IT-EXPLORE
Jurnal Penerapan Teknologi Informasi dan Komunikasi
ISSN 2828-7940, e-ISSN 2829-1727, DOI: 10.24246/itexplore.v4i1.2025.pp106-115
Volume 04 Nomor 01 Tahun 2025
Journal homepage: https://ejournal.uksw.edu/itexplore

Implementasi *port knocking* dinamis berbasis waktu pada router untuk pengamanan akses SSH

David Christian Richardo Junga [1], Wiwin Sulistyo [2]
[1,2] Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Salatiga, Jawa Tengah, Indonesia

Informasi Artikel

Riwayat Artikel:

Dikirim:
11 Desember 2024
Direvisi:
10 Januari 2025
Diterima:
7 Februari 2025

Kata Kunci:

Port Knocking,
Mikrotik,
SSH,
Keamanan Jaringan,
Autentikasi Dinamis

ABSTRAK

Penelitian ini mengusulkan teknik *port knocking* dinamis berbasis waktu pada router Mikrotik sebagai upaya untuk meningkatkan keamanan akses *SSH*. Sistem ini dirancang menggunakan algoritma perhitungan *port* yang berubah secara otomatis sesuai dengan waktu *server*, sehingga setiap sesi akses memerlukan serangkaian *knocking* yang berbeda. Teknik ini melibatkan tiga tahap *knocking*, di mana setiap tahap harus dilewati secara berurutan agar akses *SSH* dapat diberikan. Hasil penelitian menunjukkan bahwa metode ini efektif sebagai mekanisme autentikasi dua lapis yang mampu membatasi akses hanya untuk *client* yang memahami algoritma perhitungan *port* yang digunakan. Selain itu, sistem ini juga dapat mencegah serangan *brute force* secara lebih optimal karena perubahan *port* dilakukan setiap satu menit, sehingga menyulitkan upaya penyerangan oleh pihak yang tidak berwenang. Dengan demikian, metode ini memberikan solusi pengamanan *SSH* yang lebih dinamis, adaptif, serta fleksibel dalam menghadapi berbagai ancaman keamanan siber modern.

Keywords:

Port Knocking,
Mikrotik,
SSH,
Network Security,
Dynamic Authentication

This is an open access article under the CC BY-SA license

ABSTRACT

This research proposes a time-based dynamic port knocking technique on Mikrotik routers to enhance SSH access security. The system uses an algorithm for port calculation that changes according to the server's time, requiring a different knocking sequence for each access session. The technique involves three consecutive knocking stages, where each stage must be passed in order for SSH access to be granted. The results show that this method is effective as a two-factor authentication mechanism, limiting access only to clients who understand the port calculation algorithm. Moreover, this system helps prevent brute-force attacks more efficiently by changing the port every minute, making it more difficult for unauthorized parties to launch an attack. Therefore, this method offers a more dynamic, adaptive, and flexible SSH security solution that can effectively address modern cybersecurity threats.

Corresponding Author:

David Christian Richardo Junga
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro No.52-60, Sidorejo, Salatiga, Jawa Tengah, Indonesia
Email: 672020212@student.uksw.edu

1. PENDAHULUAN

Di tengah pesatnya perkembangan era digital, keamanan informasi menjadi aspek yang sangat krusial untuk diperhatikan. Kemajuan teknologi dan informasi memberikan kesempatan serta tantangan yang mengakibatkan perubahan dalam semua aspek kehidupan, dari individu hingga tingkat global, termasuk dalam ekonomi, politik, sosial, dan keamanan. Perkembangan yang sangat cepat dalam teknologi dan informasi juga berdampak signifikan pada semua bagian kehidupan [1].

Perlindungan jaringan adalah sebuah sistem yang digunakan untuk mencegah potensi ancaman eksternal yang dapat menyebabkan kerusakan pada jaringan serta pencurian data perusahaan dengan memberikan perlindungan atau proteksi [2]. Keamanan jaringan mencakup berbagai Langkah dan praktik yang bertujuan melindungi sistem komputer serta data dari serangan yang berpotensi merusak integritas, kerahasiaan, dan ketersediaannya. Ancaman-ancaman tersebut dapat berupa serangan siber, seperti *Brute force*. Selain itu, keamanan jaringan juga melibatkan pencegahaan akses tanpa izin, pengawasan aktivitas jaringan, serta penerapan kebijakan keamanan yang sesuai [3].

Teknologi jaringan, khususnya dalam konteks jaringan komputer, memerlukan metode yang efektif untuk melindungi data dan sistem dari serangan yang mungkin terjadi. Salah satu metode yang digunakan adalah *port knocking*, metode *port knocking* sudah banyak digunakan dalam pengembangan keamanan jaringan, penggunaan metode *port knocking* cukup terstruktur dimana pengguna harus melakukan serangkaian tindakan yang memungkinkan perangkat menerima tindakan yang sesuai dengan aturan yang sudah ditentukan sebelumnya [4].

Penelitian ini bertujuan untuk menunjukkan bahwa pendekatan metode *port knocking* dapat dikombinasikan dengan metode dinamis yang dapat menciptakan suatu paradigma keamanan yang lebih kokoh dan efektif. Melalui eksperimen dan simulasi yang cermat, diharapkan dapat ditemukan solusi-solusi inovatif yang mampu mengatasi tantangan keamanan yang lebih kompleks dalam konteks keamanan jaringan komputer modern. Oleh karena itu, pada penelitian ini *port knocking* dikembangkan dengan membuat *port knocking* secara dinamis, Dimana *port* akan terus berubah sesuai dengan waktu *server* pada Router dengan menggunakan logika tertentu dalam *script*.

