

**PERANCANGAN DAN IMPLEMENTASI AUTENTIKASI REMOTE  
SERVER DENGAN MENGGUNAKAN METODE PORT KNOCKING  
BERBASIS LOADABLE KERNEL MODULE**

Skripsi

untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA**

**YOGYAKARTA**

**2013**

**PERANCANGAN DAN IMPLEMENTASI AUTENTIKASI REMOTE  
SERVER DENGAN MENGGUNAKAN METODE PORT KNOCKING  
BERBASIS LOADABLE KERNEL MODULE**

Skripsi

untuk memenuhi sebagian persyaratan  
mencapai derajat Sarjana S-1



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA**

**YOGYAKARTA**

**2013**

**PENGESAHAN SKRIPSI/TUGAS AKHIR**

Nomor : UIN.02/D.ST/PP.01.1/3290/2013

Skripsi/Tugas Akhir dengan judul

: Perancangan dan Implementasi Autentikasi Remote Server  
Dengan Menggunakan Metode Port Knocking Berbasis  
Loadable Kernel Module

Yang dipersiapkan dan disusun oleh :

Nama : Rois Awang Rimbayani

NIM : 07650082

Telah dimunaqasyahkan pada : Jum'at, 18 Oktober 2013

Nilai Munaqasyah : A -

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

**TIM MUNAQASYAH :**

Ketua Sidang

Bambang Sugiantoro, M.T  
NIP. 19751024 200912 1 002

Penguji I

Sumarsono, M.Kom  
NIP. 19710209 200501 1 003

Penguji II

Nurochman, M.Kom  
NIP. 19801223 200901 1 007

Yogyakarta, 28 Oktober 2013

UIN Sunan Kalijaga

Fakultas Sains dan Teknologi

Dekan

H. Akh. Minhaji, M.A, Ph.D  
NIP. 19580919 1986031 002

**SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR**

Hal : Permohonan

Lamp :-

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

*Assalamu'alaikum wr. wb.*

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Rois Awang Rimbayani

NIM : 07650082

Judul Skripsi : **Perancangan dan Implementasi Autentikasi Remote Server  
Dengan Menggunakan Metode Port Knocking Berbasis  
Loadable Kernel Module**

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapan terima kasih.

*Wassalamu'alaikum wr. wb.*

Yogyakarta, 7 Oktober 2013

Pembimbing

Bambang Sugiantoro, M. T., Comp. TIA.  
NIP: 19751024 200912 1 002

## PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Rois Awang Rimbayani  
Nim : 07650082  
Program Studi : Teknik Informatika  
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **Perancangan Dan Implementasi Autentikasi Remote Server Dengan Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module** tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 20 September 2013

Yang Menyatakan,



Rois Awang Rimbayani  
NIM : 07650082

## KATA PENGANTAR

*Bismillahirrahmanirrahiim*

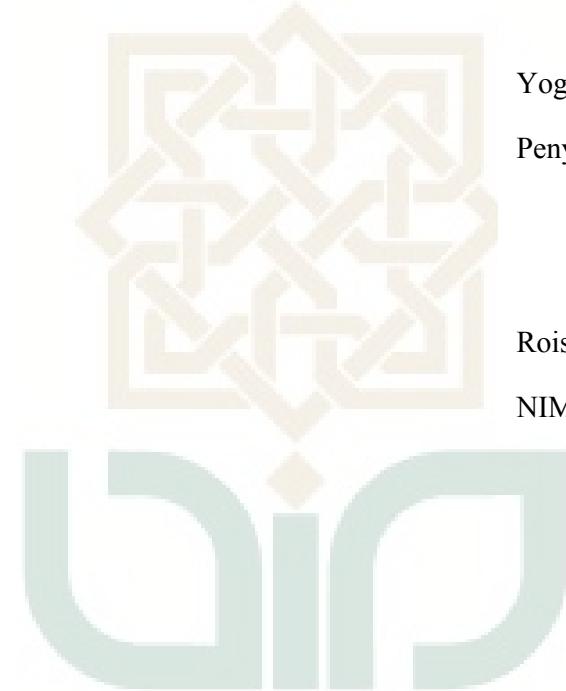
Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya sehingga penyusun dapat menyelesaikan skripsi dengan judul **“Perancangan dan Implementasi Autentikasi Remote Server dengan Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module”** sebagai salah satu syarat untuk mencapai gelar kesarjanaan pada program studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta. Shalawat serta salam semoga tetap tercurahkan kepada baginda Nabi Muhammad SAW beserta keluarga dan sahabatnya.

Dalam penyelesaian skripsi ini ada beberapa pihak yang telah membantu menyusun serta mengoreksi dan member dukungan baik moril maupun materiil. Sebagai rasa hormat dan ucapan terima kasih penyusun sampaikan kepada :

1. Bapak Prof. Dr. H. Musa Asy’arie, M.A., selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Bapak Prof. Dr. H. Akh. Minhaji, M.A., Ph.D., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Agus Mulyanto, S.Si., M.Kom., selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
4. Bapak Nurochman, M.Kom., selaku Sekretaris Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta.
5. Ibu Ade Ratnasari, S.Kom., M.T., selaku dosen pembimbing akademik selama penyusun menyelesaikan kuliah di Prodi Teknik Informatika.

