Majalah Ilmiah Teknologi Elektro, Vol. 19, No. 1, Januari - Juni 2020
DOI: https://doi.org/10.24843/MITE.2020.v19i01.P14

Analisis Data Log IDS Snort dengan Algoritma Clustering Fuzzy C-Means

Ida Ayu Shinta Dewi Paramitha[1], Gusti Made Arya Sasmita[2], I Made Sunia Raharja[3]

Submission: 28-04-2020, Accepted: 20-06-2020

***Abstract*— Snort is one of open source IDS to detect intrusion or potentially malicious activity on network traffic. Snort will give alert for every detected intrusion and write the alerts in log. Log data in IDS Snort will help network administrator to analyze the vulnerability of network security system. Clustering algorithm such as FCM can be used to analyze the log data of IDS Snort. Implementation of the algorithm is based on Python 3 and aims to cluster alerts in log data into 4 risk categories, such as low, medium, high, and critical. The result of analysis shows that the observed network security system is still vulnerable as IDS Snort records 30% of medium risk attacks. The evaluation with Modified Partition Coefficient (MPC) obtains clustering validity value of 98%.**

***Intisari*— Snort merupakan salah satu sistem deteksi intrusi (IDS) *open source* yang banyak digunakan untuk mendeteksi intrusi atau aktivitas mencurigakan pada lalu lintas jaringan. Snort akan memberikan *alert* atau peringatan apabila terdapat serangan yang terdeteksi, kemudian mencatatnya pada *log*. Data *log* IDS Snort tersebut dapat digunakan oleh administrator jaringan untuk menganalisis kerentanan sistem keamanan jaringan. Analisis data *log* dapat dilakukan dengan berbagai cara, salah satunya mengimplementasikan algoritma *clustering* seperti Fuzzy C-Means (FCM). Implementasi algoritma pada penelitian ini berbasis Python 3 dan bertujuan untuk mengelompokkan serangan pada data *log* menjadi 4 kategori risiko, yaitu *low*, *medium*, *high*, dan *critical risk*. Hasil analisis menunjukkan bahwa sistem keamanan jaringan yang diamati masih memiliki kerentanan, di mana IDS Snort mencatat adanya 30% serangan dengan kategori *medium risk*. Tahap evaluasi dengan *Modified Partition Coefficient (MPC)* memperoleh validitas *clustering* mencapai 98%.**

***Kata Kunci*— *Clustering*, Fuzzy C-Means, *Log*, Snort.**

I. PENDAHULUAN

Keamanan kini masih menjadi salah satu masalah dalam perkembangan teknologi, termasuk pada jaringan komputer. Jaringan komputer sebagai bagian dari suatu sistem harus dilindungi dari berbagai jenis ancaman dan serangan, baik yang bersifat aktif maupun pasif. Adapun ancaman dan serangan yang sering terjadi pada jaringan komputer, antara lain adalah DoS, DDoS, *scanning*, *password cracking*, dan lainnya.

Beberapa cara dapat dilakukan untuk menjaga aspek keamanan pada jaringan, salah satunya menerapkan sistem deteksi intrusi atau IDS. Kemampuan IDS dalam memberikan *alert* atau peringatan ketika terdapat intrusi atau aktivitas mencurigakan dapat membantu administrator dalam mengawasi lalu lintas jaringan. Salah satu IDS yang umum digunakan adalah Snort.

Snort merupakan sistem pencegahan dan deteksi intrusi jaringan bersifat *open source* dengan berbasis aturan (*rule-driven*) yang digunakan untuk memantau lalu lintas jaringan secara pasif dan memberikan peringatan atau *alert* ketika ancaman terdeteksi. Sistem ini secara logika dapat dibagi menjadi beberapa komponen yang bekerja sama untuk mendeteksi serangan tertentu. Snort juga mampu menghasilkan *output* dengan format yang dibutuhkan, seperti data *log* yang mencatat *alert* hasil deteksi [1].

