semanTIK, Vol.3, No.1, Jan-Jun 2017, pp. 107-114
ISSN : 2502-8928 (Online)

RANCANG BANGUN APLIKASI UNTUK MEMPERMUDAH KONFIGURASI *PORT KNOCKING* DALAM MIKROTIK

Ayu Abrianingsih*[1], LM.Fid Aksara[2], La Ode Hasnuddin S. Sagala[3]
[*1,2,3]Jurusan Teknik Informatika, Fakultas Teknik Universitas Halu Oleo, Kendari
e-mail: [*1]**ayuningsihlestari@gmail.com**, [2]fid.laode@yahoo.com , [3]hasnuddin.sagala@gmail.com

Abstrak

Port knocking merupakan suatu sistem keamanan yang bertujuan untuk membuka atau menutup akses *block* ke *port* tertentu dengan menggunakan *firewall* pada perangkat jaringan. Sehingga untuk masuk ke *port* tertentu maka *user* harus melakukan pengetukan terlebih dahulu. Konfigurasi *port knocking* dapat disetting didalam *firewall* mikrotik. Dalam melakukan konfigurasi *port knocking* seorang *user* harus membuat beberapa *rule port knocking*, sehingga membutuhkan waktu yang lama bagi *user* untuk melakukan konfigurasi.

Dalam penelitian ini akan diibuat sebuah aplikasi yang dapat digunakan untuk mempermudah dalam melakukan konfigurasi aturan *port knocking*, sehingga seorang *user* tidak perlu lagi melakukan settingan secara manual. Aplikasi ini dirancang dengan pemodelan berorientasi obyek, Aplikasi ini dibangun dengan menggunakan bahasa pemrograman java.

Berdasarkan hasil pengujian, aplikasi ini sudah berjalan dengan baik untuk melakukan konfigurasi *port knocking*, ditandai dengan kemampuanya dalam melakukan konfigurasi dan *knocking* untuk membuka akses pada port yang tertutup. Melaui aplikasi ini, seorang *user* dapat dengan mudah melakukan konfigurasi *port knocking* didalam mikrotik.

Kata kunci— *Port Knocking, User, Java, Mikrotik, Firewall.*

Abstract

Port knocking is a security system that aims to open or close block access to certain ports by using a firewall on the network device. So to get into a particular port then the user must perform tapping beforehand. Configuration port knocking can be set in the proxy firewall. In configuring port knocking a user must make some rule port knocking, so it takes a long time for the user to configure.

In this research are rendered an application that can be used to facilitate the configuration rules of port knocking, so that a user does not need to do the settings manually. This application is designed with object-oriented modeling, application was built using the Java programming language.

Based on test results, this application is already running well to configure port knocking, characterized by their ability to configure and knocking to open access to closed ports. Through this application, a user can easily configure port knocking in the proxy.

Keywords— *Port Knocking, User, Java, Mikrotik, Firewall.*

1. PENDAHULUAN

Perkembangan teknologi jaringan komputer saat ini menunjukkan bahwa sistem keamanan sangatlah penting bagi suatu sistem jaringan komputer. Celah-celah keamanan yang terdapat pada jaringan dapat dilihat oleh orang yang tidak bertanggung jawab dan dapat menjadi ancaman yang patut diperhatikan. Oleh karena itu seorang *administrator* jaringan harus menyediakan keamanan untuk mengakses ke *device* jaringan yang dikelola seperti router, PC *server*, dan lain-lain.

Mikrotik sebagai salah satu *dedicated* router yang mempunyai banyak *service remote login* seperti ssh *service* (22), *telnet* (23), *webfix* (80), *winbox* (8291) merupakan *port-port* yang dapat digunakan seorang *admin*

untuk mengendalikan router tersebut. Hal ini cukup membantu *administrator* dalam mengelolah jaringan, namun sekaligus dapat menjadi ancaman karena orang yang tidak berkepentingan dapat menjadikan *service* tersebut sebagai pintu masuk ke jaringan [1].

Salah satu upaya untuk mengamankan jaringan adalah dengan memanfaatkan *firewall* yang dapat mengatur, *memfilter* serta mengontrol lalu lintas data yang diizinkan untuk mengakses jaringan. Didalam *firewall* semua komunikasi keluar dan masuk dikontrol. Perluasan dari penggunaan *firewall* ini salah satunya dengan menggunakan metode *port knocking* [2].