6. Bapak Bambang Sugiantoro, S.Si., M.T, Comp.TIA., selaku dosen pembimbing skripsi, sehingga penyusun dapat menyelesaikan skripsi ini dengan baik.
7. Seluruh dosen Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberikan bimbingan serta inspirasi yang sangat berharga.
8. Ayahanda Suyanto serta Ibunda Siti Barokah tercinta, atas doa serta perhatian yang selama ini diberikan, serta dukungan materiil yang penyusun dapatkan.
9. Adikku Lilo Kurniawan Saputro dan Ridho Ridhiansyah yang selalu menjadi semangat kepada penyusun untuk segera lulus dengan baik.
- 10 Anisa Prihantini, beserta keluarga besar di Magelang yang selalu memberikan doa, serta dukungan dan motivasi untuk segera menyelesaikan tugas akhir ini dengan sebaik-baiknya.
11. Teman-temanku semua di Program Studi Teknik Informatika angkatan 2007 serta di AKAKOM Yogyakarta yang selalu menginspirasi penulis untuk selalu belajar dengan baik.
12. Teman-teman serta adik-adik di Masjid Ambargama dan Kos Ambarrukmo Yogyakarta, yang selalu memberikan dukungan serta semangatnya.

Akhirnya penyusun hanya bisa berharap dan berdoa kepada Allah SWT semoga apa yang telah dilakukan menjadi amal sholeh dan diberikan ridho Allah SWT. Penyusun menyadari tentu masih banyak kesalahan serta kekurangan yang terdapat dalam skripsi ini, maka pendapat, saran dan kritik untuk membangun dan perbaikan sangat penyusun harapkan. Semoga skripsi ini bermanfaat bagi penyusun pada khusunya, dan bagi para pembaca pada umumnya. Terima Kasih.



Yogyakarta, 12 Juli 2013

Penyusun,

Rois Awang Rimbayani

NIM. 07650082

## HALAMAN PERSEMPAHAN

Bismillahirrahmaanirrahiim

Puji syukur kehadiran Allah SWT dan sholawat untuk Rasul-Nya Muhammad SAW.

Selesainya penelitian ini tidak lepas dari Rahmat serta Kasih saying-Nya yang selalu tercurah untuk hamba-Nya. Pada halaman ini saya ingin mengucapkan terima kasih kepada :

Bapak, Ibu yang mendoakan dan mendidikku.

Adik-adikku, Lilo, Ridho,

Teman-temat Teknik Informatika angkatan 2007

Segenap Dosen Teknik Informatika

Temat-temat kos Ambarukmo 243 D

Teman-temat Masjid Ambargama

Semoga selalu diberi Perlindungan dan Kasih Sayang dari Allah SWT

Terima Kasih



## DAFTAR ISI

HALAMAN JUDUL .....	i
PENGESAHAN SKRIPSI .....	ii
PERSETUJUAN SKRIPSI/TUGAS AKHIR .....	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
KATA PENGANTAR .....	v
HALAMAN PERSEMBAHAN .....	viii
MOTTO .....	ix
DAFTAR ISI .....	x
DAFTAR GAMBAR .....	xiv
DAFTAR TABEL .....	xv
DAFTAR LAMPIRAN.....	xvi
INTISARI.....	xvii
ABSTRACT.....	xviii
BAB I PENDAHULUAN	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	3
1.5 Manfaat Penelitian .....	4
1.6 Keaslian Penelitian.....	4

BAB II TINJAUAN PUSTAKA DAN LANDASARN TEORI.....	5
2.1 Tinjauan Pustaka .....	5
2.2 Landasar Teori .....	6
2.2.1 Autentikasi .....	6
2.2.2 <i>Server</i> dan <i>Remote Server</i> .....	8
2.2.3 Port Knocking .....	9
2.2.4 Kernel.....	13
2.2.5 Linux dan GNU/Linux.....	17
2.2.6 Debian GNU/Linux .....	18
2.2.7. Sistem Memori di Linux .....	19
2.1.7.1. <i>User Space</i> .....	20
2.1.7.2. <i>Kernel Space</i> .....	21
2.2.8. <i>Loadable Kernel Module (LKM)</i> .....	22
2.2.9. Firewall .....	24
2.2.10. Netfilter .....	25
2.2.11. IPTables .....	29
2.2.12. SSH .....	32
2.2.13. Transmission Control Protocol (TCP).....	32
BAB III METODE PENELITIAN .....	35
3.1 Pengumpulan Data .....	35
3.2 Pengembangan Sistem .....	36
3.3. Tahapan Pengembangan Sistem.....	37

BAB IV ANALISIS DAN PERANCANGAN SISTEM .....	39
4.1. Analisis Masalah .....	39
4.2. Analisis Kebutuhan .....	39
4.3. Arsitektur Skenario Autentikasi.....	41
4.4. Diagram Alir Autentikasi Menggunakan <i>Port Knocking</i> .....	42
4.5. Diagram Alir Pengembangan Sistem Aplikasi <i>Port Knocking</i> .....	45
4.6. Perancangan Format Ketukan.....	46
4.7. Perancangan Aplikasi Server .....	48
4.8. Perancangan Aplikasi Daemon .....	50
4.9. Perancangan Aplikasi <i>client</i> .....	52
BAB V. IMPLEMENTASI DAN PENGUJIAN.....	53
5.1. Implementasi.....	53
5.1.1 Instalasi <i>virtual machine</i> .....	53
5.1.2. Instalasi GCC (GNU C Compiler) .....	54
5.1.3.Instalasi Paket Linux <i>header</i> .....	55
5.1.4. Instalasi Utilitas Make .....	56
5.1.5. Implementasi Firewall .....	56
5.1.6. Implementasi <i>Server Port Knocking</i> .....	58
5.1.7. Implementasi Daemon .....	67
5.1.8. Implementasi <i>client Port Knocking</i> .....	68
5.2. Pengujian .....	69
5.2.1. Kompilasi Kode Sumber.....	69
5.2.2. Pengujian Membuka <i>firewall</i> .....	72

