Jurnal Sistim Informasi dan Teknologi
https://jsisfotek.org/index.php
2023 Vol. 5 No. 1 Hal: 7-15 e-ISSN: 2686-3154

Sistem Deteksi Intrusi pada Server secara Realtime Menggunakan Seleksi Fitur dan Firebase Cloud Messaging

Faizal Riza✉,

[1]Dinas Kependudukan Dan Pencatatan Sipil Kabupaten Bengkalis

faizal.jaringan@gmail.com

Abstract

Intrusion detection is one of the fundamental parts of a security tool, such as adaptive security tools, intrusion detection systems, intrusion prevention systems and firewalls. There are various kinds of intrusion detection techniques used, the main problem of this intrusion technique is the performance problem. The accuracy of intrusion detection techniques greatly affects system performance, so it needs to be improved to reduce the false alarm rate and increase the detection rate in overcoming problems in performance, multilayer perceptron, and Support Vector Machine (SVM). This technique shows limitations and is inefficient for use in large data sets, such as system and network data. This study aims to improve the efficiency of classification techniques for large data sets. This technique applies machine learning, namely SVM, random forest, and extreme machine learning. Development of Network Security Layer (NSL) and data mining datasets as benchmarks in evaluating intrusion detection mechanisms. The results of this study indicate that Extreme Learning Machine (ELM) outperforms other approaches. Utilization of Firebase Cloud Messenging because it can work with multi-platforms in addition to the availability of a filestore that can store all logs created by the JALA application.

Keywords: Attack, Intrusion Detection, Firebase, Network, Security.

Abstrak

Deteksi intrusi merupakan salah satu bagian yang mendasar dari sebuah alat keamanan, seperti peralatan keamanan adaptif, sistem deteksi intrusi, sistem pencegahan intrusi serta firewall. Ada berbagai macam teknik deteksi intrusi yang digunakan, masalah utama dari teknik intrusi ini dihadapkan dengan masalah kinerja. Keakurasian teknik pendeteksian intrusi sangat mempengaruhi kinerja sistem, maka perlu ditingkatkan untuk mengurangi tingkat alarm palsu dan meningkatkan tingkat deteksi dalam mengatasi masalah pada kinerja, multilayer perceptron, dan Support Vector Machine (SVM). Teknik ini menunjukkan adanya keterbatasan dan tidak efisien untuk digunakan dalam kumpulan data yang besar, seperti sistem dan data jaringan. Penelitian ini bertujuan untuk meningkatkan efisien teknik klasifikasi terhadap kumpulan data yang besar. Teknik ini menerapkan pembelajaran mesin, yaitu SVM, hutan acak, dan mesin pembelajaran ekstrim. Pengembangan Network Security Layer (NSL) dan dataset penambangan data sebagai tolok ukur dalam evaluasi deteksi intrusi mekanisme. Hasil penelitian ini menunjukkan bahwa Extreme Learning Machine (ELM) mengungguli pendekatan lainnya. Pemanfaatan Firebase Cloud Messenging dikarenakan dapat bekerja dengan multi-platform di samping tersedianya filestore yang dapat menyimpan semua log yang di buat oleh aplikasi JALA.

Kata kunci: Serangan, Deteksi Intrusi, Firebase, Jaringan, Kemanan.

1. Pendahuluan

Konektifitas yang semakin meluas dalam sistem informasi saat ini menimbulkan tantangan baru bagi keamanan. Mekanisme keamanan konvensional telah berhasil mencapai tujuan kerahasiaan, integritas, orisinalitas, dan ketersediaan yang terdefinisi dengan baik. Namun, dengan meningkatnya kompleksitas sistem dan jangkauan serangan, memberikan keamanan melalui metode tradisional semakin sulit dicapai [1].

Penelitian Keamanan Jaringan menjadi pusat perhatian mengingat kerentanan ekosistem komputasi dengan sistem jaringan yang semakin beralih ke tangan peretas. Pada kanvas keamanan jaringan, sistem deteksi intrusi adalah alat penting yang digunakan untuk mendeteksi serangan dunia maya secara tepat waktu. *Machine Learning* sering digunakan untuk mendeteksi intrusi karena pemahaman mereka tentang sistem deteksi intrusi dalam meminimalkan ancaman keamanan. Namun, beberapa pengklasifikasi tunggal memiliki keterbatasan dan menimbulkan tantangan bagi pengembangan *Intrusion Detection System* yang efektif [2].

Identifikasi serangan dalam jaringan komputer dibagi menjadi dua kategori, yaitu deteksi intrusi dan deteksi anomali dari segi informasi yang digunakan dalam tahap pembelajaran. Deteksi intrusi menggunakan lalu lintas rutin dan lalu lintas serangan. Metode deteksi abnormal mencoba untuk memodelkan perilaku normal sistem, dan setiap kejadian yang melanggar

Diterima: 31-08-2022 | Revisi: 07-09-2022 | Diterbitkan: 31-03-2022 | doi: 10.37034/jsisfotek.v5i1.161

model ini dianggap sebagai perilaku yang mencurigakan [3].