Data *log* IDS Snort ini dapat dimanfaatkan oleh administrator jaringan untuk menganalisis performa sistem keamanan jaringan. Data yang tercatat pada *log* terdiri dari informasi mengenai *alert* serangan yang berhasil terdeteksi oleh Snort, seperti jenis serangan, waktu serangan, alamat dan *port* penyerang, serta alamat dan *port* target penyerang. Snort memberikan dan mencatat *alert* tersebut sesuai dengan *rules* atau aturan yang telah dikonfigurasi.

Analisis data *log* akan membantu administrator jaringan dalam mengetahui jenis serangan yang masih mampu melewati sistem keamanan jaringan sehingga dapat dilakukan tindakan lebih lanjut dalam mengatasi serangan tersebut. Namun, jumlah data yang tersimpan pada *log* Snort umumnya cukup besar sehingga ini dapat menjadi masalah karena administrator jaringan membutuhkan banyak waktu untuk menganalisisnya. Pendekatan *data mining*, seperti klasifikasi dan *clustering*, dapat digunakan untuk menganalisis data dengan jumlah yang besar.

Analisis *log* IDS Snort pernah dilakukan untuk seleksi notifikasi serangan dengan Algoritma K-Means sehingga hanya serangan berbahaya yang akan dikirimkan melalui SMS [2]. Penelitian lainnya melakukan evaluasi K-Means untuk IDPS pada trafik jaringan yang besar dan mendapatkan tingkat akurasi 90% [3]. Implementasi *data mining* pada IDS juga pernah dilakukan dengan Naïve Bayes, SVM, dan Random Forest yang mendapatkan hasil akurasi tertinggi oleh Algoritma Random Forest sebesar 98% [4].

Penelitian ini bertujuan untuk menganalisis data *log* IDS Snort dengan algoritma *clustering* Fuzzy C-Means (FCM). Algoritma FCM dipilih karena menggunakan pemodelan *fuzzy* yang lebih fleksibel dengan melibatkan derajat keanggotaan, memiliki waktu komputasi yang cepat, dan memiliki keunggulan dalam menghasilkan pusat *cluster* yang lebih akurat dibandingkan dengan algoritma lainnya [5],[6]. Hasil

[1]*Mahasiswa, Program Studi Teknologi Informasi Universitas Udayana, Jl. Raya Kampus Udayana, Jimbaran, Badung-Bali, 80361 INDONESIA: 0361-701806; e-mail: 9fshinta27@gmail.com)*
[2,3] *Dosen, Program Studi Teknologi Informasi Universitas Udayana, Jl. Raya Kampus Udayana, Jimbaran, Badung-Bali 80361 INDONESIA (telp: 0361-701806; e-mail: aryasasmita88@gmail.com, sunia.raharja@gmail.com)*

p-ISSN:1693 – 2951; e-ISSN: 2503-2372

clustering berupa jumlah data dan pusat setiap *cluster* akan digunakan untuk perhitungan nilai risiko sehingga didapatkan persentase jumlah serangan dengan kategori *low*, *medium*, *high*, dan *critical risk*. Penelitian ini diharapkan dapat membantu administrator dalam menganalisis kerentanan sistem keamanan jaringan di Institusi X.

II. METODE PENELITIAN

A. Fuzzy C-Means

Fuzzy C-Means (FCM) merupakan algoritma *clustering* dengan logika *fuzzy* yang keberadaan setiap data ditentukan berdasarkan derajat keanggotaannya. Algoritma FCM diperkenalkan pada 1981 oleh Jim Bezdek. *Clustering* dengan FCM berbeda dengan teknik klasik, seperti K-Means, di mana setiap data dapat menjadi anggota dari beberapa *cluster* [7].

Langkah-langkah penyelesaian dengan Algoritma FCM dapat diuraikan sebagai berikut.

