Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing

Anggit Ferdita Nugraha[1], Rifda Faticha Alfa Aziza[2*], Yoga Pristyanto[3]

[1]*Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta*
Jln. Padjajaran, Ring Road Utara, Condong Catur, Depok, Sleman, Yogyakarta
[1]*anggitferdita@amikom*.ac.id

[2]*Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta*
Jln. Padjajaran, Ring Road Utara, Condong Catur, Depok, Sleman, Yogyakarta
[2]*rifda@amikom.ac.id*

[3]*Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta*
Jln. Padjajaran, Ring Road Utara, Condong Catur, Depok, Sleman, Yogyakarta
[3]yoga.pristyanto@amikom.ac.id

***Abstrak*—** Ketidakseimbangan kelas seringkali menjadi hal yang diabaikan terutama oleh para peneliti di bidang data mining dan machine learning. padahal dengan melakukan penanganan terhadap ketidakseimbangan kelas, memungkinkan adanya peningkatan kinerja klasifikasi apabila dibandingkan dengan penggunaan model klasifikasi tunggal. Hal tersebut dikarenakan cara kerja klasifikasi tunggal yang cenderung bekerja untuk mengenali pola mayoritas dan mengasumsikan distribusi data secara relative membuat kinerja klasifikasi menjadi kurang maksimal. Oleh karena itu, pada penelitian ini diusulkan sebuah pendekatan level algoritmik yang memanfaatkan algoritma random forest, serta metode stacking yang menggabungkan algoritma decision tree dengan naïve bayes sebagai model klasifikasi pada dua jenis web phising dataset yang masing-masing memiliki *imbalanced rasio* sebesar 1.25% untuk binary class dan 6.82% untuk multiclass. Ide dasar dilakukannya pendekatan level algoritmik ini karena keunggulannya yang mampu meningkatkan dan memaksimalkan kinerja klasifikasi tanpa merubah komposisi maupun pola dataset sehingga informasi yang ada pada dataset tetap terjaga. Hasilnya, penggunaan algoritma random forest memiliki nilai akurasi tertinggi yakni sebesar 96.4% pada dataset web phising binary class, sedangkan metode stacking yang menggabungkan algoritma decision tree dengan naïve bayes menghasilkan kinerja terbaik pada dataset web phising multiclass berdasarkan nilai akurasi sebesar 88.8%.
***Kata kunci*— ketidakseimbangan kelas, klasifikasi, machine learning, random forest, metode stacking, ensemble model**

*Abstract***—** Class imbalances are often overlooked, especially by researchers in the data mining and machine learning fields. whereas by dealing with class imbalances it is possible to improve classification performance when compared to the use of a single classification model. This is because the way a single classification works, which tends to work to recognize the majority pattern and assumes a relative data distribution, makes the classification performance less than optimal. Therefore, this study proposes an algorithm-level approach that utilizes the random forest algorithm, as well as a stacking method that combines a decision tree algorithm with nave Bayes as a classification model for two types of phishing web datasets, each of which has an unbalanced ratio. from 1.25% for binary. class and 6.82% for multiclass. The basic idea of doing this algorithmic level approach is because of its advantages, namely being able to improve and maximize classification performance without changing the composition or pattern of the dataset so that the information contained in the dataset is maintained. As a result, the use of the random forest algorithm has the highest accuracy value of 96.4% on the web phishing binary class dataset, while the stacking method that combines the decision tree algorithm with nave Bayes results in the best performance on the multiclass phishing web dataset. based on an accuracy value of 88.8%.
***Keywords*— imbalanced class, classification, machine learning, random forest, stacking, ensemble model**.

I. PENDAHULUAN

Era *Pandemic* Covid-19 yang melanda masyarakat di seluruh dunia memunculkan adanya perubahan pada kebiasaan, gaya hidup, dan tingkah laku manusia [1]. Kebiasaan yang sebelumnya biasa dilakukan secara fisik dan luring (*offline*), perlahan mulai ditinggalkan dan berganti dengan kebiasaan baru yang dilakukan secara daring (*online*) melalui berbagai *platform* digital . Adanya internet juga turut berperan sebagai media pendukung yang tidak hanya memudahkan melainkan juga telah berevolusi menjadi media yang menyediakan berbagai fasilitas dan dukungan dalam beraktivitas secara *online* [2]. Salah satu perubahan yang nampak dari adanya evolusi internet adalah halaman website. Pada awal kemunculannya, website hanya digunakan sebagai media untuk penyampaian informasi satu arah, akan tetapi, seiring dengan adanya perkembangan teknologi informasi dan komunikasi yang terjadi sekarang ini, membuat website kini mampu untuk menjadi sebuah media interaksi, media komunikasi, dan bahkan bisa juga digunakan sebagai media dan sarana untuk melakukan berbagai proses transaksi [3]. Sayangnya, adanya perubahan yang terjadi pada website, justru diimbangi dengan semakin merebaknya potensi ancaman yang membahayakan dan dapat menimbulkan kerugian bagi penggunanya. Satu dari sekian banyak ancaman siber yang mengancam pengguna website adalah serangan Web Phising [3]–[5].

