

Perancangan dan Implementasi Sistem Keamanan Jaringan Komputer Menggunakan Metode Port Knocking Pada Sistem Operasi Linux

Imam Marzuki

Program Studi Teknik Elektro, Fakultas Teknik

Universitas Panca Marga Probolinggo

Jl. Yos Sudarso, Pabean, Dringu, Kab. Probolinggo 67271

imam@upm.ac.id

Abstract—Keamanan suatu server merupakan salah satu hal yang penting bagi seorang administrator. Tentunya seorang administrator adalah orang yang berhak untuk mengakses server. Apabila ada attacker yang mengambil alih posisi administrator, maka sistem dikatakan tidak aman. Pada penelitian ini difokuskan pada sistem keamanan pada server dari serangan attacker yang ingin merubah dan merusak suatu server. Studi kasus pada penelitian ini adalah ssh server yang berjalan di port 22. Hal ini dikarenakan layanan ssh yang paling banyak diincar oleh attacker. Metode keamanan yang digunakan dalam penelitian ini adalah port knocking. Dengan menggunakan metode port knocking seorang administrator dapat meningkatkan keamanan suatu server dari berbagai serangan yang ditujukan untuk layanan server. Cara kerja dari metode ini adalah server akan menerima percobaan koneksi dari client menuju port yang sudah ditentukan setelah itu firewall akan mendeteksi percobaan tersebut dan mengizinkan client untuk mengakses server. Setelah client selesai mengakses server firewall akan menutup kembali akses ke server sehingga server tidak bisa di akses kembali. Dalam penelitian ini server berhasil melindungi layanan yang ada dengan megintegrasikan aturan firewall dengan program port knocking yang digunakan. Selain itu tanpa mengirimkan ketukan yang tepat, client tidak dapat menggunakan layanan pada server.

Kata kunci : server, metode port knocking, administrator, keamanan

I. PENDAHULUAN

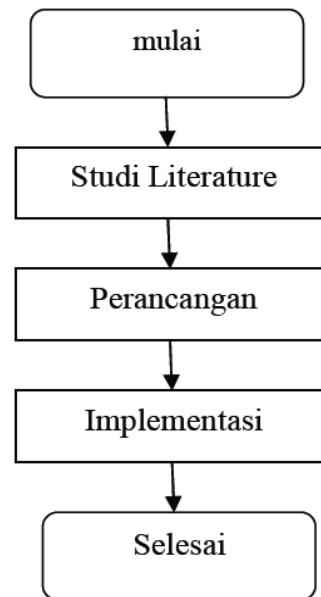
Kemajuan teknologi di bidang jaringan telah memungkinkan untuk melakukan interaksi melalui komputer. Suatu jaringan komputer biasanya terdiri dari server dan client. Server dikendalikan oleh seorang administrator. Salah satunya dengan melakukan remote server. Administrator yang meremote suatu server haruslah orang yang berhak untuk mengakses server tersebut. Namun ada juga attacker yang dengan sengaja masuk kedalam sistem dan kemudian melakukan perubahan serta pengrusakan terhadap server.

Salah satu upaya yang dilakukan untuk meningkatkan keamanan sebuah server adalah dengan menggunakan firewall. Tetapi saat ini masih memiliki kelemahan. firewall tidak mampu membedakan user yang dapat dipercaya. Firewall hanya mampu membedakan alamat IP yang diasumsikan digunakan oleh orang yang tidak dapat dipercaya. Sehingga dicari solusi untuk mengurangi kelemahan yang ada. metode ini salah satunya adalah dengan menggunakan metode port knocking.

II. METODE PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian ditunjukkan dengan flowchart pada gambar 1.



Gambar 1. Flowchart tahapan penelitian

2. 1.1 Studi Literatur

Pencarian referensi dan sumber-sumber yang dapat digunakan sebagai acuan dalam pembuatan penelitian ini

serta teori-teori dasar lain mengenai perancangan dan implementasi metode *port knocking* pada linux ubuntu.

2.1.2 Perancangan

Merancang suatu *system* yang mampu melakukan koneksi melalui port yang tertutup menggunakan metode *port knocking*.

2.1.3 Implementasi

Bagaimana mengimplementasikan metode *port knocking* untuk dapa mengamankan layanan *server* (studi kasus ssh server).

