

**IMPLEMENTASI HONEYPOT DENGAN SMS GATEWAY SEBAGAI  
LAPORAN INTRUSION DETECTOR**

**TUGAS AKHIR**

**Diajukan sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana  
Jurusan Teknik informatika**



**OLEH :**

**NAMA : Cahya Adhi Setya Nugraha  
NO. MAHASISWA : 07523154**

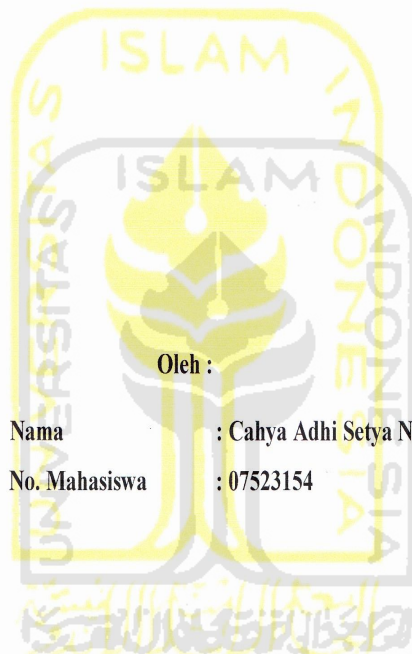
**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2012**

# LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI HONEYPOT DENGAN SMS GATEWAY SEBAGAI  
LAPORAN INTRUSION DETECTOR

TUGAS AKHIR



Oleh :

Nama : Cahya Adhi Setya Nugraha  
No. Mahasiswa : 07523154

Yogyakarta, 8 Maret 2012

Pembimbing

A handwritten signature in black ink, appearing to read 'Svarif Hidayat'.

Svarif Hidayat, S.Kom., MIT

# LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI HONEYPOT DENGAN SMS GATEWAY SEBAGAI  
LAPORAN INTRUSION DETECTOR

## TUGAS AKHIR

Oleh :

Nama : Cahya Adhi Setya Nugraha

No. Mahasiswa : 07523154

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Teknik informatika Fakultas  
Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 28 Maret 2012

Tim Penguji

Syarif Hidayat, S.Kom., MIT  
Ketua

Dr. R. Teduh Dirgahayu, ST., M. Sc  
Anggota I

Affan Mahtarami, S.Kom., MT  
Anggota II



Mengetahui,  
Ketua Jurusan Teknik informatika  
Universitas Islam Indonesia

Andi Prayudi, S.Si, M.Kom.

## LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

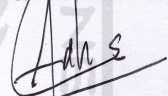
Nama : Cahya Adhi Setya Nugraha

No. Mahasiswa : 07523154

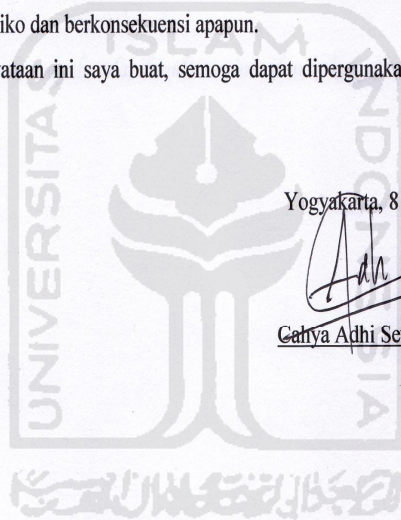
Menyatakan bahwa seluruh komponen isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, maka saya siap menanggung resiko dan berkonsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagai mana mestinya.

Yogyakarta, 8 Maret 2012



Cahya Adhi Setya Nugraha



## PERSEMBAHAN

*Untuk:*

*Kesembuhan ayahanda, ibunda tercinta dan kakak serta adikku*



## MOTTO

*" Allah telah menciptakan manusia dalam bentuk yang paling sempurna, namun ia tidak mudah bertahan dalam kesempurnaannya. Boleh jadi ia kembali (mati) dalam posisi yang sangat hina, kecuali yang menjaga dirinya dengan beriman dan beramal saleh "*

*( Mukadimah QS : AT TIIN )*

*"Jalani hidup dengan penuh keyakinan dan tidak lupa berikhtiar dan bertawakal kepada Allah SWT"*



## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Assalamu'alaikum. Wr. Wb*

Dengan mengucapkan Alhamdulillah, puji dan syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan tugas akhir ini, yang berjudul **“implementasi honeypot dengan sms gateway sebagai laporan intrusion detector”** dengan baik.

Laporan tugas akhir ini disusun untuk melengkapi salah satu syarat guna memperoleh gelar Sarjana Teknik informatika pada Universitas Islam Indonesia dan atas apa yang telah diajarkan selama perkuliahan baik teori maupun praktek, disamping laporan itu sendiri yang merupakan rangkaian kegiatan yang harus dilakukan setelah tugas akhir ini selesai.

Penulisan dan penyelesaian tugas akhir ini tidak lepas dari saran, bimbingan, dukungan serta bantuan dari berbagai pihak. Untuk itu pada kesempatan kali ini penulis menyampaikan ucapan terimakasih kepada :

1. Allah SWT. Atas segala hidayah, barokah dan taufiq-Nya
2. Bapak Gumbolo selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan Teknik informatika.
4. Bapak Syarif Hidayat, S.Kom., MIT selaku Dosen Pembimbing Tugas Akhir. Terima kasih atas segala bantuan dan dukungan yang telah diberikan kepada penulis dalam penyusunan skripsi ini.
5. Kedua orangtua yang selalu melimpahkan kasih sayang yang tulus, doa yang tiada henti-hentinya, serta dukungan yang begitu besar.
6. Eko Cahyono dan Rizqi Bagus Pamungkas, kedua saudaraku tercinta yang senantiasa memberikan dukungannya.

7. Beti Rahmasari Utami yang senantiasa terus mendukung saya, para “koplak crew dan GCS”, teman-teman LPM PROFESI FTI UII dan INCLUDE, Lantip Agil Nugroho, Yoga Febriansyah yang turut membantu saya dalam menyelesaikan tugas akhir ini.
8. Bapak R. Ratna Dewa yang telah mengenalkan dunia jaringan kepada saya dan teman-teman Cisco Networking Academy yang telah memberikan dukungan kepada saya.
9. Fauzi Arief T, Lukman Ikhwanurrahman, Yulis Kurniawan, Leyne Khurniawati, Shouma prameswari, Arsi Dea Iriani, Devi Latifahadi, Afif Tri Pudyastuti dan Noviana, terimakasih untuk semua dukungan dan doanya. “Never say goodbye cause i am sure i will meet you again”.
10. Semua pihak yang telah membantu dalam pembuatan hingga terselesaikannya tugas akhir ini, yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari dalam penulisan laporan tugas akhir ini masih jauh dari sempurna, karena keterbatasan kemampuan dan pengalaman. Penulis mengharapkan saran dan kritik yang bersifat membangun untuk memperbaiki tugas akhir ini semoga dapat bermanfaat bagi penulis khususnya dan pembaca pada umumnya.

*Wassalamu'alaikum Wr. Wb*

Yogyakarta, 8 Maret 2012

Cahya Adhi Setya Nugraha



## SARI

Pada sistem jaringan komputer, sistem keamanan merupakan hal sangat mutlak yang harus diterapkan. Informasi nyata akan sebuah serangan merupakan salah satu acuan dalam pembangunan sistem keamanan. *Honeypot* adalah *software* yang digunakan untuk mengalihkan dan menjebak tindakan tidak terotorisasi pada jaringan komputer. *Honeyd* adalah *virtual daemon* yang menciptakan *virtual host* pada jaringan komputer. *Honeypot* yang berbasis *honeyd* mampu memberikan informasi nyata mengenai serangan pada jaringan komputer. Apalagi jika dikonfigurasi dengan *sms gateway*, informasi tersebut dapat tersampaikan dengan cepat.

*Sms gateway* merupakan *software* yang digunakan untuk menghubungkan operator selular dengan *internet*. Pengkonfigurasi *honeyd* dengan *sms gateway* akan bekerja jika terdapat serangan yang masuk didalamnya. Hasil pengiriman informasi serangan akan dikirimkan melalui *sms* dan dapat dilihat setiap saat melalui *handphone*. Untuk membuktikan hal tersebut perlu dilakukan pengujian dengan melakukan serangan dengan teknik *scanning*, *brute force* dan *os fingerprinting*.

Hasilnya semua aktifitas penyerangan dari ketiga teknik tersebut dapat direkam oleh *log file honeyd*. Pola yang dihasilkan berbeda-beda dan mempermudah dalam proses analisa sebuah serangan. Dari informasi yang ada dalam *log file* akan terkirim melalui *sms* yang hanya menampilkan *protocol* dan *service* yang diakses, sehingga tercipta *alert* adanya paket yang tidak terotorisasi pada jaringan komputer.

Kata kunci : *Honeypot*, *honeyd*, *sms gateway*, *log file*

## TAKARIR

*Virtual Environment*

*Server yang dibentuk pada konfigurasi honeypot yang digunakan untuk mengelabui dan mengalihkan perhatian intruder.*

*Intruder*

*Seseorang yang mencoba masuk ke dalam sistem dan menggunakannya untuk suatu tindakan yang seharusnya bukan menjadi kewenangannya.*

*Logging*

*Kemampuan dari honeypot dalam mencatat semua traffic yang tertuju pada virtual environment*

*Field*

*Huruf, angka dan karakter dalam baris log file yang dipisahkan dengan spasi*



## DAFTAR ISI

<b>LEMBAR PENGESAHAN PEMBIMBING</b> .....	ii
<b>LEMBAR PENGESAHAN PENGUJI</b> .....	iii
<b>LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR</b> .....	iv
<b>PERSEMBAHAN</b> .....	v
<b>MOTTO</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>SARI</b> .....	ix
<b>TAKARIR</b> .....	x
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiv
<b>DAFTAR TABEL</b> .....	xvi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian .....	3
1.6 Metodologi Penelitian .....	4
1.6.1 Metode Pengumpulan Data .....	4
1.6.2 Metode Observasi .....	4
1.6.3 Implementasi dan Konfigurasi Sistem .....	4
1.7 Sistematika Penulisan .....	5
<b>BAB II LANDASANTEORI</b> .....	7
2.1 Jaringan Komputer.....	7
2.2 Serangan Jaringan Komputer .....	9
2.2.1 Jenis-jenis Serangan.....	9

2.3	Honeypot.....	11
2.3.1	Kategori <i>Honeypot</i> .....	12
2.3.2	Jenis Honeypot .....	13
2.3.3	Kelebihan dan Kekurangan Honeypot.....	14
2.3.4	Nilai guna Honeypot.....	15
2.4	Honeyd .....	18
2.5	Farpd .....	20
2.6	Gammu.....	20
2.7	Regular Expression.....	21
<b>BAB III METODOLOGI.....</b>		<b>23</b>
3.1	Analisis Masalah.....	23
3.2	Topologi Jaringan.....	24
3.3	Desain Alur Data .....	25
3.4	Analisi dan Persiapan Kebutuhan Sistem .....	26
3.4.1	Kebutuhan Perangkat Keras.....	26
3.4.2	Kebutuhan Perangkat Lunak.....	27
3.5	Instalasi dan Konfigurasi .....	28
3.5.1	Sistem Operasi.....	28
3.5.2	Honeyd .....	28
3.5.3	Farpd.....	31
3.5.4	Gammu.....	31
3.5.5	Konfigurasi <i>Honeyd</i> dengan <i>Gammu</i> .....	32
3.6	Pengujian Sistem .....	32
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>		<b>34</b>
4.1	Implementasi Sistem.....	34
4.1.1	Honeyd .....	34
4.1.2	Farpd.....	38
4.1.3	Gammu.....	38
4.1.4	Konfigurasi <i>Honeyd</i> dengan <i>Gammu</i> .....	41

4.2	Pengujian Sistem .....	45
4.2.1	Teknik Scanning .....	45
4.2.2	Teknik Brote Force .....	48
4.2.3	Teknik OS Fingerprinting .....	49
4.3	Hasil Penelitian.....	51
4.3.1	Hasil Pengujian Teknik Scanning .....	52
4.3.2	Hasil Pengujian Teknik Brute force .....	56
4.3.3	Teknik OS Fingerprinting .....	58
4.4	Pembahasan.....	59
4.4.1	Skenario Pengujian .....	60
4.4.2	Performa <i>Honeyd</i> .....	61
4.4.3	Log file.....	63
4.4.4	Pengiriman SMS.....	69
<b>BAB V</b>	<b>PENUTUP</b> .....	<b>71</b>
5.1	Kesimpulan.....	71
5.2	Saran .....	71



## DAFTAR GAMBAR

Gambar 2.1	<i>End device dan Intermediary device</i> .....	7
Gambar 2.2	<i>Network Media</i> .....	8
Gambar 2.3	<i>Network services</i> .....	8
Gambar 2.4	<i>Scaning jaringan komputer</i> .....	9
Gambar 2.5	Penempatan <i>honeypot</i> dengan <i>firewall</i> dalam sistem keamana..	12
Gambar 2.6	Topologi <i>high interaction honeypot</i> .....	13
Gambar 2.7	Topologi <i>low Interaction honeypot</i> .....	14
Gambar 2.8	Penempatan <i>honeypot</i> dekat dengan jaringan publik.....	17
Gambar 2.9	Penempatan <i>honeypot</i> didalam DMZ.....	18
Gambar 2.10	Penempatan <i>honeypot</i> dibelakang <i>gateway</i> .....	18
Gambar 2.11	<i>Unused ip address</i> .....	19
Gambar 3.1	Rancangan topologi <i>honeyd</i> .....	24
Gambar 3.2	Alur proses yang berjalan pada <i>honeyd</i> .....	25
Gambar 4.1	<i>Modem</i> telah terdeteksi.....	39
Gambar 4.2	<i>Form</i> konfigurasi <i>gammu</i> .....	39
Gambar 4.3	<i>Gammu</i> dapat membaca <i>modem</i> yang digunakan.....	40
Gambar 4.4	<i>Sms gateway</i> dapat mengirimkan <i>sms</i> .....	40
Gambar 4.5	Hasil <i>scaning network</i> 192.168.1.0/28.....	45
Gambar 4.6	hasil <i>scaning port</i> pada <i>server</i> palsu 192.168.1.6.....	46
Gambar 4.7	Pengaksesan <i>service ftp</i> pada <i>server</i> palsu.....	46
Gambar 4.8	Pengaksesan <i>service telnet</i> pada <i>server</i> palsu.....	47
Gambar 4.9	Pengaksesan <i>service ssh</i> pada <i>server</i> palsu.....	47
Gambar 4.10	Pengaksesan <i>service http</i> pada <i>server</i> palsu.....	48
Gambar 4.11	<i>Brutus alert</i> untuk melakukan serangan <i>brute force</i> .....	48
Gambar 4.12	Serangan <i>brute force</i> .....	49
Gambar 4.13	<i>Xprobe2</i> menjalankan modulnya untuk proses identifikasi.....	50
Gambar 4.14	Hasil dari serangan <i>os fingerprinting</i> .....	51

Gambar 4.15	Paket TCP yang tidak terotorisasi pada log file honyd.....	52
Gambar 4.16	Paket TCP yang dikirim pada proses <i>scanning</i> .....	54
Gambar 4.17	Layanan yang diakses pada saat proses <i>scanning</i> .....	54
Gambar 4.18	<i>Sms</i> paket tcp pada serangan <i>brute force</i> .....	57
Gambar 4.19	<i>Sms</i> layanan <i>telnet</i> yang diserang dengan teknik <i>brute force</i> ....	58
Gambar 4.20	<i>Protocol</i> yang diakses pada serangan <i>os fingerprinting</i> .....	59
Gambar 4.21	<i>Honeyd</i> mampu menjalankan layanan ICMP.....	61
Gambar 4.22	<i>Farpd</i> mengkalim <i>ip address</i> 192.168.1.6.....	62
Gambar 4.23	<i>Sms</i> yang menginformasikan protokol dan <i>service</i> yang diakses.....	70



## DAFTAR TABEL

Tabel 2.1	Tabel POSIX <i>regular expression</i> .....	22
Tabel 4.1	Tabel keterangan <i>log file honeyd</i> .....	64





# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Perkembangan teknologi informasi pada saat ini menunjukkan kemajuan yang cukup pesat. Salah satu perkembangan dari teknologi tersebut adalah jaringan *internet*. Pada zaman sekarang ini, hampir semua orang menggunakan *internet* dalam segala aspek kehidupannya. Melalui jaringan tersebut masyarakat dapat berkomunikasi, bertukar data dan berbagi informasi antara yang satu dengan yang lainnya.

Dengan adanya *internet*, masyarakat akan sangat diuntungkan bila tahu bagaimana cara mengelolanya dan bahkan mampu menciptakan lapangan kerja sendiri. Hal ini mampu menarik perhatian masyarakat dari berbagai lapisan untuk mencoba memanfaatkannya. Dari segi pemanfaatannya, *internet* dapat digunakan untuk kepentingan kerja, sekolah, bisnis, atau kepentingan yang lain. Akan tetapi, banyak pula yang memanfaatkan *internet* untuk kepentingan yang dapat merugikan orang lain.

Dari berbagai macam manfaat jaringan *internet* yang begitu besar diperlukan adanya aspek keamanan. Keamanan ini bertujuan untuk mengamankan data penting yang ada didalamnya. Apalagi yang berkaitan dengan produktifitas, sangat dibutuhkan pengamanan yang mampu mengamankan data-datanya. Untuk mengatasi hal tersebut, perlu digunakan *software* yang mampu menjaga keamanan data. Salah satu *software* yang dapat membantu menjaga sistem keamanan data tersebut adalah *Honeypot*.

*Honeypot* adalah suatu program yang dapat berjalan pada jaringan komputer yang dirancang untuk mengalihkan dan menjebak usaha-usaha yang tidak terotorisasi yang masuk ke dalam sebuah jaringan komputer. Sistem *honeypot* mampu membentuk *virtual environment* yang mampu menjalankan *service* mirip dengan sistem aslinya. *Honeypot* akan bekerja jika ada serangan yang masuk didalamnya. Jadi, bila ada seseorang yang melakukan percobaan

serangan dan masuk ke dalam *honeypot* seolah-olah dia telah berhasil masuk dan mengambil alih sistem tersebut. Akan tetapi, pada kenyataannya tidak masuk ke dalam sistem yang sebenarnya.

Sistem *honeypot* memang sengaja dibuat untuk dijadikan target serangan, sehingga sengaja dibuat lemah. Kelemahan sistem *honeypot* digunakan untuk menganalisa dan mempelajari paket-paket yang tidak terotorisasi yang terekam didalam *log file honeypot*. Dari kelemahan sistem tersebut dapat membebaskan *intruder* berkreasi di dalamnya. *Intruder* adalah seseorang yang mencoba masuk ke dalam sistem dan menggunakannya untuk suatu tindakan yang seharusnya bukan menjadi kewenangannya. *Intruder* dalam melakukan serangan pasti memiliki maksud tujuan tertentu. Melalui *honeypot* ini administrator dapat mengetahui dan mempelajari maksud dan tujuan dari *intruder* dalam melakukan penetrasi pada jaringan komputer.

Dengan menggunakan *honeypot*, administrator akan merasa lebih terbantu apabila informasi mengenai paket yang tidak terotorisasi dapat diketahui dengan cepat. Informasi tersebut dapat diketahui melalui sms yang dikirim dengan menggunakan *sms gateway*. *Sms gateway* merupakan *software* yang digunakan untuk menghubungkan operator selular dengan *internet*. Dengan demikian, administrator dapat mengetahui informasi paket yang tidak terotorisasi dengan cepat melalui *sms* yang dikirim secara otomatis ke HP-nya.

Dari penjelasan di atas dapat diketahui gambaran kinerja dari *honeypot*. *Honeypot* dapat mengalihkan perhatian dan menjebak para *intruder* yang akan mencoba masuk ke dalam sistem jaringan komputer. Pengintegrasian *honeypot* dengan *sms gateway* dapat memberikan nilai *plus* tersendiri. Administrator dapat mengetahui lewat *sms* yang terkirim ke handphonenya mengenai adanya paket tidak terotorisasi yang masuk ke dalam sistem *honeypot*. Pengintegrasian tersebut menguntungkan administrator untuk mengamati dan melakukan pengamanan pada *server* sebenarnya.

## 1.2 Rumusan Masalah

Adapun rumusan masalah dari latar belakang di atas adalah bagaimana cara menganalisa paket-paket yang tidak terotorisasi yang terekam didalam *honeypot* dan mengirimkannya dengan menggunakan *sms gateway*.

