Analisis Serangan DDoS Menggunakan Metode Jaringan Saraf Tiruan

M. Alfine Ridho[1], Molavi Arman[2]*
Department Technical Information[1], STMIK GI MDP
Department Informatics Management[2], AMIK MDP
Palembang, Indonesia
alfinridho99@mhs.mdp.ac.id[1], molavi.arman@mdp.ac.id[2]

Abstract*— DDoS attack (Distribute Denial of Service) is one of the weapons of choice from hackers because it's proven it has become threat on the internet worlds. The frequent of DDoS attacks creates a threat to internet users or servers, so that requires the introduction of several new methods that occur, one of which can use the IDS (Intrusion Detection System) method. This study took advantage of Neural Network ability to detect DDoS attack or normal based on traffic log processed statistically using Fixed Moving Window. The DDOS attack scheme uses a network topology that has been designed based on the needs and objectives that are found in monitoring network traffic. In each DDoS data and normal consist of 27 traffic log with total numbers of dataset as much as 54 data along with each testing data as much as 10 DDoS data and normal. Data collection was performed using LOIC, HOIC, and DoSHTTP with 300 seconds of traffic monitoring. The result of the Fixed Moving Window processing is the extraction value that will be put in the Neural Networks have 6 input values, one hidden layer with 300 neurons and 2 outputs which consist of a normal dataset and a DDoS dataset. The results of this study showed that Neural Network can detect DDoS and Normal in a good way with accuracy value as much as 95%.

Keywords*— DDOS, Neural Netwok, Fixed Moving Window, Traffic Log, Intrusion Detection System.

Abstrak*— Serangan DDoS (*Distribute Denial of Service*) menjadi salah satu senjata pilihan *hacker* karena telah terbukti menjadi ancaman di dunia internet. Sering terjadinya serangan DDoS membuat adanya ancaman terhadap pengguna internet atau *server*, maka dari itu diperlukannya pengenalan baru beberapa metode yang terjadi, salah satunya dapat menggunakan metode IDS (*Intrusion Detection System*). Penelitian ini memanfaatkan kemampuan *Neural Network* untuk mendeteksi serangan DDoS atau normal berdasarkan *traffic log* yang diolah secara statistik menggunakan *Fixed Moving Window*. Skema penyerangan DDoS menggunakan topologi jaringan yang telah dirancang berdasarkan kebutuhan dan tujuan yang ingin didapatkan dalam pemantauan *nettwork traffic*. Setiap data DDoS dan normal terdiri dari 27 *traffic log* dengan total jumlah dataset sebanyak 54 data dengan jumlah data uji masing – masing sebanyak 10 data DDoS dan Normal. Pengambilan dataset dilakukan menggunakan LOIC, HOIC, dan DoSHTTP dengan pemantauan *traffic* selama 300 detik. Hasil pengolahan *Fixed Moving Window* didapatkan nilai ekstraksi yang akan di masukkan ke dalam Jaringan Saraf Tiruan yang memiliki nilai input sebanyak 6 nilai, satu hidden layer dengan neuron berjumlah 300 dan 2 output yang terdiri dari dataset normal dan dataset DDoS. Hasil pengujian menunjukkan bahwa *Neural Network* dapat mendeteksi serangan DDoS dan Normal secara baik dengan nilai *accuracy* sebesar 95%.

Kata Kunci*—DDOS, Neural Network, Fixed Moving Window, Traffic Log, Intrusion Detection System.

I. PENDAHULUAN

DDoS (*Distribute Denial of Service*) merupakan jenis serangan yang terstruktur, serangan DDoS adalah serangan yang mungkin bisa sering kita jumpai diantara serangan lainnya [1]. Serangan DDoS merupakan teknik yang paling populer dan menjadi senjata pilihan *hacker* karena telah terbukti menjadi ancaman di internet, serangan ini telah ada sejak tahun 1990 [2]. Serangan DDoS mampu melumpuhkan *server* dengan membanjiri lalu lintas jaringan dan mengakibatkan *down*. Ancaman dan serangan terhadap keamanan *server* terus meningkat, banyaknya kemudahan dan ketersedian informasi mengenai *hacking* yang dapat diakses dengan mudah di internet sehingga menjadikan pelaku mudah mendapatkan informasi untuk di jadikan sebagai target kejahatan [3]. *Intrusion Detection System* (IDS) merupakan suatu sistem yang digunakan untuk mendeteksi serangan yang ada pada jaringan, jika terjadi aktivitas yang mencurigakan pada *network traffic* [4]. Sering terjadinya serangan DDoS membuat adanya ancaman terhadap pengguna internet atau *server*, maka dari itu diperlukannya pengenalan baru beberapa metode untuk mendeteksi serangan DDoS yang terjadi, diantaranya dapat menggunakan metode IDS (*Intrusion Detection System*) yang sebelumnya sudah ada lebih dulu untuk mendeteksi serangan DDoS. Pada umumnya serangan DDoS terbagi menjadi 3 jenis yaitu serangan dengan basis *bandwith*, serangan dengan lalu lintas jaringan dan serangan dengan basis aplikasi [5]. Metode Neural Network secara algoritma EM (*Expectation-Maximization*) digunakan untuk mendeteksi adanya serangan DDoS. Menurut dataset DARPA ada 21 kelompok serangan yang dapat dikelompokkan menjadi 13 kelompok serangan, sehingga terjadi kesalahan pemisahan *alert* dari jenis serangan yang sama, sehingga menjadi serangan yang berbeda [6]. Alasan digunakannya metode Jaringan Saraf Tiruan dikarenakan metode ini merupakan salah satu yang mendekati untuk digunakan dalam penelitian ini, untuk mengolah data traffic yang didapat dan

