Pengaruh Feature Selection terhadap Kinerja C5.0, XGBoost, dan Random Forest dalam Mengklasifikasikan Website Phishing

Michael Jonathan, Silvia Rostianingsih, Henry Novianus Palit
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) – 8417658
Email: michaeljonathan518@gmail.com, silvia@petra.ac.id, hnpalit@petra.ac.id

ABSTRAK

Dengan meningkatnya pengguna internet khususnya *website*, memberikan celah bagi para pelaku *phishing* untuk mendapatkan atau mencuri informasi pribadi dari para pengguna. Pada setiap *website* akan terdapat banyak informasi yang akan dijadikan *feature*, *feature* tersebut akan dimanfaatkan untuk mengklasifikasikan *website phishing*. *Feature* akan dibagi menjadi 3 yaitu *url feature*, *content feature*, dan *external feature*.

Dalam penelitian ini akan menggunakan tiga metode yaitu C5.0, XGBoost, dan Random Forest. Ketiga metode akan diuji kinerjanya untuk mencari metode terbaik dalam mengklasifikasikan *website phishing*. Selain itu penelitian ini juga akan memanfaatkan *feature selection* dengan tujuan untuk membuang *feature* yang tidak berpengaruh sehingga *training time* dapat dipersingkat.

Berdasarkan hasil pengujian yang diperoleh menunjukan, bahwa C5.0 mampu memberikan nilai *accuracy, precision, recall, & f1-score* dengan rata-rata sebesar 93,5%, XGBoost dengan rata-rata sebesar 96,6%, dan Random Forest dengan rata-rata sebesar 95,7%. Penggunaan *feature selection* pada ketiga algoritma juga menunjukan, bahwa *training time* dapat dipersingkat dengan rata-rata sekitar 3,53 kali lebih cepat dengan hanya menggunakan 15 *feature importance* saja. Namun dengan penggunaan *feature selection,* performa pada nilai *accuracy, precision, recall, & f1-score* terjadi sedikit penurunan walaupun penurunan yang diberikan tidak signifikan atau tidak berdampak besar pada proses klasifikasi.

Kata Kunci: *Feature Selection, Website Phishing,* Random Forest, C5.0, XGBoost

ABSTRACT

With the increase in internet users, especially websites, it provides an opportunity for phishing actors to obtain or steal personal information from users. On each website there will be a lot of information that will be used as a feature, this feature will be used to classify phishing websites. Features will be divided into 3, namely feature url, content feature, and external feature.

In this study, three methods will be used, namely C5.0, XGBoost, and Random Forest. The three methods will be tested for their performance to find the best method for classifying phishing websites. In addition, this research will also utilize feature selection with the aim of removing features that have no effect so that training time can be shortened.

Based on the test results obtained, it shows that C5.0 is able to provide accuracy, precision, recall, & f1-score values with an average of 93.5%, XGBoost with an average of 96.6%, and Random Forest with an average of 95.7 %. The use of feature selection in the three algorithms also shows that training time can be shortened by an average of about 3.53 times faster by using only 15 feature importance. However, with the use of feature selection, the performance on accuracy, precision, recall, & f1-score values decreased slightly even though the given decrease was not significant or had no major impact on the classification process.

Keywords: *Feature Selection, Website Phishing, Random Forest, C5.0, XGBoost*

1. LATAR BELAKANG

Dengan meningkatnya pengguna internet khususnya *website* dapat memberikan celah bagi para pelaku *phishing* untuk mencuri informasi pribadi dari pengguna. *Phishing* seringkali berhasil dilakukan karena pengguna tidak menyadari atau tidak dapat membedakan *website phishing* dan *non phishing*, bahkan ada beberapa *website phishing* yang dirancang sedemikian rupa sehingga mirip dengan aslinya [9]. Pada bulan desember 2021 tercatat sebanyak S$13,7 juta hilang dalam rentetan *phishing* yang baru terjadi dan melibatkan OCBC Bank [4]. Pelanggan yang terkena dampak juga meningkat dari 469 menjadi 790 orang dan bisa terus bertambah. Penipu memanfaatkan *log-in credentials* dan PIN yang diberikan oleh pengguna pada saat mengakses *website phishing* sehingga penipu leluasa mengambil alih rekening bank pengguna. Untuk itu perlu dilakukan klasifikasi *website phishing* agar pengguna dapat membedakan *website* yang mereka akses itu *phishing* atau *legitimate*.