2. 2 Konfigurasi IPTABLES

Pada *IPTABLES* di Linux sudah terinstall secara *default* dimana *IPTABLES* merupakan *firewall* di sistem operasi linux. Dalam konsep *port knocking*, *IPTABLES* difungsikan untuk menutup atau *DROP* semua akses yang menuju port 22. Untuk perintah yang dilakukan untuk *DROP* akses port 22 dengan menulis perintah *IPTABLES -A INPUT -p tcp -dport 22 -j DROP* dan untuk melihat aturan tersebut sudah tersimpan dengan perintah *IPTABLES -L* pada *terminal console* dan dapat dilihat hasilnya seperti pada gambar 2.

```
root@rofiq-275E4E-275E5E:~# iptables -A INPUT -p tcp --dport 22 -j DROP
root@rofiq-275E4E-275E5E:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
DROP      tcp  --  anywhere             anywhere            tcp dpt:ssh

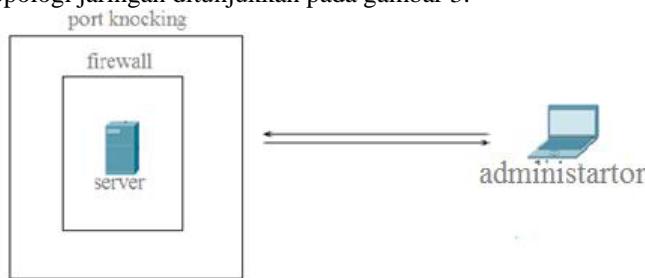
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Gambar 2. Drop Port 22 IPTABLES

2.3 Diagram Topologi Jaringan

Dalam perancangan Diagram Topologi jaringan penelitian ini menggunakan dua buah komputer, satu laptop sebagai *administrator client server* yaitu komputer yang akan mengakses atau *meremote server* melalui port 22 *server* menggunakan *system* operasi linux ubuntu. Diagram topologi jaringan ditunjukkan pada gambar 3.



Gambar 3. Diagram topologi jaringan

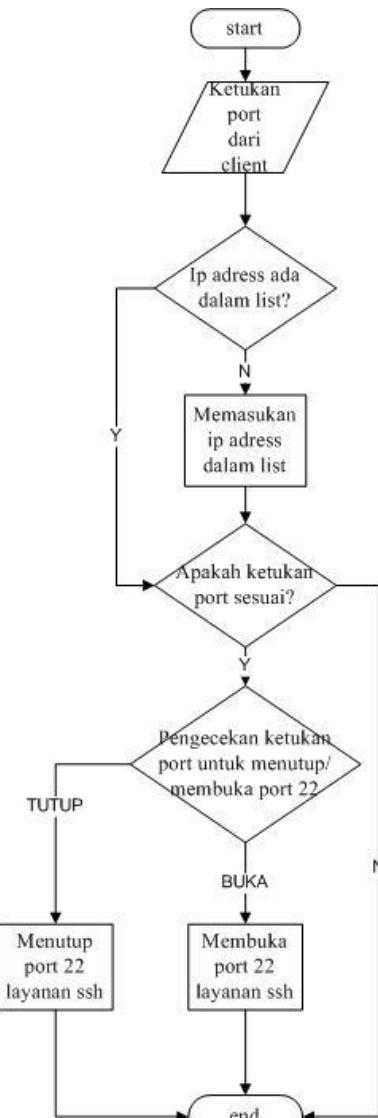
Semua koneksi ke server akan ditutup oleh firewall, Dikarernakan ditutupnya semua koneksi administrator tidak bisa melakukan koneksi ke server. Jika administrator ingin meremote server administrator harus melakukan knocking ke server. Jika knocking yang

dilakukan administrator sesuai dengan konfigurasi yang sudah di atur sebelumnya di server maka firewall akan mengizinkan administrator tersebut melakukan koneksi server menuju ke port 22.

2.3 Perancangan Sistem

2.3.1 Flowchart Aplikasi Port Knocking

Cara kerja aplikasi port knocking ditunjukkan dengan flowchart pada gambar 4.



Gambar 4. Flowchart aplikasi port knocking

Gambaran alur proses dari flowchart *aplikasi port knocking* dijabarkan sebagai berikut:

1. *Client* melakukan ketukan port untuk membuka port 22
2. Apakah IP Address *client* berada dalam *list firewall* jika tidak memasukan IP Address *client* dalam *list*.

3. Jika ketukan port dari *client* tidak sesuai maka *client* tidak diijinkan tetapi jika ketukan port sesuai maka *client* diijinkan.
4. Setelah itu ketukan port dari *client* akan di cek oleh *firewall*, apakah ketukan tersebut membuka atau menutup port 22.

2.3.2 Context Diagram

CD memperlihatkan sistem yang di rancang secara keseluruhan, semua *external entity* harus digambarkan sedemikian rupa, sehingga terlihat data yang mengalir pada input-proses output.

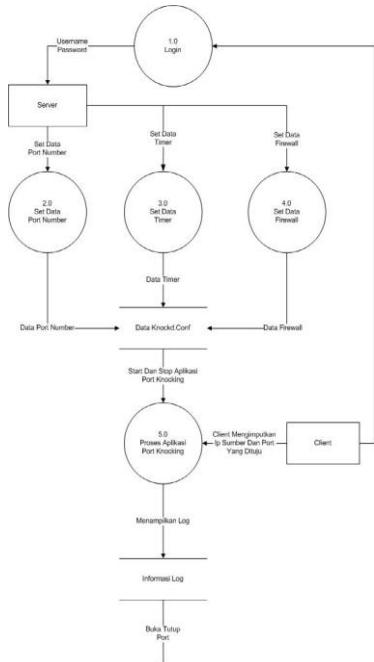


Gambar 5. Context Diagram

Gambar 5 menjelaskan proses *client* dan *server* dalam menjalankan atau menggunakan *aplikasi port knocking*. Di mana server menjalankan perintah untuk mengaktifkan *aplikasi port knocking*, sedangkan *client* menjalankan perintah untuk mengakses port yang digunakan.