5.2.3. Pengujian Menutup <i>firewall</i> .....	75
5.2.4. Pengujian Keamanan Port Knocking.....	75
5.2.1. Pengujian dengan Nmap.....	76
5.2.2. Pengujian dengan Metasploit .....	77
BAB VI. HASIL PENGUJIAN DAN PEMBAHASAN .....	79
6.1. Analisis .....	79
6.2. Perancangan .....	80
6.3. Implementasi.....	80
6.4. Pengujian .....	81
BAB VII. KESIMPULAN DAN SARAN .....	83
7.1 Kesimpulan .....	83
7.2. Saran.....	83
DAFTAR PUSTAKA .....	85
LAMPIRAN .....	87

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Salah satu komputer <i>server</i> merk HP .....	9
<b>Gambar 2.2</b> Semua <i>port</i> di <i>server</i> tertutup .....	11
<b>Gambar 2.3</b> <i>Client</i> mengirimkan paket SYN.....	12
<b>Gambar 2.4</b> Aplikasi di sisi <i>server</i> membaca dan menganalisa <i>log</i> .....	12
<b>Gambar 2.5</b> Server membuka <i>port</i> tertentu .....	13
<b>Gambar 2.6</b> Arsitektur Monolithic <i>kernel</i> .....	17
<b>Gambar 2.7</b> Arsitektur Micro-kernel.....	17
<b>Gambar 2.8</b> Logo Debian GNU/Linux .....	19
<b>Gambar 2.9</b> Bagian-bagian dari area memori di Linux.....	20
<b>Gambar 2.10</b> Hubungan antara <i>user application</i> dan <i>kernel</i> .....	21
<b>Gambar 2.11</b> Perjalanan paket data jaringan .....	26
<b>Gambar 2.12</b> Tahapan <i>three-way handshaking</i> pada protokol TCP .....	33
<b>Gambar 3.1</b> Model <i>Waterfall</i> .....	37
<b>Gambar 4.1.</b> Diagram Alir Autentifikasi dengan <i>Port Knocking</i> .....	42
<b>Gambar 4.2.</b> Diagram Alir Pembuatan Aplikasi <i>Port Knocking</i> .....	43
<b>Gambar 4.3.</b> Diagram Alir Aplikasi <i>Server Port Knocking</i> .....	45
<b>Gambar 4.4.</b> Diagram alir perancangan aplikasi <i>daemon</i> .....	49
<b>Gambar 4.5.</b> Diagram Alir aplikasi <i>client</i> .....	51
<b>Gambar 4.6.</b> Perancangan Aplikasi <i>client</i> .....	52
<b>Gambar 5.1.</b> Spesifikasi <i>guest os</i> pada VirtualBox.....	54

## DAFTAR TABEL

<b>Tabel 2.1</b> Tabel <i>return code</i> pada Netfilter .....	27
<b>Tabel 2.2</b> <i>Table-table</i> dalam IPTables beserta <i>chain</i> .....	30
<b>Tabel 2.3</b> Beberapa Layanan umum yang menggunakan protoko TCP .....	34
<b>Tabel 4.1</b> Contoh urutan port untuk membuka dan menutup <i>firewall</i> .....	47
<b>Tabel 6.1</b> Form Pengujian Fungsionalitas Sistem.....	81
<b>Tabel 6.2</b> Hasil Pengujian Fungsionalitas sistem .....	82



## **DAFTAR LAMPIRAN**

Lampiran A : Kode Sumber lengkap pk_server .....	88
Lampiran B : Kode Sumber lengkap pk_daemon .....	93
Lampiran C : Kode Sumber lengkap pk_client .....	96
Lampiran D : Form Pengujian.....	98



# **Perancangan dan Implementasi Autentikasi Remote Server Dengan Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module**

**Rois Awang Rimbayani**  
**NIM. 07650082**

## **INTISARI**

Keamanan komputer *server* merupakan salah tugas pokok dari *system administrator*. Hal ini didasarkan pada karakteristik umum dari jaringan komputer yang pada dasarnya adalah tidak aman untuk diakses secara bebas. Berbagai celah di sisi *server* membuat pihak-pihak yang tidak bertanggung jawab berusaha menerobos masuk ke sistem dengan berbagai teknik. Terbukanya *port* untuk layanan yang bersifat publik maupun layanan yang sifatnya privat, memiliki kemungkinan resiko yang tinggi untuk diserang oleh para *attacker*.

Penelitian ini lebih menekankan pada implementasi pemrograman berbasis *Loadable Kernel Module* untuk membuat aplikasi *port knocking* yang digunakan untuk autentikasi ke sebuah *server*. Meskipun di sisi *server* sudah terpasang *firewall* yang cukup canggih, namun sampai saat ini masih banyak *server* yang berhasil diterobos. Hal ini karena adanya *port* yang terbuka dan dapat dilihat oleh orang luar. Pada penelitian ini, penulis menggunakan metode pengumpulan data berupa studi pustaka, pengkodean, serta metode pengembangan sistem. Sedangkan tahapan penelitian terbagi dalam beberapa langkah yakni perancangan sistem secara umum, perancangan setiap aplikasi dan pengujian.

Hasil akhir dari penelitian adalah terbentuknya sistem autentikasi yang cukup aman, karena ditutupnya *port* yang sangat penting, yaitu *port SSH* dari orang yang tidak berhak untuk mengakses *server*. Dengan menggunakan metode *Port knocking* para pengelola jaringan lebih terbantu selama proses autentikasi ke *server* yang ia kelola. Dengan demikian akan membuat para *attacker* harus berusaha lebih keras lagi untuk bisa menembus ke dalam sistem.