2. KAJIAN PUSTAKA

2.1. Penelitian Terdahulu

Penggunaan metode *port knocking* sebagai Solusi keamanan jaringan komputer, dimulai dengan fokus pada penggunaan *Secure Shell (SSH)* sebagai *protocol* aman untuk akses remote dan peran *Network Address Translation (NAT)* dalam mencegah akses tidak sah [5]. Metode *port knocking* yang dikembangkan menggunakan MikrotikOS bertujuan untuk menutup akses *port* tertentu dan membukanya hanya melalui urutan ketukan *port* yang telah ditentukan sebelumnya, yang efektif melindungi jaringan dari potensi serangan .Penelitian ini menekankan pentingnya keamanan dalam menghadapi ancaman fisik dan logistik, menghasilkan pendekatan inovatif yang memungkinkan administrator jaringan mengontrol akses dengan lebih ketat. Implementasi metode ini tidak hanya meningkatkan keamanan akses langsung ke router, tetapi juga memberikan lapisan perlindungan tambahan melalui mekanisme *login* yang terkontrol [6],[7].

Pada penelitian lain *port knocking* dikembangkan sebagai keamanan data *transfer* pada *File Transfer Protocol (FTP) server* berbasis *Linux Ubuntu Server*. Dalam proses *transfer file*, *FTP* membutuhkan dua koneksi antara *client* dan *server*. Koneksi pertama dibuat oleh *client* ke *port TCP* 21, kehadiran *FTP* membuka celah bagi *Hacker* untuk mengganggu sistem *FTP*. Disini peran *Firewall* dan *port knocking* dibutuhkan, *firewall* digunakan untuk menutup *port* sementara *port knocking* menggunakan autentikasi untuk memberikan hak akses pengguna untuk menggunakan layanan *FTP* [8].

Pengembangan *port knocking* tidak hanya berfokus pada peningkatan keamanan jaringan saja, pada penelitian yang berjudul Analisis Optimalisasi *Port Knocking* Dan *Honeypot* Dengan *Iptables* Pada *Server* Untuk Keamanan Jaringan, penerapan *port knocking* dan *Honeypot* yang dikombinasikan dengan metode *IPTables* dapat meningkatkan keamanan jaringan dan menurunkan grafik penggunaan sumberdaya ketika terjadi penyerangan, pada *server* dengan 1 *CPU* dan 6036624 *KiB (Kibibyte)* memori, penggunaan *CPU* menurun sebesar 36%, dan penggunaan memori turun sebesar 41%, hal ini menunjukkan bahwa *IPTables* dapat dengan efisien menurunkan sumberdaya yang digunakan ketika terjadi penyerangan [9].

Penelitian ini mengembangkan metode *port knocking* yang lebih maju dibandingkan dengan studi sebelumnya, dengan mengintegrasikan aspek matematis dalam sistem keamanan jaringan. Berbeda dengan

penelitian [5],[6],[7] yang menerapkan *port knocking statis*, penelitian [8] yang mengkombinasikan *firewall* untuk *server FTP*, dan penelitian [9] yang meningkatkan *port knocking* menggunakan *Honeypot* serta *IPTables*, penelitian ini mengembangkan sistem *port knocking* dinamis berbasis waktu, dimana *port* berubah secara otomatis mengikuti perhitungan matematis berdasarkan waktu. Aspek matematis dalam sistem ini melibatkan penjumlahan antara *base port* dengan nilai waktu server router yang berjalan secara otomatis. Dengan demikian, *port* yang harus diketuk oleh *client* selalu berubah sesuai dengan perhitungan ini, sehingga hanya *client* yang memahami mekanisme perhitungan dan melakukan *knocking* tepat waktu yang dapat mengakses *Secure Shell (SSH)*, tanpa mengurangi efisiensi pengelolaan.

2.2. Landasan Teori

Router Mikrotik adalah salah satu produk perangkat keras dan perangkat lunak yang dirancang untuk mengelola fungsi router. Perangkat ini dilengkapi dengan sistem operasi khusus yang memungkinkan pengaturan jaringan komputer dengan koneksi internet, baik menggunakan kabel maupun nirkabel *(wireless)* [10]. Sistem operasi tersebut juga dapat dipasang pada komputer konvensional, mengubahnya menjadi Router yang dilengkapi dengan berbagai fitur seperti *routing, firewall*, manajemen *bandwidth*, titik akses nirkabel, dan lain sebagainya [11]. Dengan fitur-fitur yang disediakan, Mikrotik memungkinkan pengguna untuk mengoptimalkan akses internet dalam berbagai skenario, mulai dari lingkungan rumahan hingga lingkungan bisnis yang lebih kompleks.

