ISSN: 2477-5126
e-ISSN:2548-9356

Deteksi Malware menggunakan Metode Stacking berbasis Ensemble

Fauzi Adi Rafrastara[1*)], Catur Supriyanto[2], Cinantya Paramita[3], Yani Parti Astuti[4]
Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro, Semarang
Jln. Imam Bonjol, No. 207, Kota Semarang, 50131, Indonesia
email: [1]fauziadi@dsn.dinus.ac.id, [2]catur.supriyanto@dsn.dinus.ac.id, [3]cinantya.paramita@dsn.dinus.ac.id, [4]yanipartiastuti@dsn.dinus.ac.id

***Abstract* – Malware attacks are getting more and more apprehensive. The rapid and increasingly destructive evolution of malware is a cause for concern for many. Therefore, effective malware detection is urgently needed. Data mining plays a crucial role in this field, since the algorithms can be trained to produce the very high accuracy. The aim of this research is nothing but to implement the stacking method on file classification, whether it is considered as malware or benign file. Stacking method is used since it can improve the classification accuracy score compared to the conventional classifiers. Four algorithms are involved in this experiment, namely: Neural Network, Random Forest, kNN and Logistic Regression. The first three algorithms are used as a classifier at level-0, while Logistic Regression is used as classifier at level-1 (meta classifier). With the combination of these 4 algorithms, the obtained accuracy is 98.7%. This score is the highest one, especially compared to each algorithm when executed individually.**

***Intisari* – Serangan malware kian hari kian memprihatinkan. Evolusi malware yang cepat dan semakin destruktif menimbulkan kekhawatiran bagi banyak pihak. Oleh karena itu, deteksi malware yang efektif sangat dibutuhkan. Data mining memainkan peran yang krusial dalam bidang ini, mengingat algoritma-algoritma yang ada pada data mining bisa dilatih hingga menghasilkan akurasi yang sangat tinggi. Tujuan dari penelitian ini adalah untuk menerapkan metode stacking pada kasus klasifikasi suatu file, apakah tergolong malware atau tidak. Metode stacking digunakan karena dapat meningkatkan performa khususnya jika dibandingkan dengan algoritma-algoritma klasifikasi konvensional. Empat buah algoritma klasifikasi dilibatkan dalam eksperimen yang dilakukan pada penelitian ini, yaitu: Neural Network, Random Forest, kNN, dan Logistic Regression. Tiga algoritma pertama digunakan sebagai *classifier* pada level 0, sementara itu Logistic Regression digunakan *classifier* pada level 1 (*meta classifier*). Dengan kombinasi 4 algoritma tersebut, akurasi yang diperoleh adalah sebesar 98.7%, dan akurasi tersebut merupakan yang paling tinggi jika dibandingkan dengan masing-masing algoritma jika dieksekusi secara individual.**

***Kata Kunci* – deteksi malware, ensemble, stacking, data mining, neural network, random forest, kNN, logistic regression**

I. PENDAHULUAN

Malware atau *malicious software* merupakan software yang secara sengaja melakukan hal-hal jahat di dalam perangkat korban, termasuk diantaranya adalah komputer, smart phones, tablet, dll) [1]. Ada banyak jenis *malware* yang menjadi ancaman serius bagi sebagian besar orang, diantaranya yaitu virus, worm, trojan horse, rootkit, hingga ransomware. Masing-masing jenis *malware* tersebut memiliki tujuan yang berbeda. Sebagai contoh, virus memiliki tujuan destruktif yaitu merusak system. Virus bisa berjalan secara otomatis dan akan bersembunyi untuk menghindar dari deteksi antivirus. Dengan demikian dia akan leluasa menjalankan misinya, mulai dari sekedar menunjukkan eksistensinya dengan tampilan-tampilan tertentu, mengganggu kinerja system, hingga merusak data atau sistem yang dituju secara brutal [2].

Di sisi lain, terdapat sebuah malware yang bisa menggandakan dirinya sendiri kemudian menyebar layaknya cacing yang merambat ke segala wilayah yang bisa dijangkau. Malware tersebut dikenal dengan istilah Worm [3]. Worm memiliki kaitan erat dengan jaringan komputer, karena salah satu tujuan utama dari worm adalah membanjiri jaringan komputer untuk kemudian mengambil alihnya, atau hanya sekedar melumpuhkannya. Namun demikian, dikotomi antara virus dan worm belakangan telah memudar, karena malware yang dideteksi sebagai virus saat ini juga memiliki kemampuan menggandakan dan menyebarkan diri layaknya worm.

Perkembangan *malware* dengan jenis lain juga menjadi sangat mengkhawatirkan. Ransomware telah memakan banyak korban. Data-data di computer korban disandra dengan cara dienkrip, kemudian muncul nominal dalam jumlah besar untuk menebus data-data tersebut. Sayangnya, meskipun uang tebusan telah dibayar, tidak semua data bisa dikembalikan. Menurut statistik, 57% korban ransomware rela membayar uang tebusan demi kembalinya data-data mereka. Sayangnya, kurang dari 28% korban yang berhasil mendapatkan kembali akses ke semua data-datanya [4]. Pada tahun 2017, salah satu ransomware yang sangat terkenal, yaitu Wannacry, telah menginfeksi lebih dari 300.000 korban di 150 negara. Para penyerang umumnya meminta tebusan dibayar menggunakan bitcoin mengingat sifat anonimnya, sehingga segala transaksi termasuk identitas pelaku menjadi sulit untuk dilacak [5].

Tingginya penetrasi malware berbanding lurus dengan tingkat destruktifnya. *Signature-based detection* dapat dengan mudah dikelabui berkat adanya kemampuan-kemampuan khusus pada malware yang berjenis polymorphic, metamorphic, hingga oligomorphic [2]. Oleh karena itu, dibutuhkan sistem cerdas yang dapat menganalisa dan mendeteksi suatu malware, baik berbasis *static analysis* maupun *dynamic analysis*.

*) **penulis korespondensi**: Fauzi Adi Rafrastara
Email: fauziadi@dsn.dinus.ac.id

Tujuan dari penelitian ini adalah untuk mengimplementasikan metode *stacking* pada kasus klasifikasi suatu file, apakah tergolong sebagai malware atau *benign file*. Metode *stacking* dipilih karena dapat meningkatkan performa algoritma data mining, khususnya jika dibandingkan dengan penggunaan *single algorithm* secara konvensional. Dataset yang digunakan merupakan dataset public dimana fitur-fiturnya diperoleh dari hasil *capture* perilaku *malware* yang terekam berdasarkan aktifitas API Call-nya (*dynamic analysis*). Selanjutnya, algoritma data mining dan metode *stacking* akan digunakan untuk mempelajari pola perilaku antara *benign file* dengan *malware* berdasarkan nilai pada atribut-atributnya.

II. PENELITIAN YANG TERKAIT

Topik penelitian tentang deteksi malware terus berkembang hingga saat ini. Penelitian tentang deteksi malware berbasis Windows, Android dan deteksi Ransomware seperti menjadi primadona di bidang informasion security. Paper yang ditulis oleh [6] dan [7] membahas tentang deteksi android *malware* menggunakan machine learning. Metode *Stacking* digunakan oleh [6] karena dalam penelitian mereka, skor yang diperoleh *stacking* lebih baik daripada 2 metode berbasis ensemble lain, yaitu *Bagging* dan *Boosting*. Dalam penelitiannya, [6] mengklaim bahwa model yang mereka usulkan berhasil mendeteksi 96.56% malware (dengan 227 malware yang gagal dideteksi). Sementara itu, para peneliti pada [7] membandingkan kemampuan beberapa algoritma klasifikasi untuk mendeteksi malware. Algoritma yang digunakan meliputi Logistic Regression, kNN, Support Vector Classifier, Decision Tree dan MLP. Algoritma kNN berhasil mengalahkan 4 algoritma lain dengan nilai recall sebesar 85%.

Pada sisi lain, deteksi malware berbasis Windows juga dilakukan oleh peneliti lain, seperti pada [8] yang fokus pada deteksi malware berbasis Windows menggunakan metode ensemble. Pada penelitian ini, 5 classifiers digunakan untuk mendeteksi suatu file apakah tergolong sebagai malware atau tidak. Selanjutnya, metode voting diterapkan untuk mencari nilai terbanyak, sehingga menjadi hasil akhir dari metode ensemble tersebut. Dengan metode ini, diperoleh akurasi sebesar 98%.

Berdasarkan penelitian-penelitian tersebut, dibutuhkan penelitian lanjutan untuk meningkatkan performa deteksi suatu malware menggunakan machine learning. Eksperimen dengan menggunakan metode ensemble terbukti menghasilkan performa yang lebih tinggi dibandingkan dengan menggunakan metode klasifikasi dasar. Dengan demikian, penelitian ini akan menggunakan metode ensemble untuk mencapai nilai evaluasi yang optimal.

III. METODE PENELITIAN

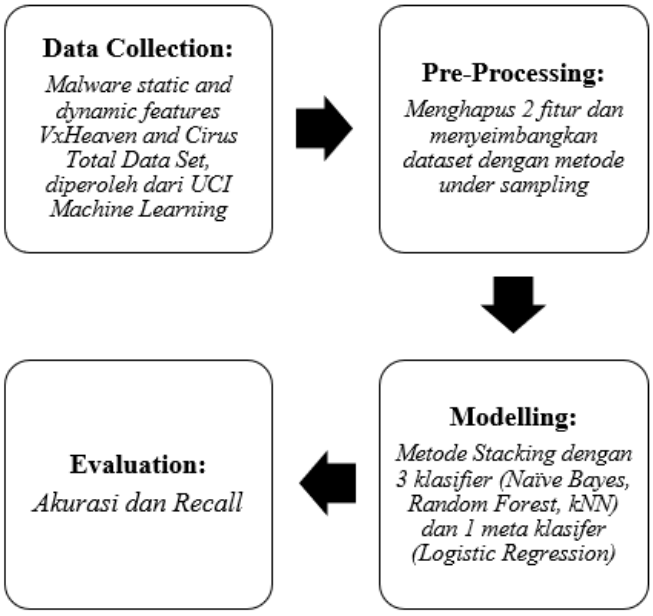
A. Hardware dan Software

Guna mendukung keberhasilan eksperimen dalam penelitian ini, dukungan software dan hardware yang mumpuni sangat penting untuk digunakan. Untuk software, Microsoft Excel digunakan untuk melakukan pre-processing, mulai dari menggabungkan dataset, mengurangi fitur, mengurangi *records*, hingga mengacak data. Selanjutnya, untuk pengaplikasian model, peneliti menggunakan software bernama Orange sebagai data mining tools-nya (https://orangedatamining.com/). Orange memiliki fitur-fitur yang lengkap untuk membangun model, baik model-model *basic* maupun berbasis ensemble, termasuk *stacking*. Dengan menggunakan Orange, nilai-nilai hasil evaluasi, seperti *Classification Accuracy*, *Recall*, *Precision, F1-Score* dapat diperoleh dengan cepat dan tepat.

Sedangkan terkait hardware, penelitian ini menggunakan sebuah computer dengan spesifikasi sebagai berikut:

Processor: Intel Xeon E5620
RAM: 16 GB
HD: 3 TB
VGA: Radeon RX550

Adapun tahapan-tahanpan yang dilakukan pada penelitian ini dapat dilihat pada Gambar 1.



Gbr. 1 Tahapan penelitian.

B. Data Collection

Dataset yang digunakan pada penelitian ini merupakan dataset public yang diperoleh dari UCI machine learning repository. Nama datasetnya yaitu 'Malware static and dynamic features VxHeaven and Virus Total Data Set'. Dataset ini diperoleh dari 3 sumber, yaitu dari VxHeaven (sebanyak 2698 *records* berlabel *malware*), Virus Total (2955 *records* berlabel *malware*), serta *benign file* dari Windows 7 & 8 (595 *records* berlabel *benign*).

C. Pre-Processing

Tiga jenis dataset yang didownload dari UCI tersebut memiliki jumlat *record* yang berbeda-beda. Apabila dataset *benign file* dengan *malware* dari Virus Total dan VxHeaven disatukan, maka akan menghasilkan *imbalanced dataset*. Untuk mengatasinya, peneliti hanya menggunakan dataset *malware* yang berasal dari Virus Total dan *benign* file. Pada kondisi ini, *imbalanced dataset* masih terjadi, mengingat jumlah record dengan label 1 (sebagai virus) masih sangat tinggi, bahkan hampir 5x lipat dari data dengan label 0 (kategori *benign*). Oleh karena itu, peneliti memotong jumlah data dari dataset Virus Total secara random dengan menggunakan metode *under-sampling* sehingga menghasilkan

dataset yang seimbang, yaitu 595 *record* berlabel 1 (*malware*) dan 595 *record* berlabel 0 (*benign*).

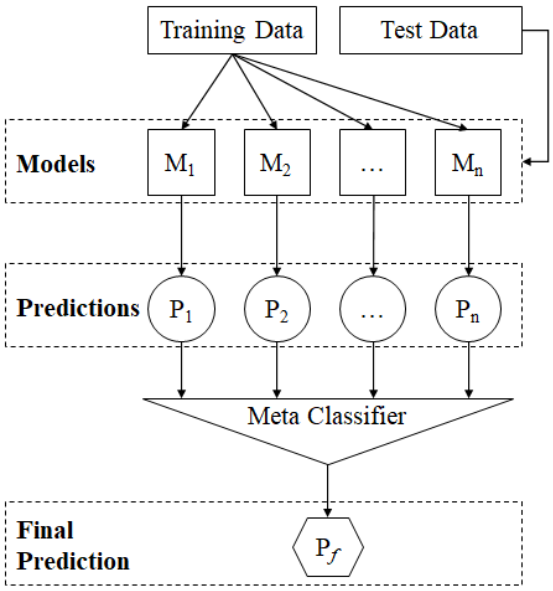
Dari sisi fitur, terdapat sedikit perbedaan dimana fitur pada dataset *benign* memiliki 1086 buah fitur, sedangkan pada dataset dari Virus Total memiliki 1089 fitur. Agar bisa dieksekusi dan menghasilkan akurasi yang baik, maka peneliti menghapus 2 fitur pada dataset dari Virus Total, yaitu fitur *__vbaVarIndexLoad* dan *SafeArrayPtrOfIndex*. Dengan demikian, kedua dataset tersebut memiliki jumlah dan nama-nama fitur yang sama persis (1086 fitur) sehingga bisa diproses lebih lanjut.

Fitur-fitur tersebut diperoleh dari hasil meng-*capture* Windows API Calls ketika file *benign* maupun *malware* dieksekusi pada sandbox yang sudah dipersiapkan. Deteksi ataupun klasifikasi *malware* dapat dilakukan dengan cara menganalisis pola API Calls pada suatu file ketika dijalankan [5].

D. Modelling

Sebelum masuk ke tahap modelling, data akan di-*sampling* terlebih dahulu menggunakan metode *cross validation*, yaitu *5-fold cross-validation*. Hal ini diperlukan untuk mengatasi kelemahan dari metode *Stacking*, yaitu *overfitting* [9]. Pada tahap Modelling, metode stacking berbasis ensemble digunakan untuk meningkatkan performa akurasi algoritma data mining.

Stacking merupakan teknik pembelajaran berbasis *ensemble* yang melibatkan beragam algoritma data mining untuk membentuk sebuah model baru. Model baru tersebut nantinya akan digunakan untuk membuat prediksi pada data uji yang telah disediakan. Pada tahap awal dalam metode *stacking*, data latih *(training data)* yang telah disampling akan diproses menggunakan beragam model atau algoritma sekaligus. Ada empat model klasifikasi yang dilibatkan pada eksperimen kali ini, yaitu Neural Network, Random Forest, kNN, dan Logistic Regression. Ke-empat model tersebut bertindak sebagai *base-classifier* pada level 0 di metode *stacking* dalam penelitian ini.



Gbr. 2 Workflow metode *stacking se*cara umum

Hasil prediksi dari masing-masing model (P_1, P_2, hingga P_n) akan dijadikan input untuk *meta classifier*, yaitu *classifier* pada level 1. Selanjutnya algoritma atau model yang dipilih untuk *meta classifier* akan melakukan *training* sekaligus *testing* dari data input yang diperoleh dari hasil prediksi beberapa model sebelumnya. Pada penelitian ini, algoritma Logistic Regression dipilih sebagai *meta classifier.* Logistic Regression merupakan algoritma yang sangat bisa diandalkan sebagai *meta classifier*. Hal ini dibuktikan dengan banyaknya penelitian yang menggunakan Logistic Regression sebagai *meta classifier* dan diterapkan pada area yang beragam, seperti pada: [6], [10], [11], [12], [13], [14]. Adapun *workflow* metode *stacking* secara umum dapat dilihat pada gambar 2.

E. Evaluasi

Evaluasi merupakan tahap akhir dari eksperimen yang dilakukan pada penelitian ini, yaitu dengan menghitung nilai akurasi dan juga recall. Akurasi dibutuhkan untuk mendapatkan gambaran dengan jelas tentang seberapa sering suatu model memprediksi/mengklasifikasi dengan tepat. Nilai akurasi diperoleh dari hasil pembagian antara jumlah True Positive dan True Negative dengan jumlah antara True Positive, False Positive, True Negative, dan False Negative *(formula 1)*.

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

Sementara itu, Recall merupakan rasio model memprediksi *malware* dengan tepat terhadap total keseluruhan sampel *malware*. Singkatnya, Recall digunakan untuk mengetahui seberapa sering model mengidentifikasi *malware* dengan tepat. Dalam hal deteksi *malware*, Recall *(formula 2)* memiliki peran yang sangat penting mengingat false negative dalam kasus ini dapat memberi dampak yang fatal [7]. Suatu file virus akan sangat berbahaya jika dideteksi sebagai bukan virus, namun tidak terlalu berbahaya untuk sebaliknya. Nilai Recall diperoleh dari hasil pembagian antara TP dengan jumlah TP dan FN.

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Berdasarkan kebutuhan pada penelitian ini, maka jenis metrik yang digunakan adalah Akurasi dan Recall. Kedua metrik tersebut akan dimanfaatkan untuk mengukur performa masing-masing algoritma secara mandiri, juga mengukur beberapa performa algoritma stacking untuk mencari yang terbaik.

IV. HASIL DAN PEMBAHASAN

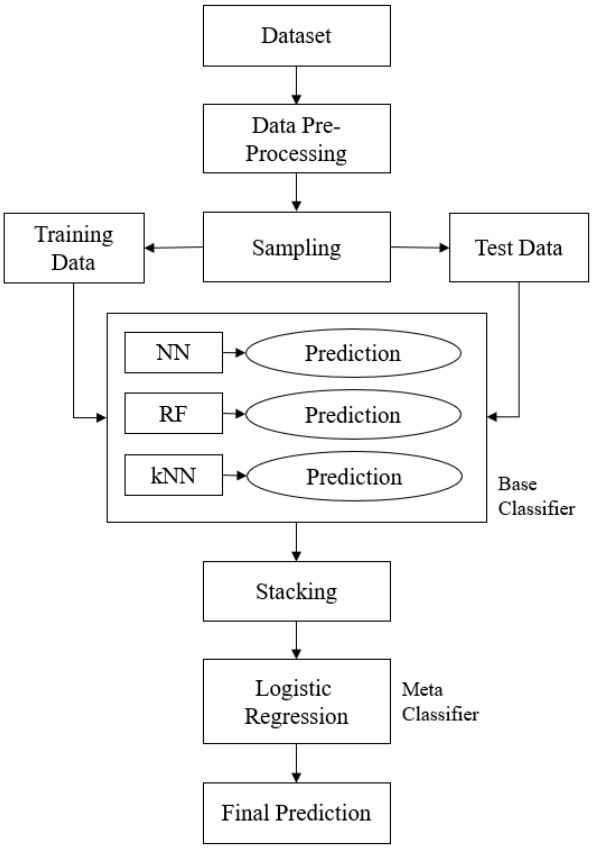
Pada penelitian ini, empat metode klasifikasi data mining digunakan, baik secara individu maupun secara bersamaan dalam bentuk metode *stacking*. Empat metode tersebut yaitu: Neural Network, Random Forest, kNN, dan Logistic Regression. Ketika masing-masing algoritma tersebut dieksekusi pada dataset yang telah disiapkan, hasil akurasi dan recall yang diperoleh dapat dilihat pada Tabel 1.

Ke-empat algoritma tersebut memiliki skor akurasi dan recall yang tergolong tinggi, yaitu di atas 90%. Nilai tertinggi dimiliki oleh Random Forest, dengan skor 98.2%, baik untuk akurasi maupun recall.

TABEL I
PERBANDINGAN 4 ALGORITMA YANG DIEKSEKUSI SECARA INDIVIDU

Algoritma	Akurasi	Recall
Neural Network	0.967	0.967
Random Forest	0.982	0.982
kNN	0.955	0.955
Logistic Regreession	0.922	0.922

Mengingat pada kasus ini objek yang digunakan adalah *malware*, dimana false negative memiliki resiko yang besar, maka angka recall 98.2% masih perlu ditingkatkan. Dengan angka tersebut, maknanya masih ada sekitar 1.8% peluang suatu virus lolos dari deteksi system, dan tentunya hal itu sangat berbahaya. Recall yang dimiliki oleh system pendeteksi virus, seharusnya mencapai 100% untuk mencapai *zero tolerance* false negative. Oleh karena itu, perlu upaya lebih agar angka 98.2% yang dimiliki oleh Random Forest bisa ditingkatkan lagi. Salah satu cara untuk mengoptimasi algoritma klasifikasi adalah dengan teknik ensemble, yang satu diantaranya bernama *Stacking*. *Framework* metode *stacking* yang digunakan pada penelitian ini dapat dilihat pada gambar 3.



Gbr. 3 Framework metode *stacking* yang digunakan.

Ensemble merupakan metode yang menggunakan beberapa algoritma *machine learning* atau model untuk menghasilkan model prediksi yang optimal. Umumnya hasil dari metode ensemble ini memiliki performa yang lebih bagus daripada *base learner* atau algoritma dasar *machine learning* apabila dieksekusi secara individu.

Dibandingkan dengan metode jenis *ensemble* yang lain seperti *Bagging* dan *Boosting*, *Stacking* memiliki performa yang lebih baik. Hal ini terjadi karena pada teknik *Stacking* terdapat *meta learner*/*classifier* yang mampu menentukan *classifier* mana yang cenderung berhasil pada fitur tertentu, kemudian *classifier* mana yang bagus pada fitur yang lain, kemudian menggabungkannya dengan tepat [6].

Di dalam penelitian ini, yang bertindak sebagai *base classifier* pada teknik *stacking* adalah algoritma Neural Network (NN), Random Forest (RF) dan k-Nearest Neighbor (kNN). Masing-masing algoritma tersebut akan menghasilkan prediksinya masing-masing sebelum akhirnya diproses lebih lanjut oleh *meta classifier* yang dalam hal ini memanfaatkan algoritma *Logistic Regression*.

Dengan metode *stacking*, nilai akurasi dan *recall* untuk deteksi *malware* berhasil dinaikkan, meskipun tidak terlalu signifikan mengingat nilai yang dihasilkan oleh *base classifier* sudah sangat tinggi. Hasil perbandingan dari nilai akurasi dan recall pada metode stacking dengan 4 algoritma lain dapat dilihat pada tabel II.

Akurasi dan Recall pada tahap evaluasi ini memiliki peranan yang berbeda. Akurasi digunakan untuk mengetahui dengan jelas tentang seberapa sering suatu model memprediksi/mengklasifikasi dengan tepat. Sementara itu, nilai Recall pada penelitian ini juga menjadi krusial, mengingat perannya dalam memberikan gambaran tentang seberapa sering model mengidentifikasi *malware* dengan tepat. Deteksi *malware* perlu terus ditingkatkan dari sisi recall mengingat tingkat fatalitasnya yang sangat tinggi apabila ada kasus *misclassification*, khususnya file *malware* dideteksi sebagai bukan *malware*.

Pada eksperimen yang telah dilakukan, peningkatan yang berhasil diperoleh adalah sebesar 0.005, baik untuk skor akurasi maupun *recall*. Meskipun terlihat kecil, namun angka ini semakin mendekatkan nilai *recall* ke arah 100% yang notabene menjadi target untuk kasus deteksi *malware*. Sekecil apapun celah atau peluang yang dapat membuat virus lolos dari pantauan harus ditutup, sehingga dibutuhkan recall dengan nilai 100%.

TABEL III
PERBANDINGAN 4 ALGORITMA YANG DIEKSEKUSI SECARA INDIVIDU

Algoritma	Akurasi	Recall
Stacking	0.987	0.987
Random Forest	0.982	0.982
Neural Network	0.967	0.967
kNN	0.955	0.955
Logistic Regreession	0.922	0.922

Eksperimen lain juga dilakukan dengan mengganti *meta classifier* dengan beragam algoritma, yaitu Random Forest, Neural Network dan juga kNN. Hasil yang diperoleh terlihat pada tabel III berikut ini.

TABEL IIIII
PERBANDINGAN ALGORITMA STACKING DENGAN BERAGAM META CLASSIFIER

Base Classifier	Meta Classifier	Akurasi	Recall
RF, NN, kNN	LR	0.987	0.987
kNN, LR, RF	NN	0.986	0.986
NN, kNN, LR	RF	0.984	0.984
LR, RF, NN	kNN	0.982	0.982

Dari eksperimen di atas, diperoleh nilai akurasi dan recall yang ternyata tidak lebih baik dari Logistic Regression sebagai *meta classifier*-nya. Logistic Regression merupakan algoritma yang cukup populer untuk dijadikan sebagai *meta classifier* karena kehandalannya. Namun demikian, berdasarkan skor yang ada pada tabel III, dapat diketahui bahwa skor akurasi dan recall untuk metode *stacking* cenderung lebih baik dari algoritma-algoritma klasifikasi dasar yang dieksekusi secara individu, seperti Logistic Regression, kNN dan Neural Network. Random Forest memiliki nilai yang cukup tinggi juga (meskipun masih kalah dengan *stacking*), karena Random Forest juga merupakan algoritma yang mengusung salah satu konsep *ensemble*, yaitu Bagging (*Bootstrap Aggregating*).

Pada penelitian yang dilakukan oleh [7], skor *recall* tertingginya hanya berada dalam angka 85% pada kasus klasifikasi *malware* pada system operasi Android, yaitu dengan menggunakan algoritma kNN. Pada eksperimen yang lain [8], Neural Network memiliki skor yang paling tinggi dalam mengklasifikasi ransomware, yaitu 98%, mengalahkan 4 algoritma lain, seperti Support Vector Machine (SVM), k-Nearest Neighbor (kNN), XGboost, dan Random Forest (RF). Ketika dua metode terbaik tersebut diterapkan pada dataset yang digunakan di penelitian ini, skor akurasi dan *recall* yang diperoleh adalah sebesar 95.5% (kNN) dan 96.7% (NN). Berdasarkan data tersebut, metode stacking yang digunakan pada eksperimen ini dapat menghasilkan skor akurasi dan juga recall yang sama-sama lebih baik, yaitu 98.7%. Adapun metode stacking tersebut, terdiri dari Random Forest, Neural Network, dan kNN sebagai *base classifiernya*, serta Logistic Regression sebagai *meta classifier*.

V. KESIMPULAN

Perkembangan *malware* yang massive dengan dampak yang semakin mengkhawatirkan memaksa para peneliti untuk turut berkontribusi, yang salah satunya yaitu dengan menghasilkan model-model yang efektif untuk mendeteksi suatu file apakah tergolong *malware* atau tidak. Dalam penelitian ini, dataset publik digunakan dengan terlebih dahulu melakukan *pre-processing* agar data tersebut siap untuk digunakan. Selanjutnya, 4 algoritma klasifikasi (Neural Network, Random Forest, k-Nearest Neighbor dan Logistic Regression) dieksekusi secara individu untuk melihat hasil akurasi dan recallnya, sebelum dibandingkan dengan nilai yang didapat pada metode stacking, khususnya dengan Logistic Regression sebagai *meta classifier*-nya. Hasilnya, metode stacking dengan meta classifier Logistic Regression berhasil mengungguli 4 algoritma yang dieksekusi secara individu. Terakhir, stacking dengan logistic regression juga dibandingkan dengan metode *stacking* lain, namun dengan *meta classifier* yang berbeda-beda, yaitu Random Forest, kNN dan Neural Network. Hasilnya, metode *stacking* dengan *meta classifier* Logistic Regression masih menjadi yang terbaik, dengan tingkat akurasi serta recall sebesar 98.7%.

Meskipun nilai akurasi dan recall sudah mencapai 98.7%, namun angka tersebut dirasa masih perlu untuk dinaikkan, mengingat false negative yang ada pada kasus *malware* ini memiliki dampak yang sangat signifikan. Oleh karena itu, nilai 98.7% bisa dimaknai bahwa masih ada celah bagi virus untuk lolos dan merusak sistem target. Dengan angka 98.7%, artinya ada sekitar 7 virus yang gagal dideteksi oleh mesin, dan ini merupakan angka yang sangat besar apabila melihat dampak yang bisa dihasilkan. Nilai 100% pada bagian recall menjadi tujuan penelitian selanjutnya. Salah satu caranya adalah dengan menambah satu tahapan pada pre-processing, yaitu feature selection.

DAFTAR PUSTAKA

[1] O. Aslan and R. Samet, "A Comprehensive Review on Malware Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[2] F. A. Rafrastara and F. M. A, "Advanced Virus Monitoring and Analysis System," 2011. [Online]. Available: http://sites.google.com/site/ijcsis/

[3] A. Pratama and F. A. Rafrastara, "Computer Worm Classification," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 10, no. 4, pp. 21–24, 2012.

[4] M. Robles-Carrillo and P. García-Teodoro, "Ransomware: An Interdisciplinary Technical and Legal Approach," *Security and Communication Networks*, vol. 2022, 2022, doi: 10.1155/2022/2806605.

[5] W. Z. A. Zakaria, M. F. Abdollah, O. Mohd, M. S. M. M. Yassin, and A. Ariffin, "RENTAKA: A Novel Machine Learning Framework for Crypto-Ransomeware Pre-encryption Detection," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022, [Online]. Available: www.ijacsa.thesai.org

[6] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic android malware detection system with ensemble learning," *IEEE Access*, vol. 6, pp. 30996–31011, 2018, doi: 10.1109/ACCESS.2018.2844349.

[7] S. Shakya and M. Dave, "Analysis, Detection, and Classification of Android Malware using System Calls," *arXiv preprint arXiv:2208.06130*, 2022.

[8] S. Aurangzeb, H. Anwar, M. A. Naeem, and M. Aleem, "BigRC-EML: big-data based ransomware classification using ensemble machine learning," *Cluster Comput*, vol. 25, no. 5, pp. 3405–3422, 2022, doi: 10.1007/s10586-022-03569-4.

[9] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, and S. Gordon, "A tree-based stacking ensemble technique with feature selection for network intrusion detection," *Applied Intelligence*, vol. 52, no. 9, pp. 9768–9781, 2022, doi: 10.1007/s10489-021-02968-1.

[10] A. A. ABRO, E. TAŞCI, and A. UGUR, "A Stacking-based Ensemble Learning Method for Outlier Detection," *Balkan Journal of Electrical and Computer Engineering*, vol. 8, no. 2, pp. 181–185, 2020, doi: 10.17694/bajece.679662.

[11] J. Yi, H. Zhang, H. Liu, G. Zhong, and G. Li, "Flight Delay Classification Prediction Based on Stacking Algorithm," *Journal of Advanced Transportation*, vol. 2021, 2021, doi: 10.1155/2021/4292778.

[12] S. Touileb, "LTG-ST at NADI Shared Task 1: Arabic Dialect Identification using a Stacking Classifier," *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pp. 313–319, 2020.

[13] H. Byeon, "Exploring Factors Associated with the Social Discrimination Experience of Children from Multicultural Families in South Korea by using Stacking with Non-linear Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 125–130, 2021, doi: 10.14569/IJACSA.2021.0120516.

[14] Y. Wang, Q. Pan, X. Liu, and Y. Ding, "ET-MSF: a model stacking framework to identify electron transport proteins," *Frontiers in Bioscience - Landmark*, vol. 27, no. 1, pp. 1–9, 2022, doi: 10.31083/j.fbl2701012.