Port knocking diimplementasikan dengan mengkonfigurasi *firewall* untuk permintaan koneksi dan menentukan apakah *client tersebut* diizinkan untuk mengakses *port* yang telah diblok oleh *firewall* jika telah melakukan urutan ketukan yang benar. Jika jawabannya adalah ya, *firewall* akan membuka *port* yang terkait secara dinamis. Sehingga dengan menggunakan metode ini seorang administrator jaringan tetap dapat mengakses jaringannya meskipun layanan tersebut tertutup dari luar. Jika *client* mengirimkan urutan ketukan yang salah *port* yang dilindungi tidak akan muncul atau terbuka. Dalam melakukan konfigurasi *port knocking* seorang *user* harus membuat beberapa *rule port knocking difirewall* mikrotik, *rule port knocking* dibuat dengan melakukan settingan didalam *firewall*. Penulisan skripsi ini bertujuan untuk Membangun aplikasi untuk mempermudah dalam mengkonfigurasi aturan *port knocking* dimikrotik.

Penelitian yang berjudul "Aplikasi *Remote* Mikrotik Berbasis *Java* untuk Keperluan *Setting Hotspot* Mikrotik", dalam penelitiannya penulis membangun sebuah aplikasi dengan menggunakan bahasa pemrograman java untuk melakukan *setting* pada *hotspot* mikrotik . Hasil dari penelitian ini bahawa aplikasi yang dibangun dapat digunakan untuk mempermudah dalam melakukan *setting hotspot* mikrotik [3].

Penelitian yang berjudul "Implementasi *Simple Port Knocking* pada *Dynamic Routing* (OSPF) Menggunakan simulasi GNS3" dalam penelitian ini penulis mengimplementasikan metode *simple port knocking* menggunakan sebuah jaringan antar sesama *routing* yang dikenal sebagai akses *Dynamic Routing* (OSPF) sebagai *cloud*, dan perancangan jaringannya menggunakan aplikasi GNS3. Hasil dari penelitian ini bahawa pengimlementasian *simple port knocking* menggunakan *Dynamic Routing* (OSPF) dapat melindungi jaringan dari pihak lain yang tidak dipercaya [4].

Penelitian yang berjudul Implementasi *Remote Server* Menggunakan Metode *Port Knockig* dengan *asymmetric encryption*. Dalam penelitian ini penulis mengimplementasikan metode *port knocking* pada proses *autentikasi host* yang akan mengakses *server*. Hasil yang didapatkan dari penelitian ini yaitu bahwa pengimplementasian metode *port knocking* dapat menjadi sebuah *security layer* tambahan pada suatu *host* [5]. Penulis menggunakan referensi tersebut dikarenakan ada kesamaan dengan metode yang digunakan yaitu metode *port knocking*.

2. METODE PENELITIAN

2.1. *Unified Process* (UP)

Unified Process dikembangkan sebagai metodologi yang bersifat *use-case driven*, berpusat pada arsitektur, iteratif dan *incremental* didasarkan pada UML (*Unified Modeling Language*) yaitu bahasa model *standar* untuk desain berorientasi objek. *Unified Process* dikembangkan oleh *Rational software* (yang sekarang bagian dari IBM) tahun 1999. *Unified Process* mendeskripsikan pemberian dan pengelolaan tugas serta tanggung jawab dalam sebuah organisasi pengembang perangkat lunak. *Unified Process* yang dilaksanakan dengan produk-produk bantu yang dibuat oleh *Rational Software. IBM*) disebut *Rational Unified Process* (*RUP*).

1. *Inception*

Tahap *inception*, adalah tahap persiapan. Hal-hal yang perlu ditentukan dalam tahap *inception* ini dalah jadwal kerja, pembentukan tim, dan ruang lingkup perangkat lunak yang akan dikembangkan

2. *Elaboration*

Tahap *elaboration*, adalah tahap perencanaan dimana penekanan dilakukan pada terselesaikannya deskripsi kebutuhan perangkat lunak, analisis dan desain arsitekturserta pembangunan kerangka

dasar aplikasi dan metode pengujiannya.

3. *Construction*

Tahap *construction*, adalah tahap pembangunan yang dilakukan penekanan pada desain teknis, pemrograman dan pengujian perangkat lunak.

4. *Transition*

Tahap *transition*, adalah tahap penerapan dilakukannya uji coba oleh calon pengguna, pelatihan, persiapan pemakaian dan diakhiri dengan pemakaian sistem oleh pengguna.