Keamanan komputer didefinisikan sebagai teknik dan prosedur administratif yang diterapkan pada sistem komputer untuk memastikan ketersediaan, efektivitas, dan kerahasiaan transfer informasi dalam sistem komputer dan memastikan akses. Keamanan komputer dapat diklasifikasikan menjadi tiga bidang yaitu pencegahan, deteksi dan reaksi. Area kedua dicakup oleh sistem deteksi intrusi atau biasa disebut *Intrusion Detection System* (IDS), didefinisikan sebagai identifikasi dan respons dari setiap perilaku jahat yang menargetkan sumber daya komputer dan jaringan. Ini dapat diklasifikasikan menjadi dua jenis utama: basis tanda tangan dan basis penyalahgunaan. Basis tanda tangan bergantung pada tanda tangan penyerang, sedangkan basis penyalahgunaan bertindak untuk menemukan perilaku abnormal dari pengguna [4].

IDS didefinisikan sebagai pekerjaan yang menggunakan teknik dan mekanisme khusus untuk mendeteksi gangguan. Intrusi didefinisikan sebagai upaya untuk mengkompromikan keamanan, efektivitas, atau mekanisme keamanan yang tumpang tindih dalam suatu sistem atau jaringan. IDS memonitor dan menganalisa kejadian-kejadian dalam sebuah komputer atau sistem jaringan untuk mendeteksi tanda-tanda penyusupan, dimana sistem deteksi penyusupan adalah komponen yang dapat diprogram atau fisik yang bekerja secara otomatis pada pemantauan seperti yang disebutkan di atas untuk mengidentifikasi masalah keamanan. Penyerang yang mengganggu sistem dengan masuk sebagai pengguna jaringan yang sah untuk mendapatkan hak istimewa tambahan dari pengguna biasa yang tidak sah dan pengguna sah yang menyalahgunakan hak yang diberikan kepada mereka. Namun IDS mendeteksi serangan atau perilaku abnormal yang terjadi di dalam jaringan dan segera mengeluarkan alarm yang mengetahui orang yang bertanggung jawab atas keamanan di jaringan saat serangan terjadi dan memintanya untuk mengambil tindakan yang diperlukan [5].

Menguji sistem pendeteksian serangan atau penyusupan menggunakan metode seleksi fitur dengan menggunakan dataset UNSW-NB-15. Hasil penelitian menunjukkan bahawa metode *XGBoost-based feature selection* meningkatkan ketepatan pengujian dari 88.13% hingga 90.85% untuk skema binary *classification* [6].

Membandingkan Teknik *Machine Learning*, yaitu, SVM, *Random Forest*, dan *Extreme Learning Machine* (ELM) yang diterapkan pada Dataset NSL-KDD. Ketiga teknik yang disebutkan diuji kemampuannya dalam klasifikasi. *Dataset* NSL-KDD digunakan karena dianggap sebagai penanda aras dalam penilaian mekanisme pendeteksian pencerobohan. Hasilnya menunjukkan bahawa ELM mengatasi metode lainnya [7].

Pengusulkan penggunaan algoritma metode seleksi *hybrid* dengan *Guided Regularized Random Forest-Feature Weighting* SVM (GRRF-FWSVM) + Cat Boost dan diterapkan pada dataset KDD 99 Cup. Hasil penelitian menunjukkan bahwa metode pemilihan fitur hybrid yang diusulkan dengan klasifikasi GRRF-FWSVM + *CatBoost* sebesar 98,55% pada set pengujian dibandingkan dengan dua model benchmark lainnya. Model yang diusulkan telah mencapai tingkat kinerjanya dengan tingkat akurasi yang tinggi [8].

Pengusulkan penggunaan algoritma *Convolutional Neural-based learning Classifier System* (CN-LCS) yang merupakan model gabungan antara *Learning Classifier System* (LCS) konvensional dengan *Convolutional Neural Network* (CNN) yang diterapkan pada *synthetic query dataset*. Kombinasi LCS gaya Pittsburgh yang diubah sesuai untuk pengoptimuman peraturan *feature selection* dan penggunaan CNN satu dimensi untuk pemodelan dan klasifikasi sebagai ganti peraturan tradisional mengatasi pengeluar pembelajaran mesin yang lain [9].

Penggunakan algoritma *Convolutional Neural Networks* (CNN) dengan membandingkannya dengan *Recurrent Neural Network* (RNN) pada dataset NSL-KDD. Sejumlah besar *Data Mining* (DM), *Machine Learning* (ML), dan teknik kecerdasan buatan digunakan dalam pengembangan IDS, banyak dari penelitian sebelumnya di bidang ini telah berfokus pada penggunaan algoritma klasifikasi dan teknologi agregasi untuk meningkatkan operasi deteksi intrusi. Penelitian ini digunakan untuk meningkatkan kinerja IDS (akurasi tinggi, tingkat deteksi, dan mengurangi tingkat alarm palsu) [10].

Peneliti mencadangkan model bersatu yang menggabungkan *Multiscale Convolutional Neural Network dengan Long Short-Term Memory* (MSCNN-LSTM). Model ini pertama kali menggunakan *Multiscale Convolutional Neural Network* (MSCNN) untuk menganalisis ciri spasial dataset, dan kemudian menggunakan Jaringan *Long Short-Term Memory* (LSTM) untuk memproses ciri temporal. Menggunakan Set Data UNSW-NB15. Hasil eksperimen menunjukkan bahawa hasil MSCNN-LSTM l tersebut dapat meningkatkan ketepatan dengan berkesan berbanding dengan metode lain yang ada dan secara efektif mengurangkan FAR kerana secara automatik mempelajari ciri-ciri spasial-temporal, yang meningkatkan prestasi keseluruhan IDS [11].

