OPTIMASI PORT KNOCKING DAN HONEYPOT MENGGUNAKAN IPTABLES SEBAGAI KEAMANAN JARINGAN PADA SERVER

(*Port Knocking and Honeypot Optimization using IPTables for Server's Network Security* **)**

Ahmad Zafrullah Mardiansyah*, Yayank Muhammad Abdussyakur, Andy Hidayat Jatmika
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Mataram
Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA
Email: zaf@unram.ac.id , yayankmuhammadabdussyakur@gmail.com, andy@unram.ac.id

**Penulis Korespondensi*

Abstract

Network security is the most important aspect of a system in maintaining data validation and integrity, as well as ensuring the availability of services for its users. The development of network security system demands a better security, especially in servers. However, there are still weaknesses in the server that controls incoming packet which results in more vulnerable it is to be stormed by Denial of Service and Brute Force attacks on the system. This study aims to improve network security performance using the Port Knocking and Honeypot methods which will be combined with the Iptables method. From the tests carried out with the addition of the IPTables method, an increase in performance of CPU usage (38.4%) and memory (44.2%) on servers and network security were obtained compared to those that only use the Port Knocking and Honeypot methods.

Keywords: *Brute force, Denial of Service, Honeypot, Iptables, Port Knocking.*

1. PENDAHULUAN

Keamanan jaringan merupakan aspek terpenting sebuah sistem dalam menjaga validitas dan integritas data, serta menjamin ketersediaan layanan bagi penggunaanya. Sistem keamanan jaringan komputer harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak[1]

Kebutuhan akan jaringan komputer semakin bertambah penting, baik dalam pendidikan, pekerjaan maupun dalam sebuah permainan, dengan banyaknya akses ke jaringan tersebut maka akan banyak pula peluang kejahatan yang terjadi didalam jaringan ataupun adanya peretas yang dapat mematikan sumber daya pada server.

Sepanjang tahun 2019, sistem monitoring mata garuda mendeteksi adanya sekitar 290,3 juta serangan siber ke jaringan internet Indonesia di antaranya 137,4 juta percobaan pembocoran data, 117,9 juta total serangan menggunakan trojan, 6,4 juta total serangan *name* server, 12,5 juta serangan terhadap *port* 80, 4224 total laporan aduan publik dan 2831 aduan publik terkait kerentanan sistem. Data tersebut merupakan data yang di dapatkan oleh Badan Siber dan Sandi Negara (BSSN) pada tahun 2019[2].

Ancaman siber akan terus ada dan semakin canggih. Bukan hanya di Indonesia, Amerika pun mengalami kesulitan menghadapi ancaman ini. Menurut FBI. Tindakan kriminal siber di Amerika sepanjang tahun 2019 telah mengakibatkan kerugian 3,5 Milyar dolar atau sekitar 47,9 trilyun rupiah. Terdapat juga beberapa kejadian penting keamanan siber di Indonesia pada tahun 2019 seperti kebocoran data 13 juta pengguna Bukalapak, Pembatasan media *social whatshapp* oleh kemkominfo, *website* Kemendagri di retas, *website* DPR tidak bisa di akses, kebocoran 7,8 juta data pribadi penumpang malindo air dan terdapat beberapa penyerangan lainnya yang terjadi di Indonesia pada tahun 2019[2].

Sehubungan dengan hal tersebut maka pada penelitian ini digunakan metode *Port knocking* dan *Honeypot* untuk keamanan server. penelitian [3] melakukan sistem autentifikasi menggunakan kombinasi lapisan-lapisan kunci untuk dapat menggunaka *port* komunikasi yang dilindungi *Port knocking*. Teknik ini mempertahankan satu atau lebih *port* yang telah dikonfigurasi sebelumnya dan hanya dapat terbuka jika menggunakan *sequence of request* [1].

Terdapat juga *Honeypot* yang bertujuan untuk menganalisis penyerangan yang dilakukan, *Honeypot*

merupakan sebuah sistem umpan atau aplikasi yang digunakan untuk melakukan simulasi seluruh jaringan untuk memikat penyerang dengan menyamarkan diri sebagai sistem yang rentan. Namun, dari hasil penelitian yang dilakukan masih terdapat kelemahan dalam mengontrol paket yang masuk, sehingga akan lebih mudah dalam melakukan penyerangan *Denial of Service* dan *Brute force* ke sistem. Hal tersebut dikarenakan tidak adanya proses yang mengatur paket yang masuk kedalam sistem keamanan. Maka dari itu, perlu adanya penangan terhadap ancaman yang dapat merusak server dan melakukan tindakan keamanan, pemantauan dan menganalisis serangan yang dilakukan oleh peretas.

Untuk mengatasi kelemahan dari penelitian [3], diusulkan menggunakan *Iptables*. *Iptables* merupakan salah satu solusi yang dapat diberikan untuk menangani serangan. *Iptables* dapat memperkuat keamanan dalam jaringan dengan cara melakukan *filtering* (penyaringan) terhadap lalu lintas (*traffic*) data. Dengan adanya *Iptables* ini bisa membuat aturan (*rule*) untuk arus lalu lintas data, aturan-aturan itu dapat mencangkup banyak hal, seperti besar data yang boleh lewat, jenis paket/datagram yang dapat diterima, mengatur *traffic* berdasarkan asal dan tujuan data, *forwarding*, NAT, *redirecting*, pengolahan *port*, dan *firewall*.

Pada penelitian ini akan dilakukan implementasi, analisis dan juga perbandingan kinerja dari keamanan jaringan menggunakan metode *Port knocking* dan *Honeypot* dengan keamanan jaringan yang menggunakan metode *Port knocking*, *Honeypot* dan *Iptables.* Simulasi akan dilakukan dengan menggunakan simulator GNS3. Hasil dari penelitian ini bertujuan untuk meningkatkan keamanan jaringan dan diharapkan dapat dijadikan sebagai perbandingan dengan penelitian sebelumnya sekaligus acuan untuk penelitian selanjutnya yang berkaitan dengan keamanan jaringan.

2. TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Terdapat beberapa penelitian yang digunakan sebagai sumber rujukkan dalam penelitian ini. Penelitian Port knocking dan Honeypot sebagai keamanan jaringan pada server dengan pengujian mengguanakan aplikasi Putty dan Moba*Xtreem*, dari hasil uji coba penyerangan dapat ditarik kesimpulan penyusup berhasil dialihkan ke *server* bayangan dengan metode *Honeypot* [3].

