Komputika: Jurnal Sistem Komputer
Volume 11, Nomor 1, April 2022, hlm. 67 - 76
Terakreditasi Peringkat 3, SK No. 28/E/KPT/2019
DOI: 10.34010/komputika.v11i1.5227

ISSN: 2252-9039 (print)
ISSN: 2655-3198 (online)

Analisis Pengamanan Jaringan Menggunakan Router Mikrotik dari Serangan DoS dan Pengaruhnya Terhadap Performansi

Arief Indriarto Haris[1*], Budhi Riyanto[2], Farry Surachman[3], Ardito Adi Ramadhan[4]

[1,2,3,4)]Pusat Teknologi Informasi dan Komunikasi Penerbangan dan Antariksa (Pustikpan),
Lembaga Penerbangan dan Antariksa Nasional (LAPAN)
Jalan Pemuda Persil 1, Jakarta Timur, 13220
**email: arief.indriarto@lapan.go.id*

(Naskah masuk: *16 Juli 2021*; diterima untuk diterbitkan: *02 September 2021*)

ABSTRAK *– Denial of Service (DoS) menjadi ancaman siber serius dan berdampak destruktif karena dapat melumpuhkan target dengan membanjirinya dengan traffic dalam jumlah besar. Router sebagai gateway dalam jaringan memegang peranan vital. Jika fungsinya terganggu, maka akan berdampak langsung terhadap performa jaringan. Penelitian ini menggunakan metodologi PPDIOO dan bertujuan untuk menganalisis pengamanan jaringan dari serangan DoS menggunakan router mikrotik dengan memanfaatkan fitur-fitur keamanan bawaan, lalu menilai tingkat keefektifannya. Serangan DoS diujikan terhadap enam kondisi router, lalu pada masing-masing kondisi dilakukan pengukuran terhadap lima indikator. Hasil yang diperoleh yaitu pengamanan dengan firewall raw pada kondisi 6 adalah yang paling efektif dan efisien dibandingkan kondisi lainnya. Konsumsi CPU berhasil diturunkan hingga 20% dan ping response time kembali ke kondisi normal, serta proses deteksi dan blokir bekerja secara otomatis. Namun secara keseluruhan, pengamanan dengan fitur-fitur keamanan bawaan dinilai tidak efektif dalam menghadapi serangan DoS. Hal ini dibuktikan dengan konsumsi CPU yang masih tinggi dan jauh dari normal, serta traffic DoS yang tidak dapat dinihilkan, hanya latensi yang dapat dinormalkan.*

Kata Kunci *– Denial of Service (DoS), Router, Firewall, Filter Rules, Raw*

Analysis of Securing Network Using Mikrotik Router from DoS Attacks and the Effect on Performance

ABSTRACT *– Denial of Service (DoS) is a serious threat and has a destructive impact because it can paralyze a target by flooding it with large amounts of traffic. Router as a gateway in the network plays a vital role. If the function is disturbed, it will have a direct impact on network performance. This study uses the PPDIOO methodology and aims to secure the network from DoS attacks using a Mikrotik router by utilizing the built-in security features, as well as assessing the level of effectiveness. DoS attacks were tested against six router conditions, then in each condition five indicators were measured. The results obtained are that security with firewall RAW in condition 6 is the most effective and efficient in other conditions. CPU consumption was reduced by 20% and ping response times returned to normal, and the detection and blocking process works automatically. But overall, security with the built-in security features that are considered ineffective in dealing with DoS attacks. This is evidenced by CPU consumption which is still high and far from normal, as well as DoS traffic that cannot be eliminated, only latency can be normalized.*

Keywords *- Denial of Service (DoS), Router, Firewall, Filter Rules, Raw*

1. PENDAHULUAN

Berdasarkan laporan insiden siber dari Gov-CSIRT Indonesia pada tahun 2019, di Indonesia terdata sebanyak 4241 aduan insiden siber terjadi di sektor pemerintah (domain *.go.id), dimana 3542 aduan terverifikasi yang terdiri dari *vulnerability*, *phishing*, *web defacement*, *malware*, dan insiden siber lainnya [1]. Berdasarkan laporan terkait serangan *Denial of Service* (DoS) yang dirilis oleh Securelist dari Kaspersky pada *Quarter* 1 tahun 2021, rata-rata serangan DoS per hari mencapai 1500 serangan,

dengan *traffic* tertinggi 800 GB per detik terjadi di sektor swasta, dan Amerika Serikat menjadi sumber serangan DoS terbesar (41,98%) dibanding dengan negara-negara lainnya [2]. Dari data-data tersebut menunjukan bahwa insiden siber dapat terjadi kapanpun, dimanapun, dan dapat mengancam pihak manapun, serta menghasilkan dampak negatif yang cukup besar, mulai dari sisi finansial, hingga reputasi organisasi.

DoS menjadi salah satu jenis serangan siber teratas dan cukup banyak digunakan oleh para *attacker* dengan tujuan untuk melumpuhkan targetnya. Serangan DoS menggunakan volume dan intensitas tertentu yang menyebabkan target menjadi kehabisan *resource* bahkan *down* ketika menangani permintaan layanan dari pengguna, sehingga membuat pengguna layanan yang sah kesulitan atau bahkan tidak dapat mengakses layanan [3]. Seiring dengan perkembangannya, DoS memiliki beberapa jenis tipe serangan, diantaranya SYN-*Flooding*, SMURF *Attack*, TCP-*Flooding*, UDP-*Flooding*, ICMP-*Flooding*, DNS-*Flooding* [4].

DoS memiliki beberapa model basis serangan, diantaranya adalah DoS berbasis *bandwidth*, dimana serangan DoS basis ini bekerja dengan mengirimkan *packet data* secara massal yang menyebabkan target menjadi *overload* dan kehabisan sumber daya *bandwidth* pada jaringan. Berikutnya adalah DoS berbasis lalu lintas jaringan, yang mana DoS basis ini membanjiri lalu lintas jaringan dengan sejumlah besar *packet* TCP, UDP, ICMP yang terlihat seolah-olah sah oleh target. Dan yang terakhir adalah DoS berbasis aplikasi, bekerja dengan memanfaatkan serangan DoS pada tingkat *layer* aplikasi (*layer* 7), seperti akses ke *database*, yang menyebabkan sumber daya pada *layer* aplikasi tersebut *overload* [5].

