ANALISA PERBANDINGAN KLASIFIER DECISION TREE, RANDOM FOREST, DAN ADABOOST DALAM MENDETEKSI SERANGAN

COMPARATIVE ANALYSIS DECISION TREE, RANDOM FOREST, AND ADABOOST CLASSIFIER ON DETECTING ATTACK

Khalidian Gustami Fiqri[1], Ahmad Tri Hanuranto[2], Casi Setianingsih[3]
[1,2] Prodi S1 Teknik Telekomunikasi, [3]Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Universitas Telkom
[1]**khalidianfiqri@telkomuniversity.ac.id**, [2]**athanuranto@gmail.com**,
[3]**setiacasie@telkomuniversity.ac.id**

Abstrak

Ancaman siber yang banyak muncul dengan bertambahnya pengguna internet membuat keamanan siber menjadi hal yang penting untuk dimiliki oleh penyedia layanan. Karena ancaman siber tidak hanya merusak sistem penyedia layanan, tapi juga dapat mengambil data yang dimiliki oleh penyedia layanan tersebut. Bila hal ini terjadi dapat merugikan pihak penyedia layanan itu sendiri dan juga pengguna layanan tersebut. Dengan *Intrusion Detection System* yang dapat mendeteksi serangan siber secara otomatis dapat membantu dalam mengurangi serangan yang dapat masuk kedalam sistem.

Didalam Tugas Akhir ini, didesain pendeteksi serangan menggunakan klasifier *Decision Tree, Random Forest,* dan *AdaBoost* dan dianalisa klasifier manakah yang paling efisien dalam hal waktu dan performa dari ketiga klasifier yang digunakan. Perbandingan klasifier ini dilakukan dengan cara mendapatkan dataset, *preprocessing* dataset, pemilihan fitur yang digunakan, *training* klasifier, *testing* klasifier, lalu yang terakhir mengevaluasi hasil klasifier. Dataset yang digunakan adalah dataset KDDcup99 dan dataset manual. Dan fitur yang digunakan berjumlah 14 dari total 41 fitur dalam KDDcup99.

Hasil yang didapatkan adalah klasifier *Decision Tree* menjadi klasifier yang paling efisien dalam hal waktu dan performa. Dengan hasil: lama melatih klasifier 9,35 detik, memprediksi serangan 1,42 detik, *Precision* 96,41%, *Recall* 100%, dan *Accuracy* 97,05%. *Random Forest* merupakan klasifier kedua yang efisien untuk mendeteksi serangan karena dibandingkan *Decision Tree, Random Forest* memiliki hasil yang fluktuatif pada performanya. *AdaBoost* kurang efisien untuk mendeteksi serangan dikarenakan waktu yang dibutuhkan untuk melatih klasifier (178.64 detik) dan memprediksi serangan (21.56 detik) terlalu lama.

Kata kunci : *Decision Tree, Random Forest, AdaBoost, Intrusion Detection System, Classifier, Confusion Matrix, KDDcup99*

Abstract

With increasing usage of internet, the cyber threat will also increasing which makes cyber security become something that must have for every service provider. Because cyber threat not only can damage service provider's system but also can steal user's personal data that use service from service provider. If this happen not only loss on service provider but also on users. That's where Intrusion Detection System comes in. IDS can detect cyber attack automatically and can help reduce attack that comes to system.

In this Final Assignment was designed Decision Tree, Random Forest, and AdaBoost classfier to detecting attack and would be analyzed which more efficient based on time and performance from those three classifiers. This comparative classifier was done by getting datasets, preprocessing datasets, features selection, training classifiers, testing classifiers, and evaluating classifiers result. Datasets used were KDDcup99 dataset and manual dataset. From 41 features in KDDcup99, chosen 14 features to be used in this Final Assignment.

The results are Decision Tree classifier is the most efficient classifier based on time and performance with the outcome in training time 9.35 second and predict time 1.42 second. The performance from classifier are Precision 96.41%, Recall 100%, and Accuracy 97.05%. Random Forest is the second most effient because compared with Decision Tree, Random

Forest performance is fluctuative. On the other hand, AdaBoost is not very efficient to detecting attack because time needed for AdaBoost to train classifier (178.64 second) and predict attack (21.56 second) are too long.

