|  |  |  |
| --- | --- | --- |
|  | Available online to [www.journal.unipdu.ac.id](http://www.journal.unipdu.ac.id)    **S2-Accredited** – [SK No. 34/E/KPT/2018](http://arjuna.ristekdikti.go.id/index.php/news/view/138)  Journal Page is available to [www.journal.unipdu.ac.id:8080/index.php/register](http://www.journal.unipdu.ac.id:8080/index.php/register) |  |

Advanced Detection Denial of Service Attack in Internet Of Things Network Based On MQTT Protocol Using Fuzzy Logic

--

--

-

|  |  |  |
| --- | --- | --- |
| A R T I C L E I N F O |  | A B S T R A C T |
| Article history:  Received xxx Revised xxx Accepted xxx Available online xxx | Message Queuing Telemetry Transport (MQTT) is one of the popular protocols used on the Internet of Things (IoT) networks because of its lightweight nature. With the increasing number of devices connected to the internet, the number of cybercrimes on IoT networks will increase. One of the most popular attacks is the Denial of Service (DoS) attack. Standard security on MQTT uses SSL / TLS, but SSL / TLS is not capable of handling DoS. The use of fuzzy logic algorithms with the Intrusion Detection System (IDS) scheme is suitable for detecting DoS because of its simple nature. This paper uses a fuzzy logic algorithm embedded in a node to detect DoS in the MQTT protocol with feature selection nodes. The contribution of this paper is that the nodes feature selection used will monitor subscribe and suback traffic and provide this information to fuzzy input nodes to detect DoS attacks. Fuzzy performance evaluation is measured against changes in the number of nodes and attack intervals. The results obtained are that the more the number of nodes and the higher the traffic intensity, the fuzzy performance will decrease, and vice versa. However, the number of nodes and traffic intensity will affect fuzzy performance. |
| Keywords: [ Key word heading]  fuzzy  message queuing telemetry transport (MQTT)  internet of things (IoT)  denial of service (DoS) |
| **IEEE style in citing this article: [citation Heading]**  M. S. Budiana, R. M. Negara, and A. I. Irawan, "Advanced detection denial of service attack in internet of things network based on MQTT protocol using fuzzy logic," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. xxx, no. xxx, pp. xxx, 2021. |
| 2021 Register: *Jurnal Ilmiah Teknologi Sistem Informasi* (Scientific Journal of Information System Technology) with CC BY NC SA license. | | |

1. Introduction

Jaringan Internet of Things (IoT) atau komunikasi Machine-to-Machine (M2M) sudah menjadi bagian penting di era sekarang [1]. IoT mampu mengkomunikasikan things atau objek (seperti perangkat, mobil, atau sensor) satu sama lain melalui media wireline ataupun wireless ke jaringan internet. Tujuan utama dari IoT yaitu untuk menghubungkan seluruh entitas satu sama lain dimana saja dan kapan saja dengan apa saja dan siapa saja [2]. Entitas pada jaringan IoT harus memiliki resource (seperti memory, storage, ataupun daya) yang terbatas [3]. Pada umumnya perangkat IoT memiliki spesifikasi 8-bit mikrokontroler dengan 20kB RAM dan 100kB ROM [2]. Agar perangkat mampu terhubung dengan jaringan internet, diperlukan pengalamatan menggunakan IP, namun penggunaan IP pada perangkat IoT tidak praktis. Akhirnya para peneliti memberikan solusi pengalamatan untuk perangkat IoT menggunakan IPv6 over Low power Wireless Personal Protocol (6LoWPAN) [4]. Beberapa protokol lainnya yang mendukung jaringan IoT ialah Contrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), dan Message Queuing Telemtry Protocol (MQTT) [1][5][6].

Protokol MQTT memiliki karakteristik yang ringan dan sederhana sehingga menjadi salah satu kandidat terbaik untuk digunakan pada jaringan dengan entitas yang bersifat constrained, low-bandwidth, high-latency, ataupun unriable network [7]. MQTT menggunakan pola komunikasi publish/subscribe [8]. Pada dasarnya komunikasi publish/subscribe memiliki tiga elemen utama, yaitu publisher, subscriber, dan broker. Publisher (contohnya ialah perangkat sensor, PC, smartphone) yang menyediakan data dan mem-publish topic ke broker. Subscriber (contohnya ialah aplikasi atau perangkat) sebagai pihak yang meminta topic dari broker. Broker berperan sebagai server dan bertanggung jawab terhadap pertukaran topic antara publisher dengan subscriber [9]. Karakteristik MQTT yang ringan dan sederhana menimbulkan permasalahan pada sisi keamanannya [10] sehingga MQTT menggunakan SSL/TLS sebagai kemanan standarnya guna mencegah terjadinya serangan eavesdropping atau mencegah kerusakan data. Bagaimanapun juga SSL/TLS tidak dapat menangani serangan DoS [11]. Salah satu cara untuk menghindari serangan DoS ialah dengan menggunakan Intrusion Detection System (IDS) [9]. Implementasi IDS dapat dilakukan menggunakan beberapa algoritma berikut, antara lain fuzzy logic [1], machine learning [8], deep learning [9], atau blockchain [10]. Penerapan algoritma fuzzy logic pada protokol MQTT merupakan pilihan yang sesuai karena memiliki beberapa keunggulan. Pertama, memiliki pendekatan yang baik untuk decision making problem. Kedua, implementasi yang mudah dan sederhana [11].

Algoritma fuzzy logic ditemukan oleh Lotfy Zadeh pada tahun 1965 [12][13], yang mana pernah diteliti untuk pertama kalinya oleh Lukasiewicz pada tahun 1920 dengan sebutan many-valued logic [14]. Fuzzy logic merupakan pengembangan dari logika boolean [15] yang mana fuzzy logic didasari oleh “penalaran manusia” dengan tujuan untuk mendapatkan output yang mendekati “true” [16]. Pendekatan menggunakan fuzzy logic akan memberikan output yang sangat luas daripada menggunakan logika boolean [17]. Pada logika boolean, hanya terdapat dua nilai kebenaran, yaitu “true” (biasanya 1) atau “false” (biasanya 0), sehingga output yang didapat adalah “completely true” atau “completely false”. Berbeda dengan fuzzy logic yang mana memiliki derajat atau degree dari 0 sampai sama dengan 1. Fuzzy logic akan memberikan output berupa “partially true” atau “partially false” [15]. Terdapat empat komponen utama dalam algoritma fuzzy logic, yaitu fuzzification, fuzzy rules, fuzzy inference engine, dan defuzzification [18]. Data input harus melewati blok fuzzifier agar data input tersebut dapat dikonversi dari bentuk numerik menjadi bentuk linguistik. Setelah tahap fuzzification, fuzzy inference engine atau Fuzzy Inference System (FIS) akan menetapkan output berdasarkan informasi linguistik. Tahap ini menggunakan pendekatan berdasarkan interpretasi manusia. Tahap terakhir ialah blok defuzzification dimana fuzzy output akan dikonversi dan ditranslasikan menjadi output yang berupa keputusan yang akan dilakukan [16][19][20].