Web Phising merupakan salah satu ancaman kejahatan siber yang tujuannya adalah untuk mengambil informasi penting dari targetnya seperti *username*, *password*, data kartu kredit, maupun data-data serta informasi pribadi lainnya [6]–[8]. Web phising bekerja dengan cara menjebak korban untuk klik sebuah tautan yang nantinya akan diarahkan pada halaman palsu yang didesain semirip mungkin dengan website asli yang diharapkan oleh targetnya. Umumnya, pengguna yang terkena serangan web-phising tidak akan menyadari bahwa dirinya sedang berada pada jebakan web phising, dan baru akan tersadar setelah mengalami berbagai kerugian material [7], [9]. Anti-Phising Working Group (APWG) mencatat adanya 245.771 halaman website yang terindikasi sebagai web-phising yang menyerang sektor lembaga keuangan, e-commerce, dan

social media pada quartil awal tahun 2021, dan diperkirakan nilainya akan terus meningkat hingga mencapai dua kali lipat pada kuartil ke-tiga di tahun 2021 [10]. Oleh karena itu, dengan semakin merebaknya serangan web phising yang terjadi, perlu adanya sebuah sistem yang dapat membantu pengguna website dalam mendeteksi adanya serangan web phising terlebih untuk meminimalisir terjadinya berbagai kerugian yang diakibatkan dari adanya serangan web phising tersebut.

Mengingat pentingnya menghadapi ancaman serangan web phising, pengembangan sistem yang dapat mendeteksi adanya web phising masih terus dilakukan oleh peneliti di berbagai sektor keilmuan, tak terkecuali bagi para peneliti di bidang data mining dan *machine learning.* Algoritma klasifikasi popular seperti K-Nearest Neighbor (K-NN) [3], [5]–[8], Decision Tree [3], [4], [7], [11], Naive Bayes [3]–[6], [12]–[14], dan Support Vector Machine (SVM) [3], [6]–[9], [14], [15] menjadi algoritma populer yang sampai dengan saat ini masih banyak digunakan dalam mendeteksi adanya serangan web phising dengan kinerja yang cukup baik. Penelitian [11] menunjukkan bahwa penggunaan algoritma Decision Tree, mampu memberikan kinerja yang baik berdasarkan nilai akurasi sebesar 94.3%. Sebanding dengan penggunaan Decision Tree, pada penelitian [3] yang menggunakan Naive Bayes sebagai algoritma klasifikasi, juga menunjukkan kinerja yang baik apabila menilik nilai akurasi yang dihasilkan yakni sebesar 94.2%, penelitian lainnya yang menggunakan metode K-Nearest Neighbor [6] juga memunculkan potensi kinerja yang baik karena nilai akurasi yang dihasilkan sebesar 94.5%.

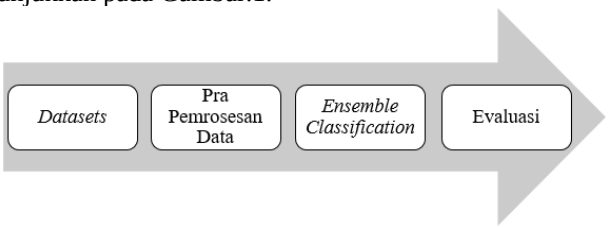
Jika didasarkan pada penelitian sebelumnya, pengembangan model *machine learning* untuk deteksi web phising masih menyisakan ruang untuk memaksimalkan potensi kinerja klasifikasinya. Hal tersebut dikarenakan sebagian besar penelitian yang dilakukan oleh para peneliti, hanya menerapkan model klasifikasi tunggal dan mengabaikan adanya ketidakseimbangan pada kelas data. Padahal, penggunaan model klasifikasi tunggal cenderung bekerja dengan mengenali pola pada kelas mayoritas yang membuat kinerja klasifikasi menjadi kurang maksimal. Terdapat dua pendekatan dalam menangani adanya ketidakseimbangan pada kelas data, yang pertama adalah melakukan penanganan pada level data menggunakan algoritma resampling seperti oversampling maupun undersampling [16]–[18], sedangkan pendekatan kedua dilakukan menggunakan ensemble learning seperti Bagging maupun Stacking [19]–[22].

Secara teori, penggunaan metode resampling memiliki resiko terjadinya duplikasi instance yang dapat menyebabkan hilangnya informasi serta dapat merubah komposisi yang terdapat pada dataset sehingga mempengaruhi kinerja klasifikasi. Sedangkan, penggunaan *ensemble learning*, diketahui mampu meminimalisir adanya resiko perubahan komposisi pada dataset.