## 1.3 Batasan Masalah

Pembahasan masalah akan lebih terperinci ketika terdapat batasan dalam penyusunan Tugas Akhir ini. Dalam hal ini terdapat beberapa batasan masalah yaitu sebagai berikut:

1. Konfigurasi *honeypot* untuk membentuk *virtual environment* (*server* palsu) yang berbasis *windows* dan *linux*.
2. Pengimplementasian *honeypot* tidak digabungkan dengan *firewall* dan IDS.
3. Pengujian terhadap *honeypot* dilakukan dengan menggunakan *tools exploit* dengan teknik *scanning*, *brute force* dan *os fingerprinting*.
4. Pengiriman *log file honeypot* melalui *sms* dan hanya menampilkan *protocol* dan *service* yang diakses.

## 1.4 Tujuan Penelitian

Tujuan penelitian ini adalah menganalisa dan mengimplementasikan *honeypot* agar dapat merekam paket yang tidak terotorisasi dan mengirimkan melalui *sms*.

## 1.5 Manfaat Penelitian

Manfaat yang dapat diberikan dalam penyelesaian tugas akhir ini adalah :

1. Administrator jaringan dapat mempelajari aktivitas yang terekam dalam *log file honeypot*.
2. Administrator dapat melakukan tindakan pencegahan agar tidak berdampak buruk pada *server* sebenarnya.
3. Memberikan peringatan dini adanya *intruder* yang mencoba melakukan penetrasi kedalam jaringan komputer.

## 1.6 Metodologi Penelitian

Metode yang penulis gunakan bertujuan agar hasil dari penelitian dan analisa tersebut lebih terarah serta data yang di peroleh lebih tepat. Kelengkapan data yang di peroleh dapat memberikan kontribusi bagi penulis dalam menyusun skripsi ini. Adapun beberapa metode yang digunakan antara lain:

### 1.6.1 Metode Pengumpulan Data

Metode pengumpulan data yang dipakai adalah melakukan studi pustaka dengan menggunakan referensi dari buku dan literature yang ada di *internet* yang dapat membantu dalam memecahkan masalah yang ada serta melakukan konsultasi secara berkesinambungan dengan dosen pembimbing.

### 1.6.2 Metode Observasi

Pengumpulan data secara langsung pada objek yang diteliti untuk memperoleh data yang tepat serta melakukan analisis masalah untuk menjawab permasalahan yang telah disebutkan pada rumusan masalah.

### 1.6.3 Implementasi dan Konfigurasi Sistem

Metode implementasi disusun berdasarkan hasil perolehan dari metode Pengumpulan data dan metode observasi:

- a. Analisis Masalah  
Tahap ini merupakan tahap pengamatan dan perincian masalah yang ada pada proses implementasi.
- b. Desain Topologi Jaringan  
Tahap ini merupakan tahap perancangan topologi jaringan komputer yang akan diterapkan dalam pengimplementasian *honeypot*.
- c. Desain Alur data proses aplikasi  
Tahap ini merupakan perancangan visualisasi data yang mengalir pada proses yang terjadi pada dari awal hingga akhir.
- d. Kebutuhan Perangkat Keras  
Tahapan ini merupakan tahap pengadaan perangkat keras yang meliputi komputer untuk berjalannya *honeypot* dan *sms gateway*.

e. Kebutuhan Perangkat Lunak

Tahapan ini merupakan tahap pengadaan perangkat lunak meliputi sistem operasi *linux* sebagai tempat berjalannya aplikasi *honeypd* dan *sms gateway* dan *software* lain untuk mendukung jalannya kedua aplikasi tersebut.

f. Instalasi dan Konfigurasi

Tahapan ini merupakan tahap instalasi dan konfigurasi dari masing-masing aplikasi yang akan digunakan termasuk sistem operasi yang dibutuhkan seperti *honeypd*, *farpd* dan *gammu* dan mengkonfigurasi *honeypd* dengan *sms gateway* untuk dapat melakukan otomatisasi *sms*.

g. Pengujian

Tahapan ini digunakan untuk melakukan pengujian pada *honeypd* agar dapat mengetahui apakah *honeypd* telah bekerja pada jaringan dan merekam paket yang tidak terotorisasi dan untuk mengetahui apakah *sms gateway* dapat mengirimkan paket-paket tidak terotorisasi tersebut.

## 1.7 Sistematika Penulisan

Sistematika yang digunakan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut :

### **BAB I Pendahuluan**

Bab ini memuat latar belakang yang menyebabkan munculnya permasalahan pengamanan sistem yang terhubung dengan jaringan dan batasan masalah yang dipergunakan, tujuan penelitian, manfaat penelitian, metodologi penelitian dan sistematika penulisan.

### **BAB II Landasan Teori**

Bab ini berisi teori yang mendasari penyusunan skripsi dan juga membahas dasar-dasar teori yang berhubungan dengan penelitian berupa teori jaringan komputer, serangan jaringan komputer, konsep *honeypot* dan *sms gateway*.

### **BAB III Metodologi Penelitian**

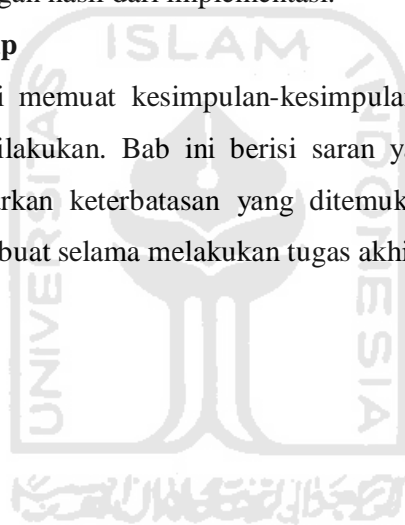
Bab ini memuat urian tentang analisis masalah, topologi jaringan, alur sistem, analisis kebutuhan sistem yang mencakup kebutuhan perangkat keras dan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir, instalasi dan konfigurasi serta pengujian sistem.

### **BAB IV Hasil dan Pembahasan**

Bab ini menjelaskan hasil dari penelitian dan pembahasannya. Pada bab ini juga memuat dokumentasi hasil implentasi keterangan hasil dari implementasi.

### **BAB V Penutup**

Bab ini memuat kesimpulan-kesimpulan dari penelitian yang telah dilakukan. Bab ini berisi saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan tugas akhir.



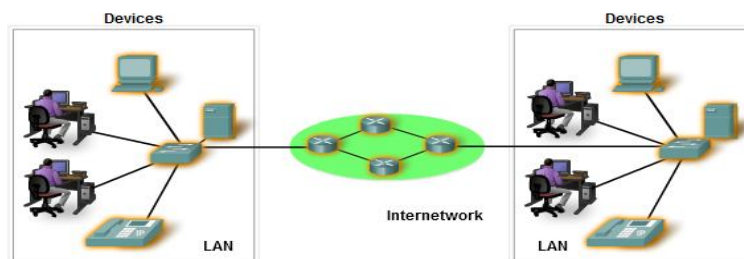
## BAB II

### LANDASAN TEORI

#### 2.1 Jaringan Komputer

Jaringan komputer adalah sekelompok komputer yang saling terhubung dengan menggunakan protokol komunikasi melalui media transmisi sehingga dapat saling berbagi data, aplikasi dan penggunaan *hardware* secara bersamaan. Dalam proses pengiriman data, jaringan komputer memiliki komponen yang digunakan untuk berjalannya proses pengiriman paket data dari sumber menuju tujuan. Komponen tersebut antara lain *devices*, media dan *services* [CIS11].

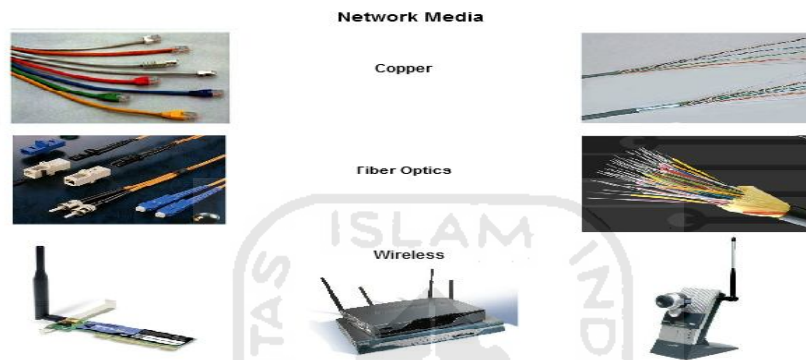
*Devices* merupakan komponen yang berbentuk fisik atau perangkat keras jaringan. Ada beberapa penggolongan *devices* dalam jaringan komputer diantaranya *end devices* dan *intermediary devices*. *End devices* adalah perangkat yang membentuk antarmuka antara pengguna (*user*) dengan sistem. Contoh *end devices* seperti laptop, PC, *server*, *ip phone*, kamera, *printer*. *Intermediary devices* adalah perangkat perantara untuk menyediakan konektivitas untuk memastikan aliran data yang melalui jaringan. Beberapa contoh *intermediary devices* seperti *hub*, *switch*, *wireless*, *router*, *modem* dan *firewall* [CIS11]. Gambaran umum *devices* dapat dilihat pada gambar 2.1.



**Gambar 2. 1** *End devices* dan *Intermediary devices*

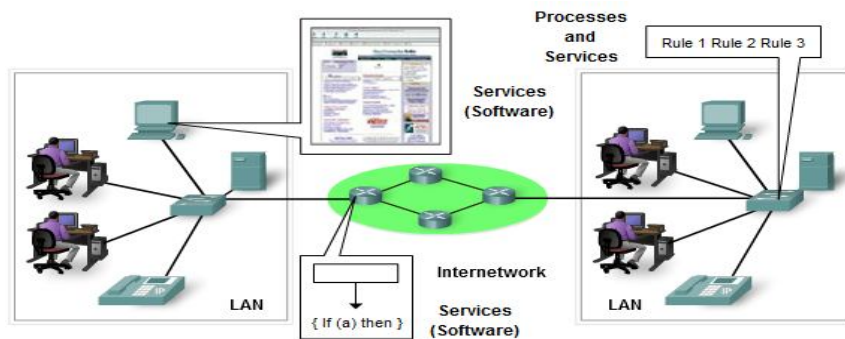
Media merupakan perangkat keras dalam jaringan komputer. Fungsi dari media adalah sebagai jalur yang digunakan untuk menghubungkan *end devices* dan berjalannya paket data dalam jaringan dari sumber menuju tujuan. Dalam jaringan komputer menggunakan tiga jenis media yaitu logam, serat optik dan

nirkabel. Sinyal yang dipancarkan oleh setiap media berbeda-beda. Pada media logam sinyal dipancarkan diubah menjadi impuls listrik dengan pola tertentu. Media serat optik sinyal dipancarkan di ubah dalam rentang cahaya sedangkan pada media nirkabel sinyal yang dipancarkan berupa gelombang elektromagnetik [CIS11]. Gambaran umum media dapat dilihat pada gambar 2.2.



**Gambar 2. 2** *Network Media*

*Services* atau layanan merupakan komponen jaringan yang berbentuk perangkat lunak yang berjalan pada media jaringan dan diproses oleh devices. *Service* jaringan memberikan informasi kepada *user* sebagai tanggapan terhadap permintaan. *Services* yang sering digunakan dalam kehidupan sehari-hari antara lain seperti *email*, *hosting* dan *web hosting* termasuk juga pemrosesan paket data oleh *intermediary device* yang berjalan di jaringan [CIS11]. Gambaran umum *services* dapat dilihat pada gambar 2.3.

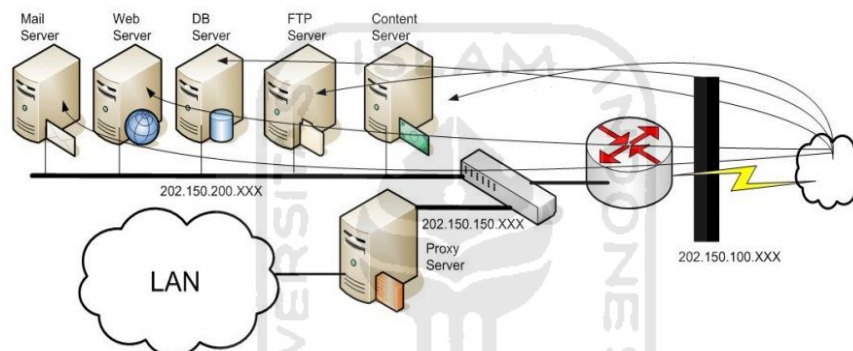


**Gambar 2. 3** *Network services.*



## 2.2 Serangan Jaringan Komputer

Komputer yang terhubung dengan jaringan komputer pasti tidak akan luput dari serangan jaringan komputer. Serangan jaringan komputer adalah segala aktivitas yang berjalan di dalam jaringan komputer dimana aktivitas tersebut bertentangan dengan aturan yang diterapkan. Serangan dapat terjadi kapan saja dan dalam bentuk yang bervariasi pada jaringan lokal maupun jaringan yang lebih besar (*internet*). Gambar 2.4 menjelaskan salah satu serangan pada jaringan komputer.



**Gambar 2.4** Serangan jaringan komputer.

Banyak sekali motif serangan yang dilakukan dan motif tersebut terbagi menjadi dua jenis, yaitu motif intelektual dan motif kejahatan [KUR11]. Serangan *intruder* dengan motif intelektual pada umumnya tidak menimbulkan kerugian yang terlalu besar. *Intruder* tersebut hanya ingin mengetahui seberapa besar kemampuan dan keberaniannya dalam menembus sistem keamanan jaringan. Pada motif kejahatan, kerugian yang ditimbulkan amat sangat besar. Motif kejahatan ini dapat merambah aspek ekonomi dan politik atau kriminal yang dapat merugikan banyak orang dan memicu *cyber war*.

### 2.2.1 Jenis-jenis Serangan

Jenis-jenis serangan menurut *National Security Agency* (NSA) dalam dokumen *Information Assurance Technical Framework* (IATF) menggolongkan lima jenis ancaman pada jaringan Komputer [WAK11]. Kelima ancaman itu adalah :

#### 1. Serangan Pasif

Tipe dari serangan pasif adalah analisa *traffic*, memonitor komunikasi terbuka, memecah kode *traffic* yang dienkrpsi dan menangkap informasi seperti *password*. Bagi *intruder*, menangkap secara pasif data-data di jaringan ini bertujuan mencari celah sebelum menyerang. Lewat serangan tersebut bisa memaparkan informasi atau data tanpa sepengetahuan pemiliknya.

#### 2. Serangan Aktif

Tipe serangan ini berupaya membongkar sistem keamanan, misalnya dengan memasukan kode-kode berbahaya (*malicious code*), mencuri atau memodifikasi informasi. Sasaran serangan aktif ini termasuk penyusupan ke jaringan *backbone*, eksploitasi informasi, penetrasi elektronik. Serangan aktif ini selain mengakibatkan terpaparnya data, juga *denial-of-service*, atau modifikasi data.

#### 3. Serangan jarak dekat

Dalam jenis serangan ini, *intruder* secara fisik berada dekat dari perangkat jaringan. Serangan ini bertujuan memodifikasi, mengumpulkan atau memblok akses pada sumber informasi. Tipe serangan jarak dekat ini biasanya dilakukan dengan masuk ke lokasi secara langsung dan tidak sah.

#### 4. Orang dalam

Serangan oleh orang di dalam organisasi ini dibagi menjadi sengaja dan tidak sengaja. Jika dilakukan dengan sengaja, tujuannya untuk mencuri, merusak informasi, menggunakan informasi untuk kejahatan. Serangan orang dalam yang tidak disengaja lebih disebabkan karena kecerobohan pengguna, tidak ada maksud jahat dalam tipe serangan ini.

#### 5. Serangan distribusi

Tujuan serangan ini adalah memodifikasi peranti keras atau peranti lunak pada saat produksi di pabrik sehingga bisa disalahgunakan di kemudian hari. Dalam serangan ini, *intruder* menyisipkan sejumlah kode ke dalam produk sehingga membuka celah keamanan yang bisa dimanfaatkan untuk tujuan ilegal.

### 2.3 Honeypot

Dalam dunia keamanan jaringan nama *honeypot* sudah tidak asing lagi. Banyak para profesional yang sangat tertarik pada *honeypot* karena seorang akan dapat melihat informasi secara nyata tentang serangan. Ada beberapa definisi *honeypot* yang disampaikan oleh profesional yang berkompeten dalam penelitian *honeypot*. Menurut *Lance Spitzner*, seorang arsitek keamanan *Sun Microsystems* sebagai berikut:

*"A honeypot is a security resource who's value lies in being probed, attacked, or compromised. This mean, that a honeypot is expected to get probe, attacked and potentially exploited. Honeypot do not fixed anything. They provide us with additional, valuable information"*

Definisi lain menurut *Retto Baumann* dan *Christian Platner* :

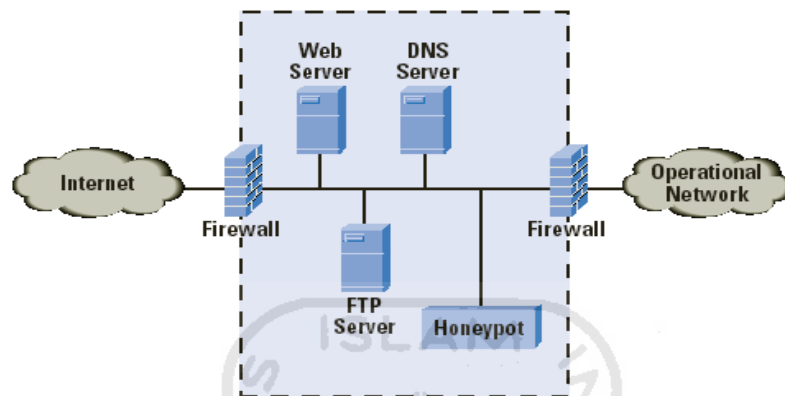
*"A honeypot is resource which pretends to be a real target. A honeypot is a expected to be attacked or compromised. The main goals are the distraction of an attacker and the gain of information about an attack and the attacker"*

*Honeypot* dapat didefinisikan sebagai sebuah *software* dimana nilai guna dari *honeypot* justru ada ketika terdeteksinya suatu tindakan yang tidak terotorisasi pada sebuah jaringan komputer. *Honeypot* memang sengaja dijadikan target serangan agar nilai gunanya itu terlihat. Perlu diketahui bahwa setiap aktifitas yang masuk kedalam *honeypot* tercatat sebagai aktifitas ilegal. Sehingga jika ada *client* yang melakukan interaksi dengan *honeypot* akan tecatat sebagai aktifitas ilegal dan perlu diwaspadai.

Tujuan utama dari *honeypot* ini adalah untuk mengumpulkan informasi sebanyak-banyaknya dari suatu tindakan penyerangan pada sistem jaringan. Dari informasi tersebut administrator dapat menganalisa dan mempelajari aktivitas yang dilakukan *intruder*. Sehingga dapat melakukan pencegahan untuk melindungi *server* yang sebenarnya.

Dalam penerapannya untuk menghasilkan sistem keamanan yang handal, *honeypot* perlu dikonfigurasi dengan firewall atau IDS. Pengkonfigurasi tersebut dapat melengkapi satu sama lain untuk menanggulagi tindakan yang tidak terotorisasi dalam jaringanserver produksi. Sistem kewanaman tersebut dapat

diterapkan pada perusahaan untuk mengamankan data yang terdapat dalam server produksi. Berikut ini adalah salah satu topologi ideal penerapan honeypot pada jaringan komputer.



**Gambar 2. 5** Penempatan *honeypot* dengan *firewall* dalam sistem keamanan.

### 2.3.1 Kategori *Honeypot*

Menurut kategorinya *honeypot* terbagi menjadi dua macam yaitu *production honeypot* dan *research honeypot* [SPI01]. *Production honeypot* digunakan untuk mengurangi resiko penyerangan pada sistem keamanan jaringan dalam sebuah organisasi atau pada *sever* produksi. *Research honeypot* digunakan untuk mendapatkan informasi sebanyak mungkin tentang aktifitas penyerangan dan juga yang sering digunakan sebagai objek penelitian.

Beberapa kalangan memperdebatkan *honeypot* sebagai suatu sistem yang dianggap mampu memberikan nilai pada suatu jaringan. Salah satu cara yang digunakan untuk mengetahui kemampuannya adalah dengan melakukan penyerangan. Dari serangan yang dilakukan seberapa lama *honeypot* mampu bertahan, Waktu bertahan tersebut yang dapat dimanfaatkan administrator jaringan untuk bereaksi atas serangan yang dilakukan. Informasi penyerang yang ada pada *log file* dapat dipelajari dan juga dapat memberikan nilai tambah pada sistem keamanan jaringan jika suatu saat terjadi serangan dan administrator dapat mengantisipasinya dengan baik.