Paper ini akan membahas mengenai kinerja dari algortima fuzzy logic saat mendeteksi serangan DoS pada jaringan IoT berbasis protokol MQTT. Algoritma fuzzy logic akan ditanam pada salah satu node jaringan sebagai subscriber dan akan mendeteksi serangan DoS dengan bantuan nodes feature selection.

1. Research Method

Keterbatasan dan kesederhanaan MQTT menimbulkan masalah pada sistem keamanannya . Hal ini menyebabkan MQTT rentan terhadap berbagai macam serangan [10]. Terdapat berbagai macam keamanan yang dapat diimplementasikan pada MQTT, tergantung dari jenis serangannya. Haripriya et al. dalam paper [1] menggunakan fuzzy logic-basedIDS untuk mengidentifikasi anomali jaringan terhadap serangan DoS berdasarkan trafik PUBLISH dan SUBSCRIBE dari tiap node client. Kedua trafik tersebut akan dijadikan sebagai variabel input oleh fuzzy. Fuzzy akan menghitung derajat kedua variabel tersebut dan akan mengeksekusi logika “IF … THEN …” menggunakan fuzzy inference system (FIS), dimana metode FIS yang digunakan adalah metode Mamdani. Selain itu, penelitian ini menggunakan fuzzy rule interpolation untuk menghasilkan rule baru berdasarkan behavior dari alir trafik jaringan. Penelitian ini akan menghasilkan sistem yang mampu mendeteksi serangan DoS pada protokol MQTT.

Harsha et al. dalam paper [6] membahas tentang identifikasi celah keamanan pada MQTT menggunakan Shodan API berdasarkan paket MQTT dan QoS level. Paper ini menunjukkan bahwa sebagian besar pengguna MQTT tidak menggunakan mekanisme autentikasi dan otoritas data, sehingga jaringan rentan terhadap sniffing atau modifikasi data. Pencegahan yang diusulkan oleh Harsha et al. untuk meningkatkan autentikasi pada MQTT antara lain adalah dengan menggunakan username dan password antara klien dengan broker, mengimplementasikan plain text pada pesan CONNECT, menggunakan jalur pribadi (TLS), atau menggunakan enkripsi pada klien dan dekripsi pada broker. Sedangkan pencegahan untuk meningkatkan otoritas data adalah dengan menggunakan ACL per topik, per method, atau per QoS.

Andy et al. dalam paper [7] membahas mengenai penanganan serangan sniffing, modifikasi data, dan Botnet pada MQTT. Andy et al. mengusulkan untuk menggunakan TLS, ECC, ataupun ECC dan RSA untuk menangani serangan sniffing dan modifikasi data pada jaringan private. Sedangkan untuk menangani serangan Botnet (DoS atau phising), Andy et al. mengusulkan untuk meletakkan MQTT broker pada jaringan publik. Potrino et al. dalam paper [9] mengusulkan Host-based IDS (HIDS) yang terpasang pada fog node untuk menangani SYN flooding attacks, CONNECT flooding attacks, High QoS messages attack, dan DoS menggunakan PUBLISH attacks pada MQTT. HIDS akan mengidentifikasi dan memvalidasi panjang buffer dan QoS level pada pesan CONNECT dan PUBLISH yang datang dari tiap klien. Ramos et al. dalam paper [21] mengusulkan framework menggunakan teknik template-based fuzzing untuk mendeteksi serangan DoS pada MQTT.

Penelitian ini bertujuan untuk mendeteksi serangan DoS pada protokol MQTT menggunakan algoritma fuzzy logic. Variabel SUBSCRIBE dan SUBACK akan dijadikan sebagai variabel input pada fuzzy logic. Fuzzy logic akan diletakkan pada node jaringan dan akan memantau alir trafik MQTT dengan bantuan Feature Selection nodes.

1. Research Method
   1. System Design

Algoritma fuzzy logic akan ditanam pada salah satu node yang dinamakan node fuzzy. Selain itu diperlukan lima nodes tambahan sebagai node pendukung untuk mengamati alir trafik MQTT pada jaringan. Gambar 1 menunjukkan diagram blok sistem keseluruhan. Nodes feature selection akan memantau alir trafik MQTT secara real-time dimana informasi alir tersebut diperoleh dari broker menggunakan topik broker status. Selanjutnya nodes feature selection akan mengirim kembali semua informasi alir trafik MQTT ke broker menggunakan topik baru sehingga node fuzzy dapat mengetahui informasi mengenai alir trafik MQTT menggunakan topik yang telah dibuat oleh nodes feature selection. Lalu node fuzzy akan memilah alir trafik MQTT untuk dijadikan sebagai variabel input pada algoritma fuzzy logic dimana trafik yang dipilah adalah SUBSCRIBE dan SUBACK menggunakan persamaan 1 dan 2. Node fuzzy akan menghitung rasio trafik SUBSCRIBE dan SUBACK menjadi Subscribe Message Ratio (SMR) menggunakan persamaan 3 dan Subscribe Acknowledgement Message Ratio (SAMR) menggunakan persamaan 4.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |
|  | (4) |

Nilai SMR dan SAMR akan dimaksukkan ke dalam algoritma fuzzy logic dan akan dikonversi menjadi variabel linguistik. Selanjutnya algoritma fuzzy logic akan menyesuaikan variabel linguistik SMR dan SAMR dengan rule bases yang telah dibuat menggunakan logika “IF ... AND ... THEN ...”. Hasil penyesuaian SMR dan SAMR dengan rule bases akan menghasilkan data linguistik tentang kondisi jaringan terhadap serangan DoS.