Oleh karena itu, pada penelitian ini akan dilakukan percobaan menggunakan metode *random forest* dan metode *stacking* yang merupakan *ensemble learning* untuk memaksimalkan kinerja klasifikasi pada proses deteksi web phising. Terdapat dua kontribusi yang diharapkan ada pada penelitian ini. Pertama, metode *ensemble* yang diusulkan mampu menjadi solusi dalam menangani ketidakseimbangan kelas pada dataset dalam mengklasifikasikan web phishing, sedangkan kontribusi kedua, metode *ensemble* yang diusulkan dapat menjadi referensi bagi penelitian berikutnya terutama dalam hal yang berkaitan dengan penanganan terhadap ketidakseimbangan kelas pada dataset pada proses klasifikasi untuk pengembangan berbagai sistem deteksi web phishing.

II. METODOLOGI PENELITIAN

Penelitian ini dilakukan berdasarkan tahapan penelitian yang dilakukan secara berurutan sesuai dengan alur yang ditunjukkan pada Gambar.1.



Gambar.1. Alur Tahapan penelitian

Alur proses penelitian sebagaimana yang ditunjukkan Pada gambar 1. Meliputi proses akuisisi dataset, dilanjutkan dengan tahapan pra-pemrosesan data, berikutnya adalah proses klasifikasi yang menjadi usulan utama pada penelitian ini, dan terakhir adalah tahapan evaluasi yang bertujuan untuk mengetahui kinerja dari model yang diusulkan dalam penelitian. Adapun penjelasan lebih lanjut dari setiap tahapan akan dijelaskan pada masing-masing sub-bab.

A. Dataset

Dataset yang digunakan pada peneltiian ini adalah Web phising dataset yang dapat diakusisi secara langsung dan gratis melalui halaman website UCI Machine Learning Repository [23]. Terdapat dua jenis dataset yang akan diujikan pada penelitian ini, dimana kedua dataset tersebut memiliki kondisi yang tidak seimbang pada kelas datanya. Hal yang membedakan kedua dataset tersebut terletak pada banyaknya kelas data, dimana pada dataset web phising Binary Class mengandung dua kelas data, yakni sah (*legitimate*) dan Phising, sedangkan pada dataset web phising Multiclass memiliki tiga kelas data, yakni sah (*legitimate*), Suspect, dan Phising. Informasi singkat dari kedua dataset yang digunakan dalam peneliain ini ditunjukkan menggunakan Tabel.I.

TABEL I
INFORMASI WEB PHISING DATASET

Datasets	Class	Instance	Attributes	Majority Class	Minority Class	Imbalanced Ratio
Web Phising Binary Class [23]	2	2456	31	1362	1094	1.25
Web Phising Multiclass [24]	3	1353	10	702	103	6.82

B. Pra-pemrosesan Data

Tahapan pra-pemrosesan data merupakan tahapan kedua yang dilakukan setelah dataset diakuisisi. Proses yang dilakukan pada tahapan ini meliputi proses untuk menelaah dan memahami karakteristik dari masing-masing dataset yang digunakan dalam penelitian, dan memastikan bahwa data tersebut siap untuk digunakan tanpa adanya missing data, tanpa adanya data yang redundan, dan tanpa adanya data yang inkonsisten.

Setelah karakteristik dataset dipahami, proses berikutnya pada pra-pemrosesan data adalah membagi dataset tersebut menjadi data latih (*training data*) dan data uji (*testing data*). Proses pembagian data dilakukan berdasarkan rasio perbandingan antara training data dan testing data sebesar 80:20. Data yang tergabung dalam data latih berikutnya akan dilakukan proses pemodelan berdasarkan skenario penelitian. Sedangkan data yang tergabung sebagai data uji, merupakan data yang nantinya akan digunakan untuk menguji skenario model yang diusulkan untuk mengetahui kinerjanya.

Terdapat tiga skenario pemodelan yang akan digunakan pada peneltian ini, yang pertama adalah skenario perancangan model *machine learning* yang mengimplementasikan model klasifikasi tunggal yang didasarkan pada penggunaan algoritma klasifikasi populer seperti *Decision Tree*, *Naive Bayes*, dan *Support Vector Machine*. Skenario kedua merupakan skenario yang menggunakan algoritma *Random Forest* sebagai model klasifikasinya, dan skenario ketiga adalah penggunaan metode stacking sebagai *ensemble learning* dalam proses klasifikasi dataset web phising.

C. Ensemble Classification

Ensemble classification merupakan teknik klasifikasi yang memanfaatkan beberapa algoritma klasifikasi untuk kemudian dikombinasikan, digabungkan, maupun dibandingkan sebagai salah satu cara untuk menemukan solusi prediksi yang terbaik [19]–[22]. Pada penelitian ini, *ensemble model* yang diusulkan adalah *Random Forest* dan metode *Stacking*.