Kata Kunci : *Port Knocking, port, firewall, autentikasi, Loadable Kernel Module*

**Perancangan dan Implementasi Autentikasi Remote Server Dengan  
Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module**

**Rois Awang Rimbayani**  
**NIM. 07650082**

**ABSTRACT**

Security server computers is one of the main tasks of the system administrator . It is based on the general characteristics of computer network that is basically not safe to be accessed freely . Various loopholes in the server side to make the parties not responsible to try to break into the system with a variety of techniques . Opening ports for services that are public and which are private services , have a high risk for the possibility of being attacked by the attacker .

This research emphasizes the implementation of loadable kernel module based programming to create applications that port knocking is used to authenticate to a server . Although the server-side firewall installed fairly sophisticated , but until now there are many servers that was breached . This is because the port is open and can be seen by outsiders . In this study , the authors use data collection methods such as literature review, coding , and system development methods . While the research stage is divided into several steps that the system design in general , design and testing each application .

The end result of the study is the formation of a fairly secure authentication system , due to the closure of a very important port , the port SSH from people who are not entitled to access the server . By using port knocking over network managers helped during the authentication process to the server that he managed. Thus would make the attacker must try harder to penetrate into the system.

**Keywords :** *Port Knocking, port, firewall, authentication, Loadable Kernel Module.*

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Dalam hal pengelolaan *server*, biasanya *administrator* sistem tidak selalu harus berada dalam ruang *server*. Hal ini karena biasanya ruangan *server* dirancang agar memiliki suhu yang cukup dingin dan stabil, dimana hal itu tentu kurang baik untuk tubuh. Sehingga biasanya seorang *administrator* menjalankan tugasnya dari luar ruang *server* dengan melalui aplikasi *remote server*. Aplikasi ini diinstal di *server* dan selalu *listen* pada *port* tertentu dan selanjutnya menunggu permintaan koneksi dari aplikasi *client*. Dengan demikian seorang *administrator* cukup melakukan proses autentikasi ke aplikasi tersebut dan jika berhasil maka *administrator* tersebut akan mendapatkan akses untuk mengelola *server*.

Berbagai metode maupun jumlah serangan pada suatu *server* semakin hari semakin meningkat. Terbukanya beberapa *port* yang *listen* secara tidak langsung akan mengundang para *attacker* maupun pihak-pihak tertentu yang tidak bertanggung jawab untuk menerobos masuk ke dalam *server* melalui *port* tersebut. Hal yang sering dilakukan oleh para *attacker* adalah mencoba untuk mengeksloitasi berbagai aplikasi yang sedang *running* melalui *port* yang terbuka pada sisi *server*. Untuk mencegah hal-hal yang tidak diinginkan, biasanya *administrator* akan memasang *firewall* dan melakukan beberapa konfigurasi yang pada intinya adalah untuk membatasi siapapun yang akan mengakses *server*.

Terbukanya *port* pada *server*, terutama *port* untuk aplikasi *remote server*, tentu akan menjadi pusat perhatian *attacker* untuk dieskploitasi. *Port knocking* hadir sebagai salah satu metode autentikasi yang dapat digunakan untuk mengatasi masalah di atas. Metode ini memiliki kemampuan untuk menentukan siapa yang memang benar-benar berhak mengakses *server*, dan biasanya diimplementasikan pada layanan-layanan yang sifatnya jarang diakses, seperti mengakses *secure shell* di *remote server*.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah sebagai berikut :

1. Bagaimana merancang sistem autentikasi *remote server* dengan menggunakan metode *Port Knocking*.
2. Bagaimana mengimplementasikan pemrograman berbasis *Loadable Kernel Module* untuk membuat aplikasi yang dapat memonitor lalu lintas paket data ketika terjadi permintaan untuk melakukan proses autentikasi ke *remote server*.
3. Bagaimana membuat aplikasi *client* yang dapat digunakan oleh *administrator* untuk mengirimkan *port-port* tertentu sebagai bentuk usaha untuk melakukan proses autentikasi *remote server*.

### 1.3 Batasan Masalah

Batasan-batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Sistem yang akan dibangun diimplementasikan pada jaringan lokal yang terdiri dari sebuah *client* dan sebuah *server* dengan memanfaatkan *virtual machine*.
2. Sistem yang dikembangkan diimplementasikan untuk aplikasi SSH (*secure shell*) dengan *port* 22.
3. Jumlah *port* yang dikirimkan ke *server* dengan menggunakan protokol TCP dan berjumlah tetap serta tidak dapat diubah pada saat aplikasi *server Port Knocking* sedang berjalan..
4. Aplikasi *server Port Knocking* dikembangkan dengan menggunakan Bahasa Pemrograman C, sedangkan aplikasi *client* dikembangkan dengan menggunakan Bahasa Pemrograman Perl.
5. Tidak ada fungsi waktu dan fungsi enkripsi pada saat mengirimkan *port-port* untuk proses autentikasi.
6. *Server Port Knocking* hanya menerima satu pengguna setiap proses autentikasi yang telah berhasil dilakukan.

### 1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut :

1. Merancang autentikasi *remote server* yang lebih aman yaitu dengan menggunakan metode *Port Knocking*.
2. Mengimplementasikan pemrograman pada tingkat *kernel*, yaitu *Loadable Kernel Module Programming* pada *kernel Linux* untuk

membuat aplikasi yang dapat memonitor serta memfilter paket-paket yang masuk ke *server* untuk digunakan sebagai proses autentikasi menggunakan metode *Port Knocking*.