Router merupakan salah satu perangkat jaringan yang memungkinkan perangkat lain untuk terhubung kedalam jaringan *intranet* maupun *internet*. Selain itu *router* juga dapat menyimpan identitas lalu lintas *packet data* yang melewatinya, beserta dengan perpindahannya [6]. *Router* sering difungsikan sebagai *gateway* bagi jaringan *internal* agar dapat terhubung ke jaringan lain atau *internet*. Oleh karena itu, jika terjadi kendala pada *router* maka secara langsung akan berdampak besar terhadap performa jaringan [7].

Adapun *Router* Mikrotik merupakan *router* yang mencangkup *Operating System* (OS) berbasis Mikrotik dengan berbagai fitur handal didalamnya, salah satunya adalah fitur *Firewall* untuk menghadapi ancaman serangan siber [8]. Pada *firewall router* mikrotik, terdapat beberapa fitur yang dapat digunakan untuk mengamankan jaringan, diantaranya adalah *Firewall Filter Rules* dan *Firewall Raw*. *Firewall filter rules* dapat dimanfaatkan untuk memblokir aktifitas jaringan yang berpotensi membahayakan, seperti memblokir *website* tertentu, memblokir penggunaan aplikasi seperti Torrent, VPN, *port scanning*, hingga *recursive* DNS [4], [9]. *Firewall raw* juga dapat melakukan blokir seperti halnya dengan *firewall filter rules*, namun dengan konsumsi *resource* yang lebih hemat. Hal ini dikarenakan *firewall raw* memungkinkan melakukan *connection tracking,* sebelum memilih antara melewatkan atau memblokir *packet* [10].

Pada penelitian sebelumnya, menunjukkan DoS dapat dideteksi dengan bantuan beberapa *tools*. Pada penelitian yang dilakukan oleh [11], serangan DoS berhasil dideteksi dengan menggunakan *Intrusion Detection System* (IDS) berbasis Snort. Sedangkan pada penelitian yang dilakukan [12], [13], dengan menggunakan *tools honeypot* berbasis Honeyd, serangan DoS dapat dideteksi secara *real time* dengan memberikan peringatan berupa *log* yang berisikan informasi serangan yang sedang terjadi, serta dapat mensimulasikan atau menduplikasi target, sehingga dapat mengecoh *attacker,* dengan membuat seolah-olah target yang diserang adalah target yang asli.

Selain dapat dideteksi, serangan DoS juga dapat diidentifikasi, salah satunya menggunakan *machine learning* dengan algoritma K-Nearest Neighbor (KKN), melalui serangkaian *dataset* yang bertujuan untuk meningkatkan tingkat akurasi [14]. Selain deteksi dan identifikasi, serangan DoS juga dapat dimitigasi dengan suatu pendekatan tertentu. Pada penelitian [15], tindakan mitigasi dilakukan dengan pendekatan dalam *hardening* yaitu *Defense-through-deception*, yang merupakan peningkatan dari pendekatan pendahulunya yaitu *defense-in-depth*. *Defense-through-deception* menerapkan perlindungan berlapis dan dapat memberikan *delay* bagi *attacker* pada saat menyerang *target*, sehingga memberikan waktu bagi *network administrator* untuk melakukan tindakan perlindungan.

Pada penelitian lainnya terkait serangan DoS terhadap *router* mikrotik, serangan DoS bertipe TCP *flooding* dengan perubahan *data size* diujikan terhadap router mikrotik. Hasil yang diperoleh adalah *router* mikrotik mengalami peningkatan konsumsi *resource*, yaitu di sisi daya listrik dan beban CPU, namun pada penelitian ini tidak memberikan solusi terkait perlindungan dari serangan DoS [16]. Pada penelitian [10], serangan DoS dengan tipe DNS *flooding* diujikan terhadap *router* mikrotik. Pada *router* dilakukan konfigurasi pengamanan dari serangan DoS menggunakan fitur *firewall filter rules* dan *firewall* raw, namun terdapat beberapa metode yang masih perlu penyesuaian dan dapat lebih dioptimalkan. Beberapa diantaranya adalah algoritma konfigurasi pengamanan dari serangan DoS yang masih belum optimal, terlihat dari konfigurasi *firewall* raw yang memblokir protocol TCP, UDP, dan ICMP. Hal ini akan membuat semua perangkat lain tidak dapat

terhubung ke jaringan karena *protocol* tersebut ditutup. Hal lainnya adalah durasi pengujian serangan DoS yang tidak diketahui, indikator yang diukur juga terbatas pada CPU, *memory*, dan *traffic* DoS. Pengukuran pada tiap indikator terlihat hanya diukur di waktu tertentu saja, bukan berdasarkan rata-rata nilai dari masing-masing indikator yang dihasilkan pada saat pengujian selama durasi waktu tertentu. Dikarenakan beberapa hal tersebut, membuat hasil akhir yang disimpulkan pada penelitian tersebut (menyatakan bahwa metode pengamanan yang dimaksud adalah efektif dalam menangani serangan DoS) menjadi bias, sehingga perlu sekiranya untuk dilakukan penelitian lanjutan dengan beberapa penyesuaian dan peningkatan.

Terkait fungsi dan perannya yang vital didalam jaringan, menyebabkan *router* sangat berpotensi besar untuk dijadikan sebagai *target* serangan DoS. Maka dalam penelitian ini, penulis bermaksud menganalisis pengamanan jaringan dari serangan DoS menggunakan *router* mikrotik. Melalui penelitian ini, diharapkan dapat diperoleh pengamanan jaringan yang optimal dengan memanfaatkan fitur-fitur keamanan bawaan pada *router* mikrotik, serta mengetahui tingkat keefektifannya berdasarkan pengaruhnya terhadap performa dari *router* tersebut.

Penulis melakukan pengujian serangan DoS terhadap *router* mikrotik yang dikonfigurasi dalam beberapa kondisi. Setiap kondisi diuji secara bergantian dalam durasi waktu tertentu, lalu dipantau dan dicatat menggunakan *tools monitoring* tambahan untuk mengukur nilai dari tiap-tiap indikator yang menggambarkan performa dari *router* tersebut. Metodologi yang digunakan pada penelitian ini adalah PPDIOO, dengan tujuan agar setiap tahapan yang dilakukan dapat berjalan konsisten, sistematis, dan berkelanjutan.