Peneliti [4] melakukan uji coba penerapan *IPTables Firewall* pada linux. Berdasarkan uji coba yang telah dilakukan, maka akan dilakukan penelitian mengenai pembuatan *filtering* paket TCP dan UDP terhadap penerapan *IPTables* dengan melakukan *block* akses web dan *game online steam* dota 2. Dari uji coba yang telah dilakukan diperoleh hasil *packet loss* dalam penerapan *IPTables.*

Penelitian [5] melakukan uji coba keamanan jaringan menggunakan *Honeypot* dan IDS pada jaringan *Nirkabel* (*Hotspot*). Berdasarkan penelitian yang dilakukan, hasil yang diperoleh adalah berupa *file log* dari aktifitas penyerang yang telah disimpan oleh *Honeyd* pada direktori /var/log/honeyd/. Setiap ada akses menuju virtual mesin (server palsu) pada alamat ip 192.168.1.100 – 192.168.1.105 akan langsung tercatat di *file log*. Kemudian setiap aktifitas yang melakukan penyerangan terhadap *hotspot* terutama pada server palsu akan terekam oleh *Honeyd* sesuai dengan jenis server palsu tersebut.

Peneliti [6] melakukan analisis perbandingan *system* keamanan jaringan menggunakan *Snort* dan *Netfilter*. Berdasarkan penelitian yang dilakukan didapatkan beberapa hasil, yaitu perangkat keras lebih banyak digunakan oleh sistem keamanan jaringan *snort.* Selain menggunakan server, sistem juga membutuhkan PC snort, sedangkan pada jaringan *netfilter* hanya menggunakan server. Berdasarkan pengamatan dalam penggunaan memori *netfilter* lebih banyak penggunaanya dibandingkan dengan *snort* yang dimanan penggunaan memori *netfilter* adalah 457968 KiB sedangkan untuk *snort* sebesar 330668 KiB dan uji coba yang terakhir yaitu pencegahan serangan yang dimana penyerangan yang di gunakan ada 3 penyerangan yaitu *ping attack*, DoS, dan *port scanning netfilter* memiliki hasil keamanan yang lebih baik Dengan persentase rata-rata 63,92% dari pada *snort* dengan persentase rata-rata 49,83%.

Peneliti [7] melakukan uji coba keamanan jaringan menggunakan *simple Port Knocking* pada *dynamic routing* (OSPF) menggunakan simulasi GNS3, Berdasarkan percobaan yang telah dilakukan diperoleh hasil yaitu bagaimana cara mengimplementasi akses *open* dan *block Disable port* dengan menggunakan metode *simple Port knocking* yang bertujuan agar menutup celah pada sisi server dengan membuat *port* pada *router* tidak terlihat oleh pihak lain yang tidak dipercaya meskipun sudah di *scanning port* , namun tetap akan terlihat terbuka dan dapat diakses oleh pihak yang sudah terautentifikasi sehingga untuk mencegah adanya serangan akses dari *attack*.

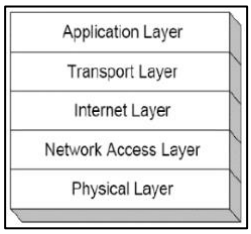
2.2. Dasar Teori

a. Jaringan Komputer

Jaringan komputer adalah hubungan dari sejumlah perangkat yang dapat saling berkomunikasi satu sama lain. Perangkat yang dimaksud pada definisi ini mencakup semua jenis perangkat komputer (Komputer desktop, laptop, *smartphone*, PC, *tablet*) dan perangkat penghubung [8].

b. TCP/IP

TCP/IP (Singkatan dari *Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain didalam jaringan internet. *Protocol* ini tidaklah dapat berdiri sendiri, karena memang *protocol* ini berupa kumpulan *protocol* (*protocol suite*). *Protocol* ini juga merupakan *protocol* yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. *Protocol* ini juga berupa *routable* yang berarti *protocol* ini cocok untuk menghubungkan sistem-sistem berbeda (seperti Microsoft Windows dan keluarga UNIX) untuk membentuk jaringan yang heterogeny [9]. Tampilan *Layer* TCP/IP dapat dilihat pada Gambar 1



Gambar 1. Layer TCP/IP

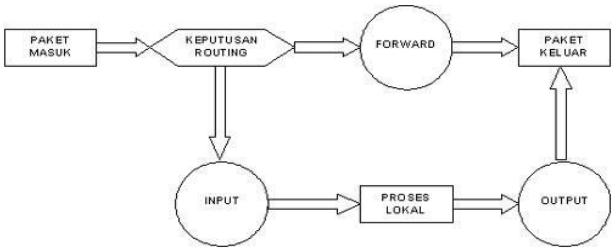
c. *Firewall*

Firewall adalah perangkat atau sebuah program yang berfungsi untuk mengontrol aliran lalu lintas jaringan antar jaringan atau *host* yang menerapkan keamanan yang berbeda. *Firewall* sering dibahas dalam sebuah konteks konektivitas internet, tetapi *firewall* juga memiliki penerapan pada konektivitas jaringan lainya [10].

d. IPTABLES

Iptables adalah suatu *tools* dalam sistem operasi *linux* yang berfungsi sebagai alat untuk melakukan *filtering* (penyaringan) terhadap (*traffic*) lalu lintas data. Secara sederhana digambarkan sebagai pengatur lalu lintas data. Dengan *iptables* inilah akan bisa mengatur semua lalu lintas dalam komputer, baik yang masuk ke komputer, keluar dari komputer, ataupun *traffic* yang sekedar melewati komputer. *Iptables* merupakan sistem *firewall* di sistem *open source* yang mendukung *layer* 3 (*Network layer*), *layer* 4 (*Transport layer*) dan *layer* 7 OSI *layer*

Dengan kemampuan *tools iptables* ini, dapat dilakukan banyak hal. Salah satunya yaitu dapat membuat aturan (*rule*) untuk arus lalu lintas data. Aturan-aturan itu dapat mencakup banyak hal, seperti besar data yang boleh lewat, jenis paket/datagram yang dapat diterima, mengatur *traffic* berdasarkan asal dan tujuan data*, forwarding*, NAT, *redirecting*, pengelolaan *port* dan *firewall.*



Gambar 2. *Chain IPTables*

Pada Gambar 2 *chain* tersebut digambarkan pada lingkaran, jadi saat sebuah paket sampai pada sebuah lingkaran, maka disitulah terjadi proses penyaringan. *Chain* akan memutuskan nasib paket tersebut apabila keputusan adalah DROP, maka paket tersebut akan di-DROP, tetapi jika *chain* memutuskan untuk ACCEPT, maka paket akan dilewati melalui diagram tersebut [4].

e. *Port Knocking*

Port knocking adalah konsep menyembunyikan layanan jarak jauh di dalam sebuah *firewall* yang memungkinkan akses ke *port* tersebut hanya untuk mengetahui *service* setelah klien berhasil diautentikasi ke *firewall*. Hal ini dapat membuat untuk mencegah pemindai untuk mengetahui *service* apa saja yang saat ini tersedia di *host* dan juga berfungsi sebagai pertahanan terhadap serangan *zero-day* [11].