3. Merancang dan membuat aplikasi *client* yang dapat memudahkan pengguna yang ingin melakukan autentikasi ke *server* dengan menggunakan metode *Port Knocking*.
4. Menambahkan *security layer* tambahan di sisi *server* yang diimplementasikan pada level *kernel*.

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah membantu para *administrator* sistem agar lebih aman dalam melakukan proses autentikasi *remote server* dengan menggunakan aplikasi berbasis *Loadable Kernel Module*.

## 1.6 Keaslian Penelitian

Penelitian dengan judul “Perancangan dan Implementasi Autentikasi Remote Server Dengan Menggunakan Metode Port Knocking Berbasis Loadable Kernel Module” sepengetahuan penulis belum pernah dilakukan sebelumnya, khususnya di Universitas Islam Negeri Sunan Kalijaga.

**Tabel 6.2** Hasil Pengujian Fungsionalitas sistem

Penilaian	SB	B	TB	STB
Total	125	78	-	-

Keterangan :

**SB** (Sangat Baik), **B** (Baik), **TB** (Tidak Baik), **STB** (Sangat Tidak Baik).

Dengan jumlah penguji sebanyak 29 orang. Maka :

$$\text{SB} : \frac{125}{203} \times 100 \% = 61,57 \%$$

$$\text{B} : \frac{78}{203} \times 100 \% = 38,42 \%$$

$$\text{TB} : 0 \%$$

$$\text{STB} : 0 \%$$

Dari tabel 6.1 dapat diperoleh hasil bahwa sebagian besar pengguna menyatakan sistem *Port Knocking* sudah berjalan dengan baik. Hal ini ditandai dengan 61,57 % pengguna menyatakan aplikasi ini berfungsi sangat baik, dan 38,42 % menyatakan bahwa aplikasi ini berjalan dengan baik. Dari hasil pengujian yang dilakukan, dapat ditarik kesimpulan bahwa aplikasi *Port Knocking* untuk autentikasi *remote server* sudah berjalan dengan baik, akan tetapi masih ada beberapa hal yang perlu perbaikan terkait masalah kualitas keamanan dan perlu ada pengembangan lebih lanjut guna penyempurnaan berikutnya.

## BAB VII

### KESIMPULAN DAN SARAN

#### 7.1. Kesimpulan

Berdasarkan kegiatan penelitian yang telah penulis lakukan, maka ada beberapa kesimpulan yang dapat diambil, diantaranya adalah :

1. Sistem *Port Knocking* ini telah diujicoba pada jaringan LAN yang terdiri dari 2 buah komputer, *client* dan *server*.
2. Sistem yang dibangun telah mampu untuk menambah keamanan dalam proses autentikasi ke *server*, karena *port* tidak terbuka secara bebas ke publik.

#### 7.2 Saran

Sistem autentikasi *remote server* dengan 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 masih memanfaatkan program *daemon* untuk memantau *log* autentikasi. Jika *daemon* mati, maka *firewall* tidak akan bisa dibuka maupun ditutup. Oleh sebab itu, perlu adanya teknik yang lebih baik dalam hal proses komunikasi antara aplikasi *server* yang berada di area *kernel space* dengan aplikasi *firewall* yang berada di area *user space*.

2. Perlu adanya proses enkripsi untuk mengamankan *port-port* yang dikirim oleh *client*.
3. Perlu adanya fungsi waktu, dimana *client* diberi tenggang waktu untuk melakukan koneksi ke SSH, meskipun *client* telah berhasil membuka *firewall*. Jika dengan tenggang waktu yang diberikan pengguna belum mampu melewati proses autentikasi SSH, maka IP dari komputer pengguna akan tercatat sebagai IP yang perlu di-*black list*.
4. Perlu adanya GUI untuk aplikasi *client*, sehingga pada saat pengguna melakukan proses autentikasi ke *server* dapat lebih interaktif dan menarik.
5. Sistem dapat dikembangkan tidak hanya untuk satu pengguna setiap mengakses *server Port Knocking*, namun sekali waktu dapat beberapa pengguna yang dapat melakukan proses autentikasi *Port Knocking*.



## DAFTAR PUSTAKA

- Azikin, Ashari. 2011. *Debian GNU/Linux*. Penerbit Informatika, Bandung.
- Forouzan, Behrouz. 2006. *TCP/IP Protocol Suite*. McGrawHill.
- Handaya, Suteja, Ashari. 2010. *Linux Ssystem Administrator*. Penerbit Informatika, Bandung.
- Haryadi, Bernardus Ivan. 2004. *Definisi dan Implementasi Dari Port Knocking*. Tugas Akhir Semester. Institut Teknologi Bandung.
- Husni. 2004. *Implementasi Jaringan Komputer dengan Linux Redhat 9*. Penerbit Andi, Yogyakarta.
- Jogiyanto. 2006. *Konsep Dasar Pemrograman Bahasa C*. Penerbit Andi, Yogyakarta.
- Kadir, Abdul. 2002. *Pengenalan Unix dan Linux*. Penerbi Andi, Yogyakarta.
- Kadir, Abdul. 2002. *Dasar Pemrograman Perl*. Penerbit Andi, Yogyakarta.
- Memahami Authentication dan Authorization.<http://www.ilmu hacking.com/basic-concept/memahami-authentication-dan-authorization/> diakses pada 31 Mei 2012.
- Perdhana, Rambu. 2011. *Harmless Hacking :Malware Analysis dan Vulnerability Development*. Penerbit Graha Ilmu, Yogyakarta.
- Port Knocking – Network Authentication Across Closed Ports.  
[www.sysadminmag.com](http://www.sysadminmag.com) diakses pada 24 Juni 2012.

Putra, Ade Eka Maulana. 2012. *Implementasi Remote Server dengan Menggunakan Metode Port Knocking Menggunakan Bahasa Python.* Tugas Akhir. Institut Teknologi Telkom.

Putra, Firman Cahaya. 2009. *Rancangan Bangun Sistem Keamanan Jaringan Komputer Dengan Menggunakan Metode Port Knocking.* Skripsi. Universitas Pembangunan Nasional Veteran Jawa Timur.

Rafiudin, Rahmat. 2004. *Panduan Menjadi Administrator Sistem Unix.* Penerbit Andi, Yogyakarta.

Rozi, M. Fahru. 2010. *Implementasi Remote Server Menggunakan Metode Port Knocking Dengan Asymmetric Encryption.* Tugas Akhir. Institut Teknologi Sepuluh November.

Simarmata, Janner. 2006. *Pengamanan Sistem Komputer.* Penerbit Andi, Yogyakarta.

Sofana, Iwan. 2008. *Mudah Membangun Server Denga Fedora.* Penerbit Informatika, Bandung.

Sofana, Iwan. 2009. *CISCO CCNA & JARINGAN KOMPUTER.* Penerbit Informatika, Bandung.

Sofana, Iwan. 2009. *Membangun Jaringan Komputer.* Penerbit Informatika, Bandung.

Thomas, Tom. 2004. *Network Security first-step.* Penerbit Andi, Yogyakarta.



**LAMPIRAN A : Kode Lengkap Aplikasi Port Knocking (pk\_server.c)**

```
#include <linux/module.h>
#include <linux/init.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/skbuff.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/if_ether.h>
#include <linux/jiffies.h>
#include <linux/timer.h>
#include <linux/proc_fs.h>
#include <asm/uaccess.h>
#include <linux/string.h>
#include <linux/vmalloc.h>

#define MAX_PORT      4

MODULE_LICENSE( "GPL" );
MODULE_AUTHOR("Rois Awang R");
MODULE_DESCRIPTION("Server Port Knocking");
/*proc file system declaration*/

#define nama_proc      "knockaddr"
#define nama_dir        "knockdir"

static char *buff_mem=NULL;
static struct proc_dir_entry *fproc=NULL;
static struct proc_dir_entry *dproc=NULL;
static char ip[17];
static char ip_tmp[17];
static char mac[17];
static char status[10];
static unsigned short
port_open[MAX_PORT]={211,212,213,214};
static unsigned short flag_open[MAX_PORT]={0,0,0,0};
static unsigned short
port_close[MAX_PORT]={311,312,313,314};
```

```

static unsigned short flag_close[MAX_PORT]={0,0,0,0};
static unsigned short count_flag_open=0;
static unsigned short count_flag_close=0;
static unsigned short count_flag_failed=0;
/*netfilter declaration*/
static struct nf_hook_ops nethook;

int read_procfs(char *buffer,char
**buffer_location,off_t offset, int buffer_length,
int *eof,void *data) {
    int ret;
    //ret = sprintf(buffer,"%s %s
%s",status,ip,mac);
    ret=sprintf(buffer,"%s %s ",status,ip);
    return ret;
}

static unsigned int fungsi_hook (
    unsigned int hooknum,
    struct sk_buff *skb,
    const struct net_device *in,
    const struct net_device *out,
    int (*okfn)(struct sk_buff *)
) {
    struct sk_buff *buff;
    struct tcphdr *header_tcp;
    struct iphdr *header_ip;
    struct ethhdr *eth;
    static unsigned short int port_tujuan;
    buff=skb;
    eth=eth_hdr(buff);

    header_ip=(struct iphdr *)skb_network_header(buff);
    snprintf(ip_tmp,16,"%pI4",&header_ip->saddr);
    //printk(KERN_INFO "IP Sumber : %pI4\n", &header_ip-
    >saddr);
    //printk(KERN_INFO "MAC Sumber :
    %0x:%0x:%0x:%0x:%0x \n",eth->h_source[0],eth-
    >h_source[1],eth->h_source[2],eth->h_source[3],eth-
    >h_source[4],eth->h_source[5]);

    if (header_ip->protocol== IPPROTO_TCP){
        header_tcp=(struct tcphdr *)((__u32
        *)header_ip+header_ip->ihl);
    }
}

```

```

    port_tujuan=htons((unsigned short int)header_tcp->dest);

    if(strcmp(status,"deactive")==0) // jika
status==deactive
    {
        if(port_tujuan==port_open[0] && flag_open[0]==0) {
            flag_open[0]=1;
            count_flag_open +=1;
        }else if(port_tujuan==port_open[1] &&
flag_open[1]==0) {
            flag_open[1]=1;
            count_flag_open +=1;
        }else if(port_tujuan==port_open[2] &&
flag_open[2]==0) {
            flag_open[2]=1;
            count_flag_open +=1;
        }else if(port_tujuan==port_open[3] &&
flag_open[3]==0) {
            flag_open[3]=1;
            count_flag_open +=1;
        }else{
            count_flag_failed +=1;
        }
    }

    if(strcmp(status,"active")==0 &&
strcmp(ip_tmp,ip)==0) // jika status==active.
    {
        if(port_tujuan==port_close[0] &&
flag_close[0]==0) {
            flag_close[0]=1;
            count_flag_close +=1;
        }else if(port_tujuan==port_close[1] &&
flag_close[1]==0) {
            flag_close[1]=1;
            count_flag_close +=1;
        }else if(port_tujuan==port_close[2] &&
flag_close[2]==0) {
            flag_close[2]=1;
            count_flag_close +=1;
        }else if(port_tujuan==port_close[3] &&
flag_close[3]==0) {
            flag_close[3]=1;
            count_flag_close +=1;
        }else{
    
```

```

        count_flag_failed +=1;
    }
}

if(count_flag_open==4 && count_flag_failed ==0 ) {
    sprintf(status,7,"active");
    sprintf(ip,16,"%pI4",&header_ip->saddr);

sprintf(mac,17,"%0x:%0x:%0x:%0x:%0x:%0x",eth-
>h_source[0],eth->h_source[1],eth->h_source[2],eth-
>h_source[3],eth->h_source[4],eth->h_source[5]);
    count_flag_open=0;
    memset(flag_open,0,sizeof(flag_open));
} else if(count_flag_close==4 &&
count_flag_failed==0){
    sprintf(status,9,"deactive");
    //memset(status,0,sizeof(status));
    memset(ip,0,sizeof(ip));
    memset(mac,0,sizeof(mac));
    count_flag_close=0;
    memset(flag_close,0,sizeof(flag_close));

} else{
    printk(KERN_ALERT "Urutan port salah...!\n");
    count_flag_failed=0;
}

return NF_ACCEPT;
} else{
    printk(KERN_INFO "bukan protokol TCP\n");
    return NF_DROP;
}
return NF_DROP;
}

static int __init pknock_in( void )
{
    strcpy(status,"deactive");
    nethook.hook=fungsi_hook;
    nethook.hooknum=NF_INET_LOCAL_IN;
    nethook.pf=PF_INET;
    nethook.priority=NF_IP_PRI_FIRST;
    nf_register_hook(&nethook);
}

```

```
// membuat berkas di /proc file system
buff_mem=(char *)vmalloc(PAGE_SIZE);
if(!buff_mem){
    printk(KERN_INFO "Gagal
mengalokasikan memori. \n");
    return 0;
}
memset(buff_mem, 0, PAGE_SIZE);
dproc=proc_mkdir(nama_dir,NULL);
if(!dproc){
    printk(KERN_INFO "Gagal membuat
direktori %s.\n",nama_dir);
    return 0;
}

fproc=create_proc_entry(nama_proc,0600,dproc);
if(fproc){
    fproc->read_proc = read_procfs;
} else{
    remove_proc_entry(nama_proc,dproc);
    printk(KERN_ALERT "Gagal membuat file
proc %s. \n",nama_proc);
}

printk(KERN_INFO "Server Port Knocking
berhasil diload...!!\n");

return 0;
}

static void __exit pknock_exit( void )
{
    nf_unregister_hook(&nethook);
    remove_proc_entry(nama_proc,dproc);

    remove_proc_entry(nama_dir,NULL);
    printk(KERN_INFO "Server Port Knocking Berhasil
ter-unload dengan baik.\n");
    return;
}

module_init(pknock_in );
module_exit(pknock_exit);
```

**LAMPIRAN B : Kode Lengkap Aplikasi Daemon (pk\_daemon.c)**

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <string.h>

int main(int argc,char *argv[]){
    struct address{
        char status[10];
        char ip[20];
        char mac[20];
    };
    pid_t process_id=0;
    pid_t sid=0;
    FILE *fplog,*fpproc;
    struct address datalog;
    int buka,tutup;
    int status_iptables=-1;

    char *newargv[]={NULL,"-A","INPUT","-p","tcp","-s",
                     NULL,"-d",NULL,"--sport","513:65535","--"
                     "dport","22","-m","state","--"
                     "state","NEW,ESTABLISHED","-j","ACCEPT",NULL};
    char *newenviron[] = { NULL };
    newargv[0]="/sbin/iptables";
    newargv[8]="192.168.57.2";

    process_id=fork();
    if(process_id <0){
        fprintf(stderr, "Gagal membuat process
baru.\n");
        exit(1);
    }
    if(process_id >0){
        printf("SUCCESS: process id dari child
```

```

process adalah %d \n",process_id);
    exit(0);
}
umask(0);
sid=setsid();
if(sid < 0){
    exit(1);
}
chdir("/");
close(STDIN_FILENO);
close(STDOUT_FILENO);
close(STDERR_FILENO);

if((fplog=fopen("logaddr.log","w"))==NULL){
    fprintf(stderr, "Gagal membuat file
/logaddr.log");
    exit(1);
}
while(1){
    sleep(2);

    if((fpproc=fopen("/proc/knockdir/knockaddr","r"))
==NULL){
        fprintf(fplog,"Gagal membuka file
/proc.\n");
    }
    while(fscanf(fpproc,"%s %s
%s",datalog.status,datalog.ip,datalog.mac)!=EOF) {
        //fprintf(fplog,"Status :
%s\n",datalog.status);
        //fprintf(fplog,"IP      :
%s\n",datalog.ip);
        //fprintf(fplog,"MAC      :
%s\n",datalog.mac);
        //fflush(fplog);
        buka=strcmp(datalog.status,"active");

        tutup=strcmp(datalog.status,"deactive");
        if(status_iptables==-1){
            status_iptables=0;
        }else if(buka==0 &&
status_iptables==1){
            system("/sbin/iptables -F");
            system("/sbin/iptables -P INPUT
DROP");
            system("/sbin/iptables -P FORWARD
ACCEPT");
        }
    }
}

```

```
        system("/sbin/iptables -P OUTPUT  
ACCEPT");  
        newargv[6]=datalog.ip;  
        execve(newargv[0], newargv,  
newenviron);  
        status_iptables=0;  
    }else{  
        if(tutup==0 &&  
status_iptables==0){  
            system("/sbin/iptables -F");  
            system("/sbin/iptables -P  
INPUT DROP");  
  
            system("/sbin/iptables -P FORWARD DROP");  
  
            system("/sbin/iptables -P OUTPUT DROP");  
  
            status_iptables=1;  
        }  
    }  
}  
fclose(fpproc);  
fclose(fplog);  
return 0;  
}
```

**LAMPIRAN C : Kode Lengkap Aplikasi *Client* (pk\_client.pl)**