Peneliti membangun DL-IDS (*deep learning-based intrusion detection system*), yang menggunakan rangkaian *hibrid Convolutional Neural Network* (CNN) dan *Long Short-Term Memory Network*

(LSTM). Menggunakan Set Data CICIDS2017. Hasilnya menunjukkan bahawa DL-IDS masing-masing mencapai 98.67% dan93.32% dalam keseluruhan ketepatan dan skor F1, yang menunjukkan prestasi yang lebih baik daripada semua model pembelajaran mesin. Juga, dibandingkan dengan model CNN-only dan model LSTM sahaja, DL-IDS mencapai lebih dari 99.50% dalam ketepatan semua jenis serangan dan mencapai prestasi terbaik di antara ketiga-tiga model ini [12].

Peneliti mengusulkan model pembelajaran mendalam yang dibangun berdasarkan lapisan jaringan saraf *convolutional* (CNN) dan menggunakan lapisan Memori Jangka Pendek (LSTM) yang disebut CNN-LSTM untuk mengklasifikasikan setiap jaringan lalu lintas. Menggunakan kumpulan data NSL-KDD. NLS-KDD memiliki dua *set test* yaitu KDDTest+ dan KDDTest-. Banyak penelitian hanya berfokus pada KDDTest+ karena lebih mudah untuk diklasifikasikan daripada KDDTest-. Namun, metode yang diusulkan dapat menggantikan metode lain yang tersedia di KDDTest+ atau KDDTest- [13].

Peneliti mengusulkan model deteksi intrusi jaringan multiklasifikasi berdasarkan jaringan saraf *convolutional*, dan algoritma yang dioptimalkan. Penelitian menggunakan *Dataset* KDD-CUP 99 dan NSL-KDD. Dalam penelitian ini, peneliti membandingkan hasil eksperimen dengan model *deep learning* DNN, LSTM-RNN, GRU-RNN, DBN, KNN, ICNN, dan sebagainya. Hasil eksperimen menunjukkan bahwa model deteksi intrusi jaringan yang diusulkan meningkatkan akurasi dan retraksi berkurangn kadar positif palsu, dan memperoleh hasil pendeteksian yang lebih baik untuk mendeteksi serangan yang tidak diketahui [14].

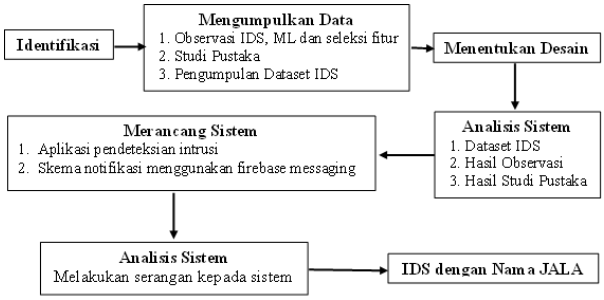
Penelitian mengusulkan sistem deteksi intrusi baru yang disebut TR-IDS, yang memanfaatkan fitur statistik dan fitur muatan dengan mengimplementasikan penyematan Word dan jaringan saraf konvolusi teks (Text-CNN) untuk mengekstrak informasi dari muatan secara efektif. Menggunakan kumpulan data ISCX2012. Teknik penyisipan kata mempertahankan hubungan semantik antara setiap *byte* dan mengurangi dimensi fitur, dan kemudian *Text*-CNN digunakan untuk mengekstrak fitur dari setiap beban. Eksperimen ekstensif menunjukkan kinerja unggul dari metode yang diusulkan [15].

Untuk membangun sebuah IDS, diperlukan kinerja sistem yang mumpuni agar dapat berjalan normal. Seluruh proses pendeteksian harus dapat dilakukan dalam jumlah besar sekaligus dengan menerapkan arus. Bahasa pemrograman Go dapat menangani kasus kasus seperti yang ditemui. Salah satu pilar inti Go adalah fitur pemrograman kurikulernya, termasuk penguncian memori bersama untuk sinkronisasi *thread*, dan penggunaan saluran penerusan pesan eksplisit [16].

Berdasarkan penelitian ini dalam Langkah memberikan perlindungan pada server terhadap intrusi yang ada digunakan IDS secara *real time* menggunakan seleksi fitur dan *firebase cloud mesaaaging.*

2. Metodologi Penelitian

Alur penelitian disajikan pada Gambar 1.