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

diolah menggunakan metode fixed moving window karena metode ini bisa digunakan untuk proses ekstraksi yang akan diinput ke dalam Jaringan Saraf Tiruan sehingga hasil olahan bisa masuk sebagai input layer pada Jaringan Saraf Tiruan, hasil akhir dari penelitian ini akan didapatkan output berupa nilai akurasi dari proses pengenalan Jaringan Saraf Tiruan tersebut.

II. TINJAUAN PUSTAKA

A. Linux

Linux merupakan sebuah *Operating System* (OS) seperti *Windows* yang *open source* atau gratis dibawah lisensi GNU, linux merupakan turunan dari Unix dan dapat dijalankan diberbagai macam *platform* perangkat keras seperti intel (x86) sampai prosesor RISC [7].

B. Neural Network

Neural Network (NN) adalah sistem pemroses informasi yang karakteristiknya mirip dengan jaringan syaraf biologi, dimana untuk melakukan tugas sederhana dalam memproses informasi, otak manusia terdiri dari sejumlah *neuron*. Dikarenakan adanya keterhubungan antar *neuron*, maka otak dapat melakukan fungsi pemrosesan yang cukup kompleks [8]. Metode ini menggunakan elemen perhitungan *non-linier* dasar yang disebut *neuron* yang diorganisasikan sebagai jaringan yang saling berhubungan seperti jaringan saraf manusia. Terdapat tiga elemen dasar dari model *neuron*, yaitu: Jalur hubungan atau sekumpulan sinapsis, di mana masing-masing sinapsis tersebut mempunyai bobot atau kekuatan hubungan. Suatu *adder* untuk menjumlahkan sinyal-sinyal *input* yang diberi bobot oleh sinapsis *neuron* agar sesuai. Operasi-operasi yang digambarkan mengikuti aturan dari *linear combiner*. Fungsi aktivasi untuk membatasi amplituda *output* setiap *neuron* [9]. Penggunaan *threshold* memberikan pengaruh adanya *affine transformation* terhadap *output* u_k dari *linear combiner* pada model gambar berikut:

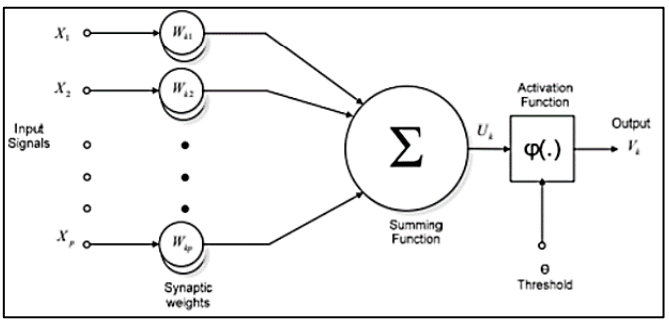

Fig. 1. Model Matematis *Neuron*

Didalam *Neural Network*, *neuron-neuron* akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut lapisan *neuron* (*neuron layers*). Kemudian *neuron-neuron* pada satu lapisan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya. Kemudian informasi yang diberikan pada jaringan syaraf akan dirambatkan dari lapisan ke lapisan, dimulai dari lapisan masukan sampai ke lapisan keluaran melalui lapisan tersembunyi (*hidden layer*) [10]. Terdapat 3 macam arsitektur *Neural Network*, yaitu:

A. Single Layer Net

Jaringan ini hanya memiliki 1 lapisan dengan bobot-bobot terhubung, dan hanya menerima masukan dan diolah menjadi keluaran tanpa melalui lapisan tersembunyi. *Neuron-neuron* pada gambar berikut menunjukkan kedua lapisan saling berhubungan dan seberapa besar hubungan antara 2 *neuron* tersebut ditentukan oleh bobot yang bersesuaian dan terlihat semua unit masukan akan dihubungkan dengan setiap unit keluaran.