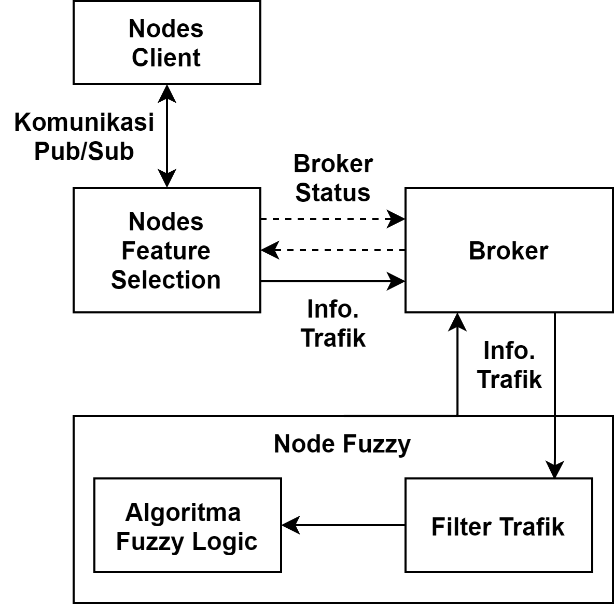


Figure 1. System design

* 1. Feature Selection (FS) Nodes

Nodes Feature Selection (FS) bertugas untuk mengamati alir trafik MQTT pada jaringan. Gambar 2 menunjukkan diagram blok dari nodes FS. Nodes FS akan melakukan subscribe ke broker mengenai informasi alir trafik menggunakan topik broker status yaitu $SYS/broker/# sehingga broker akan melakukan publish informasi yang diperlukan secara real-time. Saat nodes FS memperoleh informasi dari broker, nodes FS akan melakukan publish informasi alir trafik ke broker menggunakan topik baru, yaitu fuzzy/dtc/#, dan broker akan menyimpan topik tersebut. Jumlah nodes FS yang diperlukan ialah sebanyak lima nodes. Hal ini dikarenakan oleh karakteristik perangkat yang sederhana sehingga tiap nodes tidak dapat melakukan subscribe lebih dari satu topik.

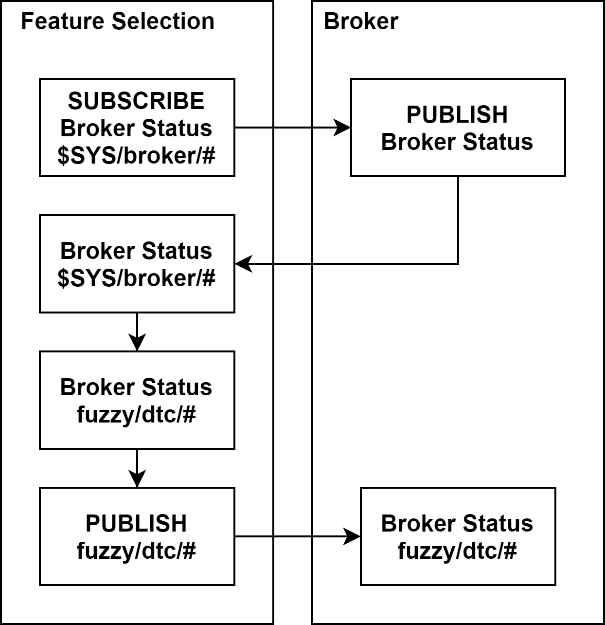


Figure 2. FS nodes model

* 1. Fuzzy Node

Node fuzzy berfungsi untuk mendeteksi serangan DoS. Gambar 3 menunjukkan diagram blok dari node fuzzy. Node fuzzy akan melakukan subscribe ke broker menggunakan topik yang telah dibuat oleh node FS yaitu fuzzy/dtc/# sehingga node fuzzy dapat mengetahui informasi mengenai alir trafik MQTT pada jaringan. Setelah broker menerima subcribe dari node fuzzy, broker akan melakukan publish topik fuzzy/dtc/# secara real-time. Apabila informasi alir trafik MQTT dari broker sudah diterima, maka node fuzzy akan memilah trafik SUBSCRIBE dan SUBACK menggunakan persamaan 1 dan 2 untuk dijadikan sebagai varibel input pada algoritma fuzzy logic. Node fuzzy akan menghitung rasio trafik SUBSCRIBE dan SUBACK menjadi SMR dan SAMR menggunakan persamaan 3 dan 4.

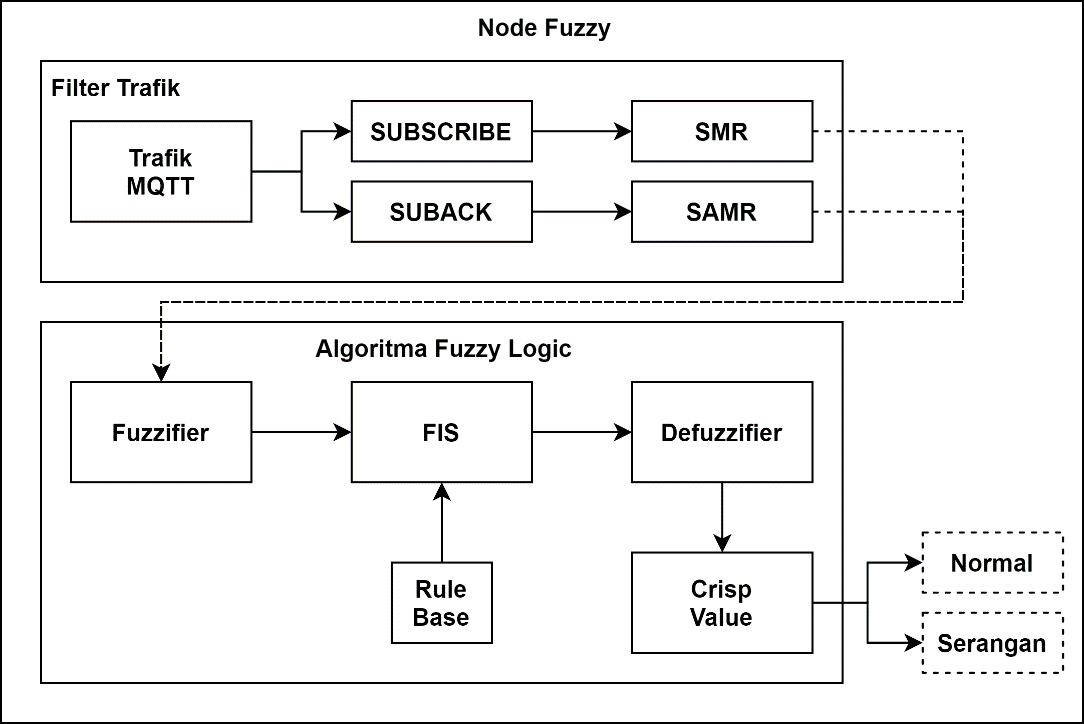


Figure 3. Fuzzy node model

Nilai SMR dan SAMR akan diteruskan ke blok fuzzifier dimana blok ini akan menghitung derajat keanggotaan atau membership function (MF) dari SMR dan SAMR. Setiap variabel akan dikelompokkan menjadi variabel linguistik berdasarkan derajat MF-nya masing-masing menggunakan proses fuzzification. Terdapat tiga macam variabel linguistik pada SMR dan SAMR, antara lain “RENDAH”, “SEDANG”, dan “TINGGI”. MF yang dihasilkan oleh fuzzififier akan diberikan kepada blok FIS. Pada tahap ini setiap MF akan disesuaikan dengan rule bases yang sudah dibuat dengan menggunakan logika “IF ... AND ... THEN ...”. Tabel 1 menunjukkan rancangan rule bases yang digunakan. Penyesesuaian MF dengan rule bases akan menghasilkan kesimpulan berupa variabel linguistik yang mengindikasikan kondisi jaringan terhadap serangan DoS. Terdapat tiga macam kesimpulan yang dihasilkan, antara lain “NORMAL”, “ABNORMAL”, dan “SERANGAN”. Kesimpulan yang diperoleh akan dikonversi menjadi crisp value oleh blok defuzzifier menggunakan proses defuzzification.

Table 1. Rule Bases

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SMR** | R | R | R | S | S | S | T | T | T |
| **SAMR** | R | S | T | R | S | T | R | S | T |
| **Output** | N | A | S | N | N | A | S | S | S |

* 1. Experiment Setup

Metode *fuzzy control* atau FIS yang digunakan ialah *Mamdani* dengan metode *defuzzification* yang digunakan ialah *centroid*. Simulasi dilakukan menggunakan simulator jaringan COOJA dengan *Mosquitto* sebagai *platform broker*-nya. Versi *MQTT* yang digunakan adalah MQTT *v.3.1* dengan MQTTQoS *level* 0. Simulasi dilakukan menggunakan skenario yang bervariasi, yaitu 15, 20, 25, dan 30 *nodes* dengan masing-masing skenario menggunakan interval serangan 3, 5, 7, dan 9 detik. Jumlah *nodes* penyerang berjumlah 20% dari jumlah *nodes* keseluruhan. Penyerang akan membanjiri jaringan (*flooding attack*) menggunakan trafik *SUBSCRIBE*.

1. Results and Analysis
   1. Membership Function (MF) of Fuzzy Logic Algorithm

Tabel 2 menunjukkan nilai SMR yang diperoleh menggunakan persamaan 3. Tabel 3 menunjukkan nilai SAMR yang diperoleh menggunakan persamaan 4. Berdasarkan tabel 2 dan 3, nilai SMR dan SAMR akan semakin meningkat seiring dengan bertambahnya jumlah *nodes* ataupun meningkatnya intensitas serangan. Hal ini disebabkan karena semakin banyak jumlah *nodes*, maka trafik pada jaringan akan semakin meningkat. Nilai SMR tertinggi diperoleh pada skenario 25 *nodes* saat interval serangan *ATK 3S* dengan nilai 0,6308; sedangkan nilai SMR terendah diperoleh pada skenario 20 *nodes* saat tidak terjadi serangan dengan nilai 0,0541. Nilai SAMR tertinggi diperoleh saat interval serangan *ATK 3S* pada 15 *nodes* dengan nilai 0,3848; sedangkan nilai SAMR terendah diperoleh saat tidak terjadi serangan pada 20 *nodes* dengan nilai 0,043.

Table 2. SMR value per scenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenario** | **15 Nodes** | **20 Nodes** | **25 Nodes** | **30 Nodes** |
| **ATK 3S** | 0,5989 | 0,5582 | 0,6308 | 0,5982 |
| **ATK 5S** | 0,4491 | 0,4961 | 0,5740 | 0,5636 |
| **ATK 7S** | 0,4131 | 0,4293 | 0,4652 | 0,4707 |
| **ATK 9S** | 0,3628 | 0,3413 | 0,4405 | 0,4078 |
| **No ATK** | 0,0724 | 0,0541 | 0,1822 | 0,1092 |

Table 3. SAMR value per scenario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Skenario** | **15 Nodes** | **20 Nodes** | **25 Nodes** | **30 Nodes** |
| **ATK 3S** | 0,3848 | 0,3455 | 0,3200 | 0,2410 |
| **ATK 5S** | 0,2951 | 0,2217 | 0,2896 | 0,1421 |
| **ATK 7S** | 0,2446 | 0,1351 | 0,2751 | 0,1988 |
| **ATK 9S** | 0,2698 | 0,1620 | 0,1918 | 0,1233 |
| **No ATK** | 0,0740 | 0,0430 | 0,0816 | 0,1654 |

Gambar 4 dan 5 menunjukkan grafik nilai rata-rata SMR dan SAMR. Hal ini dilakukan agar MF SMR maupun SAMR dapat mendeteksi serangan DoS. Nilai rata-rata SMR tertinggi terjadi pada skenario 25 *nodes* dengan nilai 0,5276; sedangkan nilai rata-rata SMR terendah terjadi pada skenario 20 *nodes* dengan nilai 0,0541. Nilai rata-rata SAMR tertinggi terjadi pada skenario 15 *nodes* dengan nilai 0,2986; sedangkan nilai rata-rata SAMR terendah terjadi pada skenario 20 *nodes* dengan nilai 0,043. Berdasarkan perolehan nilai yang ditunjukkan oleh gambar 4 dan 5, bentuk MF SMR dan SAMR yang digunakan untuk mendeteksi serangan DoS ditunjukkan oleh gambar 6 dan 7.

Figure 4. Average of SMR value

Figure 5. Average of SAMR value

Penentuan bentuk MF *output* diperoleh berdasarkan hasil percobaan dengan menggunakan *rule bases* yang ditunjukkan pada tabel 1. Terdapat tiga *MF* *output* untuk mengindikasikan kondisi jaringan, diantaranya “NORMAL”, “ABNORMAL”, dan “SERANGAN”. “R”, “S”, dan “T” pada variabel *SMR* dan *SAMR* merupakan “RENDAH”, “SEDANG”, dan “TINGGI”. “N”, “A”, dan “S” pada variabel *output* merupakan “NORMAL”, “ABNORMAL”, dan “SERANGAN”. *Crisp value* diperoleh menggunakan proses *defuzzification* dimana pada penelitian ini metode *defuzzification* yang digunakan adalah metode *centroid* yang artinya nilai *crisp* akan diperoleh berdasarkan titik tengah dari *MF output* yang dihasilkan. Untuk menentukan kondisi jaringan (dalam keadaan tidak terjadi serangan atau sedang terjadi serangan), *centroid* dari MF“*ABNORMAL*”dijadikan sebagai nilai acuan dimana titik tengah MF“ABNORMAL” bernilai 0,375. Gambar 8 merupakan MF dari variabel *output* yang menunjukkan kondisi jaringan dimana jika *crisp value* bernilai antara nol sampai kurang dari nilai acuan (0 ≤ *crisp* < 0,375), diasumsikan bahwa tidak terjadi serangan *DoS* pada jaringan. Apabila *crisp value* bernilai lebih besar dari sama dengan nilai acuan sampai kurang dari sama dengan satu (0,375 ≤ *crisp* ≤ 1), diasumsikan bahwa sedang terjadi serangan *DoS* pada jaringan.

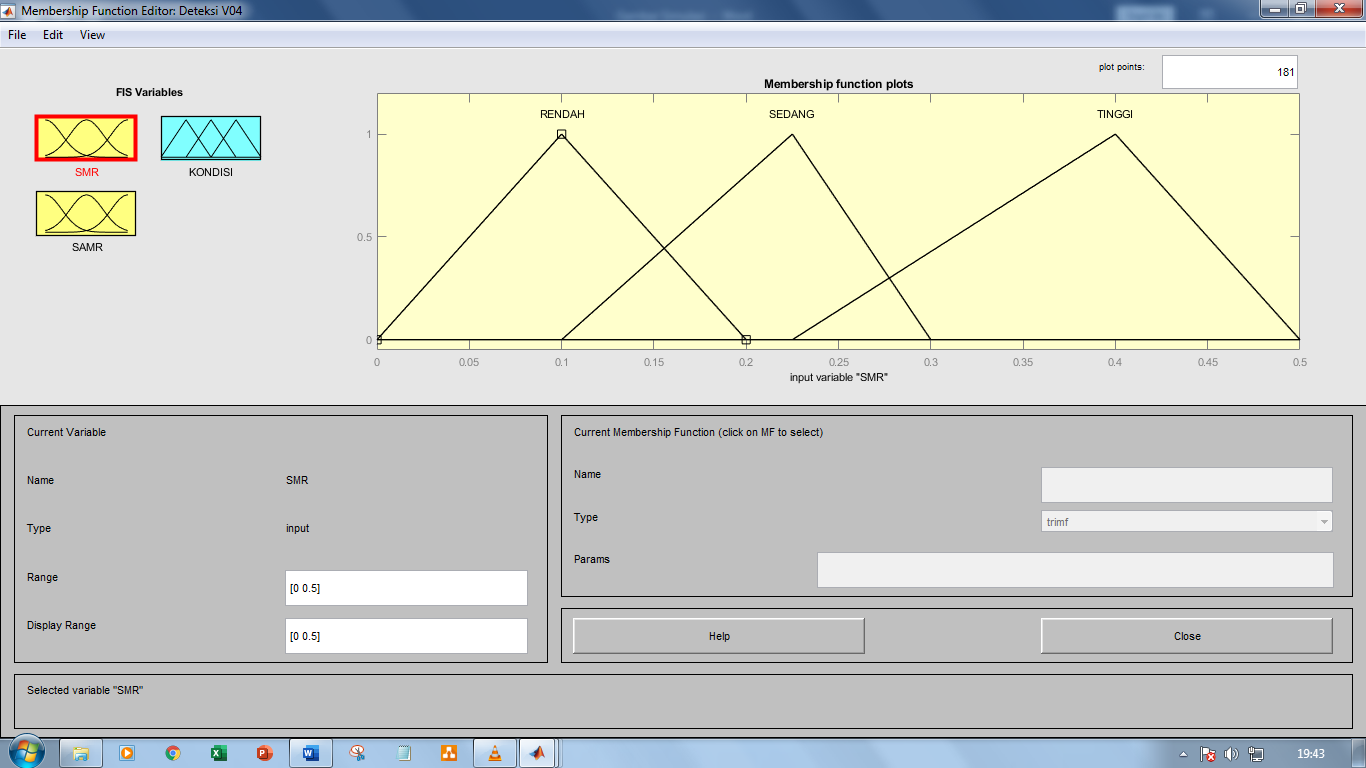


Figure 6. SMR membership function

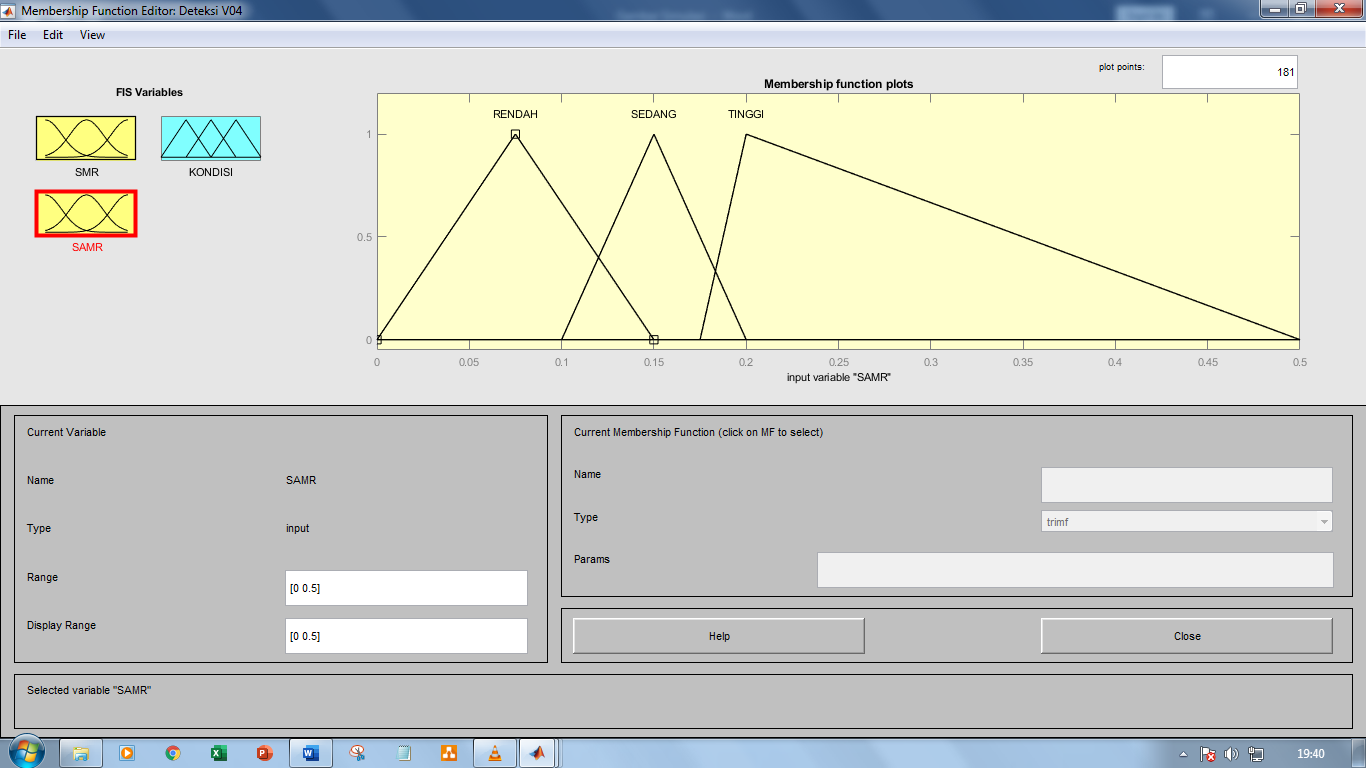


Figure 7. SAMR membership function

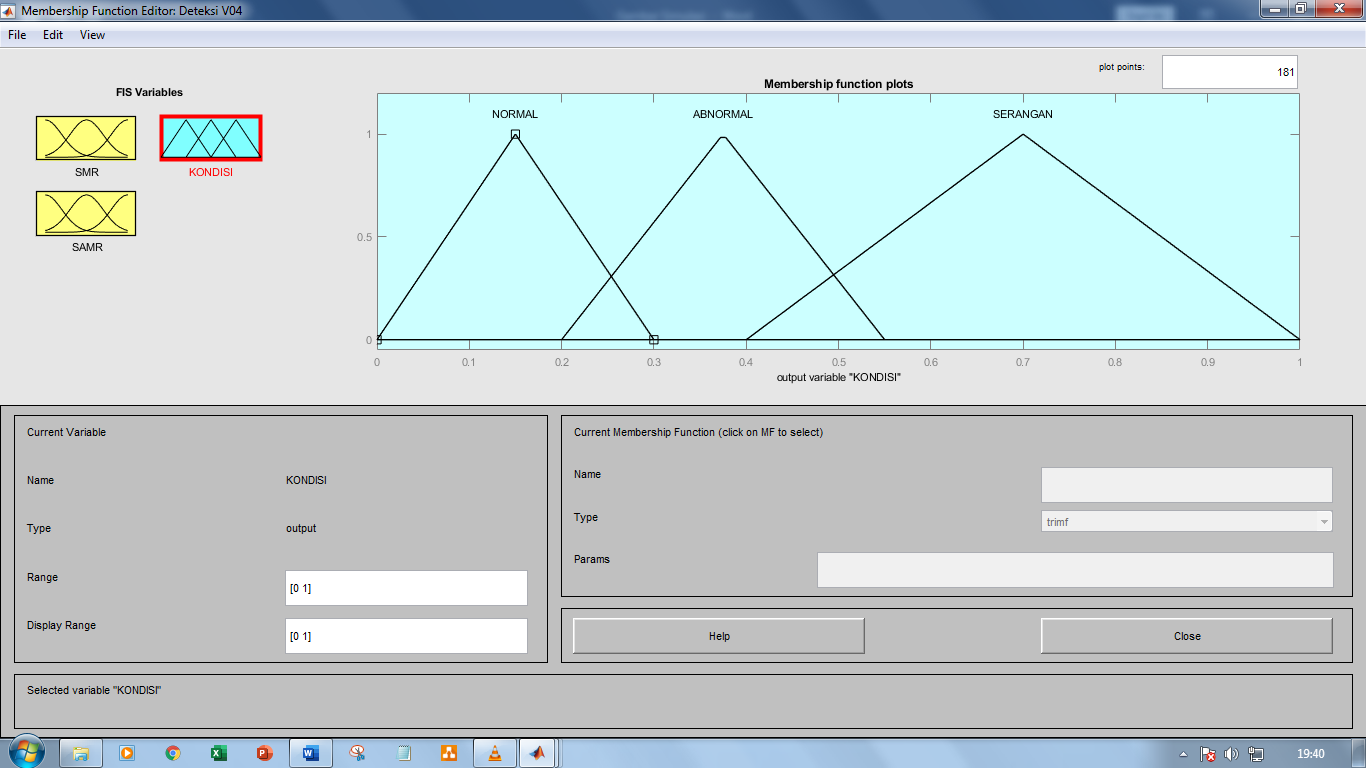


Figure 8. Output membership function

* 1. False Positive Ratio (FPR)

False Positive Ration (FPR) digunakan untuk menentukan tingkat kesalahan fuzzy saat mendeteksi serangan saat tidak terjadi serangan pada jaringan. Persamaan untuk menghitung parameter FPR tertera pada persamaan 5. Gambar 9 menunjukkan perolehan nilai FPR tiap skenario. FPR terendah diperoleh pada skenario 20 nodes dengan nilai 0,1047; sedangkan FPR tertinggi diperoleh pada skenario 30 nodes dengan nilai 0,4048.

|  |  |
| --- | --- |
|  | (5) |

Figure 9. FPR value per scenario

* 1. Accuracy

*Accuracy* digunakan untuk menentukan persentase keberhasilan *fuzzy* mendeteksi serangan saat terjadi serangan maupun tidak terjadi serangan. Persamaan untuk menghitung *accuracy* tertera pada persamaan 6. Gambar 10 menunjukkan persentase *accuracy* tiap skenario. *Accuracy* tertinggi diperoleh pada skenario 20 *nodes* saat interval serangan *ATK 5S* dengan persentase 94,74%; sedangkan *accuracy* terendah diperoleh pada skenario 30 *nodes* saat interval serangan *ATK 9S* dengan persentase 79,29%.