### 2.3.2 Jenis Honeypot

*Honeypot* dapat dibagi berdasarkan tingkat interaksi yang dimilikinya [PRO04]:

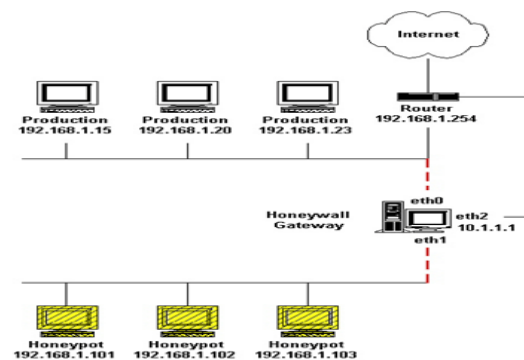
1. *High-interaction honeypot*
2. *Low-interaction honeypot*

Berikut ini adalah penjelasan dari jenis-jenis *honeypot* dapat diuraikan sebagai berikut :

1. *High interaction honeypot*

*High interaction honeypot* mensimulasikan semua aspek dari sebuah sistem yang nyata, dimana *honeypot* level ini mempunyai sistem operasi yang nyata. Sebagai contoh ketika kita akan membangun *honeypot* berbasis *linux* dan menjalankan *FTP server*, maka kita juga harus benar-benar membangun sistem *linux* yang juga benar-benar menjalankan *FTP server*.

Resiko yang dapat timbul pada *honeypot* level ini sangat tinggi karena mempunyai akses *root*. Apabila hal tersebut terjadi maka dapat menjadi ancaman pada jaringan lainnya. Namun, informasi yang didapat mengenai penyerangan akan jauh lebih banyak dikarenakan dapat berinteraksi penuh dengan sistem operasi. Disinilah letak kelebihan dari *high-interaction honeypot*. Hanya saja *high-interaction honeypot* menghabiskan banyak waktu karena harus diawasi secara terus-menerus.

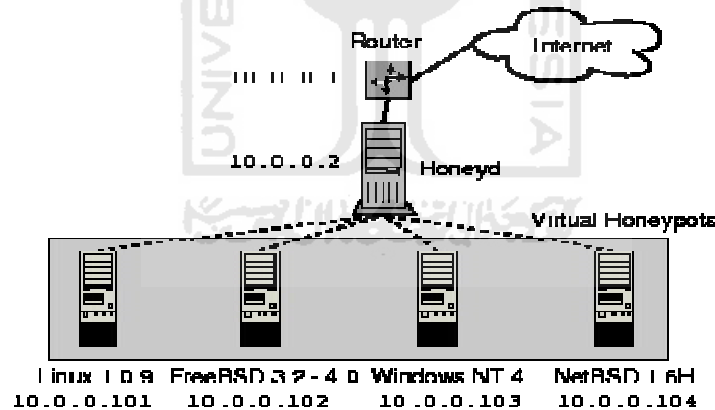


**Gambar 2.6** Topologi *high interaction honeypot*.

## 2. *Low-interaction honeypot*

*Low interaction honeypot* biasanya hanya menyediakan tiruan dari layanan tertentu. Bentuk paling sederhana dari layanan ini dapat diimplementasikan dengan memasang suatu *listener* pada suatu *port*. Pada *low interaction honeypot* tidak ada sistem operasi nyata yang digunakan. Hal ini dapat mengurangi resiko, karena tidak ada kompleksitas dari suatu sistem operasi. Keuntungannya adalah kesederhanaannya, karena dapat dengan mudah dibangun dengan resiko minimal. Kerugiannya adalah sedikitnya informasi yang terekam dari serangan yang terjadi.

*Honeypot* jenis ini bersifat koneksi satu arah karena hanya mendengarkan dan mencatat koneksi yang terjadi tanpa memberikan balasan kepada koneksi tersebut. Jika pada *honeypot* disediakan program yang dapat mengemulasikan suatu layanan, maka *intruder* akan menerima respon seperti halnya respon yang diberikan oleh layanan aslinya.



**Gambar 2.7** Topologi *low interaction honeypot*.

### 2.3.3 Kelebihan dan Kekurangan Honeypot

Sistem yang dibangun pada *honeypot* pastilah memiliki sisi kelebihan dan kelemahan. Sisi kelebihan nantinya akan memberikan manfaat dalam pengimplementasian sedangkan sisi kekurangannya yang nanti akan menjadi sebuah pelengkap untuk membentuk sistem yang sempurna. Berikut ini akan diijelaskan mengenai kelebihan dari *honeypot* [HER11].

1. Bebas dari *problem false-positive* dan *false-negatif*

*Honeypot* relatif tidak akan mengalami *problem false-positive* dan *false negative*. Hal ini dikarenakan *traffic* ke *honeypot* adalah *traffic* aktifitas yang tidak terotorisasi baik oleh pihak eksternal maupun pihak internal.

2. Bebas dari polusi data

*Traffic* ke *Honeypot* adalah *traffic* yang bertujuan untuk melakukan serangan. Hal ini memberikan kemudahan bagi administrator untuk melakukan analisa terhadap mekanisme serangan yang dilakukan *intruder*.

3. Meminimalisir Serangan.

Penggunaan *honeypot* yang umumnya dikonfigurasi dengan pengamanan yang rendah dengan tujuan agar *intruder* mampu menembus dan masuk kedalam sistem. Maka dari keberhasilannya tersebut keinginannya terpuaskan dan menghentikan serangannya tanpa mempengaruhi sistem yang sebenarnya.

4. Analisa lebih terperinci dan aman.

Semua aktifitas yang terekam dalam *honeypot* merupakan aktifitas ilegal. Dengan menggunakan *honeypot* informasi mengenai serangan dapat dipelajari dan dianalisa secara detail tanpa mempengaruhi *production system* karena informasi tersebut terisolasi di dalam *honeypot* saja.

Namun, selain kelebihan dari *honeypot* yang tersebut di atas, *honeypot* juga memiliki kelemahan. Kelemahan *honeypot* adalah bersifat pasif dalam mengamankan infrastruktur jaringan komputer. Aktifitas *hacking* yang tidak mengarah pada *honeypot* tidak akan terkendali dan tidak akan terdeteksi. Untuk itu pengamanan secara nyata pada jaringan produksi tidak bisa semata-mata hanya dilakukan dengan menggunakan *honeypot* saja akan tetapi gabungan antara *honeypot*, *firewall* dan IDS atau IPS.

#### **2.3.4 Nilai guna Honeypot**

Secara umum keamanan jaringan di bagi menjadi tiga area diantaranya *prevention*, *detection* dan *reaction* [SPI01]. Ada kaitan ketiga hal tersebut dengan fungsi *honeypot* didalam area jaringan keamanan.

### 1. Prevention

*Prevention* adalah tindakan pencegahan untuk melindungi jaringan. Salah satu nilai guna *honeypot* adalah pencegahan, karena sebagai pengalih perhatian *intruder* untuk mencegah terjadinya serangan. Hal yang paling utama dari nilai guna ini adalah pencegahan dan pengalihan perhatian. Kedua hal tersebut mampu membuat *intruder* menghabiskan waktu dengan menyerang *honeypot* sehingga mengurangi resiko penyerangan pada *server* produksi.

### 2. Detection

*Honeypot* berguna pada saat pendeteksian karena mampu menyederhanakan proses pendeteksian. Hal tersebut dapat langsung diketahui karena setiap paket data yang masuk ke dalam *honeypot* dapat dikatakan sebagai paket yang ilegal. Jadi, dalam hal ini dapat memudahkan untuk pengumpulan informasi yang diinginkan (paket ilegal).

### 3. Reaction

Nilai guna *reaction* yang dimaksud adalah tidak memberikan layanan tertentu kepada setiap *user*, dikarenakan data yang terkumpul merupakan data yang tidak terotorisasi. Selain itu jika *intruder* mengetahui akan adanya *honeypot* dan berhasil diambil alih maka *honeypot* dapat dilepas dari jaringan tanpa mempengaruhi aktivitas yang ada.

## 2.3.5 Penempatan Honeypot

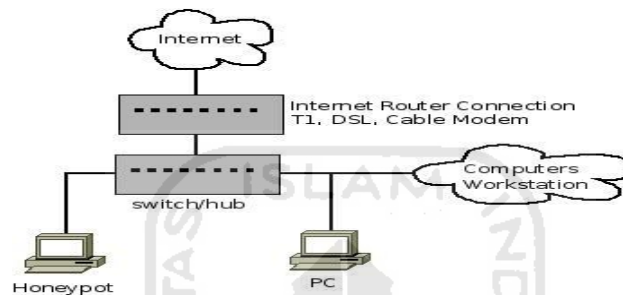
Penempatan *honeypot* dalam jaringan komputer tidaklah membutuhkan tempat khusus. Kebebasan dalam menentukan penempatan *honeypot* dikarenakan *honeypot* tidak memberikan layanan khusus bagi *client*. Namun, ada beberapa tempat yang dapat di bilang strategis dalam penempatan *honeypot*, diantaranya [ELR09]:

#### 1. Terletak dekat dengan Jaringan Publik

Pada penempatan ini *honeypot* terletak dekat dengan *internet*. Penempatan *honeypot* di tempat tersebut tidak membutuhkan adanya kombinasi dengan



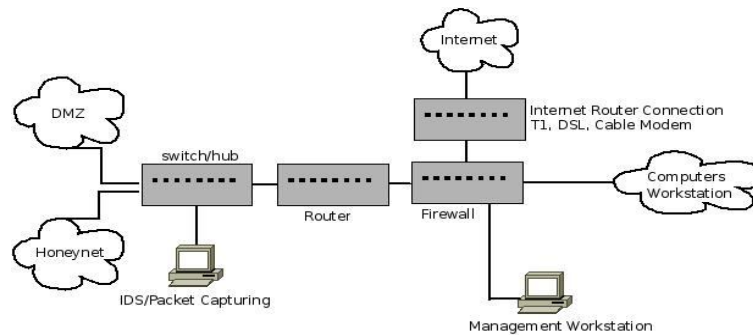
*firewall* dan IDS karena dapat dianggap bagian dari jaringan publik. Keuntungan dari penempatan tersebut ialah mengurangi resiko serangan pada jaringan lokal karena *honeypot* dapat mengalihkan perhatian dari *intruder*. Kekurangannya adalah sistem keamanan yang ada pada jaringan lokal tidak bekerja optimal, karena semua paket yang tidak terotorisasi tercatat semua dalam *log file honeypot*.



**Gambar 2.8** Penempatan *honeypot* dekat dengan jaringan publik.

## 2. Terletak di dalam DMZ

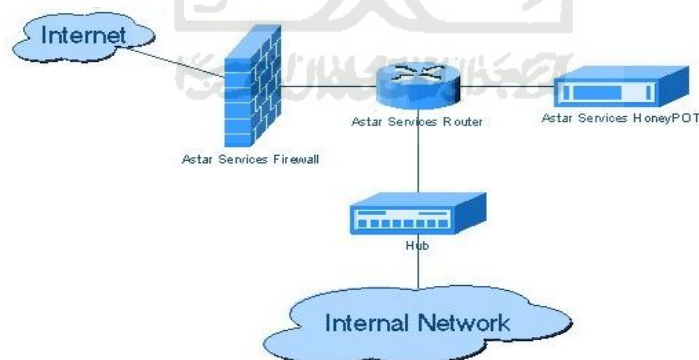
Penempatan *honeypot* didalam DMZ biasa dilakukan pada jaringan yang mempunya *traffic* yang amat besar. Biasanya penempatan tersebut di implementasikan di perusahaan besar dan *honeypot* yang digunakan berjenis *high-interaction honeypot*. Keuntungan yang diperoleh ialah *honeypot* berada dekat dengan *firewall*, sehingga *traffic* yang tertuju pada *honeypot* akan melewati *firewall* tercatat oleh *log firewall*. Kerugiannya pada penempatan ini adalah sistem yang ada dalam DMZ harus diamankan dari *honeypot*, karena jika *honeypot* berhasil di ambil alih maka akan dapat dijadikan senjata untuk menyerang sistem yang ada pada DMZ.



**Gambar 2.9** Penempatan *honeypot* didalam DMZ.

### 3. Terletak di belakang gateway

Penempatan *honeypot* dibelakang *gateway* biasa digunakan untuk mengantisipasi serangan yang berasal dari dalam jaringan lokal. Pada penempatan ini mengandung resiko yang cukup besar. Jika *honeypot* berhasil diambil alih maka *honeypot* akan dijadikan sebagai batu loncatan untuk menyerang sistem yang ada didalamnya. *Firewall* akan tetap mengizinkan *traffic* yang berjalan pada jaringan tersebut karena dianggap *traffic* tersebut tertuju pada *honeypot*.



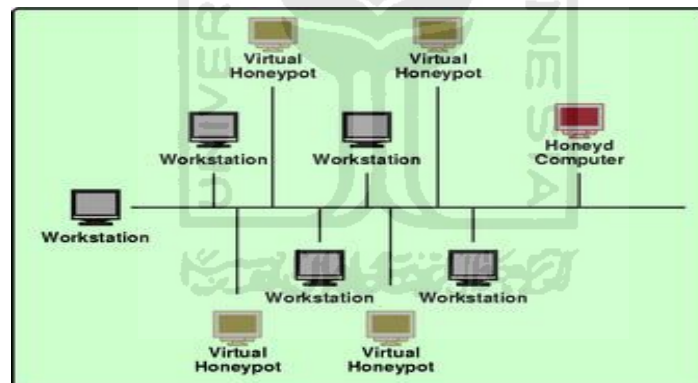
**Gambar 2.10** Penempatan *honeypot* dibelakang *gateway*.

## 2.4 Honeyd

*Honeyd* adalah *honeypot open source* yang dapat membuat *virtual host* tetap pada jaringan. *Honeyd* sendiri tergolong ke dalam *low interaction honeypot*. *Honeyd* mampu membentuk *virtual host* yang nantinya bisa dikonfigurasi

untuk menjalankan banyak *service*. *Honeyd* dapat berjalan sebagai sistem operasi tertentu yang bisa mengelabui *intruder* yang berniat masuk kedalam sistem jaringan.

*Honeyd* dapat menggunakan sejumlah alamat *ip address* yang tidak terpakai pada jaringan untuk disimulasikan menjadi *ip address* sistem operasi tertentu sehingga nampak seperti sebuah *host*. Pengklaiman tersebut dapat diumpamakan, dalam sebuah jaringan LAN dengan *ip address* 192.168.1.1-192.168.1.25 tidak semua alamat *ip address* tersebut dipakai oleh komputer yang aktif. Alamat-alamat *ip address* yang tidak aktif dapat dipakai dan dimonitor oleh *honeyd*. Karena segala macam usaha akses seperti *scanning*, *probing* yang menuju pada *unused ip address* tersebut dapat dikatan ilegal. Gambar 2.8 menjelaskan *unused ip address* pada jaringan yang dapat dipakai oleh *virtual environment* yang terbentuk oleh *honeyd*.



**Gambar 2.11** *Unused ip address*.

Konfigurasi *honeyd* dapat membentuk *virtual host* dimana *virtual host* yang ada bisa di *ping* sehingga menyerupai sistem aslinya. *Honeyd* memiliki *personality* yang berperilaku seolah-olah paket yang dikirimnya berasal dari sistem operasi. *Virtual host* disimulasikan dengan membuat *file* konfigurasi *honeyd*. *Service* yang nantinya akan dijalankan oleh *virtual host* dapat ditentukan dalam *file* konfigurasi tersebut. *Honeyd* memilih berada pada tingkatan *network* yang *stack*. *Honeyd* mampu mengemulasikan service TCP, UDP dan juga dapat merespon paket ICMP [UTD05]. Sejumlah *feature* yang di miliki oleh *honeyd*:

1. Simulasi *virtual host* pada saat bersamaan.
2. Konfigurasi bermacam-macam *service* seperti *telnet*, *apache* dan *ftp*.
3. Simulasi sistem operasi pada tingkat TCP/IP.
4. Simulasi bermacam topologi *routing* : *Latency* dan paket *loss Assymetric routing*, integrasi mesin fisik, *Distributed Honeyd via GRE tunneling*.
5. Subsistem *virtualization* : Menjalankan aplikasi UNIX sesungguhnya di bawah alamat *ip address* virtual *honeyd* (*web*, *server*, *ftp server*), *dynamic port binding* dalam *virtual address space*.

## 2.5 Farpd

*Farpd* adalah *daemon* yang mendengarkan permintaan ARP untuk mengidentifikasi alamat *ip address* tujuan kedalam alamat MAC *address*. *Farpd* dapat mengklaim *unused ip address* pada jaringan dan dapat melepaskannya ketika *ip address* tersebut terpakai oleh komputer sesungguhnya. Hal ini dapat memungkinkan sebuah *host* yang terinstal *farpd* mengklaim semua alamat *ip address* untuk dimonitor.

*Farpd* dapat dihubungkan dengan *honeyd* untuk mengklaim *unused ip address* yang belum dialokasikan dalam jaringan. Alamat *ip DHCP* yang dialokasikan memungkinkan *interfers farpd* untuk mengambil alih *ip address* tersebut. *Virtual server* yang terbentuk oleh *honeyd* masing-masing akan diberi *ip address*, dimana *ip address* tersebut diambil dari *unused ip* pada jaringan. *Ip address* tersebut kemudian dimasukkan ke dalam *interface farpd* yang kemudian digunakan untuk memonitor *traffic* sehingga seolah-olah *ip address* tersebut digunakan oleh sebuah *host*. Jadi, jika ada yang mengakses *ip address* tersebut maka akan tercatat pada *log file honeyd*.

## 2.6 Gammu

*Gammu* adalah aplikasi yang digunakan untuk *sms gateway* yang menghubungkan operator selular dengan *internet*. Aplikasi ini bersifat *open source* yang ada dibawah *license GPL*. *Gammu* dapat berjalan pada sistem operasi berbasis *linux* dan *windows*. Media yang dapat digunakan oleh *gammu* adalah *modem* dan *handphone* dengan operator GSM.

Dalam penyelesaian tugas akhir ini, *gammu* dikonfigurasi dengan *honeyd*. Konsepnya adalah *log file* yang tercatat pada *honeyd* akan dikirimkan melalui *sms* ke *handphone* Administrator. Pengkonfigurasi tersebut dapat menambah nilai guna *reaction* dari *honeypot*. Keuntungannya adalah administrator dapat dengan mudah mengetahui adanya paket yang tidak terotorisasi serta memberikan waktu yang lebih lama pada administrator untuk menghindarkan *server* produksi dari serangan yang dilakukan *intruder*.

## 2.7 Regular Expression

*Regular Expression* dalam situs resminya menjelaskan *regular expression* (*regex or regexp for short*) is a special text string for describing a search pattern [BAR01]. Dari pengertian tersebut dapat langsung dipahami bahwa fungsi utama dari *regex* adalah untuk menggambarkan sebuah pola pencarian berdasarkan *string*. *Regular expression* atau yang sering disebut *regex* sering digunakan untuk menentukan kata pada baris yang mengandung pola tertentu. Hal ini akan lebih mempermudah dalam pencarian kata menggunakan *regex* karena akan lebih cepat tanpa harus mencari *string* satu-persatu.

Pada sistem operasi *linux* banyak perintah-perintah yang menyisipkan fungsi dari *regex*. Beberapa perintah *regex* yang digunakan seperti *grep*, *sed*, *awk*. Selain memiliki perintah-perintah tersebut, *regex* mempunyai fitur *case sensitive* yang digunakan untuk menentukan pola. Didalam pencocokan suatu *string regex* menggunakan pola yang terbentuk dari karakter-karakter yang merupakan bagian dari *regex*. Karakter tersebut antara lain.

### 1. Literal Character

*Literal character* adalah karakter yang digunakan untuk melakukan pencarian sebuah *string* [AND09]. Contohnya seperti *string* “*honey*” dapat ditemukan pada kata *honeypot* dan *honeyd*.

### 2. Metacharacter

*Metacharacter* adalah sebuah karakter yang mempunyai fungsi special [AND09]. *Metacharacter* yang sering digunakan untuk membuat *regex* antara lain  $\wedge$   $\$$   $\{$   $\}$   $[$   $]$   $\cdot$   $*$   $+$   $?$   $/$   $|$ .

### 3. *Escape Sequence*

*Escape character* adalah karakter yang mempunyai fungsi untuk mengubah *metacharacter* menjadi *literal character* [AND09]. Penggunaannya dapat digabungkan dengan metacharacter contohnya seperti `^honey`.

Selain dari karakter-karakter diatas *regex* juga memiliki standar dalam penyusunannya. Standar tersebut dinamakan POSIX yang nantinya akan digunakan dalam pola pecarian. Berikut ini adalah tabel POSIX yang digunakan *regex* [AND09].