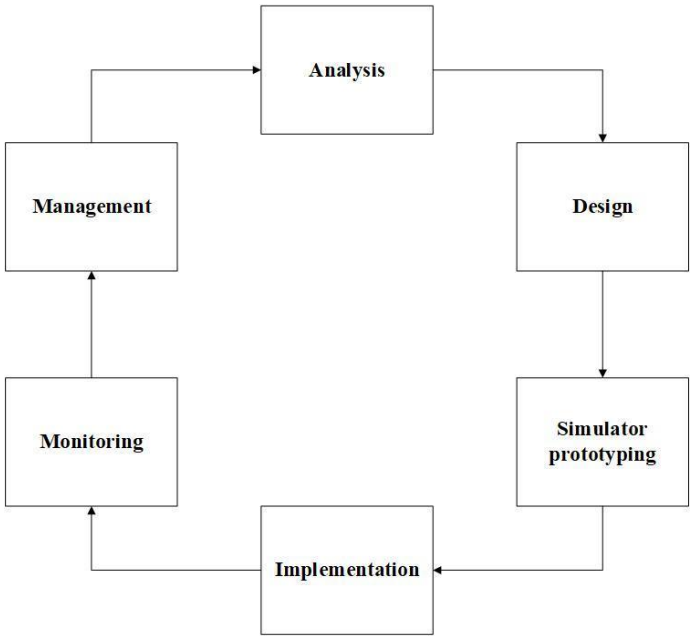
Port knocking merupakan metode yang digunakan untuk meningkatkan keamanan layanan komputer dengan mengharuskan pengguna atau perangkat yang ingin mengakses layanan tertentu melakukan serangkaian tindakan terarah. tindakan ini melibatkan pengiriman paket data ke *port* tertentu pada router atau *firewall* dalam urutan yang telah ditentukan sebelumnya [4].

Keuntungan utama *port knocking* adalah menyembunyikan keberadaan layanan, menambah lapisan keamanan tambahan, dan memungkinkan kontrol akses yang lebih terkontrol. Namun, teknik ini juga memiliki kelemahan, seperti proses *knocking* yang dapat menjadi rumit bagi pengguna awam, kebutuhan akan sinkronisasi waktu yang akurat, dan kemungkinan serangan. Implementasi *port knocking* biasanya menggunakan *script* atau fungsi pada sisi *server* untuk memantau dan mengelola proses *knocking*, dan aplikasi khusus pada sisi *client* untuk melakukan *knocking* sesuai urutan yang ditentukan.

Firewall merupakan sistem keamanan komputer yang berperan penting dalam mengawasi lalu lintas data jaringan, memastikan perlindungan, dan mencegah akses yang tidak sah. Fungsi utamanya adalah melakukan penyaringan *(filtering)* akses internet, menentukan dengan tepat pihak mana yang diperbolehkan atau dilarang melewati jaringan. Dengan kemampuan pemantauan aktivitas jaringan yang canggih, *firewall* bertindak sebagai penjaga gerbang digital yang melindungi infrastruktur teknologi informasi dari potensi ancaman keamanan [12].

3. METODE PENELITIAN

Network Development Life Cycle (NDLC) adalah metode yang digunakan dalam merancang atau mengembangkan infrastruktur jaringan. Metode ini menjelaskan tahapan siklus proses untuk membangun atau meningkatkan sistem jaringan komputer [13],[14]. *Network Development Life Cycle* terdiri dari berbagai elemen yang menjelaskan fase, tahapan, Langkah, atau mekanisme dalam suatu proses tertentu. Istilah “siklus” menjadi kata kunci yang menggambarkan proses berkelanjutan dalam pengembangan sistem jaringan, mencakup seluruh tahapan yang akan diilustrasikan pada gambar 1 [15].



Gambar 1. Metode *Network Development Life Cycle*

Tahapan-tahapan penelitian dengan metode *Network Development Life Cycle* dapat dijelaskan seperti berikut. Proses dimulai dengan tahap analisis. Pada tahap ini, kebutuhan-kebutuhan untuk pengembangan sistem jaringan diidentifikasi dan didefinisikan. Tujuan-tujuan yang ingin dicapai serta kriteria-kriteria yang harus dipenuhi oleh sistem dirumuskan dengan jelas.

Langkah selanjutnya adalah desain. Pada tahap ini, perancangan arsitektur jaringan, *script*, dan aturan *firewall* berdasarkan kebutuhan yang telah dianalisis sebelumnya. Rancangan ini disusun sedemikian rupa agar dapat memenuhi kriteria-kriteria yang telah ditetapkan.

Setelah itu, dilakukan tahap simulasi dan *prototype*. Pada tahap ini, rancangan yang telah dibuat pada tahap sebelumnya diuji dan divalidasi melalui simulasi atau pembuatan *prototype*. Tujuannya adalah untuk memastikan bahwa desain tersebut dapat berfungsi sesuai dengan yang diharapkan.

