

**PERANCANGAN IMPLEMENTASI
SISTEM ACCOUNTING USER JARINGAN KOMPUTER SECARA TERPUSAT
MENGGUNAKAN PROTOKOL RADIUS**

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh :

HANDOKO DIAN FATIKNO

NIM 0410630046

DEPARTEMEN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2010

PERANCANGAN IMPLEMENTASI
SISTEM ACCOUNTING USER JARINGAN KOMPUTER SECARA TERPUSAT
MENGGUNAKAN PROTOKOL RADIUS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik



Disusun oleh :

HANDOKO DIAN FATIKNO

NIM 0410630046

DOSEN PEMBIMBING:

Ir. Heru Nurwarsito, M.Kom
NIP 19650402 199002 1 001

R. Arief Setyawan, ST., MT
NIP 19750819 19903 1 001

**PERANCANGAN IMPLEMENTASI SISTEM ACCOUNTING USER
JARINGAN KOMPUTER SECARA TERPUSAT MENGGUNAKAN
PROTOKOL RADIUS**

Disusun oleh :

HANDOKO DIAN FATIKNO
NIM. 0410630046-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal **28 JANUARI 2010**

Majelis Penguji :

Waru Djuriatno, ST., MT.
NIP. 19690725 199702 1 001

Ir. Muhammad Aswin
NIP. 19640626 199002 1 001

Adharul Muttaqin, ST., MT.
NIP. 19760121 200501 1 001

Mengetahui :
Ketua Jurusan Teknik Elektro

Rudy Yuwono, ST., M.Sc.
NIP. 19710615 199802 1 003

PENGANTAR

Puji syukur ke hadirat Allah SWT, atas nikmat, hidayah serta kasih sayang-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul "**Perancangan Implementasi Sistem Accounting User Jaringan Komputer Secara Terpusat Menggunakan Protokol RADIUS**". Hanya kepada-Nya kita menyembah dan memohon. Teriring doa keselamatan untuk Rasulullah Muhammad SAW, keluarga, sahabat serta seluruh ummatnya. Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Program Studi Sistem Informatika dan Komputer Fakultas Teknik Universitas Brawijaya Malang.

Tidak banyak yang bisa penulis sampaikan kecuali ungkapan terima kasih kepada berbagai pihak yang telah dengan tulus ikhlas memberikan bimbingan, arahan, dan dukungan hingga penulisan tugas akhir ini dapat terselesaikan. Pada kesempatan kali ini, dengan segala kesungguhan dan rasa rendah hati, penulis mengucapkan banyak terima kasih kepada:

1. Bapak Rudy Yuwono, ST., M.Sc. dan Bapak M. Azis Muslim, ST., MT., Ph.D. selaku Ketua dan Sekretaris Jurusan Teknik Elektro serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya;
2. Bapak Ir. Heru Nurwarsito, M.Kom. selaku Dosen Pembimbing I yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan tugas akhir ini;
3. Bapak Raden Arif Setyawan ST., MT. selaku Dosen Pembimbing II yang telah banyak memberikan bimbingan, masukan dan arahan dalam penyusunan tugas akhir ini;
4. Kedua orang tua dan kedua kakakku yang sedikitpun tidak pernah lengah dari do'a, dukungan dan harapan untuk terselesaiannya tugas akhir ini.

5. Winda Septarini yang tiada henti memberikan semangat, do'a, dukungan dan harapan untuk terselesaikannya tugas akhir ini.
6. Semua teman-teman di PPTI-UB yang telah memberikan dukungan baik semangat maupun sarana untuk menyelesaikan tugas akhir ini.
7. Semua teman-teman di TPTIFT yang telah memberikan dukungan, memberikan dukungan baik semangat maupun sarana untuk menyelesaikan tugas akhir ini.
8. Semua teman-teman di RistIE UB yang telah memberikan dukungan, memberikan dukungan baik semangat maupun sarana untuk menyelesaikan tugas akhir ini.
9. Mahasiswa Jurusan Teknik Elektro Universitas Brawijaya khususnya teman-teman angkatan 2004 "Generator" yang telah saling mendukung satu sama lain agar tetap kompak dan saling membantu dalam penyelesaian tugas akhir ini.
10. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat.

Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 17 Januari 2010

Penulis

ABSTRAK

Handoko D F. 2009. : Perancangan Implementasi Sistem Accounting User Jaringan Komputer Secara Terpusat Menggunakan Protokol RADIUS. Skripsi Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing: Ir. Heru Nurwarsito, M.Kom. dan Raden Arif Setyawan ST., MT.

Pada saat ini, terdapat dua macam tipe koneksi di lokal jaringan komputer (Local Area Network [LAN]) Universitas Brawijaya yaitu melalui kabel (wired) dan tanpa kabel (wireless). Namun kita tidak dapat mengetahui dengan pasti siapa user yang sedang melakukan koneksi ke jaringan komputer Universitas Brawijaya. Hal ini dapat mengancam keamanan jaringan komputer di Universitas Brawijaya karena user yang melakukan koneksi tidak tercatat serta tidak mudah melacak dan mengetahui siapa pelakunya jika terjadi suatu serangan. Berdasarkan kebutuhan keamanan tersebut, maka dibutuhkan suatu sistem keamanan AAA untuk menjamin keamanan dan kenyamanan user jaringan komputer di Universitas Brawijaya karena dengan sistem ini kita dapat mengetahui dengan pasti siapa user yang sedang melakukan koneksi. Untuk memudahkan agar user dapat berpindah dari network yang satu ke network yang lainnya, maka dibutuhkan suatu sistem yang terpusat. Oleh karena itu pada tugas akhir ini menggunakan protokol RADIUS dan menitik beratkan pada proses accounting. Hasil accounting pada tugas akhir ini diolah dan dapat digunakan untuk keperluan tracking (siapa, kapan dan dimana permintaan berasal) jika terjadi suatu tindak kejahatan yang menggunakan akses jaringan komputer internal Universitas Brawijaya, proses penghitungan biaya akses user jika menggunakan akses Internet, serta sebagai bahan untuk perencanaan dan alokasi sumber daya berupa bandwidth pada waktu mendatang.