2.3.3 Context Diagram

Data Flow Diagram (DFD) memberikan gambaran komponen-komponen dari sebuah sistem beserta aliran data. *Data Flow Diagram* (DFD) dapat dilihat pada gambar 6.



Gambar 7. Data Flow Diagram

Gambar 7 menjelaskan proses aplikasi *port knocking*. *Server* terlebih dahulu melakukan login kedalam sistem., kemudian *server* melakukan set data *port number*, *timer*, *firewall* didalam file *knockd.conf*. setelah data diset *server* menyimpan aplikasi *port knocking* dan menjalankan atau memberhentikan aplikasi *port knocking*. *Client* melakukan login kedalam sistem agar dapat menggunakan *port* yang ingin dituju. *Client* menjalankan perintah berdasarkan IP sumber dan *Port Number* untuk mengakses *port* yang digunakan dan *Logs* akan menampilkan informasi siapa yang masuk ke *port* yang dituju dan siapa yang menutup *port* yang dituju kedalam *server*.

2.3.4 Format Ketukan Port Knocking

Format ketukan yang digunakan dalam perancangan sistem adalah format port tunggal dengan pemetaan tetap, dan hanya menggunakan tiga port ketukan sebagai tujuan pengiriman paket data untuk melakukan ketukan.

Untuk mempermudah penentuan port ketukan, maka dibuat aturan pemilihan port ketukan sesuai dengan nomor port tujuan, digit terakhir pada nomor port ketukan merujuk pada nomor port tujuan. Sebagai contoh, seorang user ingin mengakses port 22, dengan *range* port ketukan yang telah ditentukan yaitu antara port 300 sampai dengan 600, maka pemilihan port ketukan yang digunakan adalah seperti pada Gambar 8.

Nomor Port Ketukan	Port Tujuan
300+a,400+b,600+c	a,b,c

Gambar 8. Penentuan format ketukan

Nomor port 300+a, 400+b, 600+c merupakan nomor port tujuan pengiriman paket data yang berfungsi sebagai port ketukan. Nomor port ketukan menunjukkan port tujuan abc yang akan dibuka atau ditutup. Maka ketukan yang dilakukan oleh pengguna jika ingin membuka 22 adalah seperti pada gambar 9.

300,400,600

Gambar 9. Format ketukan membuka port 22

Sedangkan jika user ingin menutup port 22, maka ketukan yang dilakukan oleh client adalah seperti pada gambar 10.

600,400,300

Gambar 10. Format ketukan menutup port 22

III. HASIL DAN PEMBAHASAN

3.1 Instalasi dan Konfigurasi Program

Sebelum program *port knocking* di jalankan, pertama kali dilakukan instalasi program pada komputer *server* yang berfungsi untuk mendengarkan ketukan port dari komputer *client*. Program *port knocking* yang di gunakan pada penelitian ini menggunakan aplikasi *knockd* yang ada pada setiap *repository* linux. Untuk menginstal aplikasi *knockd* harus terhubung ke internet tetapi dalam penelitian ini penulis menggunakan repository local dan hanya menghubungkan *server* ke *repository* tersebut, setelah terhubung instalasi aplikasi menggunakan “*apt-get install knockd*”. Perintah yang dilakukan pada instalasi ini ditunjukkan pada gambar 11.

