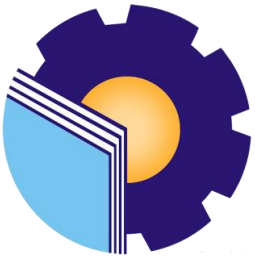
LAPORAN TUGAS AKHIR

IMPLEMENTASI *PORT KNOCKING* PADA LABORATORIUM JURUSAN TEKNIK INFORMATIKA (STUDI KASUS: LABORATORIUM *HIGH PERFORMANCE COMPUTING*)

Sebagai Salah Satu Syarat Untuk Menyelesaikan

Program Studi Diploma III Jurusan Teknik Informatika



Oleh :

MUTIARA KRISTINA BR SINAGA

6103211479

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BENGKALIS

BENGKALIS

TAHUN 2024

HALAMAN PENGESAHAN

LAPORAN TUGAS AKHIR

IMPLEMENTASI *PORT KNOCKING* PADA LABORATORIUM JURUSAN TEKNIK INFORMATIKA (STUDI KASUS: LABORATORIUM *HIGH PERFORMANCE COMPUTING*)

Oleh :

MUTIARA KRISTINA BR SINAGA
6103211479

Telah diujikan dan dinyatakan lulus ujian tugas akhir pada tanggal 08 Agustus 2024
oleh tim penguji Program Studi Diploma III
Teknik Informatika

Pembimbing

Wahyat, M.Kom
NIP. 198911262020121006

Bengkalis, 08 Agustus 2024
Anggota Tim Penguji

Lipantri Mashur Gultom, M.Kom
NIP. 198708122019031010

Tengku Musri, M.Kom
NIP. 198503082024211009

Desi Amirullah, M.T
NIP. 198712092019031010

Mengetahui,
Ketua Jurusan Teknik Informatika

Rasmawi, M.Kom
NIP. 197706072014041001

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa Laporan Tugas Akhir ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar Ahli Madya di Politeknik Negeri Bengkalis Jurusan Teknik Informatika, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Bengkalis, 08 Agustus 2024

MUTIARA KRISTINA BR SINAGA

NIM. 6103211479

PERNYATAAN PERSETUJUAN

PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN

AKADEMIS

Sebagai civitas akademik Politeknik Negeri Bengkalis, saya yang bertanda tangan di bawah ini :

Nama	:	Mutiara Kristina Br Sinaga
Nim	:	6103211479
Program Studi	:	D-III Teknik Informatika
Jenis Karya	:	Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Bengkalis **Hak Bebas Royalti Noneksklusif** *(None-exclusive Royalty Free Right)* atas karya ilmiah saya yang berjudul :

IMPLEMENTASI *PORT KNOCKING* PADA LABORATORIUM JURUSAN TEKNIK INFORMATIKA (STUDI KASUS: LABORATORIUM *HIGH PERFORMANCE COMPUTING*)

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Akses Royalti/Noneksklusif ini Politeknik Negeri Bengkalis berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data *(database)*, merawat dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini saya buat dengan sebenarnya,

Dibuat di : Bengkalis

Pada Tanggal : 08 Agustus 2024

Yang menyatakan,

(Mutiara Kristina Br Sinaga)

6103211479

LEMBAR PERSEMBAHAN

In The Name Of Jesus Christ

"Mulialah harimu dengan mempersiapkan karir untuk masa depan Mu"

Doakan apa yang dikerjakan, Kerjakan apa yang di Doakan.

"Bersukacitalah dalam Pengharapan, Sabarlah dalam Kesesakan, dan Bertekunlah dalam Doa"

(Roma 12:12)

Keluarga Tercinta

Kepada cinta pertama dalam hidup penulis Ayah Mangiring Sinaga , Ibu Tercinta Esti Nurhayati Br Simatupang, dan Saudara tersayang Tomi Sinaga, Sarianto Sinaga, Imanuel Sinaga, Firnando Sinaga. Tugas Akhir ini penulis persembahkan untuk penyemangat hidup dan sumber kebahagiaan bagi penulis. Terima kasih penulis ucapkan atas kasih sayang, semangat, doa, dan dukungan yang selalu terlimpahkan.

Keluarga Besar dan Teman – Teman

Teruntuk Seluruh keluarga besar Penulis Sinaga dan yang selalu melimpahkan kasih sayang dan dukungan, serta teman-teman penulis dan sahabat tercinta yaitu Adellia Fitri yang telah membantu dan menemani penulis dalam pembuatan Tugas Akhir.

Dosen Pembimbing dan Dosen Wali

Terimakasih kepada Bapak Wahyat M.Kom selaku dosen pembimbing dan dosen wali penulis yang selalu senantiasa memberikan ilmu dan selalu sabar memberikan arahan, memberikan waktu, pikiran untuk mengarahkan Tugas Akhir saya sehingga Tugas Akhir ini dapat diselesaikan.

IMPLEMENTASI *PORT KNOCKING* PADA LABORATORIUM JURUSAN TEKNIK INFORMATIKA (STUDI KASUS: LABORATORIUM *HIGH PERFORMANCE COMPUTING*)

Nama Mahasiswa : Mutiara Kristina Br Sinaga

Nim : 6103211479

Dosen Pembimbing : Wahyat M.Kom

ABSTRAK

Mengamati beberapa masalah dalam pengelolaan jaringan *mikrotik* di Laboratorium *High Performance Computing* (HPC) pada Politeknik Negeri Bengkalis. Tidak ada pembagian prioritas bagi pengguna misalnya tidak ada pembagian *bandwith* antara dosen dan mahasiswa, dan masalah utamanya adalah kurangnya optimalisasi keamanan jaringan *mikrotik*. Maka dari itu solusi yang tepat pada Laboratorium HPC yaitu sesuaikan prioritas dan pembagian *bandwith* sesuai kebutuhan berdasarkan pengguna dan juga pembuatan halaman *login page mikrotik* untuk melakukan autentikasi dan informasi pengguna. Fokus yang utama adalah implementasi untuk solusi meningkatkan keamanan *mikrotik* dan *web server* menggunakan metode *port knocking* dan keamanan *firewall raw*. Implementasi *port knocking* dengan berbagai *rule* diantaranya menangkap *traffic* menggunakan *protocol* ICMP (*Internet Control Message Protocol)* atau lakukan *ping* yang masuk ke *router mikrotik*, kemudian lakukan hal yang sama untuk menangkap *traffic* pada *port* Telnet dan SSH (*Secure shell*) dengan *protocol* TCP (*Transmission Control Protocol*) dengan Tujuan Implementasi *port knocking* adalah menyembunyikan layanan yang aktif dari penyerang dan memberikan lapisan tambahan verifikasi sebelum memberikan akses ke layanan yang diinginkan dan sebagai mitigasi serangan DDOS. Pengujian yang akan dilakukan melibatkan evaluasi sebelum dan setelah penerapan *Port knocking* pada *mikrotik* dengan pengujian *Port Scanning* menggunakan NMAP pada *Kali Linux* dan pengujian selanjutnya pada keamanan *web server* yaitu DVWA (*Damn Vulnerable*

Web Application) menggunakan *firewall raw* dengan pengujian serangan SYN *flood* dan HTTP *flood* menggunakan LOIC. Hasil pengujian yang dapat menunjukkan penurunan signifikan dalam upaya akses yang tidak sah dan juga mendapatkan hasil peningkatan jaringan keseluruhan.

Kata Kunci: *Port Knocking, Mikrotik*, Firewall Raw, Penyerang.

PORT KNOCKING IMPLEMENTATION IN THE LABORATORY OF INFORMATICS ENGINEERING DEPARTMENT (CASE STUDY: HIGH PERFORMANCE COMPUTING LABORATORY)

Name Of Student	: Mutiara Kristina Br Sinaga
Student Identification Number	: 6103211479
Supervisor	: Wahyat, M.Kom

ABSTRACT

Observed several problems in managing the proxy network in the High Performance Computing (HPC) Laboratory at Politeknik Negeri Bengkalis. There is no priority division for users, for example, there is no bandwidth division between lecturers and students, and the main problem is the lack of optimisation of proxy network security. Therefore, the right solution for the HPC Laboratory is to adjust priorities and bandwidth distribution according to user needs and also create a proxy login page to authenticate and user information. The main focus is the implementation of solutions to improve proxy and web server security using the port knocking method and raw firewall security. Implementation of port knocking with various rules including capturing traffic using the ICMP (Internet Control Message Protocol) protocol or pinging that enters the proxy router, then do the same to capture traffic on the Telnet and SSH (Secure shell) ports with the TCP (Transmission Control Protocol) protocol with the purpose of implementing port knocking is to hide active services from attackers and provide an additional layer of verification before providing access to the desired service and as mitigation of DDOS attacks. Tests that will be carried out involve evaluation before and after the application of Port knocking on proxy with Port Scanning testing using NMAP on Kali Linux. The next test on web server security is DVWA (Damn Vulnerable Web Application) using a raw firewall by testing SYN flood

and HTTP flood attacks using LOIC. The test results can show a significant decrease in unauthorised access attempts and also get overall network improvement results.

Keywords: Port Knocking, Mikrotik, Firewall Raw, Attacker.

KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa, karena berkat Rahmat-Nya penulis dapat menyelesaikan Tugas Akhir dengan judul **"Implementasi *Port Knocking* pada Laboratorium Jurusan Teknik Informatika**" pada Laboratorium HPC (*High Performance Computing*). Tujuan penulisan ini adalah untuk memenuhi salah satu syarat kelulusan pada Program Studi Teknik Informatika Politeknik Negeri Bengkalis Oleh karena itu penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1 Bapak Johny Custer S.T, M.T selaku Direktur Politeknik Negeri Bengkalis.
2 Bapak Kasmawi M.Kom selaku Ketua Jurusan Teknik Informatika.
3 Bapak Supria M.kom selaku Koordinator Program Studi D-III Teknik Informa tika.
4 Bapak Wahyat M.kom selaku Dosen Pembimbing Tugas Akhir.
5 Bapak Eko Prayitno M.Kom dan Bapak Nurul Fahmi M.T selaku Koordinator Proposal Tugas Akhir.
6 Kedua orang tua beserta seluruh keluarga dan teman-teman seperjuangan yang memberikan dorongan, motivasi dan semangat sehingga bisa menyelesaikan proposal ini.

Penulis menyadari bahwa masih banyak kekurangan pada Laporan Tugas Akhir (TA) ini. Maka penulis mengharapkan pembaca untuk memberikan saran dan kritik yang dapat membangun ke depannya dan semoga dapat memberikan manfaat kepada pembaca dimasa yang akan datang. Akhir kata, Penulis mengucapkan terimakasih.

Bengkalis, 08 Agustus 2024

Penulis

Mutiara Kristina Br Sinaga

Nim. 6103211479

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Jaringan Komputer adalah sekumpulan komputer yang saling terhubung melalui suatu media perantara seperti *switch*, *router*, *wireless*, kabel yang masing-masing komputer dapat bertukar informasi dan bertukar data karena berada dalam satu ruang lingkup yang disebut jaringan (Afdhol. P. Y. et al., 2023). Pesatnya perkembangan teknologi saat ini nyatanya memiliki dampak serangan terhadap jaringan pada sistem administrator. Sebagian besar jaringan komputer yang mengalami permasalahan dalam membangun sebuah jaringan komputer, hal ini dapat membuka peluang bagi para *hacker* untuk meretas dan merusak jaringan yang dibangun tersebut (Amarudin, 2018).

Sehingga bagi para pengguna teknologi yang terhubung dengan jaringan lokal maupun internet perlu waspada terhadap serangan yang dilakukan oleh pihak yang tidak bertanggung jawab. Banyak serangan-serangan yang bisa dilakukan dalam keadaan *port-port* yang terbuka diantaranya *virus, malware* dan *Trojan*. Walaupun sudah diatur dalam dalam Undang-Undang Nomor 11 Tahun 2008 Tentang Internet dan Transaksi Elektronik (ITE) pada Pasal 30 UU ITE tahun 2008 ayat 3 yang berbunyi: “Setiap orang dengan sengaja dan tanpa hak atau melawan hukum mengakses komputer dan Sistem Elektronik dengan cara apapun dengan melanggar, menerobos, melampaui, atau menjebol sistem pengaman”.

Ancaman pidana pasal 45 ayat 3 setiap orang yang memenuhi unsur sebagaimana dimaksud dalam Pasal 30 ayat 3 dipidana dengan pidana penjara paling lama 8 (delapan) atau denda paling banyak Rp 800.000.000 (delapan ratus juta rupiah) dan di dalam Kitab Undang-Undang Hukum Pidana Pasal 406 KUHP dapat di kenakan pada

kasus *deface* atau *hacking* yang membuat kerusakan atau melakukan modifikasi sistem milik orang lain. Sistem milik orang lain, yaitu :

1. Sistem *Website* atau Portal.
2. Sistem *Server*.
3. Sistem Jaringan.
4. Sistem Aplikasi.

Fokus Utama jika penyerang melakukan modifikasi atau kerusakan pada sistem jaringan khususnya pada *router mikrotik*. Keamanan jaringan *mikrotik* rentan terhadap serangan *port* yang terbuka dapat dihindari dengan menerapkan *port knocking* pada *router mikrotik*. *Simple port knocking* yang diterapkan agar sistem yang dibangun mampu mendeteksi dan menghindari serangan yang berbahaya terhadap jaringan dan langsung memberikan peringatan kepada pengelola jaringan (administrator) tentang kondisi jaringan yang sedang berjalan pada saat kejadian yang berlangsung (Saputro et al., 2020).

Port knocking merupakan suatu sistem keamanan yang bertujuan untuk membuka atau menutup akses *block* ke *port* tertentu dengan menggunakan *firewall* pada perangkat jaringan dengan cara mengirimkan paket atau koneksi tertentu. Koneksi berupa *protocol* TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), maupun ICMP (*Internet Control Message Protocol*). Sehingga untuk masuk dan menggunakan akses ke *port* tertentu yang telah dibatasi, maka *user* harus mengetuk terlebih dahulu dengan memasukan *rule* yang harus dilakukan terlebih dahulu. *Rule* yang hanya diketahui oleh pihak penyedia jaringan (administrator jaringan).

Sebuah sistem harus memiliki keseimbangan antar keamanan dan *fleksibilitas*. Satu cara untuk dapat mencapai sistem seperti demikian yaitu dengan mengakses *firewall*. Maka dengan menggunakan *firewall* secara tidak langsung kita dapat mendefinisikan *user* yang dapat dipercaya dan yang tidak dapat dipercaya dengan menggunakan alamat IP sebagai kriteria *filter* (Novianto et al., 2021). Hasilnya adalah sebuah sistem keamanan jaringan yang mengamankan *mikrotik* dengan metode *port knocking*. Hal ini

berfungsi sebagai alternatif untuk menjaga keamanan dalam jaringan komputer, mencegah penyerang untuk mengakses *router mikrotik* dan memungkinkan administrator untuk menentukan siapa mereka. Hanya mereka yang memiliki hak akses yang dapat masuk ke *port* tertentu (Blaise et al., 2020)

Saat ini permasalahan pada Laboratorium *High Performance Computing* (HPC) melibatkan sejumlah kendala yang signifikan dalam pengelolaan jaringan *mikrotik*. Salah satunya terkait masalah dalam pembagian *bandwith* antara dosen dan mahasiswa. Ketidakoptimalan dalam aspek keamanan jaringan *mikrotik* pada Laboratorium HPC menjadi fokus utama. Kurangnya tindakan keamanan dapat menyebabkan risiko tinggi terhadap akses yang tidak sah dan potensi ancaman keamanan yang serius. Oleh karena itu, diperlukan suatu solusi yang dapat meningkatkan keamanan jaringan pada *mikrotik* terutama ketika mengamankan *web server* yang berada di belakang NAT pada *router mikrotik* di Laboratorium HPC.

Solusi yang diusulkan melibatkan serangkaian tindakan konkret. Pertama, mengoptimalkan pembagian *bandwith* dan *hotspot user* dengan memperhatikan prioritas dan memastikan keseimbangan pembagian antara dosen dan mahasiswa. Pembuatan halaman *login page* pada perangkat *mikrotik* juga diperlukan sebagai langkah pengenalan pengguna. Tidak hanya memberikan lapisan keamanan tambahan, tetapi juga memberikan informasi yang berguna tentang pengguna. Selanjutnya yang menjadi implementasi utama untuk pengamanan jaringan pada *router mikrotik* untuk mengamankan dari potensi penyerang dan memberikan verifikasi tambahan sebelum memberikan akses ke layanan yang diinginkan pada *mikrotik* dan *mengamankan web server* yang berada pada NAT yang ada pada *mikrotik* dengan menggunakan *port knocking* dan *firewall raw* merupakan solusi yang sarankan untuk mengamankan *router mikrotik* yang berada pada Laboratorium *High Performance Computing* (HPC).

Mekanisme/skema sederhana ketika terjadi serangan pada *router mikrotik* di Laboratorium *High Performance Computing* (HPC). *Port knocking* diterapkan melalui 4 *rule* yaitu *rule* pertama lakukan *ping IP address router mikrotik* dengan *protocol* ICMP dan *rule* kedua lakukan *ping IP address router mikrotik* dengan *port telnet* dengan *protocol* TCP, *rule* ketiga lakukan hal yang sama untuk menangkap *traffic* pada *port* SSH dengan *protocol* TCP dan *rule* terakhir yaitu *droping packet* yang masuk kedalam *router* kecuali yang sudah melakukan *ping* kemudian *telnet* dan SSH seb elumnya. Rangkaian proses *knocking* harus sesuai dengan urutan *rule port knocking* yang telah dikonfigurasi, jika proses *knocking* tidak sesuai urutan maka tetap akan di *block* oleh *firewall filter* yang ada pada *router mikrotik*. Berdasarkan uraian di atas maka diperlukan pemanfaatan metode *port knocking* dalam sistem keamanan jaringan untuk mengurangi tingkat risiko, terutama pada jaringan yang ingin melakukan kejahatan dengan cara memindai *port-port* yang terbuka melalui perangkat *mikrotik* dan melakukan penyerangan *DDOS attack*.

Maka dari itu dilakukan pengujian pemanfaatan *Port knocking* yang dapat menjadi komponen penting dalam strategi mitigasi serangan DDoS, terutama ketika dihubungkan dengan perlindungan terhadap *web server* yang diamankan melalui *firewall raw* pada *mikrotik*. Dengan *port knocking*, akses *ke port-port* penting pada *router* hanya diberikan kepada pengguna yang mengikuti urutan *knocking* yang telah ditentukan, seperti mengirimkan paket ICMP, TCP Telnet, dan SSH secara berurutan. Hal ini membuat *port-port* tersebut tidak terlihat oleh penyerang yang mencoba melakukan *port scanning* atau melancarkan serangan DDoS seperti *SYN flood* dan *HTTP flood* dengan LOIC (*Low Orbit Ion Cannon*). Pada serangan *SYN flood*, di mana penyerang mencoba membanjiri *port* dengan permintaan koneksi yang tidak pernah diselesaikan, *port knocking* dapat mencegah serangan tersebut dengan menyembunyikan *port* dari akses langsung kecuali telah mengikuti urutan *knocking* yang benar. Sementara itu, untuk serangan *HTTP flood* yang berusaha membanjiri *web server* dengan permintaan HTTP palsu, *firewall raw mikrotik* dapat diatur untuk hanya menerima trafik dari klien yang telah diizinkan melalui proses *port knocking*. Maka

port knocking berfungsi sebagai gerbang pertama yang mencegah serangan DDoS menjangkau *web server*, sementara *firewall raw* menangani dan mem*filter* trafik yang masuk, memastikan hanya trafik yang sah yang dapat mencapai *web server*. Kombinasi dari *port knocking* dan *firewall raw* ini menciptakan lapisan keamanan yang kuat, mengurangi risiko kerusakan yang dapat disebabkan oleh serangan DDoS, dan menjaga ketersediaan serta kinerja *web server* di Laboratorium HPC. Solusi-solusi di atas yang telah dirancang dapat mengurangi resiko ancaman yang akan mengganggu aktivitas yang sedang berlangsung, disesuaikan dengan kondisi pada Laboratorium HPC.

1.2 Perumusan Masalah

Hasil dari permasalahan yang sudah diuraikan dari latar belakang masalah dapat di rumuskan permasalahaan yang ada pada pembuatan Metode Keamanan Jaringan *Port knocking mikrotik* pada Laboratorium *High Performance Computing* (HPC) yaitu sebagai berikut:

1. Bagaimana implementasi *port knocking* menggunakan rangkaian proses 3 ketukan yaitu *protocol* ICMP, lalu ketukan pada *port* Telnet dan *port* SSH yang ada pada *protocol* TCP dan implementasi *firewall filter* menggunakan *mikrotik RouterBoard* pada sistem keamanan jaringan Laboratorium *Performance Computing* (HPC)?
2. Bagaimana mengatur 4 baris *rule* tertentu, agar komputer yang tidak di kenali untuk mengakses ke *port* tujuan tertentu yang sedang terbuka tetapi tidak dapat masuk, tapi perangkat yang sudah dikenal dapat mengakses ke *port* tujuan dengan mengikuti 4 baris *rule* yang dibuat ?
3. Bagaimana efektivitas pengujian *port knocking* dalam meningkatkan keamanan *web server* dalam *mikrotik* menggunakan *firewall raw* dari serangan DDOS seperti SYN *flood* dan HTTP *flood*?

1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, maka batasan masalah dalam metode *port knocking* pada Laboratorium *High Performance Computing* (HPC) adalah sebagai berikut:

1. Implementasi *port knocking* menggunakan dengan rangkaian proses 3 ketukan yaitu *protocol* ICMP, lalu ketukan pada *port* Telnet dan *port* SSH yang ada pada *protocol* TCP yang hanya dibuat untuk Laboratorium *High Performance Computing* (HPC).
2. Pengguna hanya bisa mengakses *port-port* dengan *rules* yang dibuat untuk dapat mengaksesnya.
3. Pembuatan *port knocking* dengan *Mikrotik RouterBoard* RB951Ui 2nd HaP.
4. Uji coba *port knocking router mikrotik* dan keamanan *firewall filter* penelitian ini pada peran *Kali Linux* dengan melakukan *Port Scanning.*
5. Uji coba efektivitas *port knocking* pada keamanan *web server* yang di *forward* ke dalam NAT *mikrotik* dengan *firewall raw*. Penelitian ini pada peran LOIC sebagai sistem penyerang dengan jenis serangan DDOS *attack* yaitu berupa serangan SYN *flood* dan HTTP *flood*.

1.4 Tujuan Penelitian

Tujuan keamanan *port knocking* pada Laboratorium *High Performance Computing* (HPC) sebagai berikut:

1. Implementasi *port knocking mikrotik* menggunakan rangkaian proses 3 ketukan yaitu *protocol* ICMP, lalu ketukan pada *port* Telnet dan *port* SSH yang ada pada *protocol* TCP menggunakan *mikrotik RouterBoard* pada Laboratorium *High Performance Computing* (HPC) untuk meningkatkan keamanan jaringan.
2. Membuat 4 baris *rule* terhadap *router mikrotik*, agar pengguna yang tidak di kenali tidak dapat memiliki akses untuk masuk pada *port* tertentu yang

terbuka, tetapi dengan menganalisis dan di kenali oleh sistem keamanan jaringan komputer yang diberi akses oleh sistem administrator untuk perangkat yang sudah di kenali bisa dapat mengakses dengan menggunakan jaringan LAN (*Local Area Network*) ataupun internet dengan menggunakan *rule* yang dibuat.

3. Merancang dan melaksanakan pengujian dengan menggunakan *Kali Linux* dan LOIC dengan *Port Scanning* dan mitigasi serangan DDOS dengan jenis serangan SYN *flood* juga HTTP *flood* dalam meningkatkan keamanan *web server* pada *mikrotik* untuk mengukur efektivitas dan peran *port knocking* di *mikrotik*.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari keamanan jaringan menggunakan *port knocking* pada Laboratorium *High Performance Computing* (HPC) ini yaitu sebagai berikut:

1. Dengan adanya keamanan jaringan *port knocking mikrotik* menggunakan rangkaian proses 3 ketukan yaitu *protocol* ICMP, lalu ketukan pada *port* Telnet dan *port* SSH yang ada pada *protocol* TCP untuk mengamankan jaringan komputer yang ada dan mengurangi risiko serangan akses yang tidak sah dengan menggunakan *port knocking*.
2. Bagi administrator jaringan Laboratorium *High Performance Computing* (HPC) dapat memiliki hak untuk mengakses dan memasuki *port-port* tertentu dengan mengikuti *rule* yang dibuat.
3. Mitigasi serangan DDOS dengan kombinasi *port knocking* dan *firewall raw* memberikan perlindungan berlapis terhadap serangan DDoS seperti *SYN flood* dan *HTTP flood*, memastikan bahwa hanya trafik yang sah yang dapat mencapai *web server*, sehingga menjaga stabilitas dan ketersediaan layanan.

1.6 Metode Penyelesaian Masalah

Metode Penyelesaian masalah dalam penelitian keamanan jaringan pada Laboratorium *High Performance Computing* yaitu sebagai berikut : Melakukan Identifikasi Masalah yang terjadi pada Laboratorium HPC dengan melakukan wawancara kepada Kepala Laboratorium HPC dan Laboran Laboratorium HPC dan kepada *Network Administrator* yang membangun infrastruktur jaringan komputer dan perangkat keras pada Laboratorium HPC kemudian Perancangan dan implementasi *port knocking* dan keamanan *firewall raw* pada *router mikrotik.* Dengan menggunakan *port knocking*, hanya trafik dari klien yang sah yang dapat mencapai *port-port*, sementara *firewall raw* menangani dan memblokir serangan yang lebih spesifik seperti *SYN flood* dan *HTTP flood* berdasarkan hasil wawancara maka akan melakukan analisis pada objek penelitian yang akan diteliti, dengan berdasarkan analisis yang difokuskan pada fungsi kualitas dari yang ada pada lokasi penelitian. Pada penelitian ini akan di lakukan analisis pada sistem keamanan jaringan komputer dengan menggunakan metode *port knocking*. Kemudian melakukan pengujian *Port knocking* dengan mitigasi serangan DDoS serta peran *firewall raw mikrotik* dalam melindungi *web server* dari serangan *SYN flood* dan *HTTP flood*. Berdasarkan wawancara tersebut, *port knocking* diterapkan untuk mengamankan *port-port* sensitif pada *mikrotik*, sehingga hanya pengguna yang mengikuti urutan *knocking* yang benar dapat mengakses *port* tersebut. Ini berfungsi sebagai lapisan pertama perlindungan terhadap serangan DDoS, dengan menyembunyikan *port* dari akses umum dan mengurangi kemungkinan serangan seperti *SYN flood* dan *HTTP flood* mencapai *web server*. kesimpulan, dimana pada penelitian ini hasil dicapai adalah apakah sistem jaringan di lokasi penelitian aman melalui *port* jaringan yang sudah ditentukan.

BAB II
TINJAUAN PUSTAKA

2.1. Kajian Terdahulu

Penelitian ini memiliki beberapa referensi terkait judul penelitian terdahulu yaitu penelitian dari (Novianto et al., 2021) yang berjudul "Implementasi Sistem Keamanan Jaringan Menggunakan Metode *Simple Port Knocking* pada *Router* Berbasis *Mikrotik*". Salah satu metode yang dapat digunakan untuk meningkatan keamanan sistem jaringan komputer adalah metode *simple port knocking*. *Simple port knocking* diterapkan agar sistem yang dibangun mampu mendeteksi dan menghindari serangan yang berbahaya terhadap jaringan dan langsung memberikan peringatan kepada pengelola jaringan (administrator) tentang kondisi jaringan yang sedang berjalan pada saat kejadian berlangsung. Penerapan *simple port knocking* menggunakan media *router mikrotik* yang berfungsi untuk merubah konfigurasi *setting* dan proteksi *router* sehingga tetap aman dari serangan *cracker*.

Menurut (Santoso et al., 2022) Yang judul Penelitian "Implementasi Keamanan Jaringan Menggunakan *Port Knocking*". Teknologi informasi harus diperbarui setiap tahun karena masalah keamanan data dan informasi. Keamanan informasi menjadi semakin penting seiring dengan perubahan teknologi informasi dan masih terus berubah hingga saat ini. Serangan pada *server* telah sering dilakukan oleh pengguna yang ceroboh. Keamanan jaringan perlu ditingkatkan untuk mengurangi penyalahgunaan jaringan *hacker*. Pada penelitian ini *port knocking* digunakan untuk melakukan penelitian untuk pembuatan jaringan komputer yang aman. Berdasarkan hasil analisis dan pengujian implementasi sistem, dapat disimpulkan bahwa sistem dapat berfungsi secara efektif dan keamanan jaringan dapat ditingkatkan dibandingkan

dengan keamanan non-jaringan. Pasang keamanan *port knocking* pada tempatnya. Kehadiran otentikasi yang sesuai saat mengakses adalah buktinya.