Port knocking bekerja seperti halnya brankas dengan kunci kombinasi angka putar. Pada brankas tersebut, diharuskan memutar lapisan-lapisan kunci kombinasi beberapa kali hingga tepat seperti yang ditentukan. Sebenarnya memutar lapisan-lapisan di dalam berankas. Dalam lapisan-lapisan kunci tersebut terdapat sebuah lubang kunci. Jika sebuah putaran tepat, maka sebuah lubang terbuka. Jika seluruh putaran dilakukan dengan kombinasi yang benar, maka seluruh lubang terbuka dan menciptakan sebuah jalur khusus yang bebas tidak ada hambatan sama sekali. Jalur lubang kunci tadi tidak lagi menjadi penghalang pintu brankas untuk dibuka, sehingga pintu dapat terbuka dengan mudah [12].

f. *Honeypot*

Honeypot merupakan sebuah sistem yang di bangun menyerupai atau persis dengan sistem yang

sesungguhnya, dengan tujuan agar para *attacker* teralih perhatiannya dari sistem utama yang akan di serang, dan beralih menyerang ke sistem palsu tersebut. Saat ini *Honeypot* tidak hanya berfungsi atau bertujuan untuk menjebak *attacker* untuk melakukan serangan ke server asli, namun *Honeypot* juga bermanfaat untuk para *system administrator* atau *security analyst*, untuk menganalisa aktifitas apa saja yang dilakukan oleh *attacker/malware* yang terdapat di dalam sistem *Honeypot* tersebut [13].

g. Jenis-jenis Serangan pada keamanan jaringan

1. *Denial of Serevice*

Merupakan singkatan dari *Denial of Services*, sebuah penyerangan terhadap sebuah sistem dengan jalan menghabiskan sumber data sistem tersebut sehingga tidak dapat di akses lagi. Sumber daya dapat berupa CPU, RAM, *Swap*, *cache*, maupun *bandwidth*.

2. SSH *Brute Force*

SSH (*Secure Shell*) adalah *protocol* jaringan yang memungkinkan pertukaran data melalui saluran yang aman antara kedua perangkat jaringan. *Brute force* adalah metode trial dan error yang digunakan oleh program aplikasi untuk memecahkan data yang telah dienkripsi seperti password atau standar data enkripsi (DES). Metode ini akan mencoba semua kemungkinan yang ada dari pada menggunakan strategi yang lebih baik, berdasarkan pengertian SSH dan *Brute Force* yang telah dijelaskan sebelumnya dapat diambil kesimpulan bahwa SSH *Brute Force attack* adalah sebuah jenis serangan pada SSH dengan mencoba semua kombinasi yang memungkinkan untuk mendapatkan akses pada sebuah SSH.

h. Perangkat Lunak (*Software)* Pendukung

1. GNS3

GNS (*Graphical Network Simulator* 3) adalah satu satunya aplikasi *open source* simulasi jaringan yang dapat bekerja secara sinergi mensimulasikan hampir semua sistem operasi. GNS3 sebuah aplikasi yang masih belum bisa dikatakan sempurna, masih terdapat beberapa segmen yang terus dikembangkan namun GNS 3 adalah satu-satunya aplikasi simulasi jaringan yang mampu melakukan simulasi jaringan secara nyata. GNS3 memiliki antarmuka grafis yang mampu dipahami sehingga mempermudah untuk merancan dan mengkonfigurasi jaringan virtual [14].

2. Linux Ubuntu

Ubuntu adalah salah satu distribusi linux yang berbasis Debian dan didistribusikan menjadi perangkat lunak sistem operasi yang bebas. Secara singkat dan jelasnya yaitu ubuntu adalah sejenis sistem operasi berbasiskan linux Debian

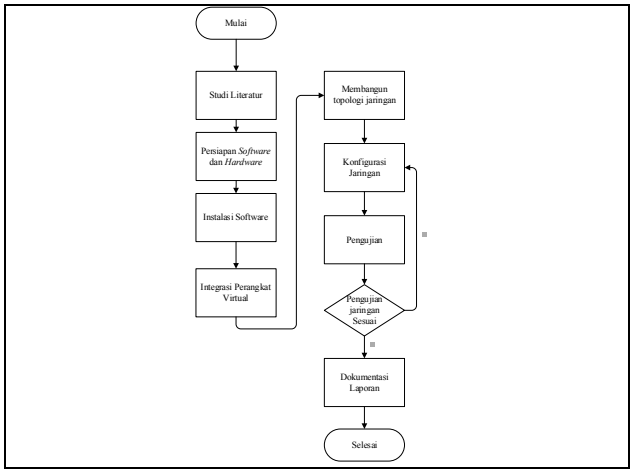
3. VirtualBox

VirtualBox adalah program untuk virtualisasi komputer yang ditujukan untuk komputer desktop, server, maupun laptop. Dengan menggunakan VirtualBox maka dapat memvirtualisasikan OS 32bit dan 64bit pada sebuah komputer yang menggunakan prosesor Inter dan AMD, baik virtualisasi perangkat lunak maupun perangkat keras. Alasan utama memilih VirtualBox merupakan perangkat lunak virtualisasi gratis dan *open source* yang menawarkan banyak kemudahan dalam melakukan virtualisasi, serta kemampuanya dalam membuat virtual *appliance* secara *native*

3. METODE PENELITIAN

3.1. Diagram Alir

Dari penelitian yang akan dilakukan, terdapat beberapa Langkah-langkah atau proses yang akan dilakukan. Langkah-langkah tersebut dilihat pada Gambar 3 berikut:



Gambar 3. Diagram Alir Penelitian.

3.2. Studi Literatur

Studi literatur merupakan tahapan pertama melakukan penelitian. Studi literatur dilakukan untuk mendapatkan gambaran umum maupun khusus mengenai objek maupun teori pendukung dalam penelitian ini. Studi literatur yang dilakukan pada penelitian ini berkaitan dengan penerapan metode *Iptables*, *Port knocking* dan *Honeypot* pada keamanan jaringan. Sumber-sumber literatur berupa jurnal ilmiah, skripsi, *paper* maupun sumber lainya yang berkaitan dengan penelitian ini.