Gambar 1. Kerangka Kerja Penelitian

2.1 Pengumpulan Data

Perancang sistem di penulisan ini, metode yang digunakan untuk mengumpulkan data ada beberapa metode, diantaranya:

a. Identifikasi Masalah

Identifikasi masalah dalam penulisan penelitian ini adalah bagaimana membuat sebuah aplikasi untuk mendeteksi serangan atau penyusupan terhadap sebuah web server dan menampilkan alert hasil deteksi kepada administrator dalam bentuk *firebase messaging*.

b. Observasi

Guna mengumpulkan informasi mengenai kebutuhan sistem penulis melakukan pengumpulan data dengan cara observasi mengenai IDS, *machine learning*, *feature selection*, dan mengamati beberapa contoh sistem yang hampir serupa sebagai perbandingan.

c. Studi Pustaka

Pengumpulan data dengan cara membaca buku dan literatur lainnya yang dapat dijadikan acuan berkaitan dengan penelitian untuk mengembangkan sistem yang baru, bf. aik membaca buku konvensional maupun *e-book*.

d. Mengumpulkan *Dataset* Statistik

Peneliti mengumpulkan data trafik jaringan dan mengklasifikasikan data berdasarkan trafik normal dan abnormal (anomali), pada penelitian ini menggunakan *datasets* yang sudah ada yaitu CICIDS-2019.

e. Menentukan Desain

Perancangan sistem ini berfokus pada desain *software*. *Software* didesain agar mudah digunakan oleh pengguna baik yang sudah profesional maupun yang masih awam. Menentukan desain ini seperti halnya menentukan bagaimana antar muka tampilan *software* ini, alur *software* dan lain sebagainya.

f. Analisis Sistem

Berdasarkan hasil observasi dan studi pustaka yang telah dilakukan penulis, masih sedikit informasi mengenai sistem deteksi serangan atau penyusupan terhadap *web server*, belum banyak yang menerapkan dan mengkaji mengenai sistem tersebut. Hal ini tentu saja banyak kendala yang dihadapi seperti mengoptimalkan *rules* deteksi sehingga sistem dapat lebih akurat mendeteksi serangan dan mengirim peringatan secara langsung kepada administrator.

g. Merancang Sistem

Berdasarkan desain yang telah ditentukan oleh penulis, selanjutnya adalah merancang sistem, dalam tahap ini penulis merancang sistem perangkat lunak pendeteksian serangan atau penyusupan pada *web server*. dalam tahap ini penulis mengumpulkan dataset, mengolah *data sampling*, *preprocessing*, dan melakukan *feature selection* menggunakan bahasa pemrograman Python pada Google Colab. Fitur dan *rules* yang telah diseleksi kemudian dimasukkan ke dalam sistem *Intrusion Detection* yang dikembangkan dengan bahasa pemrograman Go, peringatan yang muncul akan dikirimkan ke *firebase messaging*.

h. Sistem Testing

Testing adalah sebuah tahap pengujian dari sistem yang penulis rancang sebelum diimplementasikan, dalam hal ini penulis melakukan pengujian sistem pendeteksian serangan atau penyusupan pada *web server* dengan mensimulasikan serangan pada *web server* yang disiapkan dan memonitoring sistem dan kinerja *web server* hingga penulis berkesimpulan sistem ini layak untuk diimplementasikan.

i. Implementasi Sistem

Setelah sistem lolos dari tahap testing maka tahap selanjutnya adalah implementasi sistem, pada tahap ini penulis mengimplementasikan sistem yang telah dibuat di sebuah *web server* yang telah penulis siapkan, penulis melakukan instalasi perangkat lunak pada *web server* dan sistem pun sudah siap untuk diimplementasikan.

j. Desain Mekanisme yang Diusulkan

Penelitian ini,menggunakan pendekatan kuantitatif digunakan sebagai metode utama karena karakteristik tertentu, seperti ukuran kinerja, evaluasi dataset dan kegunaan hasil. Penelitian ini menggunakan siklus deduktif karena lebih tepat untuk menguji solusi yang diusulkan. Berdasarkan kerangka yang diusulkan, desain penelitian dikategorikan ke dalam 3 fase utama, yaitu *Pre-Processing* (*Feature Selection*), *Anomaly Detection* (*Rule based detection*) dan *Post-Processing* (Deteksi Serangan).

k. Fase *Pre-Processing*

Seleksi fitur merupakan bagian dari *fase pre-processing* di IDS yang bertujuan untuk mengurangi dimensi data dengan menghilangkan fitur yang tidak relevan dan berlebihan. Ada dua metode umum untuk feature selection: filter dan *wrapper*. Berdasarkan literatur, dua jenis metode filter adalah filtering rank dan FBSE. Hasil dari fase ini adalah sebuah data sampling yang siap dimasukkan ke dalam kamus aturan yang menampung aturan-aturan yang berguna untuk menandai suatu anomali sebagai suatu ancaman pada sistem jaringan [17].

l. Fase Deteksi Anomali

Pada penelitian kali ini digunakan pendekatan dalam deteksi anomali didasarkan aturan-aturan yang telah diolah pada *fase pre-processing*. Cara kerjanya deteksi anomali adalah membandingkan data yang dikumpulkan untuk memeriksa apakah perilaku yang tidak wajar dapat diberi label sebagai serangan. Dalam kasus ini anomali dideteksi dari perubahan log pada web server yang selalu terjadi terubahan dinamis pada server saat terjadi akses data. Ada dua pendekatan umum saat pembuatan rule atau aturan yaitu rule statis dan rule dinamis. Rule statis merupakan pendekatan dimana pengguna telah lebih dahulu mendefinisikan rule secara manual sebelum pengguna melakukan analisis terhadap perilaku tidak wajar pada jaringan. Misalnya dalam kasus ini menggunakan *resource rule* dari luar yang telah disediakan di forum-forum internet. Lalu pendekatan lainnya adalah rule dinamis, yang mana mengunakan algoritma data mining untuk mendefinisikan rule.