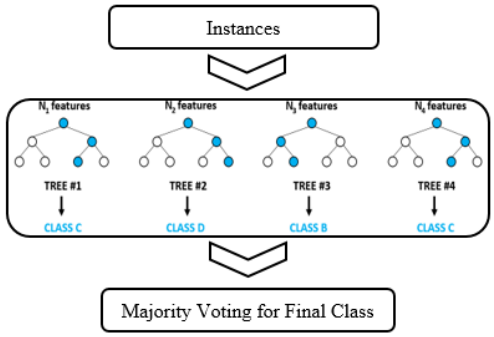
1) Random Forest

Random Forest merupakan algoritma yang pertama kali diperkenalkan [25] pada penelitian yang menyebutkan tentang keunggulannya dalam menyelesaikan masalah yang berkaitan dengan tugas klasifikasi maupun regresi. Random Forest dikembangkan berdasarkan turunan dari algoritma decision tree dengan tujuan untuk mengatasi masalah yang berkaitan dengan tingginya kemungkinan terjadinya *overfitting* apabila menggunakan algoritma decision tree [22].

Cara kerja Random Forest termasuk dalam kategori ensemble learning karena memanfaatkan kombinasi dari beberapa pohon keputusan sebagai *classifier* dasar untuk memprediksi nilai dari suatu data [22]. Terdapat tiga aspek utama yang dimiliki oleh algoritma random forest, yakni:

- Penggunaan sample individu pada setiap atribut yang dilakukan secara acak (*random*) terutama untuk membangkitkan setiap pohon keputusan.
- Setiap n pohon keputusan yang terbentuk akan melakukan prediksi dan memberikan keputusan sesuai dengan prediktor yang terbentuk.
- Random forest akan melakukan proses majority voting untuk menentukan hasil prediksi secara keseluruhan berdasarkan hasil prediksi yang diperoleh dari setiap n pohon keputusan yang dibangkitkan.

Ilustrasi yang menunjukkan proses pembentukan algoritma random forest dapat dilihat melalui Gambar 2.



Gambar 2. Ilustrasi proses dari algoritma Random Forest.

Pada Gambar 2., instance data yang akan diproses menggunakan algoritma random forest, pertama-tama akan akan diambil nilai atributnya secara acak (random) untuk membangkitkan sejumlah n pohon keputusan. Terdapat dua parameter penting yang diperlukan dalam tahap pembangkitan pohon keputusan, parameter pertama adalah berapa jumlah pohon yang akan dibangkitkan (n), dan parameter kedua adalah banyaknya fitur maksimal yang dipertimbangkan ketika melakukan proses percabangan pada saat membangkitkan setiap satu pohon keputusan (k).

Pada penelitian ini, nilai n yang digunakan adalah 11, artinya akan terbentuk 11 pohon keputusan secara *bootstrap* yang nantinya akan digunakan oleh random forest dalam menentukan hasil prediksi, sedangkan untuk nilai k akan diambil nilai secara *default* yakni sebesar nilai akar kuadrat dari jumlah keseluruhan fitur yang ada pada dataset. Setelah semua pohon keputusan terbentuk, langkah berikutnya yang dilakukan oleh masing-masing pohon keputusan tersebut adalah melakukan prediksi berdasarkan *tree predictor* yang membuat setiap pohon keputusan akan memiliki hasil prediksinya masing-masing. setelahnya, hasil prediksi dari masing-masing pohon keputusan akan dilakukan proses *majority voting* untuk memperoleh jumlah terbanyak dari kelas data yang diprediksi setiap *tree*, yang akan ditetapkan sebagai nilai prediksi final pada random forest.

Untuk itu, pada algoritma random forest yang dibangun menggunakan sebanyak pohon akan dapat dirumuskan melalui persamaan (1) sebagai berikut:

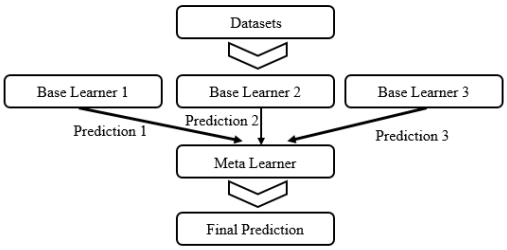
$$y = argmax(\sum_{n=1}^{n} i_{hn=c}) \quad (1)$$

Dimana y merupakan hasil prediksi akhir (final) yang akan diambil, sedangkan $argmax$ merupakan fungsi yang akan mengembalikan nilai terbesar dari setiap n pohon berdasarkan fungsi indikator i pada setiap nilai prediksi dari pohon h_n yang terbentuk [22].

