IMPLEMENTASI LINUX EMBEDDED SYSTEM UNTUK INTRUSION DETECTION SYSTEM MENGGUNAKAN OPENWRT PADA WIRELESS ROUTER WRT54G

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat

Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika

Oleh :

Nama : Arif Firmawan

NIM : 02 523 265

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

2007

LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI LINUX EMBEDDED SYSTEM UNTUK INTRUSION DETECTION SYSTEM MENGGUNAKAN OPENWRT PADA WIRELESS ROUTER WRT54G

TUGAS AKHIR

Oleh:

Nama : Arif Firmawan

NIM : 02 523 265

Yogyakarta, 12 April 2007

Pembimbing

Fathul Wahid, ST., M.Sc.

LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI LINUX EMBEDDED SYSTEM UNTUK INTRUSION DETECTION SYSTEM MENGGUNAKAN OPENWRT PADA WIRELESS ROUTER WRT54G

TUGAS AKHIR

Oleh :

Nama : Arif Firmawan

NIM : 02 523 265

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

Yogyakarta, 3 Mei 2006

Tim Penguji

Fathul Wahid, ST., M.Sc ----------
Ketua

Yudi Prayudi, S.Si., M.Kom ----------
Anggota I

Hendrik, ST ----------
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom

HALAMAN PERSEMBAHAN

Untuk kedua orangtuaku yang tiap tetes keringat dan doanya tak kan pernah dapat tergantikan dengan sesuatu apapun

Untuk uni dan adikku yang selalu menjadi semangatku

Untuk mereka yang dengan sungguh-sungguh menulisi kertas putih dihati dan akalku

MOTTO

(Yaitu) orang-orang yang mengingat ALLAH sambil berdiri atau duduk atau dalam keadaan berbaring dan mereka memikirkan tentang penciptaan langit dan bumi (seraya berkata):"Ya Tuhan Kami, Tiadalah Engkau menciptakan ini dengan sia-sia, Maha suci Engkau, Maka peliharalah kami dari siksa neraka

(QS. Ali-Imron:191)

Allahumma, Rabb Jibril, Mikail, Israfil. Yang menghamparkan langit serta bumi. Mengetahui yang ghaib dan yang terang Engkaulah yang memutuskan hukum diantara hamba-hambaMU terhadap apa yang mereka perselisihkan. Dengan IzinMU, tunjukkanlah kepadaku, dalam perselisihan itu. Sesungguhnya Engkaulah Yang Memberi petunjuk kepada siapa saja yang Engkau kehendaki

(HR MUSLIM, No 770, I/534)

Tidak ada yang penting didunia ini, kecuali berlomba lari untuk melihat telapak kaki siapa yang lebih dahulu sampai dihalaman rumahMu.

(Emha Ainun Nadjib)

Hidup adalah sebuah perjalanan, bukan sebuah perjudian. Kita tidak bertaruh untuk hidup. Mencari kemenangan sesaat dan kerugian berkepanjangan. Pilihan-pilihan yang ada adalah kemestian yang harus dipertimbangkan secara benar. Kehidupan juga bukanlah sebuah pengorbanan, karena itu sikap pesimis yang tidak pantas. Berapapun harga yang telah kita bayar itu adalah sebuah keputusan baik yang telah kita ambil. Gagal ataupun berhasil. Kita tidak pernah tau apakah keputusan itu baik dan benar atau jelek dan salah. Karen kita takkan pernah tau apa itu masa depan. Kita hanya mempelajari, memahami dengan semua derivasinya. Kebenaran akan tersingkap ketika kita sudah tidak lagi berurusan dengan lingkungan kecil ini. Meyakini ada sebuah rahasia besar dibalik semua hasil yang kita capai, merupakan kepantasan yang tidak perlu diragukan.

(Penulis)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamu'alaikum Wr. Wb.

Segala puji bagi ALLAH SWT, zat yang mengalirkan ilmu dalam aliran besar yang tak terhingga, zat yang ilmu dan nikmatnya takkan pernah dapat dituliskan meskipun air laut dijadikan tinta dan pohon-pohon dijadikan penannya. Atas rahmatNya untuk mencicipi setetes dari lautan ilmuNya yang tak terbatas sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul "Implementasi Linux Embeddded system untuk Intrusion Detection System menggunakan OpenWRT pada Wireless Router WRT54G" . Shalawat serta salam semoga senantiasa tercurah kepada Rasulullah Muhammad SAW, para sahabat dan pengikutnya yang istiqomah dijalannya.

Tugas Akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika.

Penulis dalam kesempatan ini mengucapkan terima kasih kepada :

1. Kedua Orang Tua.
2. Bapak Fathul Wahid, ST, MSc selaku Dekan Fakultas Teknologi Industri dan Dosen pembimbing.
3. Bapak Yudi Prayudi, S.Si, M.Kom selaku Ketua Jurusan Teknik Informatika

4. Seluruh staf pengajar jurusan teknik Informatika yang telah membekali ilmu selama penulis menjalani studi.
5. Fakultas Teknologi Industri UII yang memberikan kesempatan penulis untuk menjadi administrator Network Operation Center.
6. Andrey dan Yuda atas bantuan dan ilmunya selama ini.
7. Jarwo dan Andi yang sudah meminjamkan laptopnya untuk kepentigan tugas akhir penulis.
8. Forum dan teman-teman diskusi yang memberikan begitu banyak ilmu untuk kembali berpikir dan merenung atas proses yang sedang dijalani disela-sela kesibukan penulis mengerjakan tugas akhir.
9. Saudara-saudara asisten dan mantan asisten Laboratorium Sistem dan Jaringan Komputer (Yuda, AP, Ryan, Nanda, Tetra, Dinie, Iwell, Jarwo, Urip, Al, Andi, Heri, Ninki, Udoh) yang telah memcatatkan dirinya dalam proses hidup penulis.
10. Mas Andan dan teman-teman Laboratorium Informatika Terpadu.
11. Teman-teman Expertindo Dasa Pratama, yang pertanyaannya sekarang mimpi-mimpinya bagaimana ya.
12. Teman-teman VOIP02, hari jumat hari confernce sepertinya perlu direvisi.

Dalam penyelesaian Tugas Akhir ini penulis menyadari bahwa masih banyak terdapat kesalahan dan kekurangannya, untuk itu penulis mengharapkan kritik dan saran yang membangun agar bisa berguna untuk masa mendatang.

Semoga Tugas Akhir ini dapat menjadi referensi bagi administrator sistem dalam menerapkan keamanan jaringannya dan bagi pembaca dapat mengambil pelajaran serta menambah pengetahuan tentang Teknologi Informasi khususnya dibidang sistem keamanan. Amien.

Wassalaamu'alaikum Wr. Wb.

Yogyakarta, 11 April 2007

Penyusun

SARI

Salah satu tren terkini dibidang teknologi dan jaringan komputer adalah konsep dan implementasi jaringan berbasis wireless. Meningkatnya penggunaan teknologi ini diiringi dengan diciptakan dan dikembangkannya piranti wireless yang semakin baik. Salah satu piranti wireless yang banyak digunakan ialah wireless broadband router linksys WRT54G. Alat ini disebut wireless router karena didesain untuk mengerjakan proses routing antara piranti wireless dan koneksi internet. WRT54G dilengkapi dengan firmware statis yang dirasa kurang jika dibandingkan dengan pengetahuan pengguna wireless yang semakin meningkat. Untuk meningkatkan kemampuan WRT54G digunakan OpenWRT untuk menggantikan firmware statis tersebut. OpenWRT datang dengan bebagai fitur dan aplikasi yang memudahkan administrator untuk melakukan kontrol penuh terhadap jaringannya.

Isu keamanan yang menjadi pokok bahasan dalam penelitian ini dimulai dengan penerapan vlan untuk memisahkan secara logik jaringan-jaringan yang terhubung dengan WRT54G. Pemisahan ini menjadi penting karena jaringan konvensional (LAN) tidak dapat berhubungan secara langsung dengan WLAN, sehingga data-data pada WLAN tidak dapat diakses secara langsung dari LAN dan sebaliknya. Selanjutnya penerapan SSID sebagai identitas jaringan WLAN akan di sembunyikan yang dilengkapi dengan kunci enkripsi yang harus diketahui user yang berhak. SSID yang disembuyikan dimaksudkan untuk menghindari penggunaan sumber daya oleh intruder yang tidak diinginkan, kunci enkripsi sebagai salah satu otentikasi user menggunakan WPA2-PSK menggantikan kunci enkripsi standar WEP yang sangat mudah di dekripsi. Untuk memperkuat keamanan WLAN digunakan IDS yang akan memonitor dan melaporkan aktifitas mencurigakan yang dilakukan user sesuai dengan konfigurasi dan aturan yang telah didefinisikan.

Penelitian dilakukan dengan menganalisis kebutuhan sistem melalui pengumpulan data dan observasi untuk mendapatkan informasi yang akan diimplementasikan. Informasi tersebut digunakan untuk perancangan sistem menggunakan metode berarah aliran data, sehingga dihasilkan sistem yang strukturnya dapat didefinikan dengan jelas. Informasi ini juga digunakan untuk merancang arsitektur jaringan sebagai lingkungan tempat sistem diimplementasikan.

Dari penelitian yang dilakukan, dihasilkan sistem yang dapat digunakan sebagai salah satu alternatif penerapan kemanan jaringan berbasis wireless menggunakan WRT54G dan OpenWRT. Hasil pengujian membuktikan bahwa sistem dapat membantu administrator untuk mengetahui aktifitas yang terjadi dalam jaringannya sehingga administrator dapat mengambil keputusan yang tepat dari informasi yang dimilikinya.

Kata Kunci : vlan, SSID, WPA2-PSK, IDS

TAKARIR

perimeter	*garis pertahanan*
scanning	*pengintaian*
broadcast	*menyiarkan*
policy	*kebijakan*
intruder	*penyusup*
leased time	*waktu kontrak*
interface	*antarmuka*
key	*kunci*
layer	*lapisan*
remote	*jarak jauh*
request	*permintaan*
service	*layanan*

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

WRT54G adalah sebuah *hardware Wireless Broadband Router* yang dikeluarkan oleh Linksys berbasis frekuensi 2.4 GHz. Alat ini menggunakan *static firmware* sebagai sistem operasi yang telah tertanam dalam RAM dan flash memory. Fungsi standar dari firmware yang statis ini dirasa kurang jika dibandingkan dengan peningkatan penggunaan dan pengetahuan pengguna wifi yang semakain meningkat, hal tersebut berhubungan dengan administrasi dan keamanan pada jaringan berbasis wifi.

Untuk itu firmware dinamis yang menyediakan kemampuan untuk menulis file sistem, kemudahan manajemen paket dan peningkatan keamanan jaringan *wireless* layak digunakan. Implementasi *Linux Embedded System* diharapkan dapat menyelesaikan permasalahan tersebut.

Alih-alih menjejalkan semua fitur yang mungkin kedalam router, OpenWRT menyediakan fungsi untuk menambah atau mengurangi paket-paket sesuai kebutuhan. Keadaan ini memungkinkan penyesuain fitur-fitur yang layak dipakai dan menghilangkan paket-paket yang tidak dibutuhkan agar tersedia ruang bagi paket yang lain.

Penggunaan OpenWRT akan memaksimalkan kerja WRT54G yang dalam lapisan tertentu OSI layer dapat disandingkan dengan fungsionalitas router dan switch yang dikeluarkan oleh vendor terkenal seperti Cisco. Pada akhirnya kita akan mendapatkan router dengan biaya murah dengan kemampuan yang tinggi.

1.2 Rumusan Masalah

Bagaimana Mengimplementasikan *Intrusion Detection System* pada WRT54G yang telah menggunakan OpenWRT untuk membantu administrator mengetahui aktifitas yang mencurigakan dengan memberikan informasi berupa *alert* dan meningkatkan keamanan jaringan berbasis *wireless*.

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam penelitian tugas akhir ini adalah sebagai berikut :

a. Implementasi dilakukan pada WRT54G versi 2 menggunakan OpenWRT.

b. Semua konfigurasi sistem dilakukan berbasis teks.

c. Sistem dibangun dengan memperhitungkan penggunaan WRT54G untuk keperluan *Wireless Local Area network* (WLAN) yang berhubungan dengan *Local Area Network* (LAN).

c. Implementasi

Beberapa tahap dilakukan untuk mengimplementasikan desain yang pada fase sebelumnya telah didefinisikan, antara lain :

1. Membuat rancangan jaringan nirkabel.
2. Melakukan transfer OpenWRT kedalam WRT54G.
3. Mengkompilasi ulang paket-paket yang dibutuhkan dengan menyesuaikannya dengan prosesor dan kapasitas RAM yang tersedia didalam WRT54G.
4. Membangun Firewall dan IDS yang akan dijalankan
5. Mengkonfigurasi sistem secara keseluruhan.

d. Pengujian

Tahap ini dilakukan setelah selesai pembuatan/pengkodean sistem, untuk melihat sejauh mana aplikasi yang telah dibuat sesuai dengan yang dikehendaki. Tahap ini juga dilakukan uji coba dengan melakukan *scanning* terhadap sistem yang ada sebagai bagian dari pola-pola serangan jaringan komputer.

1.7 Sistematika Penelitian

Dalam sistematika penelitian tugas akhir ini diberikan uraian bab demi bab untuk mempermudah pemahaman. Penulisan laporan tugas akhir ini, disusun dalam sistematika yang terbagi menjadi lima bab.

Bab I berisi Pendahuluan yang memuat latar belakang yang menyebabkan munculnya permasalahan, asumsi dan batasan yang digunakan, tujuan yang akan dicapai, manfaat penelitian, dan bagaiman penelitian dilaksanakan.

Bab II berisi Landasan Teori yang memuat teori-teori dasar yang berhubungan dengan penelitian, berupa teori mengenai TCP/IP, jaringan nirkabel, keamanan jaringan, firewall dan IDS.

Bab III berisi Metodologi yang memuat penjelasan tahap analisis dan perancangan sistem yang akan dibangun, langkah-langkah dan hasilnya dengan menggunakan metode yang dipilih.

Bab IV berisi Hasil dan Pembahasan yang memuat batasan implementasi sistem yaitu asumsi-asumsi yang dipakai, lingkungan pengembangan, paket yang dipakai beserta alasan pemilihan. Bab ini juga memuat dokumentasi hasil pengkajian terhadap sistem yang dibandingkan kebenaran dan kesesuaiannya dengan kebutuhan sistem yang dituliskan sebelumnya.

Bab V berisi Simpulan dan Saran yang memuat kesimpulan-kesimpulan dari proses pengembangan sistem setelah dilakukan implementasi dan pengujian. Juga berisi saran yang perlu diperhatikan berdasar keterbatasan-keterbatasan yang ditemukan dan asumsi-asumsi yang dibuat selama melakukan penelitian.

BAB II

LANDASAN TEORI

2.1 Sistem Embedded

Sistem *embedded* adalah sistem komputer yang memiliki tujuan khusus, yang secara sempurna dibungkus dengan piranti untuk mengontrolnya. Sistem *embedded* berbeda dengan komputer untuk tujuan umum berdasarkan pemrogramannya untuk mengerjakan tugas-tugas tertentu. Berbagai macam tipe sistem *embedded* secara luas digunakan pada teknologi jaringan, sebagian besar diimplementasikan pada switch dan router. Selain pemrograman yang khusus, sistem *embedded* biasanya memilik karakteristik, antara lain :

a. Penggunaan daya yang rendah
b. Ukuran yang kecil dan terhitung murah
c. Penggunaan fan (pendingin) yang terbatas
d. Memiliki CPU dan memori yang terbatas

Bentuk yang kecil dengan penyuplai daya yang kecil membatasi jumlah komponen dan kecepatan CPU. Daya yang kecil berarti juga sedikit chip memori yang dapat digunakan. Seringkali, sistem *embedded* didesain menggunakan Compact Flash (CF) atau tipe yang sama dari media penyimpanan non volatile dibanding menggunakan hard drive. [HAS07]

Linux Embedded System adalah sistem operasi berbasis kernel Linux yang digunakan untuk sistem komputer embedded. Distribusi *embedded* Linux biasanya terdiri dari sebuah *development framework* dan berbagai aplikasi yang dirangkai (*software application tailored*) untuk sistem embedded, atau keduanya.

Development framework Distribution terdiri dari perkakas pengembangan yang memfasilitasi pengembangan sistem *embedded*, seperti *browser* untuk melihat sumber aplikasi tertentu, *cross-compilers*, *debugger*, *software project management*, *boot image builders* dan sebagainya. *Tailored Embedded Distribution* menyediakan sekumpulan aplikasi yang digunakan dalam sistem *embedded* tujuan, seperti library khusus, execu, dan file konfigurasi. [YAG03]

Salah satu *Linux Embedded System* adalah OpenWRT. OpenWRT merupakan distribusi Linux untuk piranti *embedded*. Alih-alih menggunakan sebuah *firmware* statis, OpenWRT menyediakan kemampuan untuk menulisi file system secara penuh dengan manajemen paket. Hal ini menyebabkan administrator bebas untuk memilih aplikasi dan konfigurasi yang disediakan oleh sebuah *vendor* (penyedia aplikasi) dan mengijinkan administrator untuk menyesuaikan paket untuk memenuhi keutuhan aplikasinya. [BAK07]

2.2 TCP/IP

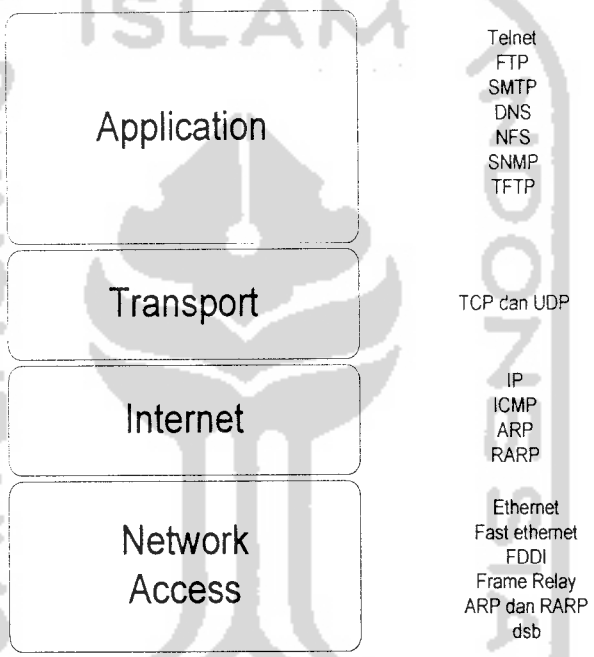
Untuk dapat saling berkomunikasi, komputer harus menggunakan suatu aturan yang dimengerti oleh semua komputer yang berhubungan dengannya. Aturan itu

disebut dengan protokol. TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di internet. Komputer-komputer yang terhubung ke internet berkomunikasi dengan protokol TCP/IP, karena menggunakan bahasa yang sama perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. TCP/IP banyak digunakan seiring dengan pertumbuhan yang cepat didunia internet dan dan makin meluasnya penggunaan intranet. Penggunaan yang luas tersebut antara lain dikarenakan fitur-fitur yang dimiliki oleh TCP/IP, antara lain : [PUR98]

1. Standar protokol yang terbuka, tersedia secara bebas dan dapat dikembangkan secara bebas tanpa mempedulikan *hardware* komputer dan sistem operasi yang dipakai. Karena dukungan yang begitu besar, TCP/IP sangat ideal untuk menyatukan berbagai *hardware* dan *software*.
2. Tidak tergantung oleh *hardware* fisik manapun. Hal ini membuat TCP/IP dapat mengintegrasikan berbagai macam jaringan. TCP/IP dapat berjalan pada *Ethernet,token ring, dial-up line* dan berbagai media transmisi fisik lainnya.
3. Skema pengalamatan yang membuat tiap *node* TCP/IP memiliki alamat unik yang berbeda dengan *node* lain dalam satu jaringan.
4. Standar protokol yang konsisten, secara luas dapat dipergunakan oleh berbagai *user*.