2.3 *Port knocking*

Port Knocking merupakan suatu sistem keamanan yang bertujuan untuk membuka atau menggunakan *firewall* pada perangkat jaringan menutup akses block ke port tertentu dengan dengan cara mengirimkan paket atau koneksi tertentu. Koneksi bisa berupa protocol TCP,UDP, maupun ICMP. Sehingga untuk masuk dan menggunakan akses ke port tertentu yang telah dibatasi,maka user harus mengetuk terlebih dahulu dengan memasukkan rule yang harus dilakukan terlebih dahulu. Rule yang dimana hanya diketahui oleh pihak administrator jaringan [4].

Dengan kata lain *port knocking* adalah sebuah metode untuk membangun sebuah komunikasi *host-to-host* dengan perangkat komputer yang tidak membuka *port* komunikasi apapun secara bebas. Port knocking diimplementasikan dengan mengkonfigurasi Sebuah program kecil yang disebut daemon guna memonitor log firewall untuk permintaan koneksi dan menentukan apakah klien terdaftar pada alamat IP yang disetujui dan telah melakukan urutan ketukan yang benar. Jika jawabannya adalah ya, firewall akan membuka port yang terkait secara dinamis [6].

2.4 Mikrotik

MikroTik *Router* OS adalah sebuah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk ip *network* dan jaringan *wireless*.

Mikrotik merupakan sistem operasi *linux base* yang dirancang secara khusus untuk keperluan *networking*. Didesain untuk memberikan kemudahan bagi penggunanya. Mikrotik dapat dilihat seperti winbox. Winbox merupakan perangkat lunak untuk me-*remote* mikrotik dalam GUI (*Graphic User Interface*) sehingga *user* dengan mudah dapat mengakses dan mengkonfigurasi *router* sesuai kebutuhan dengan mudah, efektif, dan efisien Selain itu instalasi dapat dilakukan pada *standard* PC (*Personal Computer)* [3].

2.5 *Port Scanning*

Port scanning adalah suatu kegiatan atau aktifitas atau proses untuk mencari dan melihat serta meneliti *port* pada suatu komputer atau perlengkapan dan peralatannya. Tujuan dari kegiatan *port scanning* adalah meneliti kemungkinan-kemungkinan kelemahan dari suatu sistem yang terpasang pada suatu komputer atau perlengkapan dan peralatannya melalui *port* yang terbuka. Ada dua kemungkinan *port* yang berada pada komputer atau perlengkapan dan peralatannya, yaitu karena kesalahan sistem atau *bug* dan ketidak mengertian dari pemilik atau pengguna.

2.6 *Firewall*

Firewall adalah sebuah sistem atau perangkat lunak yang mengizinkan komunikasi aliran lalu lintas jaringan yang dianggap aman untuk dapat dilaluinya dan mencegah lalu lintas jaringan yang sekiranya dianggap tidak aman. Pada dasarnya sebuah *firewall* dipasang pada sebuah *router* yang berjalan pada *gateway* antara jaringan lokal dengan jaringan internet. *Firewall* dan paket pada mikrotik digunakan untuk memilih dan memilah paket yang akan diizinkan (*accept*) dan paket yang tidak diizinkan (*drop*). Ketentuan ini merupakan kebutuhan dari konfigurasi sebuah jaringan tersebut [4].

2.7 Pemrograman *Java*

Java adalah bahasa pemrograman yang bisa dijalankan diberbagai komputer termasuk telepon genggam. Bahasa ini banyak mengadopsi sintaksis yang tedapat pada C dan C++ namun dengan sintaksis model obyek yang lebih sederhana. Aplikasi-aplikasi berbasis *java* umumnya dikompilasi kedalam *p-code(bytecode)* dan dapat dijalankan diberbagai mesin *virtual java* [3].

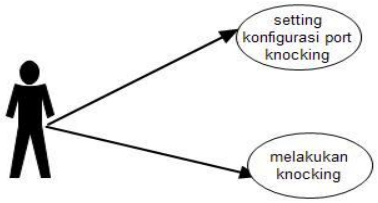
2.8 *Unified Modeling Language* (UML)

UML adalah salah satu standar bahasa yang banyak digunakan di dunia industri

untuk mendefinisikan kebutuhan, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman beorientasi objek UML muncul karena adanya kebutuhan pemodelan *visual* untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari system perangkat lunak.

1. *Use Case Diagram*

Use case diagram yaitu salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor, *use case* diagram juga dapat men-deskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. Gambar 1 menunjukkan *use case* diagram.



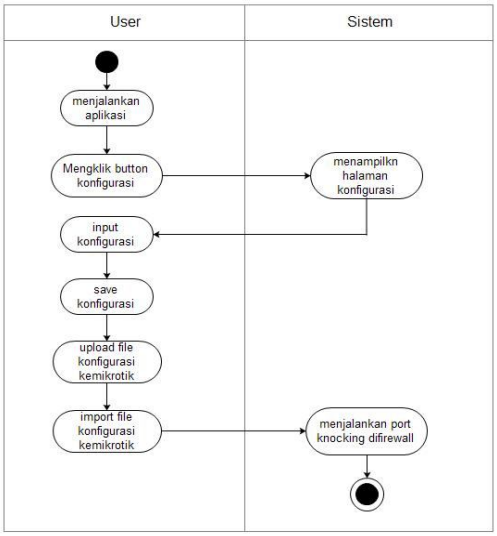
Gambar 1 *Use Case Diagram*

2. *Activity* diagram

Activity diagram atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem.

a. *Activity* diagram konfigurasi

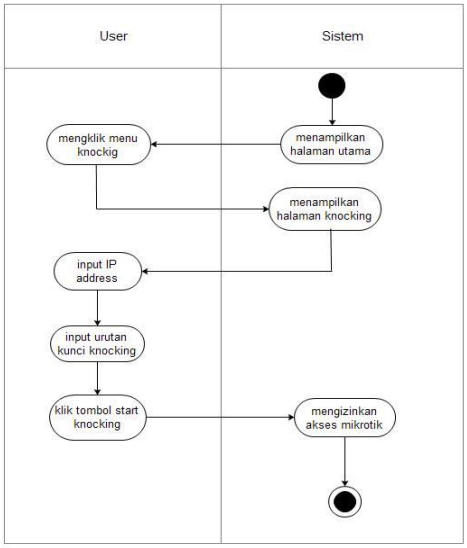
Konfigurasi merupakan suatiu aktivitas untuk menginput konfigurasi *port knocking*. *Activity* diagram konfigurasi ditunjukan pada Gambar 2.



Gambar 2 *Activity* diagram konfigurasi

b. *Activity* diagram *knocking*

Knocking merupakan suatu aktivitas untuk melakukan knocking atau pengetukan kunci. Gambar 3 menunjukkan *Activity* diagram *knocking*.



Gambar 3 *Activity diagram knocking.*

3. HASIL DAN PEMBAHASAN

3.1 Tampilan *Interface*

Sistem ini dikembangkan dengan menggunakan bahasa pemrograman *JAVA*. Adapun antarmuka (*interface*) hasil perancangan aplikasi dapat dilihat pada poin-poin berikut.

1. Tampilan *Form* Menu

Pada *from* ini terdapat dua menu utama yaitu menu *knocking* dan konfigurasi. Gambar 4 menunjukkan tampilan *from* menu.

Gambar 4 Tampilan *From* Menu

2. Tampilan *From konfigurasi*

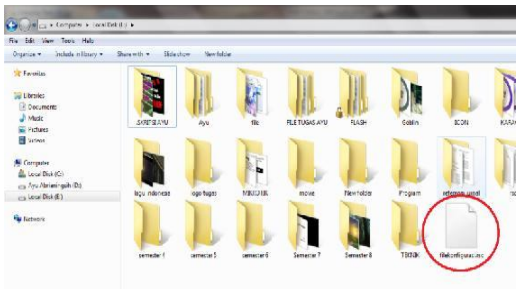
From konfigurasi, merupakan *from* yang harus diisi oleh *user* untuk mengatur konfiguasi *port knocking*. Gambar 5 menunjukkan tampilan from *konfigurasi.*

Gambar 5 Tampilan *From Konfigurasi*

Pada tampilan menu konfigurasi kita dapat menentukan *knocking port*, *knocking* interval, *bloking port* dan *time out*. Pada *knocking port* disini kita dapat menentukan urutan *port* untuk dijadikan *knocking* misalnya urutan *port* yang kita gunkanan yaitu 100,200,300 dan setiap urutan kunci yang dimasukkan harus menggunakan tanda pemisah koma (,).

Selanjutnya memasukkan *knocking* interval yaitu waktu diantara pengiriman paket *knocking* setiap *port*nya, . Bloking *port* untuk menentukan *port* yang akan diblok/di*drop diservice* mikrotik. *Time out* yaitu batas waktu yang diizinkan untuk akses *service* mikrotik setelah berhasil melakukan *knocking*,setelah settingan konfigurasi selesai selanjutnya klik tombol *generate.rsc* untuk menyimpan file konfigurasi di didrive E. Selanjutnya file konfigurasi akan tersimpan di drive E, dengan nama "filekonfigurasi.rsc".

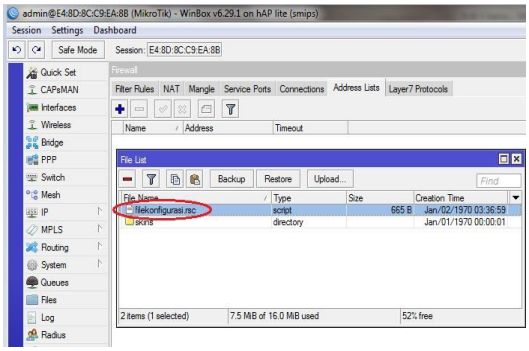
Gambar 6 menunjukkan file yang tesimpan di drive E.



Gambar 6 File yang Tesimpan di Drive E

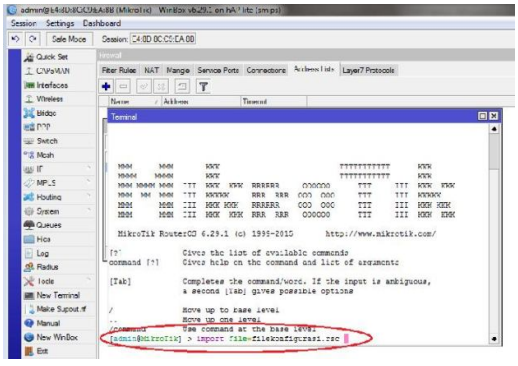
Setelah file konfigurasi tersimpan, maka file tersebut di*upload* ke dalam mikrotik. Untuk mengaploadnya terlebih dahulu membuka winbox lalu pilih menu file dan klik tombol *upload* file pada bagian atas lalu pilih "filekonfigurasi.rsc" dan tekan ok. Maka file konfugurasi tersebut akan tersimpan didalam mikrotik. Pada gambar terlihat bahwa file konfigurasi telah ter*upload* ke mikrotik dengan nama filekonfigurasi.rsc.

Gambar 7 menunjukkan file konfigurasi diupload dimikrtotik.



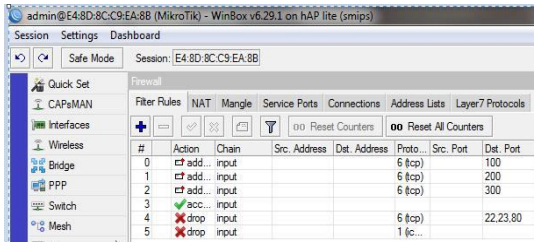
Gambar 7 File Konfigurasi Diupload di Mikrtotik

Setelah file konfigurasi tersebut di*upload* ke mikrotik maka selanjutnya file tersebut akan diim*port* ke dalam terminal mikrotik dengan memilih menu *new* terminal lalu diim*port* dengan memasukkan perintah sebagi berikut : `[admin@mikrotik] < importfile=filekonfigurasi.rsc` lalu tekan enter. Gambar 8 menunjukkan file konfigurasi di*import* di mikrotik.



Gambar 8 File Konfigurasi di*import* di Mikrotik

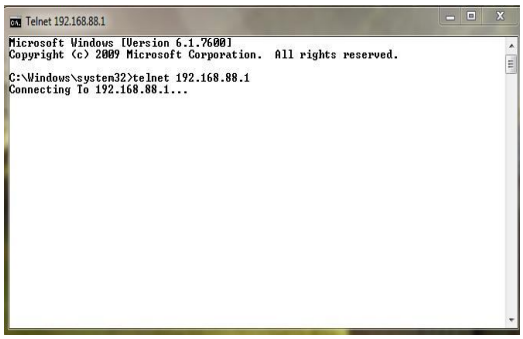
Gambar 9 menunjukkan *rule port knocking* di *firewall.*



Gambar 9 *Firewall Filter Rule* Mikrotik

Selanjutnya dilakukan pengujian untuk mengakses *port* 22 (SSH), 23 (telnet), dan 80 (webfix). Pengujian dilakukan dengan memasukkan urutan ketukan yang sesuai.