2. METODE DAN BAHAN

PPDIOO adalah singkatan dari *Prepare*, *Plan*, *Design*, *Implement*, *Operate*, dan *Optimize*. PPDIOO merupakan metodologi perancangan dan pengembangan jaringan yang didesain oleh Cisco, dimana setiap tahapannya mendefinisikan *life-cycle* yang berkelanjutan [17], [18]. Adapun deskripsi aktifitas yang dilakukan pada tiap tahapannya adalah sebagai berikut:

A. *Prepare*

Mendefinisikan kebutuhan sistem, diantaranya adalah mengidentifikasi resource hardware untuk memenuhi kebutuhan sistem dari tools utama dan pendukung. Pada penelitian ini, dibutuhkan perangkat *router* mikrotik, *server monitoring*, *switch*, dan *tools* untuk DoS. Semua perangkat tersebut berjalan dalam suatu *environment virtual* diatas *Hypervisor* berbasis *Kernel Based Virtual Machine* (KVM) (Gambar 1). Adapun spesifikasi dari masing-masing perangkat terdapat pada Tabel 1.

Tabel 1. Spesifikasi dan Peran Perangkat

Perangkat	Spesifikasi	Keterangan
Router Mikrotik	OS: MikroTik CHR 6.48 CPU: 1 Core @3,3GHz *Memory*: 224 MB *Disk*: 63,5 MB	Perangkat untuk dijadikan target dari serangan DoS.
Tools DoS	OS: Kali Linux *Tools*: Hping3 CPU: 2 Core @3,3GHz *Memory*: 3 GB *Disk*: 32 GB	Perangkat untuk melakukan serangan DoS ke target.
Server Monitoring	OS: Ubuntu Server 18.04 *Tools*: Zabbix CPU: 1 Core @3,3GHz *Memory*: 2 GB *Disk*: 50 GB	Perangkat untuk melakukan pemantauan dan perekaman dari *resource* target yang diserang DoS.
Switch	*Tools*: Linux Bridge	Perangkat untuk menghubungkan *router* mikrotik, *tools* DoS, dan *server monitoring*.

Gambar 1. Server Hypervisor

B. *Plan*

Merancang skenario pengujian dan kondisi pengamanan. Adapun untuk skenario penyerangan DoS, penulis menggunakan TCP flooding. Berdasarkan laporan serangan DoS pada *Quarter* 1 tahun 2021 yang dirilis oleh Securelist dari

Kaspersky, serangan DoS dilakukan dengan durasi rata-rata dibawah 4 jam [2]. Oleh karena itu, penulis mengambil durasi pengujian dan monitoring dari serangan DoS selama 1 jam terhadap masing-masing kondisi pengamanan, dengan asumsi bahwa nilai yang dihasilkan dari serangan DoS dapat terlihat dampaknya dan stabil trennya didalam *tools monitoring*, sehingga menghasilkan nilai dengan tingkat akurasi yang tepat dari tiap indikator.

Sedangkan kondisi pengamanan yang diujikan terhadap serangan DoS adalah berjumlah enam kondisi, diantaranya adalah:

a. Kondisi normal sebelum DoS (Kondisi 1). Pada kondisi ini, *router* mikrotik dimonitoring untuk mengetahui konsumsi *resource* ketika dalam keadaan normal.
b. Kondisi tanpa pengamanan dari DoS (Kondisi 2). Di kondisi ini, serangan DoS diujikan terhadap *router* mikrotik dengan kondisi tanpa pengamanan.
c. Kondisi dengan pengamanan dari serangan DoS menggunakan *Firewall Filter Rules. Router* diamankan menggunakan fitur *firewall filter rules*, dengan dua metode, yaitu:
    a) Pengamanan dari serangan DoS dengan proses deteksi dilakukan secara manual, sementara proses blokir dilakukan menggunakan *filter rules* (Kondisi 3).
    b) Pengamanan dari serangan DoS dengan proses deteksi dan blokir dilakukan menggunakan *filter rules* (Kondisi 4).
d. Kondisi dengan pengamanan dari DoS menggunakan *Firewall* Raw. *Router* diamankan menggunakan fitur *firewall* raw, dengan dua metode, yaitu:
    a) Pengamanan dari serangan DoS dengan proses deteksi dilakukan secara manual, sementara proses blokir dilakukan menggunakan dengan Raw (Kondisi 5).
    b) Pengamanan dari serangan DoS dengan proses deteksi dan blokir menggunakan Raw (Kondisi 6).

C. *Design*

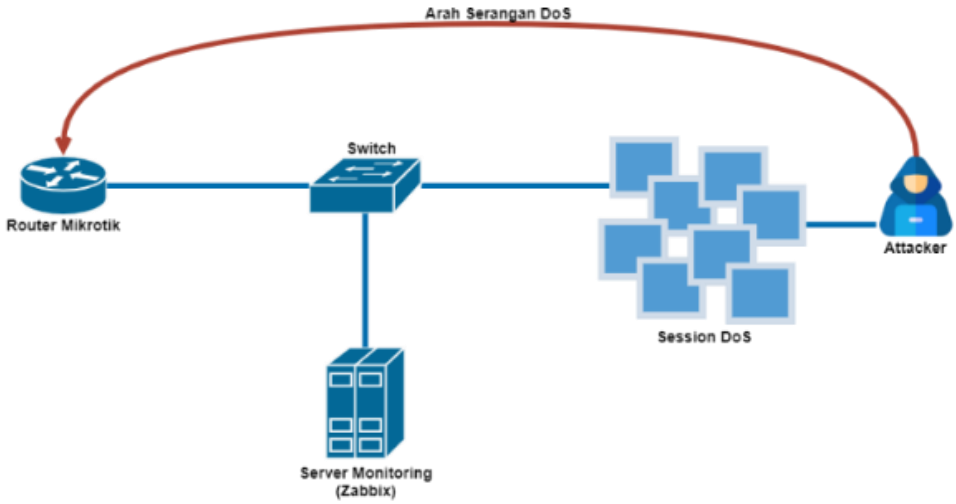
Mendesain skema pengujian jaringan dari proses pengujian serangan DoS terhadap *router* mikrotik (Gambar 2).

