Faktor Exacta

Vol. 13, No. 2, June 2021, pp. 84~91
eISSN: 2502-339X, pISSN: 1979-276X, DOI: https://doi.org/10.30998/faktorexacta.v14i2.9429

Implementasi Metode K-Medoids untuk Masalah *Intrusion Detection System* Menggunakan Bahasa Pemrograman *Matlab*

Octaviani Hutapea[1], Aini Suri Talita [2]
[1,2]Department of Informatic Engineering, Universitas Gunadarma, Indonesia

Article Info

Article history:

Received April 6, 2021
Revised July 17, 2021
Accepted July 22, 2021

Keywords:

Fuzzy K-Medoids
Intrusion Detection System
KDDCUP ' 99

ABSTRACT

Based on data from the National Cyber And Crypto Agency (BSSN) of the Republic of Indonesia from 2018 to 2021, the threat of cyber attacks continues to experience a significant increase. In 2021, a significant change that is likely to be faced is with the emergence of new smart devices, which are more than just end-users and remotely connected networked devices. Surely, gives it the attention of all parties. There are many types of cyberattacks including Malware, Phishing, Ransomeware, etc. IDS (Intrusion Detection System) is a method that can detect suspicious activity in a system or network. Implementation of the Fuzzy K-Medoids method by using the Matlab programming language that retrieves data from KDDCUP'99 which has been normalized. The data used are normal data and anomaly attack data which are categorized as DoS, Probe, R2L, and U2R. From the research conducted the accuracy percentage is around 60-89% with three types of data preprocessing.

Corresponding Author:

Octaviani Hutapea,
Department of Informatics Engineering,
Universitas Gunadarma,
Jl. Margonda Raya No.100 Depok-Jawa Barat.
Email: octaviahutapea@staff.gunadarma.ac.id

1. PENDAHULUAN

Kegiatan kejahatan siber di Indonesia saat ini cukup besar dibandingkan negara lain. Berdasarkan data yang diperoleh dari Badan Siber dan Sandi Negara (BSSN) pada tahun 2018 jumlah serangan yang menyerang Indonesia sebanyak 12.895.554 serangan, dengan jumlah serangan malware 513.863 serangan. Sedangkan pada Mei 2019 BSSN mencatat terdapat indikasi penyebaran *malware* mencapai 1.9 juta data serangan lalu terdapat kategori *attempt* yang merupakan kategori percobaan menjadi admin dalam suatu akun yang mencapai 1,1 juta data serangan. Serangan ini dilaporkan meningkat dari tahun lalu. Pada tahun 2020 tercatat oleh BSSN kurang lebih 190 juta serangan dan pada tahun 2021 ini pun diperkirakan akan mengalami peningkatan.

IDS(*Intrusion Detection System*) merupakan sebuah metode yang dapat mendeteksi aktivitas yang abnormal dalam sebuah sistem atau jaringan. IDS dapat melakukan inspeksi terhadap lalu lintas inbound dan outbound dalam sebuah sistem atau jaringan, melakukan analis dan mencari bukti percobaan intrusi penyusupan. Tugas IDS adalah memberikan peringatan kepada administrator jaringan bahwa mungkin ada serangan atau gangguan terhadap jaringan. IDS bekerja dengan cara menggunakan pendeteksian dengan pencocokan lalu lintas data pada jaringan. Hasil monitoring dari IDS bertujuan agar administrator jaringan dapat mengambil keputusan lanjutan dalam hal mengamankan informasi [1]. Data yang akan digunakan dalam penelitian ini merupakan data KDD'99. Data serangan anomali dikategorikan sebagai DoS, Probe,

R2L, dan U2R. Algoritma *Fuzzy* merupakan salah satu algoritma yang cara kerjanya merupakan memisahkan data kebeberapa kelompok-kelompok. Logika *Fuzzy* adalah suatu proses pengambilan keputusan berbasis aturan yang bertujuan untuk memecahkan masalah, dimana sistem tersebut sulit untuk dimodelkan atau terdapat ambiguitas dan ketidakjelasan yang berlimpah.