1. Masukkan data dengan matriks *X* berukuran *n x m* (*n* = jumlah data, *m* = jumlah atribut).
2. Tentukan jumlah *cluster* (*c*), pangkat (*w*), maksimum perulangan (*max_iter*), *error* terkecil (ξ), fungsi objektif awal (P_0), dan perulangan awal.
3. Bilangan random μ_{ik} dibangkitkan sebagai elemen matriks partisi awal. Jumlah setiap kolom kemudian dihitung dengan (1) dan (2).

$$\boldsymbol{Qi} = \sum_{k=1}^{c} \mu_{ik} \quad (1)$$

$$\mu_{ik} = \frac{\mu_{ik}}{Qi} \quad (2)$$

4. Pusat *cluster* ke-*k*: V_{kj} dihitung dengan (3).

$$\boldsymbol{V_{kj}} = \frac{\sum_{i=1}^{n}((\mu_{ik}^{w} * X_{ij}))}{\sum_{i=1}^{n} \mu_{ik}^{w}} \quad (3)$$

5. Hitung fungsi objektif pada iterasi ke-*t* dalam (4).

$$\boldsymbol{P_t} = \sum_{i=1}^{n}\sum_{k=1}^{c}\left(\left|\sum_{j=1}^{m}(\boldsymbol{X_{ij}} - \boldsymbol{V_{kj}})^2\right|\mu_{ik}^{w}\right) \quad (4)$$

6. Perubahan matriks partisi kemudian dihitung dalam (5).

$$\mu_{ik} = \frac{\left|\sum_{j=1}^{m}(X_{ij} - V_{kj})^2\right|^{\frac{-1}{w-1}}}{\sum_{k=1}^{c}\left|\sum_{j=1}^{m}(X_{ij} - V_{kj})^2\right|^{\frac{-1}{w-1}}} \quad (5)$$

7. Periksa kondisi berhenti:
   - Apabila *(|Pt – Pt-1| < ξ)* atau (*t* > *max_iter*) maka iterasi berhenti;
   - Apabila tidak sesuai dengan kondisi berhenti, maka: *t=t+1* dan ulangi langkah ke-4

B. Partition Coefficient Index

Partition coefficient Index (PCI) digunakan untuk mengukur validasi dari algoritma *clustering* dengan rentang nilai 0-1, di mana 1 merupakan nilai terbaik dan paling optimal. Persamaan (6) digunakan untuk menghitung PCI [8].

$$\boldsymbol{PCI} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\mu_{ij}^{2} \quad (6)$$

N merupakan jumlah data, *C* merupakan jumlah *cluster*, dan μ_{ij} merupakan derajat keanggotaan.

Metode evaluasi ini memiliki kelemahan dengan adanya perubahan yang monoton terhadap beragam jumlah *cluster*. Kelemahan tersebut dapat diatasi dengan *Modified Partition Coefficient* atau MPC yang dapat dihitung dengan (7) [9].

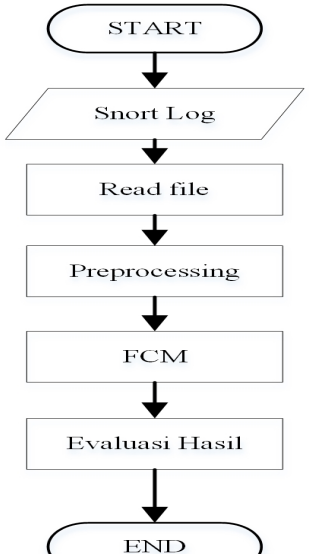
$$\boldsymbol{MPC(c)} = \mathbf{1} - \frac{c}{c-1}(\mathbf{1} - \boldsymbol{PC(c)}) \quad (7)$$

Sama halnya dengan PCI, nilai MPC memiliki rentang 0-1 di mana jumlah *cluster* (nilai *c*) terbesar menunjukkan nilai yang paling optimal. Nilai yang mendekati 0 menunjukkan keakuratan *cluster* yang semakin kabur, sedangkan nilai mendekati 1 menunjukkan semakin baik *cluster* tersebut.