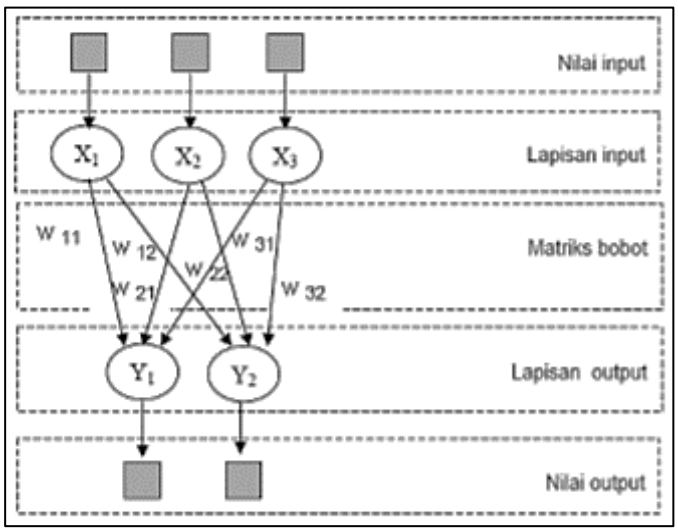

Fig. 2. Jaringan Lapisan Tunggal

2) Multi Layer Net

Jaringan ini memiliki 1 atau lebih lapisan yang terletak diantara lapisan masukan dan lapisan keluaran. Pada Gambar 3 terlihat umumnya lapisan bobot-bobot terletak antara 2 lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan tunggal, dikarenakan pembelajarannya yang lebih rumit. Pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses menyelesaikan masalah pada banyak kasus.

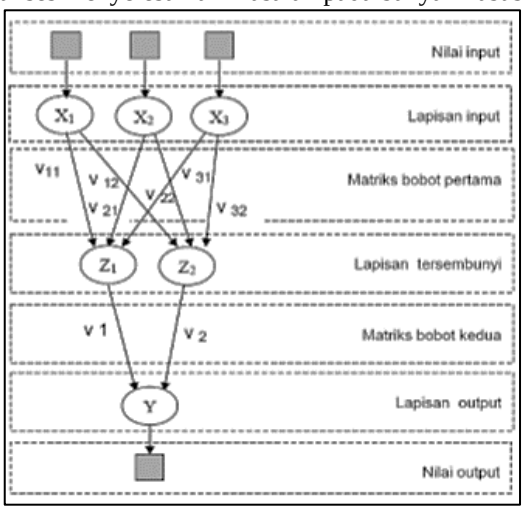

Fig. 3. Jaringan Banyak Lapisan

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

3) Single Layer Net

Pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Umumnya hubungan antar *neuron* pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Pada gambar 4 menunjukkan suatu contoh arsitektur jaringan dengan lapisan kompetitif yang mempunyai bobot.

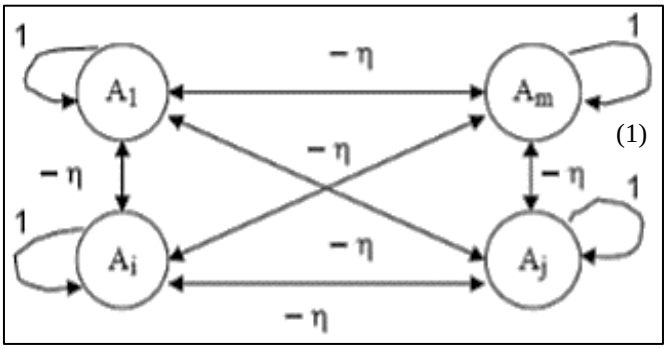

Fig. 4. Jaringan Lapisan Kompetitif

C. DDoS

DDoS (*Distributed Denial of Service*) merupakan jenis serangan yang bertujuan menggangu hak akses pengguna jaringan yang dilakukan secara massif [11]. Secara umum serangan DDoS terdiri dari beberapa jenis, serangan dengan basis *bandwidth*, serangan dengan basis lalu lintas jaringan, dan serangan dengan basis aplikasi.

D. Apache

Apache adalah sebuah nama dari *web server* yang bertanggung jawab atas permintaan, menerima HTTP dan *logging* informasi secara detail. *Apache* juga dapat diartikan sebagai suatu *web server* yang kompak, dan mengikuti standar protokol HTTP yang digemari [5].

E. Web Server

Web Server atau yang bisa juga disebut *server web* adalah perangkat lunak dalam *server* yang mempunyai fungsi untuk menerima permintaan halaman *web* melalui protokol HTTP dan HTTPS dari *client* yang dikenal dengan sebutan *browser*, setelah itu mengirimkan kembali hasil permintaan yang telah dilakukan sebelumnya dalam bentuk halaman *web* yang berbentuk sebuah dokumen HTML [2].