POSIX Standart		
<code>[:alnum:]</code>	<code>[A-Za-z0-9]</code>	alfanumerik
<code>[:word]</code>	<code>[A-Za-z0-9_]</code>	Alfanumerik dengan ‘_’
<code>[:alpha]</code>	<code>[A-Za-z]</code>	alfabet
<code>[:blank:]</code>	<code>[\t]</code> (ada spasi di antara [ dan \)	Spasi dan tab
<code>[:cntrl:]</code>	<code>[\x00-\x1F\x7F]</code>	Karakter kontrol
<code>[:digit:]</code>	<code>[0-9]</code>	Angka/digit
<code>[:graph:]</code>	<code>[\x21-\x7E]</code>	Karakter visibel
<code>[:punct:]</code>	<code>[-!\"#\$%&amp;'()^+.,/:;[\_{}~]</code>	Tanda baca

**Table 2.1** Tabel POSIX *regular expression*.

## BAB III METODOLOGI

### 3.1 Analisis Masalah

Sistem keamanan merupakan syarat mutlak yang harus ada dalam jaringan komputer. Entah seberapa besar dan kecil jaringan tersebut pasti membutuhkannya. Apalagi yang berkaitan dengan produktifitas, untuk dapat mengamankan semua data-datanya perlu dibangun sebuah sistem keamanan.

Infomasi *real* mengenai serangan jaringan komputer pasti sangat berguna dalam mengambil langkah-langkah prefentif untuk melakukan pengamanan sistem. *Honeypot* merupakan *software* yang mampu mendapatkan informasi *real* mengenai serangan. *Honeypot* yang digunakan dalam penelitian ini adalah *honeyd* yang termasuk jenis *low interaction honeypot*. *Honeyd* mampu membentuk *virtual environtment* yang membentuk *server* palsu yang digunakan mengelabui dan mengalihkan perhatian *intruder*. Semua informasi yang tersimpan dalam *log file honeyd* merupakan informasi ilegal. Apalagi jika informasi serangan tersebut dapat diketahui dengan cepat maka akan dapat mempermudah kinerja administrator jaringan untuk mengetahui aktivitas yang dilakukan *intruder*. Administrator hanya tinggal mengamati dan menganalisa *log file honeyd*.

Dalam penerapan yang dilakukan pada perusahaan-perusahaan untuk mengamankan *server* produksi, *honeypot* dikonfigurasi dengan *firewall* dan IDS. Pengkombinasian tersebut akan menciptakan sistem keamanan yang handal dan saling melengkapi satu sama lain. Namun, pada penelitian ini akan lebih difokuskan pada kinerja dari pada *honeypot* yang berbasis *honeyd* dalam merekam paket yang tidak terotorisasi yang masuk kedalamnya dan mengolahnya untuk dikirimkan melalui *sms*.

Inti dan tujuan utama dari pengimplementasian *honeypot* adalah untuk mengetahui paket yang tidak terotorisasi pada jaringan yang masuk kedalam *honeypot* dan mengirimkan paket tersebut melalui *sms*. Dari inti dan tujuan utama yang dijelaskan pada kalimat diatas, penulis mengumpulkan variabel-variabel

yang digunakan agar hasil yang diharapkan terarah, tepat pada sasaran dan akurat.

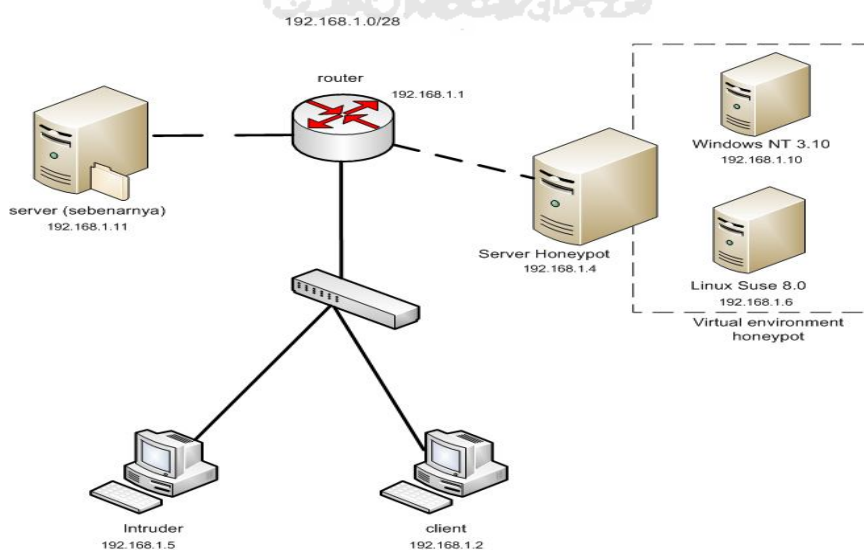
Variabel-variabel tersebut antara lain:

- Konfigurasi *honeyd*, *farpd* dan *gammu*
- Membentuk *virtual environment* yang mampu menjalankan sejumlah layanan.
- Menganalisa paket yang tidak terotorisasi yang terekam pada *log file honeypot*.
- Mengolah *log file honeypot* untuk dikirimkan melalui *sms*.

### 3.2 Topologi Jaringan

Pada analisis masalah telah dijelaskan bagaimana penggunaan *honeypot* dalam jaringan komputer untuk menciptakan sistem keamanan yang handal. Selain hal tersebut penerapan *honeypot* secara nyata yang dijelaskan pada BAB II *point 2.3 Honeypot*, juga perlu diketahui agar dapat dijadikan acuan dalam perancangan topologi yang dilakukan pada penelitian ini.

Perancangan topologi digunakan untuk membuat jaringan sebagai tempat pengimplementasian *honeypot* agar dapat berjalan optimal. Jaringan yang digunakan adalah LAN berbasis Ipv4. Pada perancangan ini tidak ada konfigurasi antara *honeypot* dengan *firewall* atau IDS. Perancangan tersebut digunakan untuk mengetahui bagaimana *honeypot* merekam aktifitas yang tidak terotorisasi.



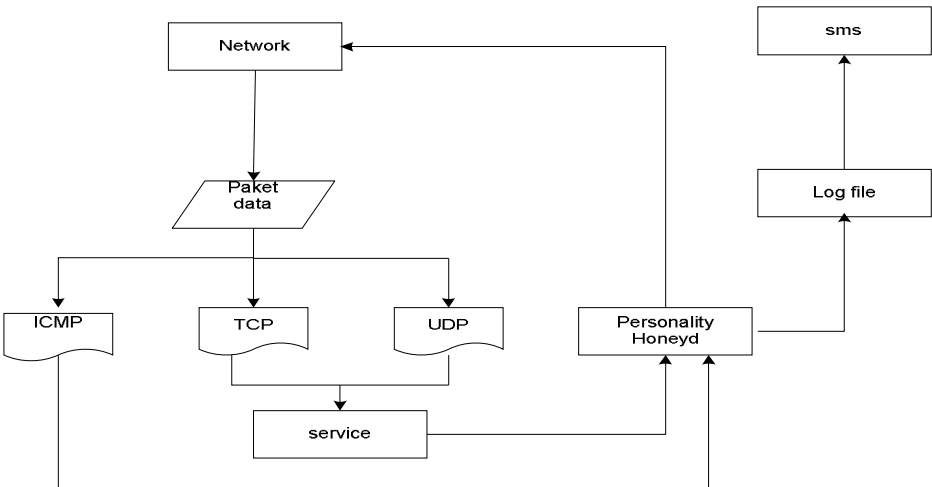
Gambar 3.1 Rancangan topologi *honeyd*.



Rancangan topologi yang digunakan dapat dilihat pada gambar 3.1 terdapat dua *client* yang masing berperan menjadi *intruder* dengan menggunakan sistem operasi *backtrack* dan satu komputer lagi *client* dengan sistem operasi *windows*. Dua *client* tersebut dihubungkan oleh *switch* yang mengarah pada *router*. Dapat dilihat di bagian kanan dan kiri terdapat dua buah *server* yaitu *server honeypot* dan *web server* (*server* sebenarnya). Bila dilihat pada *server honeypot* terdapat *virtual environment* dimana *virtual environment* tersebut berisi dua *server* palsu yaitu *Windows NT 3.10* dan *Linux Suse 8.0* yang nantinya akan menjalankan sejumlah layanan tertentu.

**3.3 Desain Alur Data**

Desain alur data ini menggambarkan proses yang berjalan pada *honeyd*. *Honeyd* dapat mengemulasikan layanan TCP, UDP dan ICMP. Ketika pada *Network* terdapat paket TCP yang menuju padanya, paket tersebut pasti akan meminta sebuah *service* pada *honeyd* yang kemudian menuju pada *personality honeyd* yang merupakan *file* konfigurasi *server* palsu. Kemudian *honeyd* merespon dan mengirimkan kembali *service* yang diminta. Untuk paket ICMP, paket tersebut langsung diteruskan karena tidak memuat layanan seperti pada TCP dan UDP. Paket yang menuju *server* palsu tersebut kemudian tercatat kedalam *log file* dan kemudian *log file* tersebut dikirimkan melalui *sms*.



**Gambar 3.2** Alur proses yang berjalan pada *honeyd*.

### 3.4 Analisi dan Persiapan Kebutuhan Sistem

#### 3.4.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras merupakan salah satu sarana pendukung yang digunakan dalam penelitian ini. Perangkat keras yang digunakan antara lain.

1. Komputer

Komputer ini berfungsi sebagai *server* yang nantinya akan membentuk *virtual environment honeypot* dan juga digunakan untuk *server* sebenarnya. Namun, karena keterbatasan penulis yang tidak memiliki komputer, maka komputer tersebut diganti dengan menggunakan laptop. Berikut ini adalah spesifikasi minimal yang dibutuhkan :

- a. Processore 1.6 GHz
- b. Memori 256
- c. Hardisk dengan kapasitas 20GB
- d. 1 buah *Network Interface Card*

2. Laptop sebagai *client*

Laptop yang digunakan ada dua, salah satu akan dijadikan sebagai *client* biasa dan yang satunya lagi akan dijadikan sebagai *intruder*. Spesifikasi laptop:

- a. Processor core i3 2.1 Ghz
- b. Memori 2GB
- c. Hardisk 500GB
- d. 1 buah *Network Interface Card*

3. *Router*

*Router* digunakan untuk membagi *ip address* pada topologi jaringan pada gambar 3.1. Adapun *router* yang digunakan adalah TP-LINK TP-MR3220

4. *Switch*

*Switch* digunakan untuk menghubungkan *client* dengan *router* sesuai dengan gambar 3.1. Adapun *switch* yang digunakan adalah TP-LINK TP-R460.

#### 5. *Modem GSM*

Modem GSM digunakan sebagai perangkat *sms gateway*. Didalam modem tersebut berisi *SIM card* dengan operator *seluler GSM 3*, yang nantinya digunakan untuk mengirim *sms*. Adapun seri *modem* yang digunakan adalah Huawei E153.

### 3.4.2 **Kebutuhan Perangkat Lunak**

Kebutuhan perangkat merupakan salah satu media yang mendukung jalannya aplikasi pada tahap implementasi ini. Adapun perangkat lunak yang dibutuhkan sebagai berikut.

#### 1. *Linux Ubuntu*

*Linux ubuntu* merupakan sistem operasi *open source*. *Linux ubuntu* yang digunakan penulis dalam melakukan penelitian ini adalah *linux ubuntu 11.04*. Dalam pengimplementasiannya *linux ubuntu 11.04* digunakan sebagai sistem operasi untuk menjalankan aplikasi *honeyd* dan *gammu* serta aplikasi pendukung lainnya. Selain itu juga sistem operasi tersebut digunakan sebagai *server* sebenarnya.

#### 2. *Linux Backtrack*

Sistem operasi *linux backtrack* merupakan salah satu *distro linux* yang menjadi favorit para *intruder* untuk melakukan serangan. Pada sistem ini telah memuat banyak *tools exploit* yang digunakan untuk melakukan serangan. Sistem operasi ini digunakan penulis untuk melakukan pengujian terhadap *honeyd*, apakah *honeyd* sudah bekerja pada jaringan dan merekam semua aktifitas serangan yang ada pada *log file* nya. Adapun versi sistem operasi *linux backtrack* yang digunakan penulis adalah *backtrack 5 revolution*.

#### 3. *Honeyd*

*Honeyd* merupakan jenis *low interaction honeypot*, dimana pada jenis ini memang sengaja digunakan untuk penelitian. *Honeyd* dapat berjalan pada sistem operasi berbasis *linux*. Pengkonfigurasiannya *honeyd* dapat membentuk

*virtual environment* sebagai *server* palsu yang digunakan untuk mengalihkan perhatian dan menjebak *intruder*.

#### 4. *Farpd*

*Farpd* adalah aplikasi yang mendengarkan permintaan ARP yang cocok dengan tujuan yang ditentukan dengan alamat *MAC address*. *Farpd* mampu memonitor semua *traffic* ARP yang ada pada jaringan. Dalam pengimplementasian ini *farpd* akan dipasang dimana sistem *honeyd* berada.

#### 5. *Gammu*

*Gammu* merupakan aplikasi *sms gateway* yang dapat menghubungkan operator *seluler* dengan *internet*. Sekarang banyak sekali pemanfaatan *sms gateway* untuk mempermudah menjalankan suatu layanan dari berbagai macam *service* yang ada. Salah satu pemanfaatannya adalah dengan menghubungkan *gammu* dengan *honeyd*. Pengkonfigurasiannya tersebut diharapkan mampu memberikan nilai guna *reaction* untuk melakukan pengamanan dengan cepat.

### 3.5 Instalasi dan Konfigurasi

Instalasi perangkat lunak yang dibutuhkan pada perangkat keras dengan cara yang berbeda-beda. Setelah proses instalasi akan dilanjutkan pada tahap konfigurasi untuk mengoptimalkan jalannya sistem.

#### 3.5.1 Sistem Operasi

Sistem operasi yang digunakan dalam untuk menyelesaikan penelitian ini menggunakan *linux ubuntu* 11.04 yang digunakan sebagai media berjalannya *honeyd* dan *gammu*. Selain itu juga sistem operasi tersebut digunakan sebagai *server* sebenarnya. Pada tahap ini juga dilakukan penginstalan sistem operasi *backtrack* yang digunakan untuk melakukan pengujian pada *honeyd*.

#### 3.5.2 Honeyd

Aplikasi *honeyd* yang digunakan pada penyelesaian tugas akhir ini merupakan paket yang sebenarnya telah ada pada *repository ubuntu*. Jadi, paket tersebut dapat diinstal dengan mudah. Setelah proses penginstalan selesai, maka

*honeyd* akan membentuk direktori *honeyd* pada sistem operasi *linux ubuntu*.

Berikut adalah beberapa direktori yang dibentuk :

- a. Direktori */etc/honeypot/* merupakan direktori yang menyimpan *file* konfigurasi *honeyd* yang digunakan untuk menjalankan *service honeyd*. Isi *file* penting yang ada di dalam direktori tersebut antara lain *honeyd.conf*, *nmap.prints* dan *xprobe.conf*.
- b. Direktori */var/log/honeypot/* merupakan direktori yang menyimpan *log file honeyd*.
- c. Direktori */etc/honeypot/scripts* merupakan direktori yang didalamnya terdapat subdirektori yang menyimpan *script* pembentuk *server-server* palsu. Subdirektori tersebut antara lain */misc*, */snmp*, */telnet*, */unix* dan */win32*. Pada */misc* terdapat *script* *base.sh* digunakan untuk *logging* tambahan, */snmp* berisi *script* yang dapat mengemulasikan *service snmp*, */telnet* berisi *script* yang mengemulasikan *service telnet*, */unix* berisi *script* yang mengemulasikan *service* sistem operasi *linux* dan */win32* berisi *scripts* yang mengemulasikan *service* sistem operasi *windows*.

Untuk membuat *virtual environment* diperlukan *template* yang didukung oleh *file -file* yang ada pada subdirektori *scripts*. Berikut ini adalah contoh *template honeyd*.

```
create suse80
set suse80 personality "Linux 2.4.7 (X86)"
set suse80 default tcp action reset
set suse80 default udp action reset
set suse80 default icmp action open
add suse80 tcp port 80 "sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/
apache.sh"
bind 192.168.1.10 suse80
```

Dapat dilihat dari contoh diatas, dalam pembentukan *template* terdapat *format* yang digunakan. Berikut ini akan dijelaskan mengenai penggunaan *format-format* tersebut.

a. *Create*

*Create* ini berfungsi untuk membantu nama *server* palsu. *Format* dari perintah *create*, ***create "template name"***. Dapat dilihat pada contoh diatas, pada baris pertama terdapat perintah ***create suse80*** yang berarti perintah untuk membuat *server* palsu dengan nama ***suse80***.

b. *Set*

*Set* ini berfungsi untuk mengambil *personality* yang berada pada *file nmap.prints* dan juga untuk mengaktifkan *service*. Didalam *file nmap.prints* terdapat nama-nama *OS fingerprint* yang dapat digunakan untuk membentuk *server* palsu tersebut. *Format* penggunaan perintah *set*.

***set template name personality "personality-name"***

***set template name default protocol action port***

Dapat dilihat pada contoh diatas dari baris kedua penggunaan *set* untuk mengambil *personality*. *Personality-name* berasal dari *file nmap.prints*. *Personality-name* tersebut yang nantinya akan menjadi nama sistem operasi pada *server* palsu. Selanjutnya penggunaan *set* juga digunakan untuk mengaktifkan *service*. Dapat dilihat pada contoh berikut ini.

***set suse80 default tcp action reset***

***set suse80 default udp action reset***

***set suse80 default icmp action open***

Perintah ***set suse80 default tcp action reset*** menjelaskan bahwa semua *port* TCP akan ditutup. Selain *reset* terdapat *action* lain untuk *port* TCP yaitu *block*, *open* dan *tarpit*. *Block* berarti semua paket akan didrop dan tidak melakukan *reply*. *Open* yang berarti *port* dibuka sedangkan *tarpit* berarti *sticky connection* (sibuk). Pada *port* TCP jika tidak dinyatakan dengan *action* apapun maka *defaultnya* adalah *open*, sedangkan pada *port* UDP *defaultnya* adalah *close*. Pada *protocol* ICMP *defaultnya* adalah

*open*. Terdapat *action* lain pada *protocol* ICMP yaitu *open* dan *block*. *Open* yang berarti *reply* sedangkan *Block* yang berarti *drop* tanpa *reply*.

c. *Add*

*Add* digunakan untuk menambahkan *file -file* yang ada pada subdirektori **scripts** dengan *service* yang akan dijalankan. Dapat dilihat pada contoh berikut ini.

```
add suse80 tcp port 80 "sh
/etc/honeypot/scripts/scripts/unix/linux/suse8.0/apache.sh"
```

Perintah tersebut digunakan untuk menjalankan *script apache.sh* pada */etc/honeypot/scripts/scripts/unix/linux/suse8.0* dengan *port* 80 yang merupakan layanan *http*.

d. *Bind*

*Bind* digunakan untuk memberikan alamat *ip address* pada *template* yang telah dibuat. Alamat *ip address* tersebut berasal dari *unused ip address* pada jaringan. Dapat dilihat pada contoh diatas, perintah ***bind 192.168.1.10 suse80*** menjelaskan *template* dengan nama *suse80* dengan *ip address* 192.168.1.10. *Ip address* tersebut yang nantinya dijadikan sebagai *ip address server* palsu.

### 3.5.3 Farpd

Pada tahap ini akan dilakukan instalasi *farpd* yang dapat langsung langsung dilakukan pada terminal *linux ubuntu*. Pengkonfigurasi dengan *honeypot* digunakan untuk mengalim *unused ip address* pada jaringan untuk dapat digunakan oleh *server* palsu dan juga untuk memonitor aktifitas yang mengarah pada *unused ip address* terutama *ip address* yang digunakan oleh *server palsu*.

### 3.5.4 Gammu

Pada tahap ini akan dilakukan instalasi *gammu* pada sistem operasi *linux ubuntu*. Untuk dapat melakukan pengiriman perlu dilakukan konfigurasi aplikasi *gammu* dengan modem sebagai media yang digunakan mengirimkan sms ke nomer tujuan.

### 3.5.5 Konfigurasi *Honeyd* dengan *Gammu*

Pada tahap ini akan dilakukan konfigurasi *honeyd* dengan *gammu*. Pengkonfigurasiannya tersebut digunakan untuk mengirimkan paket yang tidak terotorisasi berupa *sms* kepada administrator secara otomatis jika terdapat aktifitas yang tidak terotorisasi dalam jaringan komputer. Program yang dibuat menggunakan *shell programming*, dimana *script* program yang dibuat dapat berjalan pada sistem operasi *linux*.

### 3.6 Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui apakah *honeyd* dapat bekerja untuk merekam paket yang masuk di dalamnya. Serangan akan dilakukan pada jaringan tersebut untuk mengetahui kinerjanya. Pengujian sistem dalam penelitian ini menggunakan tools yang terdapat pada sistem operasi *backtrack*. *Tools exploit* yang digunakan dengan metode *scanning*, *brute force* dan *os fingerprinting*. Berikut ini adalah penjelasan dari masing-masing teknik pengujian.

1. Teknik Scanning

Teknik *scanning* digunakan untuk mengidentifikasi sistem yang akan dijadikan target serangan. *Scanning* yang dilakukan juga bertujuan untuk mengetahui apakah *ip address* yang digunakan oleh *server* palsu terdeteksi sebagai *ip address* yang sedang digunakan oleh sebuah *host*. *Tools exploit* yang digunakan pada teknik *scanning* ini adalah *nmap*, dimana tools tersebut telah tersedia didalam sistem operasi *backtrack*.