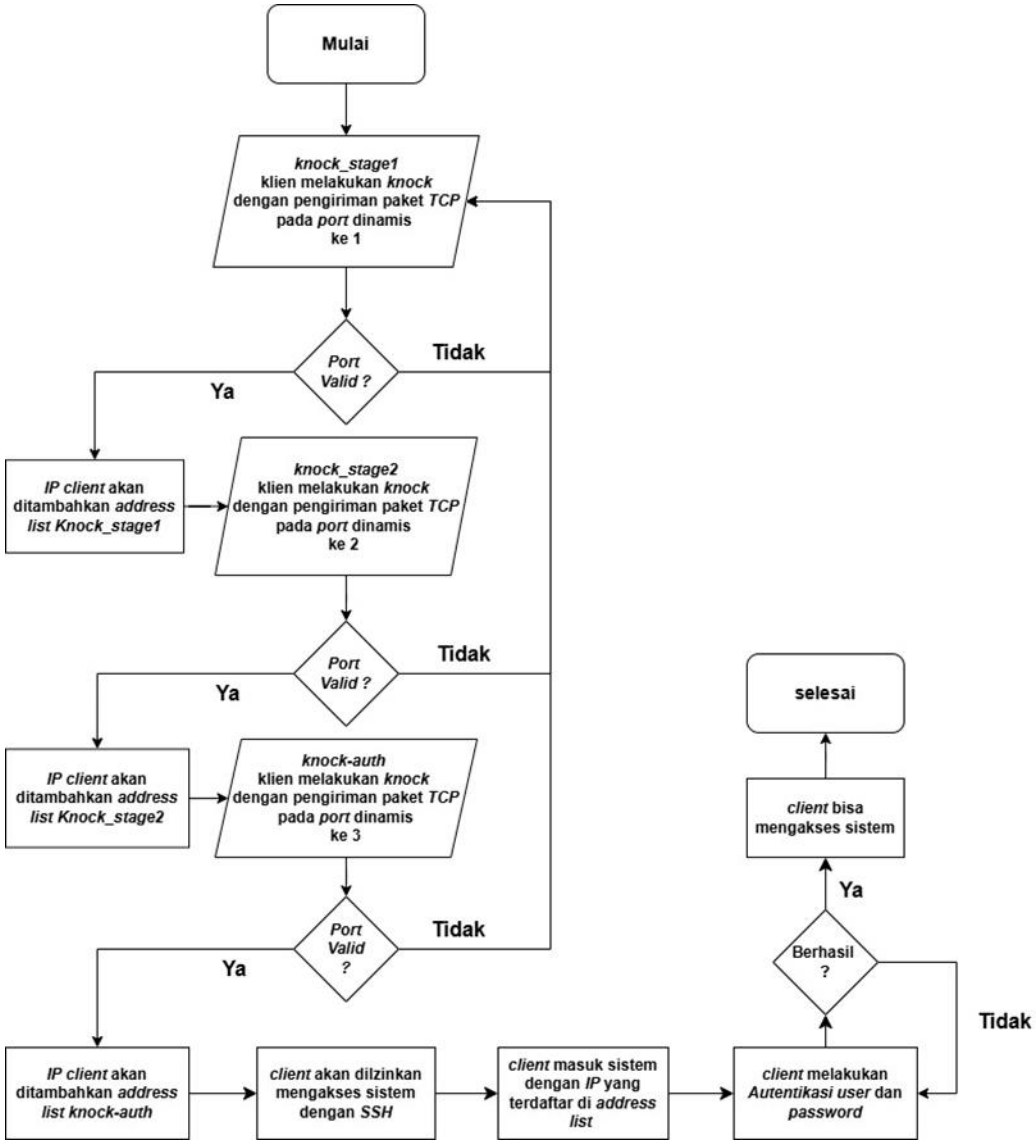
Bila hasil simulasi atau *prototype* sesuai dengan harapan, maka dilanjutkan dengan tahap implementasi. Pada tahap ini, rancangan yang telah disetujui diterapkan pada lingkungan yang sebenarnya. Proses ini memastikan bahwa sistem jaringan dapat berjalan dengan baik di lingkungan nyata.

Pada tahapan *monitoring* sistem yang telah diimplementasikan dipantau secara berkala untuk memastikan kinerjanya tetap optimal. Hasil *monitoring* ini kemudian dapat digunakan untuk melakukan manajemen, yaitu pengolahan dan perbaikan sistem sesuai dengan kebutuhan yang berubah dari waktu ke waktu.

4. HASIL DAN PEMBAHASAN

4.1. Design

Pada bagian ini membahas tentang rancangan alur sistem yang memungkinkan koneksi melalui *port* tertutup dengan memanfaatkan ketukan pada *port* tertentu.



Gambar 2. *Flowchart Port Knocking*

Flowchart pada gambar 2 menggambarkan proses *port knocking* yang digunakan untuk mengakses sistem dan bertujuan untuk meningkatkan keamanan sistem. Proses dimulai dengan *client* yang mengirimkan paket *TCP* ke *port* dinamis pada sistem untuk tahap pertama *knock*. Sistem kemudian memeriksa apakah *port* yang dikirimkan *valid* atau tidak, *port* dapat dikatakan *valid* jika *client* mengirimkan paket ke *port* dinamis yang sesuai dengan perubahan *port* saat itu. Jika *port* tersebut sesuai maka *IP client* yang berhasil melakukan *knock* akan tercatat dalam *address list* dengan nama *Knock-Satge1* dan *client* bisa melanjutkan ke tahap *knock* kedua. sebaliknya, jika tidak *valid* maka proses tidak akan tercatat dalam *address list* dan *client* harus Kembali mengirimkan paket *TCP* pada *port* pertama dengan tepat.

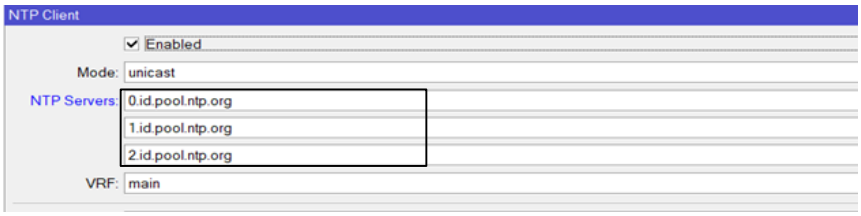
Setelah tahap pertama, *client* melanjutkan dengan mengirimkan paket *TCP* ke *Port* dinamis tahap kedua *Knock*. Sama seperti sebelumnya, sistem akan memeriksa validasi *port* yang dikirimkan. Jika *port* tersebut *valid*, *IP* akan tercatat pada *address list* sebagai *knock-stage2* dan bisa lanjut ke tahap *knock* terakhir. Namun, jika tidak *valid* maka *client* harus Kembali melakukan *knock* pada tahap pertama. Tahap terakhir juga melibatkan pengiriman paket *TCP* ke *port* dinamis yang ditentukan, Dimana sistem kembali melakukan pemeriksaan validasi dan akan ditambahkan ke *address list* dengan nama *knock-auth* jika paket yang dikirimkan *valid*. Dan jika tidak, maka *client* harus melakukan *knocking* dari awal lagi.