Penelitian ini adalah menghitung akurasi menggunakan metode *fuzzy logic* dengan cara mengklasifikasikan data dalam jumlah 40000 data dari dataset KDD'99 yang akan dikelompokan ke dalam 5 klasifikasi yang terdiri dari 4 jenis serangan anomali dan 1 normal dengan cara membandingkan beberapa parameter sehingga mendapatkan parameter yang tepat dengan nilai akurasi yang baik dari percobaan beberapa parameter tersebut.

1.1. Intrussion Detection System (IDS)

Dalam riset keamanan jaringan, IDS merupakan salah satu masalah yang perlu mendapatkan perhatian lebih. Dua pendekatan dasar IDS adalah deteksi penyalahgunaan dan deteksi anomaly [2]. IDS merupakan program yang mempertimbangkan kejadian di sistem selama eksekusi dan berdasarkan beberapa hal yang tidak biasa indikasi menemukan apakah sistem disalahgunakan. Beberapa jenis dari teknologi IDS adalah *Network-Based, Wireless IDS, Network Behavior Anomali Detection* dan *Host-Based* [3].

Beberapa jenis dari teknologi IDS adalah *Network-Based, Wireless IDS, Network Behavior Anomali Detection dan Host-Based.* Pengembangan dan implementasi teknologi IDS ke dalam sebuah sistem jaringan tergantung dari variasi konfigurasi jaringan itu sendiri. Pada dasarnya setiap jenis teknologi IDS mempunyai keuntungan dan kekurangan dalam hal pendeteksian, konfigurasi dan biaya, tetapi secara umum teknologi IDS yang paling sering digunakan adalah *Network-Based dan Host-Based*. Secara spesifik berikut penjelasan dari masing- masing tipe teknologi IDS.

a. *Network-Based*
Network Intrusion Detection System (NIDS) adalah salah satu tipe IDS yang populer atau paling banyak diimplementasikan kedalam sebuah system 6 jaringan. Tipe ini menganalisa paket-paket jaringan pada semua lapisan Open System Interconnection (OSI) dan membuat sebuah tindakan kepada paket tersebut. Kebanyakan NIDS lebih mudah diterapkan kedalam suatu jaringan dan dapat memantau paket dari banyak sistem sekaligus [3].

b. *WLAN IDS*
Wireless Local Area Network (WLAN) IDS menyerupai NIDS yang dapat menganalisa paket-paket jaringan. WLAN ini dapat menganalisa paket wireless secara spesifik, termasuk pemindaian pengguna eksternal yang mencoba untuk terhubung ke Access Point (AP). Karena WLAN IDS sendiri sebenarnya adalah NIDS dengan menggunakan wireless maka aturan-aturan keamanan yang diterapkan lebih luas [4].

c. *Network Behavior Anomaly Detection*
Network Behavior Anomaly Detection (NBAD) *atau Anomaly-based* IDS (AIDS) membandingkan *network traffic* dengan suatu *baseline* dan dapat mengeluarkan suatu peringatan apabila terdapat aktivitas yang berbeda dengan *baseline* [5].

d. *Host-Based*
Host-based Intrusion Detection System (HIDS) terdiri atas *Multi-Agents Systems* (MAS) dengan agen yang terinstall pada komputer *host* serta *Security Operation Center* (SOC) yang berkoordinasi satu sama lain untuk mengidentifikasi aktivitas terlarang [6].

1.2. Fuzzy Logic K-Medoid

Kaufman dan Rousseeuw, mengusulkan metode K-Medoid sebagai alternatif yang lebih baik dibandingkan dengan algoritma K-means [7]. Dalam metode ini, sebelum melakukan perhitungan jarak dari objek data ke pusat *clustering*, Pusat *clustering* K dipilih secara acak dari n objek data sehingga partisi awal dibuat berdasarkan kedekatan masing-masing objek dengan pusat massa untuk memulai partisi data. Kemudian, metode iterasi digunakan secara berkelanjutan sampai mendapatkan nilai partisi yang sesuai. Kemudian dalam metode ini juga setiap iterasi yang dilakukan setelahnya adalah memilih objek dari masing-masing data sampel pengelompokan yang dipilih berdasarkan peningkatan kualitas pengelompokan. Objek yang paling terpusat terletak di sebuah cluster yang diambil sebagai titik referensi yang sebenarnya merupakan medoid dan bukan nilai rata-rata elemen dalam sebuah cluster. Sehingga prinsip dasar metode K-medoid adalah meminimalisasi jumlah total jarak titik yang berbeda dari titik referensi yang harus dilakukan untuk partisi. Pengambilan data yang representatif dari setiap cluster dilakukan dengan secara empiris, total k cluster yang diambil sedemikian rupa sehingga masing-masing titik data yang tersisa dikelompokkan dengan medoid. Algoritma ini dapat bekerja secara efektif untuk dataset yang jumlahnya kecil tetapi kurang baik jika digunakan untuk *dataset* degan skala yang besar. Formula perhitungan *manhattan distance* [8]