3.3. Persiapan *Software* dan *Hardware*

Alat yang digunakan dalam perancangan sistem keamanan jaringan ini adalah sebagai berikut:

a. *Hardware*

Laptop PC dengan spesifikasi *Processor Intel Core* I5 5300U @2.3GHZ, 8 GB RAM, 256 SSD.

b. *Software*

1. *Microsoft Office Visio, Software* yang digunakan untuk mendesai alur pengujian maupun alur penelitian yang akan dilakukan
2. GNS3 sebagai *Network Simulator*
3. *Virtualbox* sebagai alat perangkat lunak Virtualisasi yang dapat mengoprasikan / menginstal beberapa *Operating System* pada sistem utama.
4. Linux Ubuntu LTS 14.0 sebagai sistem operasi yang digunakan untuk instalasi, konfigurasi dan pengujian sisem.
5. LOIC (*Low Orbit Ion Cannon*) sebagai *tools* yang digunakan untuk melakukan penyerangan *Denial of Service.*
6. *Hydra* sebagai *tools* yang digunakan untuk melakukan penyerangan *brute force*

3.4. Instalasi *Software*

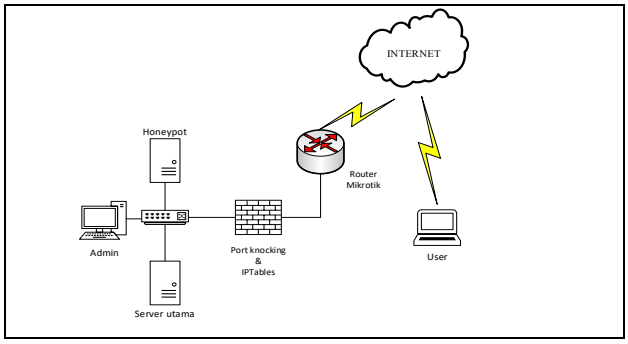
Pada tahap ini dilakukan instalasi *software* dengan menggunakan *software* GNS3, selanjutnya melakukan instalsi *Virtualbox* sebagai perangkat lunak untuk mengoperasikan/menginstal *Operating System virtual* dimana nantinya perangkat lunak ini akan diintegrasikan dengan simulator GNS3. Pada Simulator GNS3 digunakan *Operating System* Linux Ubuntu LTS 14.0 karena sistem operasi ini tidak begitu memakan banyak *resource* komputer.

3.5. Integrasi Perangkat Virtual

Pada tahapan ini dilakukanya integrasi perangkat *virtual*. Perangkat lunak GNS3 yang sudah di-*install*. Selanjutnya akan diintegrasikan dengan *VirtualBox* agar dapat menjalankan *Operating System* yang digunakan dan dapat terhubung dengan GNS3. Apabila tahapan ini telah selesai, selanjutnya dilakukan tahapan konfigurasi jaringan

3.6. Membangun Topologi Jaringan

Melakukan perancangan dan memberikan gambaran mengenai sistem keamanan yang akan dibangun.



Gambar 4. Topologi Keamanan Jaringan.

3.7. Konfigurasi Jaringan

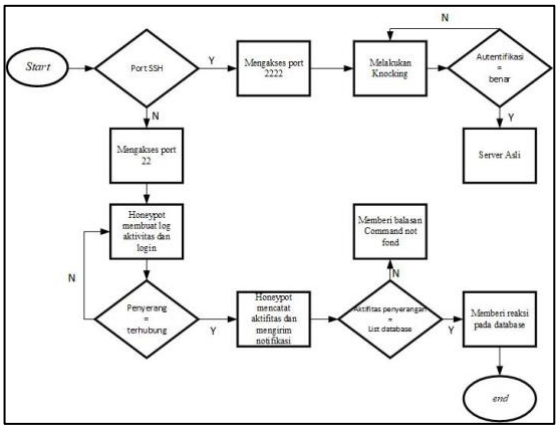
Pada tahap ini dilakukan konfigurasi keamanan jaringan *iptables*, *Port knocking* dan *Honeypot* pada server utama. Pada skenario pertama dilakukan instalasi *iptables* dan di lakukan konfigurasi untuk memfilter *port* 22 yang merupakan *port* SSH. pada skenario kedua dilakukan instalasi *Port knocking* dan dilakukan konfigurasi untuk membuka dan menutup akses menuju *port* yang telah di *block* oleh *firewall*. Pada skenario ketiga dilakukan instalasi *Honeypot* dan dilakukan konfigurasi untuk membuat server bayangan.

3.8. Pengujian

3.8.1. Skenario Pengujian Sistem

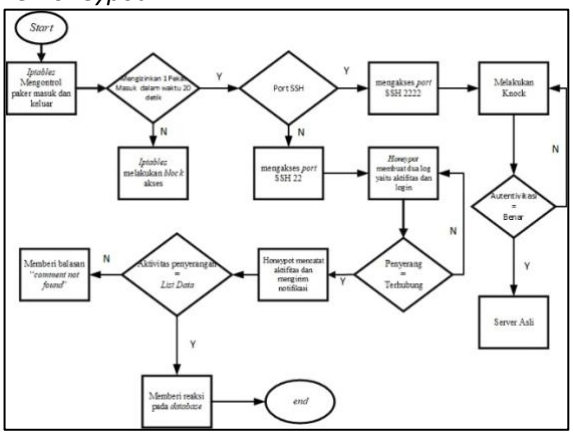
Skenario pengujian yang dilakukan pada penelitian ini menggunakan dua kondisi. Kedua kondisi tersebut berfungsi untuk menguji tingkan keamanan *port* server terhadap serangan yang datang ke server. Berikut adalah dua kondisi yang digunakan diantaranya yaitu:

a. Kondisi yang pertama adalah melakukan uji coba terhadap penyerangan server dengan menggunakan keamanan *Port knocking* dan *Honeypot*. *Honeypot* berperan sebagai pencatat hasil *log* aktifitas yang dilakukan oleh penyerang. Dari catatan *log* aktifitas tersebut menjadi masukan dalam mengamankan sistem sehingga *administrator* server dapat mempelajari penyerangan yang dilakukan. Hasil dari pengujian tersebut dilihat dari kinerja sistem *Port knocking* apakah sistem tersebut bisa di tembus oleh peretas atau tidak. Pengujian pertama dilakukan tanpa *Iptable*.



Gambar 5. Diagram Alir Skenario pengujian sistem menggunakan *Port Knocking* dan *Honeypot*

b. Kondisi pengujian yang kedua dilakukan dengan menggunakan *iptables*, *Port knocking* dan *Honeypot*. Hasil dari uji coba kedua yaitu memperlihatkan keamanan dari *Port knocking* yang telah di kombinasikan oleh *Iptables*. Penggunaan *Iptables* yaitu untuk mengatur lalu lintas paket yang masuk ke server. Dari kedua jenis pengujian tersebut dapat diketahui bahwa keamanan *firewall* yang telah dibuat dapat bekerja dengan baik dan peretas dapat di alihkan ke *Honeypot*.