Jika semua tahap *knock* berhasil dan *valid*, maka *client* diizinkan untuk mengakses sistem melalui *SSH*. Namun, sebelum akses penuh diberikan, *client* harus melakukan autentikasi dengan memasukkan *username* dan *password*. Jika autentikasi berhasil, akses penuh ke sistem akan diberikan. Jika tidak, *client* memiliki

waktu 1 jam untuk bisa melakukan autentikasi dengan benar. Dengan demikian, sistem *knock-auth* berfungsi sebagai *filter* keamanan yang hanya mengizinkan akses bagi *client* yang mengetahui urutan *knock* yang benar, meningkatkan perlindungan terhadap potensi akses yang tidak sah.

4.2. Implementation

Pada bagian ini membahas tentang bagaimana mengimplementasi rancangan alur *firewall* dan logika untuk metode *knocking* dinamis.



Gambar 3. *Network Time Protocol*

Gambar 3 menunjukkan tampilan *Network Time Protocol (NTP) Client* pada perangkat, pengaturan *NTP* dilakukan pada bagian sistem *NTP Client* router. *NTP Client* digunakan untuk menyinkronkan waktu perangkat dengan *server*, sinkronisasi waktu *server* menjadi elemen yang penting dalam *port knocking* berbasis waktu. Untuk menjamin sinkronisasi waktu yang tepat, Router Mikrotik dikonfigurasi untuk menggunakan *server Network Time Protocol (NTP)*. Sinkronisasi ini memastikan perhitungan *port knocking* tetap konsisten antara *client* dan *server*.

```
:local currentTime [/system clock get time];
:local hour [:pick $currentTime 0 2];
:local minute [:pick $currentTime 3 5];

:local basePort1 10000;
:local basePort2 20000;
:local basePort3 30000;

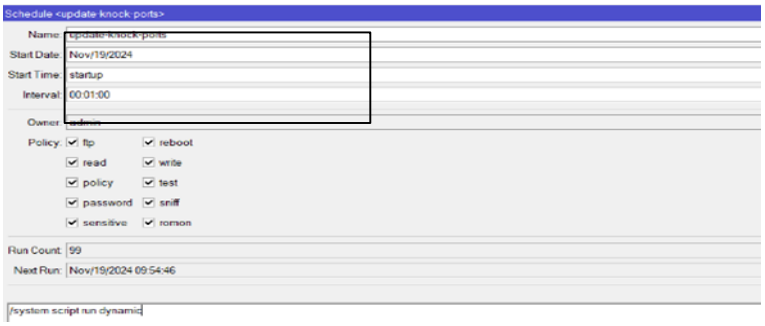
:local port1 ($basePort1 + [:tonum $hour]);
:local port2 ($basePort2 + [:tonum $minute]);
:local port3 ($basePort3 + ([:tonum $hour] * [:tonum $minute]));

# Set global variables
:global knockPort1 $port1;
:global knockPort2 $port2;
:global knockPort3 $port3;

:log info "Port Knocking Updated: $knockPort1, $knockPort2, $knockPort3";
```

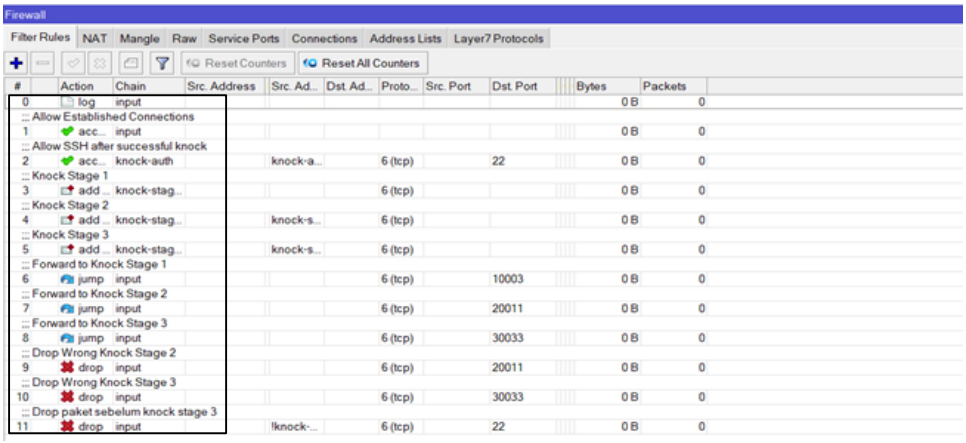
Gambar 4. *Script Port Knocking Dynamic*

Pada gambar 4 implementasi *script dynamic* dilakukan pada bagian sistem *script* Router, *script dynamic* digunakan untuk menghasilkan 3 *port* dinamis berdasarkan waktu, *Port* pertama dihitung dengan menjumlahkan jam saat ini ke *base port* 10000, *port* kedua menggunakan menit ditambah *base port* 20000, dan *port* ketiga berasal dari hasil perkalian jam dan menit ditambah *base port* 30000. Hasil perhitungan ini disimpan dalam variabel global yang dapat diakses oleh sistem *firewall*.