Pada penelitian oleh Kumar et al, pada tahun 2020 [9]. Tujuan dari penelitian ini yaitu membandingkan berbagai metode seperti Logistic Regression, Random Forest, Gaussian Naive Bayes, Decision Tree, hingga K-Nearest Neighbors dalam mengklasifikasikan *website phishing*. Hasil terbaik adalah metode Random Forest dengan *accuracy* sebesar 98,03%, *precision* sebesar 100%, *recall* sebesar 96%, dan *f1-score* sebesar 98%. Pada penelitian oleh Aminu et al, pada tahun 2019 [1]. Tujuan dari penelitian ini yaitu membandingkan ketiga algoritma yaitu Random Forest, XGBoost, dan PNN (Probabilistic Neural Network) dengan menggunakan 30 *feature* untuk mengklasifikasikan *website phishing*. Hasil terbaik adalah metode XGBoost dengan *accuracy* sebesar 97,13%, *precision* sebesar 97,13%, *recall* sebesar 97,73%, dan *f1-score* sebesar 97,43%.

Pada penelitian oleh Machado & Gadge, pada tahun 2017 [10]. Tujuan dari penelitian ini yaitu mendeteksi *website phishing* menggunakan algoritma C4.5 Decision Tree. Hasil terbaik adalah metode XGBoost dengan *accuracy* sebesar 89,30%, *precision* sebesar 87,13%, *recall* sebesar 88,60%, dan *f-measure* sebesar 88,20%.

Pada penelitian ini penulis akan membandingkan tiga metode yaitu C5.0, XGBoost, dan Random Forest dari segi *accuracy, precision, recall,* dan *f1-score* dalam mengklasifikasikan *website phishing*. Tujuan penelitian ini yaitu dapat membandingkan ketiga algoritma tersebut dikarenakan belum ada penelitian yang membandingkan ketiganya, ditambah pada masing-masing penelitian menunjukan bahwa ketiga algoritma ini memiliki tingkat *accuracy* yang tinggi. Selain itu penelitian ini juga menggunakan *dataset* dengan 87 *feature* dan 11.431 *instance*, semakin banyak *feature* digunakan dalam proses klasifikasi memang lebih baik tetapi jika menggunakan semua *feature*, waktu yang diperlukan untuk memproses data akan sangat besar. Untuk itu diperlukan *feature selection* yang dapat memilih *feature* yang berpengaruh besar saja sehingga waktu yang diperlukan untuk memproses data relatif lebih cepat dengan akurasi yang tetap tinggi. Ketiga algoritma ini akan dibandingkan dengan menggunakan atau tanpa menggunakan *feature selection* untuk mengetahui pengaruh *feature selection* terhadap kinerja ketiga algoritma.

2. LANDASAN TEORI

2.1 Website Phishing

Phishing biasanya didefinisikan sebagai perolehan data *privacy* seseorang secara curang atau tanpa seizin pengguna dan menyalahgunakan data yang dicuri tersebut. Serangan *phishing* seringkali dilakukan melalui *email*, *email* ini berisi tautan URL yang mengarahkan pengguna ke situs web lain. Situs ini sebenarnya adalah situs palsu atau modifikasi sehingga saat pengguna membuka situs ini, mereka diminta memasukkan informasi pribadi untuk diteruskan ke penyerang *phishing* [2].

2.2 C5.0

C5.0 adalah salah satu algoritma yang merupakan bagian dari decision tree. Algoritma ini dibentuk pada tahun 1987 oleh Ross Quinlan yang mana merupakan pengembangan dari metode terdahulu seperti ID3 dan C4.5. Algoritma C5.0 akan memilih atribut dengan *gain ratio* tertinggi, lalu atribut yang memiliki *gain ratio* tertinggi tersebut nantinya akan dipilih sebagai *parent* pada *node* selanjutnya. Langkah-langkah yang harus dilakukan untuk mendapatkan *gain ratio* yaitu menghitung nilai *entropy*, *information gain* kemudian *gain ratio*. Perhitungan *entropy* dan *information gain* yang pada algoritma C5.0 adalah bagian dari C4.5, perbedaan hanya terletak pada proses *boosting*, dan *voting* yang akan digunakan untuk menentukan kelas berdasarkan hasil dari perhitungan kombinasi dari beberapa tree [6].