Perancangan dan implementasi sistem accounting user jaringan komputer secara terpusat menggunakan protokol radius dilakukan dengan menggunakan sistem Operasi Linux, serta perangkat lunak freeradius, MySQL-server, chillispot sebagai NAS, dan easyhotspot. Perancangan sistem accounting user jaringan komputer secara terpusat menggunakan protokol radius ini dibagi menjadi dua tahap utama, yaitu perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras meliputi perancangan topologi prototipe jaringan komputer Universitas Brawijaya. Perancangan perangkat lunak meliputi perancangan jaringan komputer secara logical, perancangan server RADIUS menggunakan freeradius, perancangan server basis data untuk menyimpan data accounting, perancangan NAS sebagai client RADIUS.

Dari hasil pengujian dan sistem accounting user jaringan komputer secara terpusat menggunakan protokol radius, dapat disimpulkan bahwa proses yang terjadi sistem ini sesuai dengan teori yang ada pada dasar teori. Pelacakan (*tracking*) siapa dan dimana tempat *user* yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya dilakukan dengan syarat alamat IP dan waktu kejadian diketahui. Besar nilai *throughput* untuk satu *user* sesudah diterapkan sistem hampir sama dengan besarnya nilai *throughput* sebelum diterapkan sistem.

Kata Kunci: Keamanan Jaringan Internet, RADIUS, accounting, terpusat

DAFTAR ISI

PENGANTAR	i
ABSTRAK.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vii
DAFTAR TABEL.....	ix
DAFTAR LAMPIRAN.....	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan.....	4
2BAB II TINJAUAN PUSTAKA	6
2.1 Konsep Jaringan Komputer.....	6
2.1.1 Protokol Komunikasi	6
2.1.2 Topologi Jaringan Komputer	9
2.2 Mekanisme AAA (Authentication, Authorization, Accounting)	14
2.2.1 Otentikasi (Authentication).....	14
2.2.2 Otorisasi (Authorization)	14
2.2.3 Pencatatan (<i>Accounting</i>)	15
2.3 Remote Authentication Dial-In User Service (RADIUS).....	15
2.3.1 Format Paket Data RADIUS.....	15
2.3.2 Prinsip Kerja Radius	17
2.3.3 Mekanisme Akses RADIUS	17
2.3.4 Arsitektur Radius	19
2.3.5 RADIUS Accounting	20
2.4 MySQL.....	21
2.4.1 Kelebihan MySQL.....	22

BAB III METODOLOGI.....	24
3.1 Studi Literatur	24
3.2 Analisa Kebutuhan dan Perancangan Sistem.....	24
3.3 Implementasi dan Analisis Hasil.....	25
3.4 Pengambilan Kesimpulan dan Saran.....	27
BAB IV PERANCANGAN.....	28
4.1 Jaringan Komputer Eksisting Universitas Brawijaya	28
4.1.1 Topologi Jaringan Komputer Universitas Brawijaya	29
4.1.2 Spesifikasi router-router jaringan UB	31
4.1.3 Pengguna Jaringan Komputer Universitas Brawijaya	32
4.2 Analisis Kebutuhan	32
4.3 Perancangan Sistem	33
4.3.1 Perancangan Sistem Accounting Jaringan UB	33
4.3.2 Perancangan Prototipe Sistem <i>Accounting</i> Jaringan Komputer UB	34
4.3.3 Perancangan Spesifikasi Perangkat Keras	36
4.3.4 Perancangan Spesifikasi Perangkat Lunak.....	37
4.3.5 Perancangan Manajemen User.....	40
BAB V IMPLEMENTASI DAN ANALISIS HASIL.....	41
5.1 Implementasi	41
5.1.1 Server RADIUS	41
5.1.2 Server basis data dan manajemen <i>user</i>	45
5.1.3 NAS-1	47
5.1.4 NAS-2	49
5.1.5 Aplikasi Data Koneksi User	51
5.2 Pengujian dan Analisa Hasil	53
5.2.1 Proses Accounting	53
5.2.2 Pelacakan (<i>tracking</i>)	57
5.2.3 Troughput.....	62
BAB VI PENUTUP	66
6.1 Kesimpulan	66
6.2 Saran.....	67

DAFTAR PUSTAKA	68
LAMPIRAN 1	71
LAMPIRAN 2	83
LAMPIRAN 3	84
LAMPIRAN 4	109



DAFTAR GAMBAR

Gambar 2.1 Protokol TCP/IP	8
Gambar 2.2 Topologi Bintang	11
Gambar 2.3 Topologi Cincin	12
Gambar 2.4 Topologi Pohon.....	14
Gambar 2.5 Struktur paket data RADIUS	16
Gambar 2.6 Traffic pesan client NAS dengan server RADIUS	18
Gambar 2.7 Gambar 2.3.3. Arsitektur Mekanisme Autentikasi	19
Gambar 2.8 Traffic pesan client NAS dengan server RADIUS	20
Gambar 2.9 Pertukaran Message pada saat accounting session	21
Gambar 3.1 Alur Proses Analisis dan Perancangan sistem	25
Gambar 3.2 Alur Proses Implementasi	26
Gambar 3.3 Alur Proses Pengujian dan Analisis hasil	27
Gambar 4.1 Topologi Komputer Jaringan Universitas Brawijaya	29
Gambar 4.2 Potongan Topologi Komputer Jaringan Universitas Brawijaya	30
Gambar 4.3 Perancangan Topologi Sistem accounting Jaringan Komputer UB	34
Gambar 4.4 Perancangan Prototipe Sistem accounting Jaringan Komputer UB	35
Gambar 5.1 Halaman <i>login</i>	52
Gambar 5.2 Halaman data <i>accounting user</i>	52
Gambar 5.3 Halaman pencarian <i>user</i> nakal	53
Gambar 5.4 Prosedur pengujian untuk mengetahui proses accounting	55

Gambar 5.5 Proses accounting yang diharapkan	55
Gambar 5.6 Proses accounting pada pengujian	57
Gambar 5.7 Prosedur pengujian untuk pelacakan (<i>tracking</i>)	58
Gambar 5.8 Hasil query pengujian	61
Gambar 5.9 Hasil pencarian menggunakan aplikasi.....	62
Gambar 5.10 Prosedur pengujian untuk mengetahui besar throughput data	63



DAFTAR TABEL

Tabel 4.1 Spesifikasi Router Jaringan internal Universitas Brawijaya	31
Tabel 4.2 Alokasi IP Prototipe Sistem accounting Jaringan Komputer UB	35
Tabel 5.1 Fungsi tiap kolom pada tabel radacct	59
Tabel 5.2 Tabel hasil pelacakan.....	61
Tabel 5.3 Hasil pengujian besarnya throughput sebelum diterapkan sistem	64
Tabel 5.4 Hasil pengujian besarnya throughput sesudah diterapkan sistem.....	65



DAFTAR LAMPIRAN

- | | |
|------------|----------------------------------|
| Lampiran 1 | Script Konfigurasi Server RADIUS |
| Lampiran 2 | Script Konfigurasi NAS |
| Lampiran 3 | Data Pengujian Proses Accounting |
| Lampiran 4 | Data Pengujian Troughput |



DAFTAR ISTILAH

Server	Aplikasi yang memberikan pelayanan kepada user. <i>Server</i> dapat menerima permintaan (<i>request</i>), melakukan pelayanan yang diminta, kemudian mengembalikan sebagai <i>reply</i> . <i>Server</i> dapat melayani multi request bersamaan. [DHO-07].
Client	Pihak yang meminta pelayanan [DHO-07].
Router	Perangkat yang menghubungkan jaringan pada layer IP (dalam protokol TCP/IP) dan mengarahkan jalur paket data. Perangkat ini mampu memilih jalur yang terbaik untuk pengiriman data, karena memiliki <i>routing</i> [DHO-07].
Analisis Paket	Proses penangkapan dan menampilkan paket yang lewat pada suatu <i>interface</i> jaringan [STI-05].
Atribut	Salah satu bagian pada paket Radius yang membawa informasi yang spesifik mengenai autentikasi, otorisasi, dan detail konfigurasi untuk <i>request</i> dan <i>reply</i> [RFC-286].
Session	Layanan yang diberikan NAS kepada user sejak user berhasil terautentikasi hingga user <i>logoff</i> .
Log in	Proses yang bertujuan untuk mendapatkan akses pribadi oleh user yang ditandai dengan identifikasi user [WIK-10].
Log out	Proses berhenti dari sebuah akses setelah user melakukan proses log in sebelumnya [WIK-10].
DoS (Denial of Services)	Dos merupakan suatu metode penyerangan dengan membanjiri permintaan ke mesin server dengan permintaan palsu secara bertubi-tubi. Server dikirim permintaan yang terus-menerus atau permintaan yang di luar perkiraan sehingga tidak melayani permintaan lain atau bahkan sampai <i>down</i> , <i>hang</i> , <i>crash</i> [STI-05].

Ping Flooding

Serangan brute force sederhana berupa “flood” packet ICMP [STI-05].

Throughput

Throughput adalah bandwidth aktual yang terukur pada suatu ukuran waktu tertentu [DEW-03]



BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir ini, teknologi informasi dan telekomunikasi di Indonesia mulai berkembang dengan pesat. Semakin banyak dan terjangkaunya perangkat canggih dan modern yang masuk ke Indonesia seperti *notebook*, *PDA*, dan *handphone* yang dilengkapi dengan fasilitas untuk dapat terhubung ke jaringan komputer bisa menjadi salah satu penyebabnya. Beberapa kegiatan kehidupan seperti bisnis, jual beli barang, pendidikan, pertukaran informasi, berita, catatan harian, transaksi perbankan, berkomunikasi dengan orang lain yang jauh tempatnya, semuanya berkembang ke arah Internet.

Masalah yang akan dihadapi ketika terhubung dengan jaringan komputer dan Internet adalah mengenai keamanannya. Saat ini banyak ditemukan sistem jaringan komputer yang tidak menerapkan sistem keamanan yang memadai sehingga sangat memungkinkan bagi pengguna yang tidak berhak (*illegal*) dapat masuk ke dalam sistem jaringan komputer yang terhubung. Penyusup tersebut dapat saja melakukan perbuatan yang merugikan diantaranya seperti mengambil data, menyerang komputer atau perangkat lain yang terhubung dengan jaringan tersebut dan lain sebagainya. Dengan semakin berkembangnya kebutuhan akan informasi dan telekomunikasi, maka diperlukan suatu metode keamanan jaringan yang handal serta dapat memantau kegiatan *user* yang terkoneksi ke dalam jaringan komputer. Untuk itu salah satu caranya kita dapat menggunakan mekanisme keamanan AAA yang merupakan kepanjangan dari *Authentication, Authorization, dan Accounting* [WIC-06:5].