III. METODOLOGI

Tahap pertama pada penelitian ini adalah mengumpulkan data dengan memasang IDS Snort pada jaringan Institusi X selama kurang lebih 1 bulan hingga mendapatkan jumlah data yang cukup. Data dari *log* tersebut kemudian dikonversi ke format csv untuk lebih mudah untuk diproses di tahap analisis.

Gambar 1: menunjukkan diagram alir untuk tahap analisis pada penelitian ini. Secara garis besar, tahap yang dilakukan adalah pengumpulan dan pencatatan data pada *log*, *preprocessing, pengelompokkan* dengan FCM, dan evaluasi.



Gambar 1: Diagram alir analisis

Data dari Snort perlu melalui tahap *preprocessing* untuk menyesuaikan data dengan rentang parameter sehingga dapat diproses dengan algoritma *clustering*. Atribut yang digunakan pada penelitian ini adalah jenis serangan (*classtype*), *priority*, tipe *port*, dan *likelihood*. Penentuan label *critical*, *high*, *medium*, dan *low* dilakukan dengan menghitung *risk assesment* seperti yang dilakukan pada [10]. Tabel 1 menunjukkan parameter untuk menghitung nilai risiko pada penelitian ini:

TABEL I
PARAMETER PENILAIAN RISIKO

Parameter	Deskripsi
Priority	*Priority* menentukan tingkat keparahan (*severity level*) untuk setiap jenis serangan (*classtype*). Snort menyediakan klasifikasi serangan beserta *priority tag*

Parameter	Deskripsi
	dengan rentang tingkat keparahan masing-masing. Penelitian ini menggunakan rentang 1-4, semakin besar nilainya maka *priority* semakin tinggi.
Port type	*Port* terdiri dari 3 tipe, yaitu *well-known*, *registered*, dan *dynamic* [11]. Rentang tipe *port well-known* (0-1023) diberi nilai 3, *registered* (1024-49151) diberi nilai 2, dan *dynamic* (49152-65535) dengan nilai 1. Penelitian ini menambahkan 1 tipe dengan nilai 4 untuk *port* yang paling sering diserang *hacker* [12].
Likelihood	Nilai *likelihood* dapat diberikan berdasarkan frekuensi munculnya suatu *alert* [13]. Parameter ini memiliki rentang 1-5, di mana nilai 1 merupakan nilai untuk *alert* dengan frekuensi rendah dan nilai 5 untuk frekuensi tinggi. Penentuan skala frekuensi menggunakan panduan penilaian risiko dari NIST [14].

Parameter pada Tabel 1 akan digunakan untuk memberi label untuk setiap *cluster* dari algoritma FCM. Persamaan (12) digunakan untuk menghitung nilai risiko (RA) setiap *cluster* sebagai penentu label [15].

Priority (P) = [1-4]
Port Type (D) = [1-4]
Likelihood (L) = [1-5]
Max RA = 10

$$RA = \frac{P * D * R}{X} \quad (8)$$

Nilai risiko (RA) memiliki rentang dengan nilai maksimal 10 sehingga untuk mencari nilai X dengan mengacu pada (9).

$$RA = \frac{\text{Max (P)} * \text{Max (D)} * \text{Max(L)}}{\text{X}} = 10 \quad (9)$$

$$10 = \frac{4 * 4 * 5}{\text{X}} \quad (10)$$

$$X = 8 \quad (11)$$

$$RA = \frac{P * D * R}{8} \quad (12)$$

Hasil perhitungan RA untuk setiap *cluster* tersebut kemudian diberi label *low*, *medium*, *high*, dan *critical* sesuai dengan ketentuan pada Tabel 2.

TABEL II
KATEGORI NILAI RISIKO

Range	Label
0-3.9	Low
4.0-6.9	Medium
7.0-8.9	High
9.0-10.0	Critical

Implementasi FCM dan visualisasi jenis serangan pada penelitian ini menggunakan Python 3. Evaluasi dilakukan dengan perhitungan *Modified Partition Coefficient* (MPC) untuk menentukan validitas hasil *clustering*.