2.3 XGBoost

XGBoost (Extreme Gradient Boosting) merupakan *upgrade version* dari Gradien Boosting yang telah dioptimalkan dan dirancang dengan penambahan gradien agar lebih efisien, fleksibel dan portabel. XGBoost juga dapat menyediakan *boosted trees parallel* secara cepat dan akurat [13]. XGBoost adalah sebuah algoritma machine learning yang sering dimanfaatkan untuk melakukan regresi maupun klasifikasi yang berdasarkan gradient boosting tree. Pada pohon regresi, *nodes* yang ada pada bagian dalam akan mewakili nilai-nilai untuk tes atribut serta skor dari *leaf nodes* yang akan mewakili *decision* [7].

2.4 Random Forest

Random Forest Classifier merupakan salah satu metode *ensemble supervised classification* yang memanfaatkan decision tree dengan jumlah yang besar secara bersamaan. Fitur terbaik akan dipilih secara *random* lalu setiap decision tree akan digunakan dalam mengklasifikasikan objek dari *input vector*. Klasifikasi dengan *votes* terbanyak akan dipilih oleh random forest [12].

2.5 Feature Selection

Feature selection adalah metode filter untuk memilih fitur berdasarkan kinerjanya. Metode ini akan memilih fitur terbaik untuk diidentifikasi sehingga dapat digunakan untuk proses klasifikasi, selain itu metode ini menempatkan peringkat pada fitur individu atau pada *subset*. Langkah-langkah dalam mengembangkan menyaring fitur dapat diklasifikasikan kedalam: *information, distance, consistency, similarity*, dan *statistical measures*. *Feature selection* dapat dilakukan secara *univariate* atau *multivariate*. *Univariate* yaitu melakukan pemilihan fitur berdasarkan satu fitur pada satu waktu sedangkan *multivariate* yaitu mempertimbangkan *subset* fitur sekaligus [11].

Embedded method adalah metode pada *feature selection* yang berfungsi untuk pencarian atau pemilihan *subset* yang optimal untuk digunakan dalam membangun konstruksi model dari klasifikasi. Keunggulan dari metode ini yaitu dapat mengurangi waktu komputasi pada proses pencarian *subset* fitur [5]. Salah satu pendekatan pada *embedded method* adalah *feature importance*. *Feature importance* berfungsi untuk menganalisis mengenai fitur-fitur dari data dengan memanfaatkan *model.feature_importance* dalam paket *sklearn*, fitur penting atau yang berkontribusi lebih banyak akan digunakan untuk meningkatkan kinerja dari model [8].

2.6 Dataset Website Phishing

Dataset didapat dari Kaggle pada *shashwatwork/web-page-phishing-detection-dataset dalam* bentuk file.CSV yang memiliki 87 *feature* dan 11.431 *instance*. Dalam mengklasifikasikan *website phishing feature* dibagi menjadi 3 *feature* penting yaitu *url features*, *content features*, dan *external features*.

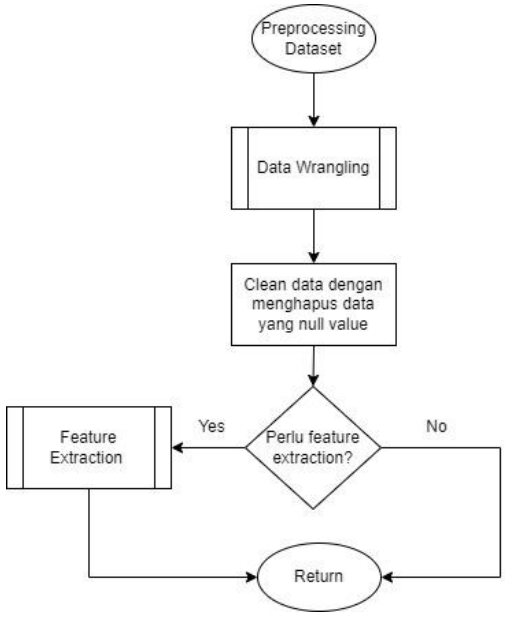
2.7 Confusion Matrix

Confusion matrix biasanya digunakan dalam menyelesaikan masalah pada klasifikasi dua atau multi kelas. Tujuan dari *confusion matrix* dapat memberikan informasi dari perbandingan hasil klasifikasi oleh model yang dibangun dengan hasil klasifikasi aktual. Pada klasifikasi dengan dua kelas akan mengklasifikasikan kasus menjadi dua set kelas dengan empat kemungkinan: kombinasi antara *True Positive*, *true negative*, *false positive*, dan *false negative* [3].