2.2.1 Model Layer (lapisan) TCP/IP

Protokol-protokol jaringan pada umumnya dikembangkan dengan layer-layer (*layers*), masing-masing layer bertanggung jawab terhadap masalah komunikasi yang berbeda. Sekumpulan protokol, dalam hal ini protokol TCP/IP, adalah kombinasi protokol-protokol yang berbeda pada berbagai layer. Model Layer TCP/IP tediri dari sistem 4 layer, seperti ditunjukkan pada Gambar 2.1. [ANO01]



Gambar 2.1. Model Layer TCP/IP

Masing-masing layer tersebut memiliki tanggungjawab yang berbeda, yaitu :

1. *Application Layer*, layer ini menangani protokol-protokol tingkat tinggi seperti Telnet untuk remote login, FTP untuk transfer file, SMTP untuk surat elektronik,

SNMP untuk mengontrol dan memonitor piranti jaringan, DNS untuk mentranslasikan alamat IP ke sebuah *domain* dan sebaliknya. [ANO01], [STE96]

2. *Transport Layer*, layer ini menyediakan aliran data antara *host* asal dan *host* tujuan untuk layer aplikasi diatasnya. Terdapat dua protokol penting dalam layer ini, yaitu :
   a. *Transmission Control Protocol* (TCP), merupakan protokol *reliable connection-oriented* yang mengijinkan aliran byte yang berasal dari suatu *host* untuk dikirimkan tanpa *error* ke host lain. TCP memecah aliran byte data menjadi pesan-pesan diskret dan meneruskannya ke internet layer. Pada host tujuan, proses TCP tujuan penerima merakit kembali pesan-pesan yang diterimanya menjadi aliran output. TCP juga menangani pengendalian aliran untuk memastikan bahwa pengirim yang cepat tidak akan membanjiri pesan-pesan yang akan diterima penerima yang lambat.
   b. *User Datagram Protocol* (UDP), merupakan protokol yang tidak reliable dan *connectionless*. UDP digunakan secara meluas pada query dan aplikasi *client/server* jenis *request-reply*, dimana pengiriman yang cepat lebih diutamakan dibanding dengan pengiriman yang akurat. [STE96]
3. *Internet Layer*, layer ini bertanggung jawab untuk memilih jalur mana yang akan digunakan dalam pengiriman paket dalam jaringan. Protokol utama dalam layer ini adalah *Internet Protocol* (IP). IP melakukan operasi seperti pendefinisian paket dan skema pengalamatan, transfer data antara *internet layer* dan *network access layer* dan merouting paket ke *host* tujuan. [ANO01]

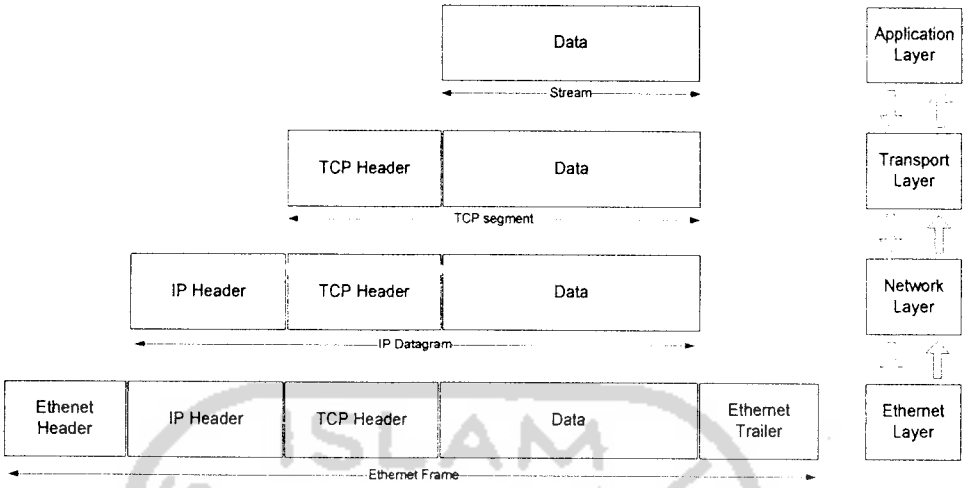
4. *Network Access Layer*, layer ini bertanggung jawab membuat jalur fisik ke media jaringan, termasuk hubungan *device driver* pada sistem operasi dengan kartu jaringan pada komputer. [STE96]

2.2.2 Enkapsulasi

Ketika suatu aplikasi mengirimkan data menggunakan TCP, data akan dikirimkan ke layer dibawahnya sampai data dikirimkan dalam bentuk sekumpulan bit melewati jaringan hingga sampai ke komputer tujuan. Pada setiap tahapan tersebut, masing-masing layer akan menambahkan informasi berupa *header* (seringkali juga informasi *trailer*) pada data yang diterimanya. Penambahan informasi agar pengiriman data berjalan baik disebut dengan enkapsulasi (*encapsulation)*.

Dalam TCP/IP, terjadi penyampain data dari protokol yang berada disatu layer ke protokol yang berada dilayer yang lain. Setiap protokol memperlakukan semua informasi yang diterimanya dari protokol lain sebagai data. Jika suatu protokol menerima data dari protokol lain dilayer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut. Informasi ini memiliki fungsi protokol tersebut. Setelah itu, data ini diteruskan lagi ke protokol pada layer dibawahnya. Hal yang sebaliknya terjadi jika suatu protokol lain yang berada pada layer dibawahnya. Jika data ini dianggap valid, protokol akan melepas informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang berada pada layer diatasnya. : [STE96], [ANO01]

Gambar 2.2 menunjukkan proses enkapsulasi dalam layer TCP/IP.



Gambar 2.2. Proses enkapsulasi data pada tiap layer

2.3 Wireless Local Area Network (WLAN)

Wireless Local Area Network (WLAN) definisikan sebagai sebuah sistem komunikasi data fleksibel yang dapat di gunakan untuk menggantikan atau menambah jaringan LAN yang sudah ada untuk memberikan tambahan fungsi dalam konsep jaringan komputer pada umumnya. Fungsi yang ditawarkan disini dapat berupa konektivitas yang andal sehubungan dengan mobilitas *user*. WLAN menggunakan gelombang radio sebagai media untuk berkomunikasi antar piranti. [GUN06]

Teknologi WLAN didefinisikan oleh *Institute of Electrical and Electronical Engineers* (IEEE) dengan standar 802.11. Terdapat beberapa spesifikasi pada standar 802.11 :

1. 802.11, menyediakan kecepatan transfer data pada 1 atau 2 Mbps pada frekuensi 2.4 Ghz.
2. 802.11a, pengembangan dari spesifikasi sebelumnya yang menyediakan kecepatan transfer data sampai 54 Mbps dan bekerja pada frekuensi 5 Ghz.
3. 802.11b, menyediakan kecepaan transfer data sampai 11 Mbps dan bekerja pada frekuensi 2.4 Ghz.
4. 802.11g, menyediakan kecepatan transfer data antara 20-54 Mbps dan bekerja pada frekusensi 2.4 GHz. [JAV07]

Terdapat beberapa keuntungan yang didapat dari penggunaan WLAN, diantaranya :

1. Mobilitas tinggi

WLAN memungkinkan *client* untuk mengkases informasi secara real time sepanjang masih dalam jangakauan WLAN, sehingga meningkatkan kualitas layanan dan produktifitas yang tidak mungkin dapat diberikan oleh jaringan LAN biasa.

2. Kemudahan dan kecepatan instalasi

Pengguanaan radio sebagai media trasnmisi menyebabkan instalasi perangkat WLAN menjadi lebih mudah dan cepat tanpa harus menarik dan memasang kabel pada tiap-tiap piranti *wireless*. Kabel digunakan hanya digunakan untuk menghubungkan *Access Point (AP)* ke jaringan (HUB/Switch/Router).

3. Fleksibel

Dengan teknologi WLAN sangat memungkinkan untuk membangun jaringan pada area yang tidak mungkin atau sulit bila dijangkau oleh kabel. WLAN juga

memungkinkan penambahan kapasitas *client* secara tiba-tiba dengan alat yang sudah tersedia.

4. Menurunkan biaya kepemilikan

Meskipun biaya investasi awal untuk perangkat keras WLAN lebih mahal dibandingkan dengan LAN konvensional, namun biaya instalasi dan perawatan jaringan lebih murah. Disamping itu, sangat cocok untuk lingkungan dinamis, dimana sering terjadi perpindahan, penambahan atau perubahan posisi kerja.

5. Skalabel

WLAN dapat digunakan dengan berbagai topologi jaringan sesuai kebutuhan instalasi atau spesifikasi. Mulai dari jaringan independen yang hanya terdiri atas beberapa *client* saja, sampai jaringan infrastruktur yang terdiri dari banyak *client*. Proses implementasi WLAN dapat dilakukan secara bertahap sesuai dengan kebutuhan.

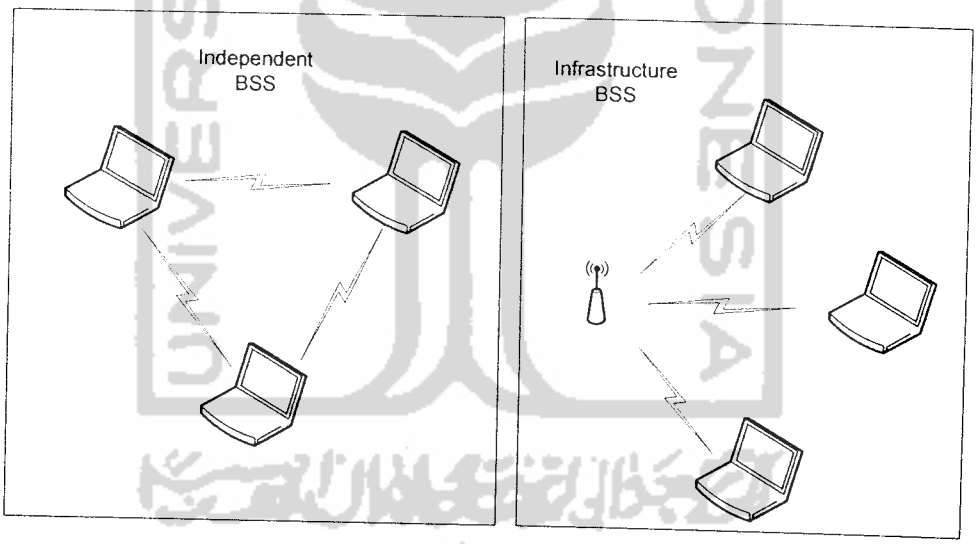
6. Produktivitas

Pengguna yang terhubung dengan WLAN dapat terus terkoneksi dalam jaringan apabila mereka bergerak dari satu tempat ke tempat lain. Bagi sektor bisnis akan berdampak pada peningkatan produktivitas karyawan yang dapat menyelesaikan pekerjaannya pada berbagai tempat. [GUN06]

2.3.1 Tipe Jaringan

Tipe jaringan adalah salah satu faktor pendukung suksesnya jaringan WLAN. Dengan menggunakan tipe jaringan yang tepat, maka akan diperoleh stabilitas dan kinerja yang terbaik.

Bentuk dasar dari jaringan berbasis 802.11 adalah *Basic Service Set (BSS)*, didalamnya terdapat sekelompok stasiun yang berkomunikasi antar sesamanya. Komunikasi dilakukan dalam sebuah area *fuzzy*, disebut dengan *basic service area*, yang akan berkembang sesuai dengan karakteristik media *wireless*. Ketika sebuah stasiun terdapat pada *basic service area*, stasiun ini dapat berkomunikasi dengan anggota lain dari BSS. Terdapat 2 macam BSS yang diilustrasikan pada Gambar 2.3.



Gambar 2.3. Independent dan Infrasructure BSS

1. *Independent BSS (IBSS)*

IBSS atau dikenal juga sebagai jaringan Ad Hoc (*peer to peer*) adalaah tipe jaringan yang masing-masing stasiun dapat berkomunikasi secara langsung dalam satu area . Jaringan Ad Hoc yang paling kecil terdiri dari dua stasiun. Umumnya jaringan ini dibentuk untuk tujuan khusus dalam periode yang relatif singkat

2. *Infrastructure BSS*

Infrastructure BSS (tidak pernah disebut sebagai IBSS) atau jaringan Infrastruktur dibedakan dengan jaringan sebelumnya oleh penggunaan AP. AP digunakan untuk semua komunikasi pada jaringan ini, termasuk komunikasi antara stasiun dalam area yang sama. Jika sebuah stasiun akan berkomunikasi dengan stasiun lain maka komunikasi akan melalui dua loncatan (*hop)*. Pertama, stasiun sumber akan mengirimkan frame kepada AP. Kedua, AP akan meneruskan frame tersebuat ke stasiun tujuan.

2.4 Teknik Keamanan WLAN

2.4.1 SSID

SSID (*Service Set Identification*) adalah nama untuk identifikasi sebuah jaringan WLAN, memiliki panjang maksimun 32 karakter. Semua piranti *wireless* pada sebuah WLAN harus memiliki SSID yang sama untuk dapat berkomunikasi. SSID berlaku seperti password sederhana untuk membagi WLAN menjadi beberapa

jaringan yang masing-masing memiliki identitas yang unik. Secara default router *wireless* atau AP akan menyiarkan (*broadcast*) signal setiap 1/10 detik, yang berisi SSID kepada semua alat pada WLAN. Konfigurasi SSID yang tepat akan mencegah seorang intruder untuk menggunakan layanan WLAN, seperti merubah nama standar SSID. [ANO03] ,[ANO04]

2.4.2 WPA2-PSK

WPA2 adalah pengembangan dari WPA (*Wi-Fi Protected Access*). Merupakan penerapan dari standar 802.11i yang diimplementasikan oleh *Wi-Fi alliance*. WPA2 memiliki kemampuan yang lebih baik dibanding WPA dikarenakan penggunaan enkripsi AES (*Advanced Encryption Standard*) yang mendukung kunci 128 bit, 192-bit dan 256 bit.

Mode PSK (*Pre-Shared Key*) atau dikenal juga dengan mode personal digunakan pada lingkungan yang tidak menggunakan server otentifikasi. Mode ini mengharuskan *user* untuk mengisikan passphrase untuk mengakses jaringan. Passphrase dapat berupa 8 sampai 63 karakter ASCII atau 64 digit hexadecimal. Passphrase akan disimpan dalam komputer *user*. [ANO06], [WIK07]

2.4.3 VLAN

VLAN (*Virtual Local Area Network*) adalah sekumpulan piranti dan *user* yang dikelompokkan secara logik, kumpulan ini di dasarkan pada kebutuhan terhadap

d. Untuk meningkatkan kemampuan keamanan digunakan Firewall dan *Intrusion Detection System* (IDS) yang akan disesuaikan dengan memori yang terdapat pada WRT54G.

1.4 Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah untuk mendapatkan nilai tambah fungsi WRT54G dengan penggunaan OpenWRT, meningkatkan keamanan dengan membuat *perimeter* langsung pada *wireless* router dan dihasilkannya log dari aktifitas user untuk memantau lalu lintas jaringan dan keperluan analisis lanjutan.

1.5 Manfaat Penelitian

Manfaat yang diperoleh bagi perusahaan, institusi, dan khususnya yang menggunakan sistem ini adalah sebagai pertimbangan dan perbandingan dengan sistem yang sudah ada dan jika belum ada dapat menggunakan sistem ini.

Manfaat yang dapat diambil dari penelitian ini bagi para pengembang atau developer sistem jaringan dapat sebagai contoh acuan untuk mengembangkan sistem yang terintegrasi untuk meningkatkan level keamanan WLAN.