Berdasarkan penelitian terdahulu (Keamanan et al., 2022) yang berjudul "Sistem Keamanan Jaringan Komputer dan Data Dengan Menggunakan Metode *Port Knocking*" Seiring dengan perkembangan teknologi informasi saat ini yang selalu berubah, menjadikan keamanan suatu informasi sangatlah penting. Banyak serangan yang dilakukan oleh orang-orang yang tidak bertanggung jawab melakukan serangan terhadap *server*. Serangan-serangan tersebut sering dilakukan pada suatu *port-port* yang dalam keadaan terbuka, sehingga nantinya akan membuat orang-orang yang tidak mempunyai hak akses maupun yang tidak berkepentingan dapat dengan mudah mengendalikan *port-port* yang telah dimasuki. Maka untuk melakukan keamanan pada jaringan komputer dalam mengatasi serangan pada *port-port*, salah satunya adalah dengan menggunakan metode *port knocking*. Untuk menghindari serangan yang dilakukan dalam keadaan *port* terbuka maka digunakan suatu metode *port knocking* dan mengatur parameter-parameter agar perangkat komputer ini tidak memiliki *port* komunikasi yang bebas untuk dimasuki, tetapi perangkat masih tetap dapat diakses dari luar. Sehingga akan membuat orang yang tidak memiliki kesempatan unruk memasuki *port-port* yang ada.

Menurut penelitian Terdahulu (Yudi mulyanto et al., 2021) yang berjudul " Implementasi Port Knocking Untuk Keamanan Jaringan SMKN1 Sumbawa Besar". Keamanan jaringan komputer atau *Computer Network Security* sangat berhubungan dengan keamanan data oleh karena itu keamanan jaringan sangat penting untuk melindungi data dari berbagai serangan pihak-pihak yang tidak bertanggung jawab. Serangan tersebut dapat ditujukan terhadap instansi, perusahaan atau lembaga tertentu, tidak terkecuali Sekolah Menengah Kejuruan Negeri 1 Sumbawa Besar yang mengalami hal tersebut. Penelitian dilakukan untuk menganalisa dan mengimplementasikan metode *port knocking* dalam keamanan jaringan dan agar dapat mencegah serangan pada *port-port* jaringan komputer SMKN 1 Sumbawa Besar. Peneliti melakukan peningkatan keamanan jaringan menggunakan metode *port*

knocking yang dapat membantu meningkatkan keamanan jaringan dan membantu administrator dalam mengamankan *Mikrotik Routerboard* pada sistem jaringan komputer SMKN 1 Sumbawa Besar. Adapun metode yang digunakan dalam pengembangan jaringan yaitu menggunakan metode *Network Development Life Cycle* (NDLC) yang terdiri dari enam tahapan yaitu analisis, perancangan, simulasi, *prototype*, penerapan dan *monitoring*.

Berdasarkan penelitian terdahulu (Setyowibowo & Moka, 2022). Kebutuhan internet yang tinggi di SMK Cakra Kusuma Jombang tidak diimbangi dengan sistem keamanan yang baik. Beberapa serangan yang digunakan antara lain *brute force*. Serangan *brute force* pada *mikrotik* mengakibatkan internet menjadi tidak stabil karena penyerang dapat mengubah konfigurasi *mikrotik*. Keamanan *admin* jaringan menjadi penting sebab *admin* merupakan *user* yang mengelola jaringan sekolah. *Simple port knocking* merupakan sebuah solusi jika terjadi serangan pihak luar yang ingin masuk lewat *port mikrotik* dengan *brute force*. Hasil pengujian dapat dilihat perbedaan tingkat keamanan dari serangan *brute force*. Dengan menerapkan *simple port knocking* dapat mencegah serangan *brute force* pada *admin mikrotik*. Walaupun dengan memasukkan *username* dan *password* yang benar tidak dapat masuk *mikrotik* karena ada *rule port* yang ada pada keamanan *simple port knocking*. Dengan *automated backup* sekolah tidak perlu lagi mendatangkan teknisi bila terjadi *error* pada *mikrotik* karena *backup* konfigurasi bisa digunakan untuk mengembalikan konfigurasi *mikrotik.*

Maka dari penelitian-penelitian terdahulu dan teori-teori yang sudah dipaparkan. Maka terdapat perbedaan dan persamaan dengan penelitian ini, yang dapat dilihat dari tabel berikut :

Table.2. 1 Penelitian- penelitian terdahulu

Sumber : (Data Olahan, 2024)

Aspek	(Novianto et al., 2021)	(Santoso et al., 2022)	(Keamanan et al., 2022)	(Yudi mulyanto et al., 2021)	(Setyowibowo & Moka, 2022)
Tema Penelitan	Implementasi Sistem Keamanan Jaringan Menggunakan Metode *Simple Port Knocking*	Implementasi keamanan jaringan Menggunakan *Port Knocking*	Sistem Keamanan Jaringan Komputer dan Data Dengan Metode *Port Knocking*	Implementasi *Port Knocking* Untuk Keamanan Jaringan SMKN1 Sumbawa Besar	*Simple Port Knocking* untuk Keamanan Jaringan pada *Mikrotik* SMK Cakra Kusuma Jombang
Metode Keamanan	*Simple Port Knocking*	*Port Knocking*	*Port Knocking*	*Port Knocking*	*Simple Port Knocking*
Tujuan Keamanan	Mendeteksi dan Menghindari serangan berbahaya terhadap jaringan	Meningkatkan keamanan jaringan dan memberikan peringatan kepada pengelola jaringan	Mengatasi serangan pada *port-port* yang dalam keadaan terbuka	Mencegah serangan pada *port-port* jaringan komputer SMKN1 Sumbawa Besar	Mencegah serangan *brute force* pada *Mikrotik*

Hasil Analisis dan Pengujian	Efektif dan meningkatkan keamanan jaringan dibandingkan dengan keamanan non-jaringan	Sistem dapat berfungsi efektif dan meningkatkan keamanan dibandingkan dengan keamanan non-jaringan	Mengamankan *port-port* agar tidak dapat dimasuki oleh pihak yang tidak berkepentingan	Peningkatan keamanan jaringan dengan metode *port knocking*	Mencegah serangan *brute foerce* pada admin *mikrotik* dengan *simple port knocking*

2.2. Landasan Teori

2.2.1. Mikrotik

Pada awalnya, *Mikrotik* merupakan perangkat lunak yang diinstal pada komputer untuk mengontrol jaringan. Namun, seiring berjalannya waktu, *Mikrotik* telah berkembang menjadi perangkat jaringan yang handal dan terjangkau, sering digunakan oleh pengguna di perusahaan penyedia jasa internet (ISP). *Mikrotik* memiliki dua jenis utama, yaitu *Mikrotik RouterOS* dan *Mikrotik RouterBoard.*

Mikrotik RouterOS adalah sistem operasi yang berfungsi sebagai *router* jaringan. *Software* ini mampu mengubah komputer biasa menjadi *router* jaringan yang handal. Di sisi lain, *Mikrotik RouterBoard* merupakan perangkat keras (*hardware*) yang diproduksi oleh *Mikrotik*. Perangkat ini menjalankan sistem operasi *RouterOS* dan mirip dengan mini PC terintegrasi, karena satu *board Mikrotik RouterBoard* memiliki *processor*, RAM, ROM, dan memori *flash* (Mustaqim, 2022).

Dari pengertian *Mikrotik* dapat disimpulkan bahwa *mikrotik* merupakan salah satu solusi untuk masalah keamanan jaringan komputer, karena *fitur-fitur* dalam *mikrotik* dapat digunakan dalam manajemen jaringan. *Mikrotik* berfungsi sebagai perangkat jaringan komputer dengan *hardware* dan *software* yang mendukung fungsi *router*, *filtering*, *switching*, pengaturan *bandwith* dan *wireless access point*. Cocok untuk digunakan dalam jaringan perusahaan, ISP, dan *provider hotspot*.

2.2.2. Port Knocking

Port Knocking adalah suatu metode yang digunakan untuk membuka akses ke *port* tertentu yang sebelumnya telah diblokir oleh *firewall* pada perangkat jaringan. Metode ini dilakukan dengan mengirimkan paket atau koneksi khusus menggunakan *protocol* TCP, UDP, atau ICMP. Apabila koneksi yang dikirimkan oleh *host* sesuai dengan aturan *knocking* yang telah ditetapkan, maka *firewall* akan secara dinamis memberikan akses ke *port* yang sebelumnya diblokir. Dengan menerapkan *port knocking*, perangkat jaringan seperti *router* dapat menjadi lebih aman.

Administrator jaringan dapat melakukan pemblokiran terhadap *port-port* yang rentan terhadap serangan, seperti *Winbox* (TCP 8291), SSH (TCP 22), *Telnet* (TCP 23), atau *webfig* (TCP 80). Dengan demikian, jika dilakukan pemindaian *port* (*port scanning*), *port-port* tersebut akan terlihat tertutup. Meskipun demikian, dari pihak administrator jaringan masih tetap memungkinkan untuk melakukan konfigurasi dan pemantauan, namun dengan langkah-langkah khusus (*knocking*) agar mendapatkan izin dari *firewall* untuk mengakses *port* seperti *Winbox*, SSH dan lainnya (Na & Hipertensiva, n.d, 2020).

Dari pengertian *port knocking* dapat menyimpulkan bahwa *port knocking* merupakan salah satu metode keamanan jaringan yang memungkinkan akses ke *router* hanya setelah menerima koneksi upaya koneksi berurutan pada satu set *port* tertutup yang ditentukan sebelumnya. Setelah urutan upaya koneksi yang benar diterima, *RouterOS* secara dinamis menambahkan IP sumber *host* ke daftar alamat yang diizinkan maka akan dapat menghubungkan *router*.

2.2.3. IP Address

Alamat *IP Address* adalah nomor yang diberikan kepada komputer dan jaringan yang menggunakan *protocol* TCP/IP. Setiap komputer yang terhubung ke internet harus memiliki alamat IP yang berbeda atau unik, karena tidak boleh ada komputer atau perangkat jaringan lain yang memiliki alamat yang sama. Alamat IP unik ini terdiri dari 32 bit yang dibagi menjadi 4 oktet, masing-masing terdiri dari 8 bit.

00000000 . 00000000 . 00000000 . 00000000

o1 o2 o3 o4

Secara umum, alamat IP dapat dikategorikan menjadi 5 jenis, yaitu kelas A, B, C, D, dan E. Kelas IP D dan E digunakan untuk tujuan khusus. Kelas IP A, B, dan C dapat dibagi menjadi dua bagian, yaitu *network bit* dan *host bit*. *Network bit* membantu mengidentifikasi jaringan yang berbeda, sementara *host bit* berfungsi untuk mengidentifikasi perangkat *host* di dalam jaringan (Unpri Press, 2024).

Dari pengertian alamat *IP Address* dapat menyimpulkan bahwa *IP Address* merupakan alamat untuk identifikasi peralatan jaringan komputer, memungkinkan pertukaran data, akses internet, koneksi jaringan dengan *protocol* TCP/IP dan juga *IP Address* terdiri dari 5 jenis yaitu A, B, C, D, E.

2.2.4. DVWA ((*Damn Vulnerable Web Application*)/*web server*

DVWA adalah aplikasi web berbasis PHP/MySQL yang berjalan pada *protocol* HTTP yang rentan terhadap berbagai jenis celah keamanan. Tujuan utamanya adalah menjadi bantuan bagi para pemula dan profesional keamanan untuk menguji skill dalam proses keamanan aplikasi web (Armadhani et al., 2022). Sebuah *server* adalah suatu perangkat komputer yang menyimpan dan menjalankan program-program yang dapat menghasilkan informasi. Informasi

tersebut kemudian didistribusikan kepada komputer-komputer klien yang mengaksesnya. Secara sederhana, dapat berupa satu komputer yang menyediakan beberapa layanan aplikasi. Namun, dalam jaringan yang lebih kompleks, server dapat diatur untuk menyediakan satu atau beberapa layanan tertentu, sementara layanan lainnya ditangani oleh *server* lain.

Dengan kata lain, terjadi kerjasama antara beberapa *server* untuk memberikan layanan dan informasi kepada sejumlah komputer klien. Konfigurasi *server* yang kompleks seperti ini biasanya diterapkan oleh organisasi besar, seperti perusahaan kelas atas. Di sisi lain, *server* yang terdiri dari satu komputer yang melayani beberapa layanan biasanya digunakan dalam lingkungan yang lebih kecil, seperti sekolah, perkantoran, atau usaha kecil dan menengah (UKM). *Web server* merupakan jenis *server* yang memiliki fungsi untuk memberikan layanan *protocol* HTTP, contoh aplikasi *web server* yaitu : *apache, Microsoft IIS, Oracle, Tomcat, Nginx,* dll (Suryana, 2018).

2.2.5. Keamanan Komputer dan Jaringan

Keamanan komputer adalah cabang teknologi yang fokus pada melindungi informasi dalam sistem komputer. Sasarannya adalah untuk menjaga informasi dari potensi pencurian, kerusakan, atau untuk memastikan ketersediaannya, sesuai dengan prinsip-prinsip yang ada dalam kebijakan keamanan. Sementara itu, keamanan jaringan adalah konsep yang melibatkan berbagai teknologi, perangkat, dan prosedur yang didesain untuk mengenali dan mencegah akses yang tidak sah ke dalam jaringan.

Dengan kata lain, sistem keamanan jaringan bertujuan untuk mencegah orang yang tidak berhak masuk ke dalam jaringan. Fokus utama dari keamanan jaringan adalah mengurangi risiko ancaman seperti pencurian data dan kerusakan fisik pada perangkat komputer (Putri et al., 2023). Dari pengertian Keamanan Komputer dan Jaringan dapat menyimpulkan bahwa Keamanan Jaringan merupakan upaya pencegahan terhadap akses tidak sah ke dalam

jaringan komputer, bertujuan melindungi dari ancaman fisik dan logis, serta menjaga integritas data dari sistem.

2.2.6. *IP Public* dan *IP Private*

IP Public adalah alamat IP yang dapat diakses di internet. *IP Public* juga di kenal sebagai alamat *IP unicast* yang dapat dirutekan secara global. Ketika sebuah perangkat memiliki *IP public* dan terhubung ke internet, perangkat tersebut dapat diakses dari mana saja melalui internet. Namun, pemberian *IP public* tidak dapat dilakukan secara manual, melainkan melalui aturan dan proses yang ditetapkan. Pengguna dapat meminjam *IP public* dari penyedia layanan internet (ISP) untuk mendapatkan alamat IP yang bersifat publik.

Sementara itu, *IP private* adalah alamat IP yang digunakan untuk jaringan lokal. *IP Private* tidak tersedia di internet dan tidak dapat diakses dari jaringan global. Dalam praktiknya, jaringan area lokal biasanya menggunakan *IP Private*, dan koneksi antar jaringan lokal dilakukan melalui *router* (Mustaqim, 2022).

Terdapat tiga jenis IP yang dapat ditetapkan dalam skema *IP address*:

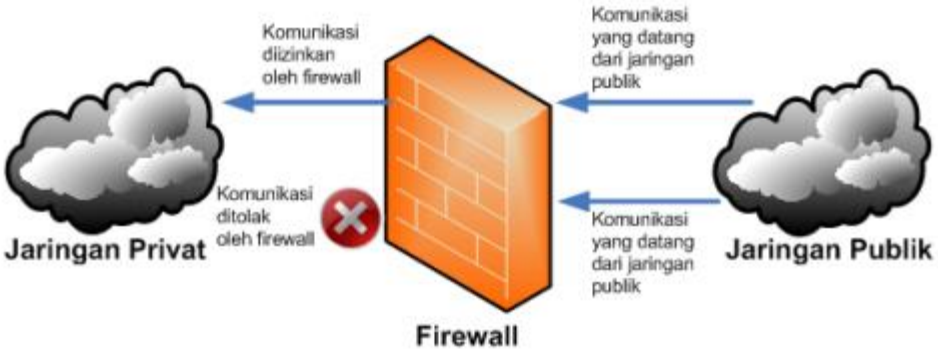
1. *Host address* : Alamat IP yang ditetapkan pada perangkat jaringan seperti komputer atau *router* agar dapat terhubung satu sama lain. Alamat *IP host* bersifat unik dalam jaringan.
2. *Network address* : Alamat IP yang mewakili alamat jaringan. Semua server dalam jaringan memiliki alamat jaringan yang sama. *Network address* adalah IP pertama dalam subnet IP.
3. *Broadcast address* : Jenis *IP address* yang digunakan untuk mengirim data ke semua *host* yang masih berada dalam jaringan yang sama. *Broadcast address* adalah IP terakhir dalam *subnet* IP.

2.2.7. Firewall

Firewall tidak hanya digunakan untuk memblokir akses *client* ke sumber daya tertentu, tetapi juga untuk melindungi jaringan lokal dari ancaman dari luar, seperti virus atau serangan dari *hacker*. Ancaman dari internet ini seringkali datang dari banyak IP yang berbeda, sehingga sulit untuk memberikan perlindungan hanya berdasarkan IP. Selain berbasis *IP Address*, *filtering* juga dapat dilakukan berdasarkan *protocol* dan *port*, sehingga memberikan variasi cara untuk meningkatkan keamanan jaringan (Na & Hipertensiva, n.d.).

Firewall Filter berfungsi sebagai penyaring atau *filter* untuk paket data yang masuk dan keluar dari jaringan, baik itu dari dalam (*local*) maupun dari luar (*internet*). Dengan kata lain, *router* akan menentukan data apa saja yang diizinkan untuk masuk atau keluar. Proses *filtering* ini umumnya melibatkan definisi *IP address*, baik yang berasal (*src-address*) maupun yang dituju (*dst-address*). Misalnya, Anda dapat memblokir komputer klien dengan IP tertentu atau memblokir akses ke suatu situs web berdasarkan IP-nya.

Firewall raw berfungsi untuk memproses paket data pada tahap paling awal sebelum paket tersebut melalui proses *connection tracking* atau NAT. firewall raw digunakan untuk tindakan cepat seperti membuang paket yang tidak diinginkan tanpa mempertimbangkan status koneksi atau membuat modifikasi lebih lanjut.



Gambar 2. 1 *Firewall*
(Sumber : https://aptika.kominfo.go.id/2017/06/keamanan-jaringan-internet-dan-firewall/)

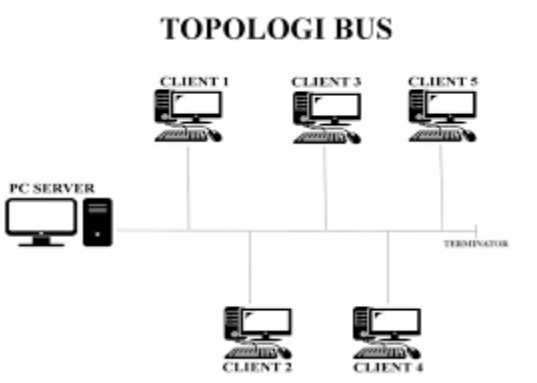
2.2.8. Topologi Jaringan

Topologi Jaringan adalah suatu cara untuk menghubungkan beberapa komputer sehingga tercipta sebuah jaringan komputer. Topologi jaringan memiliki bentuk mulai dari susunan komputer dengan jenis kabel, konektor dan spesifikasi yang berbeda (Anas et al., 2018).

Topologi Jaringan memiliki tiga jenis bentuk yang paling dasar yaitu :

1. Topologi *Bus*

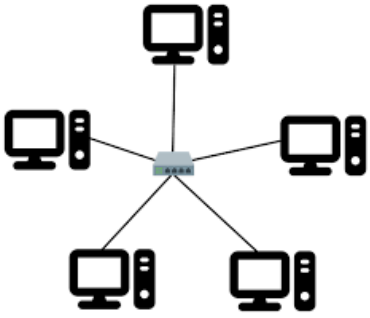
   Topologi *Bus* adalah susunan jaringan yang paling sederhana. Pada topologi ini, terdapat satu kabel utama yang mengarah ke beberapa *node* atau perangkat lain yang terhubung. Kabel ini umumnya menggunakan jenis kabel *coaxial* dengan konektor BNC. Di setiap titik sambungan antara kabel utama dan *node*, digunakan *T-Connector*, sementara ujung kabel utama yang tidak terhubung pada perangkat jaringan diberi *terminator* atau *end-connector*.



gambar 2. 2 Topologi *Bus*
(Sumber : https://itbox.id/blog/topologi-bus/)

2. Topologi *Star*

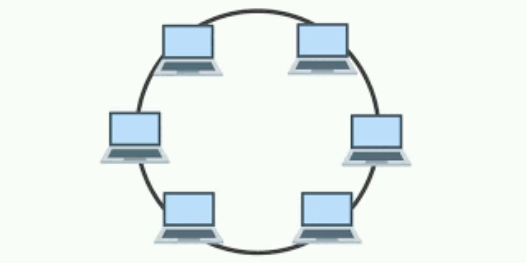
Topologi *Star* memiliki bentuk seperti bintang. Pusat dari topologi ini adalah hub/*switch* yang terletak di tengah, dan berfungsi sebagai pusat kendali. Semua perangkat jaringan terhubung langsung ke hub/*switch*, menjadikannya pusat vital dalam topologi ini. Topologi *Star* di kenal karena kemudahan perawatannya, dan menggunakan kabel UTP beserta *connector* RJ-45.



gambar 2. 3 *Topologi Star*
(Sumber : https://www.arduinoindonesia.id/2023/05/penjelasan-tentang-topologi-star.html)

3. Topologi *Ring*

Topologi *Ring* membentuk struktur lingkaran, di mana setiap perangkat terhubung langsung dengan dua perangkat lainnya, sehingga setiap *node* memiliki dua kabel. Topologi ini menggunakan kabel *coaxial* dengan konektor BNC. Berbeda dengan topologi *Bus*, *Ring* tidak memerlukan *end-connector* karena semua kabel terhubung langsung dengan perangkat jaringan.



Gambar 2.4 : Topologi *Ring*
(Sumber : https://course-net.com/blog/topologi-ring-adalah/)

2.2.9. Flowchart

Flowchart yang sering disebut sebagai bagan alir, adalah suatu bentuk representasi visual yang menggambarkan algoritma atau langkah-langkah instruksi dalam suatu sistem secara berurutan. Misalnya menggunakan diagram alir sebagai bentuk dokumentasi untuk menjelaskan gambaran logis suatu sistem. Dengan demikian, *flowchart* berfungsi sebagai bukti dokumentasi yang membantu memberikan solusi terhadap potensi masalah yang mungkin timbul dalam pengembangan sistem. *Flowchart* disusun dengan menggunakan simbol-simbol yang mewakili berbagai proses. Setiap simbol merepresentasikan suatu proses khusus untuk menghubungkan satu proses dengan proses berikutnya, digunakan garis penghubung sebagai elemen penghubung (Rosaly & Prasetyo, 2019).

2.2.10. Putty

Putty adalah sebuah *client* dari SSH dan Telnet yang dulunya dikembangkan oleh Simon Tatham untuk *platform* windows yang hanya bersifat *open source* yang memiliki tujuan untuk melakukan sebuah *protocol* jaringan SSH, Telnet dan *login Protocol* dapat digunakan ketika menjalankan sebuah sesi *remote* pada sebuah komputer melalui sebuah jaringan dengan jarak jauh (Ernawati et al., 2022).

2.2.11. DDOS attack

DDOS *attack* atau *Distributed Denial of Service* adalah sebuah jenis serangan yang mempengaruhi korban dengan sebuah tujuan menemukan kelemahan korban. DDOS *attack* dikenal serangan yang dirancang dapat melemahkan suatu layanan server. DDOS *attack* memiliki target utama pada sumber daya *bandwith*, CPU, dan sumber daya terbatas dalam jaringan (Ernawati et al., 2022). DDOS attack memiliki beberapa jenis serangan, termasuk SYN *flood* dan HTTP *flood.*

SYN *(Synchronize) flood* merupakan serangan dengan memanfaatkan paket SYN untuk membanjiri target dengan permintaan koneksi palsu, sehingga menghabiskan sumber daya *server* dan mengganggu layanan. Sedangkan HTTP (*Hypertext Transfer Protocol*) *flood* adalah serangan yang menargetkan lapisan aplikasi (*layer* 7) dari model OSI. Tujuan dari HTTP *flood* adalah membanjiri *server* dengan lalu lintas permintaan yang valid, sehingga menguras daya *server* seperti CPU, RAM, dan *bandwith* dan akhirnya menyebabkan *server* menjadi lambat atau tidak responsif.

2.2.12. Port Scanning

Port scanning adalah sebuah jenis serangan yang bertujuan untuk mengetahui aktif atau sebuah *host* target pada sebuah jaringan. Hasil dari *scanning* berupa *IP address*, sistem operasi, *service* dan juga aplikasi yang dijalankan. Informasi yang didapatkan dari serangan *port scanning* yang berguna untuk menentukan sebuah metode yang akan digunakan dalam melakukan penyerangan sistem yang akan dilakukan (Ernawati et al., 2022).

BAB III
PERANCANGAN

3.1. Bahan dan Alat Penelitian

Pengumpulan data pada penelitian ini untuk memperoleh informasi yang dibutuhkan dalam mencapai tujuan penelitian.

3.1.1 Bahan Penelitian

Pada Implementasi *Port Knocking* pada Laboratorium *High Performance Computing* (HPC), bahan penelitian yang dibutuhkan adalah :

a) Analisa

Analisa digunakan untuk menganalisa rancangan *port knocking mikrotik* yang akan dibangun pada pembuatan suatu desain keamanan jaringan, tahap pertama rancang bangun desain jaringan, hingga tahap pengujian keamanan jaringan *port knocking* tersebut untuk mengetahui apakah hasil dari rancangan yang di implementasikan pada *router mikrotik* untuk mendapatkan hasil yang baik.

b) Perancangan

Perancangan yaitu menerapkan dari tahap “analisa” kedalam bentuk desain jaringan untuk di implementasikan ke dalam sistem keamanan jaringan komputer.

c) Pengujian

Pengujian yang dilakukan pada *router mikrotik* untuk menunjukkan *port knocking* pada desain keamanan jaringan yang akan di terapkan bekerja dengan baik.

d) Dokumentasi

Proses dokumentasi dilakukan pada tinjauan pustaka, membaca dan mempelajari buku- buku, serta mencari sumber –sumber yang berkaitan dengan penelitian sebagai bahan referensi.

3.1.2 Alat Penelitian

1. Perangkat Keras

Alat yang dibutuhkan dalam proses penelitian ini adalah sebagai berikut :

a) Laptop

LAPTOP HP ENVY 13 –aq1xxx dengan spesifikasi *processor Intel* ® Core™ i5-10210U CPU @ 1.6GHz (8CPUs), ~2.1GHz dan memiliki (RAM) sebesar 8192MB.

b) Kabel UTP (*Unshileded Twisted Pair*)

Kabel UTP merupakan kabel konektor *ethernet* yang memiliki fungsi sebagai konektor topologi jaringan komputer. RJ adalah singkatan dari *Registered Jack* yang merupakan standar kepala konektor dan urutan kabel yang menghubungkan dua atau lebih peralatan komunikasi.

c) Hub

Hub atau *Network Hub* berfungsi untuk menghubungkan komputer satu ke komputer lainnya yang masih dalam satu lingkup jaringan juga dapat berbagi informasi seperti dokumen dan *file* maupun data lainnya.

d) Router Wireless

Router Wireless berfungsi sebagai *router* termasuk fungsi dari *wireless access point* yang digunakan untuk mengakses jaringan internet kemudian berfungsi sebagai jaringan LAN dan *wireless* LAN.

2. Perangkat Lunak

   Perangkat lunak/*software* yang digunakan dalam proses penelitian ini sebagai berikut :

   a. Windows

      Windows sebagai sistem operasi yang dipakai oleh komputer *client*, *operator*.

   b. Kali Linux

      Kali Linux merupakan sistem operasi yang dipakai oleh komputer penyerang (*attacker*).