$$i = \sum_{i=1}^{k} \sum_{p\Omega cj}^{n} \; \|P - Oj\| \; \ldots\ldots(1)$$

Menurut penelitian [9] didapatkan perbandingan tingkat akurasi yang diperoleh yaitu sebanyak 88.7% untuk *K-means* dan 92% untuk algoritma *K-medoids* untuk data IRIS.Dengan begitu mereka menyimpulkan bahwa algoritma *K-medoids* lebih baik dibandingkan dengan *K-means.*

1.3. Knowledge Discovery Data '99

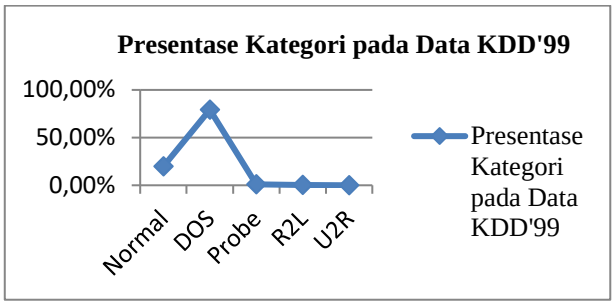
Dataset pelatihan KDD mengambil sampel dari keseluruhan dataset yang tersedia yaitu sebanyak 40000 vektor koneksi tunggal yang masing-masing berisi 41 fitur dan dilabeli dengan tepat satu jenis serangan spesifik yaitu, baik normal atau serangan. Serangan yang diberi label normal adalah rekaman dengan perilaku normal. Dataset pelatihan versi yang lebih kecil juga disediakan untuk metode pembelajaran mesin terbatas memori. Dataset pelatihan memiliki 38.67% normal dan koneksi serangan 61.33%. KDD CUP 99 telah banyak digunakan dalam serangan di jaringan. Serangan yang disimulasikan termasuk dalam salah satu dari empat kategori berikut [9]:

1. *Denial of Service Attack (DOS)*: Dalam kategori ini penyerang membuat beberapa sumber daya komputasi atau menjadikan memori terlalu sibuk atau terlalu penuh untuk menangani permintaan yang sah, sehingga menolak akses pengguna yang sah ke mesin.
2. *Users to Root Attack (U2R)*: Dalam kategori ini penyerang melakukan akses ke akun pengguna sah atau normal pada sistem dan dapat mengeksploitasi untuk mengetahui beberapa kekurangan pada sistem untuk mendapatkan akses root ke sistem.
3. *Remote to Local Attack (R2L)*: Dalam kategori ini penyerang mengirim paket ke mesin melalui jaringan tetapi yang tidak memiliki akun di mesin itu dan mengeksploitasi untuk mengetahui beberapa kekurangan pada sistem untuk mendapatkan akses lokal sebagai pengguna mesin itu.
4. *Probing Attack (PROBE)*: Dalam kategori ini penyerang berusaha mengumpulkan informasi tentang jaringan computer yang ingin diserang yg bertujuan untuk menghindari keamanannya.

Tabel 1. Pembagian Serangan Pada Data KDD'99

Target	Kategori Serangan
Back	DOS
Buffer_overflow	U2R
Ftp_write	R2L
Guess_passwd	R2L
Imap	R2L
Ipsweep	PROBE
Land	DOS
Loadmodule	U2R
Multihop	R2L
Neptune	DOS
Nmap	PROBE
Perl	U2R
Phf	R2L
Pod	DOS
Portsweep	PROBE
Rootkit	U2R
Satan	PROBE
Smurf	DOS
Spy	R2L
Teardrop	DOS
Warezclient	R2L
Warezmaster	R2L

Pada dataset KDD'99 secara keseluruhan pembagian presentasi data serangan dan normal adalah sebagai berikut, dari 494.019 catatan, di antaranya 97.277 (19,69%) adalah 'normal', 391.458 (79,24%) DOS, 4.107 (0,83%) Probe, 1.126 (0,23%) R2L dan 52 (0,01%) U2R serangan. Ditampilkan pada kurva di bawah ini.