1.6 Metodologi Penelitian

Beberapa metode yang digunakan untuk menyelesaikan penelitian ini adalah sebagai berikut :

a. Metode Pengumpulan data

Metode pengumpulan data yang dipakai adalah menggunakan landasan literatur dengan mempelajari teori-teori yang berhubungan dengan sistem keamanan jaringan, RFC (standar jaringan internet) dan literatur-literatur lain yang dapat membantu dalam memecahkan masalah yang ada serta dengan melakukan konsultasi secara berkesinambungan dengan dosen pembimbing.

b. Observasi

Pada tahap ini dilakukan observasi untuk mengetahui lingkungan yang dibutuhkan agar implementasi dapat diterapkan dan fungsi yang diharapkan dapat terpenuhi oleh sistem. Desain ini akan digunakan sebagai dasar untuk implementasi yang menghasilkan paket-paket dan fitur yang akan digunakan dalam sistem. Observasi dilakukan pada lingkungan *wireless* FTI UII.

fungsionalitas dari suatu jaringan. Semua informasi yang mengandung penandaan/pengalamatan suatu VLAN (*tagging*) di simpan dalam suatu database (tabel), jika penandaannya berdasarkan port yang digunakan maka database harus mengindikasikan port-port yang digunakan oleh VLAN. Untuk mengaturnya maka biasanya digunakan switch/bridge yang *manageable*. Switch/bridge inilah yang bertanggung jawab menyimpan semua informasi dan konfigurasi suatu VLAN dan dipastikan semua switch/bridge memiliki informasi yang sama. Switch akan menentukan kemana data-data akan diteruskan dan sebagainya.atau dapat pula digunakan suatu *software* pengalamatan (*bridging software*) yang berfungsi mencatat/menandai suatu VLAN beserta workstation yang didalamnya.untuk menghubungkan antar VLAN dibutuhkan router. [Y3D07]

Penggunaan LAN telah memungkinkan semua komputer yang terhubung dalam jaringan dapat bertukar data. Kerjasama ini semakin berkembang dari hanya pertukaran data hingga penggunaan peralatan secara bersama (*resource sharing* atau disebut juga *hardware sharing*). Beberapa LAN memungkinkan data tersebar secara *broadcast* keseluruh jaringan, hal ini akan mengakibatkan mudahnya pengguna yang tidak dikenal (*unauthorized user*) untuk dapat mengakses semua bagian dari *broadcast*. Semakin besar *broadcast*, maka semakin besar akses yang didapat. [Y3D07]

VLAN yang merupakan hasil konfigurasi switch menyebabkan setiap port switch diterapkan menjadi milik suatu VLAN. Oleh karena berada dalam satu segmen, port-port yang bernaung dibawah suatu VLAN dapat saling berkomunikasi

langsung. Sedangkan port-port yang berada di luar VLAN tersebut atau berada dalam naungan VLAN lain, tidak dapat saling berkomunikasi langsung karena VLAN tidak meneruskan broadcast. [Y3D07]

VLAN yang memiliki kemampuan untuk memberikan keuntungan tambahan dalam hal keamanan jaringan tidak menyediakan pembagian/penggunaan media/data dalam suatu jaringan secara keseluruhan. Switch pada jaringan menciptakan batas-batas yang hanya dapat digunakan oleh komputer yang termasuk dalam VLAN tersebut. Hal ini mengakibatkan administrator dapat dengan mudah mensegmentasi pengguna, terutama dalam hal penggunaan media/data yang bersifat rahasia *(sensitive information)* kepada seluruh pengguna jaringan yang tergabung secara fisik. [Y3D07]

2.4.4 Firewall

Firewall merupakan suatu cara atau mekanisme yang diterapkan baik terhadap *hardware*, software ataupun sistem itu sendiri dengan tujuan untuk melindungi, baik dengan menyaring, membatasi atau bahkan menolak suatu atau semua hubungan/kegiatan suatu segmen pada jaringan pribadi dengan jaringan luar yang bukan merupakan ruang lingkupnya. Segmen tersebut dapat merupakan sebuah *work*station, *server, router,* atau LAN . [FIR06]

Firewall secara umum di peruntukkan untuk melayani :

1. Mesin/Komputer

Setiap mesin/komputer yang terhubung langsung ke jaringan luar atau internet dan menginginkan semua yang terdapat pada komputernya terlindungi.

2. Jaringan

Jaringan komputer yang terdiri lebih dari satu buah komputer dan berbagai jenis topologi jaringan yang digunakan, baik yang di miliki oleh perusahaan, organisasi dsb.

Sebelum mendesain firewall apa yang layak digunakan perlu diketahui beberapa karakteristik firewall dan teknik yang digunakan oleh sebuah firewall.

Karakteristik sebuah firewall:

1. Seluruh hubungan/kegiatan dari dalam ke luar, harus melewati firewall. Hal ini dapat dilakukan dengan cara memblok/membatasi baik secara fisik semua akses terhadap jaringan lokal, kecuali melewati firewall. Banyak sekali bentuk jaringan yang memungkinkan agar konfigurasi ini terwujud.
2. Hanya kegiatan yang terdaftar/dikenal yang dapat melewati/melakukan hubungan, hal ini dapat dilakukan dengan mengatur *policy* pada konfigurasi keamanan lokal. Banyak sekali jenis firewall yang dapat dipilih sekaligus berbagai jenis *policy* yang ditawarkan.
3. Firewall itu sendiri haruslah kebal atau relatif kuat terhadap serangan/kelemahan. Hal ini berarti penggunaan sistem yang dapat dipercaya dan dengan sistem yang relatif aman.

Teknik yang digunakan oleh sebuah firewall :

1. *Service control* (kendali terhadap layanan)

Berdasarkan tipe-tipe layanan yang digunakan di Internet dan boleh diakses baik untuk kedalam ataupun keluar firewall. Biasanya firewall akan mencek no IP Address dan juga nomor port yang di gunakan baik pada protokol TCP dan UDP, bahkan bisa dilengkapi software untuk *proxy* yang akan menerima dan menterjemahkan setiap permintaan akan suatu layanan sebelum mengijinkannya. Bahkan bisa jadi software pada server itu sendiri, seperti layanan untuk web ataupun untuk mail.

2. *Direction Control* (kendali terhadap arah)

Berdasarkan arah dari berbagai permintaan (*request*) terhadap layanan yang akan dikenali dan diijinkan melewati firewall.

3. *User control* (kendali terhadap pengguna)

Berdasarkan pengguna/*user* untuk dapat menjalankan suatu layanan, artinya ada *user* yang dapat dan ada yang tidak dapat menjalankan suatu servis,hal ini di karenakan *user* tersebut tidak di ijinkan untuk melewati firewall. Biasanya digunakan untuk membatasi *user* dari jaringan lokal untuk mengakses keluar, tetapi bisa juga diterapkan untuk membatasi terhadap pengguna dari luar.

4. *Behavior Control* (kendali terhadap perlakuan)

Berdasarkan seberapa banyak layanan itu telah digunakan. Misal, firewall dapat memfilter e-mail untuk menanggulangi/mencegah *spamming*.[FIR06]

2.4.5 IDS

Intrusion Detection System (IDS) adalah sekumpulan set teknik dan metode yang digunakan untuk mendeteksi aktifitas yang mencurigakan baik pada level host maupun jaringan. IDS terbagi menjadi 2 kategori dasar. IDS berbasis *signature* dan IDS berbasi *anomaly*. Setiap penyusup memiliki signature, seperti virus komputer, yang dapat dideteksi menggunakan software. Berdasarkan sekumpulan signature dan rule, sistem deteksi akan dapat menemukan dan mencatat aktifitas yang mencurigakan dan mengasilkan *alert*. Deteksi intrusi berdasar anomali biasanya tergantung pada anomali paket yang terdapat pada bagian header protokol. Pada beberapa kasus metode ini memproduksi hasil yang lebih baik jika dibandingkan dengan IDS berbasis signature. Biasanya sebuah IDS mengambil data dari jaringan, kemudian menggunakan rule yang ada pada data tersebut atau mendeteksi anomali padanya. [RAF03]

Alert adalah berbagai jenis pemberitahuan terhadap aktifitas intrusi. Ketika sebuah IDS mendeteksi *intruder*, IDS akan menginformasikan administrator keamanan dengan menghasilkan alert. Alert dapat di tampilkan dalam kotak dialog, dicatat dalam konsol, dikirim ke email dan sebagainya. Alert juga dapat disimpan dalam file log atau database yang dapat dilihat kemudian. Log ini dapat dianalisa baik menggunakan program lain untuk mengetahui aktifitas intrusi dengan lebih detail. [RAF03]

Terdapat dua model dasar IDS, yaitu :

1. NIDS

Network Intrusion Detection System (NIDS) adalah IDS yang mengambil paket data yang terdapat pada media jaringan (kabel, *wireless*) dan mencocokannya dengan *signature* pada basis data. Berdasar pada apakah paket tersebut cocok dengan *signature* intrusi, *alert* akan dihasilkan atau paket akan dicatat pada sebuah file pada database. [RAF03]

2. HIDS

Host-Based Intrusion Detection Sistem atau HIDS diinstal sebagai agen pada sebuah host. IDS ini dapat melihat kedalam sistem atau file log aplikasi untuk mendeteksi aktifitas *intruder*. Beberapa dari sistem ini bersifat *reactive*, artinya informasi akan dihasilkan sistem ketika sesuatu terjadi. Beberapa HIDS yang lain bersifat *proactive*, artinya sistem akan melakukan *sniffing* lalulintas jaringan yang datang pada sebagian host dimana HIDS diinstal dan memberikan *alert* secara *real time*. [RAF03]

2.5 Port Scanning

Port adalah sebuah koneksi elektronik yang mengijinkan data untuk berjalan dari kliet ke server pada sebuah jaringan. *Port Scan* adalah data yang dikirimkan oleh seorang *attacker* melalui Internet untuk mengetahui lokasi komputer dan jaringan dan

mengetahui apakah kompuer atau jaringan tersebut memiliki port terbuka yang akan menerima koneksi.

Port Scanning dedifeinisikan sebagai salah satu teknik pengintaian paling populer yang digunakan oleh seorang *attacker* untuk menemukan *services* yang dapat dimanfaatkan atau dirusak. Semua komputer yang terhubung dengan LAN atau Internet menjalankan banyak service yang mendengarkan *well-known port*(port yang sering dipakai) dan juga *not well-known port*. Dengan *Port Scanning* seorang *attacker* akan menemukan port mana yang tersedia (sedang mendengarkan *service*). Pada dasarnya *Port Scanning* akan mengirim pesan ke masing-masin port, sekali dalam satu waktu. Berbagai macam balasan yang diterima menandakan apakah port sedang digunakan dan oleh karena itu akan diselidiki untuk mengetahui kelemahannya lebih lanjut. [PRA01]

Salah satu *Port Scanner* adalah Nmap. Nmap (Network Maper) adalah utilitas Open Source untuk memeriksa jaringan atau audit keamanan. Nmap didesain untuk melakukan *scanning* pada jaringan luas dengan cepat dan bekerja baik pada *host* tunggal. [ANO02]

2.6 Nikto

Nikto adalah *Scanner* web server dengan lisensi *Open Source* (GPL/*General Public License*) yang melakukan serangkaian test yang meliputi banyak hal untuk item yang berbeda-beda, meliputi lebih dari 3200 bahaya yang potensial terhadap

file/CGI, lebih dari 625 versi server, dan masalah-maslah khusus yang dihadapi oleh lebih dari 230 server. [ANO06]

BAB III

METODOLOGI

3.1 Metode Analisis

Metode analisis yang digunakan untuk membangun sistem ini adalah metode analisis berararah aliran data dan pendekatan struktural. Dengan metode ini diharapkan terkumpul data-data dan informasi yang berkaitan dengan alat dan teknik konfigurasi untuk membangun dan mengembangkan sistem. Sehingga hasil analisis dapat menghasilkan sistem yang strukturnya dapat didefinisikan dengan jelas. Analisis juga memperhatikan desain jaringan untuk implementasi sistem.

3.2 Hasil Analisis

Hasil analisis dari sistem yang diharapkan adalah sebagai berikut :

3.2.1 Gambaran Umum Sistem

Secara garis besar sistem jaringan WLAN yang akan dibangun masih tergabung dengan jaringan LAN. Untuk itu dirancang sebuah mekanisme yang secara logik memisahkan jaringan-jaringan tersebut dengan tetap menggunakan perangkat keras yang sama.

Setiap *user* WLAN yang akan menggunakan layanan ini harus mengetahui SSID dan kunci enkripsi yang telah dikonfigurasi dalam sistem. *User* yang mengisikan SSID dan kunci enkripsi sesuai dengan konfigurasi akan mendapat alamat IP yang diberikan oleh DHCP server yang terintegrasi dalam router.

Untuk memperkuat ketahanan router *wireless*, sistem dilengkapi dengan IDS yang akan mencatat setiap aktifitas mencurigakan yang telah didefiniskan dalam konfigurasi IDS. Catatan/log digunakan sebagai peringatan akan adanya iintrusi oleh *user*. Log tersebut akan dikirimkan ke sebuah *remote logging* untuk mengurangi beban pemakain resource yang terbatas.

Remote logging yang merupakan server log akan menyimpan dan mengolah log untuk memudahkan pembacaan oleh administrator. Log dapat dilihat secara *real time* ataupun diolah dengan bantuan aplikasi lain yang akan dikirim ke webserver. Webserver digunakan sebagai pelengkap untuk memudahkan pembacaan log, artinya sistem ini tidak membangun webserver yang kompleks tapi mengasumsikan telah terdapat webserver. Karena sistem yang dibangun berpusat pada router *wireless*.

3.2.2 Analisa Masukan Sistem

Sistem yang dibangun membutuhkan masukan-masukan sebagai berikut :

1. *Command* dari administrator untuk konfigurasi
2. Nama SSID dan kunci enkripsi
3. Alamat IP *client wireless*

4. Waktu akses
5. Intrusi oleh *user*.

3.2.3 Analisis Proses

Beberapa proses yang akan dilakukan oleh sistem antara lain :

1. Proses pembentukan 3 buah VLAN
2. Proses otentifikasi dengan mengisikan SSID dan password yang telah dienkripsi.
3. Proses intrusi dengan menggunakan aplikasi yang tersedia
4. Proses pencatatan log yang dikirm ke *remote logging* terhadap intrusi yang terjadi.
5. Proses pengolahan log menjadi file html
6. Proses pengiriman log html dan log asli ke web server.

3.2.4 Analisa Keluaran

Sistem yang akan dibangun akan menghasilkan keluaran berupa log. Log ini merupakan informasi yang didapat dari gabungan proses-proses yang terjadi dalam sistem. Log dapat dibaca secara real time pada saat terjadinya aktifitas intrusi atau disimpan dalam sebuah file. File yang terbentuk berupa file log asli dan file log yang telah diubah menjadi file html. File-file tersebut disesuaikan dengan tanggal pada saat

log dikirmkan ke web server. File log ini juga dapat digunakan untuk analisis lanjutan.

3.2.5 Kinerja yang harus dipenuhi

Sistem yang dibangun berdasarkan kemampuan router dan konfigurasi menggunakan OpenWRT ini diharapkan dapat memenuhi kebutuhan-kebutuhan antara lain :

1. Mampu menangani otentikasi berbasis WPA.
2. Mampu memberikan informasi kepada administrator akibat adanya intrusi dengan menghasilkan *alert.*

3.2.6 Fungsionalitas yang dikehendaki

Sistem yang dibangun diharapkan mampu menguatkan sistem terhadap serangan yang datang. Hal ini merupakan teknik pertahanan sebagai bagian dari sistem keamanan jaringan. Selain itu transfer data dilakukan secara terenkripsi dan pengawasan terhadap aktifitas *user* dapat dilakukan dengan baik.

3.2.7 Antarmuka yang diinginkan

Antarmuka yang diingiinkan dari sistem ini sebagi berikut :

1. Mampu menyuguhkan informasi secara real time kepada administrator

2. Mampu menyuguhkan informasi yang mudah dibaca oleh administrator dalam bentuk html.

3.3 Perancangan Sistem

3.3.1 Metode Perancangan

Metode yang digunakan dalam perancangan Implementasi *Linux Embedded System* untuk *Intrusion Detection System* menggunakan OpenWRT pada *Wireless* Router WRT54G dengan menggunakan diagram aliran data (DFD). Dimana tahapan untuk tiap-tiap proses *input* hingga *output* yang terjadi dalam sistem digambarkan dalam sebuah diagram aliran data secara jelas dan mudah untuk dimengerti.

3.3.2 Hasil Perancangan

Berikut ini adalah hasil perancangan dari sistem yang akan dibangun.

3.3.2.1 Data Flow Diagram

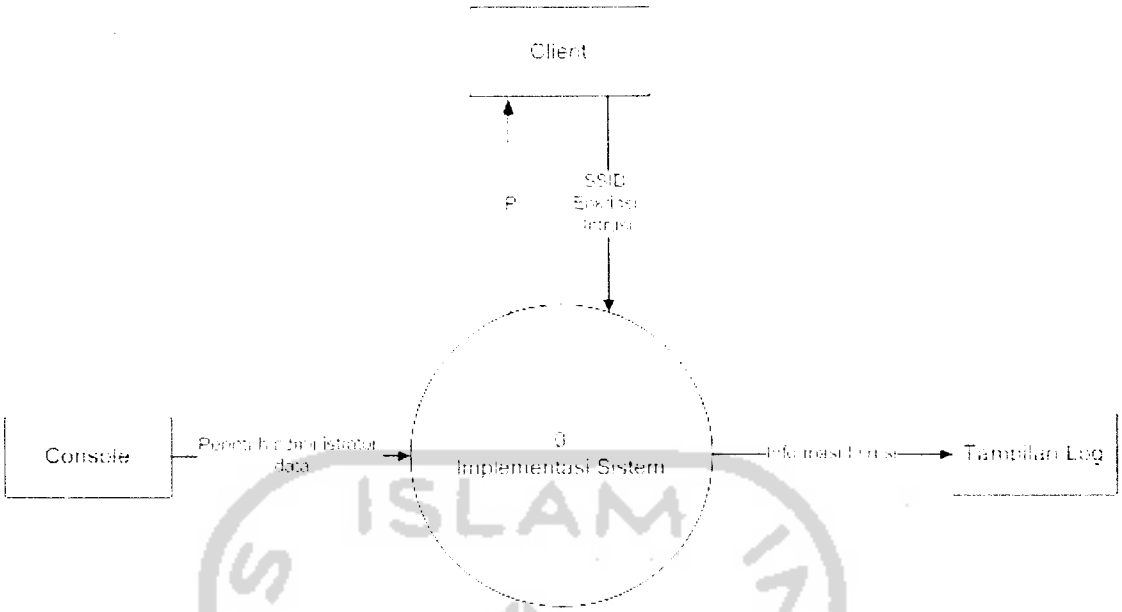
Data Flow Diagram (DFD) adalah diagram yang menggunakan notasi-notasi simbol untuk mewakili kesatuan luar atau batas sistem, arus data, proses dan simpanan data. Pendekatan DFD merepresentasikan proses-proses data di dalam sistem dan menekankan pada logika yang mendasari sistem sehingga dapat

mempermudah seseorang untuk mengerti bentuk sistem dan mempermudah dalam pengembangannya.

DFD digunakan untuk menjelaskan bagaimana data ditransformasikan dalam suatu sistem. Selain itu DFD juga menggambarkan fungsi-fungsi yang digunakan dalam mentransformasikan data dalam sistem yang telah dibangun. DFD terdiri dari beberapa tingkatan (*level*) berdasarkan kebutuhan sistem. Semakin rendah level DFD maka semakin rinci fungsi yang dijabarkan suatu proses. Berikut ini akan dijelaskan beberapa level DFD yang digunakan dalam Implementasi tugas akhir ini.

a. Diagram Konteks

Diagram konteks atau yang biasa disebut dengan DFD (*Data Flow Diagram*) *level* 0 merupakan tingkatan tertinggi dalam suatu struktur perancangan DFD. Diagram konteks hanya terdiri dari sebuah proses berupa masukan-masukan dasar, sistem umum dan keluaran proses yang merupakan gambaran sistem secara keseluruhan.