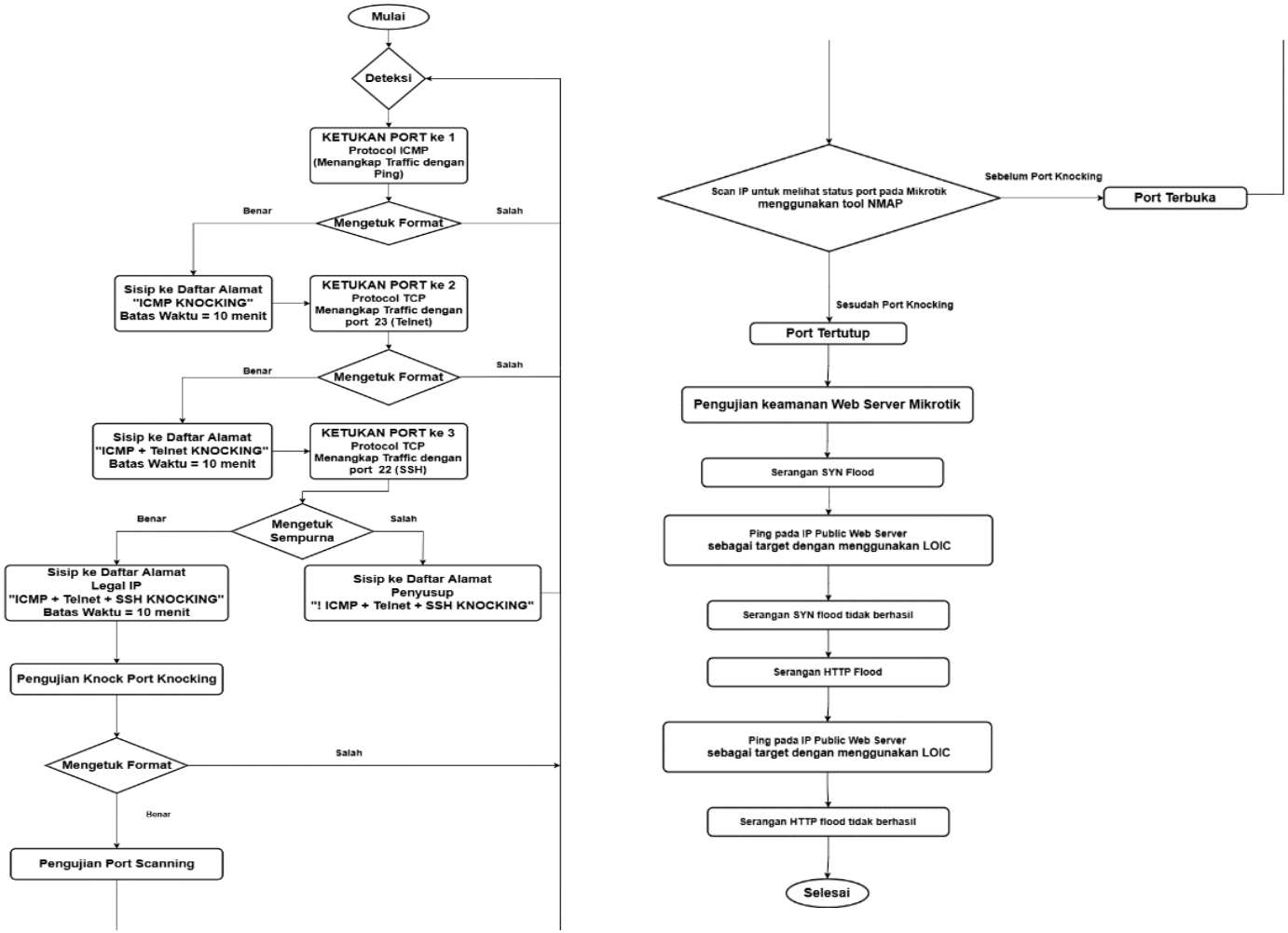
   c. Winbox

      Winbox merupakan *software* atau *unity* yang digunakan untuk melakukan *remote* sebuah *server mikrotik* ke dalam mode GUI (*Graphical User Interface*) melalui *operating system windows*.

3.2. Perancangan

3.2.1 *Flowchart*

Perancangan *flowchart* untuk menjelaskan alur proses *port knocking* pada *mikrotik* yang akan diterapkan pada laboratorium HPC :



Gambar 3. 1 *Flowchart*
(Sumber : Data Olahan, 2024)

1 Pada tahap awal akan dilakukan deteksi untuk masuk ke *mikrotik* melalui aplikasi *winbox* jika berhasil *login* maka akan dilakukan ketukan pertama menggunakan *protocol* ICMP dengan *rule* yang menangkap *traffic* ICMP (*ping*) yang masuk *router mikrotik*, kemudian secara otomatis ke *address-list* dengan nama ICMP KNOCKING selama 10 menit.

2 Kemudian akan dilakukan *rule* yang kedua yaitu untuk menangkap *traffic* kedalam *router mikrotik* dengan *protocol* TCP dan *destination port* 23 (Telnet) yang berasal dari *address-list* ICMP *Knocking*, IP yang berasal dari *address list* ICMP knocking tadi akan dimasukkan kembali ke dalam sebuah *address-list* baru yang bernama ICMP + Telnet *KNOCKING* selama 10 menit.

3 Setelah itu akan dilakukan *rule* yang ketiga yaitu penangkapan *traffic* kedalam *router mikrotik* dengan *protocol* TCP dan *destination port* 22 (SSH) yang berasal dari *address-list* ICMP + Telnet *KNOCKING*, IP yang berasal dari *address list* ICMP + Telnet *KNOCKING* akan dimasukan kedalam *address-list* yang baru bernama ICMP + Telnet + SSH *KNOCKING* selama 10 menit.

4 Tahap selanjutnya adalah *rule* yang terakhir yang berfungsi untuk melakukan *droping packet* yang masuk ke dalam *router miktotik* dengan tujuan port 8291 (*winbox*), 23 (*Telnet*), 22 (SSH) kecuali *IP user* yang sudah melakukan *ping* dan *telnet* juga SSH sebelumnya (*Src.Address List* = ! ICMP + *Telnet* + *SSH KNOCKING*).

5 Tahap pengujian untuk *knock port knocking* yaitu melihat berhasil tidaknya implementasi *port knocking* pada *mikrotik.* Pengujian ini dilakukan dengan 2 tahap yaitu tahap sebelum dan sesudah di implementasikan metode *port knocking*.

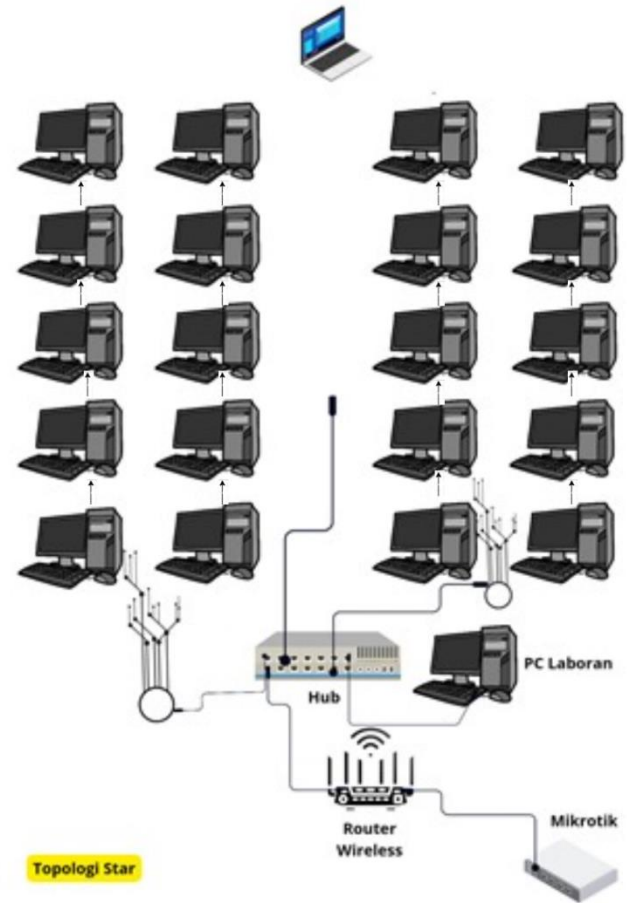
6 Tahap pengujian *Port Scanning* yaitu untuk melihat informasi dari pada *mikrotik* seperti celah *port* tujuan yang terbuka dan tertutup. Pada tahap pengujian ini menggunakan tool *NMAP* (*Network Mapper*) pada *mikrotik* dengan men-*scan* IP untuk melihat status *port*. Pengujian ini dilakukan dengan

2 tahap yaitu tahap sebelum dan sesudah di implementasikan metode *port knocking*.

7 Tahap Pengujian yang terakhir adalah DDOS *attack* yaitu untuk membuktikan keamanan *web server* dalam *mikrotik* setelah di implementasikan metode *firewall raw*. Pengujian ini dilakukan dengan cara ping pada IP dari *ISP mikrotik* pada *web server* sebagai target menggunakan SYN *flood* dan HTTP *flood* yang ada pada aplikasi LOIC.

3.2.2 Topologi Jaringan yang berjalan pada Laboratorium HPC

Menganalisa topologi jaringan yang berjalan pada Laboratorium HPC:



Gambar 3. 2 Topologi berjalan
(Sumber : Laboratorium HPC, 2024)

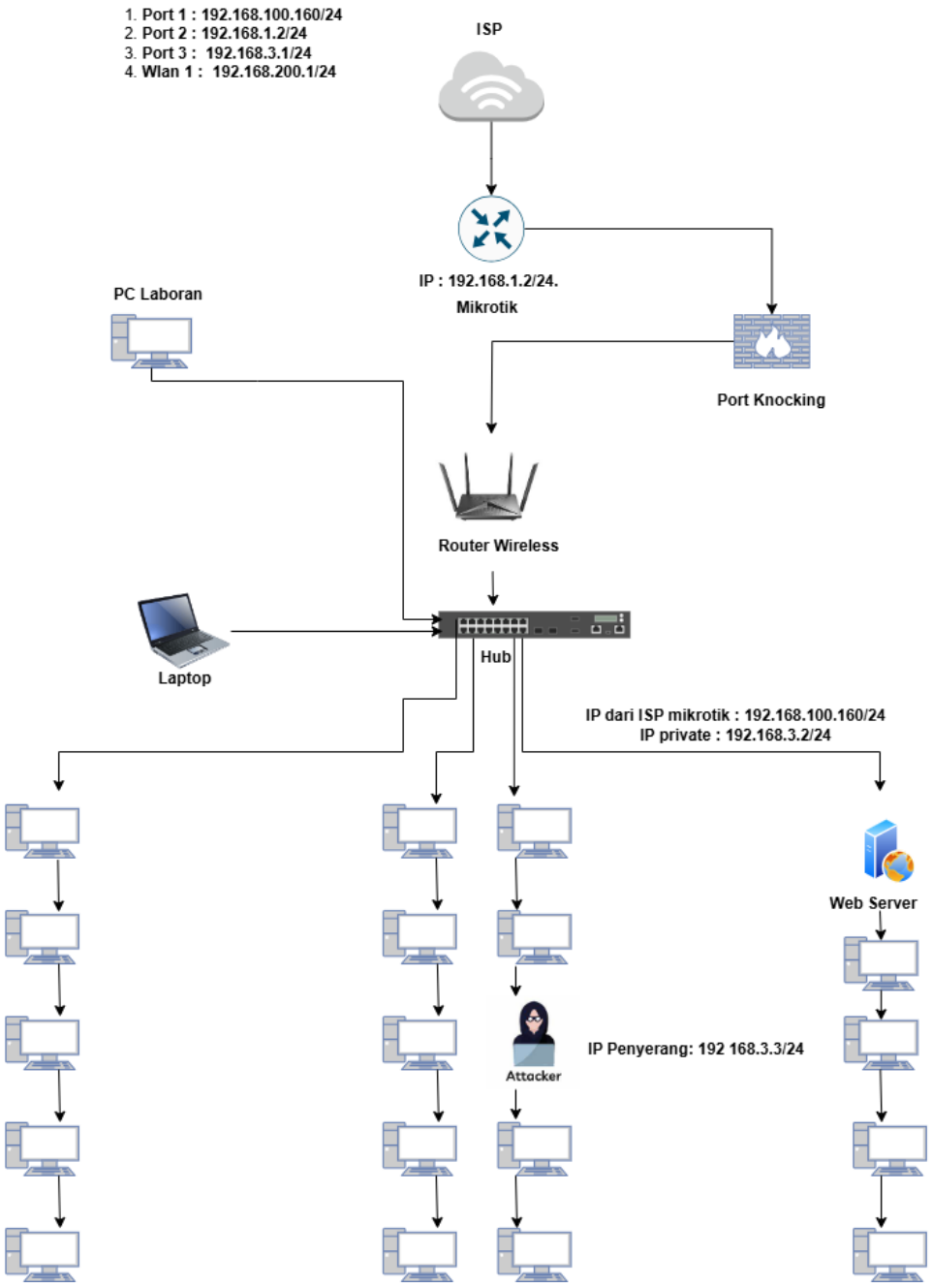
1 *Mikrotik* dan *router wireless* yang berfungsi mengatur lalu lintas data antar jaringan lokal dan jaringan luar dan berguna untuk mengelola akses ke internet dan distribusi alamat IP kepada perangkat dalam jaringan kemudian data akan diteruskan pada hub.

2 Hub akan menggabungkan atau menyambungkan beberapa perangkat untuk memberikan titik pusat koneksi pada PC Laboran, Laptop maupun PC lainnya dalam satu jaringan yang memiliki kegunaan untuk memudahkan pertukaran data diantara perangkat yang terhubung ke hub.

3 Topologi ini cukup fleksibel untuk lingkungan laboratorium HPC namun memiliki kekurangan yaitu ketersediaan dan kinerja pusat titik koneksi pada hub dan *mikrotik* maka jika pusat rusak maka seluruh jaringan dapat terpengaruh.

3.2.3 Topologi Jaringan yang diusulkan

Menganalisa topologi jaringan yang di usulkan :



Gambar 3. 3 Topologi diusulkan
(Sumber : Data Olahan, 2024)

1 ISP (*Internet Service Provider*) berfungsi untuk menyediakan koneksi internet untuk jaringan dan menyediakan jalur komunikasi ke luar jaringan lokal.

2 *Mikrotik* dan *router wireless* yang yang berfungsi mengatur lalu lintas data antar jaringan lokal dan jaringan luar dan berguna untuk mengelola akses ke internet dan distribusi alamat IP kepada perangkat dalam jaringan kemudian data akan diteruskan pada hub.

3 *Port knocking* berguna untuk memberikan lapisan tambahan keamanan dengan mengharuskan pengguna untuk melakukan serangkaian koneksi yang benar sebelum diberikan akses ke jaringan dengan melibatkan membuka akses ke *port* tertentu setelah rangkaian permintaan khusus diterima.

4 Hub akan menggabungkan atau menyambungkan beberapa perangkat untuk memberikan titik pusat koneksi pada PC Laboran, Laptop maupun PC lainnya dalam satu jaringan yang memiliki kegunaan untuk memudahkan pertukaran data diantara perangkat yang terhubung ke hub.

5 *Mikrotik* berfungsi sebagai uji coba penerapan implementasi *port knocking* pada Laboratorium HPC dimana *kali linux* sebagai penyerang dengan mencoba jenis serangan *port scanning* Hasil yang signifikan yaitu sebelum dan sesudah penerapan *port knocking* pada *mikrotik* dapat mengatasi serangan *port scanning*. Kemudian uji coba penerapan *firewall raw* untuk menguji keamanan *web server* dalam *mikrotik* dengan penyerangan DDOS *attack* dimana terdapat 2 jenis serangan yang akan di uji coba yaitu SYN *flood* dan HTTP *flood* menggunakan *LOIC*. Hasil yang signifikan setelah penerapan *firewall raw* dapat mengatasi *web server* dari serangan DDOS *attack.*

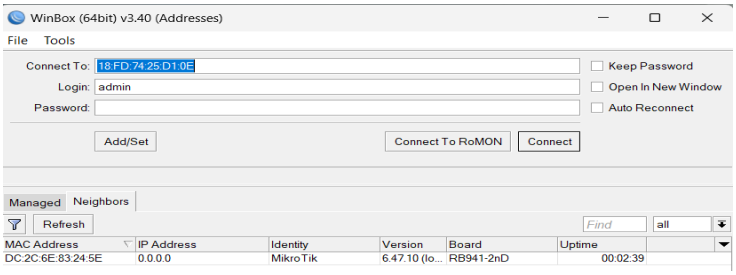
BAB IV
HASIL DAN PENGUJIAN

4.1 Hasil

4.1.1. Konfigurasi *mikrotik*

4.1.1.1. *login mikrotik* menggunakan *winbox*

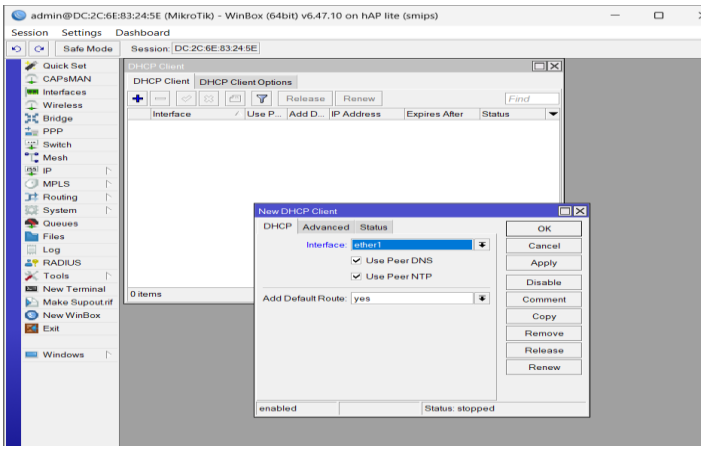
Tahap ini menampilkan *login mikrotik* menggunakan *winbox* dapat dilihat dari gambar 4.1.



Gambar 4. 1 *Login mikrotik* menggunakan *winbox*
(Sumber : Data Olahan, 2024)

4.1.1.2. Konfigurasi *ether1*

Tahap Konfigurasi *ether1* sebagai *DHCP Client* untuk mendapatkan alamat *IP address* dari ISP dapat dilihat pada gambar 4.2 dibawah ini.



Gambar 4. 2 Konfigurasi *ether1*
(Sumber : Data Olahan, 2024)

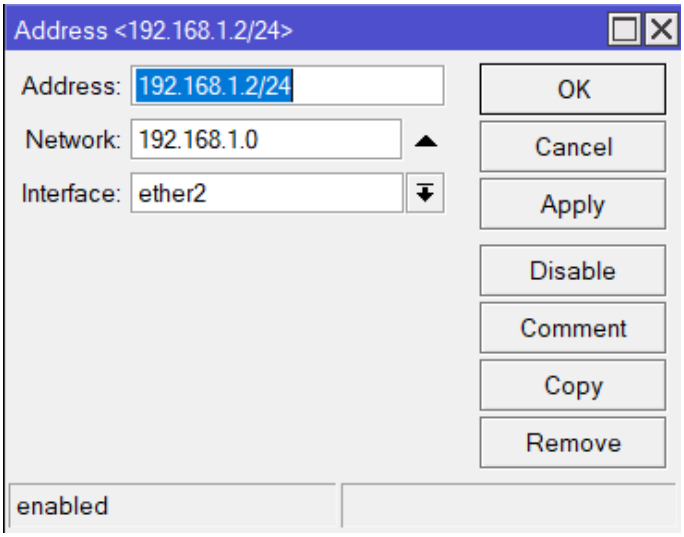
*4.1.1.3.*IP *ether1*

Maka tahap *ether1* akan mendapat alamat *IP address list* secara otomatis.

Gambar 4. 3 *IP ether1*
(Sumber : Data Olahan, 2024)

4.1.1.4.Konfigurasi *ether2*

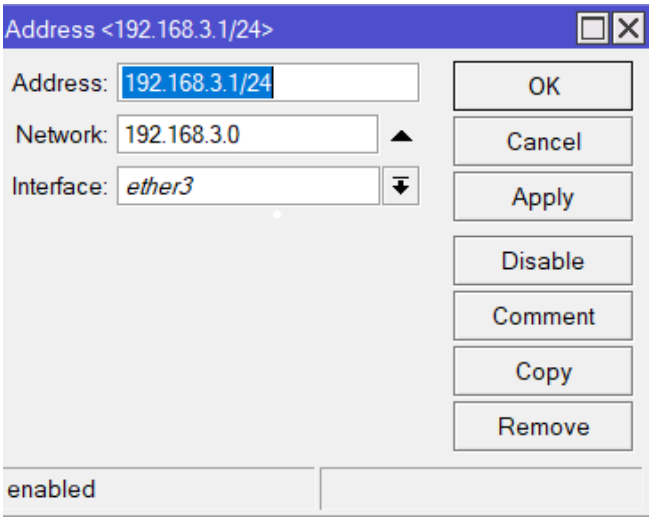
Tahap konfigurasi pada *ether2* dengan alamat *address* 192.168.1.2/24.



Gambar 4. 4 Konfigurasi *ether2*
(Sumber : Data Olahan, 2024)

4.1.1.5.Konfigurasi *ether3*

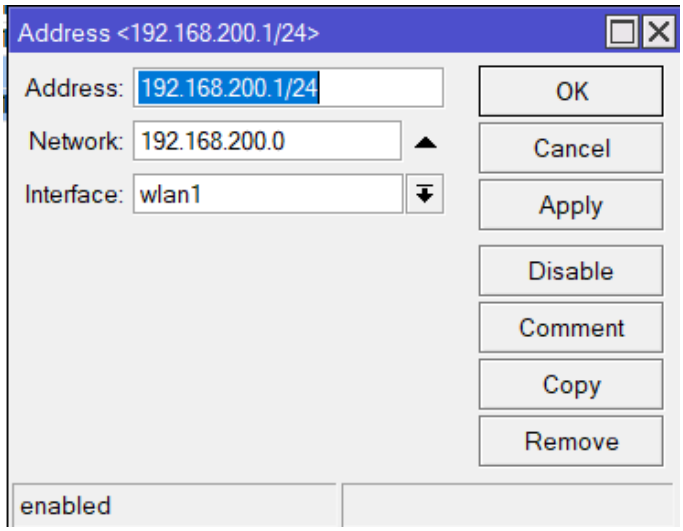
Tahap selanjutnya konfigurasi alamat IP *address* 192.168.3.1/24.



Gambar 4. 5 Konfigurasi *ether3*
(Sumber : Data Olahan, 2024)

4.1.1.6.Konfigurasi pada *wlan1*

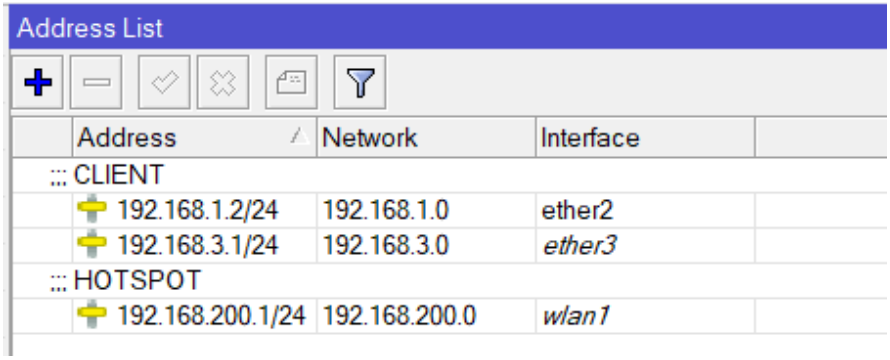
Tahap konfigurasi pada *wlan1* dengan alamat *address* 192.168.200.1/24.



Gambar 4. 6 Konfigurasi *wlan1*
(Sumber : Data Olahan, 2024)

*4.1.1.7.*Halaman *address list*

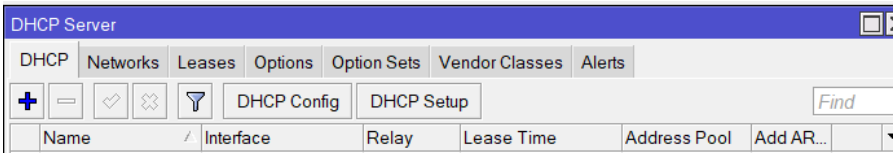
Setelah berhasil dikonfigurasi maka alamat *IP address* akan tampil pada halaman *address list.*



Gambar 4. 7 Halaman *address list*
(Sumber : Data Olahan, 2024)

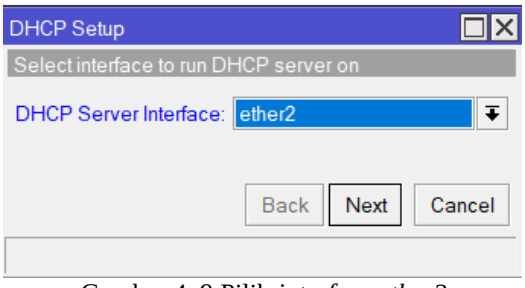
4.1.1.8.Halaman *DHCP server*

Tahap *setting* pada menu IP-*DHCP Server*- pilih menu *DHCP setup* untuk *ether2* dan *wlan1*, dan *ether3.*



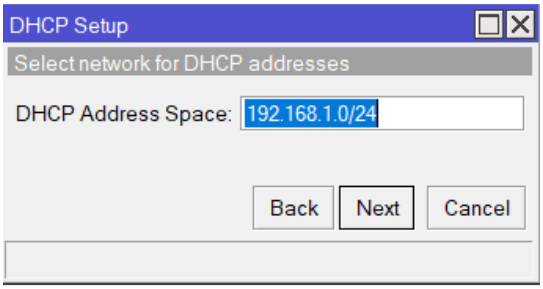
Gambar 4. 8 Halaman *DHCP server*
(Sumber : Data Olahan, 2024)

- *Interface* pada *Ether2-Next.*



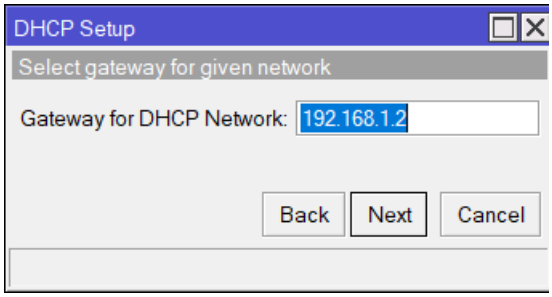
Gambar 4. 9 Pilih *interface ether2*
(Sumber : Data Olahan, 2024)

- *DHCP address space*, tahap ini akan terisi otamatis- *Next.*



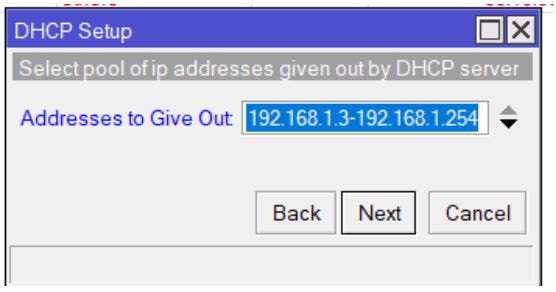
Gambar 4. 10 *DHCP address space ether2*
(Sumber : Data Olahan, 2024)

- *Gateway*, tahap ini akan terisi otomatis oleh *Ip address* dari *ether2* yaitu 192.168.1.2-*Next.*



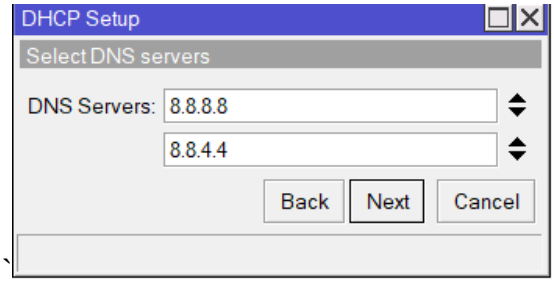
Gambar 4. 11 *Gateway ether2*
(Sumber : Data Olahan, 2024)

- *IP pool* yang akan digunakan oleh *client*, dan akan terisi otomatis sesuai *hosts* pada *prefix* yang akan digunakan.



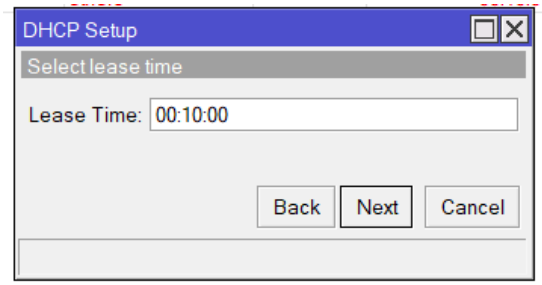
Gambar 4. 12 *IP pool ether2*
(Sumber : Data Olahan, 2024)

- DNS : 8.8.8.8 dan 8.8.4.4 yang akan digunakan otomatis pada semua *client* yang tersambung pada *ether2.*



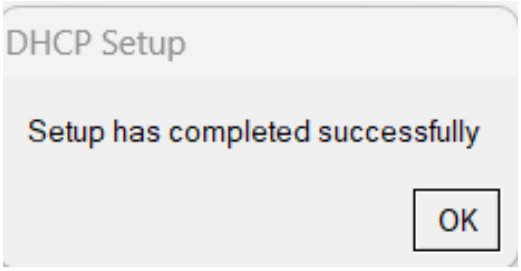
Gambar 4. 13 *DNS Ether2*
(Sumber : Data Olahan, 2024)

- *Lease Time-Next*, yaitu berapa lama *ip address* akan dipinjamkan oleh *client.*



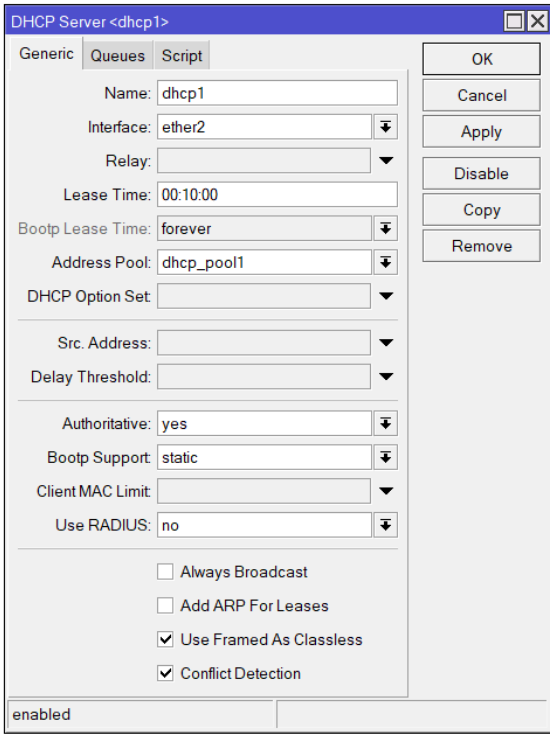
*G*ambar 4. 14 *Lease Time Ether2*
(Sumber : Data Olahan, 2024)

- Konfigurasi *DHCP Server* pada *ether2* berhasil.