```
rofiq@rofiq-275E4E-275E5E:~$ sudo bash
[sudo] password for rofiq:
root@rofiq-275E4E-275E5E:~# apt-get install knockd
```

Gambar 11. Perintah instalasi knockd

Setelah instalasi selesai maka akan tercipta file *knockd.conf*. Selanjutnya adalah mengisi file *knockd* yang berada pada directory *etc*, Dengan beberapa baris text konfigurasi dengan menggunakan *gedit* sebagai medianya. Perintah yang dilakukan ditunjukkan pada gambar 12.

```
rofiq@rofiq-275E4E-275E5E:~$ sudo bash
[sudo] password for rofiq:
root@rofiq-275E4E-275E5E:~# gedit /etc/knockd.conf
```

Gambar 12. Perintah knockd.conf

Setelah perintah diatas di jalankan maka akan menampilkan isi dari directory *knockd.conf*. beberapa baris text harus di masukan kedalam *knockd.conf* hasil perintah tersebut dapat dilihat pada gambar 13.

```
[options]
    logfile = /var/log/knockd.log

[openSSH]
    sequence   =
    seq_timeout =
    command    =
    tcpflags   =

[closeSSH]
    sequence   =
    seq_timeout =
    command    =
    tcpflags   =
```

Gambar 13. File knockd kosong

Keterangan gambar 13 :

- Bagian option yang merupakan bagian yang berfungsi untuk menunjukan letak file log yang berfungsi untuk mencatat semua aktifitas port knocking.
- Seq_timeout pada [openSSH] berfungsi untuk menentukan batas waktu yang digunakan untuk melakukan pengiriman urutan port untuk membuka port tujuan dan Seq_timeout pada [closeSSH] berfungsi untuk menentukan batas waktu yang digunakan untuk melakukan pengiriman urutan port untuk menutup port tujuan.
- Command pada [openSSH] berfungsi untuk menentukan perintah *IPTABLES* yang dijalankan, yaitu perintah untuk membuka port tujuan jika terjadi pengiriman urutan port yang benar pada komputer *server* dan Command pada [closeSSH] berfungsi untuk menentukan perintah *IPTABLES* yang dijalankan, yaitu perintah untuk menutup port tujuan jika terjadi pengiriman urutan port pada komputer *server*.
- Tcpflag pada [openSSH] dan [closeSSH] sama-sama untuk menunjukan header paket yang dikirimkan sebagai urutan port

Untuk menentukan port ketukan, Maka di tambahkan baris perintah pada file *knockd.conf*, Seperti ditunjukkan pada gambar 14.

```
*knockd.conf x
[options]
    logfile = /var/log/knockd.log
    interface = wlan0

[openSSH]
    sequence = 300,400,600
```

```
[closeSSH]
    sequence = 600,400,300
```

Gambar 14. Format ketukan

Pada bagian buka SSH di tentukan tiga nomor port ketukan, yaitu *port 300*, *port 400*, dan *port 600*. Jika terjadi ketukan *port* pada ke tiga port tersebut secara berurutan, Maka *server* akan membuka *port* tujuan.

Pada bagian tutup SSH di tentukan tiga nomor port ketukan, yaitu *port 600*, *port 400*, dan *port 300*. Jika terjadi ketukan *port* pada ke tiga port tersebut secara berurutan, Maka *server* akan menutup *port* tujuan.

Untuk menentukan perintah *IPTABLES* yang akan di jalankan jika terjadi ketukan port, maka ditambahkan baris perintah pada file *knockd.conf*. seperti ditunjukkan pada gambar 15.