Gambar 1. Kurva Presentase Kategori pada Data KDD'99

1.4. Penelitian Terkait

Pada penelitian [2] dilakukan 5 kali percobaan dari 1-20 koneksi dengan interval 5. Dengan rata-rata detection rate > 99%, Pada penelitian [10] dilakukan penelitian dan menghitung akurasi dari *feed forward neural networks* adalah 79.49%; Akurasi dari *neural network* adalah 78.1%; akurasi dari *generalized regression neural network* adalah 58.74%; akurasi dari *Probabilistic neural networks* adalah 85.56% dan akurasi dari *radial basic network* adalah 83.51%. Pada penelitian [11] Sistem defuzzifikasi pada metode Centroid, Bisector, SOM, LOM, MOM untuk menguji kekokohan aturan dan rekayasa pengetahuan alat. Pada penilitian [12] dilakukan penelitian dengan metode k-means menggunakan data KDD'99 sebanyak 13% dari total data asli dan mendapatkan tingkat akurasi terbaik sebesar 68.63%.

2. METODE

2. 1. Objek Penelitian

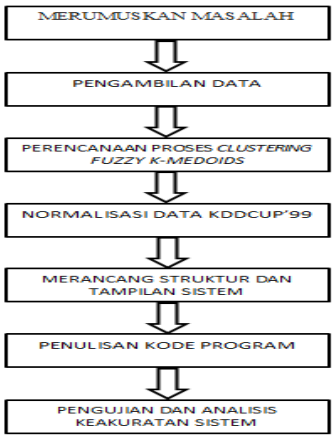
Objek penelitian ini adalah mengambil data yang bersumber dari Portal http://kdd.ics.uci.edu/ yaitu Information and Computer Science, University of California, Irvine. Terdapat sebanyak 40000 baris dan 41 kolom yang disebut atribut *(duration, protocol_type, Service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_Port_rate, dst_host_srv_diff_house_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate, attack_type).*

2.2. Waktu Penilitian

Waktu yang digunakan untuk penelitian ini pada bulan Oktober 2019 sampai dengan Januari 2020

2.3. Tahapan Penelitian

Metode penelitian digunakan sehingga pelaksanaan penelitian dapat dipertanggungjawabkan secara ilmiah. Dengan adanya metode penelitian akan mempermudah peneliti untuk memecahkan masalah yang dihadapi. Pada penelitian ini langkah-langkah yang dilakukan sebagai berikut:



Gambar 2. Tahapan Penelitian

2.4. Preprocessing Data KDD'99

Preprocessing data yang dilakukan menggunakan 2 cara, yang pertama menggunakan bantuan fungsi logaritma natural. Normalisasi data dilakukan dengan memanfaatkanperhitungan logaritma natural terhadap masing-masing data. Menggunakan *software tools* SPSS dalam pengolahanya.

a. Notasi logaritma natural adalah ln *x*.
   ln *x* > 0 ketika *x* > 0
   ln *x* < 0 ketika 0 < *x* > 1
   ln *x* = 0 ketika *x* = 1 …… (1)
   Setelah dilakukan proses *Preprocessing* terhadap data KDD'99 maka dapat mengurangi ketimpangan data dimana terdapat data yang berukuran besar dan kecil dengan range yang cukup jauh. Pada contoh diatas selisih dari data terendah yaitu 103.

b. Uji Z

$$Z = \frac{\bar{x} - x}{\sigma} \qquad \ldots\ldots (2)$$

Keterangan:
$\bar{x}$ = Rata-rata dari sampel
x = rata-rata nilai yang dihipotesiskan
σ = Standar deviasi populasi yang telah diketahui

2.5. Merancang Struktur dan Tampilan Sistem

Tahap ini merancang struktur dan tampilan sistem yang dilakukan untuk memberi gambaran awal mengenai sistem yang dibuat. Tahap pertama yang dilakukan adalah mengumpulkan *dataset* yang terdiri dari data latih dan data uji. Setelah mengumpulkan *dataset*, kemudian dilakukan pembagian atribut. Atribut terbagi menjadi dua, yaitu terdapat fitur dan target. Fitur yaitu nilai digunakan untuk menentukan ciri-ciri, sedangkan target adalah penentuan nilai 4 serangan anomali dan 1 normal yang di dapatkan dengan mempelajari dari fitur yang ada.