D. *Implement*

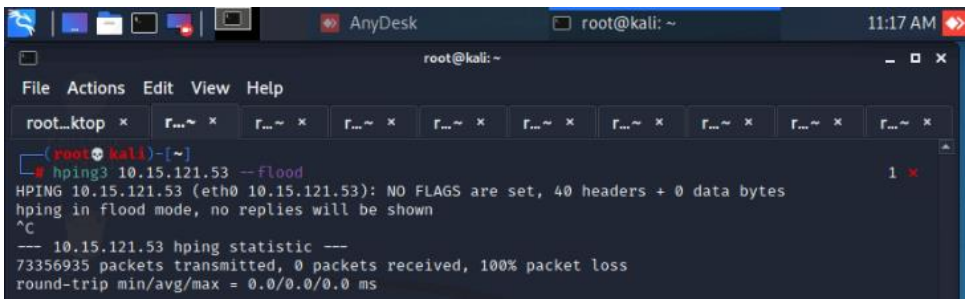
Mengimplementasikan semua hal berdasarkan desain yang dirancang di tahap sebelumnya, diantaranya adalah konfigurasi pengamanan pada router dan konfigurasi dasar operasional router lainnya.

E. *Operate*

Melakukan pengujian serangan DoS terhadap *router* mikrotik pada enam kondisi, lalu dilakukan pengukuran terhadap beberapa indikator. Beberapa



Gambar 2. Skema Pengujian



Gambar 3. DoS dengan Hping3

indikator yang diukur diantaranya adalah CPU, *memory*, *disk*, *ping response time*, dan *traffic* DoS. Indikator CPU, *memory*, dan *disk* menggambarkan terkait konsumsi *resource* komputasi pada *router*. *Ping response time* menggambarkan latensi atau ketersediaan dan kemudahan akses *router* di dalam jaringan, sedangkan *traffic* menggambarkan besaran *traffic* yang dikirimkan oleh DoS ke target. Pengujian dilakukan menggunakan Kali Linux dengan *tools* Hping3. Serangan DoS yang dilakukan berjumlah 10 *session* dan bertipe TCP *flooding* (Gambar 3).

F. *Optimize*

Mengidentifikasi dan menganalisis hasil yang didapatkan dari proses pengujian. Data diperoleh dari tools Zabbix (Gambar 4) dan diolah secara statistik, serta disajikan dalam bentuk grafik. Dengan *tools* ini, memungkinkan penulis untuk mengetahui data terkait besaran nilai yang dihasilkan dari tiap indikator dalam periode waktu tertentu.

3. HASIL DAN PEMBAHASAN

Router mikrotik dikonfigurasi pada beberapa kondisi dan metode. Selanjutnya secara bergantian diujikan dengan serangan DoS dan dicatat hasil pengujiannya. Adapun konfigurasi yang dilakukan pada masing-masing kondisi adalah sebagai berikut:

A. Kondisi normal sebelum DoS (Kondisi 1)

R*outer* hanya dikonfigurasi untuk menjalankan fungsi standar. Hal ini bertujuan untuk melihat konsumsi *resource router* pada keadaan normal dan tanpa adanya serangan DoS. Konfigurasi yang dilakukan seperti pengalamatan IP, DNS, *routing*. Tidak ada pengamanan yang dikonfigurasikan pada *router*.

B. Kondisi tanpa pengamanan dari DoS (Kondisi 2)

Serangan DoS diujikan terhadap *router* yang dalam keadaan normal dan tanpa adanya pengamanan.

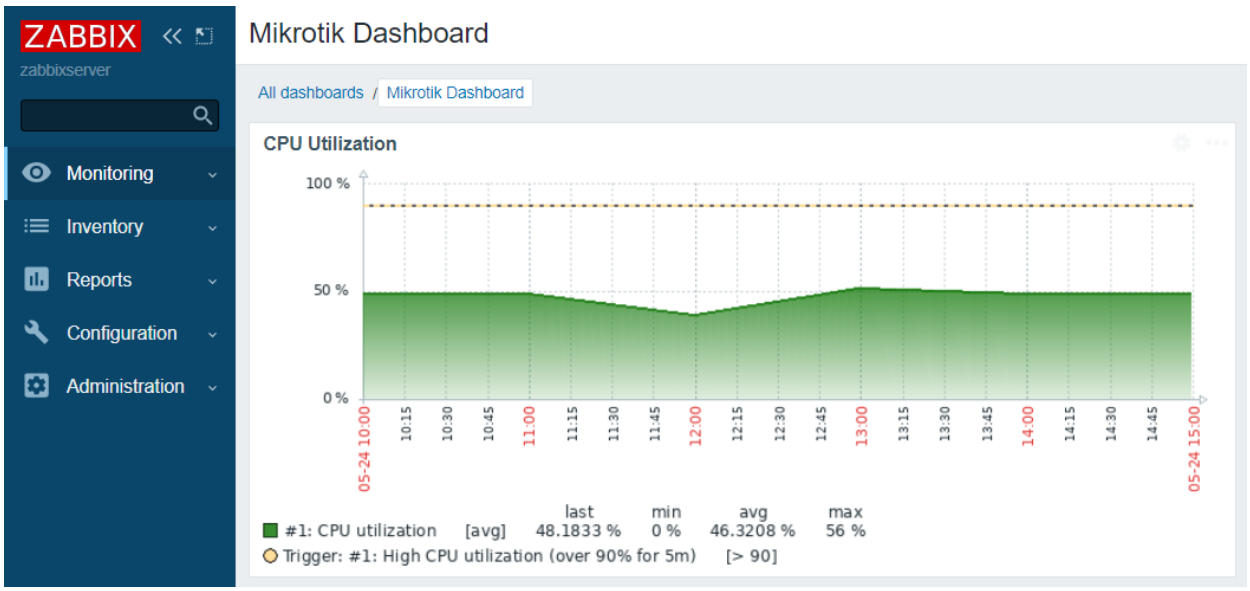
C. Kondisi pengamanan menggunakan *Firewall Filter Rules*