m. Fase *Post-Processing*

Pada IDS, fase post-processing biasanya merupakan fase ketika serangan yang telah diidentifikasi oleh sistem, diproses lebih lanjut lagi. Prosedur inisiasi ulang untuk mendeteksi serangan serupa di masa depan jarang diabaikan, yang akan memakan lebih banyak waktu dan

sumber daya. Keuntungan dari pendekatan tanda tangan atau signature adalah mengurangi waktu deteksi untuk mendeteksi serangan serupa di masa depan [18]. Jadi dalam penelitian ini, pendekatan tanda tangan diusulkan sebagai bagian dari strategi deteksi dimana serangan yang benar terdeteksi oleh sistem diubah menjadi satu set tanda tangan. Analisis lebih lanjut penting untuk memastikan serangan yang terdeteksi ditanggapi segera setelah diidentifikasi. Studi sebelumnya berfokus pada memprioritaskan serangan yang diketahui sedangkan lebih sedikit studi yang mengadopsi teknik ini untuk memprioritaskan serangan yang tidak diketahui [19], [20]. Dalam studi ini, model intrusion priority (IPM) diusulkan untuk mengurutkan serangan yang tidak diketahui dari serangan yang paling kritis diikuti oleh serangan yang kurang kritis menurut empat tingkat keparahan: tertinggi, tinggi, rendah dan terendah.

n. Rancangan Eksperimental

o. Hybrid Feature Selection

Desain HFS melibatkan kombinasi kekuatan FBSE dan WBSE. Dalam HFS, empat teknik pencarian yang berbeda dibandingkan untuk menemukan metode pencarian terbaik yang dapat menghasilkan tingkat akurasi deteksi tertinggi, yaitu *best first, greedy, genetic search*, dan PSO. Eksperimen pendahuluan dilakukan dengan menghilangkan fitur yang tidak relevan dan berlebihan untuk memilih fitur yang optimal. Awalnya, tujuan menggunakan FBSE adalah untuk mengurangi upaya komputasi WBSE dengan menyaring fitur yang tidak signifikan dan berlebihan. Selain itu, proses dilanjutkan dengan pencarian subset yang optimal untuk meningkatkan kinerja klasifikasi yang dipilih sebelumnya oleh FBSE.

p. Deteksi anomali berdasarkan rule.

Eksperimen dan analisis dilakukan untuk mendapatkan aturan-aturan berdasarkan hasil processing dari dataset. Rule yang didapatkan nantinya dimasukkan ke dalam sebuah kamus rule yang menyimpan rule-rule pada dataset sebelumnya. Kemudian nantinya akan dilakukan processing terhadap log pada webserver secara rutin untuk mendapatkan rule-rule baru lagi yang akan semakin memperkaya rule pada IDS yang dirancang.

q. Pengukuran Hasil Evaluasi

Bagian ini menganalisis metrik kinerja yang digunakan dalam penelitian ini. Metrik kinerja utama yang digunakan di bidang IDS diukur dari segi deteksi, akurasi, dan tingkat alarm palsu. Namun, untuk menghitung metrik kinerja utama, indikator utama lainnya seperti (*True Positive, True Negative, False Positive and False Negative*) juga diperlukan. Rumus disajikan pada Persamaan (1) dan Persamaan (2).

$$TPR\ (True\ Positive\ Rate) = \frac{TP}{TP + FN} \quad (1)$$

Dimana TPR (*True Positive Rate*) merupakan sebuah nilai keakuratan sebuah perbobaan, TP (*True Positive*) digambarkan sebagai jumlah data serangan sebenarnya yang telah ditandai dengan benar dan FN (*False Negative*) adalah jumlah data serangan yang salah terdeteksi sebagai data normal. Serangan dapat merusak karena kegagalannya untuk dideteksi oleh sistem. Umumnya, FN sulit untuk dihitung karena tidak ada cacat yang dapat ditemukan oleh IDS ketika itu terjadi. Sistem deteksi yang ideal harus mencapai FN yang lebih rendah, terutama mendekati nol.

$$FPR\ (False\ Positive\ Rate) = \frac{FP}{FP + TN} \quad (2)$$

Dimana FPR (*False Posite Rate*) merupakan jumlah data normal yang dianggap sebagai intrusi oleh sistem dibagi dengan jumlah data normal pada dataset. FN (*False Positive*) didefinisikan sebagai jumlah data normal yang salah terdeteksi sebagai data serangan. Idealnya, sistem deteksi harus mencapai tingkat deteksi palsu yang lebih rendah untuk respons penanganan insiden yang lebih baik dan TN (*True Negative*) mengacu pada jumlah data normal sejati yang telah diklasifikasikan dengan benar.

3. Hasil dan Pembahasan

Pada tahapan ini, peneliti melakukan analisa terhadap pemanfaatan *feature selection* dalam pembuatan aplikasi IDS *server* yang nantinya data hasil *preprocessing* saat *feature selection* akan dijadikan sebagai *rule* dari IDS. Tahapan dimulai dengan pemilihan dataset yang cocok dalam menangani serangan pada *server*, lalu proses *pre-processing dataset* tersebut yang outputnya berupa sebuah *rule* yang akan digunakan pada IDS, setelah *rule* diperoleh, kemudian dilakukan pengembangan IDS yang akan berjalan di *web server* guna memantau serangan-serangan yang ada, dan pada akhir tahapan ini adalah pengembangan aplikasi yang berjalan di sisi client dalam hal ini akan digunakan oleh admin *server*, aplikasi yang dibangun memanfaatkan firebase messaging sebagai notifikasi secara *realtime* yang handal dalam memberikan peringatan kepada admin *server* saat adanya percobaan serangan yang terjadi di *server*.