F. HTTP

Hypertext Transfer Protokol (HTTP) adalah sebuah protokol yang meminta dan menjawab *client* dan *server*. *Client* seperti *browser* memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu. Sebuah *server* di *port* tersebut menunggu *client* untuk mengirim sebuah kode permintaan seperti GET / HTTP yang akan meminta halaman yang sudah ditentukan, yang diikuti dengan pesan MIME yang mempunyai informasi kode sementara yang diperlukan oleh protokol. Setelah menerima kode permintaan, *server* akan mengirim kembali kode dan sebuah pesan yang diminta, atau sebuah pesan *error* atau pesan lainnya.

G. Fixed Moving Window

Fixed Moving Window adalah metode perhitungan statistik yang digunakan untuk mendapatkan *extraction feature* dari *log data traffic* [6]. Perhitungan statistik yang digunakan adalah sebagai berikut:

A. Rata – rata ukuran atau Panjang paket

Merupakan nilai yang menyatakan rata – rata ukuran/panjang paket dalam satu *window*/*frame* waktu tertentu. Dapat dinyatakan dengan persamaan berikut :

$$Rata - rata\ paket = \frac{\Sigma \mathrm{Pn}}{n}$$

Keterangan :
Pn = Besar data pada indeks ke-n
n = Banyak data

B. Jumlah Paket

Merupakan total paket dalam satu *window*/*frame* waktu tertentu.

C. Variansi Waktu Kedatangan Paket

Merupakan nilai akar dari deviasi waktu kedatangan paket, yang dinyatakan dengan rumus pada persamaan berikut :

$$Variansi\ Waktu = \sqrt{\frac{\sum(tn - \bar{t})^2}{n}} \qquad (2)$$

Keterangan :
tn = waktu paket diterima
$\bar{t}$ = rata-rata waktu paket diterima

D. Variansi Ukuran atau Panjang Paket

Merupakan nilai akar dari deviasi ukuran/panjang paket, yang dinyatakan dengan rumus pada persamaan berikut :

$$Variansi\ Ukuran = \sqrt{\frac{\sum(pn - \bar{\rho})^2}{n}} \qquad (3)$$

Keterangan :
pn = panjang paket diterima
$\bar{\rho}$ = rata-rata panjang paket diterima

E. Kecepatan Paket / Detik

Kecepatan Paket/Detik. Merupakan banyaknya aliran paket data dalam satu *window*/*frame* waktu tertentu, yang dihitung dengan rumus pada persamaan berikut :

$$Kp = np * \frac{1}{T.akhir - T.awal} \qquad (4)$$

F. Jumlah Bit

Merupakan jumlah total bit data yang terdapat dalam satu *window/frame* waktu tertentu.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

III. METODOLOGI PENELITIAN

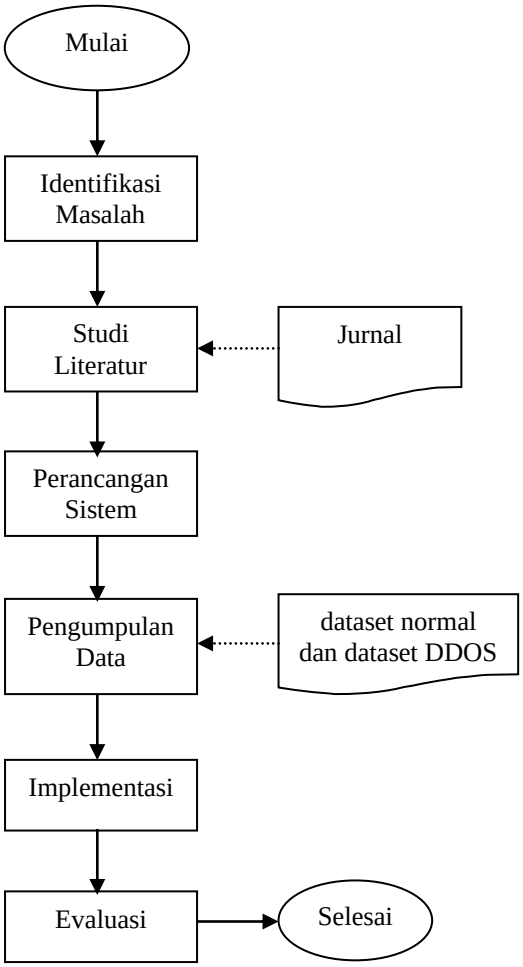

Fig. 5. Rancangan Metodologi

A. *Identifikasi Masalah*

Penelitian dimulai dengan mencari topik – topik mengenai perkembangan penelitian tentang identifikasi serangan DDoS.