|  |  |
| --- | --- |
|  | (6) |

Figure 10. Accuracy percentage per scenario

* 1. Precision

*Precision* digunakan untuk menentukan jumlah hasil deteksi seranganyang diidentifikasi benar diantara semua hasil deteksi serangan. Persamaan untuk menghitung parameter *precision* tertera pada persamaan 7. Gambar 11 menunjukkan perolehan nilai *precision* berdasarkan hasil pengujian tiap skenario. *Precision* tertinggi diperoleh pada skenario 20 *nodes* saat interval serangan *ATK 3S* dan *ATK 9S* dengan nilai 0,9082. *Precision* terendah diperoleh pada skenario 30 *nodes* saat interval serangan *ATK 9S* dengan nilai 0,7119.

|  |  |
| --- | --- |
|  | (7) |

Figure 11. Precision value per scenario

* 1. Recall

*Recall* digunakan untuk menentukan jumlah hasil deteksi seranganyang diidentifikasi benar pada saat terjadi serangan. Persamaan untuk menghitung parameter *recall* tertera pada persamaan 8. Gambar 12 menunjukkan nilai *recall* yang diperoleh dari hasil pengujian tiap skenario. Nilai *recall* pada tiap skenario memiliki nilai yang relatif sama. Apabila nilai *recall* tiap skenario dirata-ratakan, maka nilai rata-rata *recall* terbaik diperoleh pada skenario 15 *nodes* dengan nilai rata-rata 1, sedangkan nilai rata-rata *recall* terendah diperoleh pada skenario 20 *nodes* dengan nilai rata-rata 0,9914.

|  |  |
| --- | --- |
|  | (8) |

Figure 12. Average recall per scenario

* 1. F-Score

*F-score* digunakan untuk menghitung rata-rata dari *precision* dengan *recall*. Persamaan untuk menghitung parameter *f-score* tertera pada persamaan 9. Gambar 13 menunjukkan nilai *f-score* yang diperoleh dari tiap skenario pengujian. *F-score* tertinggi diperoleh pada skenario 20 *nodes* saat interval serangan *ATK 5S* dengan nilai 0,9497; sedangkan *f-score* terendah diperoleh pada skenario 30 *nodes* saat interval serangan *ATK 9S* dengan nilai 0,8276.

|  |  |
| --- | --- |
|  | (9) |

Figure 13. F-Score value per scenario

* 1. Fuzzy Logic Algorithm Analysis

Tabel 4 menunjukkan nilai rata-rata kinerja algoritma *fuzzy logic* pada tiap skenario. Saat tidak terjadi serangan, kinerja algoritma *fuzzy logic* terbaik diperoleh pada skenario 20 *nodes*. Terlihat pada tabel 4 bahwa skenario 20 *nodes* memiliki *FPR* terendah. Sedangkan saat sedang terjadi serangan, kinerja algoritma *fuzzy logic* terbaik diperoleh pada skenario 15 *nodes*. Terlihat pada tabel 4 bahwa skenario 15 *nodes* memiliki *recall* tertinggi. Hal ini disebabkan oleh keterbatasan *node fuzzy* yang hanya mampu menerima maksimal dua topik di waktu yang bersamaan. Intensitas trafik pada skenario 15 *nodes* saat tidak terjadi serangan sangat sedikit. Hal ini menyebabkan *node fuzzy* menerima lebih dari dua topik dalam waktu yang bersamaan (melebihi kapasitas penerimaan) sehingga akan ada informasi dari *broker* yang dibuang oleh *node fuzzy*. Sedangkan intensitas trafik pada skenario 15 *nodes* saat terjadi serangan sesuai dengan kapasitas penerimaan *node fuzzy* sehingga topik yang dikirim oleh *broker* dapat diterima dengan baik oleh *node fuzzy*. Berbeda pada skenario 20 *nodes* saat tidak terjadi serangan, intensitas trafik sesuai dengan kapasitas penerimaan *node fuzzy* sehingga semua informasi yang dikirimkan oleh *broker* dapat diterima dengan baik. Namun, saat terjadi serangan pada skenario 20 *nodes*, intensitas trafik cukup tinggi sehingga menyebabkan *delay* pada pertukaran informasi antara *broker* dengan *node fuzzy*.Hal ini menimbulkan kesalahan pada *node fuzzy* saat memfilter trafik dan mendeteksi serangan *DoS*. Maka dari itu *FPR* tertinggi diperoleh pada skenario 20 *nodes* dan *recall* tertinggi diperoleh pada skenario 15 *nodes*.

Table 4. Average performance of fuzzy logic per scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scenario** | **FPR** | **Accuracy** | **Precision** | **Recall** | **F-Score** |
| **15 Nodes** | 0,1591 | 92,00% | 0,8613 | 1,0000 | 0,9255 |
| **20 Nodes** | 0,1047 | 94,39% | 0,9062 | 0,9915 | 0,9469 |
| **25 Nodes** | 0,3256 | 83,72% | 0,7570 | 0,9972 | 0,8607 |
| **30 Nodes** | 0,4048 | 79,88% | 0,7166 | 0,9971 | 0,8339 |

Penentuan kondisi jaringan tertinggi yang diidentifikasi benar saat terjadi serangan maupun saat tidak terjadi serangan diperoleh pada skenario 20 *nodes*, sedangkan penentuan kondisi jaringan terendah diperoleh pada skenario 30 *nodes*. Terlihat dari perolehan *accuracy* yang ditunjukkan pada tabel 4. Tingkat hasil deteksi serangantertinggi yang diidentifikasi benar saat terjadi serangan maupun tidak terjadi serangan diperoleh pada skenario 20 *nodes*, sedangkan tingkat hasil deteksi seranganterendah diperoleh pada skenario 30 *nodes*. Terlihat dari perolehan *precision* yang ditunjukkan pada tabel 4. Secara keseluruhan, kinerja rata-rata algoritma *fuzzy logic* terbaik saat terjadi serangan maupun tidak terjadi serangan diperoleh pada skenario 20 *nodes*, sedangkan kinerja rata-rata algoritma *fuzzy logic* terburuk diperoleh pada skenario 30 *nodes*. Terlihat dari perolehan *f-score* yang ditunjukkan pada tabel 2. Hal ini disebabkan karena pada skenario 20 *nodes*, kinerja algoritma *fuzzy logic* saat tidak terjadi serangan sangat baik. Terlihat dari rendahnya *FPR* pada skenario 20 *nodes* sehingga parameter *accuracy*, *precision*, dan *f-score* akanbernilai paling tinggi. Sedangkan pada skenario 30 *nodes*,kinerjaalgoritma *fuzzy logic* saat tidak terjadi serangan sangat buruk. Terlihat dari tingginya *FPR* pada skenario 30 *nodes* sehingga parameter *accuracy*, *precision*, dan *f-score* akan bernilai paling rendah.