3. DESAIN SISTEM

3.1 Preprocessing Dataset

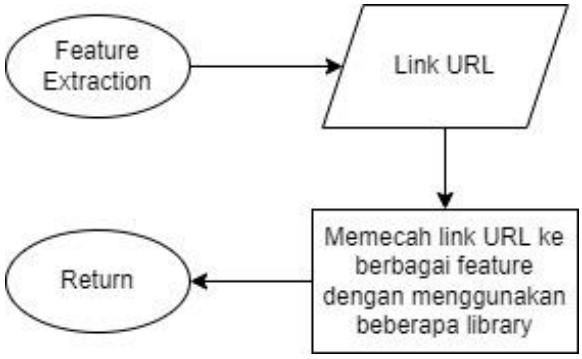
Preprocessing data merupakan sebuah proses atau langkah yang dilakukan untuk mempersiapkan data pada *dataset website phishing* sebelum digunakan untuk klasifikasi. Manfaat dari melakukan *preprocessing data* yaitu dapat menghilangkan *noise* pada data sehingga proses klasifikasi dapat berjalan tanpa ada hambatan. Proses pada *preprocessing data* dapat dilihat pada Gambar 1.



Gambar 1. Proses pada *Preprocessing Data*

3.2 Process Feature Extraction

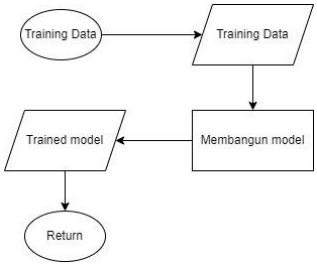
Feature extraction merupakan proses yang dilakukan pada saat *preprocessing data* dengan tujuan memecah data baru ke berbagai *feature* yang telah ada sehingga dapat dilakukan klasifikasi atau prediksi. Tahap *feature extraction* dilakukan dengan memanfaatkan berbagai *library* atau fungsi yang telah disediakan untuk mengambil informasi terkait *feature website phishing*. Beberapa *library* atau fungsi yang digunakan seperti *Regular Expression*, *HTTP Request*, *Web Scraping* (*Domain-based features*). Proses pada *Feature Extraction* dapat dilihat pada Gambar 2.



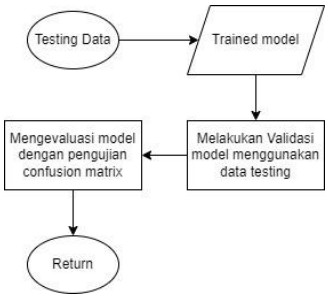
Gambar 2. Proses pada *Feature Extraction*

3.3 Training and Testing Data

Pada tahap ini diawali dengan *load dataset* yang akan digunakan untuk membangun model dalam proses klasifikasi. Data yang telah ada dibagi atau *split* menjadi dua bagian yaitu data *training* dan *testing* sesuai dengan parameter yang diberikan. Untuk data *training* akan dilatih dan diproses menggunakan ketiga algoritma yaitu C5.0, XGBoost, dan Random Forest. Setelah melakukan proses training, model dari data yang dilatih akan terbentuk. Lalu model yang telah ada akan dilakukan testing untuk prediksi dengan menggunakan data *testing*, serta melakukan pengujian menggunakan *confusion matrix* dalam mengukur ketepatan klasifikasi seperti *accuracy, precision, recall*, dan *f1-score*. Tahapan dalam *training* dan *testing* dapat dilihat pada Gambar 3 dan Gambar 4.



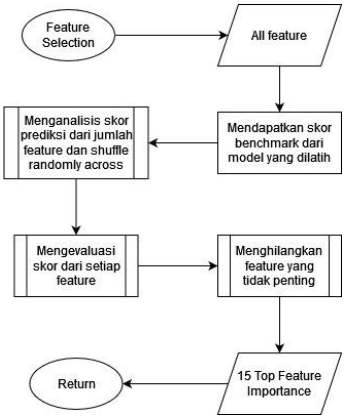
Gambar 3. Proses Training Data



Gambar 4. Proses Testing Data

3.4 Proses Feature Selection

Menggunakan pendekatan *feature importance* dengan memilih *feature* yang berpengaruh dalam membangun model untuk klasifikasi. *Feature importance* akan mengevaluasi setiap *feature* yang digunakan dengan cara melakukan percobaan misalnya menghilangkan salah satu *feature* jika skornya terganggu atau turun drastis maka *feature* tersebut sangat diperlukan dalam membangun model. 15 *feature* dengan skor tertinggi akan digunakan untuk proses klasifikasi. Proses pada *feature selection* dapat dilihat pada Gambar 5.



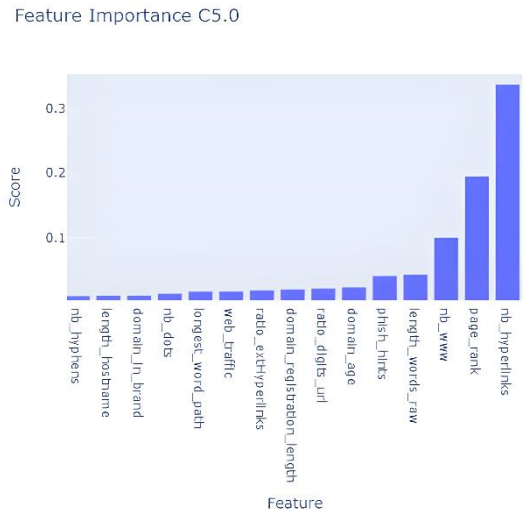
Gambar 5. Proses pada *Feature Selection*

4. PENGUJIAN SISTEM

4.1 Hasil Feature Importance C5.0

Proses dalam menentukan *feature importance* pada setiap algoritma akan menggunakan model masing-masing algoritma,

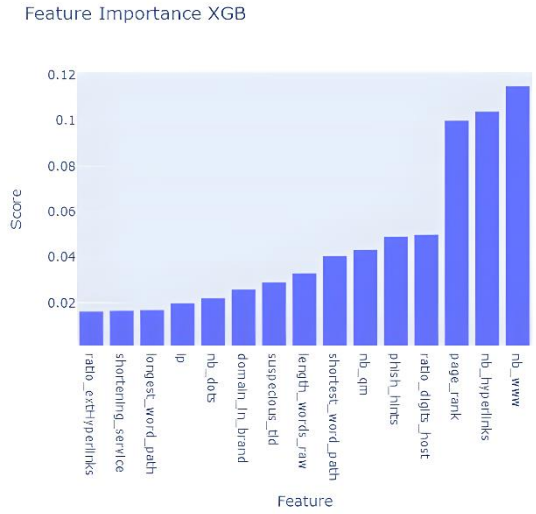
kecuali C5.0 yang akan memanfaatkan model dari Decision Tree dikarenakan menggunakan modul terpisah. Setiap *feature* akan memiliki *score* yang menunjukan seberapa penting *feature* tersebut terhadap model yang dibuat. Pada model ini sebanyak 15 dari 86 *feature* yang memiliki *score* tertinggi, visualisasi *feature importance* pada algoritma C5.0 yang terdiri dari 15 *feature* tertinggi dapat dilihat pada Gambar 6.



Gambar 6. Visualisasi *Feature Importance* pada Algoritma C5.0

4.2 Hasil Feature Importance XGBoost

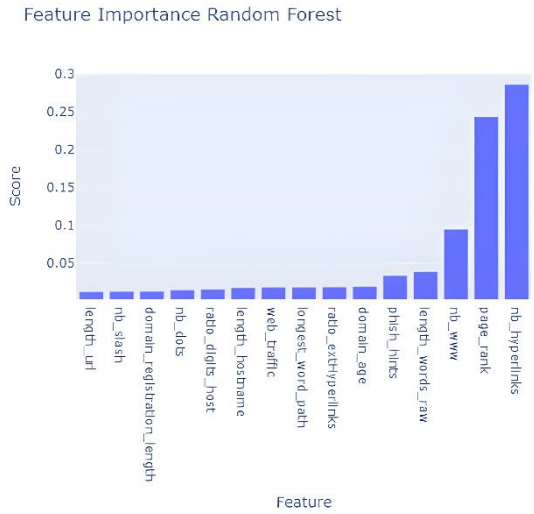
Berbeda dengan algoritma C5.0, *feature importance* pada XGBoost dapat langsung digunakan sama halnya dengan algoritma Random Forest. Pada model ini sebanyak 15 dari 86 *feature* yang memiliki *score* tertinggi, visualisasi *feature importance* pada algoritma XGBoost yang terdiri dari 15 *feature* tertinggi dapat dilihat pada Gambar 7.



Gambar 7. Hasil *Feature Importance* pada Algoritma XGBoost

4.3 Hasil Feature Importance Random Forest

Feature importance pada algoritma Random Forest memiliki *feature* terbanyak dibandingkan dengan C5.0 maupun XGBoost. Di mana pada model ini sebanyak 15 dari 86 *feature* yang memiliki *score* di tertinggi, visualisasi *feature importance* pada algoritma Random Forest yang *feature* sebanyak 15 *feature* dapat dilihat pada Gambar 8.