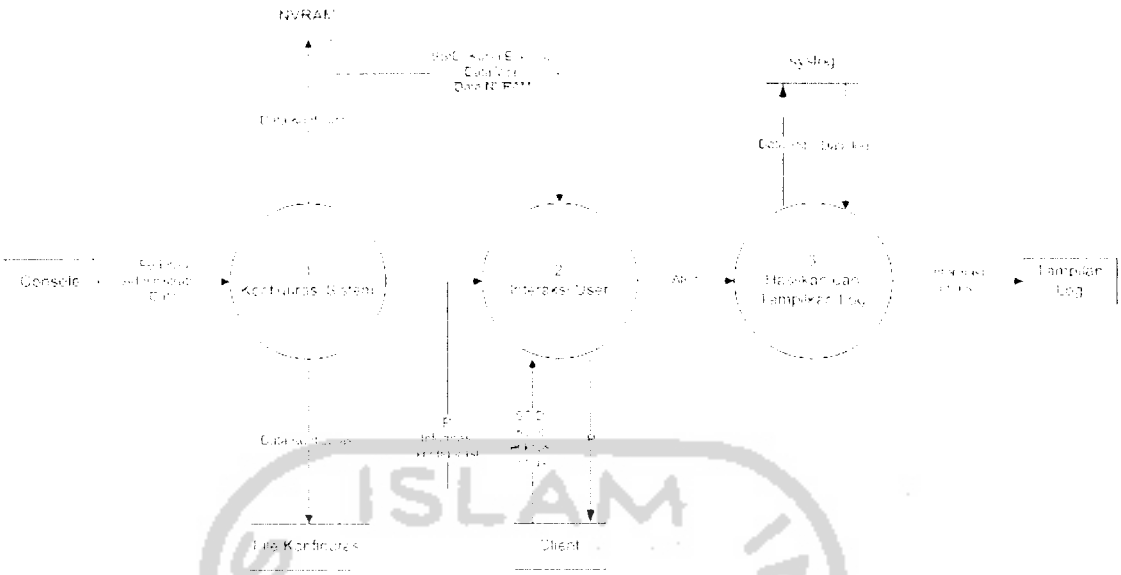


Gambar 3.1. Diagram Konteks Sistem

Diagram Konteks sistem pada Gambar 3.1 diatas menjelaskan bahwa sistem terdir dari tiga entitas objek, yaitu *console* sebagai alat administrator melakukan konfigurasi, *client* dan tampilan log yang merupakan *remote logging*. Pada suatu waktu *client* merupakan *attacker* yang akan melakukan sebuah intrusi. Oleh sistem intrusi akan diolah dan ditampilkan sebagai log yang akan memudahkan administrator mengetahui apa yang sedang terjadi.

b. DFD level 1

DFD level 1 merupakan perincian dari diagram konteks dimana proses pada diagram konteks dibagi menjadi tiga proses sistem yang terdiri dari proses konfigurasi sistem, proses interaksi *user* dan proses menghasilkan dan menampilkan log seperti yang dijelaskan pada Gambar 3.2.



Gambar 3.2. DFD Level 1

Implementasi ini dimulai dengan masukkan oleh administrator melalui *console* untuk melakukan konfigurasi sistem. Konfigurasi sistem akan disimpan dalam NVRAM (Non *Volatile Random Access Memory*) dan file-file konfigurasi. NVRAM dan file konfigurasi akan menjadi acuan bagi proses yang akan dilakukan selanjutnya. Pada proses interaksi *user*, *client* akan memasukkan SSID dan kunci enkripsi untuk mendapatkan alamat IP. NVRAM akan mencocokkan masukkan tersebut. Jika sesuai *client* akan diberikan alamat IP berdasar file konfigurasi secara dinamis.

Intrusi oleh *user* akan membangkitkan *alert* yang menghasilkan informasi log dan ditampilkan dalam tampilan *remote logging*.

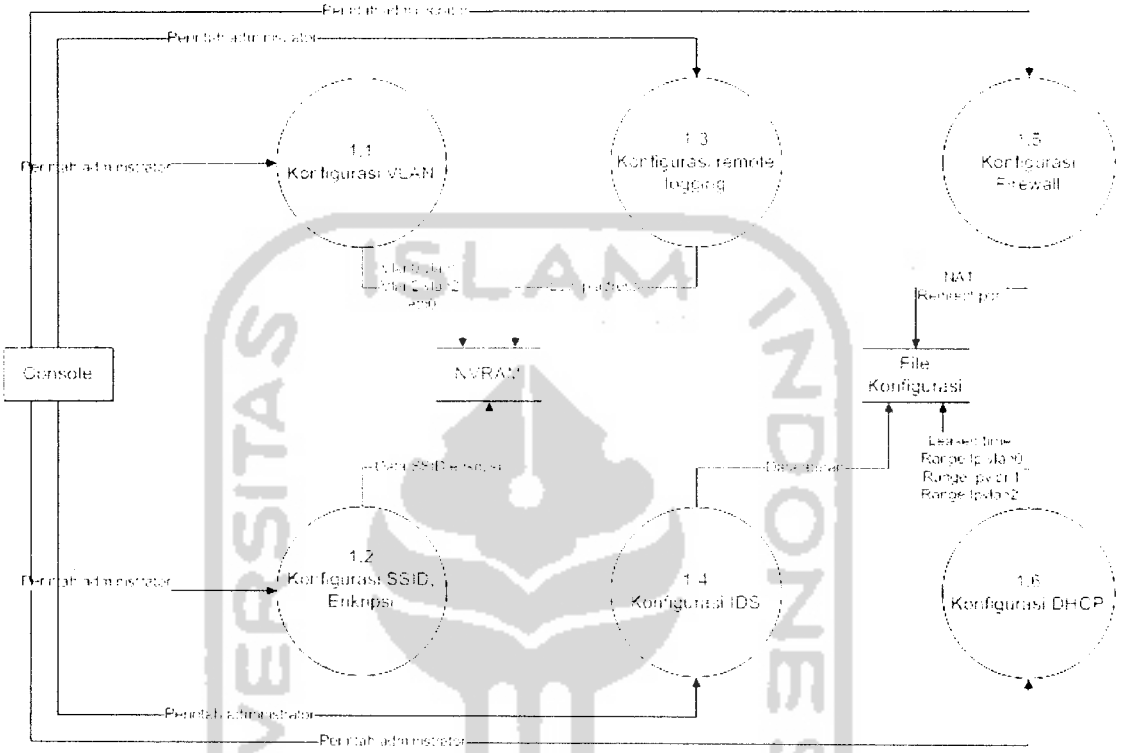
c. DFD Level 2

Pada DFD level 2 ini, setiap proses dari level sebelumnya akan dibagi kedalam proses-proses yang lebih detail. Proses konfigurasi sistem dipecah dalam enam buah

proses konfigurasi yang berbeda yaitu : konfigurasi VLAN, konfigurasi SSID dan Enkripsi, konfigurasi *remote logging*, konfigurasi IDS, konfigurasi firewall dan konfigurasi DHCP. Tiga konfigurasi awal adalah konfigurasi yang semua datanya akan disimpan dalam NVRAM dan tiga selanjutnya adalah konfigurasi yang akan disimpan dalam file-file konfigurasi.

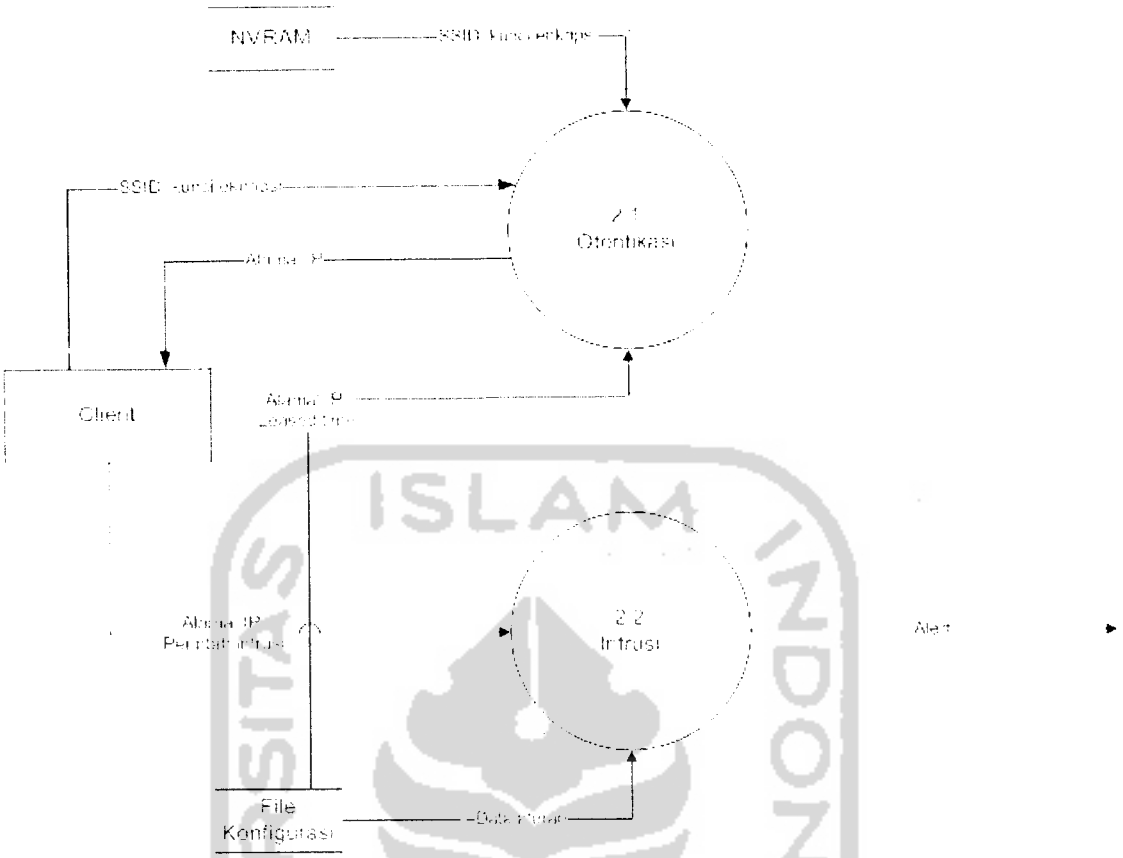
Konfigurasi VLAN dilakukan untuk membagi jaringan menjadi tiga buah jaringan yang terpisah secara logik dengan mendefinisikan jaringan vlan0, vlan1 dan vlan2. Disini juga akan didefinisikan inerface jaringan Wide Area Network (WAN) yang merupakan alamat IP untuk berhubungan dengan inernet. Konfigurasi SSID dan enkripsi adalah konfigurasi untuk mendefinisikan id jaringan dan kunci enkripsi yang harus diketahui oleh setiap *client* yang akan menggunakan fasilitas dalam sistem. Konfigurasi lain-lain adalah setiap konfigurasi yang berdiri secara terpisah tetapi memiliki peran penting karena mendukung setiap proses dalam implementasi ini. Konfigurasi IDS adalah konfigurasi yang dilakukan pada file-file yang bertugas menjalankan *monitoring* terhadap intrusi, *alert* apa yang akan dihasilkan dan dimana log akan ditampilkan. Konfigurasi ini akan menyimpan data aturan untuk kepentingan tersebut. Pada konfigurasi firewall masukkan berupa NAT (*Network address translator*) yang akam merubah setiap alamat IP dari *client* dijaringan untuk berhubungan dengan internet melalui alamat publik yang didefinisikan sebelumnya. Selain itu juga akan dilakukan pembelokan port jika terjadi permintaan pada port tertentu. Konfigurasi DHCP (*Dynamic Host Control Protocol*) berhubungan dengan tiga buah jaringan yang dibentuk sebelumnya. Konfigurasi ini akan menyimpan data

mengenai *leased time*, yaitu waktu yang diberikan setiap *client* untuk menggunakan sebuah alamat IP dan *range* alamat-alamat IP untuk tiap jaringan. Gambar 3.3 memperlihatkan penjelasan diatas.



Gambar 3.3. DFD Level 2 Proses Konfigurasi Sistem

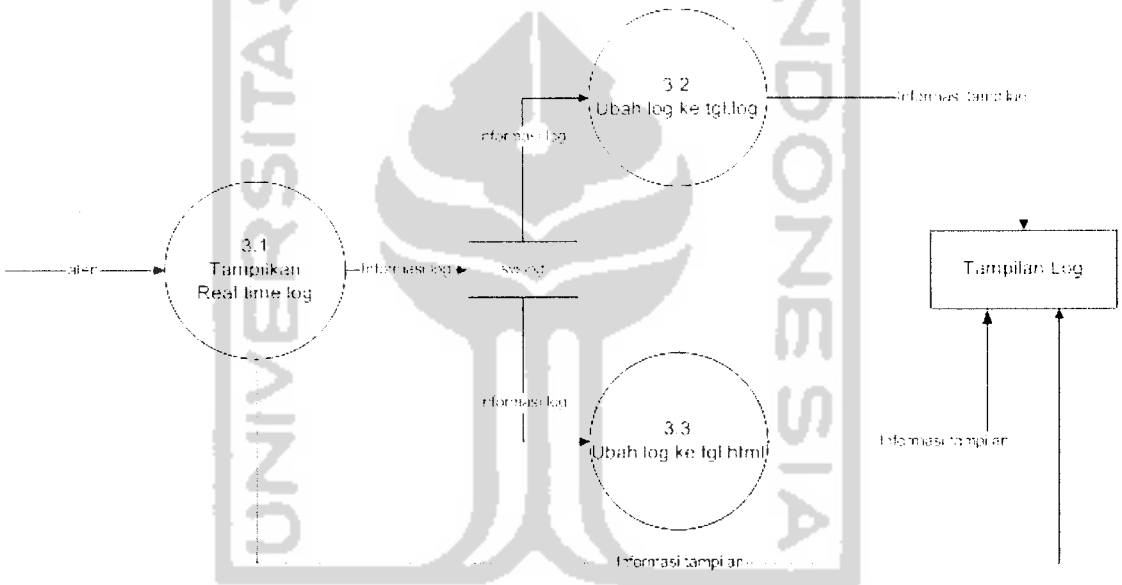
Proses selanjutnya dari DFD level 2 ini adalah proses Interaksi *User*. Proses ini dibagi dalam dua proses yang lebih detail seperti yang ditunjukkan oleh Gambar 3.4.



Gambar 3.4. DFD Level 2 Proses Interaksi User

Pada proses otentikasi, *client* akan memasukkan data berupa SSID dan kunci enkripsi, masukkan itu direspon oleh sistem dengan mengecek SSID dan kunci enkripsi pada NVRAM jika masukkan benar, maka data pada file konfigurasi akan memberikan alamat IP dan *leased time* kepada *client*. Selanjutnya adalah pembangkitan proses intrusi jika *client* melakukan aktifitas intrusi. Aktifitas ini dimulai dengan perintah-perintah intrusi yang dilakukan oleh *user*. Perintah-perintah ini menjadi masukkan bagi proses ini selain alamat IP dan informasi lainnya untuk menghasilkan alert yang akan diolah pada proses berikutnya.

Alert yang dihasilkan pada proses diatas akan menghasilkan log yang telah lengkap sebagai informasi. Proses pengumpulan data alert menjadi informasi log yang lengkap terjadi pada proses lanjutan, yaitu proses Hasilkan dan Tampilkan log. Informasi ini kemudian dikirim ke remote logging yang dapat ditampilkan dalam bentuk log *real time*. Kemudian log ini disimpan dalam *syslog* dan diolah untuk menghasilkan log sesuai tanggal log dikirim untuk keperluan analisis lanjutan. Log ini juga diolah dalam bentuk html untuk memudahkan administrator menganalisa intrusi yang terjadi. Penjelasan dari proses ini dijelaskan pada Gambar 3.5.



Gambar 3.5. DFD Level 2 Proses Hasilkan dan Tampilkan Log

3.3.2.2 NVRAM dan File Konfigurasi

NVRAM adalah memori internal pada WRT54G yang akan menjadi sumberdaya penyimpanan. Karena sifatnya yang *non volatile*, memori ini digunakan

sebagai tempat menyimpan konfigurasi yang dilakukan oleh administrator. Konfigurasi ini digunakan untuk menjalankan sistem sesuai dengan keinginan administrator. Besarnya memori yang terbatas yaitu 16MB mengharuskan semua konfigurasi dan paket yang diinstal memperhitungkan fungsi dan kebutuhan lingkungan dimana sistem akan diterapkan.

Oleh karena itu implementasi sistem ini khususnya penggunaan Snort sebagai IDS tidak menerapkan semua aturan yang terdapat dalam paket Snort. Beberapa kumpulan aturan akan dinonaktifkan untuk menghindari terjadinya error sistem akibat penggunaan memori yang lebih besar dibandingkan ketersediaan yang hanya 16MB. Selain itu NVRAM juga akan mendefinisikan dimana log akan dikirim, hal ini dilakukan agar log yang ada tersimpan pada *remote logging* dan tidak terhapus jika router WRT54G direstart.

Disamping NVRAM, file konfigurasi juga menjadi penunjang berjalannya sistem. File konfigurasi pada sistem ini antara lain :

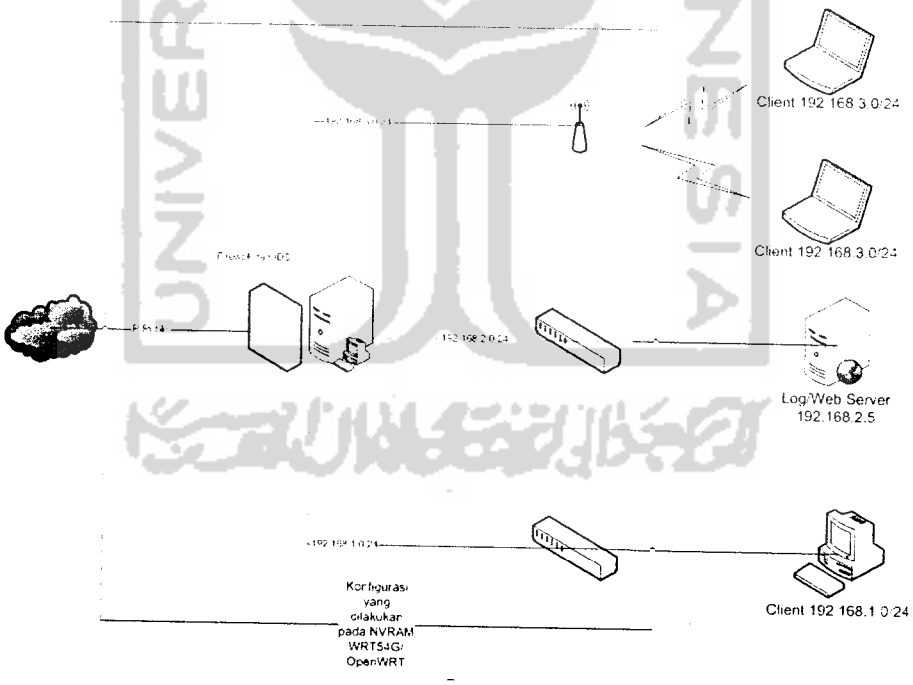
a. Firewall
b. Dhcp
c. Snort
d. Dnsmasq

3.3.2.3 Rancangan Antarmuka

Antarmuka akan disesuaikan dengan tampilan log yang dihasilkan oleh sistem secara otomatis. Pada log html antarmuka akan dihasilkan oleh aplikasi yang telah ada pada *remote logging*.