1. Conclusions

Kemampuan *node fuzzy* dalam menerima topik pada waktu yang sama berjumlah maksimal dua topik. Hal tersebut dapat menyebabkan kesalahan pada algoritma *fuzzy logic* saat medeteksi serangan DoS. FPR berbanding terbalik dengan *accuracy*, *precision*, dan *f-score*. FPR tertinggi diperoleh pada skenario 30 *nodes* dengan nilai 0,4048; sedangkan FPR terendah diperoleh pada skenario 20 *nodes* dengan nilai 0,1047. *Accuracy*, *precision*, dan *f-score* pada skenario 30 *nodes* memiliki nilai terendah, yaitu 79,88%; 0,7166; dan 0,8339. *Accuracy*, *precision*, dan *f-score* pada skenario 20 *nodes* memiliki nilai tertinggi, yaitu 94,39%; 0,9062; dan 0,9469. Rata-rata *recall* tertinggi diperoleh pada skenario 15 *nodes* dengan nilai 1 yang artinya semua hasil deteksi algoritma *fuzzy* *logic* saat terjadi serangan pada skenario 15 *nodes* memiliki tingkat deteksi yang sempurna. Intensitas trafik dan jumlah nodes berbanding lurus terhadap FPR, namun berbanding terbalik terhadap *accuracy*, *precision*, dan *f-score*.

Saran yang diberikan untuk mengembangkan penelitian ini yaitu adalah memperbarui skenario pengujian seperti menambah jumlah nodes client maupun penyerang dan/atau mengganti trafik serangan seperti *PUBLISH*, *UNSUBSCRIBE*, atau *PINGREQ*; menanamkan algoritma *fuzzy* *logic* pada *broker* atau pada *border* *router* sehingga dapat meningkatkan kinerja algoritma *fuzzy logic* dan bahkan dapat dikembangkan menjadi *IP-Sec* atau *firewall* dalam menghadapi serangan DoS; menggunakan QoS *level* MQTT 1 atau 2 sehingga dapat mendeteksi DoS menggunakan trafik MQTT lain seperti *PUBACK* atau *PUBREC* sebagai *input* pada algoritma *fuzzy* *logic*; menggunakan metode *fuzzy* *controller* lain seperti metode *Takagi-Sugeno* atau metode *Tsukamoto*.

1. References

[1] A. P. Haripriya, “Secure-MQTT : an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things,” 2019.

[2] A. Velinov and A. Mileva, “Running and Testing Applications for Contiki OS Using Cooja Simulator,” no. June 2016, 2018.

[3] M. Yassine, A. Ezzati, and M. Belaissaoui, “An Enhanced DTLS Protocol for Internet of Things Applications,” no. February 2018, 2016.

[4] P. Kasinathan, C. Pastrone, M. A. Spirito, and M. Vinkovits, “Denial-of-Service detection in 6LoWPAN based Internet of Things,” no. October, 2013.

[5] W. Li, S. Tug, W. Meng, and Y. Wang, “Designing collaborative blockchained signature-based intrusion detection in IoT environments,” *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019.

[6] M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai, “Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs,” *2018 Int. Conf. Adv. Comput. Commun. Informatics*, pp. 2244–2250, 2018.

[7] S. Andy, B. Rahardjo, and B. Hanindhito, “Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System,” no. May, 2018.

[8] S. Shin, K. Kobara, C. C. Chuang, and W. Huang, “A security framework for MQTT,” *2016 IEEE Conf. Commun. Netw. Secur. CNS 2016*, pp. 432–436, 2017.

[9] G. Potrino and A. F. Santamaria, “Modeling and evaluation of a new IoT security system for mitigating DoS attacks to the MQTT broker,” *2019 IEEE Wirel. Commun. Netw. Conf.*, pp. 1–6, 2019.

[10] S. Hameed, F. I. Khan, and B. Hameed, “Understanding Security Requirements and Challenges in Internet of Things (IoT): A Review,” *Journal of Computer Networks and Communications*, vol. 2019. 2019.

[11] G. Potrino, F. De Rango, and P. Fazio, “A Distributed Mitigation Strategy against DoS attacks in Edge Computing,” *Wirel. Telecommun. Symp.*, vol. 2019-April, pp. 1–7, 2019.

[12] S. K. Yee and J. V Milanovi, “Fuzzy Logic Controller for Decentralized Stabilization of Multimachine Power Systems,” vol. 16, no. 4, pp. 971–981, 2008.

[13] M. Alali, A. Almogren, M. M. Hassan, I. A. L. Rassan, and Z. Alam, “Improving Risk Assessment Model of Cyber Security Using Fuzzy Logic Inference System,” *Comput. Secur.*, 2017.

[14] S. Dick, “Toward Complex Fuzzy Logic,” vol. 13, no. 3, pp. 405–414, 2005.

[15] A. Machado *et al.*, “A Fuzzy Inference System to Support Medical Diagnosis in Real Time,” *Procedia Comput. Sci.*, vol. 122, pp. 167–173, 2017.

[16] K. Ozera, K. Bylykbashi, Y. Liu, and L. Barolli, “A Fuzzy-Based Approach for Cluster Management in VANETs: Performance Evaluation for Two Fuzzy-Based Systems,” *Internet of Things*, 2018.

[17] A. F. Santamaria and F. De Rango, “A real IoT device deployment for E-Health applications under lightweight communication protocols, activity classifier and Edge data filtering,” *Comput. Commun.*, 2018.

[18] M. S. Munir, I. S. Bajwa, and S. M. Cheema, “An intelligent and secure smart watering system using fuzzy logic and blockchain ✩,” *Comput. Electr. Eng.*, vol. 77, pp. 109–119, 2019.

[19] N. Upasani and H. Om, “A modified neuro-fuzzy classifier and its parallel implementation on modern GPUs for real time intrusion detection,” *Appl. Soft Comput. J.*, vol. 82, p. 105595, 2019.

[20] Rahmat, Firman A., et al. "Prototipe Robot Kapal Pengukur Tingkat PH dan Turbiditas Air Berbasis Metode Modified Fuzzy." Jurnal Rekayasa Elektrika, vol. 14, no. 1, 2018, pp. 43-50, doi:10.17529/jre.v14i1.9771.

[21] S. H. Ramos, M. T. Villalba, and R. Lacuesta, “MQTT Security : A Novel Fuzzing Approach,” vol. 2018, 2018.