Gambar 6. Diagram Alir Skenario pengujian sistem menggunakan *Iptables, Port Knocking* dan *Honeypot*

3.8.2. Skenario Pengujian Serangan

a. Pengujian DoS *Attack*

Proses pengujian *Denial of Service* (DoS) *attack* yang digunakan mengadopsi konsep penelitian sebelumnya [1]. Teknik serangan DoS ini dilakukan dengan cara menghabiskan sumber daya (*resource*) yang dimiliki oleh komputer target sampai komputer tersebut tidak dapat lagi menjalakan layanan web dengan baik. *Client* melakukan serangan DoS menggunakan *tools* LOIC. Server menjalankan *iptables* yang menangkap adanya serangan DoS melalui *protocol* TCP dengan kondisi waktu yang telah ditentukan dan menerima jumlah paket yang melebihi dari jumlah yang telah di tentukan akan langsung di blokir. Tujuan dari penyerangan DoS *attack* ini adalah untuk mengetahui ketahanan / performa sistem dalam melayani *service request* yang sah walaupun sedang mengalami serangan DoS

b. Pengujian *Brute force*

Proses pengujian *brute force attack* yang digunakan mengadopsi konsep penelitian sebelumnya [1], [3]. Pengujian serangan *brute force* dilakukan untuk menganalisa *password* dengan cara mencoba setiap kemungkinan *password*. Dalam penerapan *iptables* dan *Port knocking* pada *service* SSH yang berjalan di *port* 22 menggunakan Hydra. *Tools* ini nantinya akan mengirim banyak username dan kata sandi yang akan di coba dengan harapan dapat menebak dengan benar username dan password pada SSH, waktu yang di perlukan untuk menemukan username dan password ditentukan dengan serumit apa *username* dan *password* yang dibuat oleh admin. Akan tetapi *tools* ini nantinya tidak dapat melakukan serangan terhadap *service* SSH karena *port* SSH tersebut dilindungi *Port knocking*. Sehingga *client* tidak bisa mendapatkan informasi nama host dan *password* server.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi *Virtual machine*

Virtual *Machine* adalah *Software* virtualisasi yang dapat digunakan untuk menjalankan sistem operasi tambahan di dalam sistem operasi utama. Pada komputer penelitian, sisi *client* dan server akan dibuat dengan menggunaka media bantuan *virtual machine*

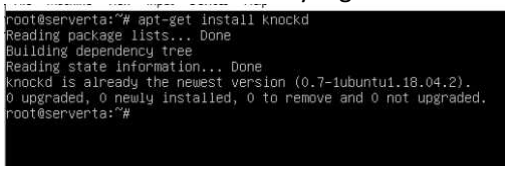
4.2. Implementasi Server

Server merupakan suatu sistem komputer yang memiliki layanan khusus berupa penyimpanan data. Data yang disimpan melalui server berupa informasi dan beragam jenis data yang kompleks. Layanan-layanan yang akan dibuat pada server diantaranya seperti DHCP Server, DNS Server, SSH, Mysql Server, FTP.

4.3. Pengujian Sistem

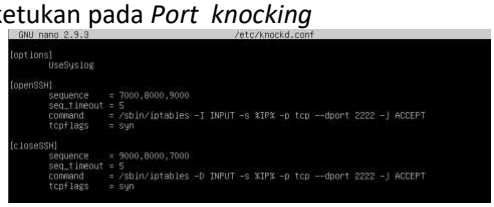
a. *Port Knocking*

Pada sisi server telah terpasang program daemon *tool* yang bertindak sebagai autentifikator yaitu knockd sebagai autentifikasi *Port knocking*. *Port knocking* diimplementasikan dengan mengkonfigurasi daemon guna memonitoring *log firewall* untuk permintaan koneksi dan menentukan apakah *client* telah melakukan urutan ketukan yang benar



Gambar 7. Proses penginstalan *Port Knocking*

Pada Gambar 7 dapat dilihat proses penginstalan knockd dan selanjutnya melakukan konfigurasi dalam menentukan ketukan yang akan di gunakan sebagai autentifikasi *client* yang akan di gunakan untuk melakuakan *remote* terhadap server. berikut merupakan konfigurasi ketukan pada *Port knocking*



Gambar 8. Proses konfigurasi Knock

b. *Honeypot*

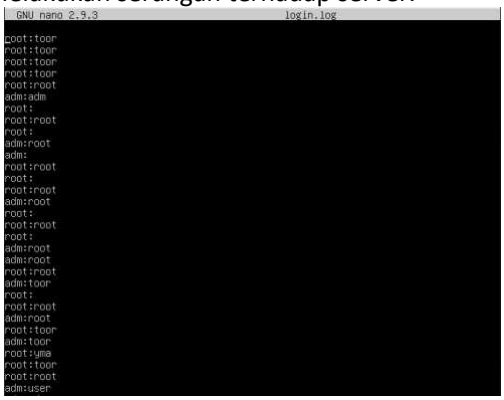
Merupakan keamanan yang dibuat untuk diserang oleh penyerang dimana pada penelitian ini *Honeypot* memiliki akses terhadap *port* 22 yang dimana *port* 22 tersebut pada awalnya merupakan *port* standar yang digunakan untuk mengakses SSH, tetapi pada penelitian ini *port* SSH yang asli telah di simpan pada *port* 2222 dan dilindungi oleh *Port knocking*

Gambar 9. Melakukan *scanning port* menggunakan Nmap

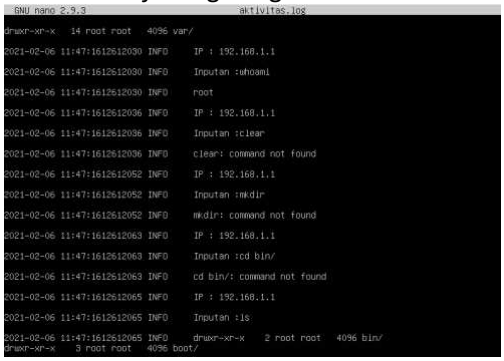
Pada Gambar 9 Menjelaskan proses untuk melakukan *scanning port* yang terbuka pada server menggunakan *tools* nmap. Dapat dilihat bahwa *status port* untuk layanan SSH dalam kondisi *open* atau terbuka dimana *port* SSH tersebut merupakan *port* dari *Honeypot* untuk mengelabui penyerang

Gambar 10. Tampilan *Honeypot* pada server

Pada Gambar 10 Menjelaskan proses *Honeypot* yang berjalan pada server yang dimana pada tampilan tersebut menunjukkan seluruh aksi penyerangan yang dilakukan akan dicatat pada *file* "aktivitas.log" yang merupakan seluruh aktivitas yang dilakukan penyerang dalam server akan di catat dan "login.log" merupakan *file* yang dibuat untuk menyimpan hasil uji coba penggunaan *user* dan *password* dalam membobol server. *Honeypot* juga mencatat IP *address* dan *port* yang dilalui oleh penyerang dalam melakukan serangan terhadap server.