Otentikasi adalah kegiatan seseorang memverifikasi bahwa sesuatu itu valid atau sah [WIC-06:5]. Otorisasi adalah penentuan apakah sesuatu (user atau peralatan) itu mempunyai ijin untuk mengakses layanan. [WIC-06] Dan *accounting* mempunyai

arti melakukan tracking tentang siapa, apa, kapan dan dimana permintaan berasal dan kemana respon akan dikirimkan [WIC-06:5]. AAA sangat penting dalam jaringan nirkabel karena adanya roaming dan identitas user yang harus selalu dijaga, dan sistem harus terus memantau aktifitas user tersebut [WIC-06:5].

Remote Access Dial-IN User Service (RADIUS) merupakan salah satu protokol jaringan AAA yang penggunaannya cukup mendominasi di lapangan. Hal ini dikarenakan protokol ini bersifat terbuka dan vendor independent [WAR-04:6]. Ada beberapa alasan kenapa RADIUS dipilih, yaitu sederhana, efisien, dan mudah diimplementasikan [SET-05:3]. RADIUS menjalankan sistem administrasi pengguna yang terpusat, sistem ini akan mempermudah tugas administrator. Dengan sistem ini pengguna dapat menggunakan hotspot di tempat yang berbeda-beda dengan melakukan autentikasi ke sebuah RADIUS server [SET-05:3].

Pada saat ini, terdapat dua macam tipe koneksi di lokal jaringan komputer (*Local Area Network [LAN]*) Universitas Brawijaya yaitu melalui kabel (*wired*) dan tanpa kabel (*wireless*). Untuk jaringan komputer yang menggunakan kabel terdapat jaringan utama (*backbone*) yang tersebar di setiap fakultas kemudian dari tiap fakultas dibagi lagi ke tiap jurusan. Setelah itu pada tingkat jurusan terdapat beberapa titik *access point* yang disediakan sebagai fasilitas *hotspot area* untuk mahasiswa yang ingin melakukan koneksi ke jaringan komputer secara *wireless*. Namun pada kenyataannya saat ini kita tidak dapat mengetahui dengan pasti siapa *user* yang sedang melakukan koneksi ke jaringan komputer Universitas Brawijaya. Hal ini dapat mengancam keamanan jaringan komputer di Universitas Brawijaya karena user yang melakukan koneksi tidak tercatat serta tidak mudah melacak dan mengetahui siapa pelakunya jika terjadi suatu serangan. Berdasarkan kebutuhan keamanan tersebut, maka dibutuhkan suatu sistem keamanan AAA untuk menjamin keamanan dan kenyamanan *user* jaringan komputer di Universitas Brawijaya karena dengan sistem ini kita dapat mengetahui dengan pasti siapa *user* yang sedang melakukan koneksi. Untuk memudahkan agar user dapat berpindah dari *network* yang satu ke *network* yang lainnya, maka dibutuhkan suatu sistem yang terpusat. Oleh karena itu pada

tugas akhir ini menggunakan protokol RADIUS dan menitik beratkan pada proses *accounting* karena untuk proses otentikasi dan otorisasi akan dibahas pada tugas akhir lain yang dikerjakan bersamaan dengan tugas akhir ini oleh saudari Winda Septarini. Hasil *accounting* pada tugas akhir ini diolah dan dapat digunakan untuk keperluan *tracking* (siapa, dan dimana permintaan berasal) jika terjadi suatu tindak kejahatan yang menggunakan akses jaringan komputer internal Universitas Brawijaya.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, maka dapat dirumuskan masalah sebagai berikut :

1. Perancangan implementasi sistem *accounting user* Jaringan Komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya Malang
2. Membuat *prototype* implementasi sistem *accounting user* Jaringan Komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya Malang
3. Menguji dan menganalisa proses *accounting* yang terjadi pada RADIUS
4. Menguji dan menganalisa pelacakan (*tracking*) siapa dan dimana tempat user yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya,
5. Menguji performansi RADIUS dari sisi *throughput*

1.3 Batasan Masalah

Mengacu pada permasalahan yang sudah dipaparkan di atas maka pembatasan masalah pada tugas akhir ini antara lain

1. Masalah yang akan diteliti adalah perancangan implementasi sistem *accounting user* jaringan komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya.
2. Jaringan komputer yang dibahas merupakan jaringan komputer di Universitas Brawijaya Malang.

3. Aplikasi yang digunakan untuk RADIUS server adalah freeradius.
4. Sistem Operasi yang digunakan pada RADIUS server adalah Sistem Operasi Linux dengan distibusi Ubuntu.
5. Sistem Operasi yang digunakan pada NAS adalah Sistem Operasi Linux dengan distibusi Centos.
6. *Access point* yang digunakan pada penelitian ini adalah *wireless access point router* dengan jenis Linksys WRT54GL.
7. IP address yang digunakan *user wireless LAN* dan *access point* adalah IP address subnet jaringan komputer *internal* di Universitas Brawijaya
8. Penggunaan NAT (*Network Address Translation*) untuk tidak diijinkan di Universitas Brawijaya.
9. Tindak kejahatan yang dilacak adalah yang tindak kejahatan yang menggunakan akses jaringan komputer internal Universitas Brawijaya.

1.4 Tujuan

Merancang implementasi sistem *accounting user* jaringan komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya.

1.5 Manfaat

Manfaat yang bisa didapatkan dari tugas akhir ini adalah :

1. Untuk mengatur user jaringan komputer Universitas Brawijaya Malang.
2. Memudahkan pelacakan user yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya.

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut :

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, pembahasan, dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perancangan implementasi sistem *accounting user* jaringan komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya Malang.

BAB III Metodologi

Berisi tentang metode-metode yang digunakan untuk melakukan penelitian dan perencanaan sistem serta pengujian..

BAB IV Perancangan

Memuat tentang tahapan beserta penjelasan perencanaan dari sistem yang akan dibangun..

BAB V Pengujian dan Analisis

Membahas tahapan realisasi dari perancangan yang telah dibuat lalu dilakukan pengujian dan analisa.

BAB VI Penutup

Memuat kesimpulan dan saran-saran.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung [ARD-04:1]. Jaringan komputer merupakan suatu koleksi komputer-komputer terpisah yang berkomunikasi satu dengan lainnya memanfaatkan media komunikasi yang dipakai bersama-sama [HUS-04:1]. Tiap komputer, printer atau periferal yang terhubung dengan jaringan disebut node. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah [ARD-04:1].

2.1.1 Protokol Komunikasi

Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras [WKI-08].

Protokol perlu diutamakan pada penggunaan standar teknis, untuk menspesifikasi bagaimana membangun komputer atau menghubungkan peralatan perangkat keras. Protokol secara umum digunakan pada komunikasi *real-time* dimana standar digunakan untuk mengatur struktur dari informasi untuk penyimpanan jangka panjang [WKI-08].

Sangat susah untuk menggeneralisir protokol dikarenakan protokol memiliki banyak variasi didalam tujuan penggunaanya. Kebanyakan protokol memiliki salah satu atau beberapa dari hal berikut:

- Melakukan deteksi adanya koneksi fisik atau ada tidaknya komputer atau mesin lainnya.
- Melakukan metode "jabat-tangan" (*handshaking*).
- Negosiasi berbagai macam karakteristik hubungan.
- Bagaimana mengawali dan mengakhiri suatu pesan.
- Bagaimana format pesan yang digunakan.
- Yang harus dilakukan saat terjadi kerusakan pesan atau pesan yang tidak sempurna.
- Mendeteksi rugi-rugi pada hubungan jaringan dan langkah-langkah yang dilakukan selanjutnya
- Mengakhiri suatu koneksi.

Untuk memudahkan memahami Protokol, kita mesti mengerti Model OSI. Dalam Model OSI terdapat 7 layer dimana masing-masing layer mempunyai jenis protokol sesuai dengan peruntukannya [WKI-08].

2.1.1.1 Model OSI

Model referensi jaringan terbuka OSI atau *OSI Reference Model for open networking* adalah sebuah model arsitektural jaringan yang dikembangkan oleh badan *International Organization for Standardization* (ISO) di Eropa pada tahun 1977. OSI sendiri merupakan singkatan dari *Open System Interconnection*. Model ini disebut juga dengan model "Model tujuh lapis OSI" (*OSI seven layer model*) [WKI-08].

Sebelum munculnya model referensi OSI, sistem jaringan komputer sangat tergantung kepada pemasok (*vendor*). OSI berupaya membentuk standar umum jaringan komputer untuk menunjang interoperabilitas antar pemasok yang berbeda. Dalam suatu jaringan yang besar biasanya terdapat banyak protokol jaringan yang berbeda. Tidak adanya suatu protokol yang sama, membuat banyak perangkat tidak bisa saling berkomunikasi [WKI-08].

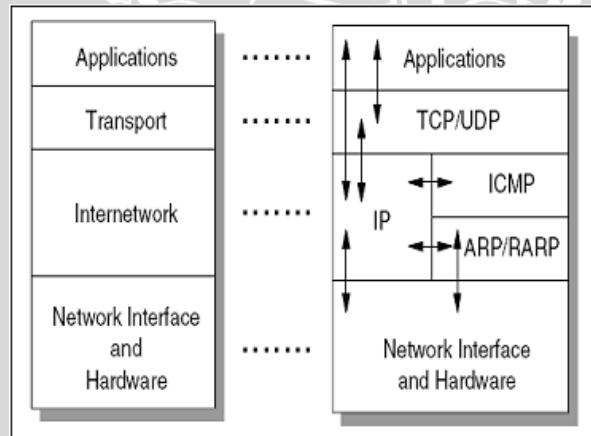
OSI Reference Model pun akhirnya dilihat sebagai sebuah model ideal dari koneksi logis yang harus terjadi agar komunikasi data dalam jaringan dapat berlangsung. Beberapa protokol yang digunakan dalam dunia nyata, semacam TCP/IP, DECnet dan IBM Systems

Network Architecture (SNA) memetakan tumpukan protokol (*protocol stack*) mereka ke *OSI Reference Model*. *OSI Reference Model* pun digunakan sebagai titik awal untuk mempelajari bagaimana beberapa protokol jaringan di dalam sebuah kumpulan protokol dapat berfungsi dan berinteraksi [WKI-08].

2.1.1.1.1 TCP/IP (Transmission Control Protocol/Internet Protocol)

TCP/IP adalah serangkaian protokol yang memungkinkan komunikasi antara beberapa komputer. Dulu TCP/IP ini tidak begitu penting bagi komputer-komputer untuk saling berkomunikasi karena bukan protokol yang umum. Tapi saat komputer saling terkoneksi dalam jaringan, kebutuhan itu muncul saat komputer-komputer tersebut sepakat untuk menggunakan protokol-protokol tertentu [BLA-04:2].

Seperti pada perangkat lunak, TCP/IP dibentuk dalam beberapa lapisan (layer). Dengan dibentuk dalam layer, akan mempermudah untuk pengembangan dan pengimplementasian. Antar layer dapat berkomunikasi ke atas maupun ke bawah dengan suatu penghubung *interface*. Tiap-tiap layer memiliki fungsi dan kegunaan yang berbeda dan saling mendukung layer diatasnya. Pada protokol TCP/IP dibagi menjadi 4 layer, tampak pada gambar berikut : [DHO-07:3]



Gambar 2.1 Protokol TCP/IP

Sumber: [DHO-07:3]

2.1.1.1.2 Layer Aplikasi (Aplications)

Layer aplikasi digunakan pada program untuk berkomunikasi menggunakan TCP/IP. Contoh aplikasi antara lain Telnet dan *File Transfer Protocol* (FTP). *Interface* yang digunakan untuk saling berkomunikasi adalah nomer port dan socket [DHO-07:3].