2) Stacking

Stacking termasuk dalam kategori ensemble learning yang didalamnya terdapat proses tumpukan dari beberapa algoritma klasifikasi tunggal [20]. Dalam penggunaannya, terdapat dua level model pembelajaran (*learner*) yang akan digunakan ketika metode stacking diterapkan, yang pertama adalah model pembelajaran level-0 yang dikenal sebagai *base learner*, dan model pembelajaran level-1 yang merupakan *meta-learner* [22].

Ilustrasi dari proses stacking digambarkan melalui Gambar 3. Sebagai berikut:



Gambar 3. Ilustrasi proses dari metode stacking.

Pada Gambar 3., proses klasifikasi yang dilakukan menggunakan metode stacking, secara sederhana dilakukan berdasarkan dataset asli yang kemudian dilakukan proses pelatihan (*training*) menggunakan beberapa model pembelajaran yang berbeda. Hasil dari masing-masing model tersebut, kemudian dikumpulkan, dan akan digunakan sebagai dataset baru, yang setiap instance datanya memiliki keterkaitan dengan nilai yang ada pada dataset asli yang seharusnya diprediksi secara langsung. Dataset baru yang terbentuk dari penggabungan beberapa model pembelajaran yang berbeda kemudian diproses kembali menggunakan model pembelajaran level-1 (*meta-learner*) untuk memperoleh hasil prediksi secara final [20], [22]. Adapun pseudocode dari proses stacking ditunjukkan menggunakan Gambar 4.

Input: Data set $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
First-level learning algorithms $\mathfrak{L}_1, \ldots, \mathfrak{L}_T$;
Second-level learning algorithm $\mathfrak{L}$.

Process:
for $t = 1, \ldots, T$: % Train a first-level learner by applying the
  $h_t = \mathfrak{L}_t(D)$; % first-level learning algorithm $\mathfrak{L}_t$
end
$D' = \emptyset$; % Generate a new data set
for $i = 1, \ldots, m$:
  for $t = 1, \ldots, T$:
    $z_{it} = h_t(\boldsymbol{x}_i)$;
  end
  $D' = D' \cup ((z_{i1}, \ldots, z_{iT}), y_i)$;
end
$h' = \mathfrak{L}(D')$; % Train the second-level learner h' by applying
% the second-level learning algorithm $\mathfrak{L}$ to the
% new data set D'.

Output: $H(\boldsymbol{x}) = h'(h_1(\boldsymbol{x}), \ldots, h_T(\boldsymbol{x}))$

Gambar 4. Pseudocode Proses Stacking.

Dari Pseudocode yang ditunjukkan pada Gambar 4. Secara teknis, metode stacking hanya memerlukan dataset, model pembelajaran level-0 (*base learner*), dan model pembelajaran level-1 (*meta learner*) sebagai input yang diperlukan. Oleh karena itu, pada penelitian kali ini, algoritma klasifikasi populer seperti Decision Tree dan Naive Bayes menjadi usulan model pembelajaran yang akan menjadi input dalam penggunaan metode stacking. Alasan utama mengapa Decision Tree dan Naive Bayes dipilih sebagai model pembelajaran karena kedua algoritma tersebut merupakan algoritma klasifikasi populer dengan kinerja yang baik pada berbagai permasalahan, namun tidak cukup baik apabila digunakan secara langsung pada data yang memiliki masalah pada ketidakseimbangan (*imbalanced*) kelas data.

D. Evaluasi

Tahapan terakhir dari penelitian ini adalah melakukan proses evaluasi serta mengukur kinerja dari berbagai skenario model yang diusulkan dalam penelitian. Confusion matrix menjadi salah satu cara yang sampai dengan saat ini masih dianggap efektif untuk mengukur dan mengevaluasi kinerja dari sebuah model klasifikasi [20].

TABEL II
CONFUSION MATRIX [20]

Aktual	Prediksi	
	Positif	**Negatif**
Positif	*TP*	*FN*
Negatif	*FP*	*TN*

Tabel II merupakan bentuk ilustrasi dari confusion matrix yang digunakan sebagai indikator dalam mengukur serta mengevaluasi kinerja klasifikasi khususnya untuk klasifikasi dua kelas (*binary classification*). Penggunaan confusion matrix bertujuan untuk mengetahui seberapa besar nilai prediksi yang dihasilkan oleh sistem untuk kemudian dibandingkan dengan nilai aktual dari datanya. Penilaian yang dilakukan berdasarkan indikator dari confusion matrix diambil dari banyaknya nilai pada komponen True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN).

TP merupakan komponen yang menunjukkan banyaknya jumlah prediksi nilai positif terhadap keseluruhan data yang secara aktual juga bernilai positif. Sedangkan FP, merupakan komponen yang menunjukkan banyaknya jumlah prediksi nilai positif namun secara aktual bernilai negatif. Berikutnya adalah FN, yang berisi banyaknya jumlah data yang diprediksi bernilai negatif, namun secara aktualnya bernilai positif, dan TN menunjukkan banyaknya jumlah prediksi sistem yang bernilai negatif, dengan data aktual juga bernilai negatif [22].