2. Teknik Brute Force

Teknik *brute force* digunakan untuk melakukan serangan kepada sistem dengan menggunakan semua kunci yang mungkin. Pada pengujian ini menggunakan sebuah *tools exploit* yaitu *brutus alert* dimana didalamnya telah terdapat daftar *username* yang terdapat pada *file user.txt* dan *password* yang terdapat pada *file words.txt*.

3. Teknik OS Fingerprinting

*Os fingerprinting* merupakan teknik yang digunakan untuk mendeteksi sistem operasi yang digunakan oleh target serangan. Sistem operasi yang



digunakan oleh target dapat membantu dalam menentukan kelemahan dan celah-celah yang digunakan untuk masuk ke dalam sistem. *Tools* yang digunakan dalam pengujian *os fingerprinting* ini adalah *xprobe2* dimana *tools* tersebut sudah tersedia di dalam sistem operasi *backtrack*.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi Sistem

Implementasi sistem merupakan tahap dimana sistem mampu dijalankan pada topologi yang dibuat. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak. Serta apakah sistem menghasilkan *output* yang sesuai dengan perancangan yang telah dibuat.

##### 4.1.1 Honeyd

Pada tahap ini akan dilakukan instalasi honeyd pada *server honeypot* yang berisi sistem operasi linux ubuntu. Berikut ini adalah perintah yang digunakan menginstal *honeyd*.

```
root@honey-Aspire:~# apt-get install honeyd
```

Seperti yang dijelaskan pada BAB III *point* instalasi dan konfigurasi *honeyd*, setelah proses instalasi selesai maka akan terbentuk direktori */etc/honeypot/*, direktori */var/log/honeypot/* dan direktori */etc/honeypot/scripts*.

Pada penelitian ini akan dibangun dua buah *server* palsu berbasis windows dan linux yang mampu menjalankan beberapa *service* layaknya sistem sebenarnya. *Server* palsu tersebut diantaranya :

a. *Windows* NT 3.10

Pada *server* ini telah dikonfigurasi beberapa *service* seperti *msftp*, *exchange-smtp* dan *iis*. *Ip address* yang diberikan pada *server* ini 192.168.1.10.

b. *Linux* Suse 8.0

Pada *server* ini telah dikonfigurasi beberapa *service* seperti *proftpd*, *ssh*, *telnetd* dan *apache*. *Ip address* yang diberikan pada *server* ini 192.168.1.6.

Konfigurasi server palsu *Windows* NT 3.10 dan *Linux* Suse 8.0 terletak didalam file *honeyd.conf* pada direktori */etc/honeypot/*. Berikut ini adalah konfigurasi server palsu yang terdapat dalam file *honeyd.conf*.

```

###Server Windows
create win2k
set win2k personality "Windows NT 3.10 (Build 528)"
set win2k default tcp action open
set win2k default udp action open
set win2k default icmp action open
set win2k uptime 3567
add win2k tcp port 21
"sh /etc/honeypot/scripts/scripts/win32/win2k/msftp.sh"
add win2k tcp port 25
"sh /etc/honeypot/scripts/scripts/win32/win2k/exchange-smtp.sh"
add win2k tcp port 80
"sh /etc/honeypot/scripts/scripts/win32/win2k/iis.sh"
bind 192.168.1.10 win2k

###Server Linux Suse 8.0
create suse80
set suse80 personality "Linux 2.4.7 (X86)"
set suse80 default tcp action reset
set suse80 default udp action block
set suse80 default icmp action open
set suse80 uptime 79239
add suse80 tcp port 21
"sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/proftpd.sh"
add suse80 tcp port 22
"sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/ssh.sh"
add suse80 tcp port 23
"sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/telnetd.sh"
add suse80 tcp port 80
"sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/apache.sh"

```

Pada konfigurasi di atas terdapat dua buah *server* palsu dimana masing-masing dari *server* tersebut dikonfigurasi untuk menjalankan beberapa *service*. Berikut ini adalah penjelasan *script* dari konfigurasi *server* palsu tersebut.

1. *Server* palsu *windows*

Pada *server* palsu *windows* ini akan menjalankan *service smtp*, *exchange-smtp* dan *iis*. *Protocol* pada *server* ini akan diset dalam keadaan terbuka (*open*). Berikut ini adalah penjelasan dari konfigurasi *server* palsu *windows*.

- a. **Create *win2k***, *server* palsu *windows* tersebut dibuat dengan nama template *win2k*.

- b. *set win2k personality "Windows NT 3.10 (Build 528)"*, server windows mengemulasikan *personality Windows NT 3.10 (Build 528)* yang nantinya akan menjadi nama dari server palsu tersebut.
  - c. *set win2k default tcp action open*, semua port TCP pada server windows dibuat terbuka untuk menerima koneksi pada port tersebut.
  - d. *set win2k default udp action open*, port udp diset terbuka untuk menerima koneksi pada port tersebut.
  - e. *set win2k default icmp action open*, ICMP juga diset terbuka agar dapat melakukan koneksi ketika dilakukan perintah *ping*.
  - f. *set win2k uptime 3567*, up time server palsu diset selama 3567s.
  - g. *add win2k tcp port 21 "sh /etc/honeypot/scripts/scripts/win32/win2k/msftp.sh"*, untuk menjalankan service *msftp.sh* dengan membuka port 21.
  - h. *add win2k tcp port 25 "sh /etc/honeypot/scripts/scripts/win32/win2k/exchange-smtp.sh"*, untuk menjalankan service *exchange-smtp.sh* dengan membuka port 25.
  - i. *add win2k tcp port 80 "sh /etc/honeypot/scripts/scripts/win32/win2k/iis.sh"*, untuk menjalankan service *iis.sh* dengan membuka port 80.
  - j. *bind 192.168.1.10 win2k*, untuk memberikan ip address pada server palsu.
2. *Server palsu linux*
- Pada server palsu linux ini akan menjalankan service *proftpd*, *ssh*, *telnet*, dan *apache*. Protocol pada TCP pada server ini akan diset *reset*, begitu juga dengan protocol udp diset *blok*, sedangkan ICMP dibiarkan dalam keadaan *open*. Berikut ini adalah penjelasan dari konfigurasi server palsu linux.
- a. *Create suse80*, server palsu linux tersebut dibuat dengan nama *template suse80*.
  - b. *set suse80 personality "Linux 2.4.7 (X86)"*, server palsu linux mengemulasikan *personality Linux 2.4.7 (X86)* yang nantinya akan menjadi nama dari server palsu tersebut.

- c. *set suse80 default tcp action reset*, semua *port* TCP pada *server* palsu *linux* dibuat tertutup.
- d. *set suse80 default udp action block*, *port* UDP diset *blok* untuk tidak dapat menerima koneksi pada *port* tersebut.
- e. *set suse80 default icmp action open*, ICMP juga diset terbuka agar dapat melakukan koneksi ketika di lakukan perintah *ping*.
- f. *set suse80 uptime 79239*, *up time server* palsu diset selama 79239s.
- g. *add suse80 tcp port 21 "sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/proftpd.sh"*, untuk menjalankan *service proftpd.sh* dengan membuka *port* 21.
- h. *add suse80 tcp port 22 "sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/ssh.sh"*, untuk menjalankan *ssh.sh* dengan membuka *port* 22.
- i. *add suse80 tcp port 23 "sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/telnetd.sh"*, untuk menjalankan *service telnet.sh* dengan membuka *port* 23.
- j. *add suse80 tcp port 80 "sh /etc/honeypot/scripts/scripts/unix/linux/suse8.0/apache.sh"*, untuk menjalankan *service apache.sh* dengan membuka *port* 80.

Setelah melakukan konfigurasi pada *file honeyd.conf*, maka *honeyd* dapat dijalankan dengan menggunakan perintah sebagai berikut.

```
root@honey-Aspire:~# honeyd -p /etc/honeypot/nmap.prints
-x /etc/honeypot/xprobe2.conf -l /var/log/honeypot/honeyd.log
-f /etc/honeypot/honeyd.conf -i eth0 192.168.1.6 192.168.1.10
```

Berikut ini adalah penjelasan dari perintah diatas:

- a. *Option -p (fingerprint)*, digunakan untuk membaca data *OS fingerprint* yang telah disimpan pada direktori *etc/honeypot/nmap.prints*.
- b. *Option -x (xprobe)*, digunakan untuk menentukan bagaimana *honeyd* bereaksi pada ICMP *fingerprinting tools*.
- c. *Option -l (logfile)*, digunakan untuk melakukan penyimpanan *log file* pada direktori */var/log/honeypot/honeyd.log*.

- d. *Option -f (file)*, digunakan untuk membaca konfigurasi yang ada pada *file honeyd.conf* dan menjalankan *file* konfigurasi tersebut.

#### 4.1.2 Farpd

*Farpd* merupakan aplikasi yang mendukung kinerja *honeyd*. *Farpd* merupakan aplikasi yang terdapat pada *repository ubuntu 11.04* jadi dapat diinstal dengan mudah. Berikut ini adalah perintah yang digunakan.

```
root@honey-Aspire:~# apt-get install farpd
```

Untuk menjalankan *farpd* dapat menggunakan perintah berikut ini.

```
root@honey-Aspire:~# farpd -d -i eth0 192.168.1.0/28
```

Pada perintah diatas *farpd* bekerja untuk memonitor *ip address* pada *network 192.168.1.0/24* yang memonitornya lewat *interface eth0*. Sehingga semua *ip address* pada jaringan tersebut akan dimonitor oleh *farpd*.

#### 4.1.3 Gammu

*Gammu* aplikasi yang digunakan untuk mengirim *sms* yang berisi paket yang tidak terotorisasi ke *handphone* administrator. Agar pengiriman dapat dilakukan, peneliti menggunakan *modem GSM Huawei*. Berikut ini akan dijelaskan langkah instalasi.

1. Menambahkan dalam *repository* dengan mengetik perintah berikut.

```
honey@honey-Aspire:~$ sudo vim /etc/apt/sources.list.d/gammu.list
```

2. Memasukan alamat *repo* pada halaman *source list* dan kemudian simpan.

```
"deb http://repo.ugm.ac.id/ppa.launchpad.net/gammu lucid main"
```

3. *Download key gammu* dengan menggunakan perintah berikut.

```
honey@honey-Aspire:~$
```

```
wget ftp://repo.ugm.ac.id/ekstra/ugos_tools/key_gammu
```

4. Menambahkan *key gammu* ke dalam sistem dengan menggunakan perintah berikut.

```
honey@honey-Aspire:~$ apt-key add gammu_key
```

5. *Update dan install* aplikasi *gammu* dengan menggunakan perintah berikut.

```
honey@honey-Aspire:~$ sudo apt-get update
```

```
honey@honey-Aspire:~$ sudo apt-get install gammu gammu-smsd
```

6. Memasukkan modem kedalam *port* USB lalu cek apakah *modem* tersebut terdeteksi atau tidak. Untuk mengeceknya dapat menggunakan perintah berikut.

```
honey@honey-Aspire:~$ dmesg | grep tty
```

```
honey@honey-Aspire:~$ dmesg | grep tty
[  0.000000] console [tty0] enabled
[ 19.516644] usb 2-6: GSM modem (1-port) converter now attached to ttyUSB0
[ 19.516857] usb 2-6: GSM modem (1-port) converter now attached to ttyUSB1
[ 19.517048] usb 2-6: GSM modem (1-port) converter now attached to ttyUSB2
```

**Gambar 4.1** Modem telah terdeteksi.

7. Melakukan konfigurasi *gammu*, dengan mengetikan perintah yang ada dibawah ini dan kemudian akan masuk kedalam konfigurasi *gammu*.

```
honey@honey-Aspire:~$ sudo gammu-config
```

```
Current Gammu configuration
P Port (/dev/ttyUSB0)
C Connection (at115200)
M Model ()
D Synchronize time (yes)
F Log file (/etc/gammulog)
O Log format (textall)
L Use locking ()
G Gammu localisation ()
H Help
S Save
<Ok> <Cancel>
```

**Gambar 4.2** Form konfigurasi *gammu*.

Ada beberapa hal yang perlu diperhatikan dalam mengisi *form* konfigurasi diatas. Berikut ini adalah penjelasan konfigurasi dari gambar 4.2.

```
port = /dev/ttyUSB0 # disesuaikan port mana Modem terdeteksi
connection = at115200 # disesuaikan dengan jenis modem yang
digunakan dan dapat dilihat pada website gammu
synchronizetime = yes
logfile = /etc/gammulog
logformat = textall
use_locking = -
```

Setelah melakukan pengkonfigurasian pilih *save* untuk menyimpan *file* konfigurasi pada direktori */home/honey/.gammurc*.

8. Konfigurasi berkas `/etc/gammu-smsdrc` dengan mengetikkan perintah berikut.

```
honey@honey-Aspire:~$ sudo nano /etc/gammu-smsdrc
```

```
[gammu]
# Please configure this!
port = /dev/ttyUSB0
connection = at115200
# Debugging
# logformat = textall
[smsd]
PIN = ''
service = sql
river = native_mysql
DeliveryReport = sms
logfile = /etc/smsdlog
# Increase for debugging information
debuglevel = 1 #
```

9. Cek *identify modem* dengan perintah berikut.

```
honey@honey-Aspire:~$ sudo gammu --identify
```

```
honey@honey-Aspire:~$ sudo gammu --identify
Device       : /dev/ttyUSB0
Manufacturer : huawei
Model        : unknown (E153)
Firmware     : 11.609.16.00.00
IMEI         : 354429042227267
SIM IMSI     : 510890980988980
```

**Gambar 4.3** *Gammu* dapat membaca *modem* yang digunakan.

Jika muncul tampilan seperti gambar 4.3 maka *gammu* dapat membaca *modem* yang digunakan.

10. Melakukan pengecekan untuk memastikan *sms gateway* dapat mengirim pesan dengan menggunakan perintah berikut.

```
honey@honey-Aspire:~$ sudo gammu --sendsms text 085739533002
```

```
honey@honey-Aspire:~$ sudo gammu --sendsms text 087735933002
Enter the message text and press Ctrl+D:
Warning: No chars read, assuming it is okay!
If you want break, press Ctrl+C..
Sending SMS 1/1...waiting for network answer..OK, message reference=156
honey@honey-Aspire:~$
```

**Gambar 4.4** *Sms gateway* dapat mengirimkan *sms*.



Pada gambar 4.4 menjelaskan bahwa *sms gateway* dapat mengirimkan pesan pada nomer yang dituju.

#### 4.1.4 Konfigurasi *Honeyd* dengan *Gammu*

Konfigurasi *honeyd* dengan *gammu* digunakan untuk mengirimkan paket yang tidak terotorisasi berupa *sms* kepada administrator. Program yang dibuat menggunakan *shell programming*, dimana *script* program yang dibuat dapat berjalan pada sistem operasi *unix* dan *linux*. Program tersebut digunakan untuk mengirimkan *sms* secara otomatis. Penulis membuat dua buah *file* yaitu *file cek.sh* yang berisi *script* program dan *file* baris yang mencatat jumlah baris *log file*. Berikut ini adalah penjelasan dari *script* program yang terdapat pada *file cek.sh* yang digunakan untuk menghubungkan *honeyd* dengan *gammu*.

```
#!/bin/bash
while :
do
a=`cat baris`
b=`wc -l /var/log/honeypot/honeyd.log | awk '{print $1}'`
c=`expr $b - $a`
if [ $a != $b ]; then
echo `tail -$c /var/log/honeypot/honeyd.log | grep -v MARK | grep '[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}' | awk '{print $2}'`
| gammu --sendsms text 087739533002`
echo `tail -$c /var/log/honeypot/honeyd.log | grep ^--MARK | awk '{print $6}' | sed -r 's/[0-9]{4}"/,/' | awk -F\, '{print $1}' | gammu --sendsms text 087739533002`
echo $b > baris
fi
sleep 120
done
```

Program tersebut menggunakan perulangan dengan menggunakan perintah *while do done*. Pengulangan dengan *while* digunakan untuk mengulang suatu perintah, selama perintah yang dijalankan bernilai benar. Dalam program ini, perulangan tersebut digunakan untuk mengecek jumlah baris pada *log file honeyd*. Berikut ini adalah penjelasan dari *script* program yang terdapat pada *file cek.sh* yang memproses pemfilteran *log file honeyd* dan melakukan pengiriman melalui *sms*.

- a. Variabel a berisi perintah **cat baris**, digunakan untuk menampilkan isi dari *file* baris. *File* baris mencatat jumlah baris *log file* sebelumnya. Ketika perintah tersebut dieksekusi di **terminal** maka akan muncul jumlah baris *log file* sebelumnya.
- b. Variabel b berisi perintah **wc -l /var/log/honeypot/honeyd.log | awk '{print \$1}'**, digunakan untuk menghitung jumlah kata atau baris *log file* yang tersimpan direktori **/var/log/honeypot/honeyd.log**. Ketika perintah tersebut dieksekusi di **terminal** maka akan muncul jumlah baris *log file* di ikuti dengan di rektori *log file* *honeyd* tersebut, seperti contoh **57423 /var/log/honeypot/honeyd.log**. Kemudian dilanjutkan dengan perintah **awk '{print \$1}'** untuk mengambil *field* pertama dari baris **57423 /var/log/honeypot/honeyd.log** sehingga yang ditampilkan hanya **57423**.
- c. Variabel c berisi perintah **expr \$b - \$a**, pada perintah tersebut terdapat operator **expr** yang digunakan untuk menghitung selisih jumlah *log file* yang terletak pada *file* *honeyd.log* dan *file* *baris*.
- d. Perintah, **if [ \$a != \$b ]; then**, menjelaskan jika jumlah *variabel a* tidak sama dengan *variabel b* maka akan dilanjutkan. Pada perintah selanjutnya terdapat dua opsi, terlihat berbeda namun, pada dasarnya sama digunakan untuk menyeleksi dan hanya mengambil *field protocol* pada baris *log file*. Perbedaan tersebut dikarenakan setiap pola serangan pasti akan menghasilkan jenis *log file* yang berbeda. Selain itu bertujuan agar dapat mengetahui informasi penting dalam baris *log file* mengingat terbatasnya jumlah karakter *sms* yang hanya memuat 140 buah karakter.
- e. Perintah **echo `tail -\$c /var/log/honeypot/honeyd.log | grep -v MARK | grep '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | awk '{print \$2}' | gammu --sendsms text 087739533002`**. Perintah tersebut untuk menyeleksi baris *log file* dengan tipe seperti di bawah ini.

2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 33250 192.168.1.6 1001: 60 S
2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 60232 192.168.1.6 16992: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 38739 192.168.1.6 5811: 60 S

Pada perintah diatas terdapat lima buah proses dari penyeleksian baris *log file* sampai melukan pengiriman.

1. Proses *pertama*, perintah `tail -$c /var/log/honeypot/honeyd.log` digunakan untuk menampilkan 10 baris terakhir dari *log file* pada perhitungan yang dilakukan pada *variable c*.
  2. Proses *kedua*, pada proses ini sampai dengan proses ke lima, terdapat aturan *regular expression* untuk menyeleksi setiap *field*. Perintah `grep -v MARK` digunakan untuk menyeleksi kata **MARK** pada setiap baris *log file*. Dari contoh baris *log file* diatas tidak terdapat kata **MARK**, namun dalam keadaan sebenarnya akan selalu nampak karena menandakan adanya pengaksesan terhadap *service* yang ada pada *server* palsu.
  3. Proses *ketiga*, perintah `grep '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'`, maksud dari perintah tersebut adalah untuk menyeleksi semua angka pada baris *log file* dari angka 0 sampai dengan 9 dengan ketentuan minimal 1 angka dan maksimal 3 angka yang di ikuti dengan tanda titik (.). Jadi, perintah tersebut untuk menyeleksi *ip address* pada baris *log file*.
  4. Proses *ke-empat*, perintah `awk '{print $2}'` digunakan untuk mengambil *field* ke dua dari baris *log file*, dimana baris kedua adalah *protocol*. Jadi hanya tinggal tulisan `tcp(6)` dari baris *log file* tersebut.
  5. Proses *kelima*, perintah `gammu --sendsms text 087739533002` digunakan untuk mengirimkan *sms* ke nomor tujuan tersebut. Sehingga jika terdapat baris *log file* seperti contoh diatas maka hanya tulisan `tcp(6)` saja yang akan terkirim melalui *sms*.
- f. Perintah `echo `tail -$c /var/log/honeypot/honeyd.log | grep ^--MARK | awk '{print $6}' | sed -r 's/[0-9]{4}"/,/' | awk -F\, '{print $1}' | gammu --sendsms text 087739533002``. Perintah tersebut untuk menyeleksi baris *log file* dengan tipe seperti di bawah ini

```
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 38739 192.168.1.6 5811: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 37589 192.168.1.6 30951: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "ssh", "", "", ""
```

Hampir sama dengan perintah pada *point e*, pada perintah ini terdapat enam proses. Berikut ini adalah penjelasan dari masing-masing proses.