IV. HASIL DAN PEMBAHASAN

Jumlah data pada penelitian ini adalah 346509 data dengan 27 atribut sesuai dengan aturan Snort secara *default*. Potongan *dataset* tersebut dapat dilihat pada Tabel 3. Data yang telah dikumpulkan perlu untuk melalui tahap *preprocessing* untuk menambahkan beberapa atribut sesuai dengan parameter penilaian risiko yang telah ditentukan.

A. Preprocessing

Tahap *preprocessing* diawali dengan menambahkan *classtype* atau klasifikasi jenis serangan untuk setiap data menggunakan *rules* Snort yang telah di-*parsing*. Penambahan atribut *classtype* ini dilakukan untuk menentukan nilai *priority* dengan rentang nilai 1-4. *Priority* untuk setiap *classtype* dapat dilihat pada Snort Manual [16].

Atribut lain yang ditambahkan pada tahap ini adalah *port type* atau tipe *port* sesuai dengan deskripsi pada Tabel 1. Penambahan nilai *likelihood* untuk setiap data didasarkan pada frekuensi setiap *alert* atau *alert rate*, di mana semakin tinggi frekuensinya maka nilai risikonya juga semakin tinggi.

B. Clustering Fuzzy C-Means

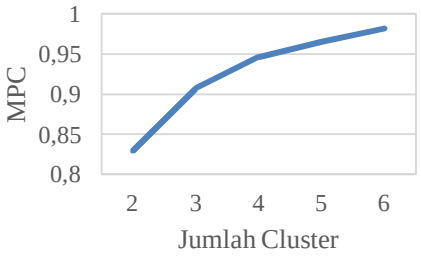
Data yang telah melalui tahap *preprocessing* kemudian dikelompokkan dengan FCM untuk mendapatkan hasil *cluster* dan pusat *cluster*. Hasil pusat *cluster* tersebut yang akan digunakan untuk perhitungan nilai risiko.

Langkah awal implementasi FCM adalah penentuan jumlah *cluster* yang dilakukan dengan rumus MPC untuk mendapatkan jumlah *cluster* terbaik. Penentuan jumlah *cluster* dilakukan dengan rentang 2-6, di mana akan dipilih jumlah *cluster* dengan nilai MPC yang paling mendekati 1.

Gambar 2: menunjukkan bahwa jumlah *cluster* atau *c=6* memiliki nilai MPC yang paling mendekati 1, sehingga *clustering* pada penelitian ini dilakukan dengan 6 *cluster*. Maksimum iterasi yang digunakan adalah 300 dan error terkecil yang diharapkan adalah 0,00001.

TABEL III
POTONGAN DATASET

timestamp	sig_gen	sig_id	sig_rev	msg	proto	src	srcport	dst	dstport
11/27-12:47:01	1	2402000	5371	ET DROP Dshield Block Listed Source group 1	TCP	89.248.168.69	41373	114.5.36.70	6777
11/27-12:48:11	1	2403368	53375	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 35	TCP	45.5.37.140	38156	114.5.36.70	9001
11/27-12:48:23	1	2403450	53375	ET CINS Active Threat Intelligence Poor Reputation IP TCP group 76	TCP	80.82.70.239	52621	114.5.36.70	3677
11/27-12:48:23	1	2402000	5371	ET DROP Dshield Block Listed Source group 1	TCP	80.82.70.239	52621	114.5.36.70	3677
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
01/06-12:12:24	1	2010935	3	ET SCAN Suspicious inbound to MSSQL port 1433	TCP	114.5.230.77	56242	114.5.36.70	1433



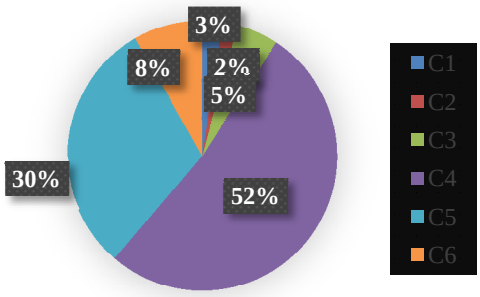
Gambar 2: Perbandingan nilai jumlah *cluster*