Gambar 11. *file* login.log



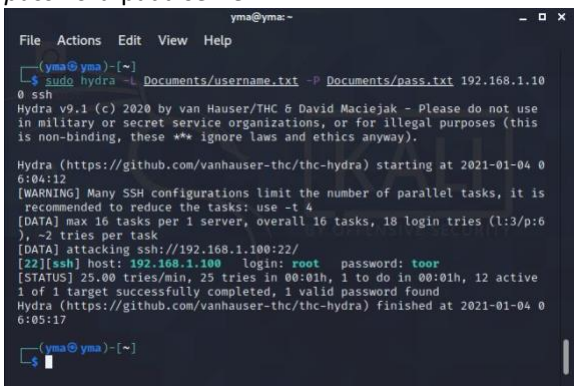
Gambar 12. *file* aktivitas.log

c. Hasil Pengujian

1. *Brute force*

Pada penelitian ini dilakukan penyerangan *brute force* menggunakan hydra terhadap *port* 22 yang merupakan *port Honeypot* dan *port* 2222 yang merupakan *port* SSH, dimana penyerangan

ini bertujuan untuk memperoleh *username* dan *password* pada server.



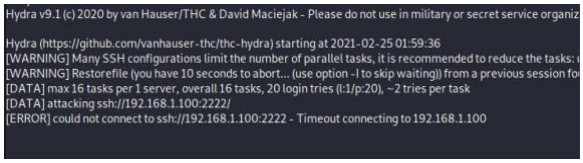
Gambar 13.Proses penyerangan menggunakan *Brute force*

Pada Gambar 13 menjelaskan proses untuk melakukan *brute force* yang dilakukan oleh *client,* Dimana *client* melakukan *brute force* pada layanan ssh dengan *port* 22 dengan IP tujuan adalah 192.168.1.100. sehingga didapatkan hasil *brute force* sukses dilakukan oleh *client* sehingga menampilkan "host: 192.168.1.100" dengan *username* "login: root" dan *password* "password: toor" yang valid

Selanjutnya dilakukan uji coba untuk mengakses server menggunakan perintah "ssh username@host -p 22" pada uji coba penyeranga sebelumnya yang dilakukan menggunakan *bruteforce*, penyerang mendapatkan hasil yaitu *username*, *host* dan *password* dalam mengakses server dan dapat digunakan untuk mengakses server seperti pada Gambar 14.

Gambar14.Melakukan remote server *Honeypot*

Pada penyerangan yang telah dilakukan oleh penyerang tidak menutup kemungkinan penyerang mengetahui *port* 22 yang pada umumnya merupakan *port* SSH bukan *port* SSH yang asli. Sehingga penyerang akan mencoba untuk menyerang *port-port* yang lain. Apabila penyerang telah mengetahui *port* 2222 merupakan *port* SSH dari server maka penyerang akan mencoba untuk melakukan penyerangan *brute force* terhadap *port* 2222 seperti pada Gambar 15.

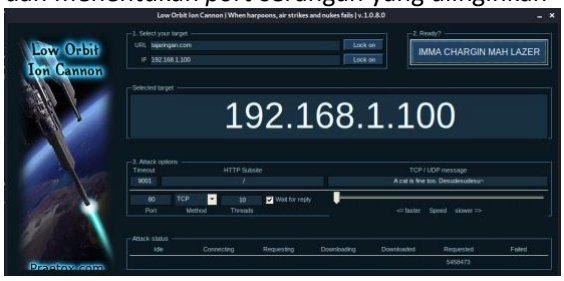


Gambar 15. Penyerangan *Brute force* menuju *port* 2222

Pada Gambar 15 dilakukan penyerangan menuju *port* 2222 yang merupakan *port* SSH dari server dimana penyerangan tersebut mengalami "*Time out connection to 192.168.1.100"* atau penyerangan gagal terhubung dikarenakan pada *port* 2222 telah dilindungi oleh *Port knocking* yang dimana untuk mengakses *port* 2222 dibutuhkan autentifikasi untuk membuka *port* tersebut

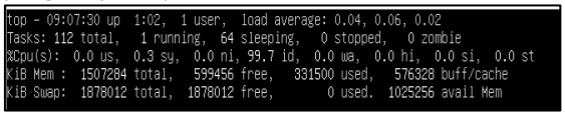
2. *Denial of Service*

Pada penelitian ini dilakukan pengujian penyerangan *Denial of Service* terhadap server menggunakan *tool* LOIC. *Software* ini merupakan *tools* yang banyak digunakan pada saat ini untuk melakukan penyerangan DoS. Dengan *tools* ini dapat melakukan penyerangan berdasarkan IP dan menentukan *port* serangan yang diinginkan

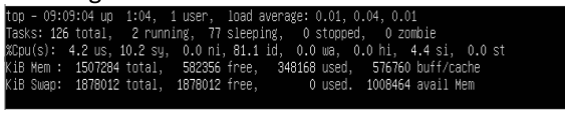


Gambar 16. Serangan DoS menggunakan LOIC

Pada Gambar 16 Menunjukkan penyerangan yang menggunaakn *tools* LOIC dimana *url* yang akan diserang adalah "tajaringan.com" dengan IP *address* "192.168.1.100" dengan tujuan *port* penyerangan yaitu *port* 80 dengan *menthod* TCP dan *thread* yang digunakan sebanyak 10. Penyerangan akan dilakuakan dalam waktu 1 menit dan akan dilihat penggunaan CPU dan Memori pada server menggunakan perintah TOP yang berfungsi sebagai memanajemen proses yang berjalan pada server



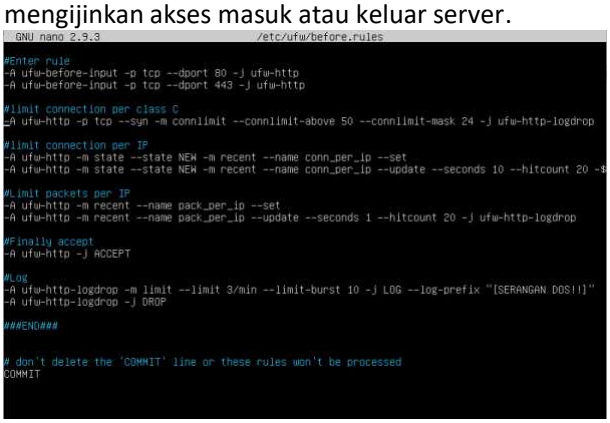
Gambar 17. Proses TOP pada saat belum diserang



Gambar 18. Proses TOP pada saat diserang

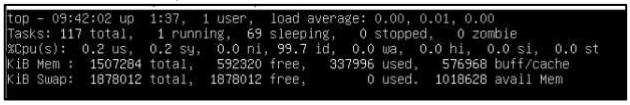
d. IPTables

Pada pengujian sistem yang telah dilakukan menggunakan *software* Hydra dan LOIC didapatkan hasil penyerangan menggunakan Hydra dapat di antisipasi oleh *Honeypot*, sedangkan penyerangan DoS menggunakan LOIC tidak dapat di antisipasi oleh server, sehingga untuk mengatasi serangan DoS dibuat *rules* IPTables yang berfungsi untuk memblokir dan mengijinkan akses masuk atau keluar server.