2.1.1.1.3 Layer Transport

Layer transport memberikan fungsi pengiriman data secara *end-to-end* ke sisi remote. Aplikasi yang beragam dapat melakukan komunikasi secara serentak (*simultaneously*). Protokol pada layer transport yang paling sering digunakan adalah *Transmission Control Protocol* (TCP), dimana memberikan fungsi pengiriman data secara *connection-oriented*, pencegahan duplikasi data, *congestion control* dan *flow control*.

Protokol lainnya adalah *User Datagram Protocol* (UDP), dimana memberikan fungsi pengiriman *connectionless*, jalur yang tidak *reliable*. UDP banyak digunakan pada aplikasi yang membutuhkan kecepatan tinggi dan dapat metoleransi terhadap kerusakan data [DHO-07:4].

2.1.1.1.4 Layer Internetwork

Layer *Internetwork* biasa disebut juga layer internet atau layer network, dimana memberikan “*virtual network*” pada internet. *Internet Protocol* (IP) adalah protokol yang paling penting. IP memberikan fungsi *routing* pada jaringan dalam pengiriman data. Protokol lainnya antara lain : IP, ICMP, IGMP, ARP, RARP [DHO-07:3].

2.1.1.1.5 Layer Network Interface

Layer network interface disebut juga layer link atau layer datalink, yang merupakan perangkat keras pada jaringan. Contoh : IEEE802.2, X.25, ATM, FDDI, dan SNA [DHO-07:3].

2.1.2 Topologi Jaringan Komputer

Topologi jaringan komputer adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan [SUP-07:1042]. Topologi jaringan adalah hal yang menjelaskan hubungan geometris antara unsur-unsur dasar penyusun jaringan, yaitu node, link, dan station [ARY-08:1]. Dalam suatu jaringan komputer jenis topologi yang dipilih akan mempengaruhi kecepatan komunikasi. Untuk itu maka perlu

dicermati kelebihan/keuntungan dan kekurangan/kerugian dari masing-masing topologi berdasarkan karakteristiknya [SUP-07:1042]. Topologi dapat dikategorikan menjadi beberapa bentuk topologi antara lain : topologi bintang, topologi cincin, topologi pohon, dsb [ARY-08:1].

2.1.2.1 Topologi Bintang

Topologi jaringan dimana setiap nodes dalam jaringan terhubung dengan node pusat dengan hubungan *point to point*. Semua data yang ditransmisikan ke node dalam jaringan selalu ditransmisikan ke node pusat yang kemudian ditransmisikan ke nodes di dalam jaringan, walupun node pusat mungkin juga sebuah titik koneksi biasa tanpa ada perangkat aktif untuk mengulang sinyal [ARY-08:2]. Topologi ini merupakan kontrol terpusat, semua link harus melewati pusat yang menyalurkan data tersebut kesemua simpul atau *client* yang dipilihnya [SUP-07:1043].

Walaupun kebanyakan jaringan yang didasarkan pada topologi ini memerlukan penggunaan hub sebagai node pusat, namun masih ada kemungkinan untuk mengimplementasikan sebuah jaringan yang didasarkan pada topologi star dengan menggunakan sebuah komputer atau bahkan titik koneksi biasa sebagai hub atau node pusat [ARY-08:2].

Model jaringan bintang ini relatif sangat sederhana, sehingga banyak digunakan oleh pihak bank yang biasanya mempunyai banyak kantor cabang yang tersebar dipelbagai lokasi. Dengan adanya konfigurasi bintang ini, segala macam kegiatan yang ada di kantor cabang dapat dikontrol dan dikoordinasikan dengan baik. Disamping itu, dunia pendidikan juga banyak memanfaatkan jaringan bintang ini guna mengontrol kegiatan anak didik mereka [ARY-08:2].

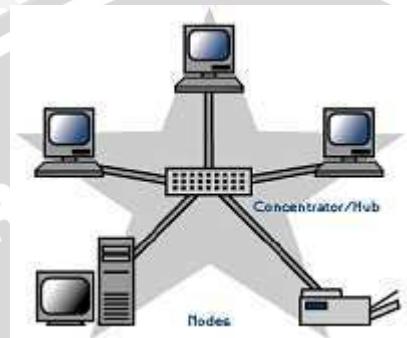
Kelebihan Topologi Bintang:

- Paling fleksibel.
- Pemasangan/perubahan stasiun sangat mudah dan tidak mengganggu bagian jaringan lain.
- Kontrol terpusat.
- Kemudahan deteksi dan isolasi kesalahan/kerusakan.

- Kemudahan pengelolaan jaringan [SUP-07:1044].

Kekurangan Topologi Bintang:

- Memerlukan kabel yang cukup panjang.
- Jika hub/concentrator gagal berfungsi maka semua jaringan akan terputus.
- Lebih mahal dengan adanya *concentrator* [ARY-08:2].