3.1 Pemilihan Dataset

Sistem Deteksi Intrusi (IDS) dan Sistem Pencegahan Intrusi (IPS) adalah alat pertahanan paling penting terhadap serangan jaringan yang canggih dan terus

berkembang. Karena kurangnya dataset pengujian dan validasi yang andal, pendekatan deteksi intrusi berbasis anomali mengalami kesulitan dalam hal performa yang konsisten dan akurat. Ringkasan data set disajikan pada Tabel 1.

Table 1. Ringkasan *Dataset* CICIDS-2017

Nama file	Jenis Trafik	Jumlah Rekod
Monday-WorkingHours.pcap_ISCX.csv	Benign	529.918
Tuesday-WorkingHours.pcap_ISCX.csv	Benign SSH-Patator FTP-Patator	432.074 5.897 7.938
Wednesday-WorkingHours.pcap_ISCX.csv	Benign Dos Hulk Dos GoldenEye Dos Slowloris Dos Slowhttptest Heartbleed	440.031 231.073 10.293 5.796 5.499 11
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	Benign Web Attack-Brute Force Web Attack-Sql Injection Web Attack-XSS	168.186 1.507 21 652
Thursday-WorkingHours-Afternoon-Infilteration.pcap_ISCX.csv	Benign Infiltration	288.566 36
Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign Bot	189.067 1.966
Friday-WorkingHours-Afternoon-Portscan.pcap_ISCX.csv	Benign Portscan	127.537 158.930
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	Benign DdoS	97.718 128.027
Jumlah Rekod		*2.830.743*

3.2 Tahapan *PreProsesing*

Pada tahap ini terdapat 4 langkah utama yang dilakukan, yaitu melakukan persiapan *dataset*, melakukan *Feature selection* pada *training* data menggunakan Information Gain. Lalu terakhir didapatkan output dari tahap ini berupa hasil analisis berupa parameter berikut ini: TPR, FPR, *Precision*, *Recall*, *Accuracy*, Persentase salah pengklasifikasian, dan waktu eksekusi. Tahap Persiapan *Dataset* disajikan pada Tabel 2.

Tabel 2. Distribusi Data dari Serangan yang telah Relabelling pada 20% total Dataset

Label Baru	Label Lama	Jumlah Contoh	Pecahan ke kelas mayoritas	Pecahan ke Jumlah Contoh
Normal	Benign	454.396	100,000	80,250
Bot	Bot	367	0,081	0,060
Brute Force	FTP-Patator, SSH-Patator	2.717	0,598	0,048
Dos/DdoS	DdoS, Dos, GoldenEye, DoS Hulk, DoS Slow, httptest, DoS slowloris, Heartbleed	76.445	16,820	13,500
Infiltration	Infiltration	6	0,001	0,000
Portscan	Portscan	31.882	7,061	5,630
Web Attack	Web Attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS	426	0,094	0,080
Total		566.239		

3.2.1. *Feature Selection* Menggunakan *Information Gain*

Information Gain adalah teknik *feature selection* yang paling banyak digunakan. Information Gain berbasis *filter* dan menggunakan peringkat atribut sederhana dan mengurangi noise yang disebabkan oleh fitur yang tidak relevan kemudian mendeteksi fitur yang memiliki sebagian besar basis informasi di kelas tertentu.

Entropi adalah ukuran ketidakpastian yang dapat digunakan untuk menyimpulkan distribusi fitur dalam bentuk ringkas, prosesnya ditunjukkan pada Algoritma 1.

```
Algoritma 1. Kalkulasi Feature Rank
1: fungsi Feature_Rank()
2:Input  Fn = Training dataset,memproses 77
fitur f1,f2,f3... f77
3:For setiap feature Fn
4:Kalkulasi Feature Information Weight dengan
Information Gain
5: Ranking fitur berdasarkan Weight
6: Simpan Rank, Feature ID, Feature name dan
feature Weight pada Feature_Ranked data
```

Fitur-fitur pada dataset diberikan ID dari 1 hingga 77. Informasi Gain memberi peringkat fitur berdasarkan nilai bobotnya dan bobot minimum ditentukan secara

manual menggunakan pendekatan *try-error* yang disajikan pada Tabel 3.

Tabel 3. Fitur yang dipilih berdasarkan Information Gain

Feature Weight	Jumlah Fitur yang dipilih
>0.6	4
>0.5	15
>0.4	22
>0.3	35
>0.2	52
>0.1	57
Semua	77

3.2.2. Analisa Performa *Feature Selection* Menggunakan *Random Forest*

Random Forest adalah salah satu metode *classifier* ensemble. Jika pengklasifikasi dalam ansambel adalah pengklasifikasi pohon keputusan, maka kumpulan dari pengklasifikasi adalah "hutan". Setiap *decision tree* dibuat melalui pemilihan atribut secara acak pada setiap node untuk pemisahan.