```

knockd.conf x
[options]
logfile = /var/log/knockd.log
interface = wlan0

[openSSH]
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

[closeSSH]
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

```

Gambar 15. Konfigurasi IPTABLES

Pada bagian buka SSH, Adalah perintah yang bertujuan untuk mengubah aturan pada *IPTABLES*, yaitu dengan membuka akses terhadap alamat IP komputer *client* agar dapat mengakses *port 22* pada komputer *server*.

Pada bagian close SSH, Adalah perintah yang bertujuan untuk mengubah aturan pada *IPTABLES*, yaitu dengan menutup kembali akses terhadap alamat IP komputer *client* sehingga tidak dapat lagi mengakses *port 22* pada komputer *server*.

Setelah konfigurasi selesai gambaran keseluruhan dari file knockd.conf dapat dilihat pada gambar 16.

```

knockd.conf x
[options]
logfile = /var/log/knockd.log
interface = wlan0

[openSSH]
sequence = 300,400,600
seq_timeout = 3
command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn

[closeSSH]
sequence = 600,400,300
seq_timeout = 3
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn

```

Gambar 16. File knockd.conf setelah dikonfigurasi

Setelah semua konfigurasi selesai, Jalankan aplikasi port knocking di *server* dengan mengetikkan perintah *knockd -D*. Seperti di tujuunjukan pada gambar 17.

```

root@roflq-275E4E-275E5E:~$ sudo bash
[sudo] password for roflq:
root@roflq-275E4E-275E5E:~# knockd -D
config: new section: 'options'
config: log file: /var/log/knockd.log
config: interface: wlan0
config: new section: 'openSSH'
config: openSSH: sequence: 300:tcp,400:tcp,600:tcp
config: openSSH: seq_timeout: 3
config: openSSH: start_command: /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
config: tcp flag: SYN
config: new section: 'closeSSH'
config: closeSSH: sequence: 600:tcp,400:tcp,300:tcp
config: closeSSH: seq_timeout: 3
config: closeSSH: start_command: /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
config: tcp flag: SYN
ethernet interface detected
could not get IP address for wlan0

```

Gambar 17. Aplikasi knockd

Gambar 17 adalah tampilan aplikasi knockd setelah di aktifkan tampilan gambar tersebut mengacu pada isi file knockd.conf dan menggunakan *wlan0* sebagai interfacesnya.

Untuk melihat Ip address *client* yang melakukan koneksi ke *server* dapat dilihat dengan menggunakan perintah “*gedit /var/log/knockd.log*”. hasilnya dapat dilihat pada gambar 18.

```

root@roflq-275E4E-275E5E:~$ sudo bash
[sudo] password for roflq:
root@roflq-275E4E-275E5E:~# cat /var/log/knockd.log
[2016-11-25 00:25] starting up, listening on wlan0
[2016-11-25 00:26] 10-42-0-67: openSSH: Stage 1
[2016-11-25 00:26] 10-42-0-67: openSSH: Stage 2
[2016-11-25 00:26] 10-42-0-67: openSSH: Stage 3
[2016-11-25 00:26] 10-42-0-67: openSSH: OPEN SESAME
[2016-11-25 00:26] openSSH: running command: /sbin/iptables -I INPUT -s 10.42.0.67 -p tcp --dport 22 -j ACCEPT
[2016-11-25 00:27] 10-42-0-67: closeSSH: Stage 1
[2016-11-25 00:27] 10-42-0-67: closeSSH: Stage 2
[2016-11-25 00:27] 10-42-0-67: closeSSH: Stage 3
[2016-11-25 00:27] 10-42-0-67: closeSSH: OPEN SESAME
[2016-11-25 00:27] closeSSH: running command: /sbin/iptables -D INPUT -s 10.42.0.67 -p tcp --dport 22 -j ACCEPT
root@roflq-275E4E-275E5E:~# 

```

Gambar 18. File knockd.log

Pada gambar 18 dijelaskan bahwa pada tanggal 25-11-2016 jam 00:26 dengan *interface wlan0* membuka port 22 dan pada tanggal 25-11-2016 jam 00:27 menutup layanan ssh yang berjalan pada port 22.

3.2 Scanning Port Tanpa Port Knocking

Pengujian *scanning port* dilakukan pada kondisi *server* tanpa menerapkan metode port knocking untuk mengetahui *service* atau layanan apa saja yang ada di *server* serta berjalan di port berapa saja, Pengujian ini di lakukan pada *server* sebelum menggunakan metode *port knocking*, Dalam pengujian *scanning port* menggunakan aplikasi scanning port.

Dalam pengujian pada penelitian ini memanfaatkan aplikasi nmap guna menemukan layanan pada *server* dan berjalan pada port berapa. Lebih jelasnya lihat gambar 19.