B. *Studi Literatur*

Penulis melakukan riset jurnal dengan mencari beberapa jurnal yang berkaitan dengan topik skripsi seperti pembahasan mengenai DDoS, *Fixed Moving Window*, Jaringan Saraf Tiruan (JST) dan *tools* yang diperlukan untuk melakukan percobaan serangan terhadap *website* dan mengamati *network traffic web server* sehingga penulis mendapatkan dataset yang diperlukan.

C. *Perancang Sistem*

Tahapan ini digunakan untuk perancangan sistem agar dapat dilakukan proses simulasi penyerangan dan mempersiapkan aplikasi yang akan dibutuhkan untuk mendapatkan dataset normal dan dataset DDoS, adapun aplikasi yang akan digunakan berupa LOIC, HOIC, dan DoSHTTP sehingga penulis dapat menentukan waktu pelaksanaan pengujian untuk melakukan serangan DDoS. Perancangan topologi dilakukan untuk menentukan langkah-langkah yang akan dilakukan dalam pelaksanaan simulasi serangan DDoS terhadap *web server*, serangan dilakukan dengan mengikuti alur pada topologi jaringan pengujian, proses pengujian dapat dilihat pada gambar 6.

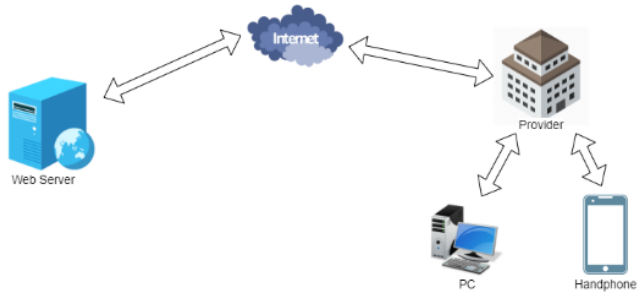

Fig. 6. Topologi Jaringan Pengujian

4) *Perancangan Sistem*

Pada tahap ini dilakukan pengumpulan data hasil dari proses penyerangan. Proses pengambilan setiap satu *log traffic* dilakukan selama 300 detik dan dilakukan penjedaan 60 detik sehingga selanjutnya dapat dilakukan kembali pengambilan data *log traffic*. Data *log traffic* yang didapatkan masih berekstensi .log dan akan diubah kedalam bentuk .csv agar mudah dilakukan pengolahan ke dalam metode *fixed moving window*. Pengumpulan *dataset* berjumlah 54 data *log traffic*, masing – masing 34 *dataset* digunakan untuk data latih yang terdiri dari 17 *data log traffic* normal dan 17 *data log traffic* DDoS. 20 data *log traffic* digunakan sebagai data uji, masing – masing 10 data uji Normal dan 10 data uji DDoS. Untuk detail pengambilan *dataset* dapat dilihat pada tabel 1 :

TABLE I. DETAIL PENGAMBILAN DATASET

Jenis *Traffic Log*	**Jumlah masing-masing *dataset***
DDoS	27 data
Normal	27 data
Total	54 data

5) *Implementasi*

Setelah didapatkan dataset, selanjutnya dilakukan pengolahan terhadap dataset tersebut menggunakan metode *Fixed Moving Window* agar memudahkan dataset tersebut mendapatkan ciri ekstraksi yang dibutuhkan untuk dikenali oleh metode Jaringan Saraf Tiruan (JST). Lalu pada tahapan ini dilakukan pelatihan data latih. Data latih tersebut didapatkan dari hasil pengolahan dataset yang telah dilakukan, setelah dilakukan pelatihan Jaringan Saraf Tiruan (JST), kemudian memasukkan nilai target pada data latih dan menggunakan *Function Training Gradient Descent with Momentum* & *Adaptive LR* (traingdx) untuk mengenali data latih dan data uji. Arsistektur Jaringan Saraf Tiruan menggunakan *single layer*. Selanjutnya melakukan pengujian Jaringan Saraf Tiruan (JST) dengan data uji apakah jaringan

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

tersebut dapat mengenali *data traffic* secara normal atau DDoS, dengan cara memasukkan hasil nilai *input* sebanyak 6 nilai, satu *hidden layer* dengan *neuron* berjumlah 300 dan 2 *output* yang terdiri dari dataset normal dan dataset DDoS. Arsitektur model Jaringan Saraf Tiruan ini menggunakan *single hidden layer*, nilai *input* pada node akan langsung melakukan proses ke *hidden layer*, node *hidden layer* pada gambar menggunakan 300 node. Epoch pada pelatihan Jaringan Saraf Tiruan sebanyak 1000 iterasi yang artinya proses *training* akan dilakukan sebanyak 1000 kali dan *goal* parameter latih sebesar 0.1. Untuk normalisasi menggunakan fungsi min-max, dengan cara merubah data dari suatu *range* ke dalam *range* baru. Hubungan *hidden layer* dan *output* terjadi pada proses komputasi terhadap bobot dan bias serta dihitung juga besarnya *output* dari *hidden layer* ke *output* berdasarkan fungsi aktivasi *logsig* dengan rentang nilai 0 sampai 1 dan menggunakan fungsi minmax pada tahap normalisasi.