Selanjutnya membagi *dataset* menjadi dua yaitu data latih digunakan untuk melatih sistem agar mengenali jenis anomali dan data uji digunakan untuk menguji sistem apakah dapat menentukan jawaban dengan benar dari hasil pembelajaran terhadap data latih.

Setelah melakukan pembagian data, kemudian melakukan *training* model dan *test* model menggunakan metode *clustering* K-Medoids karena mempunyai kemampuan mengelompokkan data dalam jumlah yang cukup besar dengan waktu komputasi yang relatif cepat dan efisien. Model optimisasi Klasifikasi k-Medoids, melibatkan fungsi *dissimilarity* untuk membandingkan kedekatan antara *prototype* dan data masukan. Pada umumnya fungsi *disimilarity* yang digunakan dalam teknik Klasifikasi berbasis *Vector Quantization* (VQ) adalah *norm* Euclidian. Jika sudah melakukan *training* model dan *test* model, maka selanjutnya menghasilkan skor hasil prediksi dari data *test* yang ditentukan.

3. HASIL DAN PEMBAHASAN

3.1. Menghitung Akurasi

Rumus :

$$\textbf{Akurasi} = \frac{\textit{Jumlah Data Yang Benar}}{\textit{Jumlah Data Uji}} \; \boldsymbol{X}\; \mathbf{100}\% \; \ldots (3)$$

Menghitung akurasi dengan cara jumlah data yang benar pada saat pengujian dibagi dengan jumlah data yang diujikan secara keseluruhan dikalikan dengan 100%. Percobaan dilakukan dengan membandingkan 10 variabel dari $0<x\leq 1$ dengan interval 0.1. Jumlah data yang digunakan sebanyak 40000 dibagi menjadi 90% data latih yaitu sebanyak 36000 dan 10% data uji yaitu sebanyak 4000. Hasil percobaan dapat dilihat pada percobaan di bawah ini:

3.2. Menguji Hasil Prediksi Data Uji

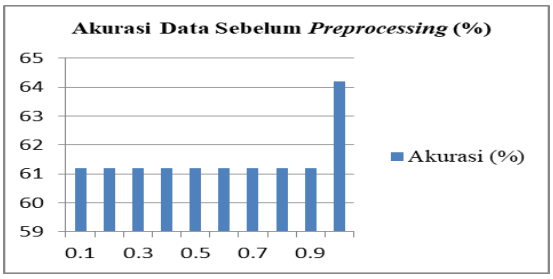
a. Data KDD'99 Sebelum *Preprocessing*

Tabel 4. Hasil Percobaan terhadap data KDD'99 sebelum *preprocessing*

Variabel m	Akurasi (%)
0.1	61.200
0.2	61.200
0.3	61.200
0.4	61.200
0.5	61.200
0.6	61.200

0.7	61.200
0.8	61.200
0.9	61.200
1	64.200

Tabel 4 berisikan informasi perhitungan akurasi seperti rumus di atas dari masing-masing variabel m dengan nilai $0<x\leq1$ dengan interval 0.1. percobaan dilakukan dengan data KDD'99 sebelum *preprocessing*.
Tampilan secara grafik perubahan dari variabel m terhadap data KDD'99 sebelum *preprocessing*



Gambar 3. Grafik Perubahan Tingkat Akurasi dari data KDD'99 sebelum *preprocessing.*

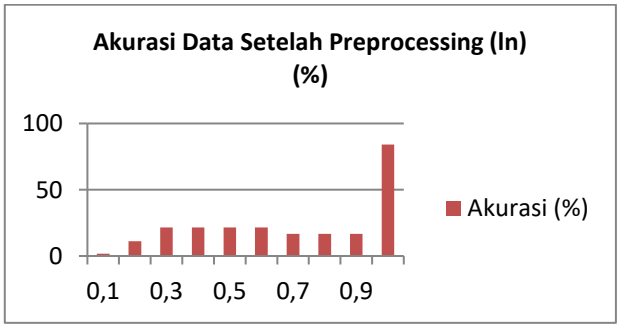
Dari percobaan yang dilakukan dapat diambil kesimpulan bahwa variabel terbaik untuk data KDD'99 sebelum *preprocessing* berada pada m = 1 dengan tingkat akurasi 64.200%.

b. Data KDD'99 Sesudah *Preprocessing (ln)*

Tabel 5. Hasil Percobaan terhadap data KDD'99 setelah *preprocessing* dengan bantuan logaritma natural

Variabel m	Akurasi (%)
0.1	1.775
0.2	11.275
0.3	21.550
0.4	21.550
0.5	21.550
0.6	21.550
0.7	16.725
0.8	16.725
0.9	16.725
1	83.975

Tabel 5 berisikan informasi perhitungan akurasi seperti rumus di atas dari masing-masing variabel m dengan nilai $0<x\leq1$ dengan interval 0.1. percobaan dilakukan dengan data KDD'99 sesudah *preprocessing* dengan logaritma natural untuk setiap datanya.
Tampilan secara grafik perubahan dari variabel m terhadap data KDD'99 yang di *preprocessing* dengan bantuan logaritma natural.



Gambar 4. Grafik Perubahan Tingkat Akurasi dari data KDD'99 setelah *preprocessing* dengan bantuan logaritma natural

Dari percobaan yang dilakukan dapat diambil kesimpulan bahwa variabel terbaik untuk data KDD'99 dengan olah data menggunakan fungsi logaritma natural berada pada m = 1 dengan tingkat akurasi 83.975%.

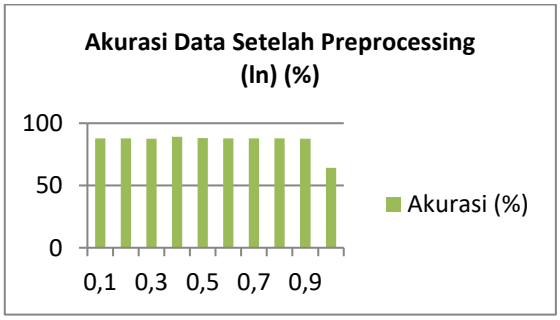
c. Data KDD'99 Sesudah *Preprocessing* (uji Z)

Tabel 5. Hasil Percobaan terhadap data KDD'99 setelah *preprocessing* dengan bantuan perhitungan Z

Variabel m	Akurasi (%)
0.1	87.725
0.2	87.725
0.3	87.525
0.4	89.050
0.5	88.150
0.6	87.775
0.7	87.800
0.8	87.800
0.9	87.650
1	64.200

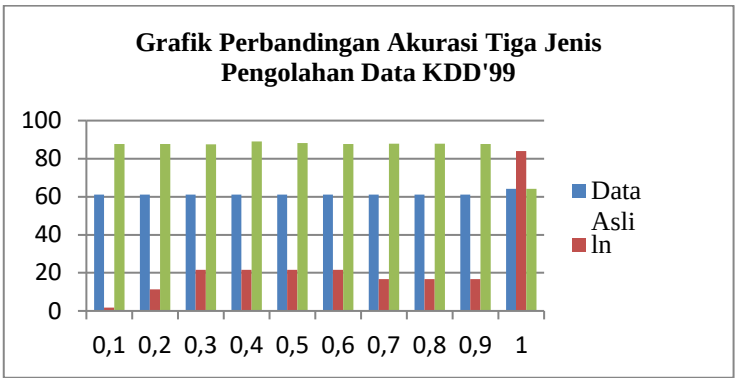
Tabel 6 berisikan informasi perhitungan akurasi seperti rumus di atas dari masing-masing variabel m dengan nilai $0<x\leq1$ dengan interval 0.1. percobaan dilakukan dengan data KDD'99 sesudah *preprocessing* dengan Uji Z untuk setiap datanya.

Tampilan secara grafik perubahan dari variabel m terhadap data KDD'99 yang di *preprocessing* dengan bantuan perhitungan Z.