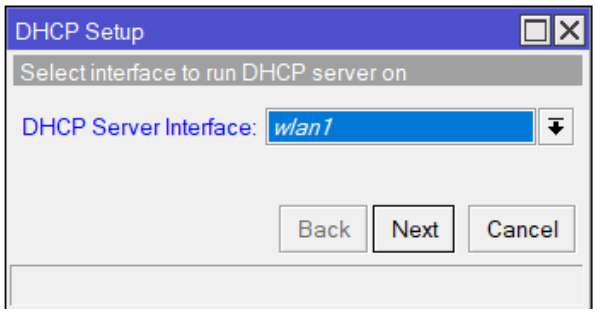
Gambar 4. 15 *DHCP Server ether2* berhasil
(Sumber : Data Olahan, 2024)

- Sampai tahap ini *client* akan mendapatkan akses internet dan *IP address* otomatis 192.168.1.2 sampai 192.168.1.254.



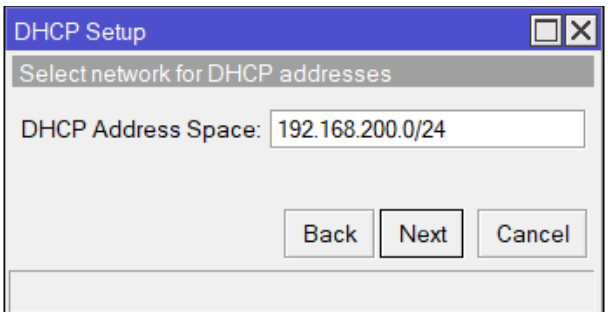
Gambar 4. 16 *DHCP Server ether2*
(Sumber : Data Olahan, 2024)

- *Interface* pada *wlan1-Next.*



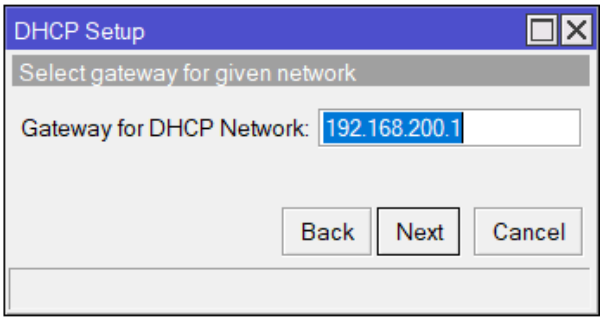
Gambar 4. 17 Pilih *interface wlan1*
(Sumber : Data Olahan, 2024)

- *DHCP address space*, tahan ini akan terisi otamatis- *Next.*



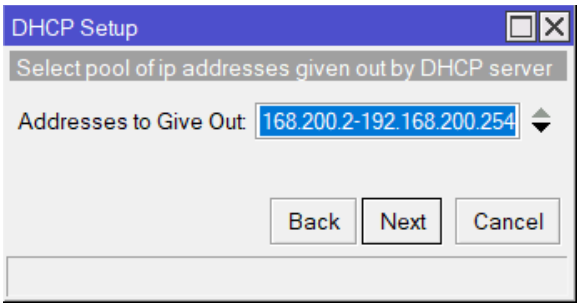
Gambar 4. 18 *DHCP address space wlan1*
(Sumber : Data Olahan, 2024)

- *Gateway*, tahap ini akan terisi otomatis oleh *Ip address* dari *wlan1* yaitu 192.168.200.1-*Next.*



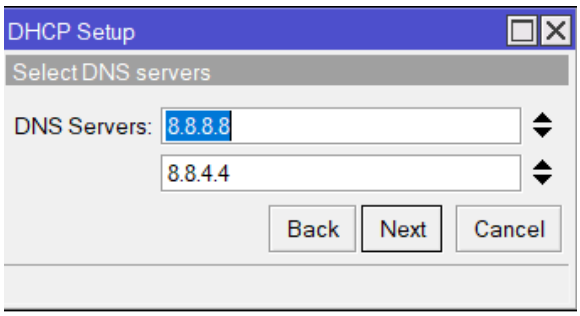
Gambar *4. 19 Gateway wlan1*
(Sumber : Data Olahan, 2024)

- Pada Tahap *IP pool* yang akan digunakan oleh *client,* dan akan terisi otomatis sesuai *hosts* pada *prefix* yang akan digunakan.



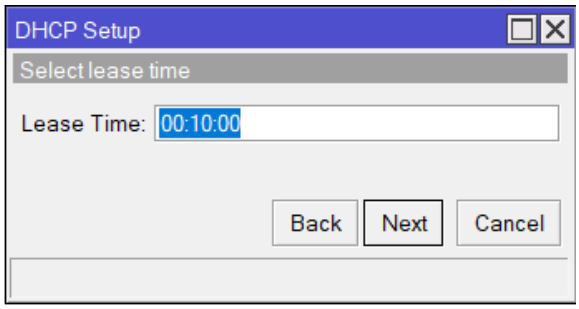
Gambar 4. 20 *IP Pool wlan1*
(Sumber : Data Olahan, 2024)

- DNS : 8.8.8.8 dan 8.8.4.4 yang akan digunakan otomatis pada semua *client* yang tersambung pada *wlan1*.



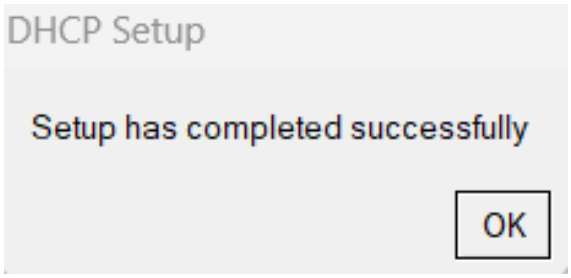
Gambar 4. 21 *DNS wlan1*
(Sumber : Data Olahan, 2024)

- *Lease Time-Next*, yaitu berapa lama *ip address* akan dipinjamkan oleh *client.*



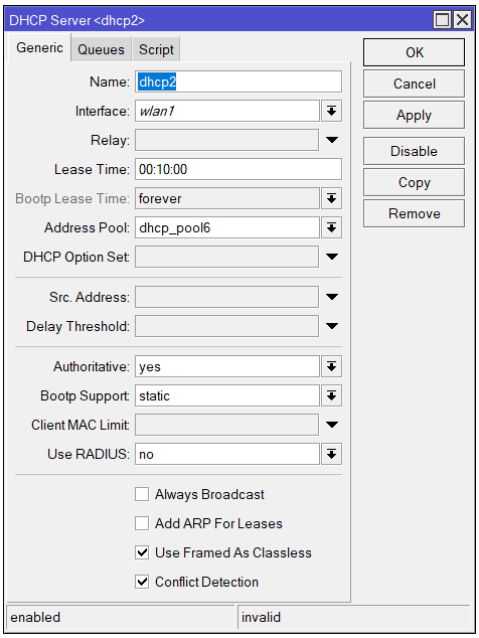
Gambar 4. 22 *Lease Time wlan1*
(Sumber : Data Olahan, 2024)

- Konfigurasi *DHCP Server* pada *wlan1* berhasil.



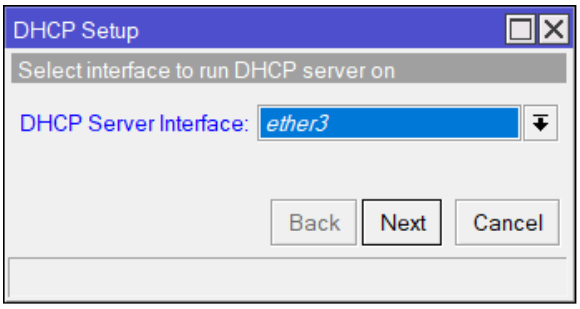
Gambar 4. 23 *DHCP Server wlan1* berhasil
(Sumber : Data Olahan, 2024)

- Sampai tahap ini *client* akan mendapatkan akses internet dan *IP address* otomatis 192.168.200.1 sampai 192.168.200.254.



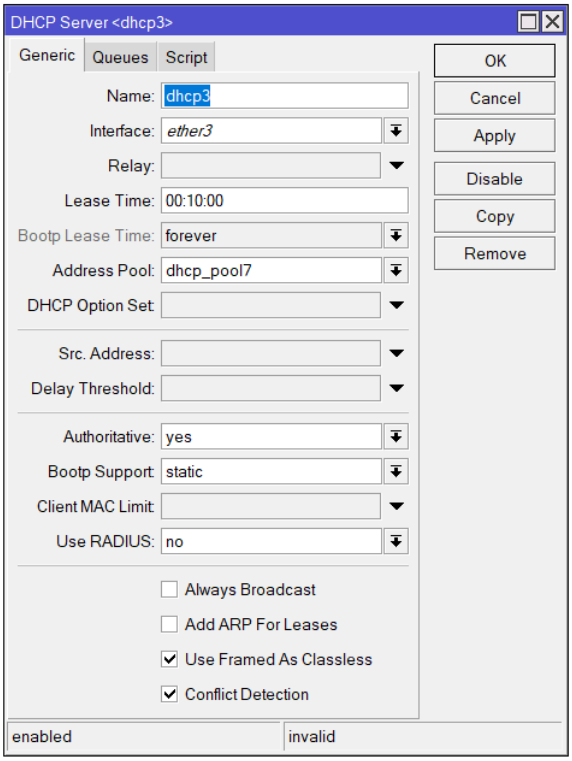
Gambar 4. 24 *DHCP server wlan1*
(Sumber : Data Olahan, 2024)

- Tahap *Interface* pada *Ether3-Next* pada halaman *DHCP Setup.*



Gambar 4. 25 Pilih *interface ether3*
(Sumber : Data Olahan, 2024)

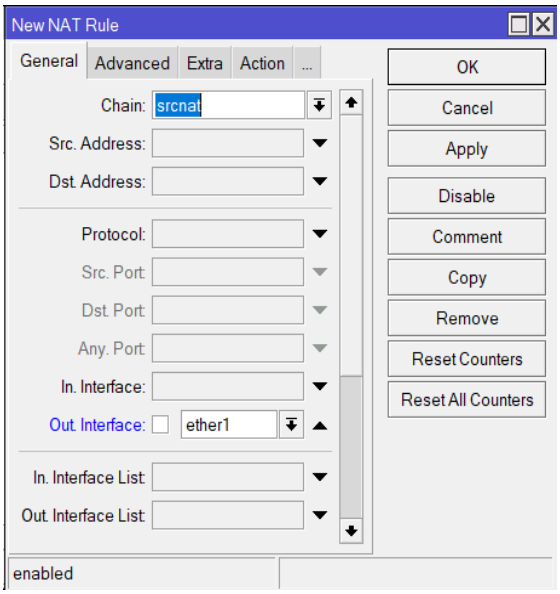
- Tahap *DHCP server* konfigurasi pada *ether3*.



Gambar 4. 26 *DHCP server ether3*
(Sumber : Data Olahan, 2024)

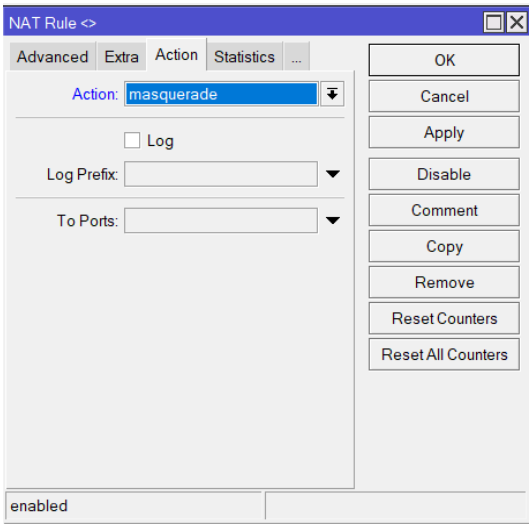
4.1.1.9.*Setting* NAT pada *ether1*

Setelah itu, Tahap *setting* NAT agar *IP client* bisa terkoneksi ke internet.



Gambar 4. 27 *Setting NAT ether1*
(Sumber : Data Olahan, 2024)

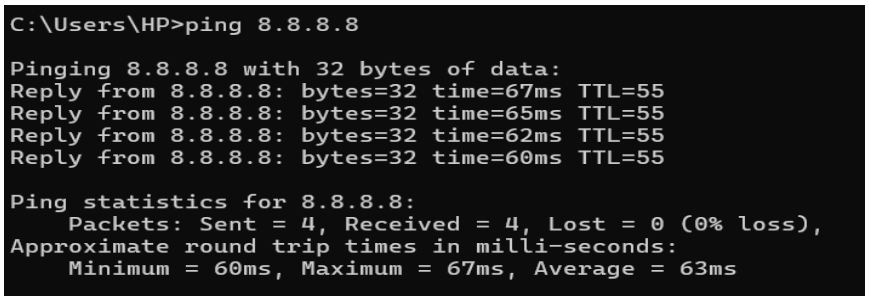
- Selanjutnya tahap pada *tab action*.



Gambar 4. 28 *Tab action Ether1*
(Sumber : Data Olahan, 2024)

4.1.1.10. *Ping* koneksi internet pada *Client*

Kemudian pada tahap ini adalah melakukan pengecekan apakah koneksi internet berjalan dengan lancar pada *client* dengan melakukan perintah *ping* 8.8.8.8 pada CMD.

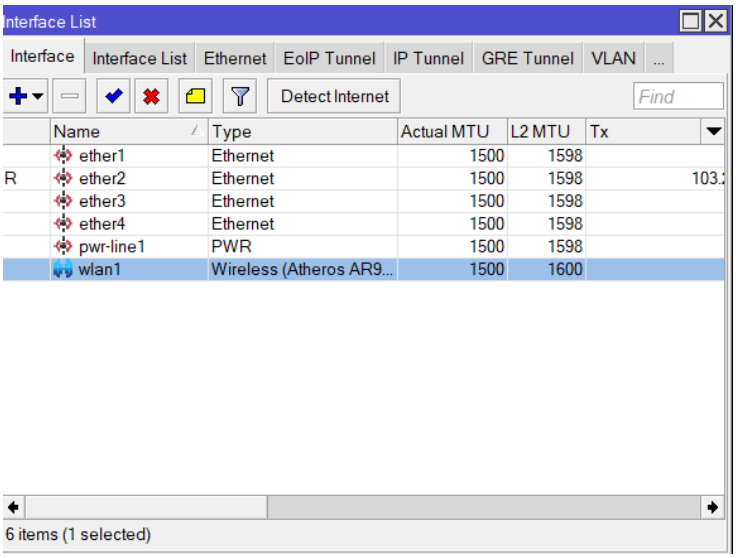


Gambar 4. 29 *ping 8.8.8.8*
(Sumber : Data Olahan, 2024)

4.1.2. Manajemen *hotspot user*

4.1.2.1 Manajemen *hotspot* pada *wlan1*

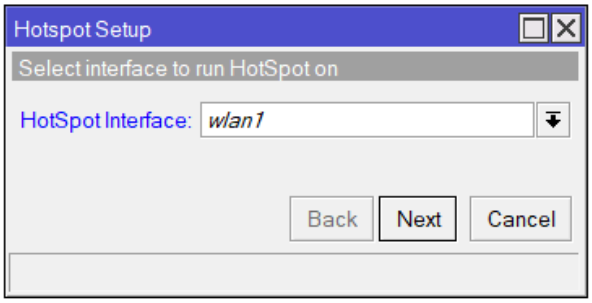
Pada *interface wlan1* akan digunakan untuk *hotspot* dan aktifkan *interface wlan1* nya dan setting menggunakan *mode Apbridge*, pilih frekuensi dan beri nama SSID Wifinya.



Gambar 4. 30 Aktifkan *wlan1* pada *interface*
(Sumber : Data Olahan, 2024)

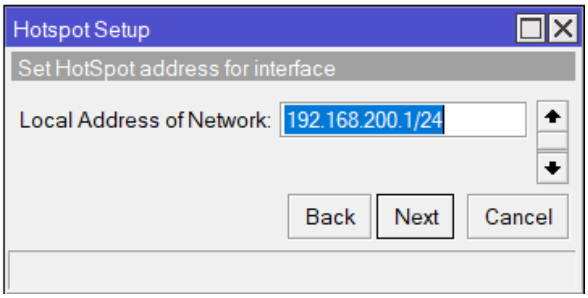
4.1.2.2 Setting Hotspot Setup

Tahap *setting Hotspot Mikrotik* dapat dilihat pada tahap ini menggunakan *interface wlan1*.



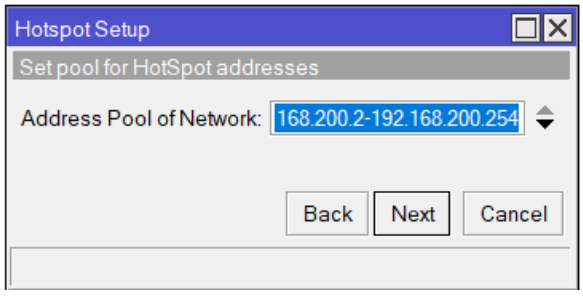
Gambar 4. 31 *Setting Hotspot Mikrotik*
(Sumber : Data Olahan, 2024)

- Tahap *IP Address* dapat dilihat pada gambar 4.32.



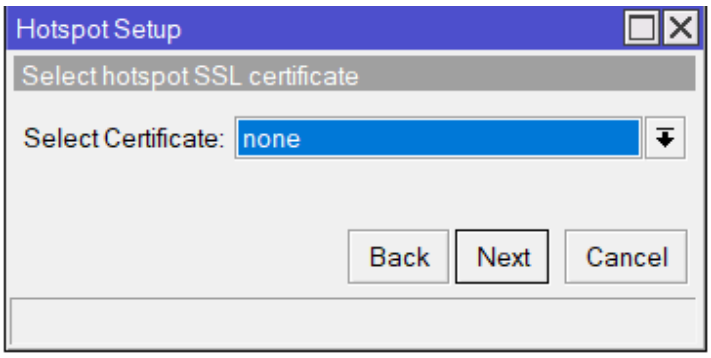
Gambar 4. 32 Isikan *IP Address*
(Sumber : Data Olahan, 2024)

- Tahap *range IP Address* yang akan diberikan ke *client hotspot* pada *DHCP Server*. Opsi ini sudah terisi secara otomatis, namun jika ingin mengganti *IP Address* nya silakan saja disesuaikan dengan kebutuhan.



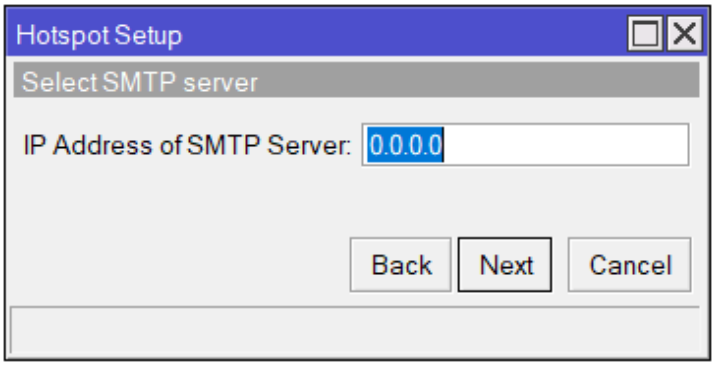
Gambar 4. 33 Tentukan *range IP Address*
(Sumber : Data Olahan)

- Tahap pilih *SSL Certificate* yang akan digunakan.



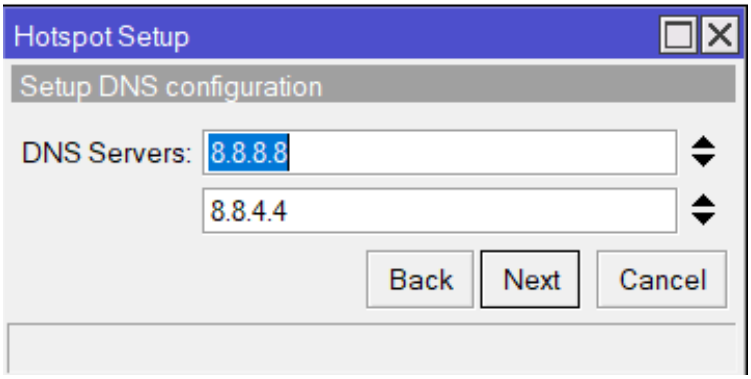
Gambar 4. 34 Pilih *SSL Certificate*
(Sumber : Data Olahan)

- Tahap Memasukkan *IP Address* untuk *SMTP Server*.



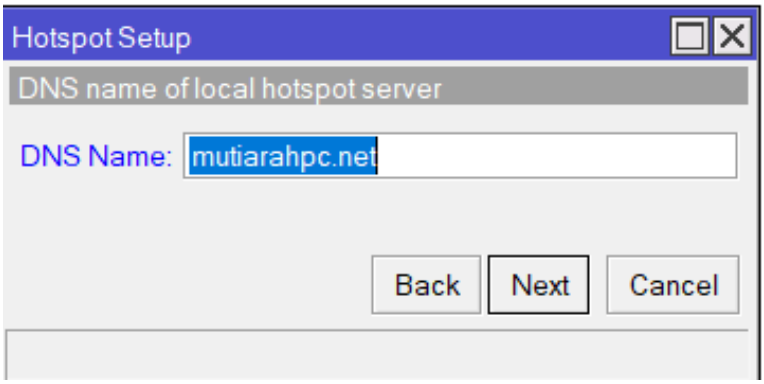
Gambar 4. 35 Memasukkan *IP Address*
(Sumber : Data Olahan)

- Tahap memasukkan *DNS Server*, dengan *DNS Google* 8.8.8.8.



Gambar 4. 36 Memasukkan *DNS Server*
(Sumber : Data Olahan, 2024)

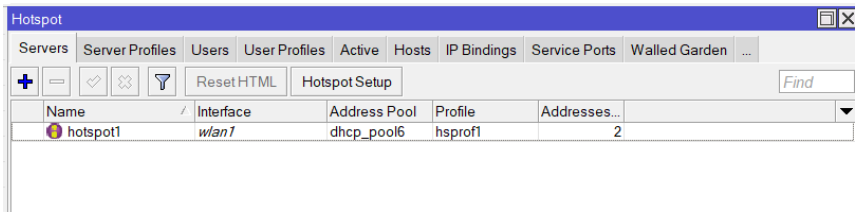
- Tahap memasukkan *DNS Name* untuk menggunakan nama domain pada *Hotspot Server Mikrotik* nya dengan nama mutiarahpc.net->*Next.*



Gambar 4. 37 Memasukkan *DNS Name*
(Sumber : Data Olahan, 2024)

4.1.2.3 *Hotspot Mikrotik* berhasil

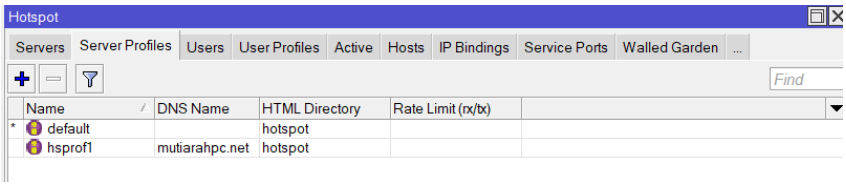
Pada Tahap ini *Hotspot Mikrotik* sudah berhasil dibuat.



Gambar 4. 38 *Hotspot Mikrotik* berhasil
(Sumber : Data Olahan, 2024)

4.1.2.4 *Tab server profiles*

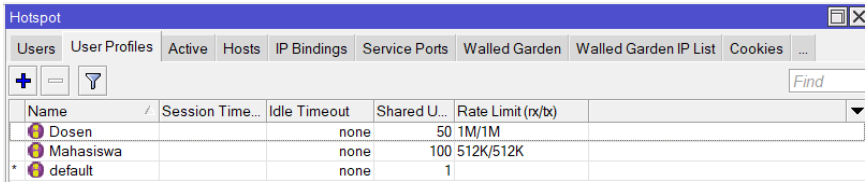
Dapat dilihat pada *tab server profiles* terdapat *DNS Name* yang sudah diatur sebelumnya.



Gambar 4. 39 *Tab server profiles*
(Sumber : Data Olahan, 2024)

4.1.2.5 *Tab user profiles*

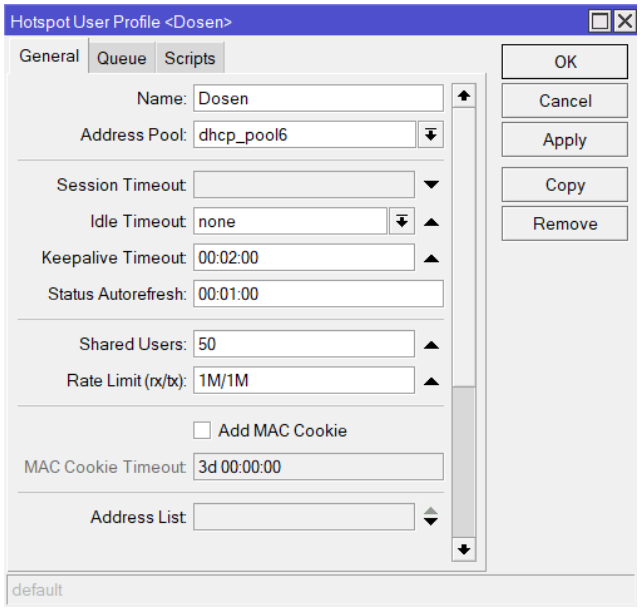
Pada tab *user profiles* tambahkan *2 user* pengguna *hotspot* yaitu Dosen dan Mahasiswa dengan cara *klik* tambah.



Gambar 4. 40 *Tab user profiles*
(Sumber : Data Olahan, 2024)

4.1.2.6 Profile Dosen

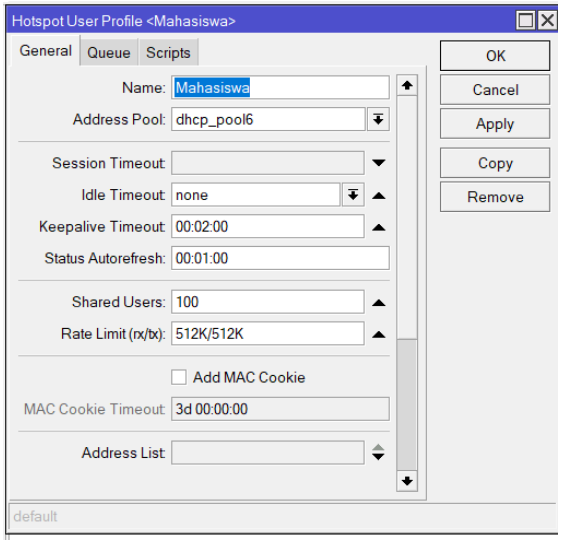
Pada tahap ini masukan nama yang diinginkan,Membuat *profile* untuk Dosen dengan pengaturan *Rate Limit* (rx/tx) 1 Mbps yang artinya setiap *user* pada *profile* ini akan mendapatkan kecepatan akses 1 Mbps untuk upload/download. *Shared user* untuk dosen dengan jumlah 100 perangkat yang berarti 1 *user* Dosen dapat digunakan untuk 100 perangkat.



Gambar 4. 41 Membuat *profile* untuk Dosen
(Sumber : Data Olahan, 2024)

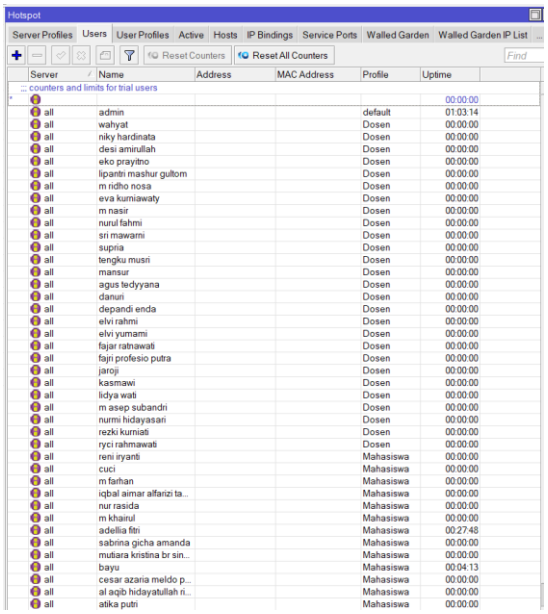
4.1.2.7 *Profile* Mahasiswa

Pada tahap ini tambahkan *profile* mahasiswa dengan kecepatan 512K/512K dan dengan *shared user* 50 perangkat.



Gambar 4. 42 Membuat *profile* untuk Mahasiswa
(Sumber : Data Olahan)

- Setelah itu tambahkan setiap *user-user* yang dibagi untuk 2 *profile* yang berbeda.



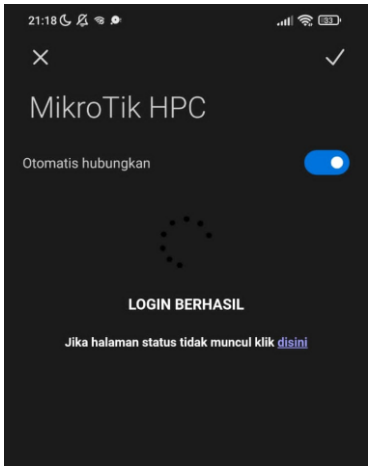
Gambar 4. 43 Tambahkan *user-user*
(Sumber : Data Olahan, 2024)

4.1.2.8 Uji Coba Konfigurasi *Hotspot*

Setelah Konfigurasi *hotspot* berhasil maka pada tahap ini mencoba memasukan *username* dan *password* untuk login ke *hotspot* laboratorium HPC.

Gambar 4. 44 Mencoba memasukan *username* dan *password*
(Sumber : Data Olahan, 2024)

- Dapat dilihat bahwa *user* dapat berhasil *login* kedalam *hotspot* laboratorium HPC.

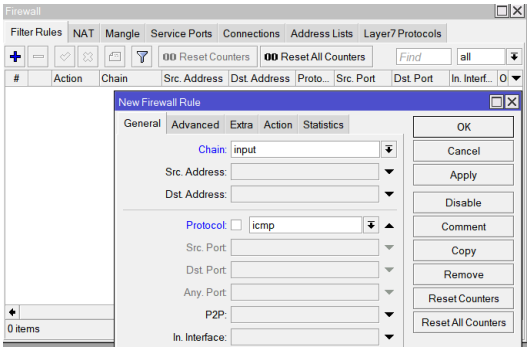


Gambar 4. 45 *User* dapat berhasil *login*
(Sumber : Data Olahan, 2024)

4.1.3. *Port Knocking*

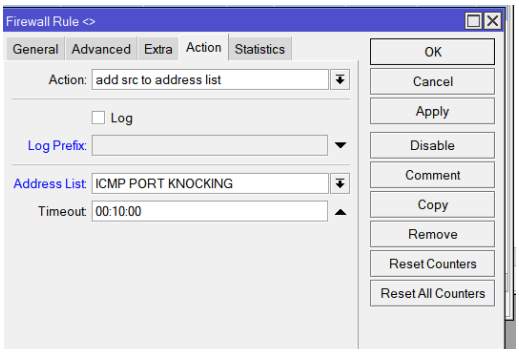
4.1.3.1 Konfigurasi *Rule Knocking* Pertama

Pada *Mikrotik* Pertama pada bagian *chain* pilih "*Input*", setelah itu pada bagian "*protoco*l" pilih "ICMP".



Gambar 4. 46 Konfigurasi *rule knocking* pertama
(Sumber : Data Olahan, 2024)

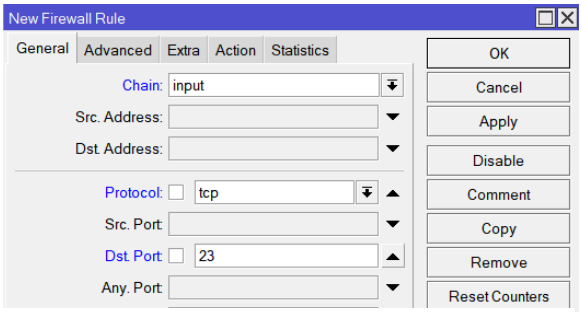
- kemudian pada tahap ini untuk memilih tab *action* pilih "*add src to addres list*" dan pada bagian *addres list* di isi dengan "*ICMP PORT KNOCKING*" dengan *time out* "10 menit" untuk mengaksesnya .



Gambar 4. 47 *Tab action ICMP port knocking*
(Sumber : Data Olahan, 2024)

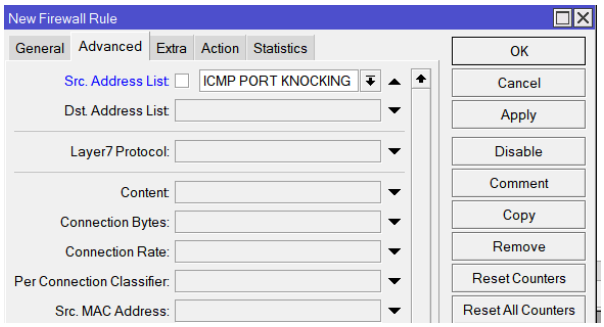
4.1.3.2 Konfigurasi *Rule Knocking* Kedua

Konfigurasi *rule knocking* kedua, pada tahap ini menampilkan bagian *chain* pilih tanda "*input"* setelah itu pada bagian *protoco*l "6 tcp" dan pada *Dst.port* isikan "23", *port* 23 ini untuk ketukan kedua dalam mengamankan *port* 23 (telnet).



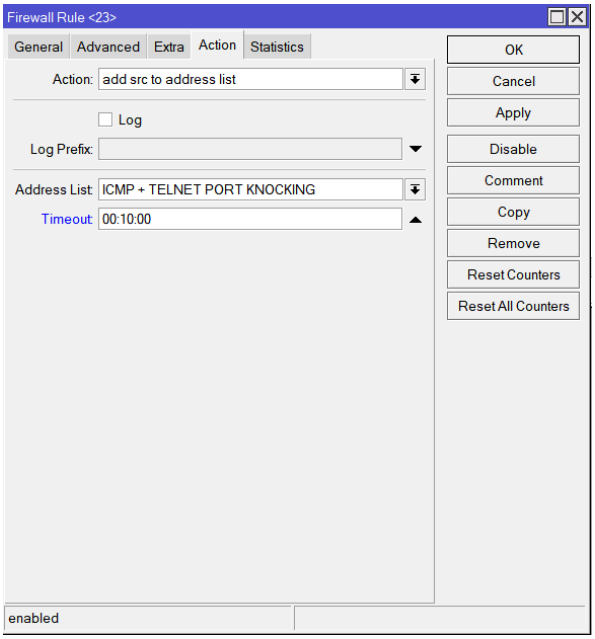
Gambar 4. 48 *Rule knocking* kedua
(Sumber : Data Olahan, 2024)

- pada bagian *Src Address list* di menampilkan pilihan "*ICMP PORT KNOCKING*".



Gambar 4. 49 *Src. Address List port* kedua
(Sumber : Data Olahan, 2024)

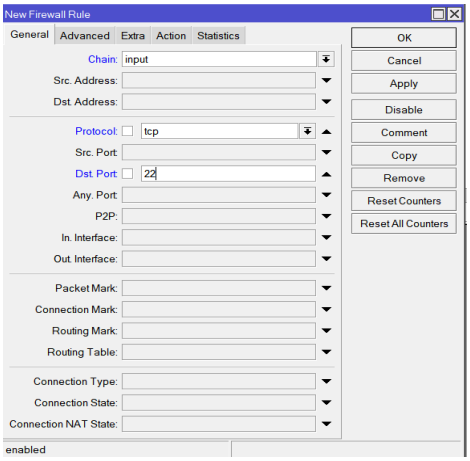
- Kemudian beralih pada *tab action* pilih " *add src to address list*" dan pada bagian *address list* isi dengan " *ICMP+TELNET PORT KNOCKING*" dengan " *timeout* " 10 menit untuk mengaksesnya.



Gambar 4. 50 Konfigurasi *rule knocking* kedua
(Sumber : Data Olahan, 2024)

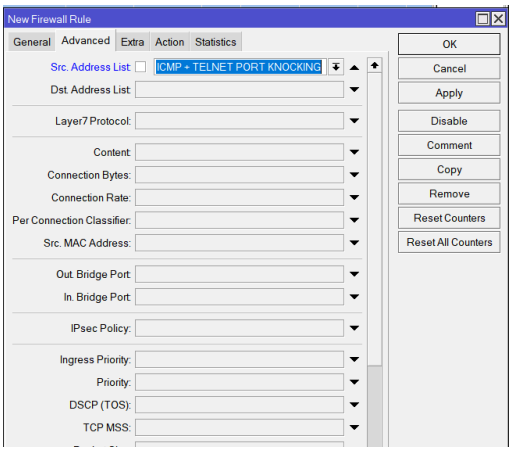
4.1.3.3 Konfigurasi *port knocking* pada SSH

Pada tahap ini menampilkan bagian *Chain* pilih "*input*", setelah itu pada bagian *protocol* pilih "6 tcp" dan pada *Dst.port* isikan *port* "22", *port* 22 ini untuk mengamankan *port* 22 (SSH). Kemudian beralih pada tab *Action* pilih "*add src to address list*" dan pada bagian *address list* isi dengan " *ICMP+TELNET+SSH PORT KNOCKING*" dengan *timeout* "10 menit" untuk mengaksesnya.



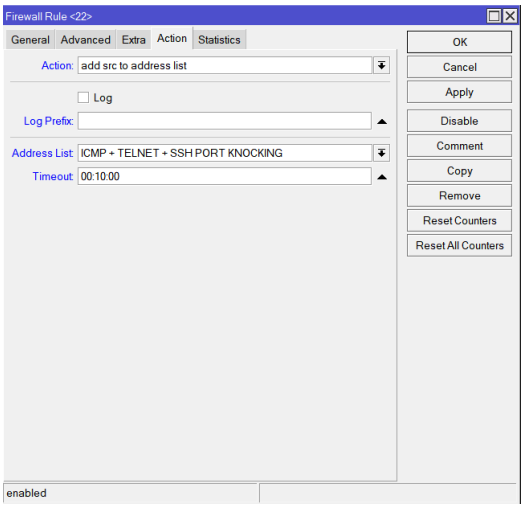
Gambar 4. 51 Konfigurasi *rule knocking* ketiga
(Sumber : Data Olahan, 2024*)*

- pada bagian *Src Address list* menampilkan untuk memilih "*ICMP TELNET PORT KNOCKING*".



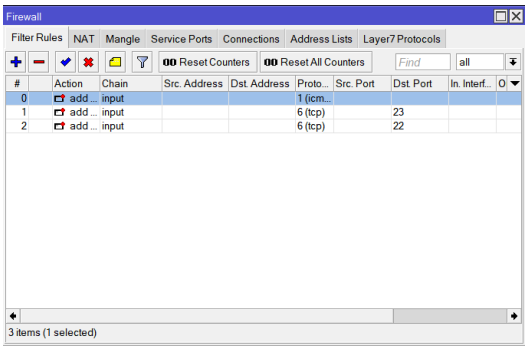
Gambar 4. 52 *Src. Address List ICMP* dan *Telnet*
(Sumber : Data Olahan, 2024*)*

- Kemudian beralih pada tab *Action* pilih "*add src to address list*" dan pada bagian *address list* isi dengan " *ICMP+TELNET+SSH PORT KNOCKING*" dengan *timeout* "10 menit" untu mengaksesnya.



Gambar 4. 53 *Tab Action Telnet*
(Sumber : Data Olahan, 2024*)*

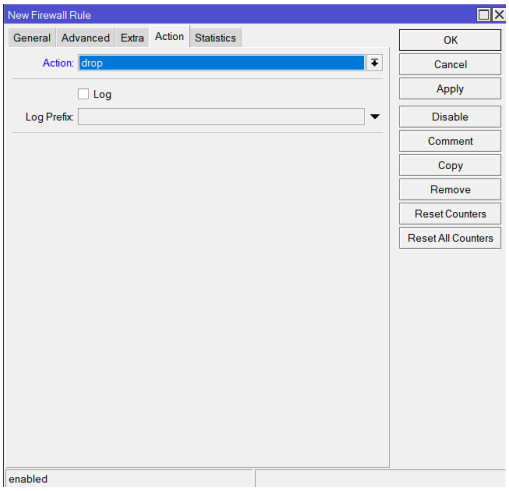
- Setelah berhasil maka pada halaman *tab filter rules* dapat dilihat ketukan pertama, ketukan kedua dan ketukan ketiga.



Gambar 4. 54 Halaman *Filter Rules*
(Sumber : Data Olahan, 2024*)*

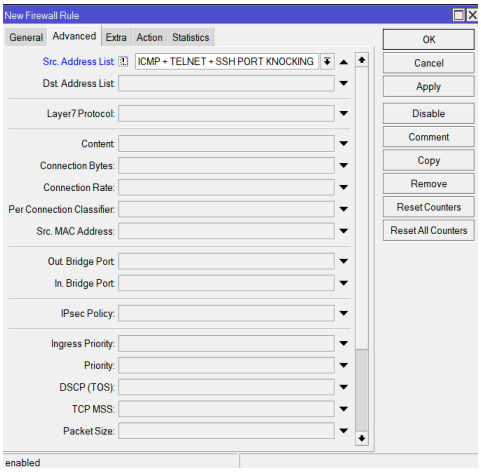
4.1.3.4 Konfigurasi *drop* pada *port* 8291,80,21,23,21

Pada bagian *chain* pilih"*input*", setelah itu pada bagian *Protocol* pilih "6 tcp" dan pada *Dst.port* isikan *port* 8291, 80, 21, 20, 23, 22. Kemudian beralih pada *Action* pilih "*drop*".



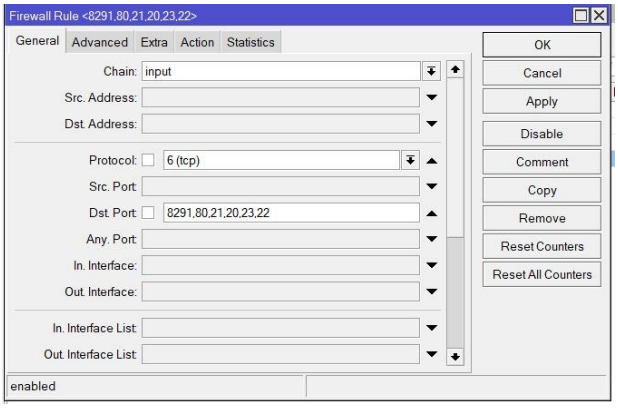
Gambar 4. 55 Perintah *drop port-port*
(Sumber : Data Olahan, 2024)

- lalu beralih ke *tab Advanced* pada bagian *Src.Address List* pilih "*ICMP+TELNET+SSH PORT KNOCKING*", klik centang pada kotak kecil (mengecualikan).



Gambar 4. 56 *Drop* selain *port knocking*
(Sumber : Data Olahan, 2024)

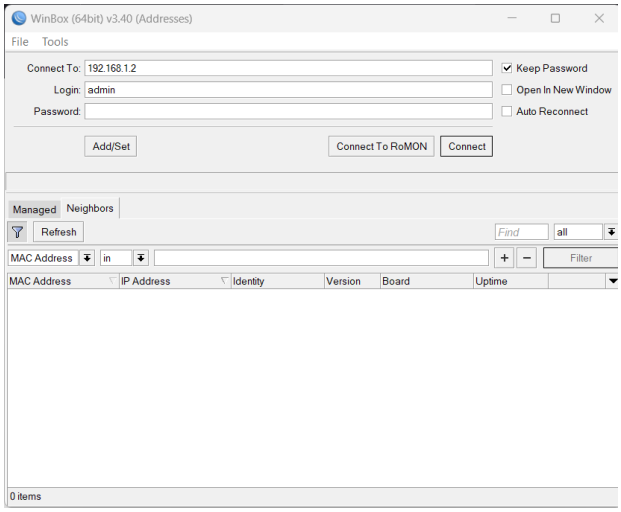
- Pilih *port-port* apa saja yang ingin di *drop.*



Gambar 4. 57 Konfigurasi *drop port knocking*
(Sumber : Data Olahan, 2024)

4.1.3.5 *Login mikrotik* setelah *port knocking*

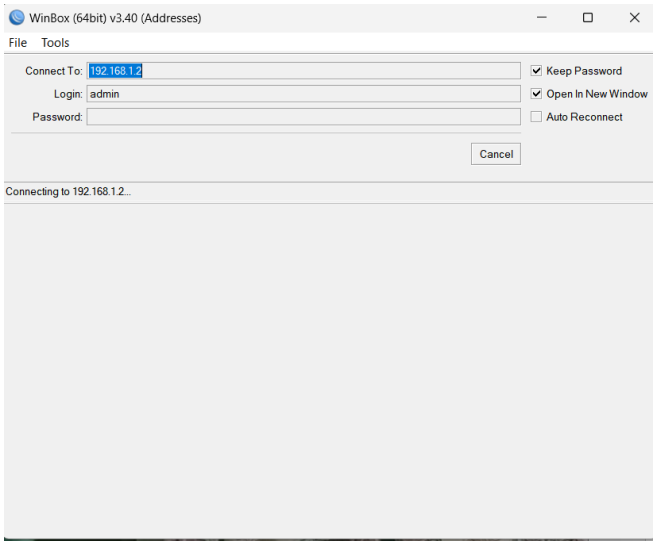
Maka pada *mikrotik* tidak akan bisa masuk dan juga *mikrotik* tidak akan terdeteksi pada *winbox.*



Gambar 4. 58 *Mikrotik* tidak akan bisa masuk *winbox*
(Sumber : Data Olahan, 2024)

4.1.3.6 Mencoba *login mikotik* pada *winbox*

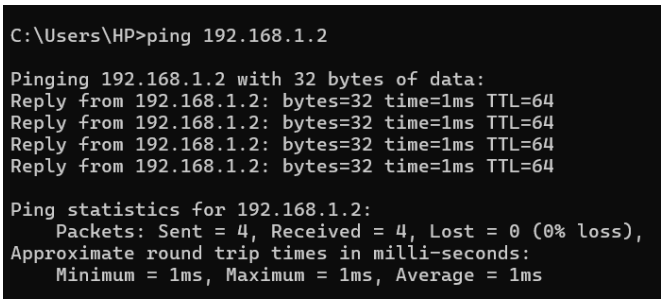
Ketika mencoba masuk dengan IP *mikrotik* yang diketahui melalui *winbox* maka *winbox* tidak akan memberikan izin mengakses *mikrotik* sebelum melakukan ketukan *port knocking* yang sudah dikonfigurasi sebelumnya.



Gambar 4. 59 Mencoba masuk dengan *IP mikrotik*
(Sumber : Data Olahan, 2024)

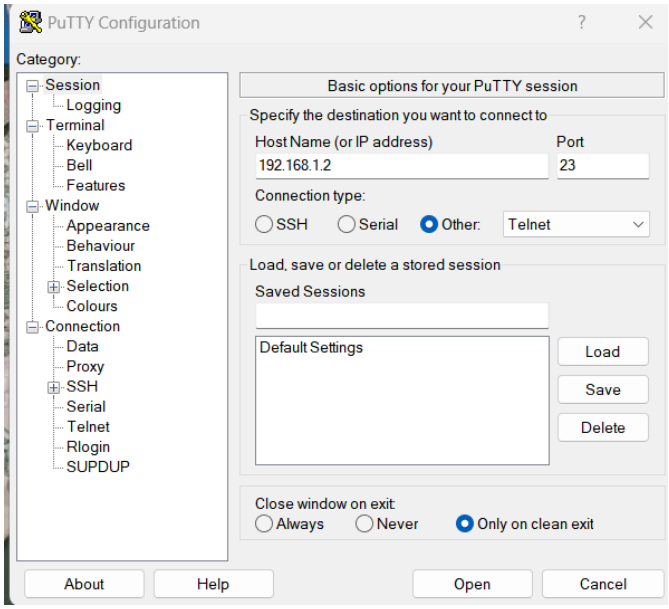
4.1.3.7 Uji Coba *Port Knocking*

a. Langkah pertama untuk menguji coba rule pertama dengan *ping ip mikrotik* (ICMP) dan memastikan konektivitas antara *client* dan *router mikrotik* perlu dilakukan *testing* dengan melakukan *ping* melalui CMD dengan perintah *ping* 192.168.1.2.



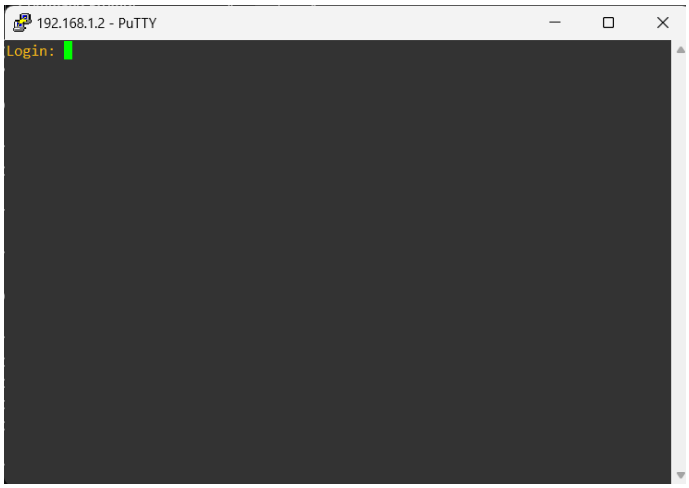
Gambar 4. 60 *Ping ip mikrotik* melalui CMD
(Sumber : Data Olahan, 2024)

b. Uji coba *rule* kedua dengan mengetuk "*port* 23" atau Telnet ketujuan IP *mikrotik* yaitu 192.168.1.2 dengan menggunakan aplikasi *PuTTY*.



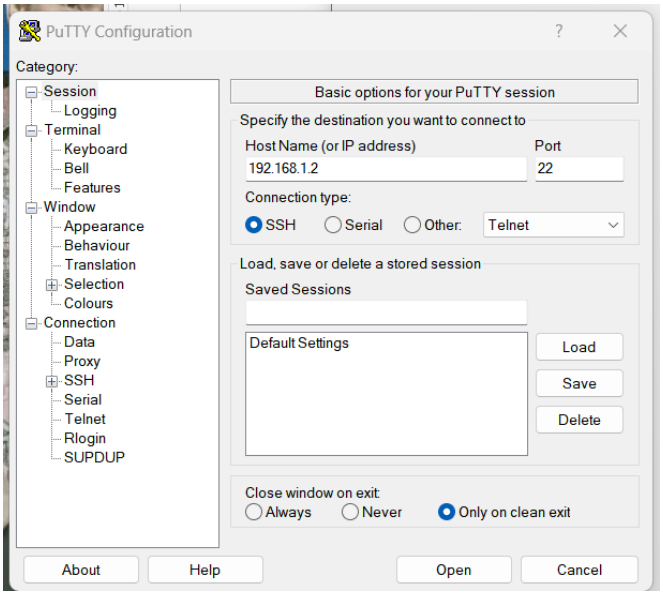
Gambar 4. 61 *Mengetuk "port 23"*
(Sumber : Data Olahan, 2024)

- Maka ini hasil ketika mencoba melakukan ketukan pada *telnet* atau *port* 23.



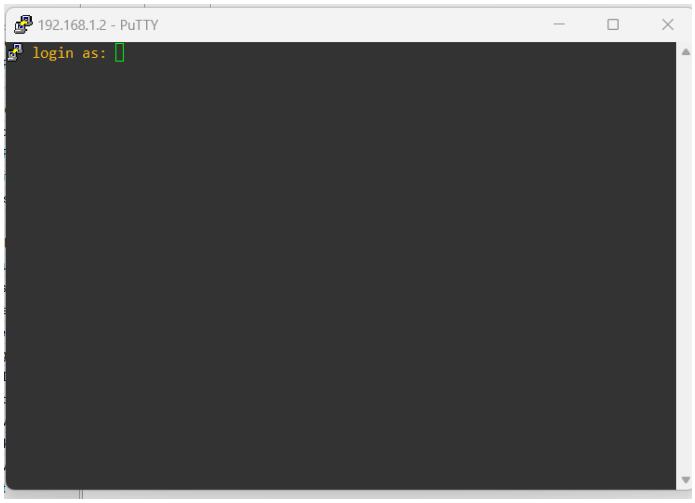
Gambar 4. 62 Ketukan pada *telnet*
(Sumber : Data Olahan, 2024)

- Lalu uji coba *rule* ketiga dengan mengetuk "*port* 22" atau SSH ketujuan ip *mikrotik* yaitu 192.168.1.2 dengan menggunakan aplikasi *PuTTY*.



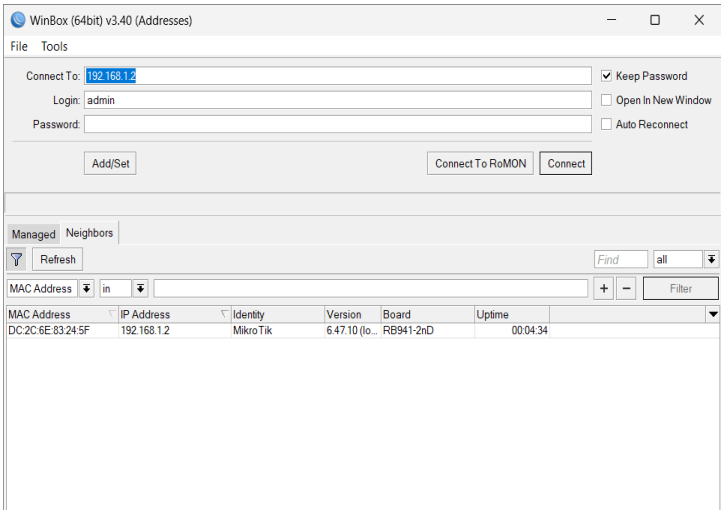
Gambar 4. 63 Mengetuk *"port 22"*
(Sumber : Data Olahan, 2024)

- Maka ini hasil ketika mencoba melakukan ketukan pada SSH atau *port* 22.



Gambar 4. 64 Ketukan pada SSH
(Sumber : Data Olahan, 2024)

- Hasil pengujian diatas yang telah dilakukan menunjukkan IP yang melakukan *port knocking* berhasil dan *mikrotik* berhasil terdeteksi pada aplikasi *winbox.*



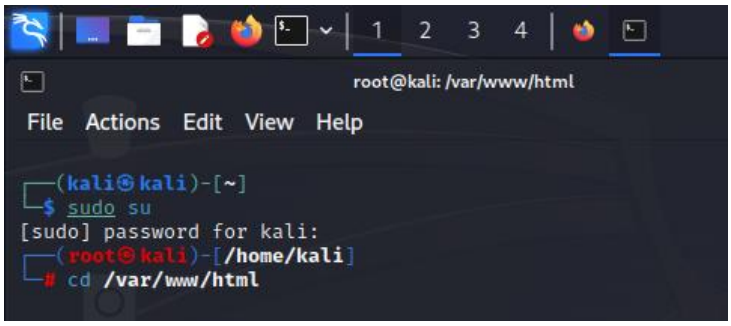
Gambar 4. 65 *Mikrotik* berhasil terdeteksi pada aplikasi *winbox*
(Sumber : Data Olahan, 2024)

4.1.4. Install DVWA pada *Kali Linux*

DVWA membutuhkan *Apache* sebagai *web server*, MySQL sebagai *database*, dan PHP untuk pemrograman *server-side*.

4.1.4.1. Konfigurasi DVWA

- Pada tahap ini menampilkan Install DVWA Pada *Kali Linux.*
- Pada tahap ini menampilkan perintah ***cd /var/www/html*** *digunakan* untuk mengubah direktori kerja saat ini di terminal ke direktori ***/var/www/html.***



Gambar 4. 66 Perintah masuk ke direktori
(Sumber : Data Olahan, 2024)

- Kemudian pada tahap ini menampilkan lokasi direktori tempat dimana berada saat ini dalam struktur *file* sistem gunakan perintah ***pwd.***

Gambar 4. 67 *Pwd*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan unduh DVWA dari repositori *GitHub.*

Gambar 4. 68 Unduh DVWA
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan perintah ***Sudo chmod –R 777 DVWA/*** akan memberikan izin penuh untuk membaca, menulis dan mengeksekusi semua *file* dan direktori di dalam direktori **DVWA/.**

Gambar 4. 69 Izin penuh pada *direktori DVWA*
(Sumber : Data Olahan,2024)

- Pada tahap ini menampilkan ubah dan masuk ke dalam direktori kerja ***config*** yang berada dalam direktori DVWA dengan perintah ***cd DVWA/config.***

Gambar 4. 70 Masuk ke direktori kerja *config*
(Sumber : Data Olahan, 2024)

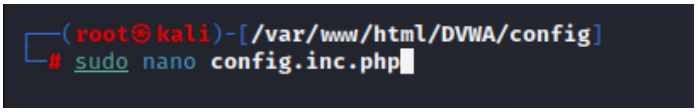
- Pada tahap ini menampilkan salinan *file* ***config.inc.php.dist*** dengan nama baru ***config.inc.php*** dengan perintah ***sudo cp config.inc.php.dist config.inc.php.***

Gambar 4. 71 Salinan *file config*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan perintah ***ls*** untuk melihat ***file config.inc*** yang telah kita buat.