Pengujian pertama kita lakukan dengan mecoba mengakses *port* 23 (telnet) sebelum di*knocking*. Masuk pada *command promt* dan ketik perintah telnet 192.168.88.1 yang merupakan IP *Address* mikrotik yang akan diakes. Terlihat bahwa koneksi tidak dapat dibuat karena *port* yang akan kita akses telah diblok oleh *firewall*. Gambar 10 terlihat bahwa telnet tidak dapat diakses.



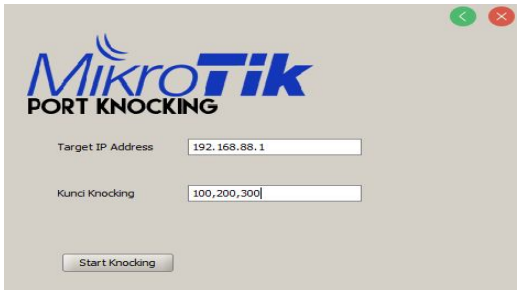
Gambar 10 Telnet Tidak Dapat Diakses

Selanjutnya akan coba mengakses *port* 22 (SSH) sebelum melakukan *knocking*, maka akan terlihat seperti gambar dibawah ini dimana *user* tidak dapat mengakses *port* 22 karena diblokir oleh oleh *firewall*. Gambar 11 menunjukkan bahwa putty (SSH) tidak dapat diakses.

Gambar 11 Putty (SSH) Tidak Dapat Diakses

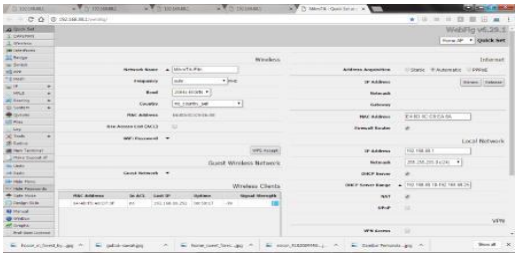
3. Tampilan *From Knocking*

From knocking,merupakan *from* untuk melakukan *knocking*, pada *from* ini *client* harus memasukan target IP address mikrotik yang akan di*knocking*. Selanjutnya masukkan urutan *port* (urutan kunci) untuk *knocking*.Apabila kombinasi urutan kunci yang dimasukan benar sistem mengizinkan koneksi untuk akses mikrotik ke *port* yang diinginkan. Gambar 12 adalah tampilan *from knocking*.



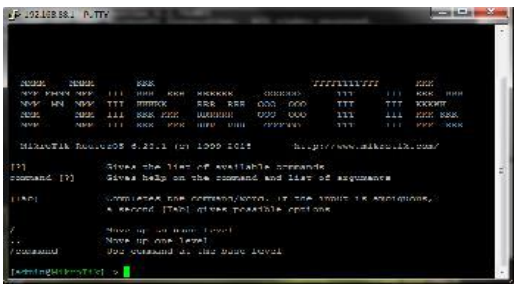
Gambar 12 Tampilan *From Knocking*

Setelah *knocking* berhasil dilakukan maka *port* 22,23, dan 80 yang diblok oleh *firewall* dapat diakses. Pertama kita coba mengakses *port* 80 (webfix) maka hasilnya *port* 80 dapat diakses. Gambar 13 menunjukkan webfix berhasil diakses.



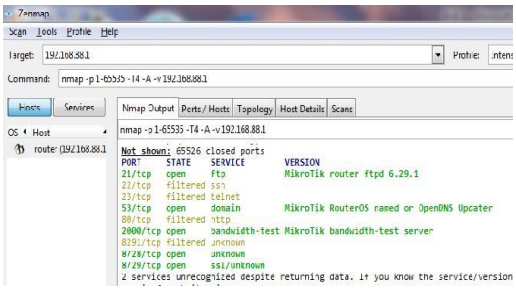
Gambar 13 Webfix Berhasil Diakses

Selanjutnya kita coba mengakses *port* 22 (SSH) maka hasilnya *port* 22 dapat diakses. Gambar 14 menunjukkan SSH berhasil diakses.