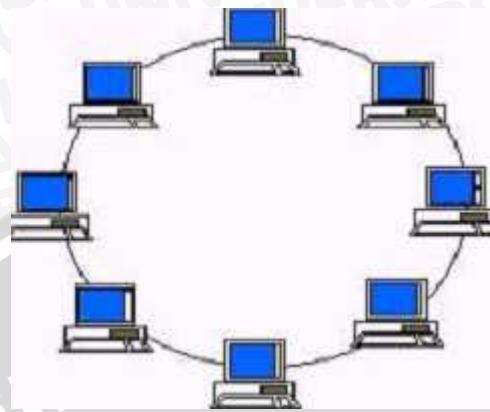
Gambar 2.2 Topologi Bintang
Sumber : [ARY-08:2]

2.1.2.2 Topologi Cincin

Topologi Cincin merupakan topologi jaringan yang tertua. Topologi Ring Network ini diperkenalkan dengan penyesuaian analog dan digital yang digunakan dalam sistem telepon. Sesuai dengan namanya, strukturnya berbentuk seperti cincin [ARY-08:2]. Topologi jaringan ini memiliki struktur dengan setiap nodes dalam jaringan terhubung dengan kedua nodes yang lain di jaringan dan dengan node pertama dan terakhir saling terhubung satu sama lain, membentuk cincin. Semua data yang ditransmisikan diantara nodes dalam jaringan berjalan dari satu node ke node berikutnya dengan pola sirkuler dan data umumnya lompat secara searah [ARY-08:3]. Jaringan akan disebut sebagai loop, data dikirimkan kesetiap simpul dan setiap informasi yang diterima simpul diperiksa alamatnya apakah data itu untuknya atau bukan [SUP-07:1043].

Pada jaringan ini terdapat beberapa peralatan yang saling dihubungkan satu dengan lainnya dan pada akhirnya akan membentuk bagan seperti halnya sebuah cincin. Jaringan cincin tidak memiliki suatu titik yang bertindak sebagai pusat ataupun pengatur lalu lintas data, semua simpul mempunyai tingkatan yang sama. Data yang dikirim akan berjalan

melewati beberapa simpul sehingga sampai pada simpul yang dituju. Dalam menyampaikan data, jaringan bisa bergerak dalam satu ataupun dua arah [ARY-08:3].



Gambar 2.3 Topologi Cincin

Sumber : [ARY-08:3]

Data yang dikirim atau diterima tetap bergerak satu arah dalam satu saat. Pertama, pesan yang ada akan disampaikan dari titik ke titik lainnya dalam satu arah. Apabila ditemui kegagalan, misalnya terdapat kerusakan pada peralatan yang ada, maka data yang ada akan dikirim dengan cara kedua, yaitu data kemudian akan ditransmisikan dalam arah yang berlawanan, dan pada akhirnya bisa berakhir pada tempat yang dituju [ARY-08:3].

Konfigurasi semacam ini relatif lebih mahal apabila dibanding dengan konfigurasi jaringan bintang. Hal ini disebabkan karena setiap simpul yang ada akan bertindak sebagai komputer yang akan mengatasi setiap masalah yang dihadapi, serta harus mampu membagi sumber daya yang dimilikinya pada jaringan. Di samping itu, sistem ini lebih sesuai digunakan untuk sistem yang tidak terpusat (decentralized-system), dimana tidak diperlukan adanya suatu prioritas tertentu [ARY-08:3].

Kelebihan Topologi Cincin:

- Aliran data cepat.
- Mampu melayani lalu lintas data yang padat.
- Waktu yang diperlukan dalam mengakses data optimal.
- Komunikasi antar terminal mudah.
- Tidak terjadi *data-collision* [ARY-08:3].

Kekurangan Topologi Cincin:

- Jika kabel utama bermasalah maka semua jaringan akan terputus.
- Penambahan dan pengurangan terminal sukar dilakukan [ARY-08:3].
- Peka kesalahan.
- Pengembangan jaringan lebih kaku [SUP-07:1043].

2.1.2.3 Topologi Pohon

Topologi jaringan pohon memadukan karakteristik dari jaringan linier dan jaringan bintang. Jaringan ini terdiri dari sekumpulan workstation berkonfigurasi dengan struktur bintang yang terhubung dengan kabel bus backbone. Jaringan pohon memungkinkan perluasan dari sub jaringan yang telah ada [ARY-08:4].

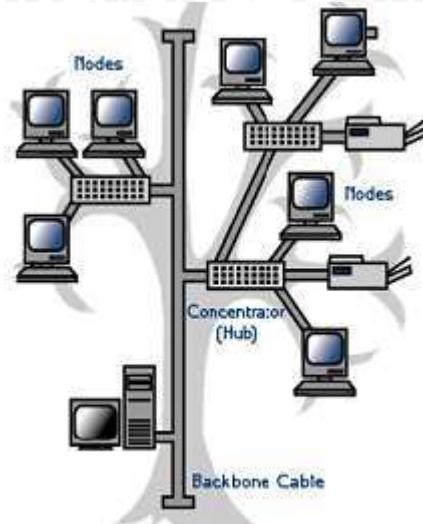
Pada jaringan pohon, terdapat beberapa tingkatan simpul (node). Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Keunggulan jaringan model pohon seperti ini adalah dapat terbentuknya suatu kelompok yang dibutuhkan setiap saat. Sebagai contoh, perusahaan dapat membentuk kelompok yang terdiri atas terminal pembukuan, serta kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah apabila simpul yang lebih tinggi tidak berfungsi, maka kelompok lain yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif lambat [ARY-08:4].

Kelebihan Topologi Pohon:

- Koneksi secara langsung (point to point) pada segmen jaringan tunggal.
- Topologi jaringan ini didukung oleh beberapa *vendor hardware* dan *software* [ARY-08:4].

Kekurangan Topologi Pohon:

- Cakupan segmen jaringan tergantung dari kabel.
- Jika jalur backbone putus, seluruh segmen jaringan akan putus.
- Pengimplementasiannya sulit [ARY-08:4].