Gambar 5. Otomatisasi *Scheduler*

Pada gambar 5 otomatisasi *script* dilakukan dengan menggunakan Fitur *scheduler*. Dalam konfigurasi *scheduler*, ditambahkan sebuah tugas baru dengan nama *"update-knock-ports"* yang akan dijalankan setiap 1 menit. Dengan mengatur *"startup"*, pengaturan ini akan berjalan saat router dinyalakan. Dengan konfigurasi *scheduler* ini, *script "dynamic-ports"* akan selalu dijalankan setiap menit untuk memperbarui perhitungan *port knocking* sesuai dengan waktu sistem terkini. Hal ini memastikan *port knocking* selalu *up to date* dan dapat digunakan oleh *client* untuk melakukan proses autentikasi.

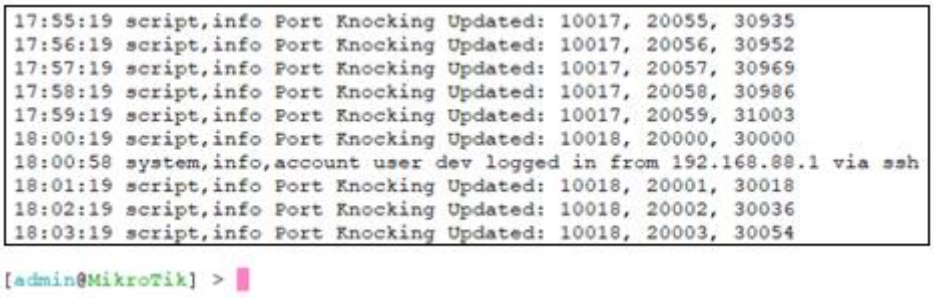


Gambar 6. *Rule Firewall*

Firewall merupakan inti dari sistem keamanan ini. Secara *default*, koneksi *SSH (port 22)* diblokir. Sistem *knocking* sendiri terdiri dari tiga tahapan yang harus dilalui dalam urutan yang benar, dengan batas waktu 1 menit untuk setiap tahap. Jika urutan *knocking* yang dilakukan *client* berhasil. *IP address client* akan ditambahkan ke daftar *"knock-auth"* dengan durasi satu jam. Hanya *IP* dalam daftar ini yang diizinkan untuk mengakses *SSH* pada router.

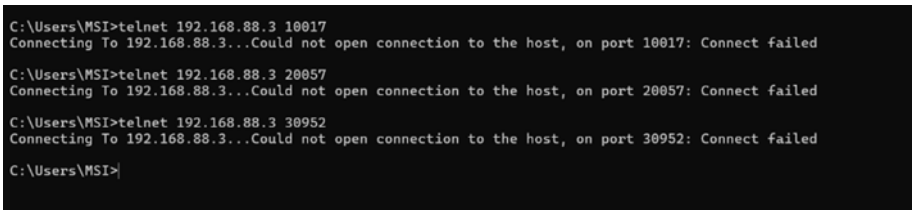
4.3. Pengujian dan Hasil

Pada penelitian ini pengujian dilakukan untuk membuktikan bahwa *port knocking* dinamis pada Router Mikrotik berhasil diimplementasikan, Adapun tahapannya sebagai berikut :



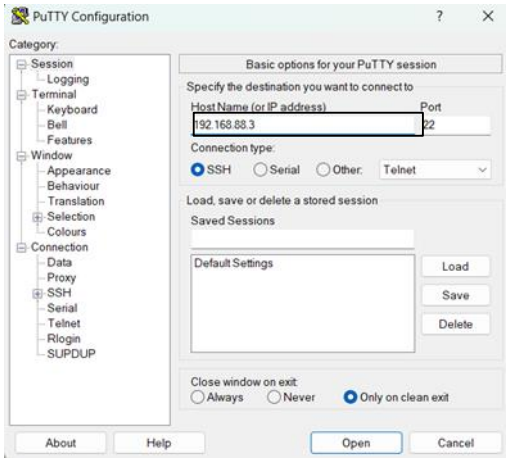
Gambar 7. *Log Print Port*

Pada gambar 7 adalah tahap dimana *log* sistem diperiksa dengan perintah */log print* untuk memastikan bahwa *script port knocking* bekerja sesuai jadwal. *Log* ini digunakan untuk memantau pembaharuan *port* dinamis yang dihasilkan oleh *script dynamic*. Keberhasilan pada tahap ini menunjukkan bahwa mekanisme pembaruan *port* berjalan secara otomatis dan konsisten setiap menit.



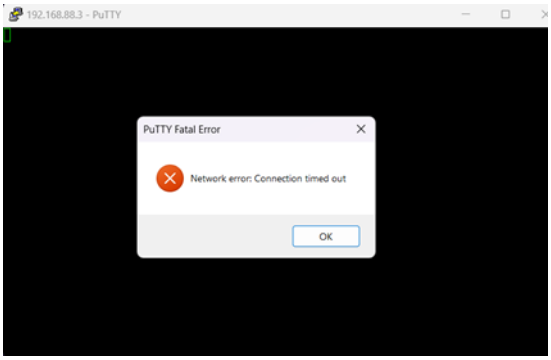
Gambar 8. Uji *Port Knocking*

Selanjutnya pada gambar 8 pengujian *port knocking* menggunakan *telnet*. Yaitu dengan mengirimkan paket *TCP* ketiga *port* dinamis yang telah dihitung berdasarkan waktu *server*. Tes ini bertujuan untuk memastikan urutan *knocking* dapat dikenali oleh sistem *firewall* dan dapat mengotentikasi *client* yang mengikuti aturan *knocking* dengan benar.