Gambar 4. 72 Perintah *ls*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan edit *file* ***config.inc.php*** dengan perintah ***sudo nano config.inc.php.***

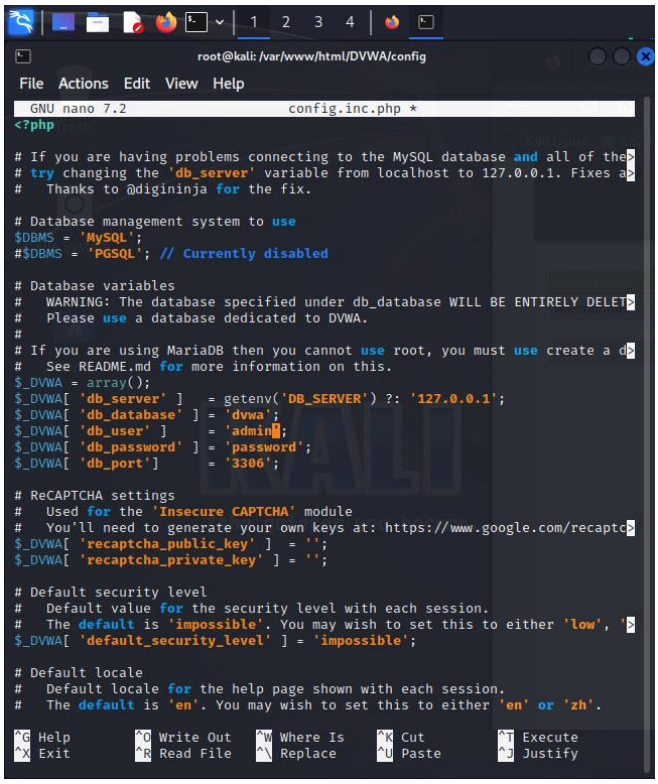


Gambar 4. 73 Perintah edit *file config*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan *file* terbuka lalu ubah

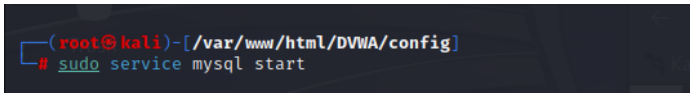
  db_user : admin.

  db_password : password.



Gambar 4. 74 Ubah *username* dan *password*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan *install MySQL* dalam *Kali Linux* dengan perintah ***sudo service mysql start.***



Gambar 4. 75 Jalankan MySQL
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan *prompt* MySQL untuk menjalankan perintah SQL dan mengelola *database*, tabel, pengguna dengan masukan kata sandi MySQL yaitu 123. Kemudian tambahkan *database* pada DVWA.

Gambar 4. 76 Mengelola *database*
(Sumber : Data Olahan, 2024)

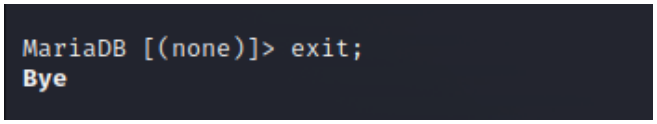
- Pada tahap ini menampilkan perintah SQL untuk membuat pengguna baru di MySQL dengan nama pengguna admin, yang diidentifikasi dengan kata sandi *password*, dan membatasi akses hanya dari alamat IP 127.0.0.1 (*localhost*).

Gambar 4. 77 Perintah SQL
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan perintah SQL untuk memberikan semua hak istimewa pada semua tabel dalam *database* dvwa kepada pengguna admin yang hanya dapat terhubung dari alamat IP 127.0.0.1.

Gambar 4. 78 Perintah SQL untuk hak istimewa ke semua tabel
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan perintah ***exit*** untuk keluar dari klien MySQL.

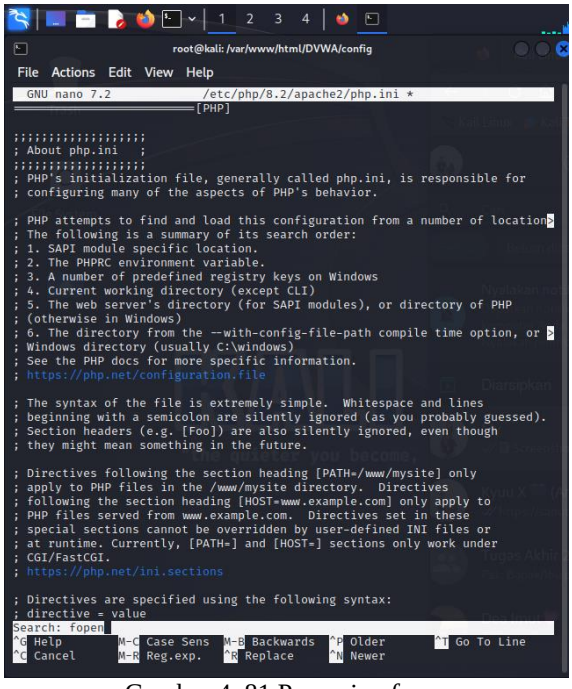


Gambar 4. 79 Perintah *exit*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan perintah ***sudo nano etc/php/8.2/apache2 /php.ini*** digunakan untuk membuka file konfigurasi php.ini untuk PHP versi 8.2 yang digunakan oleh ***server web Apache.***

Gambar 4. 80 Perintah untuk membuka *file php*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan ***fopen*** untuk pada *file* php.ini kemudian aktifkan ***allow_url_include.***



Gambar 4. 81 Pencarian *fopen*
(Sumber : Data Olahan, 2024)

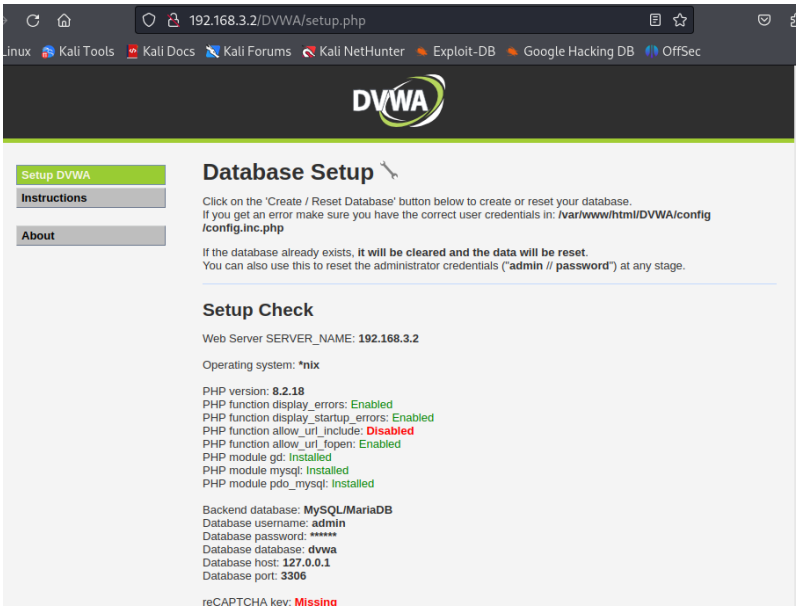
- Pada tahap ini menampilkan layanan *apache HTTP server* pada sistem dengan perintah ***sudo service apache2 start.***

Gambar 4. 82 Jalankan *apache2*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan *Reload* konfigurasi *apache HTTP Server* tanpa memutuskan koneksi yang ada atau menghentikan layanan dengan perintah ***sudo service apache2 reload.***

Gambar 4. 83 Konfigurasi *apache2*
(Sumber : Data Olahan, 2024)

- Pada tahap ini menampilkan DVWA sudah berhasil di *install* pada *Kali Linux* lalu ketikan 192.168.3.2/DVWA untuk melihat apakah DVWA berhasil dimana 192.168.3.2 adalah *IP Private* dari *web server*.



Gambar *4. 84 DVWA* berhasil di *install*
(Sumber : Data Olahan, 2024)

4.1.5. *Forward Web Server* ke dalam NAT *mikrotik*

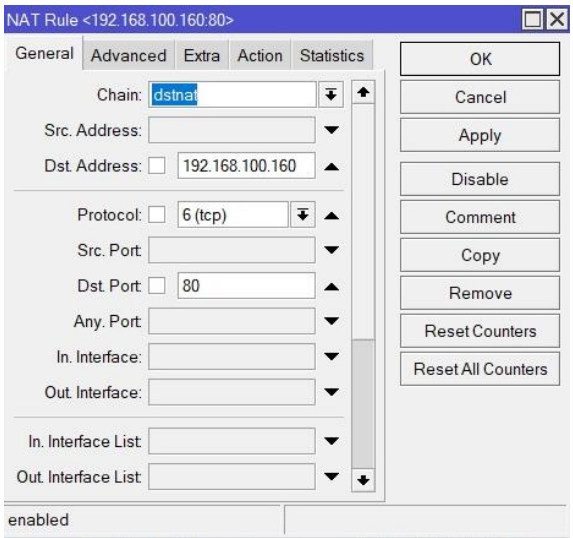
Agar *Web server* bisa diakses dari internet, *set fowarding* di router *mikrotik* dengan fitur *firewall* NAT. *Fowarding* ini akan membalokkan *traffic* yang menuju ke IP publik yang terpasang di router menuju ke IP lokal *web server*. Dengan begitu, seolah-olah *client* dari internet berkomunikasi dengan *web server* meminjam IP dari *ISP mikrotik router mikrotik*. Langkah pembuatan *rule*, masuk ke menu **IP --> Firewall --> klik tab "NAT"**, tambahkan *rule* baru dengan menekan tombol "*add"* atau tanda "+" berwarna merah.

4.1.5.1. Konfigurasi NAT

a) *rule* pada *tab general*

Pada tahap ini menampilkan *rule* pada *tab general* dengan keterangan perintah :

- ***Chain****:* Pilih *dstnat*.
- ***Dst. Address****:* Masukkan IP publik 192.168.100.160 .
- ***Protocol****:* Pilih tcp.
- ***Dst. Port****:* Tentukan *port* yang ingin Anda *forward*, misalnya 80 untuk HTTP.

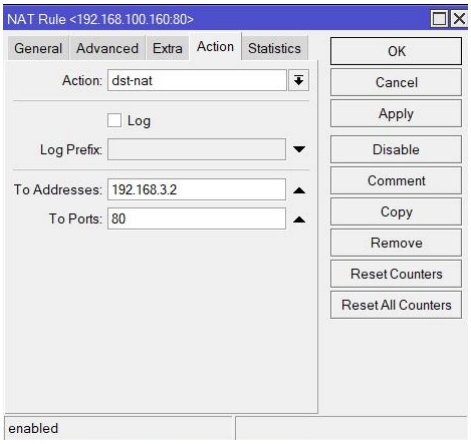


Gambar 4. 85 *Forward* pada *tab general*
(Sumber : Data Olahan, 2024)

b) *rule* pada *tab chain*

Pada tahap ini menampilkan *rule* pada *tab general* dengan keterangan perintah :

- ***Action****:* Pilih *dst-nat*.
- ***To Addresses****:* Masukkan alamat IP lokal 192.168.3.2.
- ***To Ports****:* Masukkan *port* tujuan yang sama dengan *port* pada ***Dst. Port.***



Gambar *4. 86 Forward* pada *tab action*
(Sumber : Data Olahan, 2024)

4.1.5.2. Verifikasi dan Uji Coba

Pada tahap ini menampilkan mengatur NAT maka untuk menguji konfigurasi dari IP publik dengan menggunkan *browser web*.

Gambar 4. 87 Verifikasi dan Uji Coba NAT
(Sumber : Data Olahan)

4.1.6. Keamanan IP *Private web server* menggunakan *IP Tables mikrotik*

Pada tahap ini menampilkan keamanan pada *web server* maka gunakan *iptables* untuk melakukan *drop* pada IP penyerang dengan IP penyerang 192.168.3.3.

Penjelasan dari perintah ini:

- ***iptables***: Perintah untuk mengonfigurasi aturan *firewall* di *Linux*.
- ***-A INPUT***: Menambahkan aturan ke rantai INPUT, yang menangani paket masuk ke sistem.
- **-s 192.168.3.4**: Menentukan alamat ***IP address*** penyerang yaitu 192.168.3.3 yang ingin diblokir.
- ***-j DROP***: Menentukan tindakan yang diambil terhadap paket yang cocok dengan aturan ini. *DROP* berarti paket akan dihapus dan tidak akan diproses lebih lanjut.

Gambar 4. 88 *Ip tables web server*
(Sumber : Data Olahan, 2024)

Table 4. 1 Keamanan *Web server* dengan *iptables*

IP address Web Server (alamat IP lokal)	192.168.3.4
IP address Penyerang (LOIC)	192.168.3.3

4.1.7. Keamanan IP dari *ISP* pada *web server* menggunakan *firewall raw mikrotik*

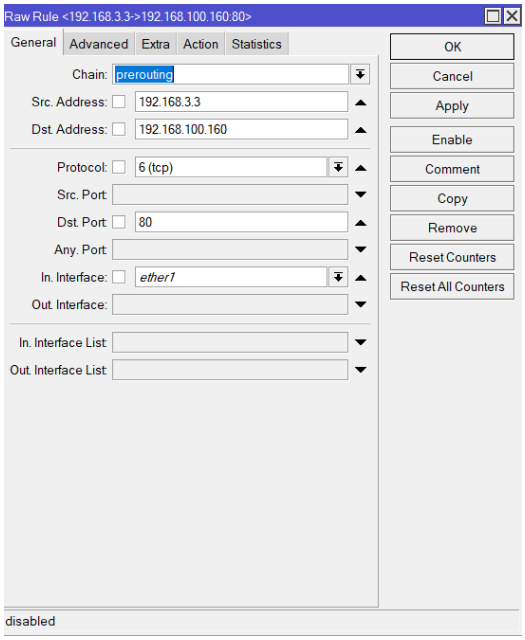
4.1.7.1 *Firewall Raw* mengatasi serangan SYN *Flood*

a) Tab General

Pada tahap ini menampilkan penggunaan aturan *raw* untuk membatasi koneksi baru dan mengurangi beban pada *web server.*

Dengan IP penyerang 192.168.3.3.

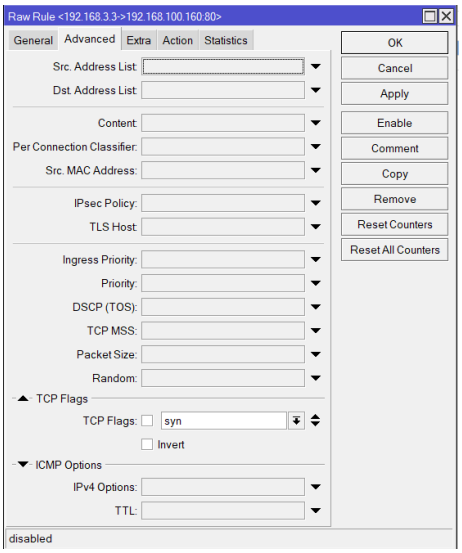
- ***chain=prerouting***: Aturan ini diterapkan sebelum paket diteruskan ke proses *routing.*
- ***protocol=tcp***: Aturan ini berlaku untuk paket TCP.
- ***dst-port=80***: Paket yang ditargetkan adalah yang menuju *port* 80 *port* umum untuk HTTP.
- ***In.Interface=ether1****:* Aturan ini akan diterapkan hanya untuk paket yang masuk melalui antarmuka *ether1*.



Gambar 4. 89 Keamanan *firewall raw* untuk serangan *SYN flood*
(Sumber : Data Olahan, 2024)

b) *Tab Advanced*

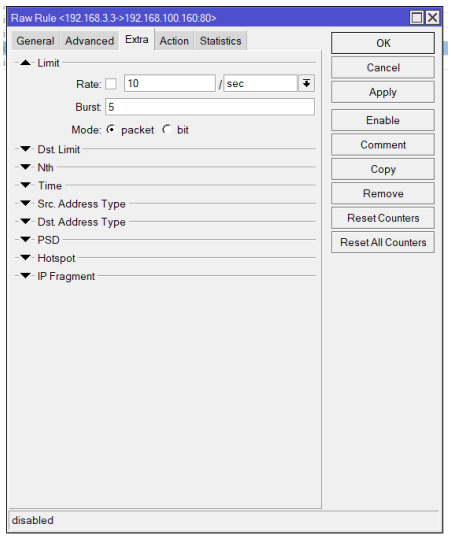
Pada *tab advanced* menampilkan pilih ***tcp-flags=syn***: Aturan ini hanya berlaku untuk paket SYN (bagian dari tiga arah *handshake* TCP).



Gambar *4. 90 Tab advanced syn*
*(*Sumber : Data Olahan, 2024)

c) *Tab Extra*

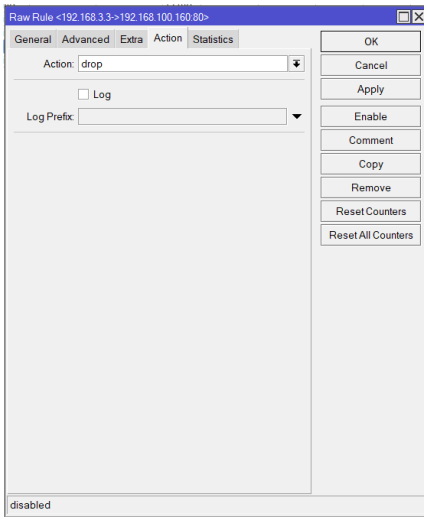
Pada *Tab Extra* menampilkan ***limit=10,5***: Membatasi jumlah koneksi baru dari IP tertentu menjadi 10 per detik dengan burst hingga 5.



Gambar *4. 91 Tab extra Syn*
(Sumber : Data Olahan, 2024)

d) *Tab Action*

Pada *Tab Action* menampilkan ***action=drop***: Paket yang sesuai aturan ini akan dibuang, sehingga tidak akan mencapai *web server*.



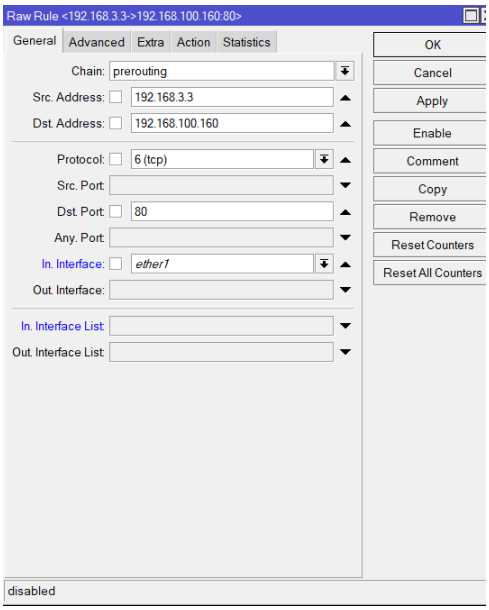
Gambar *4. 92 Tab action Syn*
(Sumber : Data Olahan, 2024)

4.1.7.2 *Firewall Raw* mengatasi serangan HTTP *flood*

a. Tab *General* untuk keaamanan HTTP *flood*

Pada tahap ini menampilkan cara untuk mengatasi serangan HTTP *flood* menggunakan aturan *raw* untuk membatasi jumlah paket atau koneksi.

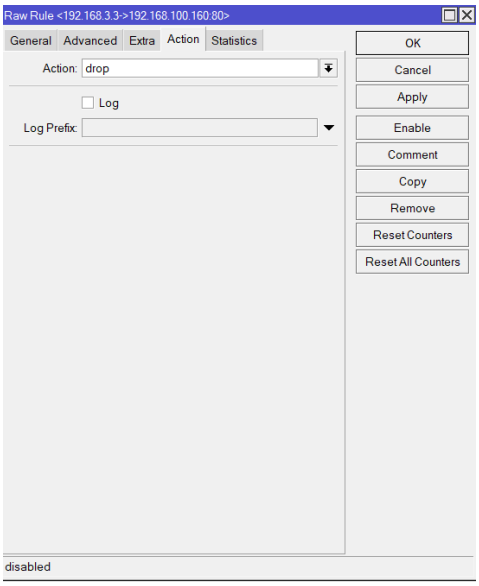
- ***chain=prerouting***: Aturan ini diterapkan sebelum paket diteruskan ke proses *routing.*
- ***protocol=tcp***: Aturan ini berlaku untuk paket TCP.
- ***dst-port=80***: Paket yang ditargetkan adalah yang menuju *port* 80 *port* umum untuk HTTP.
- ***In.Interface=ether1****:* Aturan ini akan diterapkan hanya untuk paket yang masuk melalui antarmuka *ether1*.



Gambar 4. 93 *Tab general* HTTP
(Sumber : Data Olahan, 2024)

b. Tab *Action* untuk keamanan HTTP *flood*

Pada *Tab Action* menampilkan pilihan ***action=drop***: Paket yang sesuai aturan ini akan dibuang, sehingga tidak akan mencapai *web server*.



Gambar 4. 94 *Tab action drop* HTTP
(Sumber : Data Olahan)

4.2 Pengujian

4.2.1. Port Scanning

Port Scanning menggunakan *nmap. Nmap* berfungsi untuk mengidentifikasi port yang terbuka pada *host* atau jaringan.

Penyerang :

OS : *Kali Linux.*

IP : 192.168.3.253.

Target

Mikrotik RouterBoard (RB941-2Nd) HAP Lite.

IP :192.168.100.158 *(ether1).*

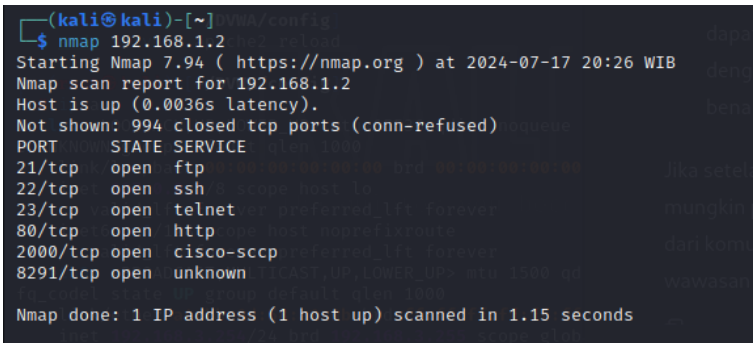
IP : 192.168.1.2 *(ether2).*

IP : 192.168.3.1 *(ether3).*

IP : 192.168.200.1 *(wlan1).*

4.2.1.1. *Port Scanning* IP *mikrotik*

Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.1.2 sebelum diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* terbuka.

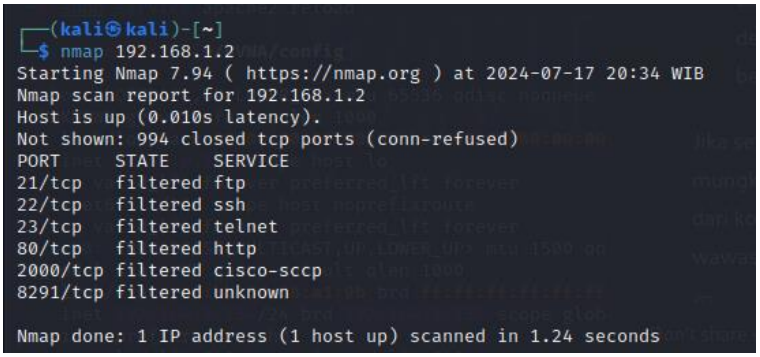


Gambar 4. 95 *Nmap* 192.168.1.2 Sebelum diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 2 Sebelum Penerapan *Port Knocking* pada *ether2*

Hasil Sebelum keamanan *port knocking* pada *ether2*	Deskripsi
1) 21 (*File Transfer Protocol*) -> *Open* 2) 22 (*Secure Shell*) -> *Open* 3) 23 (*Telnet*) -> *Open* 4) 80 (*HyperText Transfer Protocol*) -> *Open* 5) 2000 (*Skinny Client Control Protocol*) -> *Open* 6) 8291 (*Winbox*)-> *Open*	1 *FTP server* berjalan dan siap menerima koneksi untuk transfer file. 2 *SSH server* berjalan dan siap menerima koneksi untuk akses shell yang aman. 3 *Telnet server* berjalan dan siap menerima koneksi untuk akses shell yang tidak terenkripsi. 4 *Web server* berjalan dan siap menerima koneksi HTTP untuk melayani halaman web. 5 Layanan yang menggunakan SCCP berjalan dan siap menerima koneksi. 6 *MikroTik Winbox service* berjalan dan siap menerima koneksi untuk konfigurasi *router* melalui utilitas *Winbox.*
Pada hasil diatas untuk alamat IP 192.168.1.2 dengan semua port yang tidak dilakukan penerapan port knocking menunjukan status *open*	Jika semua port tersebut dalam keadaan "*open*", maka layanan yang berjalan pada *port* tersebut terbuka untuk koneksi dan dapat diakses dari jaringan. Ini juga berarti bahwa *firewall* tidak memblokir atau melakukan *filter* akses ke *port-port* tersebut.

- Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.1.2 setelah diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* tertutup.



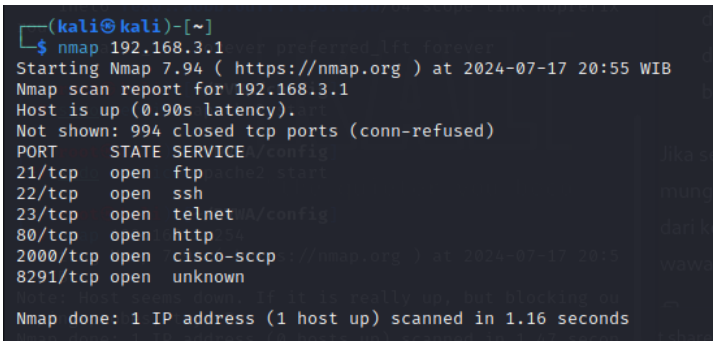
Gambar 4. 96 *Nmap* 192.168.1.2 Setelah diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 3 Setelah Penerapan *Port Knocking* pada *ether2*

Hasil Setelah keamanan *port knocking* pada *ether2*	Deskripsi
Pada hasil scan *nmap* 192.168.1.2 yang sudah diterapkan *port-port* 21, 22, 23, 80, 2000, dan 8291 dalam keadaan tertutup (*filtered*)	Pada hasil di atas, semua *port* yang tertutup oleh aturan firewall menunjukkan status "*filtered*". Maka *firewall MikroTik* menghalangi akses ke *port-port* tersebut sampai urutan *port knocking* yang benar dilakukan.

4.2.1.1. *Port Scanning* IP *ether3*

Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.3.1 sebelum diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* terbuka.



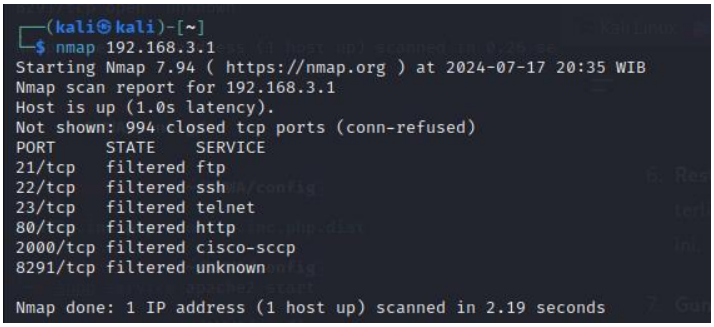
Gambar 4. 97 *Nmap* 192.168.3.1 Sebelum diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 4 Sebelum Penerapan *Port Knocking* pada *ether3*

Hasil Sebelum keamanan *port knocking*	Deskripsi
7) 21 (*File Transfer Protocol*) -> *Open* 8) 22 (*Secure Shell*) -> *Open* 9) 23 (*Telnet*) -> *Open* 10) 80 (*HyperText Transfer Protocol*) -> *Open* 11) 2000 (*Skinny Client Control Protocol*) -> *Open* 12) 8291 (*Winbox*)-> *Open*	7 *FTP server* berjalan dan siap menerima koneksi untuk transfer file. 8 *SSH server* berjalan dan siap menerima koneksi untuk akses shell yang aman. 9 *Telnet server* berjalan dan siap menerima koneksi untuk akses shell yang tidak terenkripsi. 10 *Web server* berjalan dan siap menerima koneksi HTTP untuk melayani halaman web. 11 Layanan yang menggunakan SCCP berjalan dan siap menerima koneksi. 12 *MikroTik Winbox service* berjalan dan siap menerima koneksi untuk konfigurasi *router* melalui utilitas *Winbox.*

Pada hasil diatas untuk alamat IP 192.168.3.1 dengan semua port yang tidak dilakukan penerapan port knocking menunjukan status *open*	Jika semua port tersebut dalam keadaan "*open*", maka layanan yang berjalan pada *port* tersebut terbuka untuk koneksi dan dapat diakses dari jaringan. Ini juga berarti bahwa *firewall* tidak memblokir atau melakukan *filter* akses ke *port-port* tersebut.

- Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.3.1 setelah diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* tertutup.