1. Proses pertama, perintah ***tail -\$c /var/log/honeypot/honeyd.log*** digunakan untuk menampilkan 10 baris terakhir dari *log file* pada perhitungan yang dilakukan pada *variable c*.
  2. Proses kedua, perintah ***grep ^--MARK*** digunakan untuk menyeleksi baris yang dimulai dengan kata ***--MARK*** pada setiap baris *log file*.
  3. Proses ketiga, perintah ***awk '{print \$6}'*** digunakan untuk mengambil *field* ke-6, jadi hanya tinggal tulisan ***2012", "ssh", "", "", ""***.
  4. Proses ke-empat, perintah ***sed -r '[0-9]{4}'*** digunakan untuk menyaring/menghilangkan semua angka dari 0 sampai dengan 9 yang berjumlah 4 digit yang di ikuti dengan tanda (") pada baris ***2012", "ssh", "", "", ""***. Sehingga hanya tinggal tulisan ***"ssh", "", "", ""***.
  5. Proses kelima, perintah ***awk -F\, '{print \$1}'*** digunakan untuk mengolah kata pada baris ***"ssh", "", "", ""*** yang nantinya akan diambil dari kata pertama yang dipisahkan oleh tanda petik. Jadi, baris *log file* diatas hanya tinggal tersisa kata ***"ssh"***.
  6. Proses ke-enam ***gammu --sendsms text 087739533002*** digunakan untuk mengirim *sms*. Jadi, jika terdapat contoh baris *log file* seperti contoh diatas maka yang akan terkirim hanya tulisan ***"ssh"***.
- g. Pada baris berikutnya terdapat perintah ***echo \$b > baris***. Perintah tersebut digunakan untuk menampilkan baris dengan mengeceknya terlebih dahulu. Jika baris *log file* dalam *honeyd.log* lebih besar dengan jumlah baris yang tercatat sebelumnya pada *file* baris, maka selisih dari jumlah baris tersebut yang akan di kirimkan lewat *sms*.

- h. Baris terakhir terdapat perintah *sleep 120*, perintah tersebut digunakan untuk mengatur pengiriman *sms*. Jika dalam rentang waktu 120 detik tidak ada penambahan jumlah baris *log file* maka pengiriman tidak akan dilakukan.

## 4.2 Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui apakah *honeypot* dapat bekerja untuk merekam paket yang tidak terotorisasi pada jaringan komputer. Teknik yang digunakan ada 3 yaitu teknik *scanning*, *brute force* dan *os fingerprinting*. Dalam pengujian ini juga akan dilihat apakah *honeypot* dapat mengirimkan *sms* tersebut atau tidak. Berikut ini adalah langkah-langkah pengujian yang dilakukan.

### 4.2.1 Teknik Scanning

Percobaan yang dilakukan pada pengujian teknik *scanning* ada dua tahap. Tahap pertama untuk mengetahui jumlah *host* yang aktif pada jaringan tersebut. Tahap kedua untuk mengetahui *port* mana saja yang terbuka pada *host* di jaringan tersebut, terutama yang menjadi pusat perhatian disini adalah pendeteksian *port* yang terbuka pada *server* palsu.

Pengujian ini menggunakan *tools nmap* yang telah terinstal pada *backtrack*. Pengujian tahap pertama dilakukan dengan melakukan *scanning* terlebih dahulu pada *network* 192.168.1.0/24 menggunakan perintah sebagai berikut.

```
root@bt:~# nmap -sT 192.168.1.0/28 -Pn
```

Hasil dari perintah tersebut akan menampilkan *ip address* yang aktif pada jaringan tersebut.



```
Nmap scan report for 192.168.1.15
Host is up.
All 1000 scanned ports on 192.168.1.15 are filtered

Nmap done: 16 IP addresses (16 hosts up) scanned in 138.96 seconds
root@bt:~#
```

**Gambar 4.5** Hasil *scanning network* 192.168.1.0/28.

Pada gambar diatas dapat dilihat terdapat keterangan 16 *host up*. Berarti dapat disimpulkan bahwa *ip address server* palsu 192.168.1.6 dan 192.168.1.10

juga terdeteksi sebagai *host up* pada jaringan 192.168.1.0/28. Kemudian pada tahap kedua akan coba *scanning port* yang aktif pada salah satu *server* palsu dengan menggunakan perintah sebagai berikut.

```
root@bt:~# nmap -sT 192.168.1.6 -Pn
```

Dari *scanning* menggunakan perintah di atas diperoleh hasil seperti gambar di bawah ini yang menunjukkan *port* yang terbuka pada *host* dengan *ip address* 192.168.1.6 yang merupakan *server* palsu.

```
root@bt:~# nmap -sT 192.168.1.6 -Pn
Starting Nmap 5.51 ( http://nmap.org ) at 2012-03-03 01:21 WIT
Nmap scan report for 192.168.1.6
Host is up (0.011s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
```

**Gambar 4.6** Hasil *scanning port* pada *server* palsu 192.168.1.6.

Pada proses *scanning* diatas diketahui terdapat *port* yang terbuka diantaranya *port 21(ftp)*, *22(ssh)*, *23(telnet)* dan *80(http)*. Untuk membuktikan *port* tersebut benar-benar aktif maka akan dilakukan akses ke semua *port* diatas. Berikut ini adalah pengaksesan yang dilakukan pada *port* yang terbuka pada *host* 192.168.1.6.

#### 1. Akses FTP

Pada terminal *backtrack* akan dicoba akses pada *port 21* yang menjalankan *service ftp*. Perintah yang digunakan *ftp 192.168.1.6* kemudian tekan *enter*, maka akan terdapat keterangan “*connected to 192.168.1.6*” yang menandakan *service ftp* pada *host* tersebut aktif.

```
root@bt:~# ftp 192.168.1.6
Connected to 192.168.1.6.
-e 220 ProFTPD 1.2.4rc1 Server (SuSE) [bps-pc10.local.mynet] ready.
Name (192.168.1.6:root):
```

**Gambar 4.7** Pengaksesan *service ftp* pada *server* palsu.

## 2. Akses Telnet

Pengaksesan *telnet* pada *host* 192.168.1.6 untuk mengetahui apakah *service* tersebut aktif atau tidak. Pengaksesan menggunakan perintah *telnet 192.168.1.6* pada terminal *backtrack*. Setelah menjalankan perintah tersebut pada gambar dibawah ini terdapat keterangan “*connected to 192.168.1.6*” yang menandakan *service telnet* aktif pada *host* tersebut.



```

root@bt: ~
File Edit View Terminal Help
root@bt:~# telnet 192.168.1.6
Trying 192.168.1.6...
Connected to 192.168.1.6.
Escape character is '^]'.
Welcome to SuSE Linux 8.0 (i386) - Kernel 2.4.18-64GB-SMP (8)
bps-pc10 Login:

```

**Gambar 4.8** Pengaksesan *service telnet* pada *server* palsu.

## 3. Akses SSH

Pengaksesan *ssh* ini juga ditujukan untuk mengetahui apakah *service ssh* pada *host* 192.168.1.6 dapat berjalan. Pada terminal *backtrack* pengujian dilakukan dengan menggunakan perintah *ssh 192.168.1.6* lalu tekan *enter*. Berikut ini adalah tampilan dari hasil pengaksesan tersebut.



```

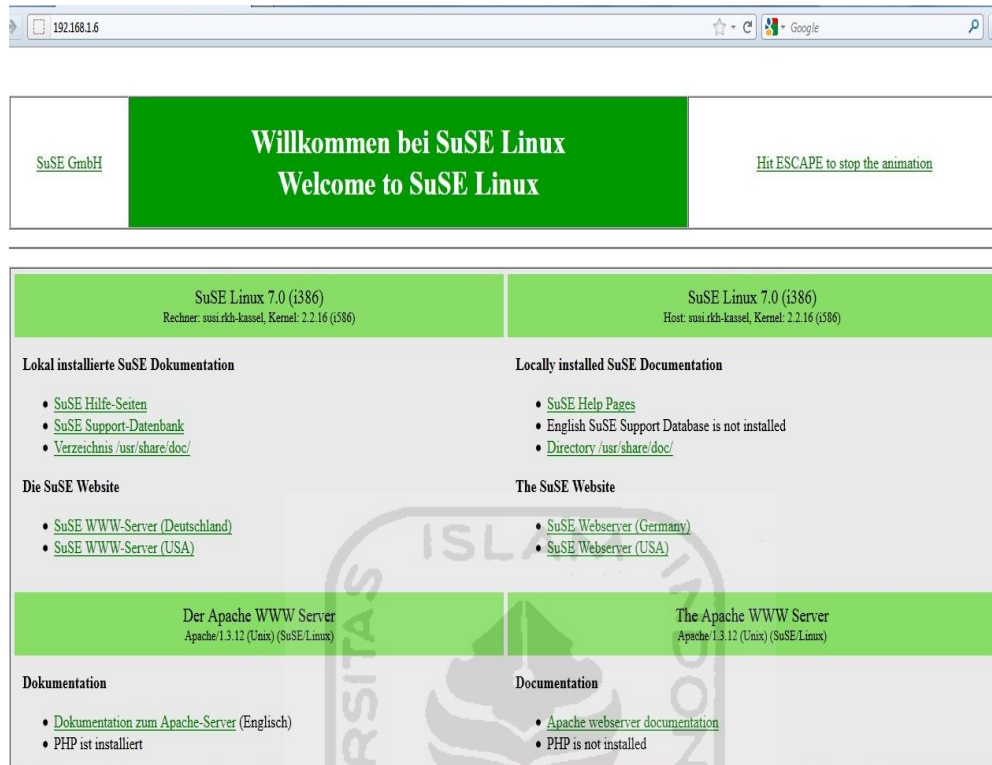
root@bt: ~
File Edit View Terminal Help
root@bt:~# ssh 192.168.1.6
Bad packet length 1397966893.
Disconnecting: Packet corrupt
root@bt:~#

```

**Gambar 4.9** Pengaksesan *service ssh* pada *server* palsu.

## 4. Akses HTTP

Pengaksesan *port 80* yang menjalankan *service http* diakses menggunakan *browser*. Pada *browser* ketikkan *ip address* 192.168.1.6 maka akan didapat hasil seperti gambar dibawah ini.

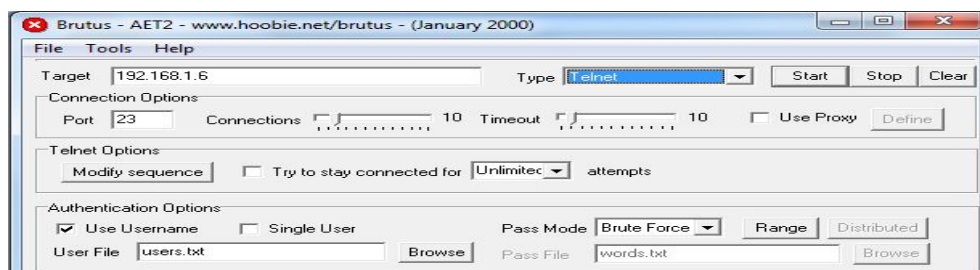


**Gambar 4.10** Pengaksesan *service http* pada *server palsu*.

Pada gambar 4.10 *service http* pada *host* 192.168.1.6 dapat berjalan. Setiap *link* yang ada pada halaman tersebut dapat juga diakses dan juga akan tercatat dalam *log file honeyd*.

#### 4.2.2 Teknik Brote Force

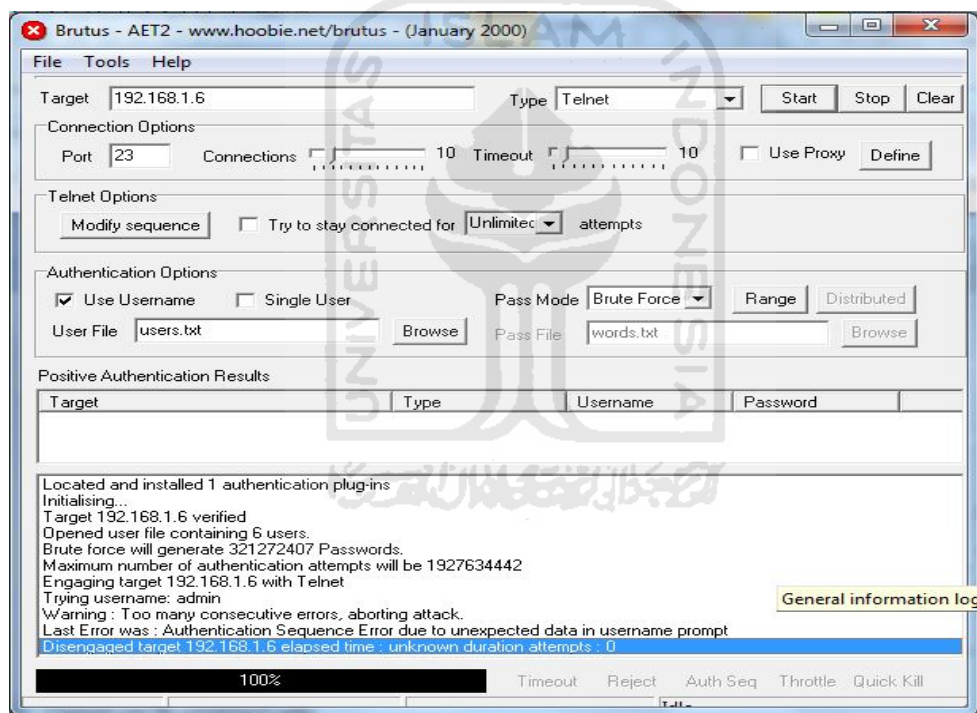
Pada pengujian ini menggunakan sebuah *tools exploit* yaitu *brutus alert*. Setelah melihat dari hasil *scanning port* percobaan ini dilakukan untuk mencoba mendapatkan hak akses terhadap *service telnet* pada *ip address* 192.168.1.6 yang merupakan *ip address* dari *server palsu*.



**Gambar 4.11** *Brutus alert* untuk melakukan serangan *brute force*.



Cara menggunakan *brutus alert*, masukan *ip address* target ke dalam *form* target dan kemudian pada *form type* pilih *service* yang akan dicoba untuk diserang. Dalam pengujian kali ini peneliti menggunakan *service telnet*, dan kemudian klik tombol *start*. Maka proses pengidentifikasian *password* dan *username* yang digunakan untuk melakukan *telnet* berjalan. Dapat dilihat pada gambar, terdapat keterangan “*Trying username: admin*” yang berarti sedang mencoba memasukan username admin. Namun, pada percobaan ini hasil dari serangan *brute force* tidak diprioritaskan, tetapi hanya untuk mengetahui kinerja dari *honeypot* dalam merekam serangan ke dalam *log file honeypot*.



Gambar 4.12 Serangan *brute force*.

#### 4.2.3 Teknik OS Fingerprinting

Pengujian dengan teknik *os fingerprinting* digunakan untuk mengetahui salah satu sistem operasi server palsu yang digunakan oleh *ip address* 192.168.1.10. Perintah yang digunakan adalah sebagai berikut.

```
root@bt:~# ./xprobe2 -v -F 192.168.1.10
```

Setelah perintah tersebut dijalankan *xprobe2* akan menjalankan modul-modulnya yang berjumlah 16 modul sebagai proses pengidentifikasian sistem operasi yang digunakan. Berikut ini adalah proses pengidentifikasian sistem operasi.

```

root@bt:/pentest/scanners/xprobe2# ./xprobe2 -v -F 192.168.1.6
Xprobe-ng v.2.1 Copyright (c) 2002-2009 fyodor@00o.nu, ofir@sys-security.com, meder@
[+] Target is 192.168.1.6
[+] Loading modules.
[+] Following modules are loaded:
[x] ping:icmp_ping - ICMP echo discovery module
[x] ping:tcp_ping - TCP-based ping discovery module
[x] ping:udp_ping - UDP-based ping discovery module
[x] infogather:ttl_calc - TCP and UDP based TTL distance calculation
[x] infogather:portscan - TCP and UDP PortScanner
[x] fingerprint:icmp_echo - ICMP Echo request fingerprinting module
[x] fingerprint:icmp_tstamp - ICMP Timestamp request fingerprinting module
[x] fingerprint:icmp_amask - ICMP Address mask request fingerprinting module
[x] fingerprint:icmp_info - ICMP Information request fingerprinting module
[x] fingerprint:icmp_port_unreach - ICMP port unreachable fingerprinting module
[x] fingerprint:tcp_hshake - TCP Handshake fingerprinting module
[x] fingerprint:tcp_rst - TCP RST fingerprinting module
[x] app:smb - SMB fingerprinting module
[x] app:snmp - SNMPv2c fingerprinting module
[x] app:ftp - FTP fingerprinting tests
[x] app:http - HTTP fingerprinting tests
[+] 16 modules registered
[+] Initializing scan engine
[+] Running scan engine
fingerprint:icmp_tstamp has not enough data
Executing ping:icmp_ping
Executing fingerprint:icmp_port_unreach
Executing fingerprint:icmp_echo
fingerprint:tcp_hshake has not enough data
Executing fingerprint:tcp_rst
Executing fingerprint:icmp_amask
Executing fingerprint:icmp_info
Executing fingerprint:icmp_tstamp

```

**Gambar 4.13** *Xprobe2* menjalankan modulnya untuk proses identifikasi.

Hasil dari perintah tersebut akan menampilkan jenis sistem operasi yang digunakan oleh *ip address* 192.168.1.10. Berikut ini adalah tampilan hasil *os fingerprinting*.

```

executing fingerprint:icmp_tstamp
app:smb has not enough data
Executing app:snmp
ping:tcp_ping has not enough data
ping:udp_ping has not enough data
infogather:tll_calc has not enough data
Executing infogather:portscan
Executing app:ftp
Executing app:http
[+] Signature looks like:
[+] "Microsoft Windows NT 4 Workstation Service Pack 6a" (95%)
[+] Generated signature for 192.168.1.10:
fingerprint {
  OS_ID =
  #Entry inserted to the database by:
  #Entry contributed by:
  #Date:
  #Modified:
  icmp_addrmask_reply = n
  icmp_addrmask_reply_ip_id = !0
  icmp_addrmask_reply_ttl = <255
  icmp_echo_code = 0
  icmp_echo_df_bit = 1
  icmp_echo_ip_id = !0
  icmp_echo_reply = u

```

**Gambar 4.14** Hasil dari serangan *os fingerprinting*.

Pada gambar 4.14 dapat dilihat terdapat keterangan *Signature looks like: "Microsoft Windows NT 4 Workstation Service Pack 6a" (95%)*. Pada keterangan tersebut terdapat prosentase yang menunjukkan kecocokan dengan sistem aslinya. Sistem operasi palsu yang dijalankan dengan *ip address* 192.168.1.10 adalah *Windows NT 3.10*. Hasil yang diperoleh mendekati kebenaran hanya versi dari sistem operasi tersebut yang berbeda.

#### 4.3 Hasil Penelitian

Pengimplementasian *honeypot* dilakukan untuk memperoleh hasil penelitian dengan menjalankan metode-metode yang telah di jelaskan pada BAB III. Hasil yang diperoleh pada penelitian ini berupa *log file* dan pengiriman *sms*. Penelitian yang dijalankan lebih menitikberatkan pada *honeypot* untuk dapat merekam paket tidak terotorisasi dan mengirimkannya melalui *sms*.

Paket yang tidak terotorisasi diperoleh dari koneksi yang tertuju pada *server* palsu. *Server* palsu yang dibentuk antara lain *Windows NT 3.10* dengan *ip address* 192.168.1.10 dan *Linux Suse 8.0* dengan *ip address* 192.168.1.6. Setiap ada koneksi yang menuju *server* palsu tersebut maka akan tercatat sebagai koneksi yang ilegal dan terekam dalam *log file honeypot*.