Secara keseluruhan, pengklasifikasi RF mampu mendeteksi dengan baik lalu lintas normal, DoS/DDoS, *Port Scan*, *Brute Force*, dan lalu lintas serangan *Web* menggunakan subset fitur 35, 52, dan 77. Studi literatur mendukung temuan ini sebagai pengklasifikasi menggunakan algoritma pembelajaran *decision tree* yang kuat.

3.3 Jala IDS

Pada tahap ini dilakukan pengembangan IDS yang peneliti memberi nama JALA. IDS ini dibangun dengan menggunakan bahasa pemrograman Golang untuk mendukung kinerja yang maksimal saat berjalan di server. Secara garis besar, cara kerja JALA IDS adalah dengan membaca *log* yang tercatat pada *server*, dalam penelitian ini server yang digunakan adalah Apache. Berikut adalah konfigurasi pembacaan *log* pada *server* yang disajikan pada Gambar 4.

```
log_format: |
  $remote_addr $remote_user - [$time_local] "$request_method $request_uri $request_protocol"
  $status $body_bytes_sent "$http_referer" "$http_user_agent"
```

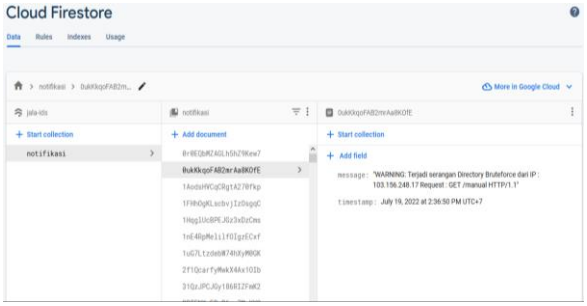
Gambar 4. Format *Log* Jala pada *Apache Server*

Log dibaca dengan mengambil data seperti pada gambar di atas, yaitu dimulai dari *remote address* hingga HTTP user agent yang digunakan oleh penyerang. Setelah *log* dibaca, kemudian IDS akan membandingkan *log* tersebut dengan model pengenalan *patern regex* yang telah diatur di IDS. Setelah didapatkan hasil rule dari tahap pre-processing, rule tersebut akan dipasang pada *config* JALA IDS. Secara *default*, JALA IDS juga mengambil *rule* dari *resources* luar yang lazim dalam penyerangan pada *server*.

Jika ada serangan yang terdeteksi oleh JALA IDS, maka IDS akan menampilkan pesan serangan pada terminal dengan warna-warna pada tiap informasi yang tercatat dan mengirimkan notifikasi berupa pesan realtime kepada aplikasi client berbasis Android menggunakan *Firebase Cloud Messaging*. Adapun pesan yang dikirim nantinya berupa notifikasi *broadcast* dan juga riwayat serangan yang dapat diperiksa kapan saja oleh admin jaringan.

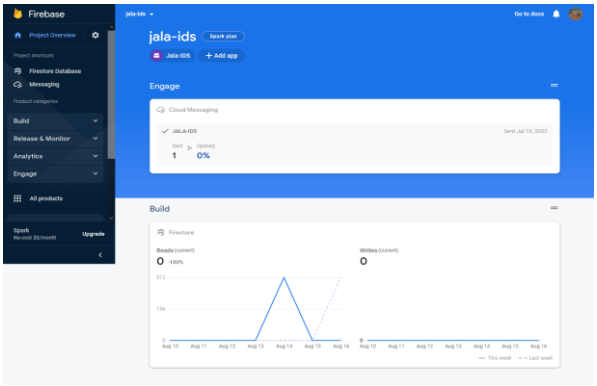
3.4 Client Jala IDS

Pada tahap ini dilakukan pengembangan aplikasi *Client* JALA IDS yang berjalan pada *platform mobile* Android. Aplikasi ini berguna sebagai penerima notifikasi serangan secara *realtime* yang dikirimkan oleh JALA IDS yang berjalan pada *server*. Aplikasi ini memanfaatkan *Firebase Cloud Messaging* sebagai pengirim notifikasi yang menghubungkan dengan JALA IDS. Lalu *Cloud Firestore* digunakan sebagai penyimpanan riwayat serangan yang disajikan pada Gambar 5.



Gambar 5. Cloud Firestore Penyimpanan Riwayat Serangan dari JALA IDS

Aplikasi *client* JALA IDS dibangun menggunakan *framework* Flutter SDK yang berguna agar kode bisa dibuild menjadi platform yang bervariatif, mulai dari Android, iOS, *Windows Desktop*, bahkan ke Web. Namun pada penelitian kali ini, platform difokuskan ke Android. Sebagai penghubung antara JALA IDS pada server dan aplikasi client ini, digunakan *Firebase* SDK yang dikembangkan oleh Google. Dimana didalam *Firebase* SDK ini bisa saling mengintegrasikan berbagai layanan *Firebase* yang ada, dimana dalam penelitian kali ini yang digunakan adalah *Firestore* dan *Firebase Cloud Messaging* yang disajikan pada Gambar 6.



Gambar 6. Tampilan *Dashboard Firebase* JALA IDS

Dari beberapa tahapan proses di atas, maka dapat menghasilkan notifikasi secara *real time* terkait keamanan *server* sebagaimana terlihat pada Gambar 7.