6) Evaluasi

Tahap ini dilakukan setelah pengujian pada matlab untuk mengetahui hasil yang didapatkan sudah mendapatkan hasil yang memuaskan dan optimal atau belum, jika belum mendapatkan hasil yang diharapkan maka akan dilakukan pengoptimal ulang. Untuk menghitung tingkat keberhasilan dari metode yang digunakan, dapat dilakukan dengan menggunakan metode *Confusion Matrix*, metode ini akan menghitung nilai *Accuracy*. Cara penghitungan *Confusion Matrix* tersebut bisa dilihat pada persamaan (3.1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \; x \; 100\% \qquad (5)$$

Keterangan :
TP = Jumlah data positif DDoS yang terklasifikasi dengan benar oleh sistem.
TN = Jumlah data negatif DDoS (Normal) yang terklasifikasi dengan benar oleh sistem.
FN = Jumlah data negatif DDoS (Normal) namun terklasifikasi salah oleh sistem
FP = Jumlah data positif DDoS namun terklasifikasi salah oleh system.

Berikut adalah contoh tabel *Confusion Matrix* yang dapat dilihat pada tabel 2.

TABLE II. CONFUSION MATRIX

		Actual Class	
		DDoS	Normal
Predicted class	DDoS	*True Positive*	*False Positive*
	Normal	*False Negative*	*True Negative*

IV. HASIL DAN PEMBAHASAN

A. Implementasi Penyerangan DDoS dan Pengambilan Dataset

Pada tahap ini *server* mempersiapkan *tools* yang digunakan untuk melihat *traffic log data*. Selanjutnya *server standby* untuk memonitoring *data traffic log* selama 300 detik untuk mendapatkan *traffic data* normal sehingga didapatkan *data traffic normal*. Kemudian *client* membuka *tools* untuk melakukan serangan DDoS menggunakan aplikasi LOIC, HOIC dan DoSHTTP selama 300 detik. Selama 300 detik tersebut *server* masih memonitoring data sehingga didapatkan data DDoS. Tahap selanjutnya ketika sudah mendapatkan data mentah *traffic log* Normal dan DDoS, *server* menyimpan data *log* tersebut kedalam format *.log*. Data mentah *traffic log* tersebut dikonversi kedalam format *.csv* yang selanjutnya akan dirapikan menggunakan excel menjadi data yang terstruktur sehingga data mentah *traffic log* tersebut dapat diolah kedalam matlab untuk dilakukan pengolahan menggunakan metode *Fixed Moving Window*

Time	Jenis Paket	Interface	Jumlah Paket	Info				
1	TCP	enp1s0	1440	from 114.125.236.14:49324 to 202.45.67.2:443	first packet			
1	TCP	enp1s0	40	from 202.45.67.2:443 to 114.125.236.14:49324	first packet			
1	TCP	enp1s0	96	from 202.45.67.2:22 to 202.45.67.10:33468	first packet			
1	TCP	enp1s0	52	from 202.45.67.10:33468 to 202.45.67.2:22	first packet			
1	TCP	enp1s0	52	from 114.5.216.200:29630 to 202.45.67.2:443	first packet			
1	TCP	enp1s0	1440	from 202.45.67.2:443 to 114.5.216.200:29630	first packet			
1	TCP	enp1s0	923	from 158.140.187.249:58448 to 202.45.67.2:443	first packet			
1	TCP	enp1s0	40	from 202.45.67.2:443 to 158.140.187.249:58448	first packet			
1	TCP	enp1s0	60	from 202.45.67.2:10908 to 202.45.67.9:3306	first packet (SYN)			
1	TCP	enp1s0	60	from 202.45.67.9:3306 to 202.45.67.2:10908	first packet (SYN)			
1	TCP	enp1s0	52	from 202.45.67.2:10908 to 202.45.67.9:3306	FIN sent	30 packets, 3653, avg flow rate	3.57 kBps	
1	TCP	enp1s0	52	from 202.45.67.9:3306 to 202.45.67.2:10908	FIN acknowleged			
1	TCP	enp1s0	52	from 202.45.67.9:3306 to 202.45.67.2:10908	FIN sent	25 packets, 16790, avg flow rate	16.40 kBps	