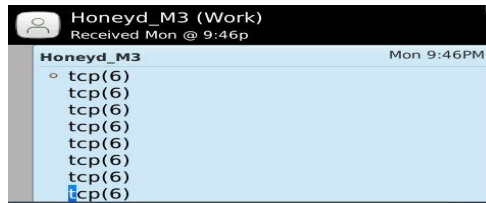


```

2012-03-03-15:23:06.1243 tcp(6) - 192.168.1.5 51918 192.168.1.6 4006: 60 S
2012-03-03-15:23:06.1244 tcp(6) - 192.168.1.5 49852 192.168.1.6 2492: 60 S
2012-03-03-15:23:06.1244 tcp(6) - 192.168.1.5 42917 192.168.1.6 1069: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "telnet", "", "", "",
",
--ENDMARK--
2012-03-03-15:23:06.1250 tcp(6) - 192.168.1.5 53625 192.168.1.6 7001: 60 S
2012-03-03-15:23:06.1251 tcp(6) - 192.168.1.5 55123 192.168.1.6 1048: 60 S
.....
2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 33250 192.168.1.6 1001: 60 S
2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 60232 192.168.1.6 16992: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 38739 192.168.1.6 5811: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 37589 192.168.1.6 30951: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "ssh", "", "", ""
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 52346 192.168.1.6 1556: 60 S
"2012-03-03-15:23:06.1692 tcp(6) - 192.168.1.5 55377 192.168.1.6 1721: 60 S
",
2012-03-03-15:23:06.1692 tcp(6) - 192.168.1.5 34647 192.168.1.6 89: 60 S
--ENDMARK--
.....
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 42118 192.168.1.6 5925: 60 S
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 48982 192.168.1.6 2135: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "pro-ftp/FTP", "", "", ""
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 37893 192.168.1.6 8873: 60 S
"2012-03-03-15:23:06.1709 tcp(6) - 192.168.1.5 45653 192.168.1.6 8011: 60 S
2012-03-03-15:23:06.1709 tcp(6) - 192.168.1.5 33277 192.168.1.6 32771: 60 S
2012-03-03-15:23:06.1710 tcp(6) - 192.168.1.5 57887 192.168.1.6 2604: 60 S
",
2012-03-03-15:23:06.1712 tcp(6) - 192.168.1.5 32888 192.168.1.6 6007: 60 S
--ENDMARK--
....
2012-03-03-15:23:06.1905 tcp(6) - 192.168.1.5 34231 192.168.1.6 1045: 60 S
2012-03-03-15:23:06.1905 tcp(6) - 192.168.1.5 56356 192.168.1.6 81: 60 S
2012-03-03-15:23:06.1906 tcp(6) - 192.168.1.5 40299 192.168.1.6 2006: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "apache/HTTP", "", "", ""
--ENDMARK--

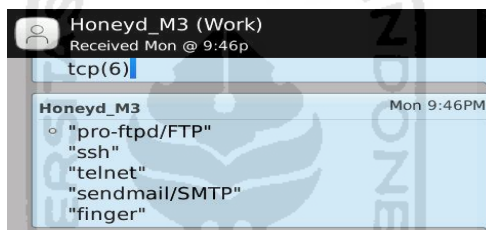
```

Dari *log file* diatas, dapat dilihat semua service yang diakses oleh 192.168.1.3 dalam waktu yang bersamaan. Berikut ini adalah tampilan *sms* yang dikirim yang menginformasikan adanya rekaman *log file honeyd* dari teknik *scanning*.



**Gambar 4.16** Paket *tcp* yang dikirim pada proses *scanning*.

Pada bagian metodologi telah dijelaskan, bahwa pemfilteran dilakukan dalam dua perintah. Gambar 4.16 menjelaskan *sms* yang dikirimkan memuat informasi paket TCP dikirimkan, sedangkan gambar 4.17 memuat informasi *service* yang diakses. Hasil dari pengiriman *sms* yang mengandung informasi *scanning* dapat ditandai dengan terkirimnya semua layanan yang diakses.



**Gambar 4.17** Layanan yang diakses pada saat proses *scanning*.

Selain melakukan *scanning* pada salah satu *server* palsu, pengujian juga dilakukan pada *port* yang terbuka pada *host* 192.168.1.6. Hasilnya, *log file* akan merekam aktifitas sesuai dengan *service* yang diakses. Berikut ini adalah hasil dari *log file* pengujian tersebut.

```

2012-03-12-01:26:45.5049 tcp(6) S 192.168.1.5 34495 192.168.1.6 22
--MARK--, "Mon Mar 12 01:26:45 WIT 2012", "ssh", "", "", "",
"SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu6
",
--ENDMARK--
2012-03-12-01:26:45.5527 tcp(6) E 192.168.1.5 34495 192.168.1.6 22: 871 64
2012-03-12-01:27:21.9752 tcp(6) S 192.168.1.5 33558 192.168.1.10 22
2012-03-12-01:28:20.9433 tcp(6) E 192.168.1.5 33558 192.168.1.10 22: 0 0
2012-03-12-01:28:32.3758 tcp(6) S 192.168.1.5 33935 192.168.1.6 21
--MARK--, "Mon Mar 12 01:28:32 WIT 2012", "pro-ftp/FTP", "", "", "",
"USER quit
SYST
QUIT

```

```

",
--ENDMARK--
2012-03-12-01:32:34.4819 tcp(6) E 192.168.1.5 33935 192.168.1.6 21: 23 189
2012-03-12-01:33:02.7893 tcp(6) S 192.168.1.5 52402 192.168.1.6 22
--MARK--, "Mon Mar 12 01:33:02 WIT 2012", "ssh", "", "", "",
"SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu6
",
--ENDMARK--
2012-03-12-01:33:02.8094 tcp(6) E 192.168.1.5 52402 192.168.1.6 22: 871 64
2012-03-12-01:47:04.8072 tcp(6) E 192.168.1.3 51852 192.168.1.6 80: 364 646
2012-03-12-02:32:00.2111 tcp(6) S 192.168.1.5 35752 192.168.1.6 23
--MARK--, "Mon Mar 12 02:32:00 WIT 2012", "telnet", "", "", ""

```

Log file diatas adalah hasil dari pengujian pada port 21(ftp),22(ssh) dan 23(telnet). Berbeda dengan baris log file yang merekam pengujian pada port 80(http), karena memuat halaman localhost pada host 192.168.1.6. Berikut ini adalah log dari pengaksesan port 80(http).

```

2012-03-12-01:40:17.1011 tcp(6) S 192.168.1.3 51819 192.168.1.6 80 [Windows
2000 RFC1323]
--MARK--, "Mon Mar 12 01:40:17 WIT 2012", "apache/HTTP", "", "", "",
"GET / HTTP/1.1
Host: 192.168.1.6
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101
Firefox/8.0.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
",
--ENDMARK--
2012-03-12-01:40:17.4573 tcp(6) S 192.168.1.3 51822 192.168.1.6 80 [Windows
2000 RFC1323]
2012-03-12-01:40:17.4588 tcp(6) S 192.168.1.3 51825 192.168.1.6 80 [Windows
2000 RFC1323]
2012-03-12-01:40:17.4599 tcp(6) S 192.168.1.3 51826 192.168.1.6 80 [Windows
2000 RFC1323]
2012-03-12-01:40:17.4601 tcp(6) S 192.168.1.3 51827 192.168.1.6 80 [Windows
2000 RFC1323]
--MARK--, "Mon Mar 12 01:40:17 WIT 2012", "apache/HTTP", "", "", "",
"--MARK--, "Mon Mar 12 01:40:17 WIT 2012", "apache/HTTP", "", "", "",
"--MARK--, "Mon Mar 12 01:40:17 WIT 2012", "apache/HTTP", "", "", ""

```

```

--MARK--, "Mon Mar 12 01:40:17 WIT 2012", "apache/HTTP", "", "", "",
""GET /gif/suse_150.gif HTTP/1.1
GET /gif/apache_pb.gif HTTP/1.1
GET /gif/penguin.gif HTTP/1.1
GET /gif/suse_button.gif HTTP/1.1
Host: 192.168.1.6
Host: 192.168.1.6
Host: 192.168.1.6
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101
Firefox/8.0.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101

```

Dari pengujian yang dilakukan terhadap *port* 80 yang ada pada *host* 192.168.1.6 membuktikan bahwa *service* tersebut dapat berjalan. Pola pada *log* yang merekam aktifitas terhadap *port* 80 berbeda dengan *port* 21, 22 dan 23. Namun, *sms* yang akan dikirimkan tetap sama yaitu *service* yang sedang diakses yaitu *apache/HTTP*.

#### 4.3.2 Hasil Pengujian Teknik Brute force

Pengujian dengan teknik *brute force* dilakukan dengan *tools exploit brutus alert*. Pada metode pengujian telah dijelaskan bagaimana cara melakukan percobaan serangan dengan menggunakan teknik tersebut. Secara teori, teknik *brute force* akan memasukan *username* dan *password* secara acak dan terus menerus kepada target yang diserang. Berikut ini akan ditampilkan sebagian dari isi *log file honeyd* yang merekam informasi serangan dengan teknik *brute force*.

```

2012-02-27-01:58:31.6574 honeyd log started -----
2012-02-27-02:05:38.2146 tcp(6) S 192.168.1.3 29901 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2165 tcp(6) S 192.168.1.3 29902 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2167 tcp(6) S 192.168.1.3 29903 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2167 tcp(6) S 192.168.1.3 29904 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2169 tcp(6) S 192.168.1.3 29905 192.168.1.6 23
[Windows 2000 RFC1323]
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", ""

```



```

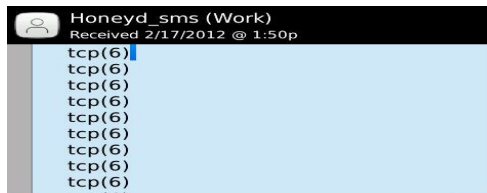
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"",
2012-02-27-02:05:38.2609 tcp(6) E 192.168.1.3 29887 192.168.1.6 23: 0 79
2012-02-27-02:05:38.2610 tcp(6) - 192.168.1.3 29887 192.168.1.6 23: 40 R
[Windows 2000 RFC1323]
--ENDMARK--
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"admin
admin
admin
admin
admin

aaa
access
abc
adrian
academic
ada
academia
admin

--MARK--, "Mon Feb 27 02:05:41 WIT 2012", "telnet", "", "", "",
"",
--ENDMARK--
2012-02-27-02:05:41.2209 tcp(6) E 192.168.1.3 29905 192.168.1.6 23: 0 79

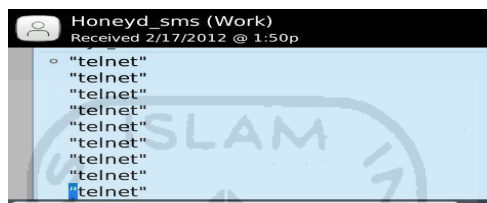
```

Pada *log file* tampak pengaksesan yang mencoba memasukan *username* dan *password*. Berikut ini adalah tampilan dari *sms* yang menginformasikan adanya rekaman *log file honeyd* dari teknik *brute force*.



**Gambar 4.18** Sms paket TCP pada serangan *brute force*.

Gambar 4.18 menginformasikan *sms* yang dikirimkan memuat informasi paket TCP dikirimkan, sedangkan pada gambar 4.19 menginformasikan layanan yang sedang diakses menggunakan teknik *brute force*. Hasil dari pengiriman *sms* yang mengandung informasi *telnet* ditandai dengan tertulisnya layanan *telnet* secara berulang-ulang pada *sms* yang dikirim. Hal itu juga berlaku untuk *service* yang lainnya. Jika yang diserang adalah *port 80* yang menjalankan *service apache* maka ada didapati *sms* yang bertuliskan *apache/HTTP*.



**Gambar 4.19** *Sms* layanan *telnet* yang diserang dengan teknik *brute force*.

### 4.3.3 Teknik OS Fingerprinting

Pengujian dengan teknik *os fingerprinting* menggunakan *tools xprobe2* yang mana *tools* tersebut telah ada pada sistem operasi *backtrack*. Pada metode pengujian serangan diarahkan kepada *ip address* 192.168.1.10 yang merupakan *ip address server* palsu Windows NT 3.10. Berikut ini akan ditampilkan sebagian hasil dari *log file honeyd* yang merekam serangan *os fingerprinting*.

```

2012-03-04-02:45:51.6715 tcp(6) S 192.168.1.5 43824 192.168.1.10 65535
2012-03-04-02:45:51.6716 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6717 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6717 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6721 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.7991 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7992 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7993 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7994 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:54.5114 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:54.5114 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28

```

```

2012-03-04-02:45:54.5116 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:54.5117 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:57.5115 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5115 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5117 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5117 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:46:00.4313 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4313 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4315 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4316 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:49.2796 udp(17) E 192.168.1.5 53 192.168.1.10 65534: 76 0
2012-03-04-02:46:49.2796 udp(17) E 192.168.1.5 53 192.168.1.10 65534: 76 0
2012-03-04-02:46:49.2798 udp(17) E 192.168.1.5 53 192.168.1.10 65534: 76 0
2012-03-04-02:46:49.2802 udp(17) E 192.168.1.5 53 192.168.1.10 65534: 76 0
2012-03-04-02:47:05.4379 udp(17) E 192.168.1.5 47122 192.168.1.10 161: 259 0
2012-03-04-02:47:05.4380 udp(17) E 192.168.1.5 47122 192.168.1.10 161: 259 0
2012-03-04-02:47:05.4384 udp(17) E 192.168.1.5 47122 192.168.1.10 161: 259 0
2012-03-04-02:47:05.4385 udp(17) E 192.168.1.5 47122 192.168.1.10 161: 259 0

```

Pada *log file* diatas, sekilas jika dilihat akan tampak seperti akses *ping* biasa. Namun, jika diamati semua *protocol* yang diakses TCP, UDP dan ICMP. Hal ini disebabkan karena proses identifikasi yang dilakukan oleh modul *xprobe2* dengan menjalankan *scan engine* untuk mengetahui sistem yang digunakan. Berikut ini adalah hasil pengiriman *sms* yang memuat informasi *os fingerprinting*.



**Gambar 4.20** *Protocol* yang diakses pada serangan *os fingerprinting*gn.

#### 4.4 Pembahasan

Pengkonfigurasi *honeyd* dan *sms gateway* pada penelitian ini digunakan untuk mengetahui paket yang tidak terotorisasi yang terekam pada *honeyd* dan mengirimkannya melalui *sms*. Detail dari setiap proses yang ada pada penelitian

ini akan dapat dilihat pada penjelasan yang akan dijabarkan pada subbab pembahasan yang ada dibawah ini.

#### 4.4.1 Skenario Pengujian

Dalam pengujian pada sebuah objek, perlu dirancang sebuah skenario. Skenario sendiri memiliki arti sebuah alur yang disusun sedemikian hingga untuk menghasilkan sebuah peristiwa sesuai dengan yang diinginkan. Pada penelitian ini skenario yang dimaksud adalah rangkaian pengujian dengan menggunakan *tools exploit* yang ditujukan pada *honeypot* agar dapat mengetahui kinerja dari *honeypot* dalam menangkap paket yang tidak terotorisasi serta mengetahui kemampuannya dalam mengirimkan paket tersebut melalui *smtp*.

Berikut ini akan dijelaskan mengenai scenario pengujian yang akan dilakukan :

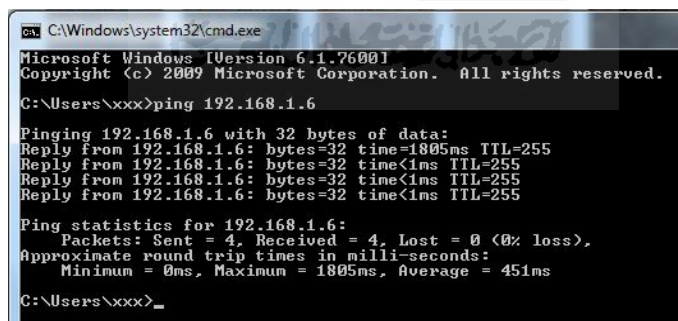
1. Merujuk pada topologi yang telah dijelaskan pada gambar 3.1 terdapat dua *server* dan dua *client*. Dua *server* tersebut yaitu *server* sebenarnya dan *server honeypot*. Sistem operasi yang digunakan pada *server honeypot* adalah *linux ubuntu* dimana di dalamnya telah terkonfigurasi *server* palsu. *Server* palsu yang terbentuk antara lain *Windows NT 3.10* dengan *ip address* 192.168.1.10 dan *Linux 2.4.7* dengan *ip address* 192.168.1.6. Kemudian terdapat dua *client* yang akan berperan sebagai *intruder* yang akan melakukan serangan pada jaringan dengan *ip address* 192.168.1.3 dan 192.168.1.4. Sistem operasi yang digunakan oleh *intruder* adalah *linux backtrack 5 revolution*. Pada sistem operasi *backtrack 5 revolution* telah terdapat *tools exploit*, namun tidak semua *tools* akan dicoba untuk melakukan serangan. Teknik yang akan digunakan dalam penyerangan antara lain *scanning*, *brute force* dan *os fingerprinting*. *Tools* yang akan dipakai antara lain *nmap* untuk melakukan serangan *port scanning*, *xprobe2* untuk melakukan serangan *os fingerprinting* dan *brutus alert* untuk melakukan serangan *brute force*.
2. Pada *server honeypot* akan dijalankan konfigurasi *honeypot* yang mengemulasikan *server* palsu. Pengaktifan *honeypot* juga harus di ikut

dengan pengaktifan *file cek.sh* untuk menjalankan *service sms gateway* yang digunakan untuk mengirimkan *log file*.

3. Pada komputer *intruder* akan di jalankan *tools exploit* yang disebutkan pada *point* pertama untuk melakukan pengujian.
4. Pada proses selanjutnya akan dilakukan pengamatan pada *server honeyd* untuk mengetahui apakah *honeyd* dapat bekerja melakukan *logging* dan mampu mengirimkan *log file* yang sudah terfilter melalui *sms* yang berisi informasi paket yang tidak terotorisasi.

#### 4.4.2 Performa Honeyd

*Honeyd* akan berkerja jika menerima serangan yang tertuju pada *server* palsu yang dibentuknya. *Server* palsu yang terbentuk memiliki *personality* dan mampu menjalankan *service* layaknya sistem sebenarnya. *Honeyd* mampu mensimulasikan *service* TCP, UDP dan ICMP maka dari itu *server* palsu yang terbentuk dapat menjalankan *service* tersebut seperti sistem aslinya. Misalkan saja *honeyd* dapat merespon layaknya *service* ICMP. Berikut ini adalah gambar 4.21 yang menjelaskan *honeyd* mampu merespon ICMP dengan melakukan tes *ping* ke *ip address* 192.168.1.6 yang merupakan *ip address server* palsu *Linux 2.4.7*.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\xxx>ping 192.168.1.6

Pinging 192.168.1.6 with 32 bytes of data:
Reply from 192.168.1.6: bytes=32 time<1ms TTL=255
Reply from 192.168.1.6: bytes=32 time<1ms TTL=255
Reply from 192.168.1.6: bytes=32 time<1ms TTL=255
Reply from 192.168.1.6: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.1.6:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1805ms, Average = 451ms

C:\Users\xxx>_
  
```

**Gambar 4.21** *Honeyd* mampu menjalankan layanan ICMP.

*Honeyd* mampu berinteraksi dengan banyak penyerang sekaligus dan mampu mengemulasikan *service* yang terbentuk secara bersamaan. Ketika terdapat *intruder* masuk di dalamnya dan berinteraksi dengan *honeyd* kemudian melakukan serangan dan setelah puas intruder akan menyudahi serangan tersebut dan memutuskan koneksi dengan *server* palsu. Namun, *service* yang dijalankan

pada *honeyd* tidak akan putus dan akan menunggu koneksi berikutnya. Jadi, *service* yang dijalankan *honeyd* akan tetap berjalan terus selama *honeyd* aktif.

*Server* palsu yang dibentuk *honeyd* memerlukan *ip address* yang berasal dari *unused ip address* pada jaringan. Untuk dapat mengkalim *unused ip address* tersebut *honeyd* memerlukan *arpd* atau *farpd*. *Farpd* ini digunakan untuk mengklaim *unused ip address* yang nantinya akan dimasukkan ke dalam *MAC address* dari *server honeypot*. Sehingga seolah-olah *unused ip address* yang ada pada jaringan dipakai oleh sebuah *host*.