Keywords: *Decision Tree, Random Forest, AdaBoost, Intrusion Detection System, Classifier, Confusion Matrix, KDDcup99*

1. Pendahuluan

Seiring dengan bertambahanya pengguna internet, ancaman siber yang dapat terjadi didalamnya pun ikut bertambah. Ancaman siber ini dapat menyerang penyedia layanan seperti *website, email,* dan *cloud* dengan cara meretas sistem tersebut ataupun mengambil data pengguna layanan tersebut. Pada tahun 2019 ini, telah terjadi serangan siber sebanyak 129 juta kali dari bulan Januari sampai dengan bulan September[3].

Karenanya pendeteksian serangan siber ini menjadi hal yang penting untuk dimiliki penyedia layanan karena tidak hanya mengamankan sistem dari penyedia layanan itu saja, tetapi juga mengamankan data pengguna dari layanan tersebut. Pendeteksian ini akan sulit dilakukan secara manual oleh manusia, karenanya pendeteksian ini dapat dilakukan oleh Intrusion Detection System (IDS) dimana akan dilakukan pendeteksian secara otomatis[1].

Telah dilakukan riset pada IDS menggunakan klasifier seperti: Decision Tree[2], Pattern Matching[5], dan Naïve Bayes[6]. Penelitian ini dilakukan oleh: Vibha Gupta, yang membandingkan klasifier-klasifier jenis Pattern Matching dan membandingkannya untuk mendapatkan yang paling efisien[5]. Ketan Sanjay Desale, melakukan penelitian yang membandingkan klasifier Naïve Bayes, Hoeffding Tree, Accuracy Updated Ensemble, dan Accuracy Weighted Ensemble. Didapatkan Naïve Bayes dan Hoesffding Tree memiliki akurasi dan kecepatan yang lebih tinggi ketimbang Accuracy Updated Ensemble, dan Accuracy Weighted Ensemble[6].

Didalam Tugas Akhir ini akan didesain pendeteksi intrusi menggunakan klasifier Decision Tree, Random Forest, dan AdaBoost dan membandingkan klasifier manakah yang paling efisien berdasarkan waktu dan performanya.

2. Konsep Dasar

2.1 Intrusion Detection System

Intrusion Detection System (IDS) adalah sebuah system pertahanan pada jaringan komputer untuk melawan beragam serangan. IDS digunakan untuk mendeteksi serangan dan memberikan respon kepada administrator. Selain memberi respon kepada administrator, IDS dapat memberikan aksi untuk mengatasi serangan yang masuk tersebut. IDS berdasarkan cakupannya dapat dibagi menjadi dua, yaitu: *Network Intrusion Detection System* (NIDS), dan *Host-base Intrusion Detection System* (HIDS)[4]. IDS memiliki beberapa teknik pendeteksian, yaitu: *Misuse/Signature based Detection*, dan *Anomaly based Detection*. *Signature Detection* merupakan pendeteksian pada serangan yang polanya sudah diketahui. *Anomaly Detection* adalah pendeteksian pada serangan yang melihat keanehan pada tingkah laku komputer atau data yang ada pada jaringan.

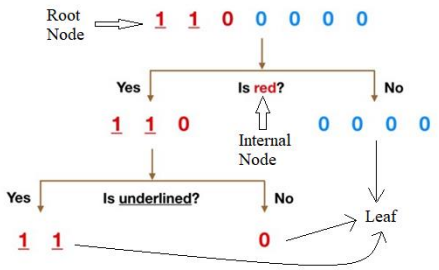
2.2 Algoritma

Algoritma adalah prosedur atau langkah komputasi yang diambil untuk mengubah input menjadi output[8]. Istilah algoritma itu sendiri diambil dari ilmuan matematika Muhammad ibn Musa Al-Khwarizmi. Algoritma didalam bidang Teknologi Informasi sudah digunakan secara luas. Misalnya Mesin Pencari yang mengambil beberapa karakter dari keyword dari input untuk mencari hasil yang relevan dari database dan menampilkannya.

Algoritma ini juga dapat digunakan sebagai classifier yang dapat mengklasifikasi data sesuai dengan atribut yang dimiliki data tersebut.