Gambar 9. Akses *SSH*

Pada gambar 9 Setelah *knocking*, koneksi *SSH* diuji untuk memverifikasi bahwa hanya *client* yang sah yang mendapatkan akses ke router. Pengujian ini mencakup percobaan akses *SSH* dari *IP* yang telah berhasil melakukan *knocking* dan memastikan *IP* lain yang tidak memenuhi prosedur tetap ditolak oleh sistem.



Gambar 10. Akses *SSH* gagal

Pada gambar 10 diperlihatkan bahwa *client* gagal mengakses sistem router dengan menggunakan *SSH*, ini menunjukkan bahwa *client* gagal melakukan *knocking* ke *port* dinamis yang tepat, *client* harus kembali melakukan *knocking* dengan *port* yang sesuai agar dapat mengakses sistem router dengan *SSH*.



Gambar 11. Akses *SSH* berhasil

Pada gambar 11 *client* berhasil masuk ke dalam sistem router dan mengaksesnya setelah *client* berhasil melakukan *knocking* dan terdaftar pada di sebagai *IP* yang dapat mengakses router lewat *SSH*. Tahapan ini memastikan keamanan sistem bekerja secara optimal sesuai desain dan konfigurasi *firewall* yang telah diterapkan.

Hasil analisis implementasi *port knocking* dinamis berbasis waktu yang telah diterapkan memiliki beberapa keunggulan utama. Pertama, sistem ini menyediakan keamanan berlapis, dimana *port* yang digunakan untuk akses *SSH* berubah secara dinamis berdasarkan waktu sistem. Hanya *client* yang mengetahui formula perhitungan *port* yang dapat melakukan proses *knocking* dengan benar, sehingga meningkatkan keamanan. Sistem juga menyediakan fitur *monitoring* dan *logging*, *IP* yang berhasil melakukan *knocking* akan bisa keluar dan masuk sistem tanpa melakukan *knocking* dalam jangka waktu 1 jam setelah *IP* tercatat pada *address list.*

Sistem ini juga dilengkapi dengan kemampuan audit yang baik, Dimana setiap perubahan *port* dan *IP address* yang berhasil melakukan *knocking* tercatat dalam *log* sistem. Hal ini sangat membantu dalam proses *monitoring* dan *troubleshooting*. Fleksibilitas sistem juga menjadi poin penting di mana formula perhitungan *port*, interval perubahan, dan durasi *timeout* dapat disesuaikan dengan kebutuhan spesifik organisasi. Namun demikian, sistem ini juga memiliki beberapa kekurangan yang perlu diperhatikan. Kompleksitas implementasi menjadi tantangan utama karena membutuhkan sinkronisasi waktu yang akurat antara *server* dan *client*. Ketergantungan pada waktu juga menjadi isu kritis, dimana akurasi perhitungan *port* sangat bergantung pada ketepatan waktu sistem dan ketersediaan *NTP server*.

Potensi masalah keamanan juga perlu diwaspadai, dimana kebocoran formula perhitungan *port* dapat mengkompromikan keseluruhan sistem. Serangan *replay* dalam *window timeout* yang sama masih mungkin dilakukan, dan ada potensi *Denial of Service* dari percobaan *knocking* yang salah. Keterbatasan akses juga menjadi *concern* dimana tidak tersedianya akses darurat jika sistem bermasalah dan sulitnya memberikan akses temporal kepada pihak ketiga.

Dari sisi skalabilitas, sistem ini memiliki keterbatasan dalam penerapannya untuk banyak *user* dengan pola akses berbeda. Manajemen *user* dan hak akses menjadi lebih kompleks dan sulit diintegrasikan dengan sistem otentikasi terpusat. dengan mempertimbangkan semua aspek tersebut, implementasi *port knocking* dinamis berbasis waktu pada MikroTik dapat menjadi solusi keamanan yang efektif jika diterapkan dengan tepat dan didukung dengan pemahaman yang baik tentang kelebihan dan keterbatasannya. Pemilihan implementasi ini harus disesuaikan dengan kebutuhan spesifik organisasi dan kapasitas tim dalam mengelola kompleksitas sistem.

V. SIMPULAN DAN SARAN

Berdasarkan hasil yang didapatkan dari penelitian ini bahwa implementasi *port knocking* dinamis berbasis waktu meningkatkan keamanan akses *SSH* pada router Mikrotik. Sistem yang dirancang menciptakan mekanisme autentikasi dua lapis, Dimana *client* harus mengetahui perubahan sesuai waktu *server* Mikrotik. Penggunaan waktu server sebagai variabel perhitungan *port* memberikan keunggulan keamanan, karena *port knocking* yang berubah setiap menit membuat sistem lebih sulit diserang menggunakan *brute force* atau *packet sniffing*. Selain itu, sistem ini secara efektif membatasi akses *SSH* hanya untuk *client* yang berhasil melakukan *knocking* dengan urutan yang benar, dengan pengaturan waktu akses *(timeout)* yang mencegah potensi serangan berkelanjutan.