Pada kondisi ini terdapat dua metode pengamanan:

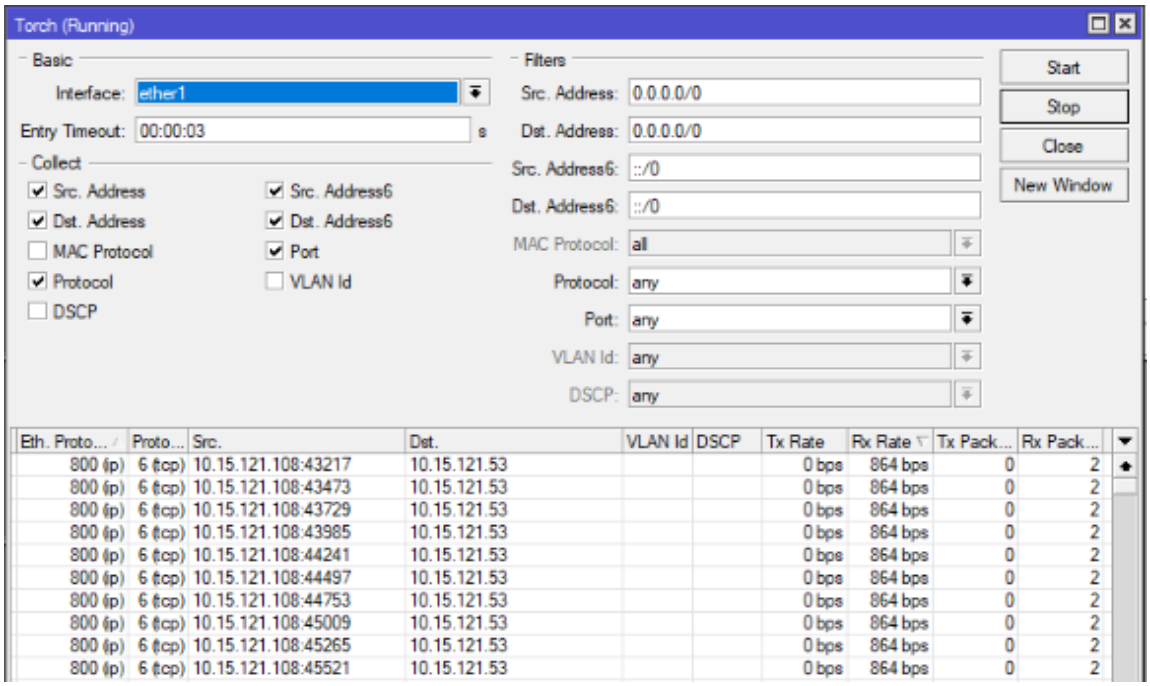
a. Pengamanan dengan proses deteksi dilakukan secara manual, sementara proses blokir dilakukan menggunakan *filter rules* (Kondisi 3). Deteksi terhadap serangan DoS dilakukan secara manual dengan fitur *Torch* (Gambar 5). Setelah serangan DoS dilakukan, fitur *torch* digunakan untuk mendeteksi IP *address* penyerang. Ketika IP *address* penyerang diketahui, tindakan pengamanan dilakukan adalah menambahkan *rule* untuk memblokir serangan DoS (Gambar 6). Adapun konfigurasi yang dilakukan terdapat pada Tabel 2.

Tabel 2. *Rule* Blokir dengan *Firewall Filter Rules*

Rule	Keterangan
chain=input action=drop src-address=xxx.xxx.xxx.xxx log=no log-prefix=""	Segala *traffic* masuk yang berasal dari IP *address* sumber serangan DoS akan diblokir.



Gambar 4. *Monitoring* dengan Zabbix



Gambar 5. Deteksi dengan *Torch*

```
[labtik@MikroTik] > ip firewall filter print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=input action=drop src-address=[redacted] log=
      log-prefix=""
```

Gambar 6. Implementasi Pengamanan pada Kondisi 3

b. Pengamanan dengan proses deteksi dan blokir menggunakan *filter rules* (Kondisi 4). Tindakan deteksi dan blokir dilakukan dengan menggunakan *filter rules*, sehingga proses untuk deteksi dan blokir menjadi terotomasi (Gambar 7). Adapun konfigurasinya terdapat di Tabel 3.

Tabel 3. *Rule* Deteksi dan Blokir dengan *Firewall Filter Rules*

Rule	Keterangan
chain=input action=jump jump-target=ddos-rule log=no log-prefix=""	Segala *traffic* yang masuk akan ditandai dan diarahkan ke *rule* berikutnya.
chain=ddos-rule action=return dst-limit=32,32,src-and-dst-addresses/1s log=no log-prefix=""	*Rule* ini untuk melakukan deteksi serangan DoS. *Traffic* yang ditandai, akan disesuaikan dengan *parameter destination limit* berdasarkan *source* dan *destination address* dengan nilai *rate* dan *burst* sebesar 32 *packet* per 1 detik. Jika nilainya sama atau lebih besar, maka dilanjutkan ke *rule* berikutnya. Namun jika kurang dari itu, maka diabaikan.
chain=ddos-rule action=add-src-to-address-list address-list=ddoser-rule address-list-timeout=10m log=no log-prefix=""	*Source* IP *address* pada t*raffic* yang sesuai dengan *rule* deteksi menunjukkan IP *address* dari penyerang, berikutnya dimasukan ke dalam *address list* ddoser-rule, dan akan kadaluarsa selama 10 menit.
chain=ddos-rule action=add-dst-to-address-list address-list=ddosed-rule address-list-timeout=10m log=no log-prefix=""	*Destination* IP *address* pada t*raffic* yang sesuai dengan *rule* deteksi menunjukkan IP *address* dari *target* serangan DoS, berikutnya dimasukan ke dalam *address list* ddosed-rule, dan akan kadaluarsa selama 10 menit.
chain=ddos-rule action=drop src-address-list=ddoser-rule dst-address-list=ddosed-rule log=no log-prefix=""	*Traffic* dengan *source address list* dari ddoser-rule dan *destination address list* ke ddosed-rule maka diblok.

```
1   chain=input action=jump jump-target=ddos-rule log=no
    log-prefix=""

2   chain=ddos-rule action=return
    dst-limit=32,32,src-and-dst-addresses/1s log=no
    log-prefix=""

3   chain=ddos-rule action=add-src-to-address-list
    address-list=ddoser-rule address-list-timeout=10m
    log=no log-prefix=""

4   chain=ddos-rule action=add-dst-to-address-list
    address-list=ddosed-rule address-list-timeout=10m
    log=no log-prefix=""

5   chain=ddos-rule action=drop src-address-list=ddoser-rule
    dst-address-list=ddosed-rule log=no log-prefix=""
[labtik@MikroTik] >
```

Gambar 7. Implementasi Pengamanan pada Kondisi 4

D. Kondisi pengamanan menggunakan *Firewall* Raw

Seperti halnya pengamanan dengan *firewall filter rules*, pengamanan dengan *firewall* Raw juga menggunakan dua metode yang sama, namun dengan beberapa penyesuaian konfigurasi. Pada *firewall* Raw di bagian parameter *chain*, tersedia fitur *prerouting*. Penulis menggunakan fitur tersebut, sehingga memungkinkan *action* dilakukan sebelum *connection tracking* maka dapat menghemat *resource*. Adapun konfigurasi pada masing-masing metode pengamanan menggunakan *firewall* Raw, antara lain:

a. Pengamanan dengan proses deteksi dilakukan secara manual, sementara proses blokir dilakukan menggunakan dengan Raw (Kondisi 5). Deteksi serangan DoS menggunakan fitur *Torch*, sedangkan untuk proses blokir menggunakan Raw (Gambar 8) dengan konfigurasi pada Tabel 4.

Tabel 4. *Rule* Blokir dengan *Firewall* Raw

Rule	Keterangan
chain=prerouting action=drop log=no log-prefix="" src-address=xxx.xxx.xxx.xxx	*Traffic* yang berasal dari IP *address* sumber serangan DoS akan diblokir, sebelum *connection tracking*

```
[labtik@MikroTik] > ip firewall raw print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=prerouting action=drop log=no log-prefix=""
      src-address=
```

Gambar 8. Implementasi Pengamanan pada Kondisi 5

b. Pengamanan dengan proses deteksi dan blokir menggunakan Raw (Kondisi 6). Proses deteksi dan blokir terhadap serangan DoS dilakukan secara otomatis oleh Raw (Gambar 9). Konfigurasi yang dilakukan tersedia pada Tabel 5.

Tabel 5. *Rule* Deteksi dan Blokir dengan *Firewall* Raw

Rule	Keterangan
chain=prerouting action=jump jump-target=ddos-raw log=no log-prefix=""	Sebelum *connection tracking*, semua *traffic* yang masuk akan ditandai ddos-raw dan dilanjutkan ke *rule* berikutnya.
chain=ddos-raw action=return dst-limit=32,32,src-and-dst-addresses/1s log=no log-prefix=""	*Rule* ini untuk melakukan deteksi serangan DoS. *Traffic* yang ditandai, akan disesuaikan dengan *parameter destination limit* berdasarkan *source* dan *destination address* dengan nilai *rate* dan *burst* sebesar 32 *packet* per 1 detik. Jika nilainya sama atau lebih besar, maka dilanjutkan ke *rule* berikutnya. Jika tidak, maka diabaikan.
chain=ddos-raw action=add-src-to-address-list log=no log-prefix="" address-list=ddoser-raw address-list-timeout=10m	*Source* IP *address* pada *traffic* yang sesuai dengan *rule* deteksi menunjukkan IP *address* dari penyerang, berikutnya dimasukan ke dalam *address list* ddoser-raw, dan akan kadaluarsa selama 10 menit.
chain=ddos-raw action=add-dst-to-address-list log=no log-prefix="" address-list=ddosed-raw address-list-timeout=10m	*Destination* IP *address* pada *traffic* yang sesuai dengan *rule* deteksi menunjukkan IP *address* dari *target* serangan DoS, berikutnya dimasukan ke dalam *address list* ddosed-raw, dan akan kadaluarsa selama 10 menit.
chain=ddos-raw action=drop log=no log-prefix="" src-address-list=ddoser-raw dst-address-list=ddosed-raw	*Traffic* dengan *source address list* berasal dari ddoser-raw dan *destination address list* ke ddosed-raw maka akan diblokir.

```
1   chain=prerouting action=jump jump-target=ddos-raw log=n
    log-prefix=""

2   chain=ddos-raw action=return
    dst-limit=32,32,src-and-dst-addresses/1s log=no
    log-prefix=""

3   chain=ddos-raw action=add-src-to-address-list log=no
    log-prefix="" address-list=ddoser-raw
    address-list-timeout=10m

4   chain=ddos-raw action=add-dst-to-address-list log=no
    log-prefix="" address-list=ddosed-raw
    address-list-timeout=10m

5   chain=ddos-raw action=drop log=no log-prefix=""
    src-address-list=ddoser-raw
    dst-address-list=ddosed-raw
[labtik@MikroTik] >
```

Gambar 9. Implementasi Pengamanan pada Kondisi 6

Pengujian DoS bertipe TCP *flooding* secara bergantian dilakukan pada tiap kondisi dengan durasi selama 1 jam, bersamaan dilakukan juga pemantauan dan pencatatan terhadap nilai rata-rata dari masing-masing indikator yang diukur. Adapun rekapitulasi hasil pengambilan dan pengolahan data dari pengujian tersebut terdapat pada Gambar 10.

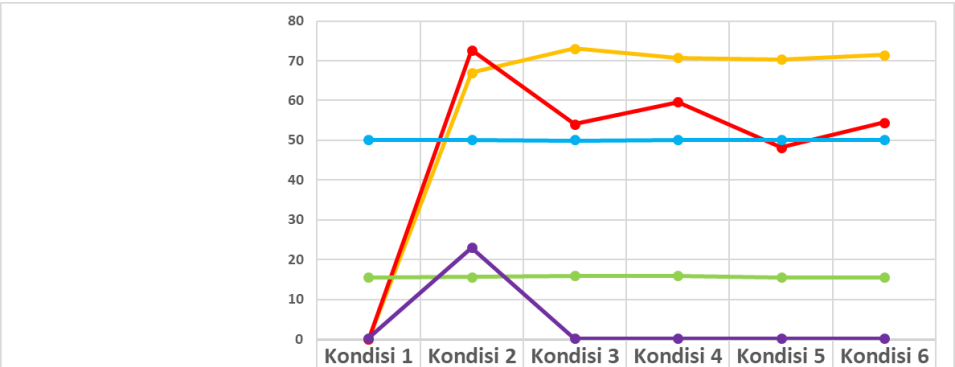
Pada indikator *traffic* dari kondisi 1 ke kondisi 2 hingga kondisi 6, terlihat terjadi peningkatan yang sangat signifikan. Hal ini menggambarkan adanya serangan DoS yang membanjiri target dengan *traffic* dalam jumlah sangat besar. Walaupun sudah dilakukan pengamanan terhadap *router* mikrotik seperti pada kondisi 3 hingga kondisi 6, tetap tidak dapat memberikan perubahan yang signifikan dalam menurunkan nilai *traffic* hingga ke kondisi normal seperti pada kondisi 1, dimana *traffic* tetap dalam nilai yang tinggi pada kisaran nilai 70Mbps.

Berikutnya pada indikator CPU, jika dibandingkan antara kondisi 1 dengan kondisi 2 hingga kondisi 6, terlihat peningkatan yang sangat signifikan. Hal tersebut terjadi dikarenakan sejumlah besar *request* yang dihasilkan oleh serangan DoS, sehingga membuat konsumsi CPU meningkat pesat untuk melayani *request* tersebut. Pada sisi lain, jika dibandingkan antara kondisi 2 dengan kondisi 3 hingga kondisi 6 yang sudah dilakukan pengamanan, nilai konsumsi terhadap CPU mengalami penurunan berkisar 10% hingga 20%. Namun tidak sampai berhasil menurunkan nilainya hingga ke keadaan normal seperti pada kondisi 1.

Pada indikator *ping response time*, dari kondisi 1 ke kondisi 2 mengalami peningkatan yang signifikan, berkisar 22ms. Hal ini pula disebabkan oleh adanya serangan DoS yang membanjiri target dengan sejumlah besar *packet* sehingga membuat latensi ikut meningkat. Dampak ini juga terlihat dari koneksi ke target yang menjadi *intermittent* (tidak stabil), sehingga menyebabkan *router* mikrotik sulit diakses. Namun ketika dilakukan pengamanan seperti pada kondisi 3 hingga 6, terjadi penurunan yang signifikan jika dibandingkan dengan kondisi 2. Bahkan pada kondisi 3 hingga kondisi 6, nilai *ping response time* dapat kembali pada keadaan normal (Kondisi 1).

Adapun untuk indikator lainnya yaitu *memory* dan *disk*, dari kondisi 1 hingga kondisi 6 tidak menunjukan perubahan yang berarti. Dengan kata lain, DoS dengan tipe TCP *flooding* tidak memberikan dampak yang signifikan terhadap konsumsi *router* dari sisi *memory* dan *disk*.

Perbandingan pengamanan router mikrotik dari kondisi 3 hingga kondisi 6, hanya pada indikator CPU yang menunjukan perbedaan yang cukup kontras, dimana pada kondisi 5 memiliki nilai CPU yang paling rendah dibandingkan kondisi lain, dengan keberhasilan menurunkan nilai konsumsi CPU sebesar 20% dari kondisi 2. Perbedaan lainnya terdapat pada proses deteksi, dimana proses deteksi pada kondisi 3 dilakukan secara manual, sedangkan



	Kondisi 1	Kondisi 2	Kondisi 3	Kondisi 4	Kondisi 5	Kondisi 6
Traffic (Mbps)	0,0087	67,03	73,12	70,8	70,4	71,48
CPU (%)	0	72,71	54,08	59,68	48,18	54,48
Memory (%)	15,57	15,61	15,92	15,97	15,55	15,55
Disk (%)	50,13	50,13	50,12	50,13	50,13	50,13
Ping Response Time (ms)	0,27	22,99	0,14	0,14	0,14	0,13

Gambar 10. Rekapitulasi Hasil Pengujian

pada kondisi 6 dilakukan secara otomatis menggunakan *raw*, sehingga lebih efisien dan membuat kondisi 6 menjadi yang paling baik dalam mengamankan diantara kondisi 3, 4, dan 5. Adapun kondisi 4 memiliki tingkat keefektifan yang paling rendah diantara kondisi pengamanan lainnya, terbukti dari konsumsi CPU hanya mampu diturunkan sekitar 10%.

4. KESIMPULAN

Serangan DoS terbukti memberikan dampak destruktif terhadap target, dimana konsumsi *resource* target mengalami peningkatan yang sangat signifikan, terutama dari sisi CPU dan latensi, hal ini sejalan dengan penelitian sebelumnya [10], [12], [13], [16]. Diantara semua kondisi pengamanan pada *router* mikrotik, kondisi 6 merupakan yang paling efektif dan efisien. Hal ini terbukti dari konsumsi CPU berhasil diturunkan hingga 15%, *ping response time* turun kembali ke kondisi normal, dan proses deteksi serta blokir dapat dilakukan oleh *firewall raw* secara otomatis.

Namun, secara keseluruhan pengamanan yang dilakukan menggunakan fitur bawaan router mikrotik (*firewall filter rules* dan *firewall raw*) menunjukan tingkat efektifitas yang cukup rendah dalam menghadapi serangan DoS. Hal tersebut dibuktikan dari tiap kondisi pengamanan (kondisi 3 hingga kondisi 6), dimana nilai CPU yang masih tinggi dan masih jauh dari kondisi normal, serta *traffic* yang dikirimkan oleh DoS tidak dapat dinihilkan, hanya dari sisi latensi yang dapat kembali ke kondisi normal. Oleh sebab itu, untuk mengamankan jaringan dari serangan DoS tidak cukup hanya dengan *router*, tapi juga membutuhkan dukungan perangkat tambahan lainnya.

Pada penelitian berikutnya disarankan untuk melakukan penelitian terkait hal-hal seperti metode, arsitektur, ataupun integrasi dengan perangkat tambahan lainnya yang memungkin untuk menghadapi serangan DoS, baik dalam hal deteksi maupun mitigasi.