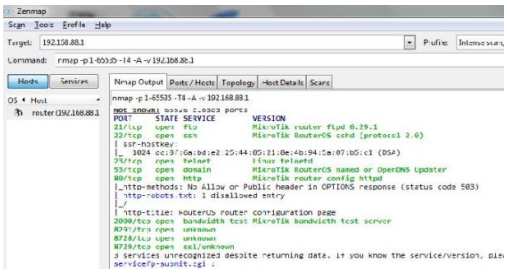
Gambar 14 SSH Berhasil Diakses

Pengujian *port scanning* dengan menggunakan Nmap sebelum melakukan *knocking*. Gambar 15 terlihat bahwa *port* 22,23, dan 80 mempunyai *state filtered*.



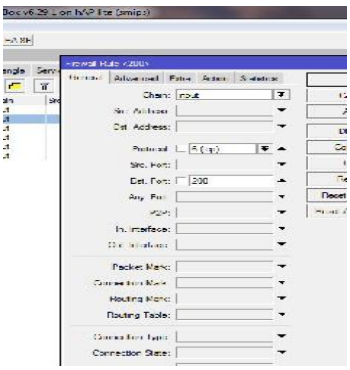
Gambar 15 *Port Scanning* pada Tampilan *Nmap Output* sebelum Melakukan *Knocking*

Pengujian setelah *knocking* Terlihat bahwa port 22,23, dan 80 mempunyai *open*, dapat dilihat pada Gambar 16.



Gambar 16 *Port Scanning* pada Tampilan *Nmap Output* setelah *Knocking* Dilakukan

Port knocking dimikrotik yang dilakukan secara manual di winbox. Gambar 17 menunjukkan settingan *rule port knocking* di winbox.



Gambar 17 Settingan *Rule Port Knocking* di Winbox

Jika *user* melakukan *settingan* langsung ke dalam winbox (tanpa menggunakan aplikasi) maka akan memakan waktu yang cukup lama, dan *user* harus *menyetting* satu per satu setiap *rule port knocking.*

4. KESIMPULAN

Berdasarkan penelitian dan hasil pengujian yang dilakukan pada penelitian ini, maka dapat disimpulkan :

1. Aplikasi *port knocking* yang dibangun dapat mempermudah konfigurasi aturan *port knocking* dan sekaligus dapat melakukan *knocking* untuk membuka koneksi dengan *port* yang tertutup.
2. Pengujian menunjukan bahawa konfigurasi *port knocking* dengan menggunakan aplikasi lebih mudah dibanding tanpa menggunakan aplikasi.

5. SARAN

Saran penulis untuk penelitian selanjutnya yang berkaitan dengan penelitian ini yaitu dengan mencoba mengembangkan sebuah aplikasi *port knocking* yang dapat melakukan import konfigurasi *port knocking* secara langsung kemikrotik.

DAFTAR PUSTAKA

[1] Prihanto, Agus. 2013. "*Implementasi Port Knocking di Mikrotik dengan Menggunakan Komponen Delphi TcpClient*". Universitas Negeri Surabaya. Surabaya.

[2] Rozi M. Fahru, Muslim Royyana , Anggoro Radityo. 2010. Implementasi Remote Server Menggunakan Metode Port Knocking dengan Asymmetric Encryption. Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember.

[3] Palimirmanto, Juventus Robbing . 2014. *Aplikasi remote Mikrotik Berbasis Java Untuk Keperluan Setting Hotspot.* Program Studi Teknik Informatika, Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma, Yogyakarta.

[4] Kusuma, Aprianto Puji Adi. 2016. *Implementasi Simple Port Knocking pada Dynamic Routing (OSPF) menggunakan Simulasi GNS3*, Manajemen Informatika, Fakultas Teknik, Universitas Negeri Surabaya.

[5] Muzawi, Rometdo. 2016. *Aplikasi Pengendalian Port dengan Utilitas Port Knocking untuk Optimalisasi Sistem Keamanan Jaringan Konputer*. Jurusan Manajemen Informatika, STIMIK Amik Riau. Diakses pada tanggal 8 juni 2016.

[6] Haryanto, Edi. 2013. "*Meningkatkan Keamanan Port SSH dengan Metode Port Knocking Menggunakan Shorewall Pada Sistem Operasi Linux"*. Amikom Yogyakarta. Yogyakarta.