```

root@honey-Aspire:~# farpd -d -i eth0 192.168.1.0/24
arpd[26295]: listening on eth0: arp and (dst net 192.168.1.0/24) and not ether src 00:1b:24:92:91:5c
arpd[26295]: arpd_lookup: no entry for 192.168.1.6
arpd[26295]: arpd_send: who-has 192.168.1.6 tell 192.168.1.5
arpd[26295]: arpd_send: who-has 192.168.1.6 tell 192.168.1.5
arpd[26295]: arpd_rcv_cb: 192.168.1.6 still discovering (2)
arpd[26295]: arp_reply 192.168.1.6 is-at 00:1b:24:92:91:5c
arpd[26295]: arp_reply 192.168.1.6 is-at 00:1b:24:92:91:5c
arpd[26295]: arp_reply 192.168.1.6 is-at 00:1b:24:92:91:5c
arpd[26295]: arp_reply 192.168.1.6 is-at 00:1b:24:92:91:5c

```

**Gambar 4.22** *Farpd* mengklaim *ip address* 192.168.1.6 .

Pada gambar 4.22 menunjukkan bahwa *ip address* 192.168.1.6 yang merupakan *ip address* dari *server* palsu diklaim oleh *farpd*. Jadi, seolah-olah memang ada *host* yang menggunakan *ip address* tersebut padahal pada kenyataannya digunakan oleh *server* palsu yang dibentuk oleh *honeyd*.

*Farpd* akan diletakkan dimana lokasi *honeyd* berada guna mempermudah monitoring *ip address* pada jaringan terutama untuk memonitor *ip address* yang dipakai oleh *server* palsu. Jika terdapat *intruder* yang masuk kedalam jaringan dan mengirimkan paket data, maka *farpd* akan mengidentifikasinya dan melakukan *ARP replies* terhadap paket data tersebut. Jika terdapat respon maka akan dihubungkan dengan *ip address honeyd*. Jika tidak mendapat respon maka *farpd* akan melakukan *arp spoofing* untuk menjawab respon paket data *intruder* bahwa *MAC address* server palsu *honeyd* memiliki *ip address* komputer *intruder*. Dari hal tersebut maka akan dapat dipercaya *arp* respon yang berasal dari *honeyd* dan kecocokan *ip address* dengan *MAC address* yang dimiliki *honeyd*. Jadi, paket data tersebut akan diteruskan menuju *honeyd* dan dapat berinteraksi dengan *intruder*.

#### 4.4.3 Log file

*Log file honeyd* adalah tempat yang mencatat semua aktifitas yang tertuju pada *honeyd*. Segala macam koneksi yang tercatat ke dalam *log file honeyd* merupakan suatu aktifitas ilegal. Hal tersebut dikarenakan, *honeyd* akan bekerja jika menerima serangan. *Log file* yang tercatat pada *honeyd* berasal dari koneksi yang tertuju pada *server* palsu yang dibentuk *honeyd*. Bentuk dari *log* yang tercatat memiliki pola yang berbeda-beda. Perbedaan tersebut dikarenakan *service* yang diakses dan juga serangan yang dilakukan.

Pada *log file* memuat informasi paket yang tidak terotorisasi. Informasi paket data tersebut diketahui dengan melihat susunan yang ada pada *log file* tersebut. Berikut ini adalah penjelasan dari informasi yang termuat dalam *log file*.

- a. *Field* waktu koneksi, menjelaskan informasi waktu yang terjadinya koneksi ke *server* palsu yang meliputi tahun, bulan, tanggal dan jam terjadinya koneksi.
- b. *Field* layanan *protocol* dan *port*, menjelaskan *port* dan *protocol* yang digunakan yaitu TCP, UDP dan ICMP.
- c. *Field* keterangan koneksi, menjelaskan keterangan koneksi. Pada kolom koneksi terdapat keterangan S, E dan “-“, keterangan tersebut mempunyai tersendiri. S merupakan *start new connection*, E merupakan *end connection* dan “-“ bukan koneksi apapun, tetapi diakhir baris akan terdapat angka yang menandakan TCP *flag*. Pada keterangan E juga menjelaskan data yang dikirim pengakses dan data yang diterima *server* palsu yang terletak pada akhir baris *log file*.
- d. *Field* alamat *ip address* asal (*intruder*) dan tujuan (*ip Server* palsu) yang diikuti dengan alamat *port* asal dan tujuan.
- e. *Field* sistem operasi, menjelaskan sistem operasi yang digunakan untuk mengakses *server* palsu *honeyd*.

Berikut ini adalah table yang berisi informasi keterangan *log file* dari kelima point diatas.

Waktu koneksi	Port dan protocol	koneksi	ip asal dan port asal	ip tujuan dan port tujuan	Sistem operasi
2012-02-27-02:05:38	2165 tcp(6)	S	192.168.1.3 29902	192.168.1.6 23	[Windows 2000 RFC1323]
2012-02-27-02:05:38	2167 tcp(6)	-	192.168.1.3 29903	192.168.1.6 23	[Windows 2000 RFC1323]
2012-02-27-02:05:38	2609 tcp(6)	E	192.168.1.3 29887	192.168.1.6 23: 0 79	-

**Table 4.1** Tabel keterangan *log file honeyd*.

Selain dari kelima informasi yang termuat tersebut, pada *log file honeyd* juga mampu menangkap informasi yang dilakukan secara manual. Misalkan jika *intruder* mencoba mengakses *ftp*, melakukan *telnet* kepada *server* palsu dan mengakses *service apache*. *Honeyd* akan menampilkan informasi *log* yang mengakses layanan tersebut. Berikut ini adalah penjelasan dari *log file* hasil pengujian yang dilakukan dengan teknik *scanning*, *brute force* dan *os fingerprinting*.

#### 1. *Log file* Teknik Scanning

*Log file* yang merekam aktifitas *scanning* akan memuat informasi pengaksesan terhadap semua layanan yang ada. Berdasarkan hasil penelitian yang diperoleh maka dapat dilakukan pembahasan untuk mengetahui informasi yang ada pada *log file* tersebut. Pada pembahasan ini akan diambil beberapa contoh dari ratusan baris *log file* yang memuat informasi pengaksesan layanan pada *server* palsu. Berikut ini adalah pembahasannya.



```

2012-03-03-15:23:06.1243 tcp(6) - 192.168.1.5 51918 192.168.1.6 4006: 60 S
2012-03-03-15:23:06.1244 tcp(6) - 192.168.1.5 49852 192.168.1.6 2492: 60 S
2012-03-03-15:23:06.1244 tcp(6) - 192.168.1.5 42917 192.168.1.6 1069: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "telnet", "", "", "",
",
--ENDMARK--
2012-03-03-15:23:06.1250 tcp(6) - 192.168.1.5 53625 192.168.1.6 7001: 60 S
2012-03-03-15:23:06.1251 tcp(6) - 192.168.1.5 55123 192.168.1.6 1048: 60 S
....
2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 33250 192.168.1.6 1001: 60 S
2012-03-03-15:23:06.1690 tcp(6) - 192.168.1.5 60232 192.168.1.6 16992: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 38739 192.168.1.6 5811: 60 S
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 37589 192.168.1.6 30951: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "ssh", "", "", "",
",
2012-03-03-15:23:06.1691 tcp(6) - 192.168.1.5 52346 192.168.1.6 1556: 60 S
"2012-03-03-15:23:06.1692 tcp(6) - 192.168.1.5 55377 192.168.1.6 1721: 60 S
",
2012-03-03-15:23:06.1692 tcp(6) - 192.168.1.5 34647 192.168.1.6 89: 60 S
--ENDMARK--
....
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 42118 192.168.1.6 5925: 60 S
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 48982 192.168.1.6 2135: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "pro-ftp/FTP", "", "", "",
",
2012-03-03-15:23:06.1708 tcp(6) - 192.168.1.5 37893 192.168.1.6 8873: 60 S
"2012-03-03-15:23:06.1709 tcp(6) - 192.168.1.5 45653 192.168.1.6 8011: 60 S
2012-03-03-15:23:06.1709 tcp(6) - 192.168.1.5 33277 192.168.1.6 32771: 60 S
2012-03-03-15:23:06.1710 tcp(6) - 192.168.1.5 51753 192.168.1.6 4900: 60 S
2012-03-03-15:23:06.1711 tcp(6) - 192.168.1.5 45591 192.168.1.6 9618: 60 S
2012-03-03-15:23:06.1711 tcp(6) - 192.168.1.5 32827 192.168.1.6 4445: 60 S
2012-03-03-15:23:06.1712 tcp(6) - 192.168.1.5 59750 192.168.1.6 20005: 60 S
2012-03-03-15:23:06.1712 tcp(6) - 192.168.1.5 34594 192.168.1.6 35500: 60 S
",
2012-03-03-15:23:06.1712 tcp(6) - 192.168.1.5 32888 192.168.1.6 6007: 60 S
--ENDMARK--
....
2012-03-03-15:23:06.1905 tcp(6) - 192.168.1.5 34231 192.168.1.6 1045: 60 S
2012-03-03-15:23:06.1905 tcp(6) - 192.168.1.5 56356 192.168.1.6 81: 60 S
2012-03-03-15:23:06.1906 tcp(6) - 192.168.1.5 40299 192.168.1.6 2006: 60 S
--MARK--, "Sat Mar 3 15:23:06 WIT 2012", "apache/HTTP", "", "", "",
"

```

Dari log file diatas, pengaksesan server palsu terjadi pada 03-03-2012 pada pukul 15:23.06. Pengaksesan tersebut dilakukan oleh komputer 192.168.1.3 yang mengirimkan paket menuju 192.168.1.6. Koneksi

menggunakan layanan *protocol* TCP. Jika diamati dari semua baris diatas banyak sekali *destination port* yang diakses. Terjadinya koneksi tersebut dalam waktu yang hampir bersamaan dapat mengakses semua *service telnet, ftp, ssh* dan *apache*. Dari hasil *log* tersebut maka sms yang dikirimkan berisi *service telnet, ftp, ssh* dan *apache* yang diakses oleh *host* 192.168.1.3. Jadi, kecenderungan dari teknik *scanning* adalah dalam akan melakukan identifikasi untuk mengetahui *host* dan *port* mana saja yang terbuka dalam jaringan. Sehingga dalam *log file honeyd* dapat ditandai dengan diaksesnya semua *port* yang terbuka dalam waktu yang hampir bersamaan.

## 2. Log file Teknik Brute force

*Log file* yang merekam aktifitas *scanning* akan memuat informasi pengaksesan terhadap satu *service* yang diserang. Pada pengujian ini serangan dilakukan pada *service telnet* yang diketahui dari *host* dengan *ip address* 192.168.1.6 pada proses *scanning*.

```

2012-02-27-01:58:31.6574 honeyd log started -----
2012-02-27-02:05:38.2146 tcp(6) S 192.168.1.3 29901 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2165 tcp(6) S 192.168.1.3 29902 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2167 tcp(6) S 192.168.1.3 29903 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2167 tcp(6) S 192.168.1.3 29904 192.168.1.6 23
[Windows 2000 RFC1323]
2012-02-27-02:05:38.2169 tcp(6) S 192.168.1.3 29905 192.168.1.6 23
[Windows 2000 RFC1323]
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
,,,,
2012-02-27-02:05:38.2609 tcp(6) E 192.168.1.3 29887 192.168.1.6 23: 0 79
2012-02-27-02:05:38.2610 tcp(6) - 192.168.1.3 29887 192.168.1.6 23: 40 R
[Windows 2000 RFC1323]
--ENDMARK--
--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",
"--MARK--, "Mon Feb 27 02:05:38 WIT 2012", "telnet", "", "", "",

```



### 3. Log file Teknik OS Fingerprinting

Pada *log file* yang dihasilkan dari serangan *os fingerprinting*, sekilas sangat mirip dengan pengaksesan biasa seperti melakukan *ping* pada target. Namun, jika diamati lebih lanjut terdapat layanan *protocol* yang diakses yaitu TCP, UDP dan ICMP. Berikut ini adalah penjelasan dari *log file* di bawah ini.

```

2012-03-04-02:45:51.6715 tcp(6) S 192.168.1.5 43824 192.168.1.10 65535
2012-03-04-02:45:51.6716 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6717 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6717 tcp(6) E 192.168.1.5 43824 192.168.1.10 65535: 0 0
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6720 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6721 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6721 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6722 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6722 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6722 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6723 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6724 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.6724 tcp(6) - 192.168.1.5 43824 192.168.1.10 65535: 40 R
2012-03-04-02:45:51.7991 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7992 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7993 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:51.7994 icmp(1) - 192.168.1.5 192.168.1.10: 17(0): 32
2012-03-04-02:45:54.5114 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:54.5114 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:54.5116 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:54.5117 icmp(1) - 192.168.1.5 192.168.1.10: 15(0): 28
2012-03-04-02:45:57.5115 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5115 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5117 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:45:57.5117 icmp(1) - 192.168.1.5 192.168.1.10: 13(0): 40
2012-03-04-02:46:00.4313 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4313 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4315 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:00.4316 udp(17) S 192.168.1.5 47122 192.168.1.10 161
2012-03-04-02:46:49.2796 udp(17) E 192.168.1.5 53 192.168.1.10 65534: 76 0

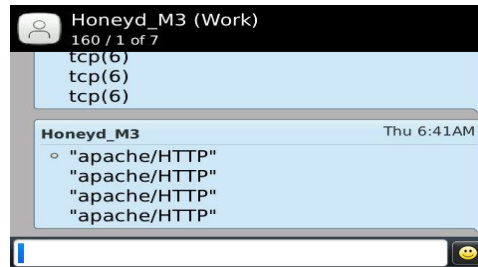
```

Sebenarnya banyak sekali baris *log file* yang merekam serangan *os fingerprinting*. Hal yang menarik dari *log file* diatas adalah, sekilas hanya terjadi proses koneksi biasa seperti dilakukan *ping*. Namun yang menjadi tanda disini adalah *host* dengan *ip address* 192.168.1.5 mengirimkan paket kepada *host* 192.168.1.10 untuk mengakses semua layanan *protocol* TCP, UDP dan ICMP dalam satu waktu. Sehingga dapat diketahui dari pola yang tercatat pada *log file* diatas adalah serangan dengan tipe *os fingerprinting*.

#### 4.4.4 Pengiriman SMS

Merujuk pada BAB III, telah dijelaskan secara detail mengenai konfigurasi program yang digunakan. Pengiriman *sms* yang berisi informasi paket yang tidak terotorisasi dilakukan dengan mengkonfigurasi *honeyd* dengan *gammu sms gateway*. Paket tersebut dapat diketahui pada *log file* yang merekam segala aktifitas yang terhubung dengan *honeyd*. Pada *log file* terdapat beberapa informasi yang termuat, waktu koneksi, *port*, *protocol*, *ip* asal, *port* asal, *ip* tujuan dan *port* tujuan, serta sistem operasi yang digunakan. Namun, tidak semuanya dari satu baris *log file* akan dikirimkan. Hal tersebut dikarenakan jumlah karakter pada *sms* yang hanya berjumlah 140 karakter. Jika satu baris dikirimkan semua maka akan menghabiskan semua karakter *sms* serta tidak memuat informasi penting dari serangan yang dilakukan.

Untuk mengantisipasi hal tersebut maka diperlukan sebuah *filtering* pada baris *log file honeyd*. Teknik yang digunakan untuk melakukan *filtering* adalah *regular expression*. *Filtering* yang dilakukan hanya akan diambil *field protocol* dan *service*. Pengambilan *Field* tersebut dikarenakan kedua *field* tersebut yang menandakan *protocol* apa yang *intruder* gunakan dan *service* apa yang diakses. Dalam program yang dibuat, pemfilteran dilakukan dengan dua perintah, perintah pertama digunakan untuk memfilter *protocol* yang digunakan, sedangkan perintah kedua digunakan untuk memfilter *service* yang diakses yang terdapat pada baris MARK dan ENDMARK. Jadi, akan terkirim dua buah *sms* yang menginformasikan *protocol* yang diakses dan *service* yang diakses.



**Gambar 4.23** Sms yang menginformasikan *protocol* dan *service* yang diakses.

Pada konfigurasi program waktu pengiriman dilakukan setiap 2 menit. Jika dalam waktu 2 menit terdapat selisih jumlah baris *log file* maka selisih tersebut yang akan dikirimkan. Namun, jika dalam waktu 2 menit tidak ada perbedaan jumlah baris *log file* maka program tidak akan mengeksekusi apapun.

Semua hasil pengujian yang dilakukan dengan menggunakan tiga jenis teknik dan *tools* yang berbeda, maka dapat diketahui pola-pola serangan yang terekam pada *log file honeyd*. Secara mendasar, pola pada *log file* dipengaruhi oleh teknik yang digunakan dan *service* yang coba diserang. Dari pola-pola tersebut dapat mempermudah dalam melakukan analisa terhadap sebuah koneksi/serangan yang dilakukan oleh *intruder*. Selain itu, *sms* yang menginformasikan adanya koneksi yang ilegal juga sangat membantu dalam proses analisa. Dari *sms* yang terkirim dapat diketahui *service* apa yang diakses. Jadi, pengkonfigurasi *honeyd* dengan *sms gateway* sangat membantu dalam pendeteksian paket yang tidak terotorisasi secara cepat.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan hasil yang diperoleh pada penelitian *honeyd* dan *sms gateway* maka dapat diambil kesimpulan :

1. Hasil yang diperoleh, *honeyd* mampu merekam serangan dengan teknik *scanning*, *brute force* dan *os fingerprinting* yang tercatat dalam *log file* dan *log file* yang dihasilkan dapat memberikan kemudahan dalam proses analisa serta mampu mengirimkan *sms* yang menginformasikan *service* yang sedang diakses oleh *intruder*.
2. Konfigurasi *honeyd* dengan *sms gateway* mampu menghasilkan sebuah sistem yang menandakan adanya paket tidak terotorisasi yang mencoba melakukan penetrasi terhadap jaringan komputer.
3. Kelebihan dari konfigurasi *honeyd* dengan *sms gateway* adalah informasi adanya paket yang tidak terotorisasi dapat diketahui dengan cepat melalui *sms* yang manginformasikan adanya paket tersebut serta dapat langsung mengetahui *service* apa yang sedang diakses oleh *intruder*.
4. Kekurangan dari konfigurasi *honeyd* dengan *sms gateway* adalah notifikasi *sms* yang dikirimkan belum berupa informasi terolah yang menerangkan adanya aktifitas tidak terotorisasi secara spesifik dalam jaringan komputer dan masih berupa cuplikan dari *log file honeyd*.

#### **5.2 Saran**

Terdapat beberapa hal yang perlu diperhatikan dalam pengimplementasian *honeypot* untuk menghasilkan sistem keamanan tangguh. Supaya dapat terbentuk sistem keamanan yang tangguh ada beberapa saran yang perlu diperhatikan :

1. Pada penelitian *honeypot* selanjutnya alangkah baiknya di konfigurasikan dengan *firewall* dan IDS, dikarenakan pengkonfigurasian tersebut mampu menciptakan sistem keamanan yang lebih baik.

2. Akan lebih baik jika dalam pengimplementasian *honeypot* diterapkan pada jaringan perusahaan atau pada tempat umum yang memiliki koneksi internet agar dapat mengetahui informasi dari tindakan penyerangan sesungguhnya.
3. Melakukan analisa terhadap *log file honeypot* secara mendetail untuk mempelajari dan mengidentifikasi jenis serangan yang dilakukan pada jaringan komputer





## DAFTAR PUSTAKA

- [AND09] Andreas, 2009. Regex tidak susah kok (online) <http://sl4y3r.blog.com/2009/>.
- [BAR01] Barnett, Bruce, 2001. What is regular expression (online) <http://grymoire.com/Unix>.
- [CIS11] Cisco Networking Academy, 2011. Network Fundamental (online) <http://www.cisco.com/web/learning>.
- [DIO12] Dion, Andrew, 2012. Membuat sms server dengan gammu, kalkun dan linux ubuntu (online) <http://andrewdion.staff.ugm.ac.id>.
- [ELR09] Erlangga, 2009. Konsep Honeypot (online) <http://rangga07.wordpress.com/2009>.
- [GAM12] Gammu, 2012. Gammu Phone Database (online) <http://wammu.eu>.
- [HER11] Hernawan, Budi, 2011. Keunggulan honeypot (online) <http://www.budihermawan.net>.
- [KUR11] Kurniawan, Zuhdi, 2011. Kejahatan Menggunakan Internet di Indonesia (online) <http://blog.ub.ac.id/kurniawanzuhdi>.
- [PRA12] Prasetyo, Yudi, 2012. Macam Serangan terhadap Jaringan Komputer (online) <http://yudi-prasetyo.blogspot.com>.
- [PRO04] Provos, Niels, 2004. Developments of the Honeyd Virtual Honeypot (online) <http://honeyd.org>.
- [RWT05] RWTH Aachen University. 2005. Hand on Honeypot Technology. German. RWTH Aachen University.
- [SID04] Sidik, Betha. 2004. UNIX dan LINUX : Panduan Bekerja dalam Lingkungan Unix dan Linux. Bandung: Informatika Bandung.
- [SPI01] Spitzner, Lance, 2001. The Value of Honeypots, Parts One: Deffinition and Value of Honeypots (online) <http://symantec.com/connect/articles>.

- [SPI03] Spitzner, Lance, 2003. Definitions and Value of Honeypots (online) <http://www.tracking-hackers.com>.
- [UTD05] Utdirartatmo, Firrar. 2005. Trik Menjebak Hacker dengan Honeypot. Yogyakarta: Andi Yogyakarta.
- [WAK11] Wakhyuni, Aprilia, 2011. Modus Kejahatan Teknologi Informasi (online) <http://apriawakhyuni.blogspot.com>.



## LAMPIRAN



### SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Cahya Adi Sn

No. Mhs. : 07 523 154

Judul TA : \_\_\_\_\_

- > ditambahkan ~~seta~~ jenis serangan yang akan digunakan.
- > di lanjutkan lagi tugas akhirnya.



Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 12 Juli 2011

Dosen,

Balang  
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran



SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Cahya Ali SU

No. Mhs. : 07523154

Judul TA : \_\_\_\_\_

\* Batasi masalah di :

- Virtual server platform yg digunakan
- serangan yg digunakan
- aplikasi honey pot yg dipakai



Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 17/7/11.....

Dosen,

Dhamas Hattar F.  
(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran