Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 2 April 2025, pp: 531-540
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Application of Classification Algorithms in Machine Learning for Phishing Detection

Penerapan Algoritma Klasifikasi pada Machine Learning untuk Deteksi Phishing

Rizky Fauzan[1], Anik Vega Vitianingsih[2*], Dwi Cahyono, Anastasia Lidya Maukar[4], Yoyon Arie Budi Suprio[5]

[1,2,3]Informatics Department, Universitas Dr. Soetomo, Surabaya, Indonesia
[4]Industrial Engineering Department, President University, Bekasi, Indonesia
[5]Informatics Department, STIKOM PGRI Banyuwangi, Indonesia

E-Mail: [1]fauzanriz497@gmail.com, [2]vega@unitomo.ac.id, [3]dwikk@unitomo.ac.id, [4]almaukar@president.ac.id, [5]yoyonstikom@gmail.com

Received Jan 11th 2025; Revised Feb 26th 2025; Accepted Mar 6th 2025; Available Online Mar 12th 2025, Published Mar 12th2025
Corresponding Author: Anik Vega Vitianingsih

Abstract

Phishing is a form of cybercrime aimed at stealing sensitive information through fraudulent methods, such as fake websites that mimic official pages. Therefore, a more accurate and efficient detection system is needed to identify this threat. This study aims to analyze the application of classification algorithms in machine learning for phishing URL detection. The algorithms used in this research are Naïve Bayes, Random Forest, and Decision Tree, applied to a dataset collected from various sources. This dataset is analyzed using Term Frequency - Inverse Document Frequency (TF-IDF) based features and numerical features such as URL length, the number of digits, special characters, and the presence of keywords commonly found in phishing sites. The model evaluation is conducted using accuracy, precision, recall, and F1-score metrics to measure the effectiveness of the developed detection system. Experimental results show that the Random Forest model performs best with an accuracy of 97.2%, followed by Decision Tree (96.3%), while Naïve Bayes has a lower accuracy of 85.3%. The Random Forest model also maintains a good balance between precision and recall, making it more reliable in detecting phishing URLs. The use of machine learning algorithms has been proven to significantly enhance phishing detection effectiveness.

Keywords: Cyber Security, Decision Tree, Naïve Bayes, Phishing, Phishing URL Detection, Random Forest

Abstrak

Phishing merupakan salah satu bentuk kejahatan siber yang bertujuan mencuri informasi sensitif melalui metode penipuan, seperti situs web palsu yang menyerupai halaman resmi. Maka diperlukan sistem deteksi yang lebih akurat dan efisien untuk mengidentifikasi ancaman ini. Penelitian ini bertujuan untuk menganalisis penerapan algoritma klasifikasi dalam machine learning guna mendeteksi URL phishing. Algoritma yang digunakan dalam penelitian ini adalah Naïve Bayes, Random Forest, dan Decision Tree, yang diterapkan pada dataset yang dikumpulkan dari berbagai sumber. Dataset ini dianalisis menggunakan fitur berbasis Term Frequency - Inverse Document Frequency (TF-IDF) serta fitur numerik, seperti panjang URL, jumlah angka, karakter khusus, dan keberadaan kata kunci yang sering ditemukan dalam situs phishing. Evaluasi model dilakukan menggunakan metrik akurasi, precision, recall, dan F1-score untuk mengukur efektivitas sistem deteksi yang dikembangkan. Hasil eksperimen menunjukkan bahwa model Random Forest memiliki performa terbaik dengan akurasi mencapai 97,2%, diikuti oleh Decision Tree (96,3%), sementara Naïve Bayes memiliki akurasi lebih rendah (85,3%). Model Random Forest juga memiliki keseimbangan yang baik antara precision dan recall, sehingga lebih andal dalam mendeteksi URL phishing. Penggunaan algoritma Machine Learning terbukti dapat meningkatkan efektivitas deteksi phishing secara signifikan.

Kata Kunci: Cyber Security, Decision Tree, Deteksi URL Phishing, Naïve Bayes, Phishing, Random Forest

1. PENDAHULUAN

Dalam era digital yang semakin berkembang, penggunaan internet telah menjadi bagian tak terpisahkan dari kehidupan manusia. Kemudahan akses informasi dan komunikasi yang ditawarkan oleh teknologi ini membawa dampak positif, namun di sisi lain juga meningkatkan ancaman keamanan siber. Salah satu bentuk kejahatan siber yang semakin marak terjadi adalah *phishing*, yaitu upaya penipuan yang bertujuan untuk mencuri informasi sensitif seperti username, password, dan data finansial pengguna [1]. Menurut laporan dari Anti-Phishing Working Group (APWG), jumlah insiden *phishing* mengalami peningkatan yang signifikan dalam beberapa tahun terakhir. Data menunjukkan bahwa serangan *phishing* tidak hanya menargetkan individu, tetapi juga organisasi besar, termasuk institusi keuangan dan perusahaan teknologi [2]. Hal ini menunjukkan urgensi dalam mengembangkan sistem yang mampu mendeteksi dan mencegah serangan *phishing* secara efektif.

Berbagai metode telah dikembangkan untuk mengidentifikasi situs web *phishing*, salah satunya adalah penggunaan machine learning. Machine learning merupakan cabang dari kecerdasan buatan (Artificial Intelligence) yang memungkinkan komputer untuk belajar dari data dan membuat prediksi tanpa perlu diprogram secara eksplisit[3]. Dengan memanfaatkan algoritma klasifikasi, sistem deteksi *phishing* dapat dikembangkan untuk mengenali pola dan karakteristik situs web yang berpotensi berbahaya [4].

Beberapa penelitian telah dilakukan terkait klasifikasi phishing menggunakan *algoritma machine learning*. Penelitian oleh Putri N dan Wijayanto A membandingkan algoritma klasifikasi data mining dalam mendeteksi *website phishing*. Hasilnya menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi sebesar 90,77%, mengungguli *Naïve Bayes* (82,31%), *Decision Tree*, dan SVM. Selain itu, *Random Forest* juga memiliki presisi tertinggi (87,90%) dan sensitivitas sebesar 95,61% [5].

Penelitian lain oleh Irawan A, Heryana N, Hopipah H, dan Rahma D membandingkan empat algoritma, yaitu *Support Vector Machine, Decision Tree, Random Forest*, dan *Multilayer Perceptron*. Dari hasil penelitian ini, *Multilayer Perceptron* menunjukkan performa terbaik dengan akurasi 93,15% dan nilai AUC 0.976 [6]. Sementara itu, penelitian yang dilakukan oleh Mahmud A dan Wirawan S membandingkan tiga algoritma untuk mendeteksi phishing berdasarkan fitur URL. Hasilnya menunjukkan bahwa *Random Forest* memiliki performa terbaik dengan akurasi 0.834, presisi 0.86, *recall* 0.83, dan *F1-score* 0.83, yang mengungguli *Decision Tree* (0.833) dan *K-Nearest Neighbors* (0.482) [7].

Kencana A, Ananda F, Hartanto A, dan Hartatik H dalam penelitiannya menerapkan metode *Random Forest* untuk klasifikasi *phishing* dan *non-phishing* website. Hasilnya, metode ini mencapai akurasi 94,36% pada 2.457 data, yang kemudian diterapkan dalam ekstensi browser. Ekstensi ini menampilkan persentase risiko suatu website serta memberikan peringatan jika website tersebut terdeteksi sebagai *phishing*. Faktor utama yang menyebabkan suatu website terindikasi sebagai *phishing* meliputi ketiadaan SSL, banyaknya *script*, dan panjang *Uniform Resource Locator* (URL) [8].

Terakhir, penelitian oleh Fatiha M, Setiawan I, Ikhsan A, dan Yunita I berfokus pada pengembangan sistem deteksi phishing berbasis web dengan algoritma *Decision Tree*. Sistem ini mencapai akurasi 95,07%, membuktikan efektivitas metode tersebut. Dikembangkan dengan metode *Rapid Application Development* (RAD), sistem ini bersifat adaptif terhadap ancaman *phishing*. Penelitian lanjutan disarankan untuk mengeksplorasi fitur lebih dalam, menerapkan teknik pemrosesan data lanjutan, serta menggunakan dataset yang lebih besar untuk meningkatkan akurasi dan efektivitas sistem deteksi *phishing* [9].

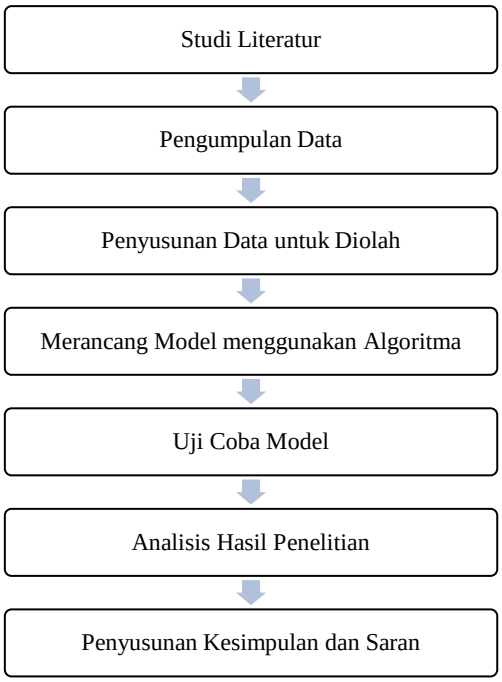
Penelitian ini berfokus pada *Naïve Bayes Classifier* untuk mengklasifikasikan phishing website, berbeda dengan penelitian sebelumnya yang lebih banyak menggunakan *Random Forest, Decision Tree, K-Nearest Neighbors*, dan *Multilayer Perceptron*. Dari segi objek penelitian, penelitian ini menganalisis empat kategori website diantaranya *benign, phishing, defacement*, dan *malware* dengan dataset 651.191 URL, sementara penelitian sebelumnya umumnya hanya membandingkan website *phishing* dan *non-phishing* dengan dataset lebih kecil. Dalam evaluasi performa, penelitian ini mempertimbangkan akurasi, *precision, recall, F1-score, macro average*, dan *weighted average*, sedangkan penelitian lain lebih banyak menitikberatkan pada akurasi dan AUC.

Naïve Bayes Classifier ini dipilih karena kemampuannya dalam menangani data dengan jumlah besar serta keakuratannya dalam mengklasifikasikan kategori data [10]. *Decision Tree* dipilih dalam penelitian ini karena memiliki keunggulan meliputi kemampuan interpretasi, kemampuan menangani data kategori dan numerik, serta kecenderungan untuk tidak memerlukan normalisasi atau standarisasi data [11]. *Random Forest* adalah kombinasi dari *tree* yang ada di dalam *Decision Tree*, semakin banyak *tree* maka akan semakin baik tingkat akurasi hasilnya. Dengan melakukan hal tersebut, *Random Forest* memastikan bahwa data tidak overhit seperti pada *Decision Tree* [12]. Dengan menggunakan dataset yang dikumpulkan dari *Kaggle*, penelitian ini akan mengevaluasi efektivitas *Naïve Bayes Classifier* berdasarkan metrik seperti akurasi, *precision*, *recall*, dan F1-*score* [13].

Tujuan utama dari penelitian ini adalah untuk mengembangkan sistem deteksi *phishing* yang lebih akurat dan efisien. Selain itu, penelitian ini juga bertujuan untuk membandingkan hasil yang diperoleh dengan metode deteksi lainnya guna menentukan pendekatan terbaik dalam mengidentifikasi situs *phishing*.

2. METODOLOGI PENELITIAN

Metode yang digunakan dalam penelitian ini adalah metode eksperimental, yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap yang lain dengan kondisi yang terkendalikan. Metode penelitian eksperimen adalah metode yang dilakukan melalui proses uji coba, yang dirancang untuk menemukan jawaban yang ideal [14]. Dengan beberapa langkah yang telah disusun berdasarkan metode eksperimental seperti yang dijelaskan dalam gambar 1.

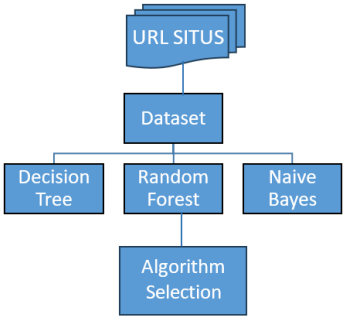


Gambar 1. Metodologi Penelitian Metode Eksperimental

Studi literatur tahap ini dilakukan dengan mengumpulkan dan menganalisis penelitian terdahulu. Data yang digunakan dalam penelitian ini dapat diperoleh dari dataset publik seperti Kaggle. Selanjutnya penyusunan data untuk diolah dilakukan sebelum model *machine learning* diterapkan, data di pre-proses agar dapat digunakan secara optimal. Model deteksi *phishing* dirancang menggunakan algoritma *Naïve Bayes* sebagai model utama dan *Random Forest* serta *Decision Tree* sebagai perbandingan. Setelah model dilatih, dilakukan uji coba menggunakan data uji untuk mengevaluasi performanya. Pengujian dilakukan berdasarkan metrik akurasi, *precision*, *recall*, dan F1-*score*. Hasil dari uji coba dianalisis untuk memahami kelebihan dan kekurangan masing-masing algoritma. Berdasarkan hasil analisis, disusun kesimpulan mengenai algoritma yang paling efektif untuk deteksi *phishing*. Selain itu, diberikan saran untuk pengembangan lebih lanjut.

2.1. Pemilihan Algoritma

Dalam penelitian ini, algoritma *Naïve Bayes Classifier* digunakan untuk mengklasifikasikan situs web sebagai *phishing* atau non-*phishing*. Algoritma ini bekerja berdasarkan teori probabilitas yang menghitung kemungkinan suatu sampel termasuk dalam kategori tertentu berdasarkan fitur yang dimilikinya [10]. Untuk mengetahui seberapa mampu kah Algoritma Klasifikasi mendeteksi dan melakukan peningkatan keamanan terhadap situs *phishing*, nantinya akan ada perbandingan dengan pengujian mengginakan algoritma klasifikasi lainnya yaitu *Decision Tree* dan *Random Forest*.



Gambar 2. Model Klasifikasi Deteksi Situs Phishing

Desain dari model klasifikasi menggunakan beberapa *classifier* yang ditunjukkan pada gambar 2, memilih *classifier* dengan kinerja terbaik (*algoritm selection*). Uji coba ini dilakukan untuk menentukan algoritma mana yang akan dipakai oleh model klasifikasi dan memastikan bahwa model klasifikasi yang dibuat mampu mendeteksi situs *phishing* dengan kinerja baik. Algoritma dengan hasil uji coba terbaik nantinya akan digunakan dalam penelitian ini, karena dengan hasil uji coba yang baik, maka dipastikan model klasifikasi yang dibuat dapat meningkatkan kinerja deteksi situs *phishing*.

Naïve Bayes Classifier merupakan sebuah metoda klasifikasi yang berakar pada pengklasifikasian dengan menggunakan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya sehingga dikenal sebagai Teorema Bayes [15]. Teori *Naïve bayes Classifier* bekerja sangat baik dibanding dengan model *classifier* lainnya yang dibuktikan oleh Xhemali, Hinde dan Stone bahwa *Naïve bayes Classifier* memiliki tingkat akurasi yang lebih baik dibanding model *classifier* lainnya [16]. *Naïve bayes* juga termasuk dalam supervised learning di mana setiap data harus mempunyai label agar dapat dilakukan pelatihan [17]. *Naïve Bayes* didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output [5].

Decision Tree merupakan model prediksi yang bersifat *supervised* yang berarti memerlukan *training dataset* yang perannya menggantikan pengalaman manusia di masa lalu dalam membuat keputusan [18], bertujuan untuk meningkatkan akurasi dan efisiensi deteksi melalui penggunaan *dataset* yang lebih representatif dan teknik pengembangan yang responsif [9]. *Decision Tree* melakukan strategi pencarian secara *top-down* untuk solusinya [19]. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari node akar (*root*) sampai node akhir (*leaf*) dan kemudian akan diprediksi kelas yang dimiliki oleh suatu data baru tertentu [5].

Random Forest adalah teknik dalam *machine learning* yang masuk dalam kategori *ensemble learning*, suatu pendekatan yang menggabungkan beberapa model pembelajaran mesin untuk meningkatkan performa dan keakuratan prediksi [20]. *Random Forest* merupakan algoritma yang pertama kali diperkenalkan pada penelitian yang menyebutkan tentang keunggulannya dalam menyelesaikan masalah yang berkaitan dengan tugas klasifikasi maupun regresi [21]. Dalam *Random Forest*, banyak pohon ditumbuhkan sehingga terbentuk hutan (*forest*), kemudian analisis dilakukan pada kumpulan pohon tersebut [5]. Klasifikasi yang dilakukan di *Random Forest* yaitu dengan penggabungkan pohon (*tree*) lalu dilakukan training pada data yang ada. Jika pohon (tree) semakin bertambah banyak maka tingkat akurasi yang di dapatkan akan semakin tinggi. Hasil klasifikasi random forest di ambil dari setiap pohon (tree) yang di bentuk. Pengutamaan dari tree dibentuk berdasakan vote terbanyak [8].

2.2. Evaluasi Model

Model yang dikembangkan dievaluasi menggunakan beberapa metrik utama [22], yang ditunjukkan pada paersamaan 1-4.

1. Akurasi: Mengukur persentase prediksi yang benar

$$Accuracy = \frac{TP}{TP+FN+FP+TN} \quad (1)$$

2. *Precision*: Mengukur ketepatan model dalam mengklasifikasikan situs *phishing*

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3. *Recall*: Mengukur sejauh mana model dapat menangkap situs *phishing* yang sebenarnya

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

4. F1-*score*: Rata-rata harmonik antara *precision* dan *recall*

$$F1 - Score = \frac{2\ x\ Precision\ x\ Recall}{Precision\ x\ Recall} \quad (4)$$

5. *Macro average* menghitung rata-rata *precision*, *recall*, dan F1-*score* untuk setiap kelas, tanpa memperhatikan jumlah data di setiap kelas, sehingga memberikan penilaian yang merata terhadap semua kelas, sehingga kinerja model pada kelas minoritas tidak diabaikan[23].
6. *Weighted average* menghitung rata-rata *precision*, *recall*, dan F1-*score* dengan memperhitungkan proporsi jumlah data di setiap kelas, sehingga memberikan gambaran performa model yang lebih

realistis pada dataset yang tidak seimbang, karena kelas dengan lebih banyak data akan memiliki bobot lebih besar [23].

3. HASIL DAN PEMBAHASAN

Pada bagian ini, dijelaskan hasil yang diperoleh dari pengujian model klasifikasi terhadap dataset yang telah diproses. Hasil yang akan dibahas meliputi distribusi dataset, kinerja algoritma klasifikasi, dan perbandingan algoritma.

3.1 Distribusi Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari 651.191 URL yang telah dikategorikan berdasarkan jenis ancaman siber yang diidentifikasi. Proses pengambilan data dilakukan dengan cara mengunduh dataset dari sumber-sumber tersebut secara berkala dan menyimpannya dalam format CSV untuk kemudian diproses lebih lanjut. Setelah data dikumpulkan, dilakukan pre-*processing* data agar dataset dapat digunakan untuk pelatihan model.

Kategori-kategori tersebut mencakup *benign*, *phishing*, *defacement*, dan *malware*, yang masing-masing memiliki jumlah data sebagai berikut.

1. *Benign* dengan jumlah 428.103 URL (65,74%) adalah URL yang aman dan tidak mengandung ancaman siber.
2. *Phishing* dengan jumlah 94.111 URL (14,46%) adalah URL yang dirancang untuk menipu pengguna agar memberikan informasi pribadi.
3. *Defacement* dengan jumlah 96.457 URL (14,81%) adalah URL yang telah diretas dan diubah tampilannya oleh pihak tidak bertanggung jawab. Biasanya terdapat pesan dari hacker, seperti slogan atau klaim serangan.
4. *Malware* dengan jumlah 32.520 URL (4,99%) adalah URL yang mengandung kode berbahaya yang dapat menginfeksi perangkat pengguna. Bisa berupa situs yang menyebarkan virus, trojan, atau ransomware.

3.2 Kinerja Algoritma Klasifikasi

Dalam analisis performa model klasifikasi, dilakukan pengujian menggunakan beberapa algoritma yang berbeda untuk menentukan tingkat akurasi serta efektivitas model dalam melakukan prediksi terhadap dataset yang digunakan. Pengujian ini bertujuan untuk mengidentifikasi algoritma mana yang memberikan hasil terbaik berdasarkan metrik evaluasi tertentu. Adapun metrik yang digunakan meliputi *precision*, *recall*, F1-*score*, *Macro average*, dan *Weighted average*. Masing-masing metrik memiliki fungsi tertentu dan memberikan wawasan yang berbeda tentang hasil klasifikasi.

Tahap pre-*processing* data dalam penelitian ini dilakukan untuk memastikan dataset siap digunakan dalam pelatihan model klasifikasi. Proses ini dimulai dengan pembersihan data, yaitu menghapus URL duplikat, menghilangkan karakter tidak valid, dan memeriksa validitas *Uniform Resource Locator* (URL) menggunakan pustaka di Python. Selanjutnya, dilakukan ekstraksi fitur URL, seperti panjang URL, jumlah tanda hubung (-), jumlah titik (.), keberadaan https, jenis domain, serta panjang subdomain. Setelah itu, dilakukan *labeling* data berdasarkan kategori *phishing*, *benign*, *defacement*, dan *malware*. Tahap berikutnya adalah normalisasi data, di mana semua teks URL diubah menjadi huruf kecil dan data kategorikal dikonversi ke format numerik agar dapat digunakan dalam model *machine learning*. Terakhir, dataset dibagi menjadi training set (80%) untuk melatih model dan testing set (20%) untuk menguji performa model.

3.2.1 Hasil Akurasi Pengujian Algoritma Naïve Bayes

Algoritma *Naïve Bayes* digunakan untuk memprediksi data dengan pendekatan probabilistik. Hasil akurasi menunjukkan sejauh mana algoritma ini efektif dalam klasifikasi dataset yang digunakan. Hasil akurasi Naïve Bayes dapat ditunjukkan pada tabel 1.

Tabel 1. Hasil Akurasi Pengujian Algoritma *Naïve Bayes*

Label	Precision	Recall	F1-score	Support
Benign	0.60	0.34	0.42	85,778
Defacement	0.59	0.35	0.42	19,104
Malware	0.62	0.32	0.41	6,596
Phishing	0.48	0.24	0.32	18,764
Macro Average	0.68	0.60	0.62	130,242
Weighted Average	0.82	0.85	0.83	130,242

Algoritma *Naïve Bayes* menghasilkan akurasi sebesar 85% dalam mengklasifikasikan dataset. Hal ini menunjukkan bahwa algoritma ini mampu memprediksi dengan benar sebagian besar data dalam pengujian. Berikut merupakan detail metrik evaluasi atas pengujian tersebut.

1. Hasil *precision* dari akurasi pengujian *Naïve Bayes* pada kelas *benign* adalah 60% prediksi benar. Kelas *defacement* memiliki *precision* 59%, menunjukkan algoritma cukup akurat dalam mengenali kelas ini. *Precision* tertinggi, 62%, terlihat pada kelas *malware*. Namun, pada kelas *phishing*, *precision* menurun menjadi 74%, menunjukkan kesulitan algoritma dalam membedakan *phishing* dari kelas lainnya.
2. *Recall* tertinggi terlihat pada kelas *defacement* (35%), menunjukkan kemampuan algoritma menangkap hampir semua data *defacement*. Kelas *benign* juga memiliki *recall* yang baik (34%). *Recall* menurun pada kelas *malware* (32%) dan *phishing* (24%), menandakan beberapa data dari kedua kelas ini tidak terdeteksi dengan baik.
3. F1-*score* menunjukkan keseimbangan antara *precision* dan *recall*. Nilainya berkisar antara 32% (*phishing*) hingga 42% (*benign* dan *defacement*). Kelas *malware* memiliki F1-*score* 41%, menunjukkan performa yang cukup baik meskipun *recall* lebih rendah.
4. *Macro average* dari hasil akurasi pengujian *Naïve Bayes*, mendapatkan nilai presentase atas *Precision* sebesar 68%, *recall* sebesar 60%, dan F1-*score* sebesar 62%.
5. *Weighted average* dari hasil akurasi pengujian *Naïve Bayes*, ditemukan bahwa seluruh metrik memiliki nilai sekitar 82% hingga 85%, mencerminkan kinerja keseluruhan model pada dataset yang tidak seimbang.

Sehingga secara keseluruhan, *Naïve Bayes* memiliki performa yang sangat baik untuk kelas *benign* dan *defacement*. Namun, performanya sedikit menurun pada kelas *phishing*, yang mungkin memerlukan teknik tambahan untuk meningkatkan prediksi pada kelas tersebut, seperti melakukan *oversampling* atau menambah fitur relevan. Analisis detail dari pengujian tersebut dapat melihat bagaimana model *Naïve Bayes* mengklasifikasikan data dengan benar maupun salah.

3.2.2 Hasil Akurasi Pengujian Algoritma Decision Tree

Setelah dilakukan pelatihan dan pengujian menggunakan algoritma *Decision Tree*, diperoleh hasil evaluasi model berdasarkan classification report ditunjukkan pada tabel 2.

Tabel 2. Hasil Akurasi Pengujian Algoritma Decision Tree

Label	Precision	Recall	F1-score	Support
Benign	0.93	0.96	0.94	85,778
Defacement	0.95	0.97	0.97	19,104
Malware	0.94	0.92	0.92	6,596
Phishing	0.88	0.87	0.87	18,764
Macro Average	0.93	0.92	0.93	130,242
Weighted Average	0.96	0.96	0.96	130,242

Pada pengujian ini, model *Decision Tree* memperoleh akurasi sebesar 96%, yang menunjukkan bahwa model memiliki tingkat prediksi yang cukup baik dalam mengklasifikasikan data. Berikut merupakan detail metrik evaluasi atas pengujian tersebut.

1. *Precision* tertinggi diperoleh pada kelas *Defacement* 95%, yang berarti model jarang salah dalam mengklasifikasikan data sebagai *Defacement*. *Precision* terendah diperoleh pada kelas *Phishing* 88%, yang menunjukkan masih adanya data yang salah diklasifikasikan sebagai *phishing*.
2. *Recall* tertinggi juga terdapat pada kelas *Defacement* 97%, yang berarti hampir semua data yang termasuk kategori ini dapat diklasifikasikan dengan benar. *Recall* terendah terdapat pada kelas *Phishing* 87%, yang menunjukkan masih adanya sampel *phishing* yang salah diklasifikasikan ke kategori lain.
3. Nilai F1-*score* tertinggi diperoleh pada kelas *Defacement* 97%, menunjukkan bahwa model sangat baik dalam mendeteksi kategori ini. F1-*score* terendah terdapat pada kelas *Phishing* 87%, yang mengindikasikan bahwa model masih memiliki sedikit kelemahan dalam mengidentifikasi *phishing* secara akurat.
4. *Macro average* dari hasil akurasi pengujian *Decision Tree*, mendapatkan nilai presentase atas *Precision* sebesar 96%, *recall* sebesar 93%, dan F1-*score* sebesar 95%.
5. *Weighted average* dari hasil akurasi pengujian *Decision Tree*, ditemukan bahwa seluruh metrik memiliki nilai sekitar 96%, mencerminkan kinerja keseluruhan model pada dataset yang tidak seimbang.

Karena nilai *Macro average* lebih rendah dari *Weighted average*, maka kemungkinan ada ketidakseimbangan data, di mana kelas *Phishing* memiliki jumlah data lebih sedikit dibanding kelas lainnya dan menjelaskan bahwa performanya kurang optimal dibanding kelas lainnya.

3.2.3 Hasil Akurasi Pengujian Algoritma Random Forest

Pengujian dilakukan menggunakan algoritma *Random Forest* untuk mengklasifikasikan data. Akurasi sebesar 97% menunjukkan bahwa model dapat mengklasifikasikan data dengan tingkat kesalahan yang rendah. Hasil akurasi *Random Forest* dapat ditunjukkan pada tabel 3.

Tabel 3. Hasil Akurasi Pengujian Algoritma Random Forest

Label	Precision	Recall	F1-score	Support
Benign	0.97	0.99	0.98	85,778
Defacement	0.98	0.98	0.98	19,104
Malware	0.99	0.97	0.98	6,596
Phishing	0.94	0.86	0.90	18,764
Macro Average	0.96	0.93	0.94	130,242
Weighted Average	0.97	0.97	0.97	130,242

Pada pengujian *Random Forest*, terindikasi bahwa model memiliki performa yang tinggi dalam mengenali pola dalam data. Selain akurasi, evaluasi model dilakukan menggunakan metrik lain seperti *precision*, *recall*, F1-*score*, *Macro average*, dan *Weighted average*. Hasil pengujian metrik ini adalah sebagai berikut.

1. *Precision* tertinggi diperoleh pada kelas *Defacement* (98%), yang berarti model jarang salah dalam mengklasifikasikan data sebagai *Defacement*. *Precision* terendah diperoleh pada kelas *Phishing* (91%), yang menunjukkan masih adanya data yang salah diklasifikasikan sebagai *phishing*.
2. *Recall* tertinggi diperoleh pada kelas *Benign* (99%), yang berarti hampir semua data *benign* berhasil diklasifikasikan dengan benar. *Recall* terendah ada pada kelas *Phishing* (86%), yang menunjukkan model masih memiliki kesulitan dalam mengenali semua data *phishing*.
3. F1-*score* tertinggi didapatkan pada kelas *Defacement* (98%), menunjukkan keseimbangan yang sangat baik antara *precision* dan *recall*. F1-*score* terendah ada pada kelas *Phishing* (90%), menunjukkan bahwa model masih kurang optimal dalam mengenali *phishing* secara seimbang.
4. *Macro average* memiliki nilai *precision* (96%), *recall* (93%), dan F1-*score* (94%), yang mengindikasikan performa rata-rata model di semua kelas cukup baik.
5. *Weighted average* memiliki nilai *precision* (97%), *recall* (97%), dan F1-*score* (97%), yang menunjukkan keseimbangan dalam menangani distribusi data berdasarkan jumlah *support* masing-masing kelas.

Kesalahan terbesar terlihat pada kelas *phishing*, yang sering salah diklasifikasikan sebagai *benign*, *defacement*, atau *malware*. Hal ini menunjukkan bahwa model perlu dioptimalkan lebih lanjut untuk meningkatkan akurasi dalam mendeteksi *phishing*.

3.3 Pembahasan

Pada bagian ini, hasil penelitian dianalisis dan dikaitkan dengan literatur yang relevan. Pada penelitian ini, tiga algoritma yang sering digunakan dalam klasifikasi data telah diuji dan dibandingkan, yaitu *Naïve Bayes*, *Decision Tree*, dan *Random Forest*. Masing-masing algoritma memiliki pendekatan yang berbeda dalam menangani data serta dalam proses pengambilan keputusan terhadap prediksi kelas suatu objek. Berdasarkan hasil pengujian akurasi dan metrik evaluasi lainnya, akan dilakukan perbandingan performa tiga algoritma klasifikasi tersebut.

3.3.1 Perbandingan Algoritma

Akurasi adalah metrik utama yang digunakan untuk mengukur performa model klasifikasi. Hasil akurasi dari ketiga algoritma ditunjukkan pada tabel 4.

Tabel 4. Perbandingan Algoritma

Algoritma	Akurasi
Naïve Bayes	85%
Decision Tree	96%
Random Forest	97%

Hasil pengujian menunjukkan bahwa *Random Forest* memiliki akurasi tertinggi sebesar 96%, diikuti oleh *Decision Tree* dengan 96%, sementara *Naïve Bayes* mencapai 85%. Meskipun *Naïve Bayes* unggul dalam kecepatan pemrosesan, algoritma ini mengalami kesulitan dalam mengklasifikasikan kelas tertentu, terutama *Phishing*. Sementara itu, *Decision Tree* menunjukkan kinerja yang sangat baik, tetapi rentan terhadap *overfitting*, yang kemudian diminimalkan oleh *Random Forest* melalui kombinasi banyak pohon keputusan.

Dengan membandingkan metrik seperti *Precision*, *Recall*, dan F1-*Score*, analisis ini bertujuan untuk mengidentifikasi kelebihan serta keterbatasan masing-masing algoritma dalam menangani dataset yang diuji.

Tabel 5. Perbandingan Metrik

Metrik	Naïve Bayes	Decision Tree	Random Forest
Precision (Macro Average)	68%	93%	96%
Recall (Macro Average)	60%	92%	93%
F1-Score (Macro Average)	62%	93%	94%
Precision (Weighted Average)	82%	96%	97%
Recall (Weighted Average)	85%	96%	97%
F1-score (Weighted Average)	83%	96%	97%

Berdasarkan hasil perbandingan metrik *Macro Average* dan *Weighted Average* pada tabel 5, model *Random Forest* menunjukkan performa terbaik dibandingkan dengan *Decision Tree* dan *Naïve Bayes* dalam mendeteksi website phishing. *Precision Macro Average* pada *Naïve Bayes* hanya 68%, menunjukkan bahwa model ini kurang akurat dalam membedakan antara website phishing dan non-phishing. Sementara itu, *Decision Tree* dan *Random Forest* memiliki *Precision Macro Average* yang lebih tinggi, yaitu 93% dan 96%, yang berarti model berbasis pohon keputusan lebih unggul dalam menjaga tingkat keakuratan prediksi di setiap kelas. *Recall Macro Average* juga menunjukkan bahwa *Naïve Bayes* hanya memiliki *recall* sebesar 60%, yang cukup rendah karena banyak website *phishing* yang gagal terdeteksi. Sebaliknya, *Decision Tree* dan *Random Forest* mencapai 92% dan 93%, yang menandakan bahwa model ini lebih efektif dalam mengenali website *phishing* tanpa terlalu banyak kesalahan negatif.

Selain itu, *F1-Score Macro Average* dan *Weighted Avera*ge memperkuat keunggulan *Random Forest* dan *Decision Tree* dibandingkan *Naïve Bayes*. F1-Score *Macro Average* pada *Naïve Bayes* hanya 68%, sementara *Decision Tree* mencapai 92% dan *Random Forest* 93%, yang berarti model berbasis pohon keputusan memiliki keseimbangan yang lebih baik antara *precision* dan *recall. Weighted Average* juga menunjukkan tren yang serupa, di mana *Naïve Bayes* hanya mencapai 85%, sedangkan *Decision Tree* dan *Random Forest* berada di kisaran 96%-97%. Dengan demikian, dapat disimpulkan bahwa *Random Forest* merupakan model terbaik untuk deteksi website phishing, karena memiliki presisi tinggi, mampu mendeteksi lebih banyak kasus *phishin*g dengan benar, serta menjaga keseimbangan antara *precisio*n dan *recal*l yang optimal.

Tabel 6. Kekurangan dan keunggulan algoritma klasifikasi berdasarkan hasil pengujian akurasi model

Kriteria	Naïve Bayes (NB)	Decision Tree (DT)	Random Forest (RF)
Akurasi Keseluruhan	85% (Cukup baik, tetapi lemah pada kelas minoritas)	96% (cukup baik, tetapi rentan overfitting)	97% (Tertinggi, perlu di optimalkan)
Ketahanan terhadap Overfitting	Tidak mengalami overfitting karena pendekatan probabilistik	Cenderung mengalami overfitting, terutama tanpa pruning	Mengurangi overfitting dengan kombinasi banyak pohon keputusan
Kecepatan Training	Sangat cepat karena perhitungan probabilitas sederhana	Waktu pelatihan sedang, tergantung kompleksitas pohon	Lebih lambat karena harus membangun banyak pohon keputusan
Ketepatan Klasifikasi	Kurang akurat pada kelas yang tidak seimbang, terutama Phishing	Sangat baik untuk pola data yang jelas	Lebih akurat dan stabil dibandingkan DT, karena menggabungkan beberapa model
Ketepatan Menangani Dataset Besar	Efisien dalam komputasi, cocok untuk dataset besar	Bisa menangani dataset besar, tetapi performa bisa menurun tanpa optimasi	Bisa menangani dataset besar lebih baik dari DT, tetapi lebih lambat dibandingkan NB
Kemampuan Menangani Fitur yang Berhubungan	Kurang optimal karena mengasumsikan fitur saling independen	Sangat baik dalam menangani fitur yang memiliki hubungan	Sangat baik, bahkan lebih stabil dibandingkan DT karena agregasi model

Berdasarkan hasil perbandingan pada tabel 6, *Naïve Bayes* (NB) unggul dalam kecepatan pelatihan dan efisiensi komputasi, menjadikannya pilihan yang baik untuk dataset besar dengan fitur independen. Namun, NB kurang akurat dalam menangani data yang tidak seimbang dan fitur yang memiliki hubungan. *Decision Tree* (DT) memiliki akurasi yang tinggi dan dapat menangani pola data yang kompleks, tetapi cenderung mengalami overfitting tanpa teknik pruning. Meskipun performanya cukup baik pada dataset besar, kecepatan trainingnya lebih lambat dibandingkan NB. *Random Forest* (RF) menunjukkan performa terbaik secara keseluruhan, dengan akurasi tinggi dan ketahanan terhadap overfitting karena kombinasi banyak pohon

keputusan. RF juga unggul dalam menangani dataset besar dan fitur yang saling berhubungan, meskipun waktu pelatihannya lebih lama dibandingkan NB dan DT. Dengan demikian, pemilihan algoritma bergantung pada kebutuhan spesifik, seperti kecepatan, akurasi, atau ketahanan terhadap *overfitting*.

4. CONCLUSION

Penelitian ini mengembangkan sistem deteksi URL *phishing* menggunakan model *machine learning*, yaitu *Naïve Bayes, Random Forest*, dan *Decision Tree*. Hasil penelitian menunjukkan bahwa *Random Forest* memberikan performa terbaik dengan akurasi 97%, diikuti oleh *Decision Tree* 96%, dan *Naïve Bayes* 85%. *Random Forest* memiliki keseimbangan *precision* dan *recall* yang baik, sehingga lebih andal dalam mendeteksi URL phishing dengan tingkat *false positive* dan *false negative* yang lebih rendah. Sebaliknya, *Naïve Bayes* menunjukkan kelemahan signifikan dengan *recall* yang rendah sebesar 24%, membuatnya kurang efektif dalam mengenali URL berbahaya. *Decision Tree* memiliki performa yang cukup baik, tetapi masih sedikit lebih rendah dibandingkan *Random Forest* dalam deteksi phishing secara konsisten. Selain itu, penggunaan ekstraksi fitur berbasis TF-IDF dan fitur numerik terbukti meningkatkan performa model, terutama pada *Random Forest* dan *Decision Tree*.

Untuk penelitian selanjutnya, disarankan menggunakan dataset yang lebih besar dan beragam, mengeksplorasi model *deep learning* seperti LSTM atau CNN, serta mengembangkan sistem deteksi *phishing* real-time dalam bentuk plugin *browser* atau API. Dengan perbaikan ini, diharapkan sistem deteksi URL phishing dapat semakin efektif dalam melindungi pengguna dari ancaman siber yang terus berkembang.

REFERENSI

[1] M. H. dan F. N. Wibowo, "Ancaman Phishing Terhadap Pengguna Sosial Media Dalam Dunia Cyber Crime," *JOEICT(Jurnal of Education and Information Communication Technology)*, vol. 1, no. 1, hlm. 1–5, 2017.

[2] A. S. Y. dkk Irawan, "Identifikasi Website Phishing dengan Perbandingan Algoritma Klasifikasi," *Syntax: Jurnal Informatika*, vol. 10, no. 1, hlm. 57–67, 2021.

[3] S. , & N. M. J. Das, "A survey on types of machine learning techniques in intrusion prevention systems," *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, hlm. 2296–2299, 2017.

[4] S. Hendrian, "Algoritma Klasifikasi Data Mining Untuk Memprediksi," *Faktor Exacta*, vol. 11, no. 3, hlm. 266–274, 2018.

[5] N. B. Putri dan A. W. Wijayanto, "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing," *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, hlm. 59–66, Jan 2022, doi: 10.34010/komputika.v11i1.4350.

[6] A. S. Y. Irawan, N. Heryana, H. S. Hopipah, dan D. Rahma, "Identifikasi Website Phishing dengan Perbandingan Algoritma Klasifikasi," *Syntax : Jurnal Informatika*, vol. 10, no. 01, hlm. 57–67, Jun 2021, doi: 10.35706/syji.v10i01.5292.

[7] A. F. Mahmud dan S. Wirawan, "Phishing Website Detection Using Machine Learning Classification Method," *SISTEMASI*, vol. 13, no. 4, hlm. 1368, Jul 2024, doi: 10.32520/stmsi.v13i4.3456.

[8] A. K. Kencana, F. D. Ananda, A. D. Hartanto, dan H. Hartatik, "Implementasi Metode Random Forest Klasifikasi untuk Phishing Link Detection," *Intechno Journal (Information Technology Journal)*, vol. 4, no. 2, hlm. 55–59, Des 2022, doi: 10.24076/intechnojournal.2022v4i2.1562.

[9] M. R. Fatiha, I. Setiawan, A. N. Ikhsan, dan I. R. Yunita, "Optimisasi Sistem Deteksi Phishing Berbasis Web Menggunakan Algoritma Decision Tree," *Jurnal Ilmiah IT CIDA*, vol. 10, no. 2, hlm. 97, Des 2024, doi: 10.55635/jic.v10i2.212.

[10] A. dan P. E. Fatkhurohman, "Penerapan Algoritma Naïve Bayes Classifier Untuk Meningkatkan Keamanan Data Dari Website Phising," *RESPATI : Jurnal Ilmiah Teknologi Informasi*, vol. XIV, no. 1, hlm. 115–124, 2019.

[11] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, dan U. Firdaus, "Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank," *Karimah Tauhid*, vol. 3, no. 2, hlm. 1860–1874, Feb 2024, doi: 10.30997/karimahtauhid.v3i2.11952.

[12] V. A. Windarni, A. F. Nugraha, S. T. A. Ramadhani, D. A. Istiqomah, F. M. Puri, dan A. Setiawan, "Deteksi Website Phishing Menggunakan Teknik Filter Pada Model Machine Learning," *Information System Journal*, vol. 6, no. 01, Agu 2023, doi: 10.24076/infosjournal.2023v6i01.1268.

[13] I. A. I. A. dan A. A. S. A. A. S. Willy Sutina, "Pengaruh Algoritma Sequential Minimal Optimization Pada Support Vector Machine Untuk Klasifikasi Data (Influence Of Sequential Minimal Optimization Algorithm On Support Vector Machine For Data Classification)," *Telkom University*, 2010.

[14] K. Bebhe, R. R. Widjaja, dan L. M. F. Purwanto, "Model Penelitian Eksperimen Pada Penelitian Tentang Bahan Dinding Bata Intercocking Tanah Putih Dan Sampah Plastik," *AKSELERASI: Jurnal Ilmiah Nasional*, vol. 5, no. 2, hlm. 151–156, Okt 2023, doi: 10.54783/jin.v5i2.738.

[15] J. Sihombing, "Klasifikasi Data Antroprometri Individu Menggunakan Algoritma Naïve Bayes Classifier," *BIOS : Jurnal Teknologi Informasi dan Rekayasa Komputer*, vol. 2, no. 1, hlm. 1–10, Mar 2021, doi: 10.37148/bios.v2i1.15.
[16] Y. S. Sari, "Penerapan Metode Naïve Bayes Untuk Mengetahui Kualitas Air Di Jakarta," *Jurnal Ilmiah FIFO*, vol. 13, no. 2, hlm. 222, Nov 2021, doi: 10.22441/fifo.2021.v13i2.010.
[17] J. Alvares dan U. Anggoro Saputro, "Klasifikasi Short Message Service Spam Menggunakan Algoritma Naïve Bayes Classifier," *Smart Comp: Jurnalnya Orang Pintar Komputer*, vol. 12, no. 4, Okt 2023, doi: 10.30591/smartcomp.v12i4.4503.
[18] R. N. Ramadhon, A. Ogi, A. P. Agung, R. Putra, S. S. Febrihartina, dan U. Firdaus, "Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank," *Karimah Tauhid*, vol. 3, no. 2, hlm. 1860–1874, Feb 2024, doi: 10.30997/karimahtauhid.v3i2.11952.
[19] D. Yusuf dan E. Sestri, "Metode Decision Tree Dalam Klasifikasi Kredit Pada Nasabah PT Bank Perkreditan Rakyat (Studi Kasus : PT BPR Lubuk Raya Mandiri)," *Jurnal Sistem Informasi (JUSIN)*, vol. 1, no. 1, hlm. 21–28, Okt 2020, doi: 10.32546/jusin.v1i1.855.
[20] A. D. Harahap, D. Juardi, dan A. S. Y. Irawan, "Rancang Bangun Sistem Pendeteksi Link Phishing Menggunakan Algoritma Random Forest Berbasis Web," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 3, Agu 2024, doi: 10.23960/jitet.v12i3.4858.
[21] A. Ferdita Nugraha, R. F. A. Aziza, dan Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *Jurnal Infomedia*, vol. 7, no. 1, hlm. 39, Jun 2022, doi: 10.30811/jim.v7i1.2959.
[22] N. Hernandoko, P. W. Laksono, dan C. N. Rosyidi, "Penerapan Sistem Kontrol Kualitas dengan Mengunakan Model CNN Transfer Learning VGG 19 pada Inspeksi Kain di Industri Tekstil," *Performa: Media Ilmiah Teknik Industri*, vol. 23, no. 2, hlm. 166, Sep 2024, doi: 10.20961/performa.23.2.86589.
[23] I. K. S. S. A. W. Fitra A. Bachtiar, "Perbandingan Algoritme Machine Learning untuk Memprediksi Pengambil Matakuliah," *Jurnal Teknologi Informasi dan Ilmu Komputer*, hlm. 543–548, 2019.