2.3 Decision Tree

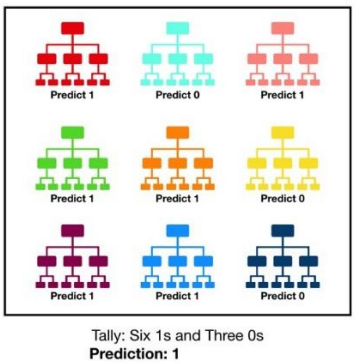
Decision Tree merupakan algoritma yang mengambil keputusan berdasarkan aturan yang ditetapkan. Aturan-aturan ini dapat digambarkan sepeti pohon yang daunnya merupakan aturan dan cabangnya merupakan keputusan yang diambil oleh algoritma ini[9].



***Gambar 2-1** Ilustrasi* Decision Tree

2.4 Random Forest

Random Forest adalah algoritma yang dikembangkan berdasarkan algoritma *Decision Tree*. Algoritma *Random Forest* ini dibentuk dengan membuat lebih dari satu algoritma *Decision Tree* yang dapat meminimalisir kesalahan pengambilan keputusan yang terjadi pada satu *Decision Tree*[10]. Dengan membentuk *Decision Tree* lebih dari satu ini juga dapat meningkatkan akurasi algoritma, dan juga memberikan fleksibilitas karena *Decision Tree* kurang fleksibel saat mengklasifikasi sampel tertentu.



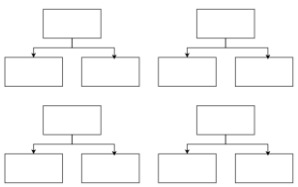
***Gambar 2-2** Ilustrasi* Random Forest

2.5 AdaBoost

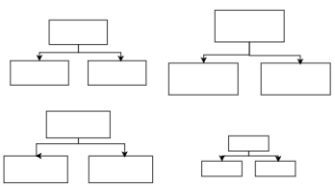
AdaBoost atau *Adaptive Boosting*, adalah sebuah *machine learning* yang dapat digunakan bersamaan dengan algoritma lain untuk meningkatkan kemampuan algoritma tersebut. *Boosting* itu sendiri merupakan sebuah teknik yang menggabungkan performa dari klasifier atau pengambil keputusan yang lemah untuk membentuk klasifier yang kuat[11]. *AdaBoost* ini akan menggunakan *Decision Tree* sebagai klasifier lemah.

AdaBoost memiliki tiga ide dasar[12], yaitu:

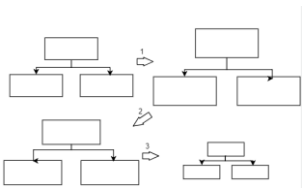
1. *Stump* : *AdaBoost* terdiri dari *Decision Tree* kecil yang hanya memiliki satu daun, dan dua node yang disebut sebagai *Decision Stump*.
2. *Value* : Setiap *Decision Stump* memiliki nilai yang berbeda. Nilai ini akan berpengaruh terhadap keputusan yang akan diambil oleh *AdaBoost*.
3. *Order* : Setiap pembentukan *Decision Stump* akan mempengaruhi *Decision Stump* berikutnya. Jadi bila ada kesalahan pada *Decision Stump* yang pertama, akan mempengaruhi pembentukan *Decision Stump* yang berikutnya.



Gambar 2-3 Ilustrasi *Stump*



Gambar 2-4 Ilustrasi *Value*



Gambar 2-5 Ilustrasi *Order*

2.6 KDDcup99 Dataset

KDDcup99 Dataset adalah kumpulan data yang digunakan pada kompetisi ketiga International Knowledge Discovery and Data Mining Tools Competition yang mana kompetisi ini dilakukan bersamaan dengan The Fifth International Conference on Knowledge Discovery and Data Mining. Didalam kompetisi tersebut, diperintahkan untuk membentuk pendeteksi intrusi yang dapat membedakat koneksi baik atau data normal, dan koneksi buruk atau data serangan. Didalam dataset ini memiliki variasi data yang disimulasikan dari intrusi pada jaringan militer[7]. Data set ini berisikan 41 atribut yang dijabarkan pada tabel dibawah.

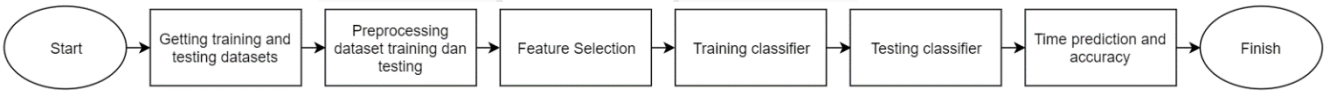
Table 2-1 Tabel Fitur didalam KDDcup99 Dataset[7]

No.	Nama Atribut	No.	Nama Atribut	No.	Nama Atribut
1	duration	15	su_attempted	29	same_srv_rate
2	protocol_type	16	num_root	30	diff_srv_rate
3	service	17	num_file_creations	31	srv_diff_host_rate
4	flag	18	num_shells	32	dst_host_count
5	src_bytes	19	num_access_files	33	dst_host_srv_count
6	dst_bytes	20	num_outbound_cmds	34	dst_host_same_srv_rate
7	land	21	is_host_login	35	dst_host_diff_srv_rate
8	wrong_fragment	22	is_guest_login	36	dst_host_same_src_port_rate
9	urgent	23	Count	37	dst_host_srv_diff_host_rate
10	hot	24	srv_count	38	dst_host_serror_rate
11	num_failed_logins	25	serror_rate	39	dst_host_srv_serror_rate
12	logged_in	26	srv_serror_rate	40	dst_host_rerror_rate
13	num_compromised	27	rerror_rate	41	dst_host_srv_rerror_rate
14	root_shell	28	srv_rerror_rate		

3. Pembahasan

3.1 Desain Sistem

Desain sistem pada penulisan Tugas Akhir ini dilakukan dengan langkah berikut:



Gambar 3-1 Diagram Alur Desain Sistem[5]

Pada langkah pertama yaitu mendapatkan dataset training dan testing dilakukan dengan cara mengunduh dari website resmi KDDCUP99[7], selanjutnya dilakukan *Preprocessing* dataset training dimana *preprocessing* dataset ini mengolah data training mentah menjadi data yang telah disortir. Didalam proses ini juga dilakukan penyocokan data dengan atribut milik KDDcup99. Setelah itu data akan dikelompokan menjadi lima kelas, yaitu: Normal, DoS, Probe, R2L, dan U2L. Dalam langkah Seleksi Fitur dilakukan pemilihan dari 41 fitur yang ada didalam KDDcup99. Fitur yang dipilih pada Tugas Akhir ini adalah 14 fitur yang pernah digunakan oleh peneliti sebelumnya[c].

Number	Feature Name	Number	Feature Name
2	protocol_type	23	count
3	service	24	srv_count
5	src_bytes	25	serror_rate
6	dst_bytes	26	srv_serror_rate
7	land	28	srv_rerror_rate
8	wrong_fragment	29	same_srv_rate
		30	diff_srv_rate

Gambar 3-2 *Fitur yang ipilih*

Setelah Seleksi Fitur, akan dilakukan *Training Classifier* yang akan melatih klasifier *Decision Tree*, *Random Forest*, dan *AdaBoost* menggunakan dataset training yang telah disiapkan. Pada langkah terakhir, klasifier akan dievaluasi berdasarkan kemampuan dan waktu yang dibutuhkannya dalam mengklasifikasi data. Kemampuan ini dilihat dari nilai *Precision*, *Recall*, dan *Accuracy* tiap klasfier yang diuji.

3.2 Hasil Pengujian dan Analisa Desain Klasifier

Hasil pengujian menggunakan dataset testing memiliki hasil sebagai berikut:

Tabel 3-1 Hasil Dataset Testing *Decision Tree*

	Attack	Normal
Attack	229850	3
Normal	8567	52026
Precision	0,964	
Recall	0,9999	
Accuracy	0,9704	
Train Time	9,359066667	
Prediction Time	1,4198	

Tabel 3-2 Hasil Dataset Testing *Random Forest*

	Attack	Normal
Attack	223893	5960
Normal	701	59892
Precision	0,9968	
Recall	0,974	
Accuracy	0,977	
Train Time	6,730933333	
Prediction Time	0,3942	

Tabel 3-3 Hasil Dataset Testing *AdaBoost*

	Attack	Normal
Attack	222951	6902
Normal	439	60154
Precision	0,998	
Recall	0,9699	
Accuracy	0,9747	
Train Time	178,64575	
Prediction Time	21,5692	

Dari ketiga hasil klasifier ini, *Random Forest* memberikan kemampuan yang lebih baik ketimbang *Decision Tree* dan *AdaBoost*. Karena *Random Forest* dapat meningkatkan akurasi sebesar 0.66% dari *Decision Tree* dengan waktu yang tidak terlalu jauh dengan *Decision Tree* itu sendiri. *AdaBoost* kurang baik untuk melakukan klasifikasi serangan dikarenakan waktu yang dibutuhkan oleh *AdaBoost* itu sendiri terbilang terlalu lama dibandingan dengan kedua klasifier sebelumnya.

Kemudian hasil analisa manual dengan sampel 1000, 5000, 10000, 15000 dan 20000 pada klasifikasi data serangan dan data normal memiliki hasil sebagai berikut:

Tabel 3-4 Hasil Dataset Manual *Decision Tree*

Decision Tree							
Data	TP	FP	TN	FN	Precision	Recall	Accuracy
1000	727	64	172	37	80,87%	91,91%	76,40%
5000	3912	136	658	294	85,60%	96,64%	84,12%
10000	8015	136	1029	820	88,62%	98,33%	88,35%
15000	9487	2596	1616	1301	85,45%	78,52%	71,92%
20000	13641	2944	1904	1511	87,75%	82,25%	75,76%
				Rata-Rata	85,66%	89,53%	79,31%

Tabel 3-5 Hasil Dataset Manual *Random Forest*

Random Forest							
Data	TP	FP	TN	FN	Precision	Recall	Accuracy
1000	478	313	138	71	77,60%	60,43%	54,90%
5000	4043	5	668	284	85,82%	99,88%	86,54%
10000	7824	327	763	1086	91,11%	95,99%	89,10%
15000	9327	2756	1098	1819	89,47%	77,19%	74,31%
20000	13402	3183	1155	2260	92,07%	80,81%	78,31%
				Rata-Rata	87,21%	82,86%	76,63%

Tabel 3-6 Hasil Dataset Manual *AdaBoost*

AdaBoost							
Data	TP	FP	TN	FN	Precision	Recall	Accuracy
1000	791	0	209	0	79,10%	100,00%	79,10%
5000	4048	0	952	0	80,96%	100,00%	80,96%
10000	7024	1127	1003	846	87,50%	86,17%	78,70%
15000	12082	1	2916	1	80,56%	99,99%	80,55%
20000	16577	8	3414	1	82,92%	99,95%	82,89%
				Rata-Rata	82,21%	97,22%	80,44%

Hasil analisis dari uji manual dapat disimpulkan sebagai berikut:

- *AdaBoost* terlihat tidak dapat melakukan klasifikasi data dengan baik terlihat pada banyaknya nilai yang kosong di kolom *False Positive* dan *False Negative*. Dikarenakan hal tersebut, hasil *Precision*, *Recall*, dan *Accuracy* pada klasifier AdaBoost terlihat stabil.
- Pada 1000 data, Decision Tree memiliki hasil yang lebih tinggi dengan *Precision* 80.87%, *Recall* 91.91%, dan *Akurasi* 76.4%. Sedangkan *Random Forest* memiliki hasil *Precision* 77.6%, *Recall* 60.43%, dan *Accuracy* 54.90%.
- Pada 5000 data, *Random Forest* memiliki hasil yang lebih tinggi dengan *Precision* 85.82%, *Recall* 99.88%, dan *Accuracy* 86.54%. Sedangkan *Decision Tree* memiliki hasil *Precision* 85.60%, *Recall* 96.64%, dan *Accuracy* 84.12%.
- Pada 10000 data, *Random Forest* memiliki hasil yang lebih tinggi dengan *Precision* 91.11%, *Recall* 95.99%, dan *Accuracy* 89.1%. Sedangkan *Decision Tree* memiliki hasil *Precision* 88.62%, *Recall* 98.33%, dan *Accuracy* 88.35%.
- Pada 15000 data, *Random Forest* memiliki hasil yang lebih tinggi dengan *Precision* 98.47%, *Recall* 77.19%, dan *Accuracy* 74.31%. Sedangkan *Decision Tree* memiliki hasil *Precision* 85.45%, *Recall* 78.52%, dan *Accuracy* 71.92%.

- Pada 20000 data, *Random Forest* memiliki hasil yang lebih tinggi dengan *Precision* 92.07%, *Recall* 80.81%, dan *Accuracy* 78.31%. Sedangkan *Decision Tree* memiliki hasil *Precision* 87.75%, *Recall* 82.25%, dan *Accuracy* 75.76%.

Rata-rata *Precision* terbaik dimiliki oleh *Random Forest* dengan hasil 87.21%, sedangkan Rata-rata *Recall* dan *Accuracy* terbaik dimiliki oleh *Decision Tree* dengan hasil masing-masing 89.53% dan 79.31%.

Kemudian hasil lama waktu prediksi menggunakan dataset manual seperti sebelumnya sebagai berikut:

Tabel 3-7 Hasil Waktu Prediksi dengan Dataset Manual (satuan detik)

Dataset	Decision Tree	Random Forest	AdaBoost
1000 data	0,0216	0,3748	0,1130
5000 data	0,0616	0,3962	0,3562
10000 data	0,0436	0,3778	0,7322
15000 data	0,1126	0,3836	0,9864
20000 data	0,1200	0,3994	1,1894

Dari hasil tersebut didapatkan bahwa Decision Tree memiliki waktu yang paling cepat ketimbang dua klasifier lainnya. Waktu yang dibutuhkan oleh Decision Tree dan AdaBoost untuk memprediksi data bertambah seiring dengan bertambahnya jumlah data didalam dataset. Random Forest tidak terpengaruh dengan jumlah data yang ada didalam dataset.

Kemudian hasil analisa berdasarkan jumlah fitur yang digunakan sebagai berikut:

Table 3-1 Analisa berdasarkan Jumlah Fitur

	Decision Tree			Random Forest			AdaBoost		
Feature Selected	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
7	69,20%	72,20%	67,30%	83,00%	99,40%	82,99%	82,60%	93,40%	78,41%
14	85,66%	89,53%	79,31%	87,21%	82,86%	76,63%	82,21%	97,22%	80,44%
21	85,80%	88,40%	79,31%	87,00%	95,00%	84,00%	85,20%	95,00%	80,44%
28	85,80%	87,80%	79,31%	85,00%	85,60%	76,00%	83,80%	96,60%	81,18%
35	85,80%	89,60%	79,31%	70,80%	51,80%	52,55%	83,80%	93,80%	79,58%

Hasil tersebut menunjukan bahwa dengan bertambahnya jumlah fitur yang digunakan membuat performa pada klasifier *Decision Tree* lebih baik dan stabil pada angka 85% di *Precision*, lebih dari 70% di *Recall* dan stabil pada 79% di *Accuracy*. Jumlah fitur yang semakin banyak mengurangi performa dari klasifier *Random Forest* sampai ke angka 70% pada *Precision*, 50% pada *Recall*, dan 52% pada *Accuracy*. *AdaBoost* tidak terlihat terpengaruh dengan jumlah fitur yang digunakan

4. Kesimpulan

Bedasarakan hasil desain klasifier yang dibentuk dihasilkan kesimpulan diantaranya:

a. Klasifier yang paling efisien untuk mendeteksi serangan adalah *Decision Tree* dengan hasil menggunakan dataset Testing *Precision* 96.41%, *Recall* 100%, dan *Accuracy* 97.05%. Hasil rata-rata menggunakan dataset manual *Precision* 85.66%, *Recall* 89,53%, dan *Accuracy* 79.31%. Dan juga waktu yang dibutuhkan untuk melatih dan memprediksi adalah 9,36 detik dan 1,42 detik.
b. *Random Forest* merupakan klasifier paling efisien kedua setelah *Decision Tree* dikarenakan walaupun *Random Forest* memiliki nilai yang lebih tinggi saat menggunakan dataset testing yang ditunjukan dengan nilai *Precision* 99.68%, *Recall* 97.4%, dan *Accuracy* 97.7% serta dataset

manual dengan nilai *Precision* 87.21%, *Recall* 82.86%, dan *Accuracy* 76.63%. Hasil yang diberikan oleh *Decision Tree* lebih stabil walaupun waktu yang dibutuhkan oleh *Random Forest* untuk memprediksi serangan lebih cepat dibandingan dengan *Decision Tree* dengan hasil lama waktu training 6.73 detik dan lama waktu prediksi 0.39 detik.