UCAPAN TERIMA KASIH

Penulis sangat mengapresiasi dan berterima kasih atas dukungan dan keterlibatan seluruh pihak, khususnya kepada Kepala Pusat Teknologi Informasi dan Komunikasi Penerbangan dan Antariksa LAPAN yang telah memfasilitasi kegiatan penelitian ini.

DAFTAR PUSTAKA

[1] GOV-CISRT, "Laporan GOV-CSIRT 2019," 2019. [Daring]. Tersedia pada: https://govcsirt.bssn.go.id/laporan-tahunan-gov-csirt-2019/.

[2] A. Gutnikov, O. Kupreev, dan E. Badovskaya, "DDoS attacks in Q1 2021," *Securelist*, 2021. https://securelist.com/ddos-attacks-in-q1-2021/102166/ (diakses Mei 17, 2021).

[3] A. Fadlil, I. Riadi, dan S. Aji, "Review of Detection DDOS Attack Detection Using Naive Bayes Classifier for Network Forensics," *Bull. Electr. Eng. Informatics*, vol. 6, no. 2, hal. 140–148, 2017, doi: 10.11591/eei.v6i2.605.

[4] D. Aprilianto, T. Fadila, dan M. A. Muslim, "Sistem Pencegahan UDP DNS Flood dengan Filter Firewall pada Router Mikrotik," *Techno.Com*, vol. 16, no. 2, hal. 114–119, 2017, doi: 10.33633/tc.v16i2.1291.

[5] S. Geges dan W. Wibisono, "Pengembangan Pencegahan Serangan Distributed Denial of Service (DDoS) pada Sumber Daya Jaringan dengan Integrasi Network Behavior Analysis Dan Client Puzzle," *JUTI J. Ilm. Teknol. Inf.*, vol. 13, no. 1, hal. 53–67, 2015, doi: 10.12962/j24068535.v13i1.a388.

[6] F. Ridho, A. Yudhana, dan I. Riadi, "Analisis Forensik Router Untuk Mendeteksi Serangan Distributed Danial of Service (DDoS) Secara Real Time," 2016, vol. 2, no. 1, hal. 111–116.

[7] R. Pambudi dan M. A. Muslim, "Implementasi Policy Base Routing dan Failover Menggunakan Router Mikrotik untuk Membagi Jalur Akses Internet di FMIPA Unnes," *J. Teknol. dan Sist. Komput.*, vol. 5, no. 2, hal. 57, 2017, doi: 10.14710/jtsiskom.5.2.2017.57-61.

[8] A. Muzakir dan M. Ulfa, "Analisis Kinerja Packet Filtering Berbasis Mikrotik Routerboard Pada Sistem Keamanan Jaringan," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 10, no. 1, hal. 15–20, 2019, doi: 10.24176/simet.v10i1.2646.

[9] E. S. R. O. B. Langobelen, Y. Rachmawati, dan C. Iswahyudi, "Analisis Dan Optimasi Dari Simulasi Keamanan Jaringan Menggunakan Firewall Mikrotik Studi Kasus Di Taman Pintar Yogyakarta," *J. JARKOM*, vol. 7, no. 2, hal. 95–102, 2019.

[10] B. Jaya, Y. Yunus, dan S. Sumijan, "Peningkatan Keamanan Router Mikrotik Terhadap Serangan Denial of Service (DoS)," *J. Sistim Inf. dan Teknol.*, vol. 2, no. 4, hal. 5–9, 2020, doi: 10.37034/jsisfotek.v2i4.81.

[11] F. Ridho, A. Yudhana, dan I. Riadi, "Implementasi Log dalam Forensik Router Terhadap Serangan Distributed Denial of Service(DDoS)," *J. TIMES*, vol. VI, no. 2, hal. 15–21, 2017.

[12] B. Mardiyanto, T. Indriyani, dan I. M. Suartana, "Analisis dan Implementasi Honeypot dalam Mendeteksi Serangan Distributed Denial-Of-Services (DDOS) pada Jaringan Wireless,"

Integer J., vol. 1, no. 2, hal. 32–42, 2016.

[13] S. Dwiyatno, A. P. Sari, A. Irawan, dan S. Safig, "Pendeteksi Serangan DDoS (Distributed Denial of Service) Menggunakan Honeypot di PT. Torini Jaya Abadi," *J. Sist. Inf. dan Inform.*, vol. 2, no. 2, hal. 64–80, 2019, doi: 10.47080/simika.v2i2.606.

[14] M. M. Azis, Y. Azhar, dan Saifuddin, "Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network(SDN)," *J. Repos.*, vol. 2, no. 7, hal. 915–922, 2020, doi: 10.22219/repositor.v2i7.762.

[15] M. A. Naagas, E. L. Mique, T. D. Palaoag, dan J. S. Dela Cruz, "Defense-through-deception Network Security Model: Securing University Campus Network from DOS/DDOS Attack," *Bull. Electr. Eng. Informatics*, vol. 7, no. 4, hal. 593–600, 2018, doi: 10.11591/eei.v7i4.1349.

[16] R. Adrian dan H. N. Isnianto, "Analisa Pengaruh Variasi Serangan DDoS pada Performa Router," in *Prosiding Seminar Nasional Teknologi Terapan SV UGM 2016*, 2016, vol. 6, no. November, hal. 1257–1259, [Daring]. Tersedia pada: https://www.researchgate.net/profile/Ronald_Adrian/publication/311219100_ANALISA_PENGARUH_VARIASI_SERANGAN_DDOS_PADA_PERFORMA_ROUTER/links/583f7f0608ae2d217557e6cf.pdf.

[17] B. Sivasubramaniam, E. Frahim, dan R. Froom, "Analyzing the Cisco Enterprise Campus Architecture _ Introduction to Enterprise Campus Network Design," *Cisco Press*, 2010. https://www.ciscopress.com/articles/article.asp?p=1608131&seqNum=3 (diakses Mei 17, 2021).

[18] D. Yuliana dan I. K. A. Mogi, "Computer Network Design Using PPDIOO Method With Case Study of SMA Negeri 1 Kunir," *J. Elektron. Ilmu Komput. Udayana*, vol. 9, no. 2, hal. 235–240, 2020.