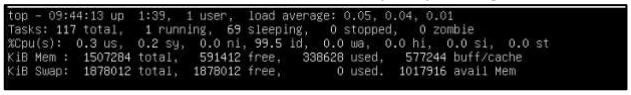


Gambar 19 Rules IPTables

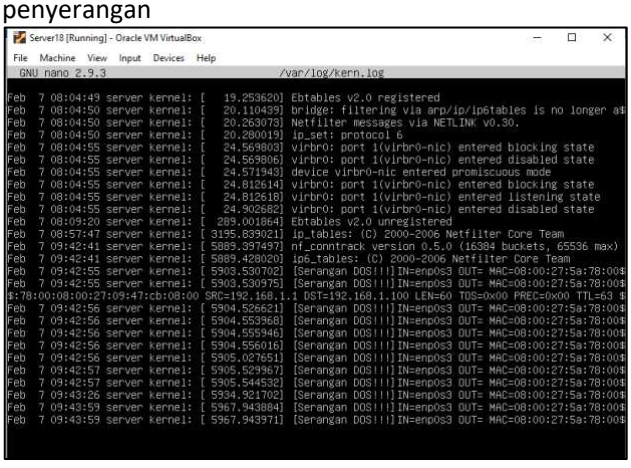
Setelah melakukan konfigurasi IPTables selanjutnya dilakukan uji coba penyerangan *Denial of Service* (DoS) terhadap server dengan menggunakan *tools* LOIC, sehingga didapatkan hasil penggunaan CPU dan Memori seperti berikut:



Gambar 20. Proses TOP sebelum penyerangan



Gambar 21. Proses TOP setelah dilakukan penyerangan



Gambar 22. Hasil Log kernel pada server

TABEL I. Hasil Pengujian *Port Knocking* dan *Honeypot*

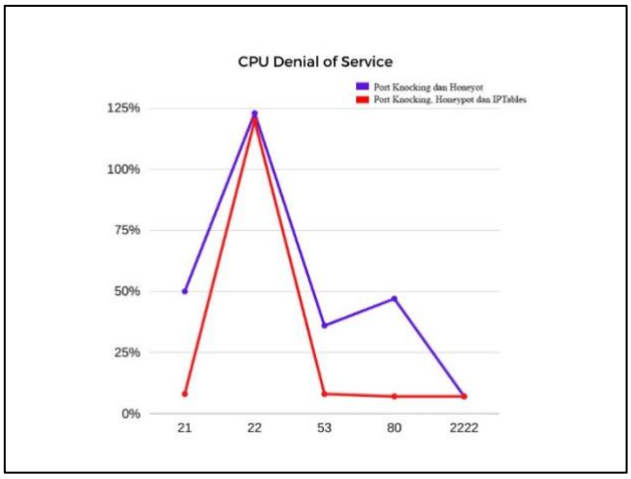
Serangan	*Port*	CPU	Memory	Keterangan	
				Berhasil	Gagal
Denial of Service	21	16,0sy	672,8	✓	
	22	0,9sy	900,8	✓	
	53	1,0sy	341,2	✓	
	80	0,8sy	684,6	✓	
	2222	0.7sy	0		✓
Bruteforce	21	5,0sy	10983,4	✓	
	22	12,3sy	18763,3	✓	
	53	3,6sy	274,6		✓
	80	4,7sy	3755,4		✓
	2222	0.7sy	0		✓

Berdasarkan hasil pengujian menggunakan metode *Port knocking* dan *Honeypot* pada Tabel 1, didapatkan hasil yaitu pada penyerangan menggunakan *Denial of Service* menuju *port* 21,22,53,80 dan 2222 menyebabkan peningkatan performa CPU dan Memori pada *server* serta penyerangan terhadap *port* 21,22,53 dan 80 berhasil dilakukan oleh penyerang. Sedangkan pada penyerangan menggunakan *Bruteforce* menuju *port* 21 dan 22 berhasil dilakukan oleh penyerang

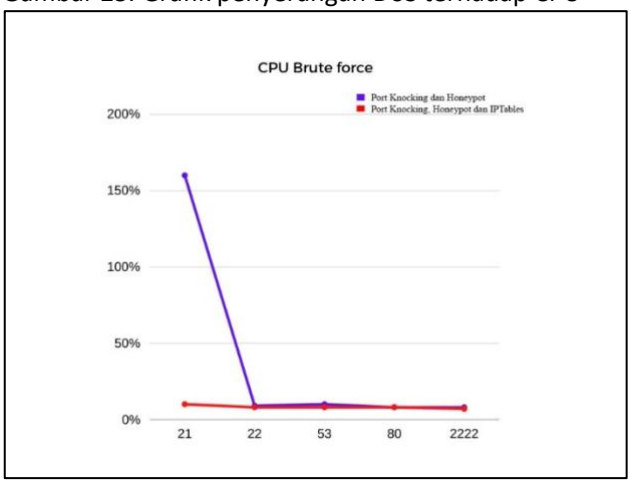
TABEL II. Hasil Pengujian *Port Knocking, Honeypot* dan IPTables

Serangan	*Port*	CPU	Peningkatan%	Memory	Peningkatan%	Keterangan	
						Berhasil	Gagal
Denial of Service	21	0.9sy	94%	493,6	26%		✓
	22	0,7sy	22%	476,4	47%	✓	
	53	0,8sy	20%	243,6	28%		✓
	80	0,8sy	0%	101,8	85%		✓
	2222	0,7sy	0%	0	0%		✓
Bruteforce	21	0.8sy	84%	101,6	99%		✓
	22	12,0sy	2%	17155,1	8%	✓	
	53	0,8sy	77%	132,6	51%		✓
	80	0.7sy	85%	49,6	98%		✓
	2222	0.7sy	0%	0	0%		✓
Jumlah rata-rata peningkatan persentase		384/10 = 38,4%		442/10 =44,2%			

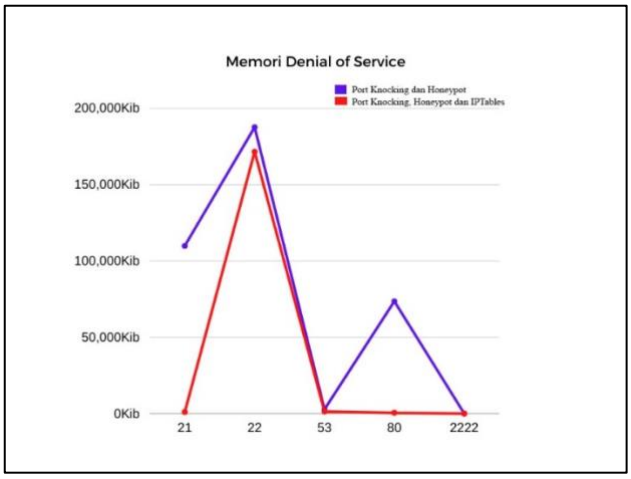
Berdasarkan hasil pengujian menggunakan metode *Port knocking*, *Honeypot*, dan IPTables pada Tabel 2, didapatkan hasil yaitu penggunaan CPU dan memory pada saat server diserang tidak mengalami peningkatan performa dan proses penyerangan menuju *port* 21,53,80 dan 2222 juga gagal dilakukan oleh penyerang