```

ulum@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# nmap 10.42.0.1
Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-24 06:29 EST
Nmap scan report for 10.42.0.1
Host is up (0.059s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
53/tcp    open  domain
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 24:F5:AA:6D:6C:17 (Samsung Electronics Co.)
Nmap done: 1 IP address (1 host up) scanned in 25.18 seconds
root@ulum-K43U:~# 

```

Gambar 19. Scanning port tanpa metode port knocking

Dari hasil gambar 19 dapat dijelaskan *scanning port* menggunakan nmap dapat menemukan layanan SSH pada port 22 yang ada di *server* dalam kondisi terbuka serta nmap dapat membaca *operating system* yang di gunakan *server* yang tanpa menggunakan metode port knocking.

3.3 Scanning Port Dengan Port Knocking

Pengujian *scanning port* dilakukan pada kondisi *server* menggunakan metode port knocking untuk mengetahui *service* atau layanan apa saja yang ada di *server* serta berjalan di port berapa saja, Pengujian ini di lakukan pada *server* sebelum menggunakan metode *port knocking*.

Pengujian *scanning port* menggunakan Nmap yang di lakukan pada *server* dalam kondisi menggunakan metode *port knocking* untuk mengetahui apakah Nmap masih bisa menemukan *service* atau layanan pada *server*.

```
ulum@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# nmap 10.42.0.1

Starting Nmap 6.40 ( http://nmap.org ) at 2016-11-24 12:03 EST
Nmap scan report for 10.42.0.1
Host is up (0.0030s latency).
All 1000 scanned ports on 10.42.0.1 are filtered
MAC Address: 24:F5:AA:6D:6C:17 (Samsung Electronics Co.)

Nmap done: 1 IP address (1 host up) scanned in 21.16 seconds
root@ulum-K43U:~#
```

Gambar 20. Scanning port dengan metode port knocking

Dari hasil gambar 20 dapat dijelaskan setelah melakukan *scanning port* dengan menggunakan Nmap pada *server* menggunakan metode *port knocking* dengan status *filtered*.

3.4 Pengujian Koneksi

Pengujian koneksi dilakukan untuk mengetahui apakah sistem *port knocking* yang berjalan di *server* sesuai dengan rule atau aturan yang sudah ditetapkan pada aplikasi *knockd*.

3.4.1 Pengujian Koneksi Langsung

Pada pengujian ini dilakukan dengan cara melakukan percobaan koneksi koneksi langsung SSH dalam kondisi semua *port* menuju *server* di tutup oleh *firewall* ke *server*. Jelasnya bisa dilihat pada gambar 21.

```
root@ulum-K43U:~#
ulum@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# ssh rofiq@10.42.0.1
[  ]
```

Gambar 21. Koneksi langsung menuju server

Pada gambar 21 dapat dijelaskan *client* melakukan koneksi ke *server* dengan kondisi semua port di tutup dengan menerapkan aturan DROOP aturan ini dimana semua koneksi yang melewati *firewall* dan tidak sesuai dengan aturan *firewall* akan langsung di blok tanpa memberi peringatan apapun

Pada pengujian yang kedua di mana *client* melakukan koneksi ke *server* untuk mengakses SSH yang berjalan di port 22 dengan kondisi *server* semua koneksi di tutup *firewall* dengan menggunakan aturan REJECT.

```
root@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# knock -v 10.42.0.1 600 400 300
hitting tcp 10.42.0.1:600
hitting tcp 10.42.0.1:400
hitting tcp 10.42.0.1:300
root@ulum-K43U:~# ssh rofiq@10.42.0.1
ssh: connect to host 10.42.0.1 port 22: Connection refused
root@ulum-K43U:~# ssh rofiq@10.42.0.1
ssh: connect to host 10.42.0.1 port 22: Connection refused
root@ulum-K43U:~#
```

Gambar 22. Koneksi dengan aturan reject

Pada gambar 22 dapat di jelaskan di mana *client* melakukan koneksi ke *server* di blok oleh *firewall* dengan memberikan peringatan berupuan CONECTION REFUSED.

3.4.2 Pengujian Dengan Ketukan Yang Salah

Pengujian koneksi yang dilakukan pada bahasan ini adalah untuk mengetahui bila ketukan yang dilakukan client salah atau berbeda dengan aturan yang di tetapkan pada *port knocking*, ataupun ketukan yang dilakukan kurang, misalnya pada rule *port knocking* menggunakan aturan ketukan tiga port dan client mengetuk dengan 2 port maka *firewall* tidak akan mengijinkan *client* tersebut mengakses SSH.