Fig. 7. Contoh *Traffic Log Data*

2. Implementasi Metode Fixed Moving Window

Pada tahap ini dilakukan pengolahan data untuk mendapatkan *feature extraction* agar dapat di *input* ke dalam arsitektur Jaringan Saraf Tiruan (JST). Untuk hasil ekstraksi ciri *Fixed Moving Window* pada data latih disimpan dengan nama "dataLatih" dan hasil ektraksi ciri *Fixed Moving Window* pada data uji disimpan dengan nama file "dataUji".
Berikut ini adalah jenis statistik nilai *feature extraction* yang digunakan sebagai *input* pada Jarigan Saraf Tiruan (JST) :

- Rata – rata ukuran Panjang
- Jumlah Paket
- Variansi Waktu
- Variansi Ukuran
- Kecepatan Paket
- Jumlah Bit

Masing – masing nilai *Feature Extraction* didapatkan dari hasil pengolahan *log traffic data* yang digunakan sebagai dataset atau datauji.

3. Implementasi Jaringan Saraf Tiruan

Pada tahap implementasi Jaringan Saraf Tiruan menggunakan *train tool* pada aplikasi Matlab 2017b terhadap hasil *feature extraction* metode *Fixed Moving Window* yang disimpan dengan nama "dataLatih" sehingga Jaringan Saraf Tiruan dapat mengenali data latih. Kemudian untuk nilai target

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

pada data latih disimpan dengan nama "target" dapat dilihat pada tabel berikut :

TABLE III. CONFUSION MATRIX

Normal (1-17)	**DDoS (18-34)**
0	**1**
1	**0**

Pada saat pelatihan model Jaringan Saraf Tiruan, digunakan *Function Training Gradient Descent with Momentum & Adaptive LR (traingdx)* untuk mengenali data latih dan data uji. Arsitektur model Jaringan Saraf Tiruan menggunakan *single layer*.

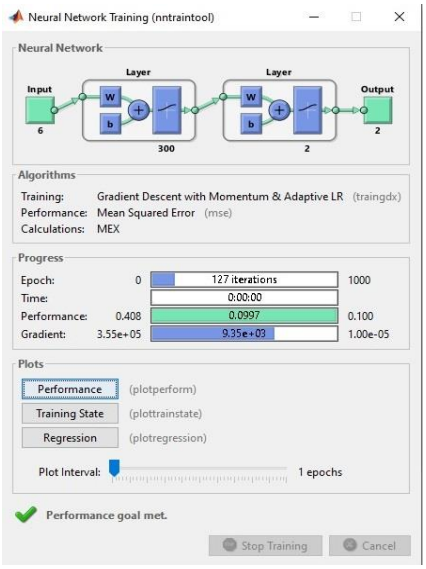

Fig. 8. Pelatihan Model Jaringan Saraf TIruan

Pada Gambar 8 ditunjukkan bahwa arsitektur Jaringan Saraf Tiruan telah melakukan proses pelatihan dan mencapai performance 0.0997 hasil terbaik. Setelah itu dilakukan pengujian pelatihan Jaringan Saraf Tiruan sehingga mendapatkan hasil nilai *input* sebanyak 6 nilai, satu *hidden layer* dengan *neuron* berjumlah 300 dan 2 *output* yang terdiri dari dataset normal dan dataset DDoS. Arsitektur model Jaringan Saraf Tiruan ini menggunakan *single hidden layer*, nilai *input* pada node akan langsung melakukan proses ke *hidden layer*, node *hidden layer* pada gambar menggunakan 300 node. Epoch pada pelatihan Jaringan Saraf Tiruan sebanyak 1000 iterasi yang artinya proses *training* akan dilakukan sebanyak 1000 kali dan *goal* parameter latih sebesar 0.1. Untuk normalisasi menggunakan fungsi min-max, dengan cara merubah data dari suatu *range* ke dalam *range* baru. Hubungan *hidden layer* dan *output* terjadi pada proses komputasi terhadap bobot dan bias serta dihitung juga besarnya *output* dari *hidden layer* ke *output* berdasarkan fungsi aktivasi *logsig* dengan rentang nilai 0 sampai 1.

4. Implementasi Jaringan Saraf Tiruan

Pada tahapan analisis hasil pengujian dilakukan dengan 2 jenis dataset yaitu dataset normal dan dataset DDoS. Setiap masing-masing jenis atau kelompok dataset memiliki jumlah 27 dataset berbentuk csv, sehingga total seluruh dataset adalah 54 dataset csv. Dataset tersebut kemudian dibagi menjadi 2 jenis yaitu dataset untuk "dataLatih" dan dataset untuk "dataUji", masing-masing berisikan dataset normal dan dataset DDoS, dataset untuk "dataLatih" berjumlah 34 dataset dan dataset untuk "dataUji" berjumlah 20 dataset.