Nilai yang diperoleh dari setiap komponen tersebut, kemudian dilakukan perhitungan untuk mengetahui kinerja klasifikasi berdasarkan metrik akurasi, presisi, recall, F1-Score, dan Area Under Curve (AUC). Kelima metrik tersebut dipilih karena sifatnya yang komprehensif terutama pada kasus yang memiliki bentuk ketidakseimbangan (*imbalanced*) kelas data.

Akurasi merupakan metrik yang digunakan untuk mengetahui proporsi dari jumlah prediksi sistem yang bernilai benar (TP dan TN) dengan jumlah dari keseluruhan hasil prediksi (TP, FP, FN, TN). Persamaan untuk menghitung nilai akurasi dalam bentuk prosentase ditunjukkan melalui persamaan (2).

$$akurasi\ (\%) = \frac{TP+TN}{TP+FP+FN+TN} * 100 \quad (2)$$

Metrik berikutnya yang digunakan untuk mengukur kinerja model adalah Presisi, yang merupakan metrik untuk mengukur seberapa besar rasio prediksi nilai positif yang sesuai dengan nilai aktual (TP) dibandingkan dengan keseluruhan hasil

prediksi yang bernilai positif (TP dan FP). Untuk menghitung prosentase besaran nilai presisi, digunakan persamaan (3).

$$Presisi\ (\%) = \frac{TP}{TP+FP} * 100 \quad (3)$$

Selanjutnya ada metric pengukuran menggunakan Recall, yang bertujuan untuk mengetahui proporsi jumlah data yang diprediksi bernilai positif (TP) dibandingkan dengan seluruh data yang secara aktual bernilai positif (TP dan FN). Persamaan (4) menunjukkan formula yang digunakan untuk menghitung nilai Recall dalam bentuk prosentase.

$$Presisi\ (\%) = \frac{TP}{TP+FP} * 100 \quad (4)$$

Kemudian ada F1-Score, yang merupakan metric pengukuran untuk mengetahui perbandingan rata-rata antara nilai Presisi dan nilai Recall. Persamaan (5) digunakan untuk memperoleh prosentase nilai F1-Score.

$$F1\ (\%) = \frac{2 * (Recall * Presisi)}{Recall + Presisi} * 100 \quad (5)$$

Terakhir ada metrik pengukuran menggunakan Area Under Curve (AUC) yang merupakan metrik untuk mengetahui apakah kinerja model yang dihasilkan termasuk dalam representasi klasifikasi yang baik (*good classifier*) atau klasifikasi yang buruk (*bad classifier*).

III. Hasil dan Pembahasan

Sebagaimana telah disampaikan pada bagian sebelumnya yaitu metodologi penelitian, bahwasanya, terdapat dua buah dataset yang menjadi bahan utama dalam penelitian ini, dataset tersebut adalah Web phising dataset binary class, dan web phising dataset multiclass. Kedua dataset tersebut dapat diunduh dengan mudah dan gratis melalui halaman website UCI Machine Learning Repository [23], [24].

Masing-masing dataset tersebut kemudian dilakukan pra-pemrosesan terutama untuk dilakukan proses pembagian (data splitting) kedalam dua set data dengan proporsi sebesar 80% untuk data latih (training data), dan 20% digunakan sebagai data uji (testing data). Skenario yang diusulkan pada penelitian ini meliputi tiga bentuk skenario model klasifikasi, yang pertama adalah skenario yang memodelkan secara langsung menggunakan algoritma klasifikasi tunggal, skenario kedua adalah dengan menggunakan random forest sebagai algoritma untuk klasifikasi, dan skenario ketiga adalah dengan menggunakan metode stacking dalam memprediksi adanya web phising.

Tabel III
Perbandingan Kinerja dari skenario model klasifikasi

Datasets	Algoritme	Akurasi	Presisi	Recall	F1-Score	AUC
Web Phishing Binary Class IR = 1,25	Decision Tree	94.3%	94.3%	94.3%	94.3%	96.4%
	Naïve Bayes	94.2%	94.2%	94.2%	94.2%	98.7%
	Support Vector Machine [SVM]	83.5%	83.5%	83.5%	83.5%	91.9%
	Stacking [Decision Tree + Naive Bayes]	95.1%	95.1%	95.1%	95.1%	99.0%
	Random Forest	96.4%	96.4%	96.4%	96.4%	99.3%
Web Phishing Multi Class IR = 6,82	Decision Tree	87.7%	87.7%	87.7%	87.7%	93.5%
	Naïve Bayes	83.6%	81.6%	83.6%	82.6%	94.3%
	Support Vector Machine [SVM]	84.5%	83.5%	84.5%	84.0%	94.3%
	Stacking [Decision Tree + Naive Bayes]	88.8%	87.9%	88.8%	88.3%	96.1%
	Random Forest	88.0%	87.9%	88.0%	87.9%	95.7%

Hasil penelitian pada Tabel III., menunjukkan bahwa, secara umum, peningkatan kinerja klasifikasi terjadi di semua metrik pengukuran ketika algoritma random forest dan metode stacking digunakan sebagai model klasifikasinya. Hal tersebut terlihat pada pengujian yang dilakukan menggunkaan dataset pertama yaitu dataset web phising binary class, yang memiliki nilai imbalance rasio (IR) sebesar 1.25%, penggunaan algoritma random forest menjadi yang terbaik kinerjanya karena menghasilkan nilai sebesar 96.4% ketika dievaluasi menggunakan metrik akurasi, presisi, recall dan F1-Score, sedangkan ketika dievaluasi menggunakan metrik AUC, nilai kinerja yang dihasilkan sebesar 99.3% yang juga merupakan nilai terbesar jika dibandingkan dengan penggunaan algoritma klasifikasi tunggal yang popular seperti Decision Tree, Naïve Bayes, maupun Support Vector Machine (SVM).

Sama halnya dengan pengujian pada dataset web phising binary class, pengujian yang dilakukan menggunakan dataset web phising multiclass juga menunjukkan adanya peningkatan kinerja klasifikasi ketika menerapkan algoritma random forest dan metode stacking dibandingkan dengan hasil yang hanya menerapkan model klasifikasi tunggal. Dengan nilai imbalance rasio sebesar 6.82 % yang dimiliki dataset web phising multiclass, nilai kinerja yang dihasilkan dari penggunaan random forest dan metode stacking terlihat mengalami adanya peningkatan hingga mencapai 4% pada metrik akurasi, peningkatan kinerja yang mencapai hampir 5% pada metrik presisi, peningkatan kinerja hingga 3% pada metrik recall, peningkatan kinerja hampir sebesar 2% pada F1-Score, dan peningkatan kinerja sebesar hampir 3% pada AUC.

Sedangkan jika menilik secara keseluruhan metrik yang digunakan sebagai evaluasi kinerja model, penggunaan metode stacking yang menggabungkan antara algoritma Decision Tree dan Naïve bayes memiliki nilai tertinggi dibandingkan dengan penggunaan model klasifikasi lainnya.

Oleh karena itu, apabila melihat dari keseluruhan hasil yang diperoleh, maka dapat diketahui adanya keunggulan yang

dimiliki oleh algoritma random forest maupun metode stacking yakni mampu menangani adanya ketidakseimbangan pada kelas data yang dapat mempengaruhi proses klasifikasi sehingga menghasilkan kinerja yang kurang maksimal. Selain itu, keunggulan lain yang dapat ditunjukkan menggunakan algoritma random forest dan metode stacking ialah penanganan ketidakseimbangan kelas pada dataset, tidak merubah komposisi dan pola yang terdapat pada dataset sehingga informasi pada dataset tetap terjaga sebagaimana dataset yang aslinya.

IV. KESIMPULAN

Dari berbagai pengujian yang dilakukan pada dua jenis dataset yang keduanya memiliki masalah pada ketidakseimbangan (*imbalanced*) kelas data, penggunaan algoritma random forest dan metode stacking yang menggabungkan antara decision tree dan naïve bayes menghasilkan kinerja yang lebih baik dibandingkan dengan penggunaan algoritma popular seperti Decision Tree, Naïve Bayes, dan Support Vector Machine (SVM) yang diproses sebagai model klasifikasi tunggal. Nilai akurasi yang dihasilkan dari penggunaan random forest sebesar 96.4% merupakan nilai kinerja terbaik pada dataset dengan imbalanced ratio sebesar 1.25%, sedangkan nilai akurasi sebesar 88.8% yang dihasilkan dari dataset dengan imbalanced ratio sebesar 6.82%, diperoleh menggunakan metode stacking yang menggabungkan decision tree dan naïve bayes. Oleh karena adanya hasil tersebut, maka penggunaan algoritma random forest dan metode stacking yang memanfaatkan decision tree dan naïve bayes sebagai model klasifikasi memiliki potensi untuk dapat digunakan sebagai solusi dalam menangani adanya ketidakseimbangan pada dataset khususnya yang berkaitan dengan pengembangan model klasifikasi untuk proses deteksi web phising. Untuk penelitian selanjutnya, perlu dilakukan percobaan menggunakan hybrid method yang memanfaatkan ensemble model dengan teknik resampling yang bertujuan untuk semakin meningkatkan dan memaksimalkan kinerja model klasifikasi terutama yang berkaitan dengan pengembangan sistem untuk deteksi web phising.