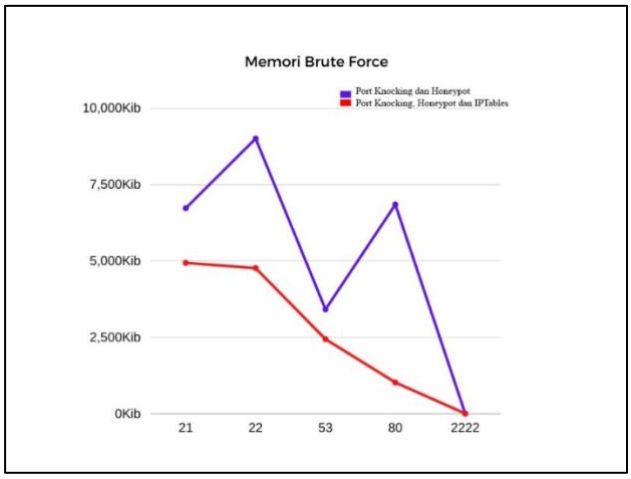
Gambar 23. Grafik penyerangan DoS terhadap CPU



Gambar 24. Grafik penyerangan *Brute force* terhadap CPU



Gambar 25. Grafik penyerangan DoS terhadap Memori



Gambar 26. Grafik penyerangan *Brute force* terhadap Memori

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengujian keamanan jaringan yang telah dilakukan, dapat disimpulkan bahwa dengan adanya penambahan metode IPTables dapat meningkatkan kinerja baik dari segi penggunaan CPU (38,4%) dan Memory (44,2%) pada *server* maupun keamanan jaringan dibandingkan dengan hanya menggunakan metode *Port knocking* dan *Honeypot* saja. Pada penelitian ini juga terdapat peningkatan maupun penurunan pada Memori dan CPU server yang disebabkan oleh berjalanya sistem *Honeypot* pada saat penyerangan masuk menuju server

5.2. Saran

Beberapa saran yang dapat dijadikan pertimbangan dalam mengembangkan penelitian ini adalah:

a. Menambah aturan dalam sistem basis data yang dapat memberikan respon terhadap penyerang pada *Honeypot*
b. Membuat aplikasi atau *scrip* agar dapat berjalan di *background* server

DAFTAR PUSTAKA

[1] R. Apriani, A. H. Jatmika, dan I. W. A. Arimbawa, “Implementasi Metode Intrusion Detection System (IDS) dan Port Knocking Pada Serangan Sistem Keamanan Dalam Jaringan Komputer.”

[2] Pusat Operasi Keamanan Siber Nasional Badan Siber dan Sandi Negara, “Indonesia Cyber Security Monitoring Report 2019,” *Indones. Secur. Incid. Response Team Internet Infrastruct.*,

p. 42, 2020.

[3] W. Wilman, I. Fitri, dan N. D. Nathasia, "Port Knocking Dan Honeypot Sebagai Keamanan Jaringan Pada Server Ubuntu Virtual," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 3, no. 1, pp. 27–33, 2018, doi: 10.37438/jimp.v3i1.86.

[4] Hawari. Mizan Syarif, "Penerapan Iptables Firewall Pada Linux Dengan Menggunakan Fedora," *J. Manaj. Inform.*, vol. 6, no. 1, pp. 198–207, 2016.

[5] P. Soepomo, "Penerapan Sistem Keamanan Honeypot dan Ids pada Jaringan Nirkabel (Hotspot)," vol. 1, no. 1, pp. 111–118, 2013, doi: 10.12928/jstie.v1i1.2512.

[6] M. Suyuti Ma'sum, M. Azhar Irwansyah, dan H. Priyanto, "Analisis Perbandingan Sistem Keamanan Jaringan Menggunakan Snort dan Netfilter," *J. Sist. dan Teknol. Inf.*, vol. 5, no. 1, pp. 56–60, 2017.

[7] A. P. A. Kusuma and Asmunin, "Implementasi Simple Port Knocking Pada Dynamic Routing (OSPF) Menggunakan Simulasi GNS3," *J. Manaj. Inform.*, vol. 5, no. 2, pp. 7–17, 2016.

[8] I. Sari, M. Yamin, L. M. F. Aksara, J. T. Informatika, F. Teknik, and U. H. Oleo, "Sistem Monitoring Serangan Jaringan Komputer Berbasis WEB Service Menggunakan Honeypot Sebagai Intrusion Prevention System," vol. 5, no. 1, pp. 35–44, 2019.

[9] G. Sondakh, M. E. I. Najoan, dan A. S. Lumenta, "Perancangan Filtering Firewall Menggunakan Iptables Di Jaringan Pusat Teknologi Informasi Unsrat," *E-Journal Tek. Elektro Dan Komput.*, vol. 3, no. 4, pp. 19–27, 2014.

[10] E. Sularno, "Analisa Dan Implementasi Iptables Dengan Debian Server Sebagai Filtering Firewall Web," vol. 3, no. 1, pp. 106–121, 2016.

[11] A. Amarudin, "Desain Keamanan Jaringan Pada Mikrotik Router OS Menggunakan Metode Port Knocking," *J. Teknoinfo*, vol. 12, no. 2, p. 72, 2018, doi: 10.33365/jti.v12i2.121.

[12] P. Riska, P. Sugiartawan, dan I. Wiratama, "Sistem Keamanan Jaringan Komputer Dan Data Dengan Menggunakan Metode Port Knocking," *J. Sist. Inf. dan Komput. Terap. Indones.*, vol. 1, no. 2, pp. 53–64, 2018, doi: 10.33173/jsikti.12.

[13] N. Arkaan dan D. V. S. Y. Sakti, "Implementasi Low Interaction Honeypot Untuk Analisa Serangan Pada Protokol SSH," *J. Nas. Teknol. dan Sist. Inf.*, vol. 5, no. 2, pp. 112–120, 2019, doi: 10.25077/teknosi.v5i2.2019.112-120.

[14] I. G. L. P. E. Prismana dan B. Chilmi, "Implementasi Simulasi Jaringan Komputer Multi Device Dengan Menggunakan GNS3," 2015.