3.3.2.4 Rancangan Arsitektur Jaringan

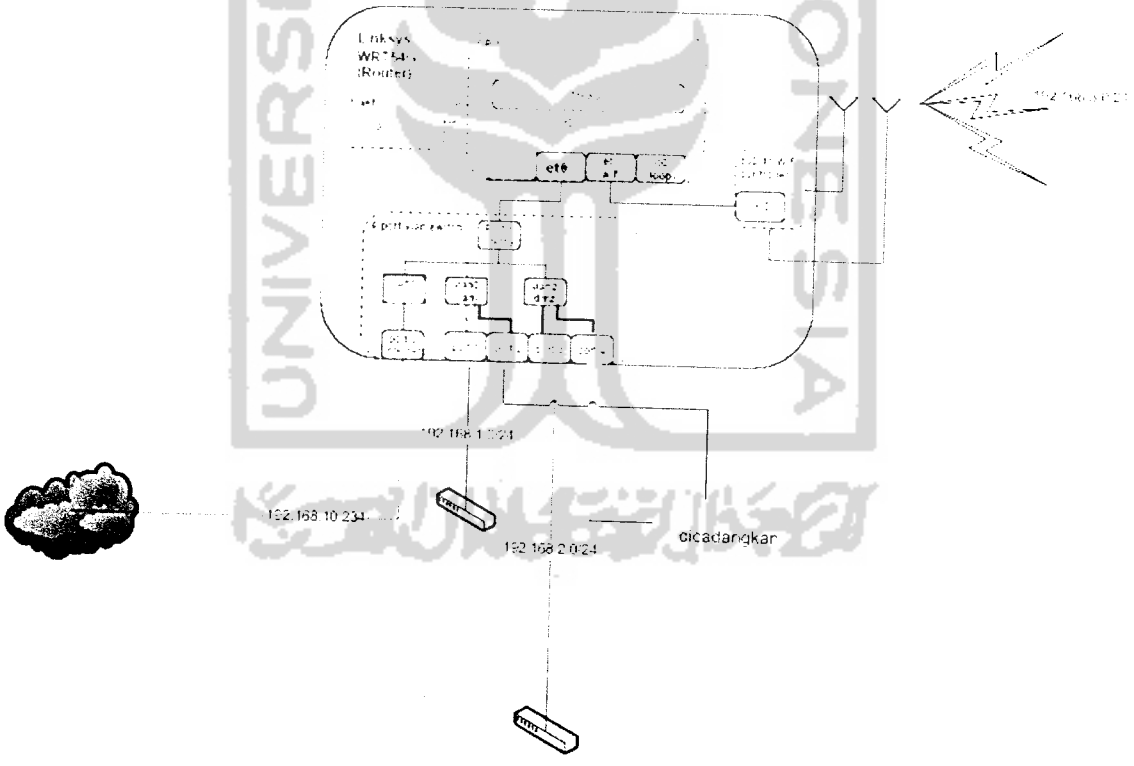
Arsitektur Jaringan dibangun untuk memudahkan implementasi secara logik pada WRT54G. Pada analisis sebelumnya akan dibangun tiga buah vlan yang akan memisahkan vlan0, vlan1 dan vlan2. Gambar 3.6 memperlihatkan rancangan arsitektur jaringan yang mendukung kebutuhan tersebut.



Gambar 3.6. Arsitektur Jaringan

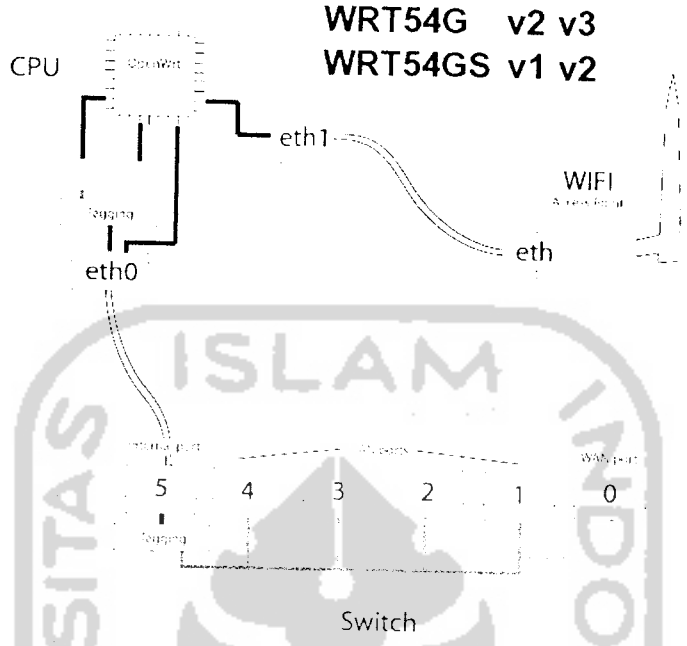
Gambar 3.6 menjelaskan bahwa konfigurasi NVRAM akan membagi jaringan menjadi tiga buah jaringan berbeda. Proses ini dapat dilakukan karena kemampuan WRT54G dan fungs-fungsi yang diberikan oleh OpenWRT sebagai sistem operasi didalamnya Jaringan-jaringan tersebut adalah jaringan LAN pada 192.168.1.0/24, jaringan 12.168.2.0/24 yang akan digunakan oleh server, dan jaringan WLAN pada 192.168.3.0/24. Selain itu terdapat juga IP publik yang akan menghubungkan jaringan internal ke internet.

Pada perangkat keras WRT54G hal ini dapat dimungkinkan dengan membagi port-port yang tersedia untuk digunakan oleh masing-masing jaringan. Penggunaan port-port tersebut ditunjukkan oleh Gambar 3.7.



Gambar 3.7 Arsitektur Port WRT54G. [MAR06]

Sebagai perbandingan arsitektur WRT54G ver 2 yang dijadikan *hardware* pada implementasi ini ditunjukkan oleh Gambar 3.8 sesuai dengan dokumentasi Openwrt.



Gambar 3.8 Arsitektur awal Port WRT54G ver 2 [ANO07]

BAB IV

HASIL DAN PEMBAHASAN

4.1 Batasan Implementasi

Implementasi sistem adalah sebuah proses penerjemahan rancangan yang telah dibuat atau didesain sebelumnya. Pada tahap ini sistem telah siap untuk dioperasikan sesuai dengan fungsi dan tujuan sistem dibangun. Tujuan dari tahapan ini adalah untuk memastikan bahwa sistem yang telah dibuat dapat bekerja dengan baik sesuai dengan yang diinginkan pada saat analisis perangkat lunak. Dalam Implementasi *Linux Embedded System* untuk *Intrusion Detection System* menggunakan *OpenWRT* pada *Wireless* router WRT54G ini terdapat beberapa batasan antara lain :

4.1.1 Asumsi yang Dipakai

Dalam tahap implementasi ini digunakan asumsi-asumsi sebagai berikut :

1. *Wireless* Router yang digunakan adalah WRT54G Ver 2.
2. Firmware standar akan digantikan dengan OpenWRT
3. Enam Port jaringan pada WRT54G akan dibagi dalam tiga VLAN yaitu, VLAN0, VLAN1 dan VLAN2. Jaringan WLAN menggunakan *interface* eth0.

4. VLAN0 merupakan *interface* jaringan yang menghubungkan dengan LAN dengan alamat 192.168.1.1. VLAN1 menghubungkan dengan WAN dengan alamat 192.168.10.234, VLAN2 menghubungkan dengan DMZ dengan alamat 192.168.2.1 dan eth1 yang menghubungkan dengan WLAN memiliki alamat 192.168.3.1.
5. Sistem Operasi komputer Client bersifat multi platform.
6. Digunakan sebuah komputer yang digunakan untuk *remote logging*.
7. Semua konfigurasi dilakukan berbasis teks.
8. Tidak semua aturan pada Snort digunakan karena memori yang terbatas..
9. Untuk masuk ke sistem WRT54G digunakan protokol SSH pada port 22 dengan username *root* dan password *ndeso25*.
10. Tidak dilakukan konfigurasi pada Web server karena web hanya digunakan untuk menampilkan file log.
11. Alamat IP *remote logging* 192.168.2.5
12. Untuk kepertingan uji coba, alamat IP WAN menggunakan alamat IP jaringan FTI selatan yaitu 192.168.10.234.

4.1.2 Lingkungan Pengembangan

Semua konfigurasi sistem baik pada WRT54G maupun server log dikembangkan di lingkungan Linux. Untuk mengkonfigurasi WRT54G dilakukan

dengan *remote acess* melalui Windows dengan aplikasi *putty* atau melalui Linux dengan console.

4.1.3 Perangkat Lunak yang Digunakan

Perangkat lunak yang digunakan untuk mendukung implementasi dan pengembangan sistem ini adalah :

1. OpenWRT, adalah sistem operasi berbasis Linux yang akan menggantikan peran firmware standar pada WRT54G.
2. CentOS, adalah sistem operasi Linux turunan *RedHat* yang akan digunakan sebagai sistem operasi pada *Log Server*.
3. Iptables, adalah aplikasi firewall standar yang terdapat pada OpenWRT.
4. Snort, adalah aplikasi IDS yang akan ditambahkan pada OpenWRT untuk mengetahui intrusi oleh *user*.
5. Nas, adalah paket tambahan untuk mendukung penggunaan WPA2 PSK bagi enkripsi dilingkungan WLAN.
6. Dnsmasq, adalah paket server DHCP dan DNS sederhana yang digunakan untuk memberikan alamat IP pada client secara dinamis.
7. Apache, adalah web server yang akan digunakan sebagai pendukung setelah pengolahan log.

4.1.4 Perangkat Keras yang Digunakan

Perangkat keras yang digunakan untuk implementasi sistem ini adalah :

1. Linksys Broadband Router WRT54G Ver 2.
2. Satu unit komputer dengan spesifikasiyang dimiliki adalah prosesor AMD Barton 2500+, RAM 512MB, Harddisk 40GB.
3. Monitor, Keyboard, Mouse.
4. Kartu Jaringan Realtek 8139.

4.2 Implementasi Sistem

Secara garis besar implementasi sistem sebagai berikut :

1. Melakukan instalasi dengan transfer OpenWRT kedalam WRT54G.
2. Mengkonfigurasi Jaringan.
3. Installasi paket NAS dan konfigurasi parameter untuk wifi.
4. Mengkonfigurasi iptables dan alokasi IP secara dinamis.
5. Membangun IDS dengan Snort.
6. Membangun Remote Logging.
7. Mengolah Data Log.

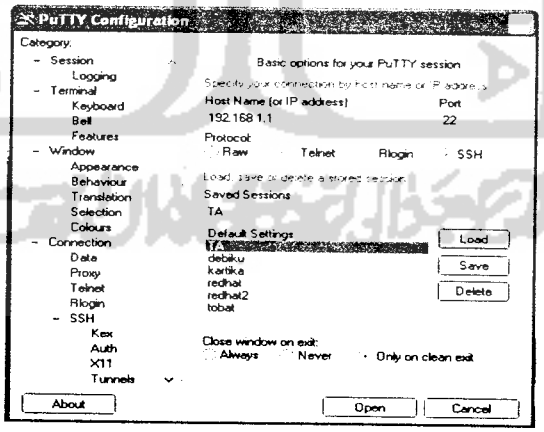
4.2.1 Instalasi OpenWRT

Langkah awal untuk mengimplementasikan sistem ini adalah dengan mengganti firmware standar pada WRT54G dengan OpenWRT yang dirancang untuk arsitektur prosesor MIPS. Transfer dilakukan dengan menggunakan aplikasi *Trivial File Transfer Protokol* (TFTP) yang secara default disediakan oleh Windows dan Linux. TFTP adalah aplikasi yang berjalan pada layer teratas model TCP/IP.

Alamat IP pada WRT54G adalah 192.168.1.1 (ip default) sehingga komputer *remote* diset dengan ip 192.168.1.3. Pada *command prompt* ketikkan perintah

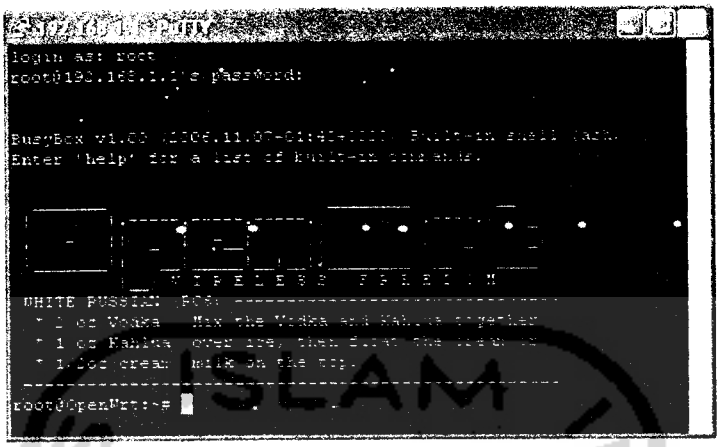
```
tftp -i 192.168.1.1 put openwrt-wrt54g-squashfs.bin
```

Image openwrt akan melakukan installasi secara otomatis setelah diletakkan pada WRT54G. Login ke router dengan telnet dan ganti password sistem. Setelah password diganti maka login ke sistem menggunakan ssh melalui *putty*. Login melalui putty ditunjukkan pada Gambar 4.1.



Gambar 4.1. Login melalui putty

Jika login berhasil maka akan tampak *console* sepertiGambar 4.2.



Gambar 4.2. Tampilan Console OpenWRT

4.2.2 Mengkonfigurasi Jaringan

Sesuai dengan Gambar 3.6 dan Gambar 3.7 pada bab sebelumnya, maka implementasi dilakukan dengan menambahkan parameter yang akan disimpan ke dalam NVRAM.

```
nvram unset lan_ifnames                 # melepas bridging dari interface br0

nvram set vlan0ports="1 2 5*"           # port 1 dan 2 dihubungkan dengan vlan0
nvram set vlan0hwname=et0               # vlan0 menggunakan et(h)0 sebagai real
                                          interface pada WRT54G
nvram set vlan1ports="0 5"              # port 0 dihubungkan dengan vlan1
nvram set vlan1hwname=et0

nvram set vlan2ports="3 4 5"            # port 3 dan 4 dihubungkan dengan vlan2
nvram set vlan0hwname=et0

nvram commit
```

Jika login berhasil maka akan tampak *console* sepertiGambar 4.2.

Gambar 4.2. Tampilan Console OpenWRT

4.2.2 Mengkonfigurasi Jaringan

Sesuai dengan Gambar 3.6 dan Gambar 3.7 pada bab sebelumnya, maka implementasi dilakukan dengan menambahkan parameter yang akan disimpan ke dalam NVRAM.

```
nvram unset lan_ifnames                 # melepas bridging dari interface br0

nvram set vlan0ports="1 2 5*"           # port 1 dan 2 dihubungkan dengan vlan0
nvram set vlan0hwname=et0               # vlan0 menggunakan et(h)0 sebagai real
                                          interface pada WRT54G
nvram set vlan1ports="0 5"              # port 0 dihubungkan dengan vlan1
nvram set vlan1hwname=et0

nvram set vlan2ports="3 4 5"            # port 3 dan 4 dihubungkan dengan vlan2
nvram set vlan0hwname=et0

nvram commit
```

```
### INPUT
###  (connections with the router as destination)
# base case
  iptables -P INPUT DROP
  iptables -A INPUT -m state --state INVALID -j DROP
  iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
  iptables -A INPUT -p tcp --tcp-flags SYN SYN --tcp-option \! 2 -j
DROP

#
# insert accept rule or to jump to new accept-check table here
#
  iptables -A INPUT -j input_rule

# allow
  Iptables -A INPUT -j LAN_ACCEPT          # allow from int wifi/lan .
  iptables -A INPUT -p icmp -j ACCEPT      # allow ICMP
  iptables -A INPUT -p gre -j ACCEPT       # allow GRE

# reject (what to do with anything not allowed earlier)
  iptables -A INPUT -p tcp -j REJECT --reject-with tcp-reset
  iptables -A INPUT -j REJECT --reject-with icmp-port-unreachable

# OUTPUT
# (connections with the router as source)
# base case
  iptables -P OUTPUT DROP
  iptables -A OUTPUT -m state --state INVALID -j DROP
  iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# insert accept rule or to jump to new accept-check table here
  iptables -A OUTPUT -j output_rule

# allow
  iptables -A OUTPUT -j ACCEPT          #allow everything out

# reject (what to do with anything not allowed earlier)
  iptables -A OUTPUT -p tcp -j REJECT --reject-with tcp-reset
  iptables -A OUTPUT -j REJECT --reject-with icmp-port-unreachable

### FORWARDING
### (connections routed through the router)

# base case
  iptables -P FORWARD DROP
  iptables -A FORWARD -m state --state INVALID -j DROP
  iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --
clamp-mss-to-pmtu
  iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
# insert accept rule or to jump to new accept-check table here
  iptables -A FORWARD -j forwarding_rule
```

```
# Forward request ke interface WAN
  iptables -A FORWARD -i $LAN -j ACCEPT
  iptables -A FORWARD -i $WIFI -o $WAN -j ACCEPT
  iptables -A FORWARD -i $DMZ -o $WAN -j ACCEPT

# reject (what to do with anything not allowed earlier)
# uses the default -P DROP

### MASQ
  iptables -t nat -A PREROUTING -j prerouting_rule
  iptables -t nat -A POSTROUTING -j postrouting_rule
  iptables -t nat -A POSTROUTING -o $WAN -j MASQUERADE

## USER RULES
[ -f /etc/firewall.user ] && . /etc/firewall.user
[ -e /etc/config/firewall ] && {
        awk    -f    /usr/lib/common.awk    -f    /usr/lib/firewall.awk
/etc/config/firewall | ash
}
```

Selain itu konfigurasi juga dilakukan pada `/etc/dnsmasq.conf`. Konfigurasi ini untuk memberikan alamat IP secara otomatis kepada client. Pada sistem ini alamat IP untuk LAN adalah 192.168.1.25 sampai 192.168.1.35, untuk wifi adalah 192.168.3.25 sampai 192.168.3.35. Semua alamat IP yang diberikan secara dinamis dapat digunakan selama 12 jam. Berikut konfigurasi tambahan yang diberikan pada file `/etc/dnsmasq.conf` untuk kepentingan diatas :

```
#LAN

dhcp-range=192.168.1.25,192.168.1.35,255.255.255.0,12h

#Wifi

dhcp-range=192.168.3.25,192.168.3.35,255.255.25.0,12h
dhcp-leasefile=/tmp/dhcp.leases
```