Gambar 7. Notifikasi Hasil Deteksi Final

4. Kesimpulan

Studi ini mengembangkan algoritma efektif yang dapat diterapkan pada IDS praktis dan memainkan peran penting dalam mendeteksi aktivitas ilegal melalui jaringan komputer. Berkembangnya jaringan serta teknologi informasi akan mengakibatkan berkembangnya jenis serangan pada masa yang akan datang sehingga sehingga perlu dilakukan penelitian lebih lanjut.

Daftar Rujukan

[1] Anwar, F., Khan, B. U. I., Olanrewaju, R. F., Pampori, B. R., & Mir, R. N. (2020). A comprehensive insight into game theory in relevance to cyber security. Indonesian Journal of Electrical Engineering and Informatics, 8(1), 189–203.

https://doi.org/10.11591/ijeei.v8i1.1810

[2] Bamhdi, A. M., Abrar, I., & Masoodi, F. (2021). An ensemble based approach for effective intrusion detection using majority voting. 19(2), 664–671. https://doi.org/10.12928/TELKOMNIKA.v19i2.18325

[3] Majidpour, J., & Hasanzadeh, H. (2020). Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks. Bulletin of Electrical Engineering and Informatics, 9(3), 1137–1148.

https://doi.org/10.11591/eei.v9i3.1724

[4] Thabit H Thabit and Yaser A Jasim. (2017). Applying IT in Accounting Environment and Computer Science Studies. In Environment and Computer Science Studies, LAP-Lambert Academic Publisher, Germany. Scholars' Press

[5] Jasim, Y. A. (2018). Improving Intrusion Detection Systems Using Artificial Neural Networks. ADCAIJ: Advances in Distributed Computing and Articial Intelligence Journal, 7(2), 49–65. https://doi.org/http://dx.doi.org/10.14201/ADCAIJ2018714965

[6] Kasongo, S. M., & Sun, Y. (2020). Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. Journal of Big Data, 7(1). https://doi.org/10.1186/s40537-020-00379-6

[7] Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection. IEEE Access, 6(c), 33789–33795.

https://doi.org/10.1109/ACCESS.2018.2841987

[8] Kavitha, G., & Elango, N. M. (2020). An approach to feature selection in intrusion detection systems using machine learning algorithms. International Journal of E-Collaboration, 16(4), 48–58. https://doi.org/10.4018/IJeC.2020100104

[9] Bu, S. J., & Cho, S. B. (2020). A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. Information Sciences, 512, 123–136. https://doi.org/10.1016/j.ins.2019.09.055

[10] Wu, K., Chen, Z., & Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. IEEE Access, 6(October 2017), 50850–50859. https://doi.org/10.1109/ACCESS.2018.2868993

[11] Zhang, J., Ling, Y., Fu, X., Yang, X., Xiong, G., & Zhang, R. (2020). Model of the intrusion detection system based on the integration of spatial-temporal features. Computers and Security, 89, 101681. https://doi.org/10.1016/j.cose.2019.101681

[12] Sun, P., Liu, P., Li, Q., Liu, C., Lu, X., Hao, R., & Chen, J. (2020). DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. Security and Communication Networks, 2020. https://doi.org/10.1155/2020/8890306

[13] Hsu, C. M., Azhari, M. Z., Hsieh, H. Y., Prakosa, S. W., & Leu, J. S. (2020). Robust Network Intrusion Detection Scheme Using Long-Short Term Memory Based Convolutional Neural Networks. Mobile Networks and Applications.

https://doi.org/10.1007/s11036-020-01623-2

[14] Liu, G., & Zhang, J. (2020). CNID: Research of Network Intrusion Detection Based on Convolutional Neural Network. Discrete Dynamics in Nature and Society, 2020.

https://doi.org/10.1155/2020/4705982

[15] Min, E., Long, J., Liu, Q., Cui, J., & Chen, W. (2018). TR-IDS: Anomaly-Based Intrusion Detection through Text-Convolutional Neural Network and Random Forest. Security and Communication Networks, 2018. https://doi.org/10.1155/2018/4943509

[16] Gabet, J., & Yoshida, N. (2020). Static Race Detection and Mutex Safety and Liveness for Go Programs (extended version). ArXiv.

[17] Liu, Z., Chang, B., & Cheng, F. (2021). An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection. Swarm and Evolutionary Computation, 65(August 2020), 100925. https://doi.org/10.1016/j.swevo.2021.100925

[18] Meng, Y., & Kwok, L. F. (2014). Adaptive blacklist-based packet filter with a statistic-based approach in network intrusion detection. Journal of Network and Computer

Applications, 39(1), 83–92. https://doi.org/10.1016/j.jnca.2013.05.009

[19] Noel, S., & Jajodia, S. (2008). Optimal IDS sensor placement and alert prioritization using attack graphs. Journal of Network and Systems Management, 16(3), 259–275. https://doi.org/10.1007/s10922-008-9109-x

[20] Zomlot, L., Sundaramurthy, S. C., Luo, K., Ou, X., & Rajagopalan, S. R. (2011). Prioritizing intrusion analysis using dempster-shafer theory. Proceedings of the ACM Conference on Computer and Communications Security, October, 59–69. https://doi.org/10.1145/2046684.2046694