Implementasi FCM pada penelitian ini dilakukan dengan Python 3 menggunakan modul dari PyPi [17] dan mendapatkan hasil berupa jumlah data dan pusat setiap *cluster*. Tabel 4 berikut menunjukkan jumlah data pada masing-masing *cluster*:

TABEL IV
JUMLAH DATA SETIAP CLUSTER

Cluster ke -i	Jumlah Data
1	8421
2	6208
3	17245
4	180431
5	105432
6	28772

Gambar 3: menunjukkan visualisasi jumlah data hasil *clustering* FCM dengan *pie chart*. Terlihat bahwa *Cluster* 4 memiliki jumlah data terbesar dan *Cluster* 2 memiliki jumlah data terkecil.



Gambar 3: Hasil *clustering* FCM

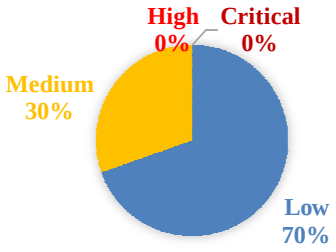
C. Perhitungan Nilai Risiko

Perhitungan nilai risiko dilakukan terhadap pusat *cluster* hasil *clustering* FCM. Pusat *cluster* ini mewakili data yang berada pada setiap *cluster*. Adapun pusat *cluster* dan perhitungan nilai risikonya (RA) dapat dilihat pada Tabel 5:

TABEL V
PERHITUNGAN NILAI RISIKO

Cluster	P	D	L	RA	Kategori
Cluster 1	1.016146	1.772347	1.000000	0.23	Low
Cluster 2	1.006239	3.989993	1.000000	0.5	Low
Cluster 3	2.999490	3.971179	1.000000	1.49	Low
Cluster 4	2.000094	1.000051	2.999332	0.75	Low
Cluster 5	2.999961	3.999668	2.999601	4.5	Medium
Cluster 6	2.998362	1.976388	1.000000	0.74	Low

Hasil perhitungan nilai risiko menunjukkan bahwa data yang berada pada *Cluster* 1,2,3,4, dan 6 merupakan data dengan kategori *low risk*, sedangkan data pada *Cluster* 5 merupakan data dengan kategori *medium risk*. Visualisasi terhadap perhitungan nilai risiko ini dapat dilihat pada Gambar 4:



Gambar 4: Hasil perhitungan nilai risiko

D. Visualisasi Data Log

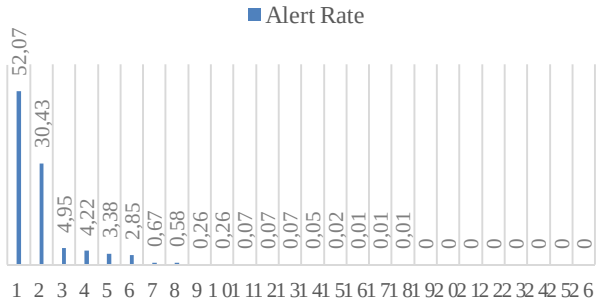
Visualisasi data *log* dilakukan berdasarkan jenis serangan dan *alert rate* yang merupakan frekuensi munculnya *alert* untuk setiap serangan pada data *log* IDS Snort. Adapun jenis serangan dan frekuensinya ditunjukkan pada Tabel 6:

TABEL VI
FREKUENSI ALERT JENIS SERANGAN

No	Jenis Serangan	Freq	Alert Rate
1.	ICMP test detected	180431	52.07
2.	ET SCAN Suspicious inbound to MSSQL port 1433	105432	30.43
3.	ET DROP Dshield Block Listed Source	17135	4.95
4.	(spp_reputation) packets blacklisted	14629	4.22
5.	ET CINS Active Threat Intelligence Poor Reputation IP TCP	11715	3.38
6.	ET SCAN Potential SSH Scan	9890	2.85
7.	ET COMPROMISED Known Compromised or Hostile Host Traffic TCP	2307	0.67
8.	ET CINS Active Threat Intelligence Poor Reputation IP UDP	2025	0.58
9.	ET SCAN Sipvicious Scan	905	0.26
10.	ET SCAN Sipvicious User-Agent Detected (friendly-scanner)	899	0.26
11.	ET DROP Spamhaus DROP Listed Traffic Inbound	258	0.07
12.	ET SCAN Suspicious inbound to mySQL port 3306	245	0.07
13.	ET SCAN SipCLI VOIP Scan	231	0.07
14.	ET SCAN Suspicious inbound to PostgreSQL port 5432	172	0.05
15.	ET SCAN Suspicious inbound to Oracle SQL port 1521	68	0.02
16.	ET TOR Known Tor Relay/Router (Not Exit) Node TCP Traffic	41	0.01
17.	ET INFO UPnP Discovery Search Response vulnerable UPnP device 2	37	0.01
18.	ET SCAN NMAP OS Detection Probe	27	0.01
19.	ET SCAN Suspicious inbound to mSQL port 4333	17	0
20.	ET VOIP Modified Sipvicious Asterisk PBX User-Agent	16	0
21.	ET COMPROMISED Known Compromised or	13	0

No	Jenis Serangan	Freq	Alert Rate
	Hostile Host Traffic UDP		
22.	ET SCAN HID VertX and Edge door controllers discover	9	0
23.	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection	2	0
24.	ET SCAN Potential VNC Scan 5900-5920	2	0
25.	ET VOIP REGISTER Message Flood UDP	2	0
26.	ET DOS Possible NTP DDoS Inbound Frequent Un-Authed MON_LIST Requests IMPL 0x03	1	0

Gambar 5: menunjukkan visualisasi terhadap jenis serangan yang tercatat pada data *log* sesuai dengan Tabel 6. Terdapat 26 jenis serangan berbeda dengan *alert rate* tertinggi adalah "ICMP Test Detected" dengan klasifikasi *icmp-event*.



Gambar 5: Visualisasi data *log*

E. Evaluasi Algoritma FCM

Evaluasi dilakukan dengan menghitung validitas *clustering* menggunakan persamaan MPC terhadap 6 *cluster*, di mana didapatkan nilai 0,98185802. MPC dihitung berdasarkan derajat keanggotaan setiap data pada hasil *clustering* iterasi terakhir. Perhitungan ini dilakukan melalui Python 3 dengan mengacu pada (7). Nilai ini menunjukkan bahwa tingkat validitas Fuzzy C-Means untuk analisis data *log* IDS Snort adalah 98%.

V. KESIMPULAN

Berdasarkan hasil pengujian dan pembahasan yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut.

1. Jenis serangan dengan frekuensi tertinggi yang tercatat pada data *log* IDS Snort adalah "ICMP Test Detected" yang termasuk klasifikasi *icmp-event*.
2. Perhitungan nilai risiko terhadap hasil *clustering* menemukan bahwa jumlah serangan terbesar berada pada kategori *low risk*, yaitu sebanyak 70%. Serangan dengan kategori *medium risk* mencapai 30% dan tidak ada serangan dengan kategori *high* maupun *critical*.
3. Evaluasi dengan MPC menunjukkan bahwa Algoritma FCM dapat digunakan untuk analisis data *log* IDS Snort dengan tingkat validitas sebesar 98%.
4. Hasil analisis data *log* menunjukkan bahwa sistem keamanan jaringan pada Institusi X masih perlu untuk terus dipantau mengingat adanya serangan tingkat medium yang terdeteksi dengan frekuensi yang cukup banyak.

REFERENSI

[1] G. D. Kurundkar, N. A. Naik, and S. Khamitkar, "Network Intrusion Detection using SNORT," *Int. J. Eng. Res. Appl.*, vol. vol.2, no. Issue 2, pp. 1288–1296, 2012.