4.2.5 Membangun IDS dengan Snort

Installasi Snort merupakan langkah awal untuk membangun IDS. Installasi dilakukan dengan menggunakan *ipkg*. Setelah installasi dilakukan maka perlu dilakukan konfigurasi pada `/etc/snort/snort.conf`.

```
# Step #1: Set the network variables:

var HOME_NET any
var EXTERNAL_NET any
var DNS_SERVERS $HOME_NET
var HTTP_SERVERS $HOME_NET
var HTTP_PORTS 80
var SHELLCODE_PORTS !80
var RULE_PATH /etc/snort/rules

config disable_decode_alerts
config detection: search-method lowmem

# Step #2: Configure preprocessors
preprocessor flow: stats_interval 0 hash 2
preprocessor frag2

preprocessor frag3_global: max_frags 65536
preprocessor frag3_engine: policy first detect_anomalies
preprocessor stream4: disable_evasion_alerts
preprocessor stream4_reassemble
preprocessor sfportscan: proto  { all } \
                         memcap { 10000000 } \
                         sense_level { low }
preprocessor xlink2state: ports { 25 691 }
preprocessor antistumbler: probe_reqs 90, probe_period 30,
expire_timeout 3600

# alert_syslog: log alerts to syslog
output alert_syslog: LOG_AUTH LOG_ALERT

include classification.config
include reference.config

# Step #4: Configure snort with config statements
config flowbits_size: 64

# Step #5: Customize your rule set

#include $RULE_PATH/local.rules
# include $RULE_PATH/bad-traffic.rules
```

```
# include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
#include $RULE_PATH/finger.rules
#include $RULE_PATH/ftp.rules
#include $RULE_PATH/telnet.rules
#include $RULE_PATH/rpc.rules
#include $RULE_PATH/rservices.rules
#include $RULE_PATH/dos.rules
#include $RULE_PATH/ddos.rules
#include $RULE_PATH/dns.rules
#include $RULE_PATH/tftp.rules

#include $RULE_PATH/web-cgi.rules
#include $RULE_PATH/web-coldfusion.rules
#include $RULE_PATH/web-iis.rules
#include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
#include $RULE_PATH/web-client.rules
#include $RULE_PATH/web-php.rules

#include $RULE_PATH/sql.rules
#include $RULE_PATH/x11.rules
#include $RULE_PATH/icmp.rules
#include $RULE_PATH/netbios.rules
#include $RULE_PATH/misc.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/oracle.rules
#include $RULE_PATH/mysql.rules
#include $RULE_PATH/snmp.rules

#include $RULE_PATH/smtp.rules
#include $RULE_PATH/imap.rules
#include $RULE_PATH/pop2.rules
#include $RULE_PATH/pop3.rules

#include $RULE_PATH/nntp.rules
#include $RULE_PATH/other-ids.rules
#include $RULE_PATH/web-attacks.rules
# include $RULE_PATH/backdoor.rules
# include $RULE_PATH/shellcode.rules
# include $RULE_PATH/policy.rules
# include $RULE_PATH/porn.rules
# include $RULE_PATH/info.rules
# include $RULE_PATH/icmp-info.rules
# include $RULE_PATH/virus.rules
# include $RULE_PATH/chat.rules
# include $RULE_PATH/multimedia.rules
# include $RULE_PATH/p2p.rules
# include $RULE_PATH/experimental.rules
# include $RULE_PATH/wifi.rules
```

Konfigurasi diatas hanya mengaktifkan dua set aturan yaitu scan.rules dan web-misc.rules. Masing-masing set aturan tersebut berisi aturan-aturan yang sudah didefinisikan.

Rule scan.rules :

```
alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"SCAN myscan";
flow:stateless; ack:0; flags:S; ttl:>220; reference:arachnids,439;
classtype:attempted-recon; sid:613; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 113 (msg:"SCAN ident
version request"; flow:to_server,established; content:"VERSION|0A|";
depth:16; reference:arachnids,303; classtype:attempted-recon;
sid:616; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"SCAN cybercop os
probe"; flow:stateless; dsize:0; flags:SF12;
reference:arachnids,146; classtype:attempted-recon; sid:619; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN";
flow:stateless; flags:F,12; reference:arachnids,27;
classtype:attempted-recon; sid:621; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN ipEye SYN
scan"; flow:stateless; flags:S; seq:1958810375;
reference:arachnids,236; classtype:attempted-recon; sid:622; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL";
flow:stateless; ack:0; flags:0; seq:0; reference:arachnids,4;
classtype:attempted-recon; sid:623; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN";
flow:stateless; flags:SF,12; reference:arachnids,198;
classtype:attempted-recon; sid:624; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS";
flow:stateless; flags:SRAFPU,12; reference:arachnids,144;
classtype:attempted-recon; sid:625; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap XMAS";
flow:stateless; flags:FPU,12; reference:arachnids,30;
classtype:attempted-recon; sid:1228; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN synscan
portscan"; flow:stateless; flags:SF; id:39426;
reference:arachnids,441; classtype:attempted-recon; sid:630; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN cybercop os
PA12 attempt"; flow:stateless; flags:PA12;
content:"AAAAAAAAAAAAAAAA"; depth:16; reference:arachnids,149;
classtype:attempted-recon; sid:626; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN cybercop os
SFU12 probe"; flow:stateless; ack:0; flags:SFU12;
content:"AAAAAAAAAAAAAAAA"; depth:16; reference:arachnids,150;
classtype:attempted-recon; sid:627; rev:8;)
alert udp $EXTERNAL_NET any -> $HOME_NET 10080:10081 (msg:"SCAN
Amanda client version request"; content:"Amanda"; nocase;
classtype:attempted-recon; sid:634; rev:2;)
```

```
alert udp $EXTERNAL_NET any -> $HOME_NET 49 (msg:"SCAN XTACACS
logout"; content:"|80 07 00 00 07 00 00 04 00 00 00 00 00|";
reference:arachnids,408; classtype:bad-unknown; sid:635; rev:3;)
alert udp $EXTERNAL_NET any -> $HOME_NET 7 (msg:"SCAN cybercop udp
bomb"; content:"cybercop"; reference:arachnids,363; classtype:bad-
unknown; sid:636; rev:1;)
alert udp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN Webtrends
Scanner UDP Probe"; content:"|0A|help|0A|quite|0A|";
reference:arachnids,308;
reference:url,www.netiq.com/products/vsm/default.asp;
classtype:attempted-recon; sid:637; rev:5;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"SCAN SSH Version
map attempt"; flow:to_server,established; content:"Version_Mapper";
nocase; classtype:network-scan; sid:1638; rev:5;)
alert udp $EXTERNAL_NET any -> $HOME_NET 1900 (msg:"SCAN UPnP
service discover attempt"; content:"M-SEARCH "; depth:9;
content:"ssdp|3A|discover"; classtype:network-scan; sid:1917;
rev:6;)
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SolarWinds
IP scan attempt"; icode:0; itype:8; content:"SolarWinds.Net";
classtype:network-scan; sid:1918; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"SCAN
cybercop os probe"; flow:stateless; ack:0; flags:SFP;
content:"AAAAAAAAAAAAAAAA"; depth:16; reference:arachnids,145;
classtype:attempted-recon; sid:1133; rev:12;)
```

Sebagian rule `web-misc.rules` :

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC cross site scripting attempt"; flow:to_server,established;
content:"<SCRIPT"; nocase; classtype:web-application-attack;
sid:1497; rev:7;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC cross site scripting HTML Image tag set to javascript attempt";
flow:to_server,established; content:"img src=javascript"; nocase;
reference:bugtraq,4858; reference:cve,2002-0902; classtype:web-
application-attack; sid:1667; rev:7;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC Cisco IOS HTTP configuration attempt";
flow:to_server,established; uricontent:"/level/";
pcre:"/\x2flevel\x2f\d+\x2f(exec|configure)/iU";
reference:bugtraq,2936; reference:cve,2001-0537;
reference:nessus,10700; classtype:web-application-attack; sid:1250;
rev:13;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC ftp attempt"; flow:to_server,established; content:"ftp.exe";
nocase; classtype:web-application-activity; sid:1057; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC xp_cmdshell attempt"; flow:to_server,established;
content:"xp_cmdshell"; nocase; reference:bugtraq,5309;
```

```
classtype:web-application-attack; sid:1061; rev:7;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC nc.exe attempt"; flow:to_server,established; content:"nc.exe";
nocase; classtype:web-application-activity; sid:1062; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC telnet attempt"; flow:to_server,established;
content:"telnet.exe"; nocase; classtype:web-application-activity;
sid:1066; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC net attempt"; flow:to_server,established; content:"net.exe";
nocase; classtype:web-application-activity; sid:1067; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC .htpasswd access"; flow:to_server,established;
content:".htpasswd"; nocase; classtype:web-application-attack;
sid:1071; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC queryhit.htm access"; flow:to_server,established;
uricontent:"/samples/search/queryhit.htm"; nocase;
reference:nessus,10370; classtype:web-application-activity;
sid:1077; rev:8;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC nessus 1.X 404 probe"; flow:to_server,established;
uricontent:"/nessus_is_probing_you_"; depth:32;
reference:arachnids,301; classtype:web-application-attack; sid:1102;
rev:8;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC nessus 2.x 404 probe"; flow:to_server,established;
uricontent:"/NessusTest"; nocase; reference:nessus,10386;
classtype:attempted-recon; sid:2585; rev:1;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC Netscape admin passwd"; flow:to_server,established;
uricontent:"/admin-serv/config/admpw"; nocase;
reference:bugtraq,1579; reference:nessus,10468; classtype:web-
application-attack; sid:1103; rev:11;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC BigBrother access"; flow:to_server,established;
uricontent:"/bb-hostsvc.sh?HOSTSVC"; nocase; reference:bugtraq,1455;
reference:cve,2000-0638; reference:nessus,10460;
classtype:attempted-recon; sid:1105; rev:10;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC ftp.pl attempt"; flow:to_server,established;
uricontent:"/ftp.pl?dir=../.."; nocase; reference:bugtraq,1471;
reference:cve,2000-0674; reference:nessus,10467; classtype:web-
application-attack; sid:1612; rev:8;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC ftp.pl access"; flow:to_server,established;
uricontent:"/ftp.pl"; nocase; reference:bugtraq,1471;
reference:cve,2000-0674; reference:nessus,10467; classtype:web-
application-activity; sid:1107; rev:10;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC Tomcat server snoop access"; flow:to_server,established;
uricontent:"/jsp/snp/"; uricontent:".snp"; reference:bugtraq,1532;
```

```
reference:cve,2000-0760; reference:nessus,10478;
classtype:attempted-recon; sid:1108; rev:11;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC ROXEN directory list attempt"; flow:to_server,established;
uricontent:"/%00"; reference:bugtraq,1510; reference:cve,2000-0671;
reference:nessus,10479; classtype:attempted-recon; sid:1109; rev:9;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC apache source.asp file access"; flow:to_server,established;
uricontent:"/site/eg/source.asp"; nocase; reference:bugtraq,1457;
reference:cve,2000-0628; reference:nessus,10480;
classtype:attempted-recon; sid:1110; rev:9;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC Tomcat server exploit access"; flow:to_server,established;
uricontent:"/contextAdmin/contextAdmin.html"; nocase;
reference:bugtraq,1548; reference:cve,2000-0672;
reference:nessus,10477; classtype:attempted-recon; sid:1111;
rev:10;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC http directory traversal"; flow:to_server,established;
content:"..|5C|"; reference:arachnids,298; classtype:attempted-
recon; sid:1112; rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC mlog.phtml access"; flow:to_server,established;
uricontent:"/mlog.phtml"; nocase; reference:bugtraq,713;
reference:cve,1999-0068; reference:cve,1999-0346;
classtype:attempted-recon; sid:1119; rev:7;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC mylog.phtml access"; flow:to_server,established;
uricontent:"/mylog.phtml"; nocase; reference:bugtraq,713;
reference:cve,1999-0068; reference:cve,1999-0346;
classtype:attempted-recon; sid:1120; rev:8;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC /etc/passwd"; flow:to_server,established;
content:"/etc/passwd"; nocase; classtype:attempted-recon; sid:1122;
rev:5;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC ?PageServices access"; flow:to_server,established;
uricontent:"?PageServices"; nocase; reference:bugtraq,1063;
reference:bugtraq,7621; reference:cve,1999-0269;
classtype:attempted-recon; sid:1123; rev:9;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC .htaccess access"; flow:to_server,established;
uricontent:".htaccess"; nocase; classtype:attempted-recon; sid:1129;
rev:6;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC /.... access"; flow:to_server,established; content:"/....";
classtype:attempted-recon; sid:1142; rev:5;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
MISC /~root access"; flow:to_server,established;
uricontent:"/~root"; nocase; classtype:attempted-recon; sid:1145;
rev:7;)
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-
```

```
MISC /~ftp access"; flow:to_server,established; uricontent:"/~ftp";
nocase; classtype:attempted-recon; sid:1662; rev:5;)
```

4.2.6 Membangun Remote Logging

Remote logging digunakan sebagai server log bagi alert yang dihasilkan oleh WRT54G. Log yang dihasilkan perlu disimpan untuk keperluan analisis bagi administrator. Log ini dikirim ke komputer lain. dikarenakan keterbatasan memori dari WRT54G dan akan terhapusnya file log jika WRT54G direstart.

Untuk mengirimkan log ke *remote* maka tambahkan parameter sebagai berikut:

1. Pada NVRAM

```
nvram set log_ipaddr=192.168.2.5                #set alamat IPserver log
```

2. Pastikan output alert pada `/etc/snort/snort.conf` ke syslog

```
output alert_syslog: LOG_AUTH LOG_ALERT #keluaran aler ke syslog
```

3. Konfigurasi file `/etc/sysconfig/syslog` pada server log untuk menerima file log

```
# Options to syslogd
# -m 0 disables 'MARK' messages.
# -r enables logging from remote machines
# -x disables DNS lookups on messages recieved with -r
# See syslogd(8) for more details
SYSLOGD_OPTIONS="-r"
# Options to klogd
# -2 prints all kernel oops messages twice; once for klogd to
decode, and
#    once for processing with 'ksymoops'
# -x disables all klogd processing of oops messages entirely
# See klogd(8) for more details
KLOGD_OPTIONS="-x"
```

4.2.7 Mengolah Data Log

Data Log akan diolah pada server log setelah menerima alert yang dihasilkan oleh WRT54G. Ketika dikirim ke *remote logging*, data log tersimpan dalam `/var/log/messages`. Untuk itu harus dipisahkan mana log yang dihasilkan oleh Snort dan disimpan dengan `hasil.log`. Kemudian *`hasil.log`* diubah dengan nama `YYYYMMDD.log`. File log tersebut akan dikirim ke folder `/var/www/html/log`. Selain itu log Snort yang terdapat pada `/var/log/messages` juga diubah dengan bantuan aplikasi Snortalog menjadi file html yang akan memudahkan administrator membaca aktifitas yang dilakukan oleh *intruder*. File `YYYYMMDD.html` yang telah dihasilkan juga dikirim folder `/var/www/html/log` untuk dibaca melalui browser.

Installasi Snortalog pada server log.

```
tar -xzvf Snortalog_v2.4.2.tgz
```

Memisahkan data log Snort dari `/var/log/messages` berdasarkan tanggal saat ini dengan menggunakan script bash `tugas.sh` kemudian menyimpan dalam `hasil.log`

```
#! /bin/bash

cat /var/log/messages | grep `date +"%b %d"` | grep snort >
/root/hasil.log
```

Menghasilkan file html dari syslog

```
cat /root/hasil.log | ./snortalog.pl -o hasil.html -report -2
```

Secara default vlan dan et(h)1 di *bridge* pada *interface* br0. Hal ini membuat LAN dan WLAN memiliki jaringan yang sama. Perintah `nvram unset lan_ifnames` digunakan untuk menghilangkan *bridging* pada *interface-interface* tersebut sehingga jaringan LAN dan WLAN akan terpisah.

Konfigurasi selanjutnya menunjukkan bahwa port 5 digunakan oleh semua vlan sebagai port yang terhubung pada prosessor WRT54G. Tanda asterisk pada port 5 menandakan traffic dari vlan akan melalui port ini untuk sampai ke *interface* sebenarnya dari WRT54G yaitu eth0 (Lihat Gambar 3.7 dan Gambar 3.8).

a. Konfigurasi VLAN0 yang mengacu pada LAN

```
nvram set lan_ifname=vlan0          # set interface lan dengan vlan0
nvram set lan_proto=static          # gunakan static IP
nvram set lan_ipaddr=192.168.1.1    # isi alamat IP interface vlan0
nvram set lan_netmask=255.255.255.0 # isi netmask lan
nvram commit
```

Vlan0 digunakan sebagai *interface* pada lan yang menggunakan metode static u

b. Konfigurasi VLAN2 yang mengacu pada DMZ

```
nvram set dmz_ifname=vlan2          # set interface lan dengan vlan2
nvram set dmz_proto=static          # gunakan static IP
nvram set dmz_ipaddr=192.168.2.1    # isi alamat IP interface vlan2
nvram set dmz_netmask=255.255.255.0 # isi netmask dmz
nvram commit
```

c. Konfigurasi eth1 yang mengacu pada WLAN

```
nvram set wifi_ifname=eth1          # set interface wifi dengan eth1
nvram set wifi_proto=static         # gunakan static IP
nvram set wifi_ipaddr=192.168.3.1   # isi alamat IP interface eth1
nvram set wifi_netmask=255.255.255.0 # isi netmask wifi
nvram commit
```

d. Konfigurasi vlan1 yang mengacu pada WAN

```
nvram set wan_ifname=vlan1          # set interface wan dengan vlan1
nvram set wan_proto=static          # gunakan static IP
nvram set wan_ipaddr=192.168.10.234# isi alamat IP interface vlan1
nvram set wan_gateway=192.168.10.1 # isi alamat IP gw wan
nvram set wan_dns=192.168.15.1      # isi alamat IP dns wan
nvram set wan_netmask=255.255.255.0# isi netmask wan
nvram commit
```

Variabel `*_proto` digunakan untuk mengkonfigurasi bagaimana inteface mendapatkan alamat IP, jika diisi `static` maka `*_ipaddr`, `*_netmask`, `*_gateway`, `*_dns` diisikan dengan alamat yang sudah pasti. Tetapi jika diisi dengan metode `dhcp` maka alamat IP akan diberikan secara otomatis.

4.2.3 Installasi nas dan parameter wifi

Nas adalah paket tambahan yang diinstall untuk kepentingan enkripsi dinamis (*Wireless Encryption Protocl* dan *Wi-Fi Protected Access*) pada jaringan *wireless*. Installasi paket ini menggunakan *tool* yang telah tersedia pada OpenWRT yaitu dengan *ipkg*. Ipkg akan langsung mencari dimana paket nas berdasar alamat yang terdapat pada konfigurasi ipkg.

```
ipkg install nas
```

Konfigurasi parameter wifi untuk otentikasi WPA2 PSK

```
nvram set wl0_ifname=eth1           # set interface wl0 dengan eth1
nvram set wl0_ssid=Juragan Wifi     # set SSID
nvram set wl0_mode=ap               # set mode router sebagai ap
nvram set wl0_infra=1               # 1 mode infrastruktur , 0 untuk ad-hoc
nvram set wl0_closed=1              # 1 untuk hide SSID, 0 untuk broadcast
nvram set wl0_radio=1               #  enable/disable radio (1=enable)
```

```
nvram set wl0_akm=psk2                  # set jenis enkripsi
nvram set wl0_wpa_psk=rahasia12345      # isikan WPA pre-shared key
nvram set wl0_crypto=aes                # Jenis cryptography
nvram commit
```

Meskipun variable `wifi_*` dapat digunakan untuk mengkonfigurasikan IP jaringan yang terpisah dari *interface wireless*, standar konfigurasi `wifi_*` digunakan untuk bridging *interface wireless* pada `lan_ifnames` (biasanya digunakan pada mode ad-hoc).

Konfigurasi parameter wifi diatas menggunakan variabel berbeda yang disebut wl0_*. Variabel ini akan mengkonfigurasikan karakteristik dari *interface* fisik *wireless* dan tetap dapat digunakan tanpa memperhatikan apakah *interface* wifi dibridge atau menggunakan IP jaringan yang terpisah.

Variabel `wl0_mode` dapat diisikan dengan `ap` untuk mode Access Point, `sta` untuk mode *station* dan `wet` untuk mode *wireless ethernet bridge*. Variable `wl0_akm` dapat diisikan dengan `open` untuk tidak menggunakan WPA, `psk` untuk WPA Personal/PSK, `wpa` untuk WPA dengan server radius, `psk2` untuk WPA2 PSK, `wpa2` untuk WPA2 dengan radius, `psk psk2` atau `wpa wp2` untuk penggunaan baik WPA maupun WPA2. Variable `wl0_ crypto` dapat diisikan dengan `tkip` untuk ekripsi menggunakan RC4, `aes` untuk enkripsi menggunakan AES, `aes+tkip` untuk dukungan bagi dua enkripsi tersebut.

4.2.4 Mengkonfigurasi Firewall dan alokasi IP secara dinamis

WRT54G harus menjalankan firewall, untuk mencegah akses yang tidak diijinkan sistem dan meningkatkan level keamanan router. Firewall yang diterapkan menggunakan skrip iptables. Skrip tersebut diletakkan pada path `/etc/init.d/S35firewall`. Berikut skrip firewall iptables yang digunakan :

```
#!/bin/sh

## Please make changes in /etc/firewall.user

. /etc/functions.sh

#ambil isi variabel dari nvram

WAN="$(nvram get wan_ifname)"
WANDEV="$(nvram get wan_device)"
LAN="$(nvram get lan_ifname)"
WIFI="$(nvram get wifi_ifname)"
DMZ="$(nvram get dmz_ifname)"

## CLEAR TABLES

for T in filter nat; do
  iptables -t $T -F
  iptables -t $T -X
done

iptables -N input_rule
iptables -N output_rule
iptables -N forwarding_rule

iptables -t nat -N prerouting_rule
iptables -t nat -N postrouting_rule

iptables -N LAN_ACCEPT
[ -z "$WAN" ] || iptables -A LAN_ACCEPT -i "$WAN" -j RETURN
[ -z "$WANDEV" -o "$WANDEV" = "$WAN" ] || iptables -A LAN_ACCEPT -i
"$WANDEV" -j RETURN
iptables -A LAN_ACCEPT -j ACCEPT
```

Merubah file log `hasil.log` menjadi file dengan format YYYYMMDD.log dan script python *log.py* sederhana, yaitu :

```
from datetime import datetime as waktu

timeisnow = waktu.now()
timeistuple = timeisnow.timetuple()

tahun   = str(timeistuple[0])
bulan   = str(timeistuple[1])
tanggal = str(timeistuple[2])

namalog = tahun + bulan + tanggal + ".log"

bukaloglama = open("hasil.log","rb")
bacaloglama = bukaloglama.read()
bukaloglama.close()

filelog = open(namalog,"w")
filelog.write(bacaloglama)
filelog.close()
```

Untuk merubah file log `hasil.html` menjadi file dengan format `YYYYMMDD.html` mengunakan script `loghtml.py` yang sama dengan `log.py`, perbedaan hanya pada namalog :

```
.
.
namalog = tahun + bulan + tanggal + ".html"

bukaloglama = open("hasil.html","rb")
.
.
```

Script tersebut dijalankan dengan mengetikkan perintah

```
python log.py   # python loghtml.py untuk file html
```

Kemudian pindahkan file-file tersebut ke folder web server pada `/var/www/html/log`. Untuk otomatisasi pengolahan log setiap hari digunakan file `/etc/crontab`.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/root

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly

55 23 * * * root sh /root/tugas.sh
55 23 * * * root cat /root/hasil.log | ./snortalog.pl -o hasil.html
-report -2
56 23 * * * root python /root/loghtml.py
56 23 * * * root python /root/log.py
57 23 * * * root cp 2* /var/www/html/log
```

Pengolahan log akan dilakukan pada pukul 23:55 sampai 23:57 setiap hari.

4.3 Pengujian Sistem

Pengujian sistem bertujuan untuk menganalisis kinerja sistem sebelum sistem tersebut dapat diaplikasikan. Dari hasil pengujian ini akan diketahui apakah sistem dapat bekerja dengan baik dan berjalan sesuai dengan kebutuhan atau tidak.

Pengujian juga dimaksudkan untuk mencari kekurangan-kekurangan yang terdapat dalam sistem untuk kemudian diperbaiki sehingga kesalahan pada sistem dapat diminimalisasi atau bahkan dihilangkan. Kekurangan-kekurangan yang ada akan menjadi masukan untuk kemudian diterapkan pada implementasi program selanjutnya.

Pengujian dilakukan pada dibeberapa kinerja sebagai berikut:

1. Penggunaan aturan snort.conf.

2. Otentikasi *user*

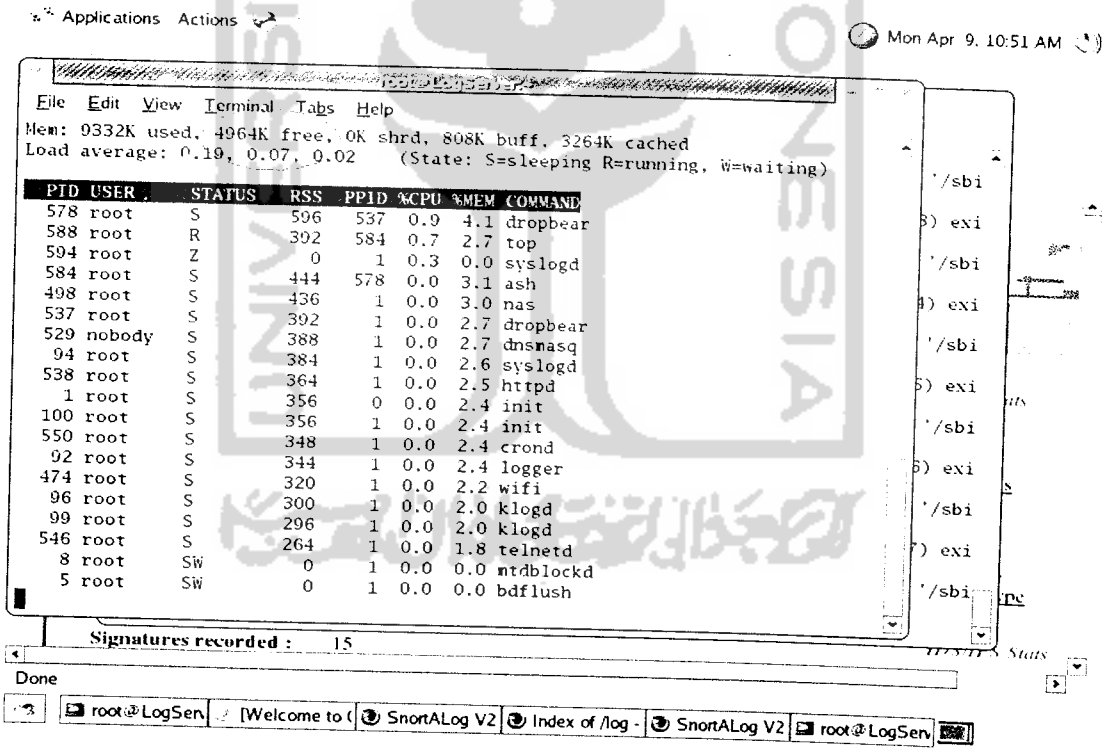
3. Intrusi *User*.

4.3.1 Penggunaan Aturan snort.conf

Pengujian ini dilakukan pada router untuk mengetahui seberapa banyak aturan yang dapat digunakan oleh Snort disesuaikan dengan ketersediaan memori pada WRT54G.

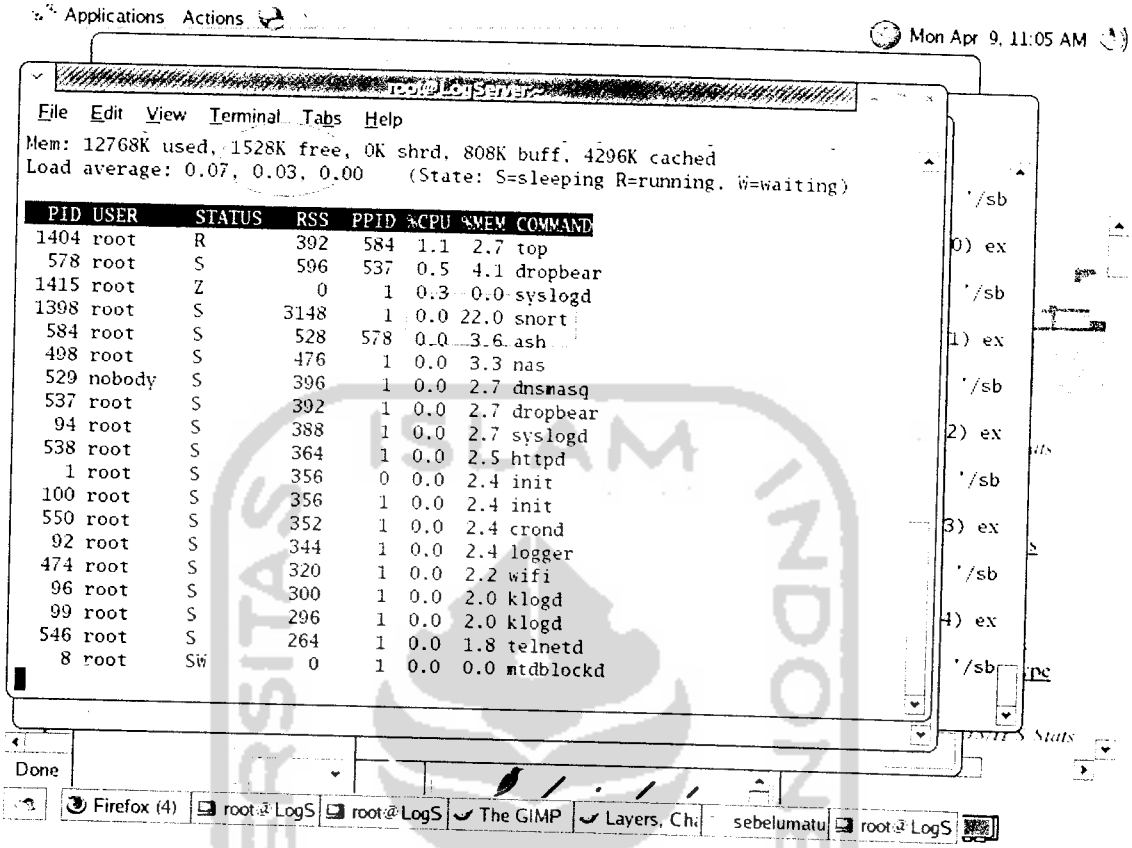
Ujicoba dilakukan dengan mencoba beberapa aturan kemudian melihat seberapa banyak konsumsi sumber daya oleh Snort.

Ketersediaan memori sebelum Snort dijalankan ditunjukkan oleh Gambar 4.3.



Gambar 4.3. Memori sebelum Snort dijalankan

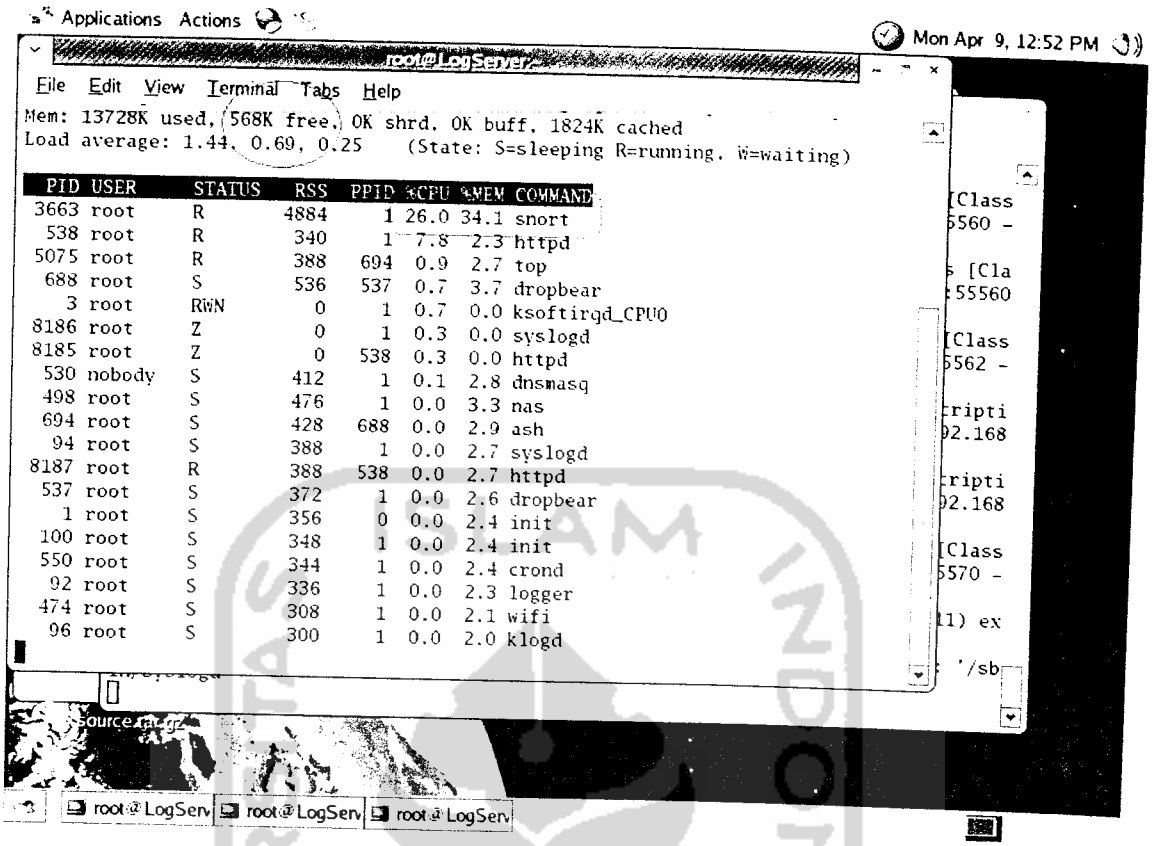
Ketersedian memori setelah Snort dijalankan ditunjukkan oleh Gambar 4.4.



Gambar 4.4. Memori setelah Snort dijalankan

Pada Gambar 4.4 terlihat bahwa Snort mengkonsumsi 22% dari memori yang ada, sehingga ketersediaan memori menjadi 1528Kb.

Ketika Snort dijalankan dan *user* melakukan intrusi kedalam sistem penggunaan sumberdaya oleh Snort menjadi meningkat seperti ditunjukkan oleh Gambar 4.5.



Gambar 4.5. Sumber daya terpakai saat intrusi

Pada saat intrusi dilakukan memori tersisa sebesar 568Kb dan Snort menggunakan 34.1 % memori dan 26% kemampuan prosessor.

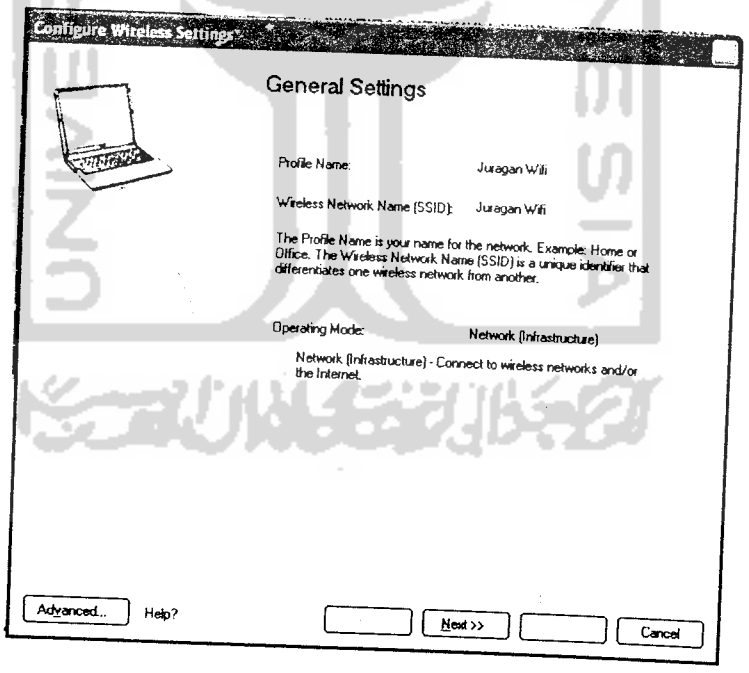
4.3.2 Otentikasi user

Pengujian otentikasi *user* ke router WRT54G.

4.3.2.1 Prosedur Normal

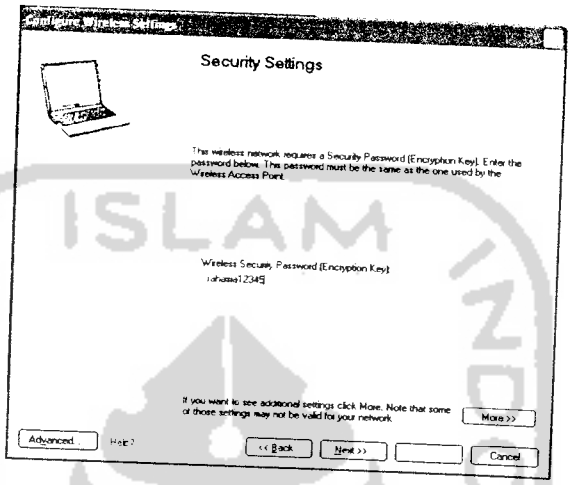
Jika *user* mengkonfigurasi *interface wireless* pada komputer secara benar, kemudian mengisikan SSID serta kunci enkripsi yang sesuai maka *interface wireless* komputer *user* akan terhubung dengan Access Point dan mendapatkan Alamat IP secara otomatis yang diberikan oleh WRT54G.