```
root@ulum-K43U:~#
ulum@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# knock -v 10.42.0.1 300 400 500
hitting tcp 10.42.0.1:300
hitting tcp 10.42.0.1:400
hitting tcp 10.42.0.1:500
root@ulum-K43U:~# ssh rofiq@10.42.0.1
ssh: connect to host 10.42.0.1 port 22: Connection refused
root@ulum-K43U:~#
root@ulum-K43U:~# ssh rofiq@10.42.0.1
ssh: connect to host 10.42.0.1 port 22: Connection refused
root@ulum-K43U:~#
```

Gambar 23. Pengujian koneksi yang salah

Pada gambar 23 dapat dijelaskan *client* tidak dapat mengakses layanan SSH karena urutan ketukan yang di lakukan *client* tidak sesuai dengan rule yang ditetapkan pada *firewall*, Dalam koneksi tersebut client mengetuk port 300,400,500 sedangkan rule yang ditetapkan adalah 300,400,600 sehingga client tidak di ijinkan oleh *firewall* untuk mengakses SSH.

3.4.3 Pengujian Sesuai Aturan

Pengujian yang dilakukan pada bahasan mengakses SSH sesuai dengan rule yang sudah ditetapkan pada *port knocking* misalnya: *client* dalam mengakses SSH dengan melakukan ketukan port pada port 300,400,600 sehingga *client* tersebut diijinkan mengakses SSH oleh *firewall*

```
rofiq@rofiq-275E4E-275E5E:~#
ulum@ulum-K43U:~$ sudo bash
[sudo] password for ulum:
root@ulum-K43U:~# knock -v 10.42.0.1 300 400 600
hitting tcp 10.42.0.1:300
hitting tcp 10.42.0.1:400
hitting tcp 10.42.0.1:600
root@ulum-K43U:~# ssh rofiq@10.42.0.1
rofiq@10.42.0.1's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation: https://help.ubuntu.com/
Last login: Thu Nov 24 18:27:37 2016 from 10.42.0.67
rofiq@rofiq-275E4E-275E5E:~#
```

Gambar 24. Koneksi sesuai aturan

Pada gambar di atas dapat di jelaskan *client* melakukan ketukan port sesuai dengan *rule* yang sudah ditetapkan sehingga *client* tersebut bisa mengakses layanan SSH.

3.4.4 Pengujian Koneksi Kembali

Pengujian pada bahasan ini menjelaskan setelah *client* berhasil mengakses layanan SSH dan setelah mengakhiri sesi koneksinya untuk melakukan koneksi kembali.