Gambar 5. Grafik Perubahan Tingkat Akurasi dari data KDD'99 setelah *preprocessing* dengan bantuan perhitungan Z

Dari percobaan yang dilakukan dapat diambil kesimpulan bahwa variabel terbaik untuk data KDD'99 dengan olah data menggunakan bantuan perhitungan Z berada pada m = 0.4 dengan tingkat akurasi 89.050%. Jika disajikan dalam satu grafik hasil dari ketiga pengolahan data di atas adalah sebagai berikut.



Gambar 6. Grafik Perubahan Tingkat Akurasi dari tiga jenis pengolahan data KDD'99

4. PENUTUP

Dari percobaan di atas dapat disimpulkan bahwa hasil terbaik pada penelitian ini adalah pada pengolahan data *preprocessing* KDD'99 dengan Uji Z dimana tingkat akurasi nya adalah 89.050% menggunakan variabel m sebesar 0.4. Variabel terbaik untuk data KDD'99 dengan olah data menggunakan fungsi logaritma natural berada pada m = 1 dengan tingkat akurasi 83.975%. Sedangkan variabel terbaik untuk data KDD'99 sebelum *preprocessing* berada pada m = 1 dengan tingkat akurasi 64.200%. Dapat dilihat perbedaanya pada Gambar 6, bahwa data asli KDD'99 memiliki tingkat akurasi yang paling rendah. Hal ini dapat disebabkan karena data asli KDD'99 masih banyak terdapat selisih data yang jauh antara satu sama lain dengan begitu dibutuhkanya *preprocessing* data. Namun pada penelitian ini belum digunakan metode optimasi parameter. Untuk penelitian selanjutnya disarankan untuk menggunakan nilai parameter lain dari metode *Fuzzy Kmedoids*

DAFTAR PUSTAKA

[1] N. P. K. S. Kadam, S. B. Bagal, Y. S. Thakare and Sonawane, "Canberra Distance Metric Based Hyperline Segment Pattern Classifier Using Hybrid Approach of Fuzzy Logic and Neural Network," 2014, pp. 28–30.

[2] H. Moudni, M. Er-rouidi, H. Mouncif, and B. El Hadadi, "Fuzzy logic based intrusion detection system against black hole attack in mobile ad hoc networks," *Int. J. Commun. Networks Inf. Secur.*, vol. 10, no. 2, pp. 366–373, 2018.

[3] Wu TM, *Intrusion Detection Systems Information Assurance Tools Report Sixth Edition September 25*, Sixth. Information Assurance Technology Analysis Center, 2018.

[4] M. Moorthy and S. Sathiyabama, "Hybrid fuzzy based intrusion detection system for wireless local area networks," *Eur. J. Sci. Res.*, vol. 53, no. 3, pp. 431–446, 2011.

[5] A. Shah, S. Clachar, M. Minimair, and D. Cook, "Building multiclass classification baselines for anomaly-based network intrusion detection systems," 2020, doi: 10.1109/DSAA49011.2020.00102.

[6] C. M. Ou, "Host-based Intrusion Detection Systems Inspired by Machine Learning of Agent-Based Artificial Immune Systems," 2019, doi: 10.1109/INISTA.2019.8778269.

[7] L. Kaufman and P. E. Rousseeuw, "Clustering by means of Medoids," *Statistical Data Analysis Based on the L1 Norm and Related Methods*. 1987.

[8] K. G. Soni and A. Patel, "Comparative Analysis of K-means and K-medoids Algorithm on IRIS Data," 2017.

[9] M. K. Siddiqui and S. Naahid, "Analysis of KDD CUP 99 Dataset using Clustering based Data Mining," *Int. J. Database Theory Appl.*, vol. 6, no. 5, pp. 23–34, 2013, doi: 10.14257/ijdta.2013.6.5.03.

[10] S. Devaraju and S. Ramakrishnan, "Detection of Accuracy for Intrusion Detection System Using Neural Network Classifier," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 1, pp. 338–345, 2013.

[11] B. Bansal, "International Journal of Advanced Research in Computer Science and Software Engineering Rule Based Intrusion Detection System to Identify Attacking Behaviour and Severity of Attacks," vol. 5, no. 1, pp. 718–724, 2015.

[12] A. Suri Talita and E. Prasetyo Wibowo, "Intrusion Detection Systems Data Classification by Possibilistic C-Means Method," *J. Eng. Appl. Sci.*, vol. 15, no. 5, pp. 1170–1174, 2019, doi: 10.36478/jeasci.2020.1170.1174.