Gambar 8. Hasil *Feature Importance* pada Algoritma Random Forest

4.4 Hasil Confusion Matrix C5.0

Untuk evaluasi hasil pada algoritma C5.0 akan terdapat beberapa bagian yang akan dievaluasi, mulai dari penggunaan *boosting* dengan beberapa jumlah *trial*, hingga penggunaan *feature importance*. Hasil dari perhitungan *accuracy, precision, recall, f1-score, & training time* pada algoritma C5.0 akan ditampilkan pada Tabel 1.

Tabel 1. Hasil Confusion Matrix pada Algoritma C5.0

No	Boosting (Trial)	Feature Importance	Accuracy	Precision	Recall	F1-Score	Training Time
1	0	No	0,918	0,930	0,909	0,920	102,393s
2	0	Yes	0,926	0,931	0,924	0,927	27,873s
3	5	No	0,931	0,940	0,925	0,932	48,690s
4	5	Yes	0,934	0,950	0,921	0,935	10,356s
5	10	No	0,930	0,919	0,946	0,933	32,944s
6	10	Yes	0,933	0,922	0,950	0,936	7,921s
7	15	No	0,932	0,934	0,935	0,934	27,835s
8	15	Yes	0,935	0,934	0,940	0,937	6,002s
9	20	No	0,935	0,937	0,937	0,937	24,259s
10	20	Yes	0,930	0,922	0,944	0,933	5,162s
11	25	No	0,931	0,931	0,935	0,933	21,688s
12	25	Yes	0,934	0,937	0,935	0,936	4,725s

Dari hasil yang diperoleh, memperlihatkan bahwa dengan penggunaan *boosting* sangat berpengaruh terhadap kinerja dari algoritma C5.0. Di mana terjadi peningkatan performa dari segi *accuracy, precision, recall, f1-score, & training time* hampir pada setiap percobaan *trial*. Jika membandingkan percobaan 1 dengan percobaan 11 yang keduanya tidak menggunakan *feature*

importance bisa sangat terlihat peningkatan performa, yang paling menarik perhatian yaitu *training time* yang berkurang drastis hingga kurang lebih 4 sampai 5 kali lebih cepat dibandingkan percobaan 1.

Selain itu, penggunaan *feature importance* juga bisa terlihat ada peningkatan performa dari segi *training time* yang dengan rata-rata sebesar 1,38 kali atau 38% lebih cepat pada setiap percobaan dibandingkan tanpa menggunakan *feature importance*. Dari segi *accuracy, precision, recall, & f1-score* tanpa menggunakan *boosting* terjadi peningkatan, namun jika menggunakan *boosting* dengan beberapa percobaan yang dilakukan, bisa terlihat model dengan tanpa menggunakan *feature importance* sedikit lebih unggul.

4.5 Hasil Confusion Matrix XGBoost

Untuk evaluasi hasil pada algoritma XGBoost, akan mengevaluasi dua hasil *confusion matrix* mengenai performa dari XGBoost dengan menggunakan parameter yang sudah ditentukan. Hasil dari perhitungan *accuracy, precision, recall, f1-score, & training time* pada algoritma XGBoost akan ditampilkan pada Tabel 2.

Tabel 2. Hasil Confusion Matrix pada Algoritma XGBoost

No	Set Parameter	Feature Importance	Accuracy	Precision	Recall	F1-Score	Training Time
1	No	No	0,965	0,965	0,965	0,965	0,949s
2	Yes	No	0,966	0,966	0,968	0,967	0,901s
3	No	Yes	0,949	0,954	0,946	0,95	0,474s
4	Yes	Yes	0,953	0,96	0,948	0,954	0,464s

Berdasarkan hasil yang diperoleh pada algoritma XGBoost, pada percobaan dengan tanpa parameter menunjukan bahwa dengan *feature importance*, *training time* dapat dipersingkat sekitar 2,00 kali atau 100% lebih cepat dibandingkan dengan tanpa *feature importance*. Sedangkan pada percobaan dengan parameter menunjukan bahwa dengan *feature importance*, *training time* dapat dipersingkat sekitar 1,94 kali atau 94% lebih cepat dibandingkan dengan tanpa *feature importance.* Dari keempat percobaan, bisa terlihat bahwa dengan *feature importance* dapat mempercepat *training time*, walaupun ada sedikit pengurangan pada *accuracy, precision, recall, & f1-score* namun penurunan performa tersebut tidak signifikan atau tidak terlalu berdampak dalam proses klasifikasi.