c. *AdaBoost* merupakan klasifier yang kurang cocok untuk mendeteksi serangan dikarenakan walaupun dengan hasil yang stabil pada dataset testing yang nilai *Precision* 99.8%, *Recall* 96.99%, dan *Accuracy* 97.47% dan pada dataset manual *Precision* 82.21%, *Recall* 97.22%, dan *Accuracy* 80.44% *AdaBoost* membutuhkan waktu yang sangat lama dibandingkan dengan klasifier *Decision Tree*, dan *Random Forest* dengan lama waktu training 178.64 detik dan lama waktu prediksi 21.56 detik.

5. Saran

a. Dataset yang digunakan sebaiknya memiliki jenis serangan yang sama atau mirip untuk mendapatkan hasil yang lebih baik.
b. Dataset yang digunakan sebaiknya dilakukan pengecekan agar dataset yang akan digunakan diketahui kecocokan pada aplikasi dan juga perangkatnya.

Daftar Pustaka:

[1] Varshovi, A., Rostamipour, M., & Sadeghiyan, B. (2014). "A fuzzy Intrusion Detection System based on categorization of attacks". *2014 6th Conference on Information and Knowledge Technology, IKT 2014*, (lKT), 50–55. https:// doi.org/10.1109/IKT.2014.7030332
[2] Shanmugam, B., & Idris, N. B. (2009). "Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks". *SoCPaR 2009 - Soft Computing and Pattern Recognition*, 212–217. https://doi.org/ 10.1109/SoCPaR.2009.51
[3] https://inet.detik.com/security/d-4791525/ada-129-juta-serangan-siber-di-indonesia-pengamat-wajar
[4] Potteti, S., & Parati, N. (2018). "Intrusion detection system using hybrid Fuzzy Genetic algorithm". *Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017*, *2018–Janua*, 613–618. https:// doi.org/10.1109/ICOEI.2017.8300775
[5] Gupta, V., Singh, M., & Bhalla, V. K. (2014). Pattern matching algorithms for intrusion detection and prevention system: A comparative analysis. *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, pp. 50–54. https://doi.org/10.1109/ICACCI.2014.6968595
[6] Desale, K. S., Kumathekar, C. N., & Chavan, A. P. (2015). Efficient intrusion detection system using stream data mining classification technique. *Proceedings - 1st International Conference on Computing, Communication, Control and Automation, ICCUBEA 2015*, pp. 469–473. https://doi.org/10.1109/ICCUBEA.2015.98
[7] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). The Floyd-Warshall algorithm. In *Introduction to Algorithms*. https://doi.org/10.1017/CBO9781107415324.004
[8] Quinlan, J. R. (1987). *Simplifying decision trees. International Journal of Man-Machine Studies, 27(3), 221–234.* doi:10.1016/s0020-7373(87)80053-6
[9] Ho, T. K. (1995). Random decision forests. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *1*, 278–282. https://doi.org/10.1109/ICDAR.1995.598994
[10] Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337–407. https://doi.org/10.1214/aos/1016218223
[11] Starner, Josh, "AdaBoost, Clearly Explained", YouTube, StatQuest with Josh Starmer, 14 Januari 2019, 20:53, https://youtu.be/LsK-xG1cLYA. [Diakses pada 10 Juni 2019]
[12] KDD-cup 1999 data set, Information and Computer Science University of California, Irvine. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Diakses 10 November 2018]
[13] Muzammil, M. J., Qazi, S., & Ali, T. (2013). Comparative analysis of classification algorithms performance for statistical based intrusion detection system. *2013 3rd IEEE International Conference on Computer, Control and Communication, IC4 2013*. https://doi.org/10.1109/IC4.2013.6653738