Tahap pertama dilakukan dengan mengisikan *profile name* dan SSID pada komputer *client*. Tidak terdapat aturan pada pengisian *profile name* sehingga *user* dibebaskan untuk mengisi sesuai dengan keinginannya. Pada pengisian SSID, *user* harus mengisikan sesuai dengan SSID router. SSID ini bersifat *case sensitive*, sehingga kesalahan pengisian mengakibatkan *client* tidak dapat menggunakan sember daya sistem. Proses ini ditunjukkan pada Gambar 4.6.



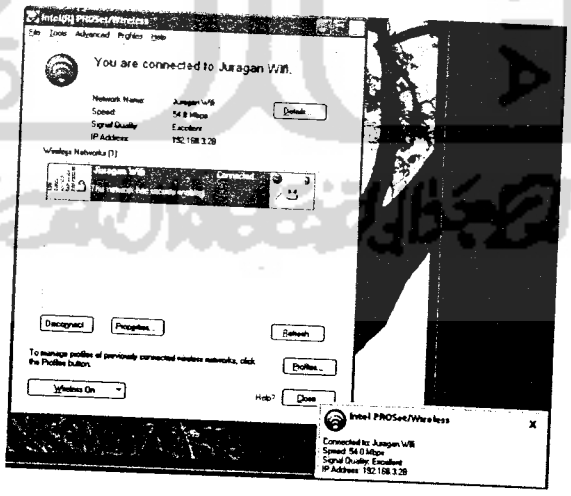
Gambar 4.6. Pengisian SSID

Setelah SSID diisikan, dilanjutkan dengan pengisian kunci enkripsi. Kunci enkripsi yang dipakai dalam sistem ini adalah WPA2-PSK. Seperti SSID, kunci enkripsi juga *case sensitive* dan harus sama dengan kunci enkripsi yang terdapat pada router. Proses ini ditunjukkan pada Gambar 4.7.



Gambar 4.7. Pengisian WPA2-PSK

Jika SSID dan ekipsi sesuai dengan konfigurasi sistem, maka *client* akan mendapatkan alamat IP secara otomatis. Hal ini ditunjukkan pada Gambar 4.8.



Gambar 4.8. Komputer user yang terotentikasi

4.3.2.2 Prosedur tidak normal

Jika *user* tidak mengkonfigurasi *interface wireless* pada komputer secara benar, tidak mengisikan SSID dan kunci enkripsi yang sesuai dengan konfigurasi router maka *interface wireless* komputer *user* tidak terhubung dengan Access Point. Keadaan semacam ini adalah salah satu penyebab proses autentikasi tidak berjalan.

4.3.3 Intrusi User

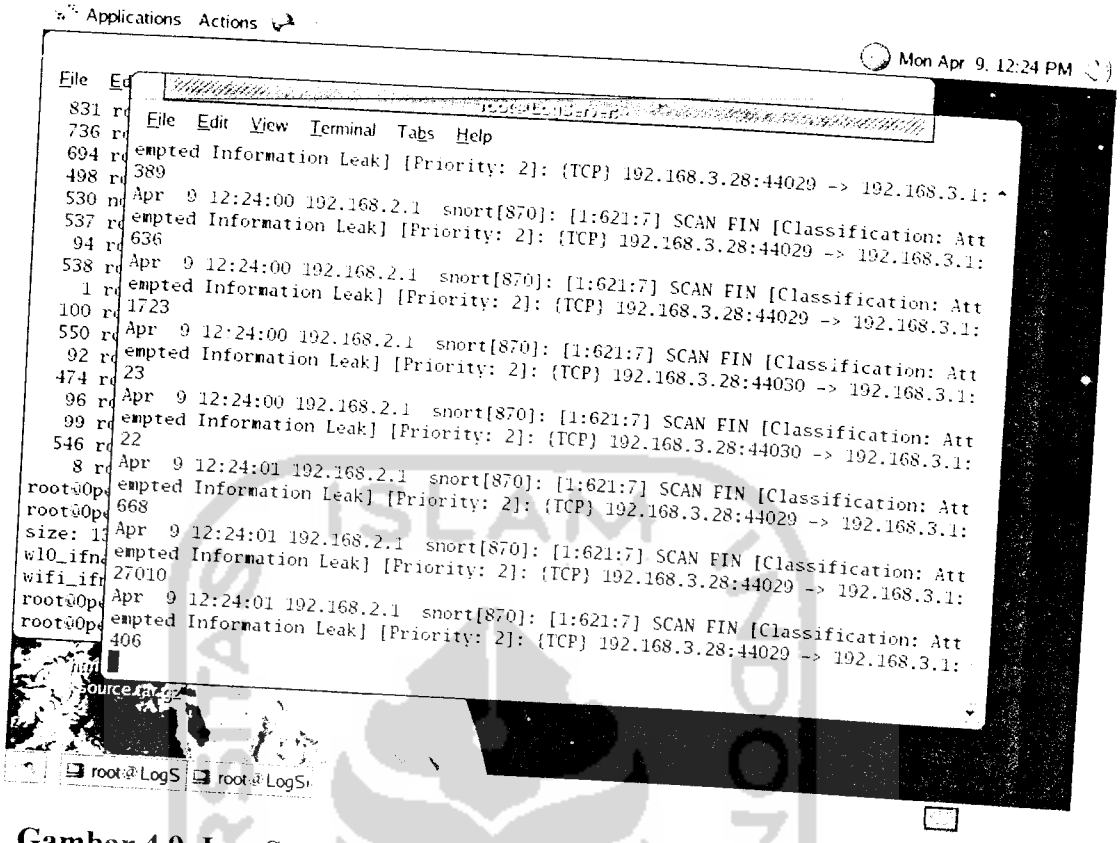
Intrusi dilakukan untuk membuktikan bahwa sistem berjalan sesuai dengan yang diinginkan. Intrusi yang diujikan menggunakan *nmap* untuk melakukan *scanning* terhadap port yang terbuka dan menggunakan *nikto* untuk verifikasi terhadap layanan pada port 80.

4.3.3.1 Pengujian dengan nmap

Pada sisi *user* diinputkan perintah :

```
nmap -sF 192.168.3.1                    # scan FIN
nmap -sX 192.168.3.1                    # scan Xmas
```

Saat *user* melakukan *port scanning* maka akan terlihat alert seperti ditunjukkan pada Gambar 4.9.



Gambar 4.9. Log Snort dari port scanning dengan mengirimkan paket FIN

Log diatas menunjukkan bahwa pada tanggal 9 April terdapat aktifitas dari komputer beralamat 192.168.3.28 yang melakukan *scanning* dengan mengirimkan paket FIN melalui port 44209 untuk mengekplorasi *port* mana yang terbuka pada WRT54G yang beralamat 192.168.3.1. Paket tersebut diterima oleh WRT54G melalui *port* 406. Hal ini terjadi karena IDS merespon aturan pada `scan.rule` yaitu pada baris yang berisi :

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN";
      flow:stateless; flags:F,12; reference:arachnids,27;
      classtype:attempted-recon; sid:621; rev:8;)
```

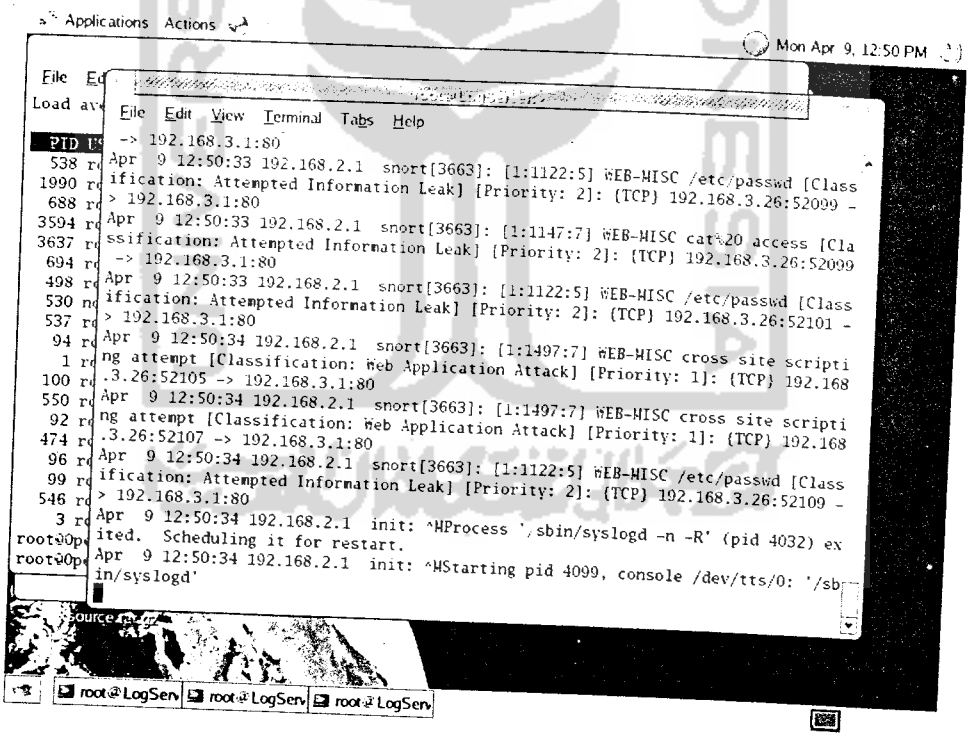
Aturan diatas menunjukkan bahwa IDS akan memberikan *alert* pada sistem jika terdapat permintaan pada protokol tcp dari jaringan diluar ke sistem yang berlamat di 192.168.3.1. Sesuai dengan signature pada aturan tersebut, aktifitas itu dikenali sebagai intrusi dengan mengirimkan paket FIN.

4.3.3.2 Pengujian menggunakan nikto

Pada sisi *user* dinputkan perintah :

```
perl nikto.pl –h 192.168.3.1
```

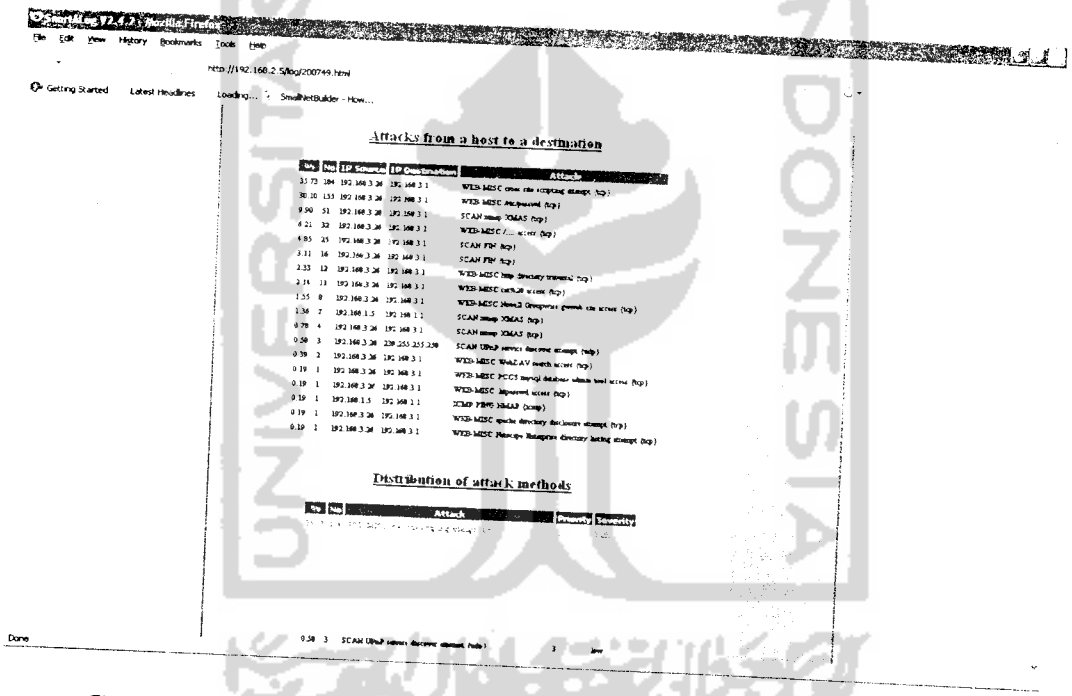
Saat *user* menjalankan perintah diatas, Snort merespon dengan menghasilkan alert seperti terlihat pada Gambar 4.10.



Gambar 4.10. Sebagian Log Snort dari alert yang dihasilkan nikto

Log diatas menunjukkan bahwa pada tanggal 9 April terdapat aktifitas dari komputer beralamat 192.168.3.26 yang melakukan *scanning* dengan memeriksa kelemahan pada file `/etc/passwd` yang terdapat pada *web server* dan kemungkinan file-file yang dapat mengakibatkan terjadinya *cross site scripting*. Alert tersebut muncul karena *signature* intrusi sesuai dengan aturan pada `web-misc.rule`.

Hasil dari instrusi diatas disimpan dalam syslog. Kemudian log seperti pada implementasi sebelumnya diolah untuk memudahkan analisis oleh administrator. Sebagian tampilan browser ditunjukkan oleh Gambar 4.11.



Gambar 4.11. Tampilan browser hasil pengolahan log oleh Snortalog

Pada sebagain tampilan browser diatas dapat diketahui alamat IP yang melakukan intrusi, alamat tujuan dan jenis serangan yang terjadi serta tingkatan serangannya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari implementasi yang telah dilakukan dapat ditarik kesimpulan sebagai berikut :

1. Sistem dapat melakukan otentikasi berbasis WPA-PSK2
2. Snort yang digunakan sebagai IDS berfungsi sebagai alarm yang akan mengingatkan administrator terhadap intrusi user.
3. Snort digunakan sebagai alat untuk mendapatkan informasi tentang apa yang terjadi sehingga administrator dapat mengambil tindakan secepat mungkin terhadap aktifitas yang terjadi dalam jaringan.
4. Tidak semua aturan Snort dapat diimplementasikan karena keterbatasan memori.
5. Pengelolaan log dalam bentuk file html sangat memudahkan pembacaan log dibandingkan dengan file log konvensional untuk memudahkan administrator menganalisis sistem.

5.2 Saran

Beberapa saran untuk pengembangan dan penelitian selanjutnya sebagai berikut :

1. Dilakukan *filtering* alamat MAC sehingga alamat IP secara otomatis diberikan pada MAC yang telah tercatat. Hal ini mencegah penggunaan sumber daya oleh user yang tidak tercatat.
2. Mengimplementasikan sistem yang dibangun pada *wireless* router yang memiliki kapasitas memori lebih besar seperti WRT54GS yang memiliki memori sebesar 32MB
3. Pencatatan log yang disimpan dalam database My SQL atau PostgreSQL, untuk memudahkan analisis yang lebih dinamis.
4. Kemampuan OpenWRT sebagai sistem operasi pada WRT54G belum dieksplorasi sepenuhnya, diharapkan pada penelitian selanjutnya kemampuan OpenWRT dapat dieksplorasi, seperti penggunaan WRT54G sebagai alat VPN (*Virtual Private Network)*, penggunaan *printer sharing* dalam jaringan, fitur *QoS* pada router, implementasi Asterisk sebagai untuk VOIP dan sebagainya.
5. Integrasi sistem saat ini dengan otentikasi berbasis *Captive Portal* yang akan menghasilkan sistem dengan level keamanan yang sangat baik. Hal ini disebabkan permintaan layanan HTTP akan di redirect ke Captive Portal terlebih dahulu

DAFTAR PUSTAKA

[ANO01] Anonim. *CCNA 1: Networking Basics v3.1.1.*

[ANO02] Anonim. *Nmap - Free Security Scanner For Network Exploration & Security Audist,* http://insecure.org/nmap/ diakses tanggal 4 Maret 2007

[ANO03] Anonim. *What is a Wireless Network's SSID?,* http://kbserver.netgear.com/kb_web_files/N100683.asp diakses tanggal 19 Maret 2007.

[ANO04] Anonim. *Disable SSID Broadcasting To Protect Your Wireless Network,* http://netsecurity.about.com/od/quicktip1/qt/qtwifinossid.htm diakses tanggal 19 Maret 2007.

[ANO05] Anonim. *Common Wireless seCurity Questions and ansWers Questions about standards: WeP WPa and* WPa2, http://eu.computers.toshiba-europe.com/Contents/Toshiba_teg/EU/GENERIC/files/C_2005_Common_wireless_Security_QnAs_EN.pdf diakses tanggal 22 Maret 2007.

[ANO06] Anonim. *Apache Webserver Security testing Using nikto,* http://www.debianhelp.co.uk/nikto.htm diakses tanggal 4 Maret 2007.

[ANO07] Anonim. *OpenWRT Documentation,* http://wiki.openwrt.org diakses tanggal 28 November 2007.

[ANO08] Anonim. *OpenWRT Forum,* http://forum.openwrt.org diakses tanggal 28 November 2007.

[BAK07] Baker, M. What is OpenWRT, http://openwrt.org diakses tanggal 30 Maret 2007

[FIR06] Firmawan, A. *Perancangan Upgrading Sistem MMI Berdadasrkan API RP 554 pada Pagar Gas Plant*, Yogyakarta, 2006.

[GUN06] Gunadi. *Teknologi Wireless LAN dan Aplikasinya.* Jakarta : Elex Media Komputindo, 2006.

[HAS07] Haskins, R. *Embedded Hardware*, https://db.usenix.org/publications/login/2005-10/pdfs/haskins.pdf diakses tanggal 30 Maret 2007

[JAV07] *WLAN: Wireless LAN by IEEE 802.11, 802.11a, 802.11b, 802.11g, 802.11n* http://www.javvin.com/protocolWLAN.html diakses tanggal 16 Maret 2007.

[MAR06] Martin, C. *Installing OpenWRT RC5 on a Liksys WRT54GSv1.1*, Ver 1.5, http://www.martin.ee/OpenWrt/OpenWrt%20Config_RC5.pdf diakses tanggal 12 Februari 2006.

[PRA01] Mateti, P. *Port Scanning*, http://www.cs.wright.edu/~pmateti/Courses/499/Probing diakes tanggal 4 Maret 2007.

[PUR98] Purbo, O.W., et. al. *TCP/IP Standar, Desain, danIimplementasi*. Jakarta : Elex Media Komputindo, 1998.

[RAF03] Raffeequrrehman. *Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID*, New jersey : Prentice hall, 2003.

[STE96] Stevens, W.R. *TCP/IP Illustrated*, Vol 1. Massachusetts : Addison-Wesley Publshing Company, 1996.

[WIK07] *IEEE 802.11i*, http://en.wikipedia.org/wiki/IEEE_802.11 diakses tanggal 22 Maret 2007.

[YAG03] Yaghmour, K. Building Embedded Linux Systems. California : O'Reilly & Associates, 2003.

[Y3D07] Y3dips, *Virtual Local Area Network*, http://ezine.echo.or.id diakses tanggal 24 Maret 2007.