[2] A. Y. Ananta, "Seleksi Notifikasi Serangan Berbasis Ids Snort Menggunakan Metode K-Means," *SMARTICS J.*, vol. 3, no. 2, pp. 31–37, 2017, doi: 10.21067/smartics.v3i2.1954.

[3] K. Nalavade and B. B. Meshram, "Evaluation of K-Means Clustering for Effective Intrusion Detection and Prevention in Massive Network Traffic Data," *Int. J. Comput. Appl.*, vol. 96, no. 7, pp. 9–14, 2014, doi: 10.5120/16804-6526.

[4] D. Mongkareng, N. A. Setiawan, and A. E. Permanasari, "Implementasi Data Mining dengan Seleksi Fitur untuk Klasifikasi Serangan pada Intrusion Detection System (IDS)," *Citee*, no. gambar 2, pp. 314–321, 2017.

[5] R. Hadi, I. K. G. D. Putra, and I. N. S. Kumara, "Penentuan Kompetensi Mahasiswa dengan Algoritma Genetik dan Metode Fuzzy C-Means," *Maj. Ilm. Teknol. Elektro*, vol. 15, no. 2, pp. 101–106, 2016, doi: 10.24843/mite.1502.15.

[6] N. Nidyashofa and Deden Istiawan, "Penerapan Algoritma Fuzzy C-Means untuk Pengelompokan Kabupaten / Kota di Jawa Tengah Berdasarkan Status Kesejahteraan Tahun 2015," *6th Univ. Res. Colloq.*, no. November, pp. 23–30, 2017.

[7] U. M. Pak and B. Darmajaya, "Penentuan Penerima Beasiswa Dengan Algoritma Fuzzy C-Means Di Universitas Megow Pak Tulang Bawang," *J. Teknol. Inf. Magister Darmajaya*, vol. 1, no. 02, pp. 158–174, 2015.

[8] M. T. A. C. Widiyanto, "Perbandingan Validitas Fuzzy Clustering pada Fuzzy C-Means Dan Particle Swarms Optimazation (PSO) pada Pengelompokan Kelas," *JISKa*, vol. 4, no. 1, pp. 22–37, 2019.

[9] M. M. A. Amirah, A. W. Widodo, and C. Dewi, "Pengelompokan Lagu Berdasarkan Emosi Menggunakan Algoritma Fuzzy C-Means," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 12, pp. 1526–1534, 2017.

[10] C. El Mostapha, M. Moughit, and Y. I. Khamlichi, "Building an efficient alert management model for intrusion detection systems," *Adv. Sci. Technol. Eng. Syst.*, vol. 3, no. 1, pp. 18–24, 2018, doi: 10.25046/aj030103.

[11] W. Goralski, "User Datagram Protocol," *Illus. Netw.*, pp. 289–306, 2017, doi: 10.1016/b978-0-12-811027-0.00011-4.

[12] K. H. Mchatta, "Ethical Hacking for Beginners (Tools , Enumeration and Exploitation)," no. July, 2019.

[13] S. G. Kassa, "IT Asset Valuation, Risk Assessment and Control Implementation Model," *ISACA*, vol. 3, pp. 1–9, 2017.

[14] NIST, "NIST Special Publication 800-30 Revision 1 - Guide for Conducting Risk Assessments," *NIST Spec. Publ.*, no. September, p. 95, 2012, doi: 10.6028/NIST.SP.800-30r1.

[15] E. M. Chakir, M. Moughit, and Y. I. Khamlichi, "An Effecient Method for Evaluating Alerts of Intrusion Detection Systems National School of Applied Sciences USMBA," *2017 Int. Conf. Wirel. Technol. Embed. Intell. Syst.*, pp. 1–6, 2017, doi: 10.1109/WITS.2017.7934678.

[16] M. Roesch, "SNORT Users Manual 2.9.13 The Snort Project," p. 269, 2019.

[17] M. L. D. Dias, "fuzzy-c-means: An implementation of Fuzzy C-means clustering algorithm." 2019.

{Halaman ini sengaja dikosongkan }