TABLE IV. DETAIL JUMLAH DATASET PENGUJIAN

Jenis Data	Total Dataset	Total Data Latih	Total Data Uji
DdoS	27	17	10
Normal	27	17	10

Pada data uji terdapat 20 data untuk digunakan sebagai proses pengujian yang terdiri dari 10 data uji DDoS dan 10 data uji Normal. Pada tahap penentuan hasil pengujian Jaringan Saraf Tiruan menggunakan *function training Gradient Descent with Momentun & Adaptive LR* (traingdx) kemudian dilakukan perhitungan secara *confusion matrix* dengan rumus yang digunakan adalah *accuracy*, untuk mengetahui masing – masing nilai tersebut.

TABLE V. HASIL PENGUJIAN JARINGAN SARAF TIRUAN

		Actual Class	
		DdoS	Normal
Predicted class	DDoS	10	
	Normal	True Positive	1

KESIMPULAN

Berdasarkan hasil pengujian pada pengenalan serangan DDoS menggunakan metode Jaringan Saraf Tiruan (JST) yang telah dilakukan maka dapat ditarik kesimpulan, pada pengujian deteksi serangan DDoS menggunakan metode *fixed moving window* dan Jaringan Saraf Tiruan didapatkan *accuracy* sebesar 95% serta penggunaan *feature extraction fixed moving window* yang dikombinasikan dengan Jaringan Saraf Tiruan mampu mengenali data normal maupun data DDoS dengan baik.

REFERENCES

[1] Hermawan, R. (2013). Analisis Konsep Dan Cara Kerja Serangan Komputer Distributed Denial of Service (DDoS). *Faktor Exacta*, *5*(1), 1–14.

[2] Muhammad, A. W. (2016). Analisis Statistik Log Jaringan Untuk Deteksi Serangan DDoS Berbasis Neural Network. *ILKOM Jurnal Ilmiah*, *8*(3), 220. https://doi.org/10.33096/ilkom.v8i3.76.220-225.

[3] Aziz, M., Umar, R., & Ridho, F. (2019). Implemetasi Jaringan Saraf Tiruan Untuk Mendeteksi Serangan DDoS Pada Forensik Jaringan. *5341*(April), 3–9.

[4] Sutarti, Pancaro, Adi, P., & Saputra, Fembi, I. (2018). Implementasi IDS (Intrusion Detection System) Pada Sistem Keamanan Jaringan SMAN 1

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020

Cikeusal. *Jurnal PROSISKO*, *5*(1). Diambil dari http://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/download/584/592.

[5] Geges, S., & Wibisono, W. (2015). Pengembangan Pencegahan Serangan Distributed Denial of Service (Ddos) Pada Sumber Daya Jaringan Dengan Integrasi Network Behavior Analysis Dan Client Puzzle. *JUTI: Jurnal Ilmiah Teknologi Informasi*, *13*(1), 53. https://doi.org/10.12962/j24068535.v13i1.a388.

[6] Muhammad, A. W., Riadi, I., & Sunardi, S. (2017). Deteksi Serangan DDoS Menggunakan Neural Network dengan Fungsi Fixed Moving Average Window. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *1*(3), 115. https://doi.org/10.14421/jiska.2017.13-03.

[7] Irsyadi, F. Y. (2015). Pengembangan Aplikasi Sistem Monitoring Keamanan Berbasis Linux, Menggunakan Cctv Dan Sms Gateway. 1–15.

[8] Prathama, A. Y., Purnamasari, R. W., Ditakristy, M. L., Saepudin, D., Nhita, F., Informatika, F., … Mubarok, M. S. (2018). Model Estimasi Neural Network, Aplikasi Peramalan Tingkat Bagi Hasil Deposito Mudharabah Dengan Variabel Makroekonomi Sebagai Penentu Skripsi. *Jurnal Teknosains*, *2*(3), 1130–1139. https://doi.org/10.1016/j.compchemeng.2005.05.025.

[9] Burhanudin, M. F. (2018). Auto Moderasi Gambar Pornografi Menggunakan Neural Network. *Universitas Mercu Buana Yogyakarta*.

[10] Wuryandari, M. D., & Afrianto, I. (2012). Perbandingan Metode Jaringan Syaraf Tiruan Backpropagation Dan Learning Vector Quantization Pada Pengenalan Wajah. *Komputa*, *1*(1), 45–51.

[11] Ananto, R. P., Purwanto, Y., & Novianty, A. (2017). Deteksi Jenis Serangan pada Distributed Denial of Service Berbasis Clustering dan Classification Menggunakan Algoritma Minkowski Weighted K-Means dan Decision Tree. *37*(3), 193–203.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v9i3.945, Copyright ©2020
Submitted : 3 Agustus 2020, Revised : 7 Oktober 2020, Accepted : 8 Oktober 2020, Published : 12 Oktober 2020