4.6 Hasil Confusion Matrix Random Forest

Sama seperti algoritma XGBoost pada algoritma Random Forest juga akan mengevaluasi dua hasil dari *confusion matrix*, dengan parameter yang sudah ditentukan. Hasil perhitungan *accuracy, precision, recall, f1-score, & training time* pada Random Forest akan ditampilkan pada Tabel 3.

Tabel 3. Hasil Confusion Matrix pada Algoritma Random Forest

No	Set Parameter	Feature Importance	Accuracy	Precision	Recall	F1-Score	Training Time
1	No	No	0,96	0,961	0,961	0,961	1,284s
2	Yes	No	0,961	0,963	0,962	0,962	2,545s
3	No	Yes	0,957	0,961	0,954	0,958	0,899s
4	Yes	Yes	0,957	0,961	0,954	0,958	0,668s

Berdasarkan hasil yang diperoleh pada algoritma Random Forest, percobaan dengan tanpa parameter menunjukan bahwa dengan *feature importance*, *training time* dapat dipersingkat sekitar 1,42 kali atau 42% lebih cepat dibandingkan dengan tanpa *feature importance*. Pada percobaan dengan parameter, hasil menunjukan bahwa dengan *feature importance*, *training time* juga dapat dipersingkat sekitar 3,80 kali atau 280% lebih cepat dibandingkan dengan tanpa *feature importance*. Kedua percobaan dengan menggunakan dan tanpa menggunakan parameter menunjukan, bahwa pada percobaan *feature importance* nilai pada *accuracy, precision, recall, & f1-score* terjadi sedikit penurunan sama dengan yang terjadi pada algoritma XGBoost.

5. KESIMPULAN

Berdasarkan hasil yang telah didapatkan dari berbagai pengujian, maka bisa disimpulkan bahwa perumusan masalah dapat terjawab dengan baik. Berikut kumpulan kesimpulan dari setiap rumusan masalah:

- Berdasarkan hasil dari *training time* pada algoritma XGBoost menunjukan dengan menggunakan dan tanpa menggunakan *feature importance*, XGBoost merupakan algoritma tercepat dibandingkan Random Forest dan C5.0. Waktu terbaik yang dibutuhkan oleh algoritma XGBoost dalam melakukan *training* adalah 0,464 detik, sedangkan untuk algoritma Random Forest membutuhkan waktu 0,668 detik dan C5.0 membutuhkan waktu 4,725 detik. Lalu untuk hasil performa dari *accuracy*, *precision*, *recall, & f1-score* pada ketiga algoritma menunjukan bahwa dengan tanpa menggunakan *feature importance*, algoritma dengan performa *confusion matrix* terbaik dengan rata-rata sebesar 96,6% adalah algoritma XGBoost. Sedangkan hasil performa dari *accuracy*, *precision*, *recall, & f1-score* pada ketiga algoritma menunjukan bahwa dengan menggunakan *feature importance,* algoritma dengan performa *confusion matrix* terbaik dengan rata-rata sebesar 95,7% adalah algoritma Random Forest.
- *Feature* yang berpengaruh besar pada setiap algoritma berbeda-beda, untuk itu hanya 15 *feature* berpengaruh besar saja yang akan disebutkan.

  Pada algoritma C5.0, *feature* yang berpengaruh besar terdiri dari *nb_hyperlinks, page_rank, nb_www, length_words_raw, phish_hints, domain_age, ratio_digits_url, domain_registration_length, ratio_extHyperlinks, web_traffic, longest_word_path, nb_dots, domain_in_brand, length_hostname, nb_hyphens*.

  Pada algoritma XGBoost, *feature* yang berpengaruh besar terdiri dari *nb_www, nb_hyperlinks, page_rank, ratio_digits_host, phish_hints, nb_qm, shortest_word_path, length_words_raw, suspecious_tld, domain_in_brand, nb_dots, ip, longest_word_path, shortening_service, ratio_extHyperlinks*.

  Pada algoritma Random Forest, *feature* yang berpengaruh besar terdiri dari *nb_hyperlinks, page_rank, nb_www, length_words_raw, phish_hints, domain_age, ratio_extHyperlinks, longest_word_path, web_traffic, length_hostname, ratio_digits_host, nb_dots, domain_registration_length, nb_slash, length_url.*
- Pengaruh yang diberikan dengan penggunaan *feature selection* khususnya *feature importance* pada algoritma C5.0 sangat besar terutama pada segi *training time* yang berkurang drastis hingga 1,38 kali atau 38% lebih cepat dari biasanya. Pada algoritma XGBoost dan Random Forest sebenarnya rata-rata juga mempercepat *training time* sekitar 2,0 hingga 3,8 kali lebih cepat, namun dikarenakan *dataset* yang digunakan