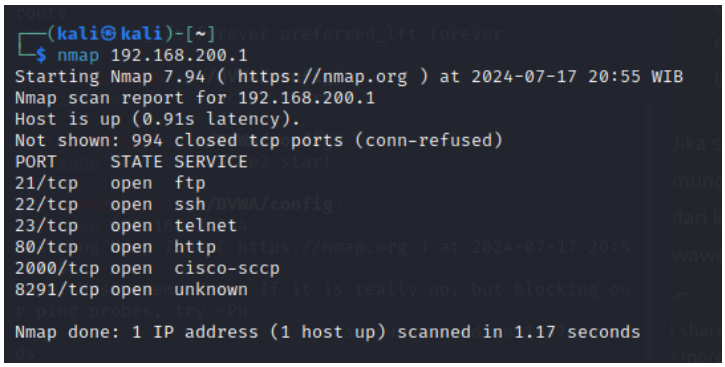
Gambar 4. 98 *Nmap 192.168.3.1* Setelah diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 5 Setelah Penerapan *Port Knocking* pada *ether3*

Hasil Setelah keamanan *port knocking* pada *ether3*	Deskripsi
Pada hasil scan *nmap* 192.168.1.2 yang sudah diterapkan *port-port* 21, 22, 23, 80, 2000, dan 8291 dalam keadaan tertutup (*filtered*)	Pada hasil di atas, semua *port* yang tertutup oleh aturan firewall menunjukkan status "*filtered*". Maka *firewall MikroTik* menghalangi akses ke *port-port* tersebut sampai urutan *port knocking* yang benar dilakukan.

4.2.1.2. *Port Scanning* IP *wlan1*

Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.200.1 sebelum diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* terbuka.



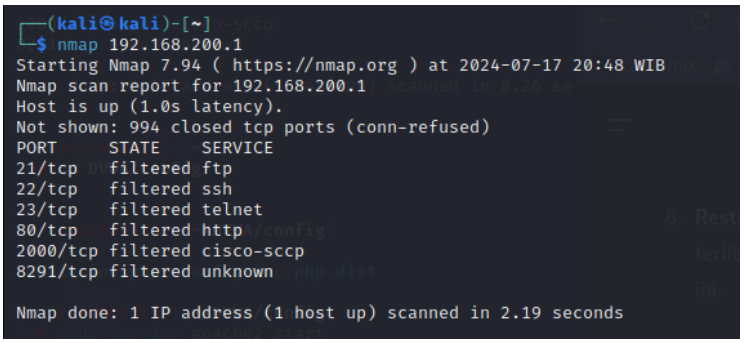
Gambar 4. 99 *Nmap* 192.168.200.1 Sebelum diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 6 Hasil Sebelum keamanan *port knocking* pada *wlan1*

Hasil Sebelum keamanan *port knocking* pada *wlan1*	Deskripsi
13) 21 (*File Transfer Protocol*) -> *Open* 14) 22 (*Secure Shell*) -> *Open* 15) 23 (*Telnet*) -> *Open* 16) 80 (*HyperText Transfer Protocol*) -> *Open* 17) 2000 (*Skinny Client Control Protocol*) -> *Open* 18) 8291 (*Winbox*)-> *Open*	13 *FTP server* berjalan dan siap menerima koneksi untuk transfer file. 14 *SSH server* berjalan dan siap menerima koneksi untuk akses shell yang aman. 15 *Telnet server* berjalan dan siap menerima koneksi untuk akses shell yang tidak terenkripsi. 16 *Web server* berjalan dan siap menerima koneksi HTTP untuk melayani halaman web. 17 Layanan yang menggunakan SCCP berjalan dan siap menerima koneksi. 18 *MikroTik Winbox service* berjalan dan siap menerima

	koneksi untuk konfigurasi *router* melalui utilitas *Winbox*.
Pada hasil diatas untuk alamat IP 192.168.200.1 dengan semua port yang tidak dilakukan penerapan port knocking menunjukan status *open*	Jika semua port tersebut dalam keadaan "*open*", maka layanan yang berjalan pada *port* tersebut terbuka untuk koneksi dan dapat diakses dari jaringan. Ini juga berarti bahwa *firewall* tidak memblokir atau melakukan *filter* akses ke *port-port* tersebut.

- Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.200.1 setelah diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* tertutup.



Gambar 4. 100 *Nmap* 192.168.200.1 Setelah diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

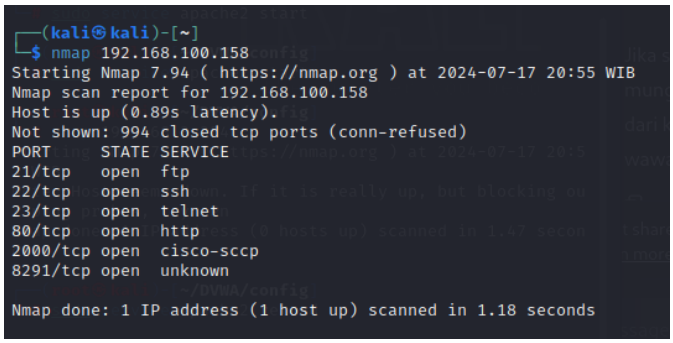
Table 4. 7 Hasil Setelah keamanan *port knocking* pada *wlan1*

Hasil Setelah keamanan *port knocking pada wlan1*	Deskripsi
Pada hasil scan *nmap* 192.168.200.1 yang sudah diterapkan *port-port* 21, 22, 23, 80, 2000, dan 8291 dalam keadaan tertutup (*filtered*)	Pada hasil di atas, semua *port* yang tertutup oleh aturan firewall menunjukkan status "*filtered*". Maka *firewall MikroTik* menghalangi akses ke *port-port*

	tersebut sampai urutan *port knocking* yang benar dilakukan.

4.2.1.3. *Port Scanning* IP *mikrotik*

Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.100.158 sebelum diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* terbuka.



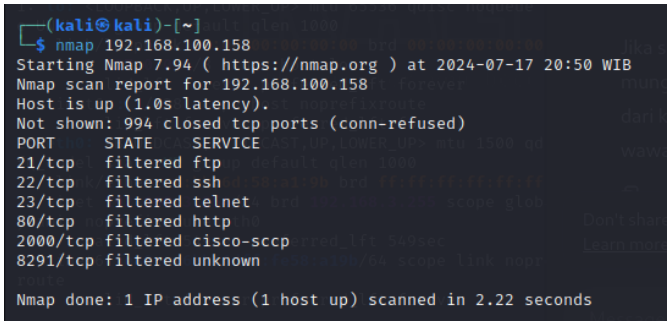
Gambar 4. 101 *Nmap* 192.168.100.158 Sebelum diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 8 Hasil Sebelum keamanan *port knocking* pada *ether1*

Hasil Sebelum keamanan *port knocking* pada *ether1*	Deskripsi
19) 21 (*File Transfer Protocol*) -> *Open* 20) 22 (*Secure Shell*) -> *Open* 21) 23 (*Telnet*) -> *Open* 22) 80 (*HyperText Transfer Protocol*) -> *Open* 23) 2000 (*Skinny Client Control Protocol*) -> *Open*	19 *FTP server* berjalan dan siap menerima koneksi untuk transfer file. 20 *SSH server* berjalan dan siap menerima koneksi untuk akses shell yang aman. 21 *Telnet server* berjalan dan siap menerima koneksi untuk akses shell yang tidak terenkripsi. 22 *Web server* berjalan dan siap menerima koneksi HTTP untuk melayani halaman web. 23 Layanan yang menggunakan SCCP berjalan dan siap menerima koneksi.

24) 8291 (*Winbox*)-> *Open*	24 *MikroTik Winbox service* berjalan dan siap menerima koneksi untuk konfigurasi *router* melalui utilitas *Winbox.*
Pada hasil diatas untuk alamat IP 192.168.1.2 dengan semua port yang tidak dilakukan penerapan port knocking menunjukan status *open*	Jika semua port tersebut dalam keadaan "*open*", maka layanan yang berjalan pada *port* tersebut terbuka untuk koneksi dan dapat diakses dari jaringan. Ini juga berarti bahwa *firewall* tidak memblokir atau melakukan *filter* akses ke *port-port* tersebut.

- Pada tahap ini menampilkan pemindaian 1000 *port* TCP pada IP target 192.168.100.158 setelah diterapkan *port knocking* maka dapat dilihat pada gambar semua *port* tertutup.



Gambar 4. 102 Setelah diterapkan *port knocking*
(Sumber : Data Olahan, 2024)

Table 4. 9 Hasil Setelah keamanan *port knocking* pada *ether1*

Hasil Setelah keamanan *port knocking* pada *ether1*	Deskripsi
Pada hasil scan *nmap* 192.168.100.158 yang sudah diterapkan *port-port* 21, 22, 23, 80, 2000, dan 8291 dalam keadaan tertutup (*filtered*)	Pada hasil di atas, semua *port* yang tertutup oleh aturan firewall menunjukkan status "*filtered*". Maka *firewall MikroTik* menghalangi akses ke *port-port* tersebut sampai urutan *port knocking* yang benar dilakukan.

4.2.2. *DDOS attack (UDP flood, SYN flood* dan *HTTP flood)* setelah penerapan *iptables*

DDOS atttack menggunakan aplikasi *LOIC* dapat mengirimkan sejumlah besar permintaan ke *server* target, menyebabkan *overload* dan mengganggu akses normal.

4.2.2.1. *DDOS attack* setelah penerapan *ip tables*

Berikut adalah jenis penyerangan DDOS *Attack* yaitu SYN *flood* dan HTTP *flood, UDP flood setelah* penerapan *iptables* pada *IP web server.*

Penyerang :

LOIC (*LOW ORBIT ION CANNON*)

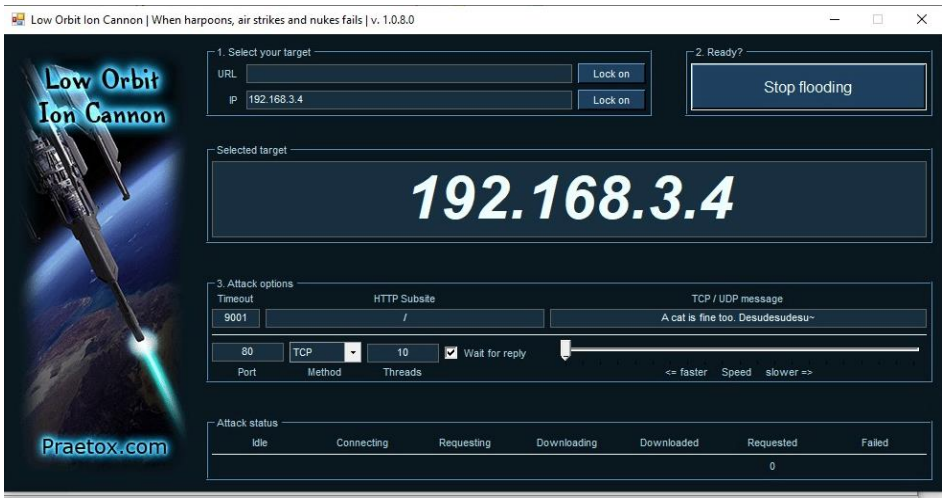
IP : 192.168.3.3

Target :

IP Lokal *Web Server*: 192.168.3.4

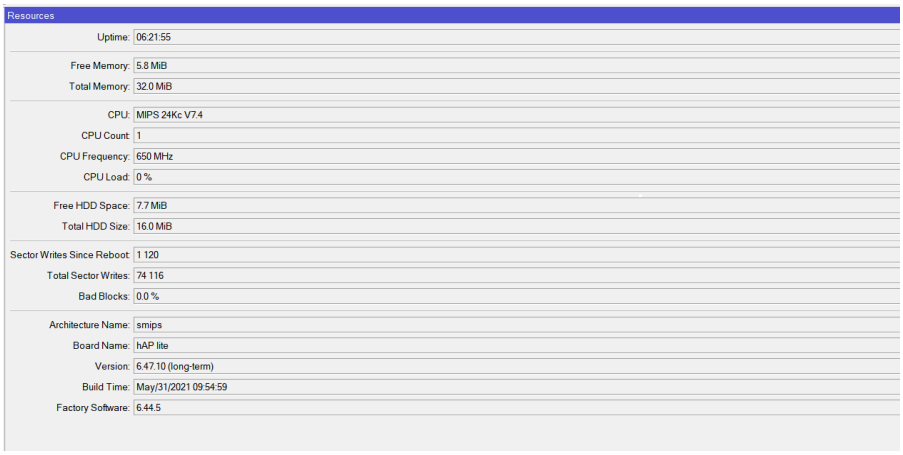
4.2.2.1.1. Serangan SYN flood attack

Pada tahap ini *SYN flood attack* adalah serangan *ddos attack* dengan mengirimkan paket SYN ke target. Dengan keterangan perintah :



Gambar 4. 103 *SYN flood attack* pada LOIC
(Sumber : Data Olahan, 2024)

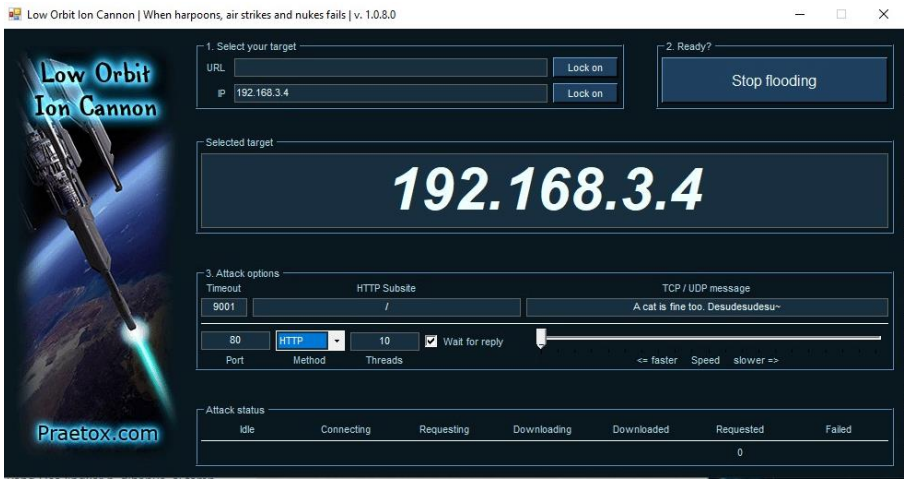
- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *flooding syn* setelah diterapkan *firewall raw* dimana *CPU Load* menjadi 0 %.



Gambar 4. 104 Hasil pada CPU untuk serangan *SYN flood attack* pada *mikrotik*
(Sumber : Data Olahan, 2024)

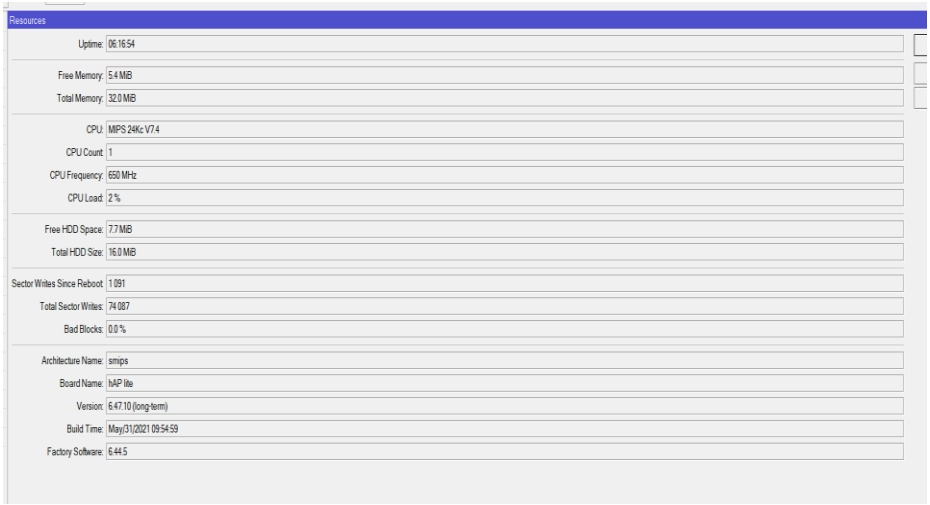
4.2.2.1.2. Serangan *HTTP flood attack*

Pada tahap ini *HTTP flood testing* adalah serangan dengan mengirimkan GET atau POST dalam jumlah besar ke *server* target. Dengan keterangan perintah :



Gambar 4. 105 *HTTP flood attack* pada LOIC
(Sumber : Data Olahan, 2024)

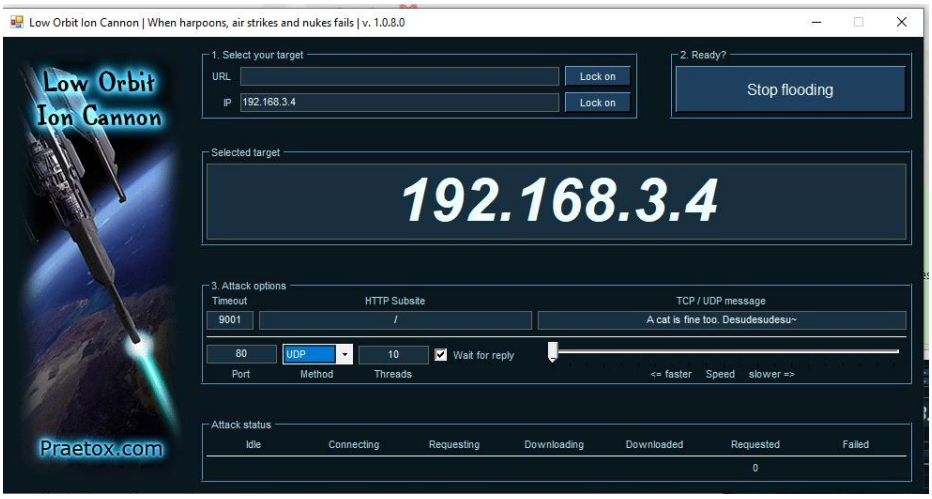
- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *HTTP flood* setelah diterapkan *firewall raw* dimana *CPU Load* turun menjadi 2 %.



Gambar 4. 106 Hasil pada CPU untuk serangan *HTTP flood attack* pada *mikrotik*
(Sumber : Data Olahan, 2024)

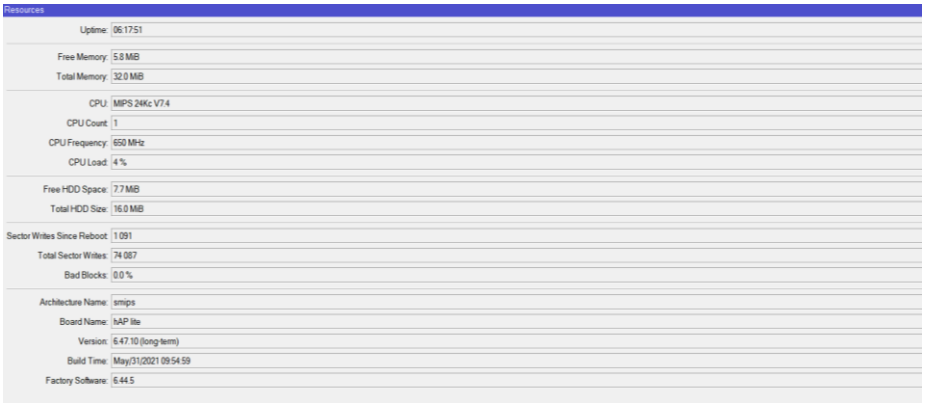
4.2.2.1.3. Serangan *UDP flood attack*

Pada tahap ini *UDP flood attack* adalah jenis serangan penolakan layanan dimana sejumlah besar paket UDP dikirim ke *server* yang diargetkan dengan tujuan melumpuhkan kemampuan *server* dalam memproses dan merespons. Dengan keterangan perintah :



Gambar 4. 107 *UDP flood attack* pada LOIC
(Sumber : Data Olahan, 2024)

- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *UDP flood* setelah diterapkan *firewall raw* dimana *CPU Load* turun menjadi 4 %.



Gambar 4. 108 Hasil pada CPU untuk serangan *UDP flood attack* pada *mikrotik*
(Sumber : Data Olahan, 2024)

Table 4. 10 keamanan *iptables*

Hasil Setelah keamanan *iptables*	Deskripsi
Serangan *SYN flood* tidak berhasil	• Menurunkan penggunaan CPU *Mikrotik* menjadi sekitar 0 % • Dengan aturan *iptables* yang tepat, maka *iptables* akan melakukan *drop* pada IP penyerang dan penyerang tidak bisa melakukan serangan *SYN flood*
Serangan *HTTP flood* tidak berhasil	• Menurunkan penggunaan CPU *Mikrotik* menjadi sekitar 2 % • Dengan aturan *iptables* yang tepat, maka *iptables* akan melakukan *drop* pada IP penyerang dan penyerang tidak bisa melakukan serangan *HTTP flood*
Serangan *UDP flood* tidak berhasil	• Menurunkan penggunaan CPU Mikrotik menjadi sekitar 4 % • Dengan aturan *iptables* yang tepat, maka *iptables* akan melakukan *drop* pada IP penyerang dan penyerang tidak bisa melakukan serangan *UDP flood*

4.2.2.2.*DDOS attack (UDP flood)* sebelum penerapan *firewall raw*

Pada tahap ini *UDP flood attack* adalah jenis serangan penolakan layanan dimana sejumlah besar paket UDP dikirim ke *server* yang diargetkan dengan tujuan melumpuhkan kemampuan *server* dalam memproses dan merespons. Berikut adalah jenis penyerangan DDOS *Attack* yaitu UDP *flood :*

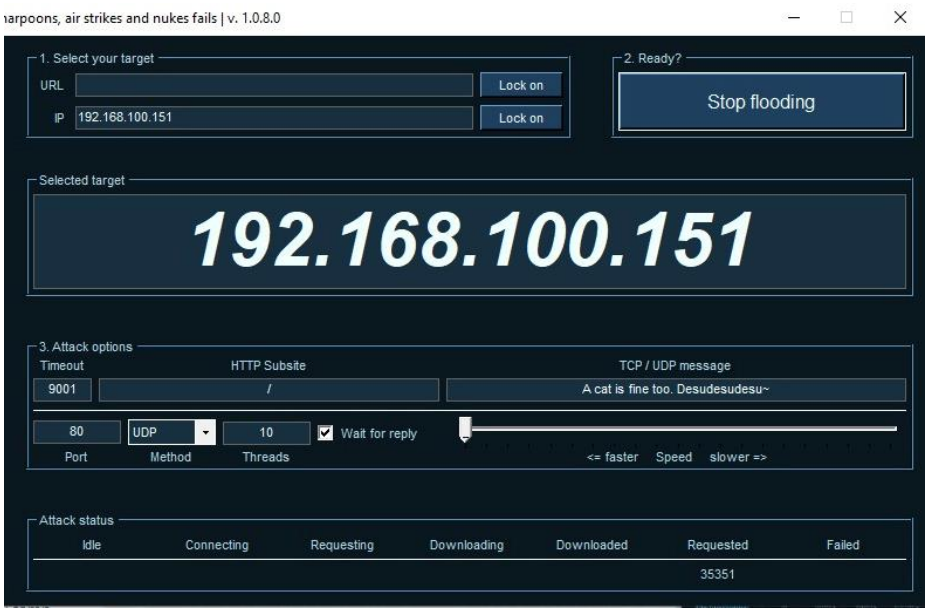
Penyerang :

LOIC (*LOW ORBIT ION CANNON*).

IP : 192.168.3.3.

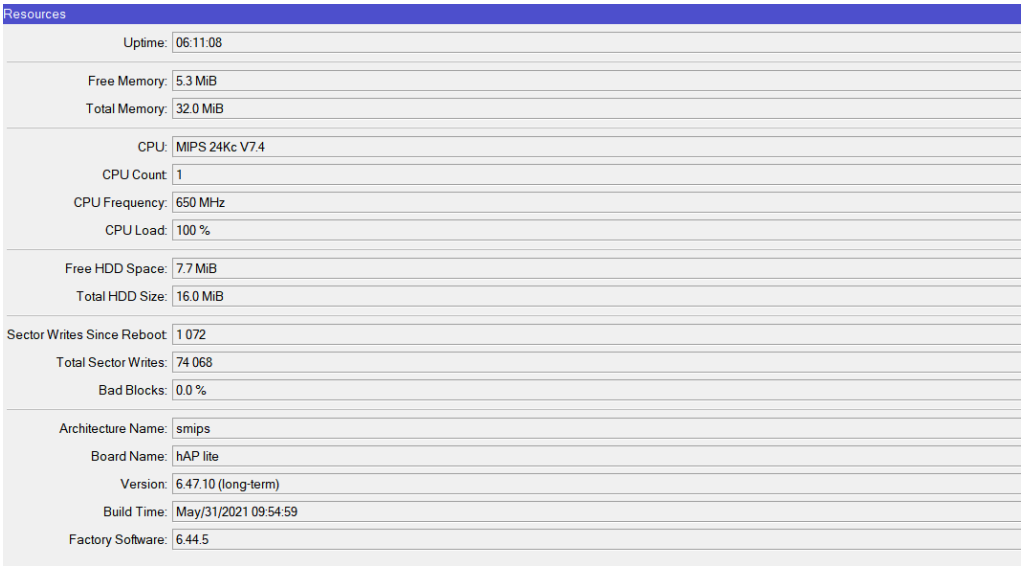
Target

IP Lokal *Web Server*: 192.168.3.4.



Gambar 4. 109 *UDP flood attack* pada LOIC
(Sumber : Data Olahan, 2024)

- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *UDP flood* setelah diterapkan *firewall raw* dimana *CPU Load* Naik menjadi 100 %.



Gambar 4. 110 Hasil pada CPU untuk serangan *UDP flood attack* pada *mikrotik*
(Sumber : Data Olahan, 2024)

Table 4. 11 Firewall raw

Hasil Setelah keamanan *firewall raw*	Deskripsi
Serangan *UDP* *flood* berhasil	• Penggunaan CPU *Mikrotik* naik menjadi sekitar 100 % • *Web server* menjadi lambat (tidak responsif) • *Router mikrotik* menjadi tidak stabil dan membutuhkan *restart*

4.2.2.3.*DDOS Attack (SYN dan HTTP flood)* setelah keamanan *firewall raw*

Pada tahap ini adalah jenis penyerangan DDOS *Attack* yaitu SYN *flood* dan HTTP *flood.*

Penyerang :

LOIC (*LOW ORBIT ION CANNON*).

IP : 192.168.3.3.

Target

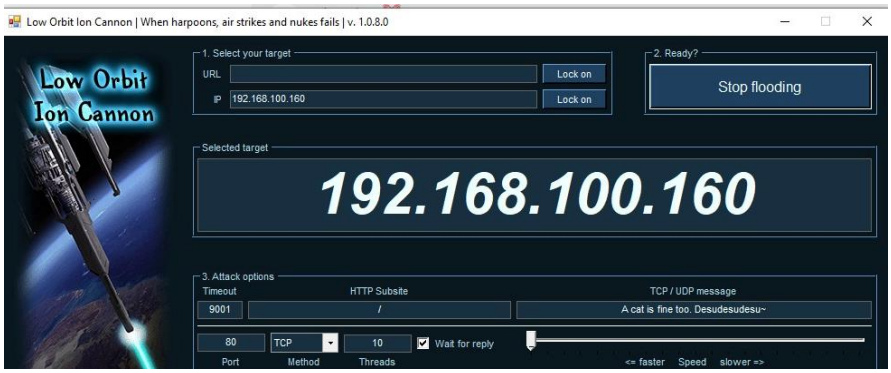
IP *Mikrotik*: 192.168.1.2.

IP ISP pada *Web Server*: 192.168.100.160.

IP *Private Web Server*: 192.168.3.2.

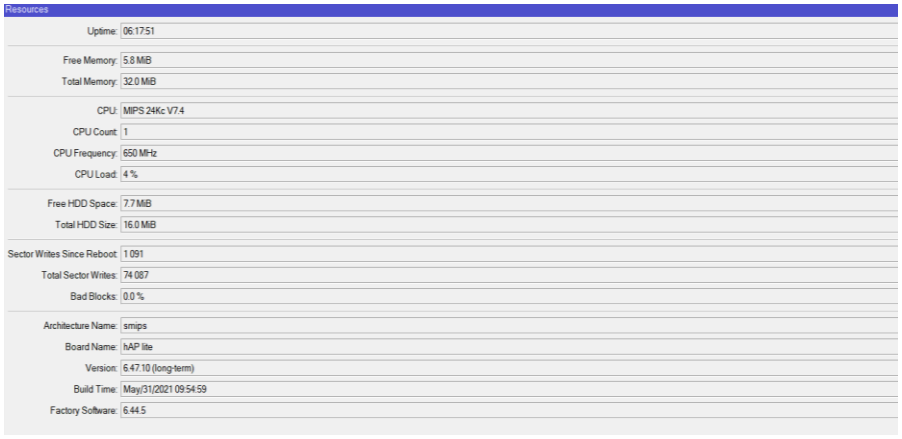
4.2.2.3.1. SYN flood attack setelah penerapan *firewall raw*

Pada tahap ini *SYN flood attack* adalah serangan *ddos attack* dengan mengirimkan paket SYN ke target. Dengan keterangan perintah :



Gambar 4. 111 Serangan *SYN flood*
(Sumber : Data Olahan, 2024)

- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *flooding syn* setelah diterapkan *firewall raw* dimana *CPU Load* menjadi 4 %.