Untuk pengembangan lebih lanjut, terdapat beberapa saran guna meningkatkan kinerja dan efektivitas sistem. Pertama, pengembangan aplikasi *client* yang dapat menghitung *knocking* secara otomatis berdasarkan waktu *server*, dengan antarmuka yang ramah. Kedua, optimasi sistem dengan menyempurnakan *script* perhitungan *port* guna mengurangi beban *server*, dan mengambangkan sistem notifikasi *real-time* untuk *administrator* jika terjadi aktivitas mencurigakan.

DAFTAR PUSTAKA

[1] M. Rizki, “POLITEIA: Jurnal Ilmu Politik Perkembangan Sistem Pertahanan/Keamanan Siber Indonesia dalam Menghadapi Tantangan Perkembangan Teknologi dan Informasi.” [Online]. Available: https://talenta.usu.ac.id/politeia diakses Desember 2024.

[2] A. Wijoyo, A. Rosadi, M. Hanafi, and R. Sidik, “Keamanan Jaringan Melindungi Sistem dari Serangan Luar.” *Jurnal Media Publikasi* [Online]. Available: https://journal.mediapublikasi.id/index.php/jriin

[3] T. Brades and Irwansyah, “Seminar Hasil Penelitian Vokasi (SEMHAVOK) PEMANFATAAN METODE PORT KNOCKING DAN BLOCKING,” 2022.

[4] I. Putri, A. Agita, and S. Soim, "Implementasi Port Knocking, Port Blocking Pada Keamanan Jaringan Komputer Berbasis Mikrotik," *Journal Resistor* , vol. 125, [Online]. Available: https://s.id/jurnalresistor

[5] P. D. Oktaviansyah, P. Korespondensi, : Paradika, and D. Oktaviansyah, "https://jurnal.netplg.com/jnca Penerapan Sistem Pengamanan Port pada Mikrotik Menggunakan Metode Port Knocking Penerapan Sistem Pengamanan Port pada Mikrotik Menggunakan Metode Port Knocking," *NetPLG Journal of Network and Computer Applications*, vol. 1, no. 2, 2022.

[6] F. Baso and M. Ardiansyah, "Implementasi Metode Port Knocking pada MikroTik RouterOS untuk Mendukung Keamanan Jaringan," *Computer, Information, Embedded, Network, and Intelligence System*, vol. 1, no. 1, p. 2023, [Online]. Available: https://journal.lontaradigitech.com/SCIENTIST

[7] D. Novianto, L. Tommy, and Y. Setiawan Japriadi, "Implementation of a Network Security System Using the Simple Port Knocking Method on a Mikrotik-Based Router Implementasi Sistem Keamanan Jaringan Menggunakan Metode Simple Port Knocking Pada Router Berbasis Mikrotik," *JURNAL KOMITEK*, vol. 1, no. 2, pp. 407–417, 2021, doi: 10.53697/jkomitek.v1i2.

[8] S. Khadafi, S. Nurmuslimah, and F. K. Anggakusuma, "IMPLEMENTASI FIREWALL DAN PORT KNOCKING SEBAGAI KEAMANAN DATA TRANSFER PADA FTP SERVER BERBASISKAN LINUX UBUNTU SERVER," 2019.

[9] A. Dermawan, Yuhandri, and Sumijan, "Analisis Perbandingan Optimalisasi Port Knocking Dan Honeypot dengan Iptables Pada Server Untuk Keamanan Jaringan," 2024.

[10] B. K. Simpony, "Simple Queue Untuk Manajemen User dan Bandwidth di Jaringan Hotspot Menggunakan Mikrotik," *JURNAL INFORMATIKA*, vol. 8, no. 1, 2021, [Online]. Available: http://ejournal.bsi.ac.id/ejurnal/index.php/ji

[11] P. Aditya, Nursobah, and Supriady, "IMPLEMENTASI JARINGAN PPPOE DAN HOTSPOT SERVER RT/RW NET BERBASIS MIKROTIK DENGAN FITUR MIKHMON DI ADINET SAMARINDA SEBERANG," *Jurnal INFORMATIKA*, vol. 13, no. 1, 2023, doi: 10.46984/inf-wcd.2204.

[12] S. N. Adzimi, H. A. Alfasih, F. N. G. Ramadhan, S. N. Neyman, and A. Setiawan, "Implementasi Konfigurasi Firewall dan Sistem Deteksi Intrusi menggunakan Debian," *Journal of Internet and Software Engineering*, vol. 1, no. 4, p. 12, Jun. 2024, doi: 10.47134/pjise.v1i4.2681.

[13] J. Penerapan Teknologi Informasi dan Komunikasi, B. Isa Mahendra Kusuma Rachman, W. Sulistyo, K. Kunci -Mekanisme Antrian, and M. Jaringan, "IT-EXPLORE ANALISIS PENGARUH IPV4 DAN IPV6 PADA QUEUING MECHANISM DALAM MANAJEMEN JARINGAN," 2024.

[14] A. Dan Optimasi Pada Teknologi Jaringan Wireless Pada Ruangan Laboratorium Dan Kantor Gedung Mangudu, H. Hasniman Harun, U. Yunan KSH, and M. Teguh Kurniawan, "Copyright ⓒ KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)."

[15] A. P. Pratama, L. Sugiyanta, and A. Idrus, "Design And Implementation of Freeradius as A User Manager on The Mikrotik Hotspot Network at PT Indotruck Utama using The NDLC (Network Development Life Cycle) Method," *International Journal of Information System & Technology Akreditasi*, vol. 7, no. 2, pp. 136–143, 2023.