memiliki ukuran yang terbatas maka pengaruh yang diberikan tidak terlihat sebesar pada C5.0, walaupun penggunaan *feature selection* pada algoritma XGBoost dan Random Forest sebenarnya memberikan rata-rata kecepatan yang lebih unggul. Nilai *accuracy*, *precision*, *recall, & f1-score* pada ketiga algoritma terjadi sedikit penurunan performa pada segi *accuracy*, *precision*, *recall, & f1-score*, hal ini wajar dikarenakan banyak *feature* yang dibuang dengan tujuan agar *training time* dapat dipersingkat.

6. SARAN

Berdasarkan hasil dan berbagai pengujian yang telah dilakukan, maka saran untuk penelitian selanjutnya yaitu:

- Dapat melakukan pengujian dengan menggunakan *dataset* dengan ukuran yang jauh lebih besar.
- Dapat menggunakan pengujian dengan metode *feature selection* yang lain, seperti metode *filter* dan metode *wrapper* untuk mengklasifikasikan *website phishing*.
- Dapat melakukan penelitian terpisah pada penggunaan *feature extraction*, dikarenakan masih terdapat *feature* pada *feature extraction* yang perlu untuk diteliti lebih jauh.

7. REFERENSI

[1] Aminu, A. A., Abdulrahman, A., Aliyu, A. Y., Aliyu, M., & Turaki, A. M. 2019. Detection of Phishing WebsitesUsing Random Forest and XGBOOST Algorithms. International Journal Of Pure And Applied Sciences, 2(3), 1-11.

[2] Baykara, M., & Gurel, Z. 2018. Detection of phishing attacks. 2018 6Th International Symposium On Digital Forensic And Security (ISDFS). DOI=10.1109/isdfs.2018.8355389.

[3] Berry, M. W., Mohamed, A., & Yap, B. W. (Eds.). 2015. Soft Computing in Data Science. Communications In Computer And Information Science, 257. DOI=10.1007/978-981-287-936-3.

[4] Chelvan, V. P. 2022. OCBC says S$13.7 million lost in phishing scams, up from S$8.5 million. CNA. URI=https://www.channelnewsasia.com/singapore/ocbc-phishing-scam-more-losses-victims-reported-2469086.

[5] Chen, C., Tsai, Y., Chang, F., & Lin, W. 2020. Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. Expert Systems, 37(5). DOI=10.1111/exsy.12553

[6] Dewi, D. A. W., Cholissodin, I., & Sutrisno. 2019. Klasifikasi Penyimpangan Tumbuh Kembang Anak Menggunakan Algoritma C5.0. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, 3(10), 10260-10261.

[7] Karo Karo. M., I. 2020. Implementasi Metode XGBoost dan Feature Importance untuk Klasifikasi pada Kebakaran Hutan dan Lahan. Journal Of Software Engineering, Information And Communication Technology, 1(1), 12-13.

[8] Khan, N., Madhav C, N., Negi, A., & Thaseen, I. 2019. Analysis on Improving the Performance of Machine Learning Models Using Feature Selection Technique. Advances In Intelligent Systems And Computing, 69-77. DOI=10.1007/978-3-030-16660-1_7.

[9] Kumar, J., Santhanavijayan, A., Janet, B., Rajendran, B., & Bindhumadhava, B. 2020. Phishing Website Classification and Detection Using Machine Learning. 2020 International Conference On Computer Communication And Informatics (ICCCI). DOI=10.1109/iccci48352.2020.9104161.

[10] Machado, L., & Gadge, J. 2017. Phishing Sites Detection Based on C4.5 Decision Tree Algorithm. 2017 International Conference On Computing, Communication, Control And Automation (ICCUBEA). DOI=10.1109/iccubea.2017.8463818.

[11] Masurkar, S., & Dalal, V. 2020. ENHANCED MODEL FOR DETECTION OF PHISHING URL USING MACHINE LEARNING. Ethics And Information Technology (ETIT), 2(2), 158-163. DOI=10.26480/etit.02.2020.158.163.

[12] Shah, K., Patel, H., Sanghvi, D., & Shah, M. 2020. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. Augmented Human Research, 5(1), 8. DOI=10.1007/s41133-020-00032-0.

[13] Zhang, L., & Zhan, C. 2017. Machine Learning in Rock Facies Classification: An Application of XGBoost. International Geophysical Conference, Qingdao, China, 17-20 April 2017. DOI=10.1190/igc2017-35.