279 LNEE, pp. 217–224. doi: 10.1007/978-3-642-41674-3_32.

[7] A. Altaher, "Phishing Websites Classification using Hybrid SVM and KNN Approach," 2017. [Online]. Available: www.ijacsa.thesai.org

[8] B. Cui, S. He, X. Yao, and P. Shi, "Biographical notes: Baojiang Cui received his BS in the Hebei University of Technology, China, in 1994, MS in the Harbin Institute of Technology, China, in 1998 and PhD in Control Theory and," 2018.

[9] C. Liu, L. Wang, B. Lang, and Y. Zhou, "Finding effective classifier for malicious URL detection," in *ACM International Conference Proceeding Series*, Jan. 2018, pp. 240–244. doi: 10.1145/3180374.3181352.

[10] "Sistem Sistem Informasi Manajemen Surat Berbasis Website di STMIK Pringsewu," *Jurnal Sains*

dan Informatika, vol. 7, no. 1, pp. 17–22, Mar. 2021, doi: 10.22216/jsi.v7i1.340.

[11] S. M. Nair, "Detecting Malicious URL using Machine Learning: A Survey," *Int J Res Appl Sci Eng Technol*, vol. 8, no. 5, pp. 2670–2677, May 2020, doi: 10.22214/ijraset.2020.5447.

[12] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset," *Expert Syst Appl*, vol. 38, no. 5, pp. 5947–5957, May 2011, doi: 10.1016/j.eswa.2010.11.028.

[13] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler," in *Proceedings of the 20th international conference on World wide web - WWW '11*, 2011, p. 197. doi: 10.1145/1963405.1963436.

[14] B. Eshete, A. Villafiorita, and K. Weldemariam, "BINSPECT: Holistic Analysis and Detection of Malicious Web Pages," 2013, pp. 149–166. doi: 10.1007/978-3-642-36883-7_10.

[15] M. Denil, D. Matheson, and N. de Freitas, "Narrowing the Gap: Random Forests In Theory and In Practice," Oct. 2013, [Online]. Available: http://arxiv.org/abs/1310.1415

[16] UNB, "Canadian Institute for Cybersecurity," 2016.