```
#!/usr/bin/perl

#####
# Practical Extraction and Report Language #
#
# Created by : Rois Awang R.                 #
# License     : GPL                          #
# Filename    : pk_client.pl                 #
# #
#####

$argc=@ARGV;

if($argc!=4 || $ARGV[0] !~ "-a" || $ARGV[2] !~ "-p") {
    print "Gunakan perintah : \nperl pk_client.pl -a
<ip_address> -p x,x,x,x\ndengan x=port \n";
    die("Error !!\n");
}

$status=validasi($ARGV[1],$ARGV[3]);

if($status==1) {
    die("IP address $ARGV[1] salah.\n");
}elsif ($status==2) {
    die("Port $ARGV[3] tidak tersedia.");
}else{
    if($status==3) {
        die("Salah, Urutan port harus berjumlah
4.\n");
    }
}

@port_seq=split(",",$ARGV[3]);
```

```

foreach $p (@port_seq) {
    system("nc", "-w", "1", "$ARGV[1]", "$p");
}

print "Proses scanning port.....\n";
system("nmap -P0 $ARGV[1] > nmap_temp.txt");
open(BERKAS, "<nmap_temp.txt");
$port_ssh="22/tcp";
while(<BERKAS>){
    if(m/$port_ssh/){
        $status_ssh=1;
    }
}

if($status_ssh){
    print "Port SSH terbuka.\n";
    print "Masukkan username remote server : ";
    $user_remote=<STDIN>;
    chomp($user_remote);
    system("ssh -l $user_remote $ARGV[1]");
} else{
    print "Port SSH tertutup\n";
}

unlink ("nmap_temp.txt");
close(BERKAS);

sub validasi {
    #validasi ip address

    @ip_array=split(/\./, @_ [0]);
    $jml_digit_ip=@ip_array;
    foreach $x (@ip_array){
        $_= $x;
        if(( $x <0 || $x>255) || (m/[^\d]/) || $jml_digit_ip !=4) {
            return 1;
        }
    }

    #validasi port

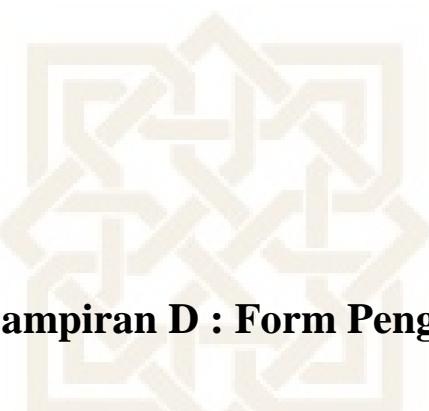
    @ports=split(", ", @_ [1]);
    $jml_port = @ports;

    foreach $x (@ports) {

```

```
$_= $x;
if (($x < 0 || $x > 65535) || ($m / [^0-9] /)) {
    return 2;
}elsif ($jml_port != 4) {
    return 3;
}
return 0;}
```





**Lampiran D : Form Pengujian**



## FORM PENGUJIAN

### PERANCANGAN DAN IMPLEMENTASI AUTENTIKASI REMOTE SERVER DENGAN MENGGUNAKAN METODE PORT KNOCKING BERBASIS LOADABLE KERNEL MODULE

---

Nama :

Pekerjaan :

Instansi :

**Petunjuk:** Berilah tanda  pada salah satu kolom penilaian SB (sangat baik), B (baik), TB (tidak baik) atau STB (sangat tidak baik) berdasarkan pernyataan yang telah disediakan.

#### **Pengujian Fungsionalitas Sistem**

No	Pernyataan	Penilaian			
		SB	B	TB	STB
1	Aplikasi <i>server</i> dapat ter-load dalam <i>kernel</i>				
2	Aplikasi <i>server</i> dapat memonitor paket jaringan.				
3	Aplikasi <i>server</i> dapat menyeleksi <i>port-port</i> untuk autentikasi <i>port knocking</i> .				
4	Berkas log pada /proc dapat terbentuk dan berfungsi sebagaimana mestinya				
5	Aplikasi <i>daemon</i> dapat terpasang dan berfungsi mengaktifkan maupun menutup <i>firewall</i>				
6	Aplikasi <i>client</i> dapat mengirim <i>port</i> untuk autentikasi				
7	Aplikasi <i>server</i> , <i>daemon</i> , serta berkas /proc dapat dihapus dari memori ketika aplikasi-aplikasi tersebut selesai digunakan				

Yogyakarta, September 2013

(.....)

## CURRICULUM VITAE



Nama : Rois Awang Rimbayani  
Tempat Tanggal Lahir: Magetan, 22 Oktober 1987  
Jenis Kelamin : Laki-Laki  
Agama : Islam  
Nama Ayah : Suyanto  
Nama Ibu : Siti Barokah  
Alamat Asal : Ds. Bulak RT 16 RW 08 Kec. Bendo, Kab. Magetan, Jawa Timur  
Email : [open.sentinel@gmail.com](mailto:open.sentinel@gmail.com)

### Riwayat Pendidikan :

1. SDN Sumengko I, Randublatung, Blora, Jawa Tengah (1993-1999)
2. MTsN Bibrik, Madiun. (1999-2002)
3. MAN 2 Madiun (2002-2005)
4. Wearnies Education Center Madiun (2005-2006)
5. UIN Sunan Kalijaga, Yogyakarta (2007-2013).