REFERENSI

[1] S. P. Kinkin Yuliaty, P. Elisabeth Nugrahaeni, and S. Dini, "Literasi media baru dan budaya baru," *Semin. Nas. Dies Fisip Unsoed Ke 34 Media, Budaya, dan Polit. di Era Milen.*, vol. 1, no. 1, pp. 229–236, 2018.

[2] M. Ganesan and P. Mayilvahanan, "Cyber Crime Analysis in Social Media Using Data Mining Technique," *Int. J. Pure Appl. Math.*, vol. 116, no. 22, pp. 413–424, 2017.

[3] M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," *6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding*, vol. 2018-Janua, pp. 1–5, 2018.

[4] R. Kiruthiga and D. Akila, "Phishing websites detection using machine learning," *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 11, pp. 111–114, 2019.

[5] A. Kulkani and L. L. Brown, "Phishing websites detection using machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 8–13, 2019.

[6] A. Razaque, M. B. H. Frej, D. Sabyrov, A. Shaikhyn, F. Amsaad, and A. Oun, "Detection of phishing websites using machine learning," *Proc. - 2020 IEEE Cloud Summit, Cloud Summit 2020*, vol. 10, no. 05, pp. 103–107, 2020.

[7] G. Harinahalli Lokesh and G. BoreGowda, "Phishing website detection based on effective machine learning approach," *J. Cyber Secur. Technol.*, vol. 5, no. 1, pp. 1–14, 2021.

[8] L. Al-Shalabi, "Comparative study of data mining classification techniques for detection and prediction of phishing websites," *J. Comput. Sci.*, vol. 15, no. 3, pp. 384–394, 2019.

[9] N. S. Zaini *et al.*, "Phishing detection system using machine learning classifiers," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1165–1171, 2019.

[10] Anti-Phishing Working Group (APWG), "PHISHING ACTIVITY TRENDS REPORT," 2021. [Online]. Available: https://docs.apwg.org/reports/apwg_trends_report_q4_2021.pdf.

[11] Y. C. G. Tomy Salim, "Data Mining Identifikasi Website Phising Menggunakan Algoritma C4.5," *J. TAM (Technol. Accept. Model)*, vol. 8, no. 2, pp. 130–135, 2017.

[12] G. Kamal and M. Manna, "Detection of Phishing Websites Using Naïve Bayes Algorithms," *Int. J. Recent Res. Rev.*, vol. XI, no. 4, pp. 34–38, 2018.

[13] A. Saifudin, F. Teknik, U. Pamulang, R. S. Wahono, F. I. Komputer, and U. D. Nuswantoro, "Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software," *J. Softw. Eng.*, vol. 1, no. 1, pp. 28–37, 2015.

[14] A. Fatkhurohman and E. Pujastuti, "Penerapan Algoritma Naïve Bayes Classifier untuk Meningkatkan Keamanan Data dari Website Phising," *J. Teknol. Inf.*, vol. 15, no. 1, pp. 115–124, 2019.

[15] A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *J. King Saud Univ. - Comput. Inf. Sci.*, 2019.

[16] S. A. Gyamerah, P. Ngare, and D. Ikpe, "On Stock Market Movement Prediction Via Stacking Ensemble Learning Method," *CIFEr 2019 - IEEE Conf. Comput. Intell. Financ. Eng. Econ.*, no. Ml, pp. 1–8, 2019.

[17] Y. Pristyanto, I. Pratama, and A. F. Nugraha, "Data level approach for imbalanced class handling on educational data mining multiclass classification," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, 2018, pp. 310–314.

[18] V. Estivill-Castro, M. Lombardi, and A. Marani, "Improving binary classification of web pages using an ensemble of feature selection algorithms," *ACM Int. Conf. Proceeding Ser.*, 2018.

[19] H. S. Hota, A. K. Shrivas, and R. Hota, "An Ensemble Model for Detecting Phishing Attack with Proposed Remove-Replace Feature Selection Technique," *Procedia Comput. Sci.*, vol. 132, pp. 900–907, 2018.

[20] A. Nurmasani and Y. Pristyanto, "Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class," *Pseudocode*, vol. 8, no. 1, pp. 21–26, 2021.

[21] V. Muppavarapu, A. Rajendran, and S. K. Vasudevan, "Phishing detection using RDF and random forests," *Int. Arab J. Inf. Technol.*, vol. 15, no. 5, pp. 817–824, 2018.

[22] A. F. Nugraha and L. Rahman, "Meta-algorithms for improving classification performance in the web-phishing detection process," *2019 4th Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2019*, vol. 6, pp. 271–275, 2019.

[23] UCI Machine Learning, "UCI Machine Learning Repository : Phising Websites Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/phishing+web%0Asites.

[24] UCI ML, "UCI Machine Learning Repository : Phishing Websites Data Set." [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Phishing+Websites.

[25] L. Breiman, "Bagging predictors: Technical Report No. 421," *Dep. Stat. Univ. Calif.*, no. 2, p. 19, 1994.