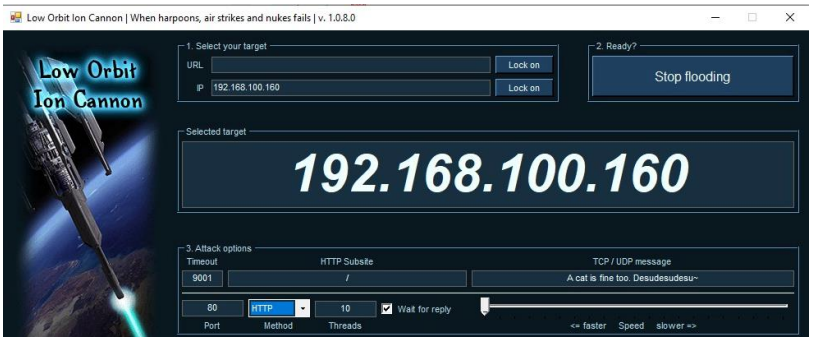
Gambar 4. 112 *Flooding Syn* setelah diterapkan *firewall raw*
(Sumber : Data Olahan, 2024)

Table 4. 12 hasil *firewal raw*

Hasil Setelah keamanan *firewall raw*	Deskripsi
Serangan *SYN flood* tidak berhasil	• Menurunkan penggunaan CPU Mikrotik menjadi sekitar 4 % • Dengan aturan firewall RAW yang tepat, paket yang mencurigakan akan didrop sebelum diproses lebih lanjut oleh firewall, yang dapat mengurangi beban CPU secara signifikan

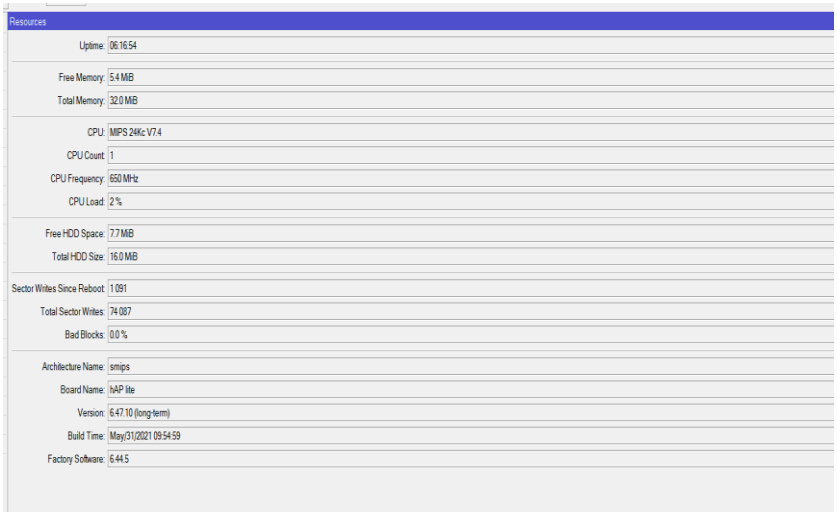
4.2.2.3.2. *HTTP flood attack* setelah penerapan *firewall raw*

Pada tahap ini *HTTP flood testing* adalah serangan dengan mengirimkan GET atau POST dalam jumlah besar ke *server* target. Dengan keterangan perintah :



Gambar 4. 113 Serangan *HTTP flood*
(Sumber : Data Olahan, 2024)

- Maka hasil menampilkan dari proses di *mikrotik* setelah dilakukan *HTTP flood* setelah diterapkan *firewall raw* dimana *CPU Load* turun menjadi 2 %.



Gambar 4. 114 *HTTP flood* setelah diterapkan *firewall raw*
(Sumber : Data Olahan, 2024)

Table 4. 13 firewall raw

Hasil Setelah keamanan *firewall raw*	Deskripsi
Serangan *HTTP flood* tidak berhasil	• Menurunkan penggunaan CPU Mikrotik menjadi sekitar 2 % • Dengan aturan firewall RAW yang tepat, paket yang mencurigakan akan didrop sebelum diproses lebih lanjut oleh firewall, yang dapat mengurangi beban CPU secara signifikan

BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan hasil dari implementasi dan pengujian *Port Knocking* yang telah dilakukan maka dapat disimpulkan :

1 *Port knocking* menggunakan dengan tiga ketukan (ICMP, *Telnet*, dan SSH) pada *MikroTik RouterBoard* dapat membantu menyembunyikan *port* terbuka dari pengguna yang tidak sah dan mengurangi risiko akses tidak diizinkan.

2 Penerapan *firewall raw* pada *MikroTik* untuk melindungi *web server* dari serangan SYN *flood* dan HTTP *flood* memberikan lapisan keamanan tambahan. Pengujian dengan LOIC menunjukkan bahwa langkah-langkah ini efektif dalam mengurangi dampak dari serangan tersebut.

3 Penerapan pada *iptables* pada *Mikrotik* dapat melindungi alamat IP lokal dari *web server* dari serangan alamat IP penyerang jika alamat penyerang diketahui dan dilakukan *drop* menggunakan *iptables*. Maka serangan *DDOS attack* yang dilakukan penyerang bisa teratasi.

5.2. Saran

Berdasarkan dari hasil implementasi dan pengujian *port knocking* menggunakan *Mikrotik Routerboard RB941-2*ND masih memiliki kinerja terbatas sehingga membuat kurang ideal untuk lingkungan yang memilki banyak perangkat terhubung. Kemudian hanya memiliki *4 port ethernet* dan jangkauan sinyal *Wi-Fi* terbatas, tidak memiliki *port* USB. Sehingga saran jika ingin melakukan penguijan pada keamanan *Mikrotik* sebaiknya melakukan pengujian dengan *Mikrotik Routerboard* yang lebih terbaru dengan kinerja, *port ethernet* yang lebih banyak, Jangkauan *Wi-Fi* yang cukup luas dan memiliki *port USB*. Contohnya ***MikroTik RB4011 (RB4011iGS+RM).***

DAFTAR PUSTAKA

Afdhol. P. Y., M. N., Anggraini Samudra, A., & Trisetyowati Untari, R. (2023). Perancangan Jaringan Komputer Menggunakan Metode Failover. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(3), 1474–1481. https://doi.org/10.36040/jati.v7i3.7313

Amarudin, A. (2018). Desain Keamanan Jaringan Pada Mikrotik Router OS Menggunakan Metode Port Knocking. *Jurnal Teknoinfo*, *12*(2), 72. https://doi.org/10.33365/jti.v12i2.121

Anas, M. A., Soepriyanto, Y., & Susilaningsih. (2018). PENGEMBANGAN MULTIMEDIA TUTORIAL TOPOLOGI JARINGAN UNTUK SMK KELAS X TEKNIK KOMPUTER DAN JARINGAN Muchammad Azwar Anas, Yerry Soepriyanto, Susilaningsih. *Multimedia Tutorial*, *1*(4), 307–314.

Blaise, A., Bouet, M., Conan, V., & Secci, S. (2020). *Detection of unknown attacks : an unsupervised port-based approach $*.

Ernawati, R., Ruslianto, I., & Bahri, S. (2022). Implementasi Metode Port Knocking Pada Sistem Keamanan Server Ubuntu Virtual Berbasis Web Monitoring. *Coding : Jurnal Komputer Dan Aplikasi*, *10*(01), 158–169. https://jurnal.untan.ac.id/index.php/jcskommipa/article/view/54226

Insani, P. P., Kanedi, I., & Akbar, A. Al. (2023). *Implementation of Snort as a Wireless Network Security Detection Tool Using Linux Ubuntu Implementasi Snort Sebagai Alat Pendeteksi Keamanan Jaringan Wireless Menggunakan Linux Ubuntu*. *3*(2), 443–458.

Keamanan, O., Komputer, J., Jamalul, A., & Nurdiawan, O. (2022). *METODE KNOCKING PORT BERBASIS MIKROTIK (Studi Kasus : CV . Mitra Indexindo Pratama)*. *6*(2), 560–570.

Mustaqim, T. M. W. (2022). IMPLEMENTASI MANAGEMEN BANDWIDTH MENGGUNAKAN METODE QUEUE TREE DI PT. JAWA POS NATIONAL NETWORK MEDIALINK (Cabang Karimun). *Jurnal TIKAR*, *3*(2), 118–130.

Na, D. E. C., & Hipertensiva, C. (n.d.). (2020) *Intrusion Detection & Prevention System* dan Keamanan Jaringan Pada Mikrotik Router

Novianto, D., Tommy, L., & Setiawan Japriadi, Y. (2021). Implementasi Sistem Keamanan Jaringan Menggunakan Metode Simple Port Knocking Pada Router Berbasis Mikrotik. *Jurnal Komitek*, *1*(2), 407–417. https://doi.org/10.53697/jkomitek.v1i2

Penerbit Unpri Press Tahun Terbit. (2024). Modul Pengajaran Jaringan Komputer.

Putri, I., Agita, A., & Soim, S. (2023). *Implementasi Port Knocking , Port Blocking Pada Keamanan Jaringan Komputer Berbasis Mikrotik*. *6*(3), 125–130.

Rosaly, R., & Prasetyo, A. (2019). Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan. *Https://Www.Nesabamedia.Com*, 2. https://www.nesabamedia.com/pengertian-flowchart/https://www.nesabamedia.com/pengertian-flowchart/

Santoso, N. A., Affandi, K. B., & Kurniawan, R. D. (2022). *Implementasi Keamanan Jaringan Menggunakan Port Knocking Network Security Implementation Using Port Knocking*. *2*(2), 90–95. https://doi.org/10.25008/janitra.v2i2.156

Saputro, A., Saputro, N., & Wijayanto, H. (2020). Metode Demilitarized Zone dan Port Knocking untuk Keamanan Jaringan Komputer. *CyberSecurity Dan Forensik Digital*, *3*(2), 22–27.

Setyowibowo, S., & Moka, N. (2022). Keamanan Jaringan Hotspot Dengan Simple Port Knocking Dan Automated Backup Menggunakan Mikrotik. *Jurnal Ilmiah Komputasi*, *21*(4), 541–552. https://doi.org/10.32409/jikstik.21.4.3109

Suryana, O. (2018). Server dan Web Server. *ResearchGate*, *August*, 14–23.

Yudi mulyanto, M. Julkarnain, & Jabi Afahar, A. (2021). Implementasi Port Knocking Untuk Keamanan Jaringan Smkn 1 Sumbawa Besar. *Jurnal Informatika Teknologi Dan Sains*, *3*(2), 326–335. https://doi.org/10.51401/jinteks.v3i2.1016

(Ernawati et al., 2022)Ernawati, R., Ruslianto, I., & Bahri, S. (2022). Implementasi Metode Port Knocking Pada Sistem Keamanan Server Ubuntu Virtual Berbasis Web Monitoring. *Coding : Jurnal Komputer Dan Aplikasi*, *10*(01), 158–169. https://jurnal.untan.ac.id/index.php/jcskommipa/article/view/54226

LAMPIRAN

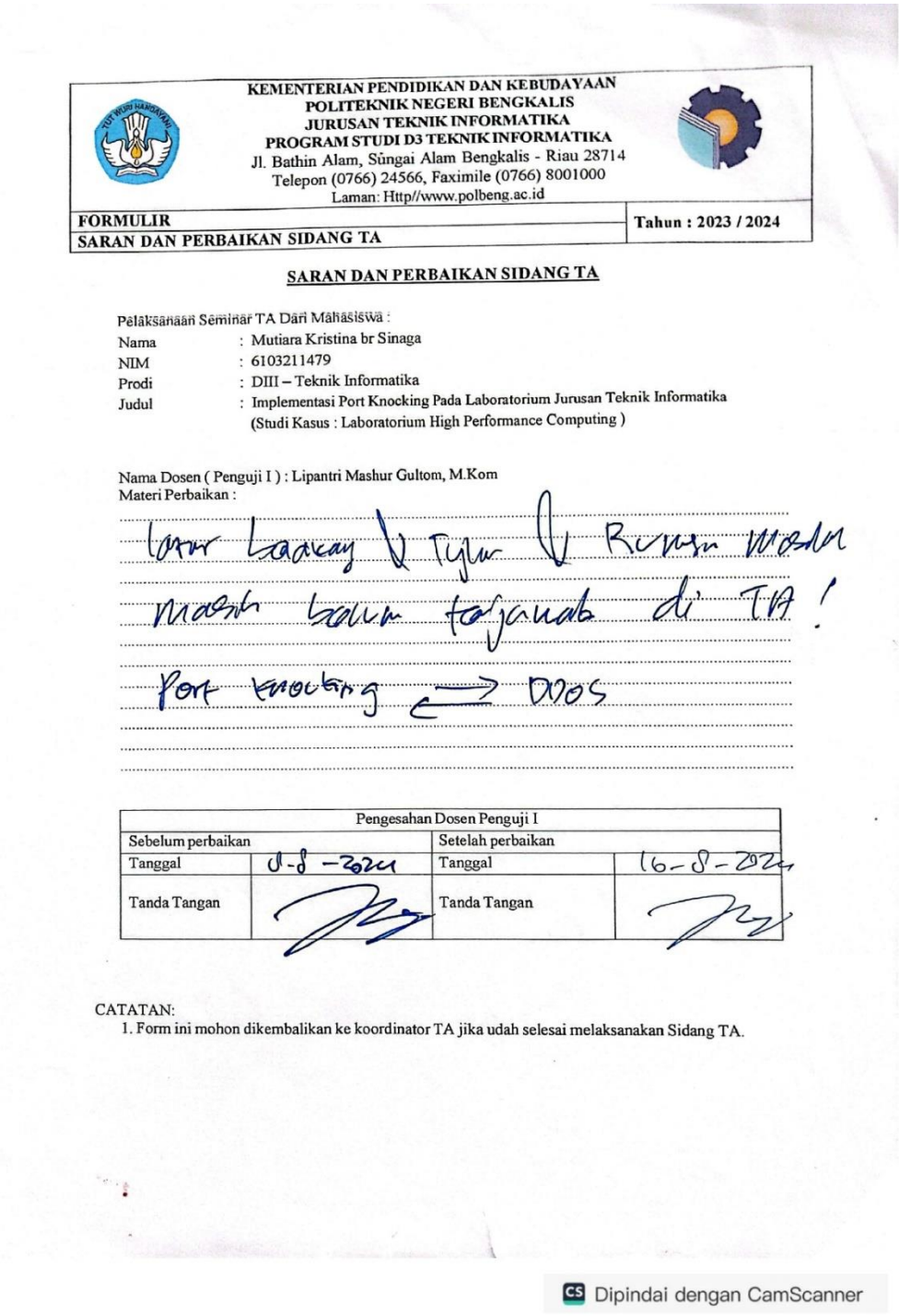
Lampiran 1 Lembar Asistensi Bimbingan

FORM LEMBAR ASISTENSI BIMBINGAN
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
TAHUN AJARAN 2023/2024

Nama : Mutiara Kristina Br Sinaga
NIM : 6103211479
Dosen Pembimbing : Wahyat, M.Kom
Judul TA : Implementasi *Port Knocking* Pada Laboratorium Jurusan Teknik Informatika (Studi Kasus: Laboratorium *High Performance Computing*)

No	Tanggal	Kegiatan	Tanda Tangan Pembimbing
1.	04 Juni 2024	memberikan arahan untuk melanjutkan progress bab 4	
2.	07 Juni 2024	lanjutkan untuk merancang mikrotik (port knocking)	
3	21 Juni 2024	- Diberikan arahan untuk memperbaiki ip address agar terhubung dengan mikrotik	
4.	03 Juli 2024	- Membuat Grup Dosen untuk mahasiswa dengan setiap user - kali-linux DDOS dengan hping port scanning dengan nmap - manajemen bandwidth	
5.	10 Juli 2024	mencoba pengujian port scanning & DDOS	
6.	26 Juli 2024	- Pengujian	
7.	1 Agustus 2024		
8.	6 Agustus 2024	ACC Seminar Tugas Akhir	

Lampiran 2 Saran dan Perbaikan Sidang TA oleh Dosen Penguji 1



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI BENGKALIS
JURUSAN TEKNIK INFORMATIKA
PROGRAM STUDI D3 TEKNIK INFORMATIKA
Jl. Bathin Alam, Sungai Alam Bengkalis - Riau 28714
Telepon (0766) 24566, Faximile (0766) 8001000
Laman: Http//www.polbeng.ac.id

FORMULIR SARAN DAN PERBAIKAN SIDANG TA	Tahun : 2023 / 2024

SARAN DAN PERBAIKAN SIDANG TA

Pelaksanaan Seminar TA Dari Mahasiswa :

Nama : Mutiara Kristina br Sinaga
NIM : 6103211479
Prodi : DIII – Teknik Informatika
Judul : Implementasi Port Knocking Pada Laboratorium Jurusan Teknik Informatika (Studi Kasus : Laboratorium High Performance Computing)

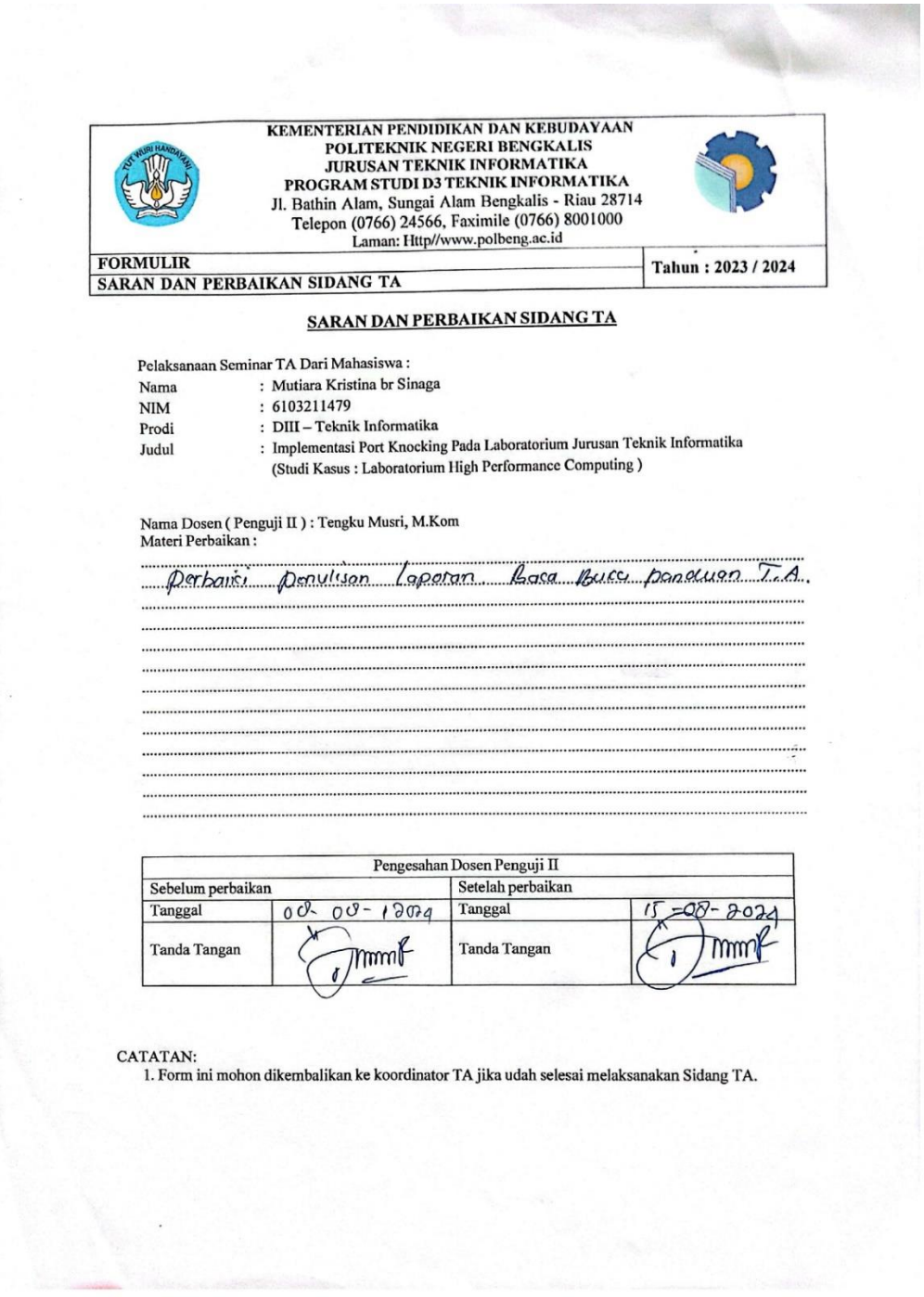
Nama Dosen (Penguji I) : Lipantri Mashur Gultom, M.Kom
Materi Perbaikan :

Latar belakang & Tujuan & Rumusan masalah masih belum terjawab di TA !
Port knocking ⟷ DOS

Pengesahan Dosen Penguji I			
Sebelum perbaikan		Setelah perbaikan	
Tanggal	8-8-2024	Tanggal	16-8-2024
Tanda Tangan		Tanda Tangan	

CATATAN:
1. Form ini mohon dikembalikan ke koordinator TA jika udah selesai melaksanakan Sidang TA.

Lampiran 3 Saran dan Perbaikan Sidang TA oleh Dosen Penguji 2



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI BENGKALIS
JURUSAN TEKNIK INFORMATIKA
PROGRAM STUDI D3 TEKNIK INFORMATIKA
Jl. Bathin Alam, Sungai Alam Bengkalis - Riau 28714
Telepon (0766) 24566, Faximile (0766) 8001000
Laman: Http//www.polbeng.ac.id

FORMULIR SARAN DAN PERBAIKAN SIDANG TA	Tahun : 2023 / 2024

SARAN DAN PERBAIKAN SIDANG TA

Pelaksanaan Seminar TA Dari Mahasiswa :

Nama : Mutiara Kristina br Sinaga
NIM : 6103211479
Prodi : DIII – Teknik Informatika
Judul : Implementasi Port Knocking Pada Laboratorium Jurusan Teknik Informatika (Studi Kasus : Laboratorium High Performance Computing)

Nama Dosen (Penguji II) : Tengku Musri, M.Kom
Materi Perbaikan :

Perbaiki penulisan laporan kasa buku panduan T.A.

Pengesahan Dosen Penguji II			
Sebelum perbaikan		Setelah perbaikan	
Tanggal	08-08-2024	Tanggal	15-08-2024
Tanda Tangan		Tanda Tangan	

CATATAN:
1. Form ini mohon dikembalikan ke koordinator TA jika udah selesai melaksanakan Sidang TA.

Lampiran 4 Saran dan Perbaikan Sidang TA oleh Dosen Penguji 3

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI BENGKALIS
JURUSAN TEKNIK INFORMATIKA
PROGRAM STUDI D3 TEKNIK INFORMATIKA
Jl. Bathin Alam, Sungai Alam Bengkalis - Riau 28714
Telepon (0766) 24566, Faximile (0766) 8001000
Laman: Http//www.polbeng.ac.id

FORMULIR	Tahun : 2023 / 2024
SARAN DAN PERBAIKAN SIDANG TA	

SARAN DAN PERBAIKAN SIDANG TA

Pelaksanaan Seminar TA Dari Mahasiswa :

Nama : Mutiara Kristina br Sinaga
NIM : 6103211479
Prodi : DIII – Teknik Informatika
Judul : Implementasi Port Knocking Pada Laboratorium Jurusan Teknik Informatika (Studi Kasus : Laboratorium High Performance Computing)

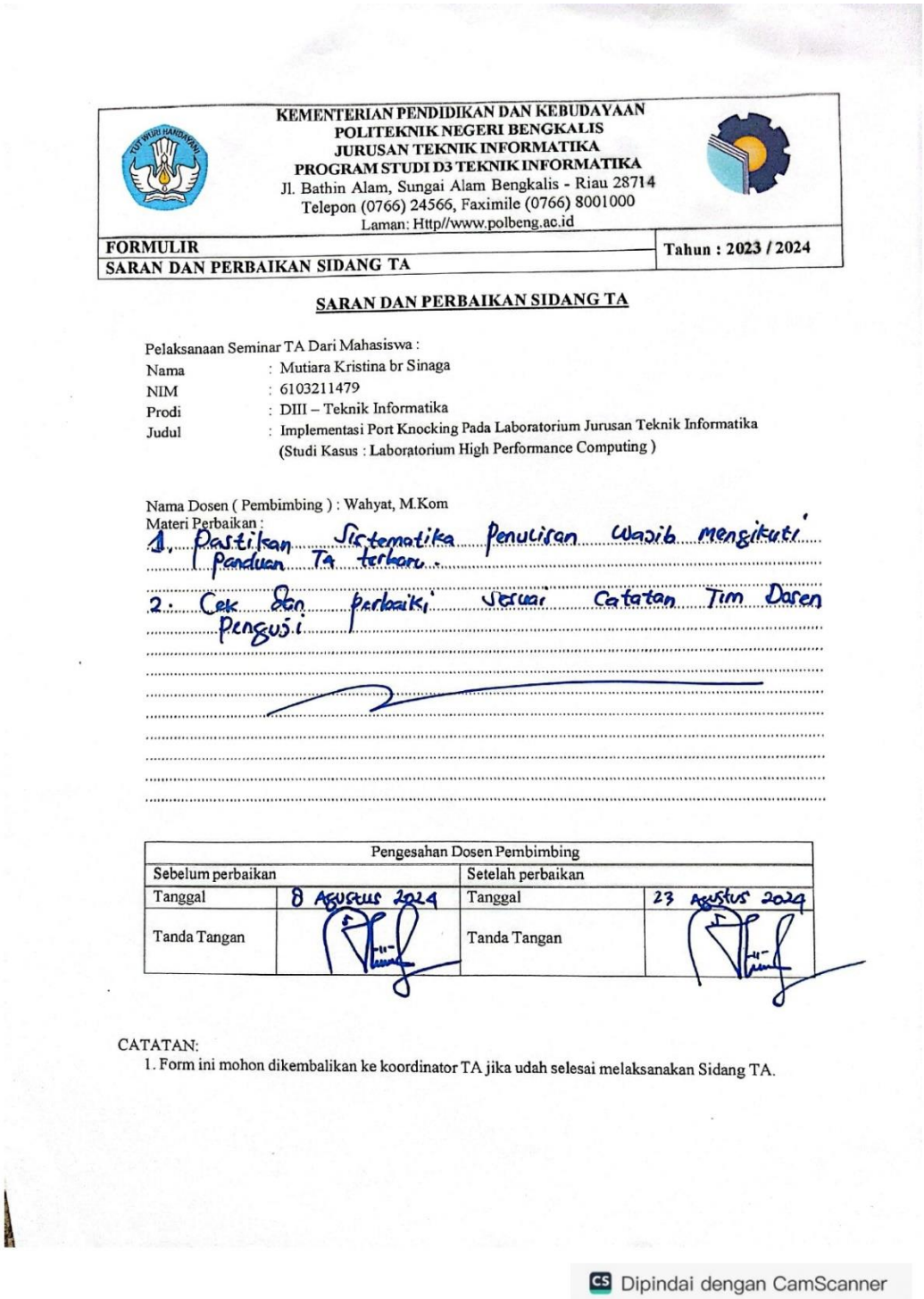
Nama Dosen (Penguji III) : Desi Amirullah, S.Kom., M.T
Materi Perbaikan :

1. Lembar 30 Perbaiki
2. Topologi yg diusulkan di Proyeksi.
3. Pd Bab IV, Gunakan Bahasa Laporan bukan blog tutorial.

Pengesahan Dosen Penguji III			
Sebelum perbaikan		Setelah perbaikan	
Tanggal	8/8/24	Tanggal	16/8/24
Tanda Tangan		Tanda Tangan	

CATATAN:
1. Form ini mohon dikembalikan ke koordinator TA jika udah selesai melaksanakan Sidang TA.

Lampiran 5 Saran dan Perbaikan Sidang TA oleh Dosen Pembimbing



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI BENGKALIS
JURUSAN TEKNIK INFORMATIKA
PROGRAM STUDI D3 TEKNIK INFORMATIKA
Jl. Bathin Alam, Sungai Alam Bengkalis - Riau 28714
Telepon (0766) 24566, Faximile (0766) 8001000
Laman: Http//www.polbeng.ac.id

FORMULIR SARAN DAN PERBAIKAN SIDANG TA	Tahun : 2023 / 2024

SARAN DAN PERBAIKAN SIDANG TA

Pelaksanaan Seminar TA Dari Mahasiswa :

Nama : Mutiara Kristina br Sinaga
NIM : 6103211479
Prodi : DIII – Teknik Informatika
Judul : Implementasi Port Knocking Pada Laboratorium Jurusan Teknik Informatika (Studi Kasus : Laboratorium High Performance Computing)

Nama Dosen (Pembimbing) : Wahyat, M.Kom
Materi Perbaikan :

1. Patuhkan Sistematika penulisan wajib mengikuti panduan TA terbaru
2. Cek dan perbaiki sesuai Catatan Tim Dosen penguji

Pengesahan Dosen Pembimbing			
Sebelum perbaikan		Setelah perbaikan	
Tanggal	8 Agustus 2024	Tanggal	23 Agustus 2024
Tanda Tangan		Tanda Tangan	

CATATAN:
1. Form ini mohon dikembalikan ke koordinator TA jika udah selesai melaksanakan Sidang TA.