```
root@roflq:~# roflq@roflq:275E4E-275E5E:~#
[roflq] password: 
root@roflq-K43U:~# knock -v 10.42.0.1 300 400 600
hitting tcp 10.42.0.1:300
hitting tcp 10.42.0.1:400
hitting tcp 10.42.0.1:600
root@roflq-K43U:~# sh roflq@10.42.0.1
root@10.42.0.1:~$ password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation: https://help.ubuntu.com/
Last login: Thu Nov 24 18:27:37 2016 from 10.42.0.67
root@roflq:275E4E-275E5E:~$ exit
[roflq] password: 
Connection to 10.42.0.1 closed.
root@roflq-K43U:~# ssh roflq@10.42.0.1
root@10.42.0.1:~$ password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation: https://help.ubuntu.com/
Last login: Fri Nov 25 00:16:09 2016 from 10.42.0.67
root@roflq:275E4E-275E5E:~$
```

Gambar 25. Koneksi kembali normal

Pada gambar 25 client telah berhasil melakukan koneksi ke SSH yang bejalan pada port 22 dan setelah itu client keluar dari SSH. Client masih bisa mengakses SSH kembali tanpa melakukan ketukan sebelumnya, Karena Ip address client masih berada dalam list IPTABLES.

IV. KESIMPULAN

Berdasarkan kegiatan penelitian yang telah penulis lakukan terkait metode *port knocking*, maka ada beberapa kesimpulan yang dapat diambil, diantaranya adalah :

1. Sistem yang rancang telah mampu untuk menambah keamanan dalam proses autentikasi ke *server*, karena port tidak terbuka secara bebas ke publik.
2. Port knocking dapat mencegah penyerang dari memindai sistem seperti *service* SSH dengan melakukan port scanning, sehingga *service* SSH tidak mudah dilacak dan diakses orang lain.
3. Metode *port knocking* untuk meningkatkan keamanan port SSH dapat menggunakan *software free* atau gratis yaitu *knockd* tanpa harus mengeluarkan biaya mahal dalam implementasinya terhadap sistem.
4. Metode *port knocking* dapat menambah keamanan disisi *server* karena *client* untuk mengakses SSH memerlukan sebuah ketukan port.

V. SARAN

Meningkatkan keamanan *remote server* menggunakan metode *port knocking* ini tentu tidak terlepas dari beberapa kekurangan. Oleh sebab itu, untuk pengembangan selanjutnya yang lebih baik, penulis menyarankan beberapa hal diantaranya adalah:

1. Sistem ini masih memanfaatkan program daemon yaitu *knockd* sehingga kalau daemon *port knocking*

2. mati server tidak dapat di akses karena tertutup oleh firewall.
3. Perlu adanya proses enkripsi untuk mengamankan *port-port* yang dikirim oleh *client*.
4. Perlu adanya tambahan layanan di *server* tidak hanya *remote server*.

DAFTAR PUSTAKA

- [1] Andri, Trismanto. 2015. Port knocking URL:<http://mastopix.blogspot.co.id/2014/11/pengertian-port-knocking-teknik-yang.html>
- [2] Edy Haryanto. 2013. *Meningkatkan keamanan port ssh dengan metode port knocking menggunakan shorewall Pada sistem operasi linux*, Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom Yogyakarta Yogyakarta
- [3] Firman, Cahaya, Putra. 2009. *Rancang Bangun Sistem Keamanan Jaringan Komputer Dengan Menggunakan Metode Port Knocking*. Universitas pembangunan nasional “veteran” Jawa timur
- [4] Guna Darma “Simbol-simbol standar dalam penggambaran flowchart” 2016 <http://tris.staf.gunadarma.ac.id>
- [5] I, Komang, Hartawan, Wijaya. 2011. *Implementasi Port Knocking Pada Sistem Keamanan Jaringan Dengan Menerapkan Algoritma Rsa (Rivest Shamir Adleman)*. Universitas pembangunan nasional “veteran” Jawa timur
- [6] Krzywinski, Martin. Juli 2016, Port Knocking. URL:<http://www.portknocking.org/view/about/summary>
- [7] Muhammad Saleh Hafizh Fajri, 2014. *Analisa Port Knocking Pada Sistem Operasi Linux Ubuntu Server 12.04 LTS*. Program Studi Teknik Informatika Jurusan Komputer Politeknik Caltex Riau
- [8] Oktaviani. 2007. *Mengenal sistem firewall*. Universitas Guna Darma.
- [9] Rois, Awang, Rimbayani. 2013. *Percangan dan Implementasi Autentikasi Remote Server Dengan Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module*. Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
- [10] Rio Handicha. 2009. *Pengertian Dan Manfaat Memakai SSH*. Di ambil 10 Oktober 2015 dari URL:<http://riohandicha.blogspot.co.id/2014/06/pengertian-dan-manfaat-memakai-ssh.html>
- [11] Syarif, Muhar. 2007-2008, *Implementasi IPTABLES sebagai Filtering Firewall*. Teknik Informatika Billngual, Fakultas Ilmu Komputer, Universitas Sriwijaya
- [12] Verysson. Juli 2016. *Cara kerja port knocking*. URL:http://cara_kerja_port Knocking _verrysoon blog's.com
- [13] WAHYU PURNAMA, 2014. *Perancangan Sistem Pengamanan Akses Otentifikasi Menggunakan Metode Port Knocking Dan Firewall Action Tarpit Pada Mikrotik Rb951-2n*. Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom Yogyakarta.