Gambar 2.4 Topologi Pohon

Sumber : ARY-08:4]

2.2 Mekanisme AAA (Authentication, Authorization, Accounting)

Mekanisme AAA (*Authentication, Authorization, Accounting*) mengatur mekanisme bagaimana tata cara berkomunikasi, baik antara *client* ke *domain-domain* jaringan maupun antar *client* dengan domain yang berbeda dengan tetap menjaga keamanan pertukaran data. [ARI-07:21]

2.2.1 Otentikasi (Authentication)

Otentikasi yaitu proses pengesahan identitas pelanggan (*end-user*) untuk mengakses jaringan. Proses ini diawali dengan pengiriman kode unik (misalnya, *username*, *password*, pin, sidik jari, dsb) oleh pelanggan kepada *server*. Di sisi *server*, sistem akan menerima kode unik tersebut, selanjutnya membandingkannya dengan kode unik yang disimpan dalam database *server*. Jika hasilnya sama, maka *server* akan memberikan hak akses kepada pelanggan. Namun jika hasilnya tidak sama, maka *server* akan mengirimkan pesan kegagalan dan menolak hak akses pelanggan [WAR-04:4].

2.2.2 Otorisasi (Authorization)

Otorisasi (*Authorization*) merupakan proses pengecekan wewenang pengguna, mana saja hak-hak akses yang diperbolehkan dan mana saja yang tidak [ARI-07:22].

2.2.3 Pencatatan (*Accounting*)

Pencatatan (*Accounting*) merupakan proses pengumpulan data informasi seputar berapa lama *user* melakukan koneksi dan *billing time* yang telah dilalui selama pemakaian. [ARI-07:22] Pencatatan ini bisa bertujuan untuk analisis, pemeriksaan, pembayaran, perencanaan dan alokasi sumber daya. Pencatatan dapat juga untuk mengamankan sistem, seperti, mengawasi pelanggan yang mencurigakan, dan sebagainya [WAR-04:5].

2.3 Remote Authentication Dial-In User Service (RADIUS)

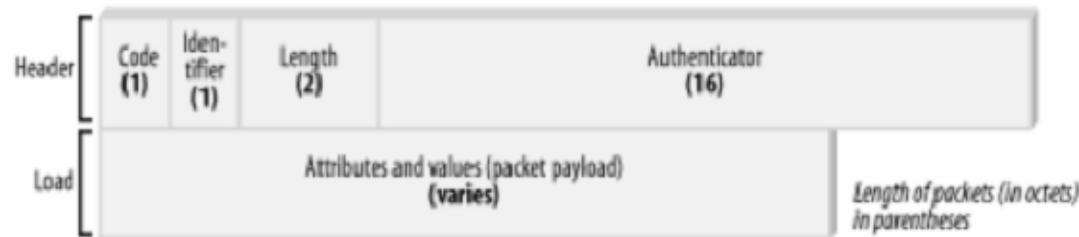
RADIUS merupakan suatu protokol yang dikembangkan untuk proses AAA (*authentication, authorization, and accounting*). RADIUS menjalankan sistem administrasi pengguna yang terpusat, sistem ini akan mempermudah tugas administrator [SET-05:3].

Remote Access Dial In User Service (RADIUS) dikembangkan dipertengahan tahun 1990 oleh *Livingstone Enterprise* (sekarang *Lucent Technologies*). Pada awalnya perkembangan RADIUS menggunakan port 1645 yang ternyata bentrok dengan layanan *datametrics*. Sekarang port yang dipakai RADIUS adalah port 1812 yang format standarnya ditetapkan pada *Request for Command* (RFC) 2138 [ARI-07:22].

Remote Authentication Dial-In User Service (RADIUS) (RFC2865) pada awalnya digunakan oleh ISP untuk otentifikasi dengan *username/password* sebelum dapat berkoneksi dengan jaringan ISP, dan biasanya adalah untuk user dial-up. RADIUS digunakan untuk *back-end authentication server* pada 802.1x. Selain RADIUS ada beberapa protokol AAA (*Authentication, Authorization, Accounting*) yang bisa juga digunakan yaitu TACACS, TACACS+, dan DIAMETER [FUA-08:6].

2.3.1 Format Paket Data RADIUS

Format paket RADIUS terdiri dari Code, Identifier, Length, Authenticator dan Attributes seperti ditunjukkan pada gambar berikut [ARI-07:22]



Gambar 2.5 Struktur paket data RADIUS

Sumber : [SET-05:3]

2.3.1.1 Code

Code memiliki panjang adalah satu oktet, digunakan untuk membedakan tipe pesan RADIUS yang dikirimkan pada paket. Kode-kode tersebut (dalam desimal) ialah:

- 1 Access-Request
- 2 Access-Accept
- 3 Access-Reject
- 4 Accounting-Request
- 5 Accounting-Response
- 11 Access-Challenge
- 12 Status-Server
- 13 Status-Client
- 255 Reserved

2.3.1.2 Identifier

Memiliki panjang 1 byte yang digunakan untuk menyesuaikan antara paket permintaan dan respon dari server RADIUS [ARI-07:22].

2.3.1.3 Length

Memiliki panjang 2 byte, memberikan informasi mengenai panjang paket. Jika paket kurang atau lebih dari yang diidentifikasi pada length maka paket akan dibuang [ARI-07:22].

2.3.1.4 Authenticator

Memiliki panjang 16 byte yang digunakan untuk mengautentikasi tanggapan dari server RADIUS [ARI-07:22].

2.3.1.5 Attributes

Berisikan informasi yang dibawa pesan RADIUS, setiap pesan dapat membawa satu atau lebih atribut. Contoh atribut RADIUS: nama pengguna, *password*, *CHAP-password*, alamat IP *access point* (AP), pesan balasan [SET-05:4].

2.3.2 Prinsip Kerja Radius

RADIUS merupakan protokol security yang bekerja menggunakan sistem *client-server* terdistribusi yang banyak digunakan bersama AAA untuk mengamankan jaringan pengguna yang tidak berhak [ARI-07:22].

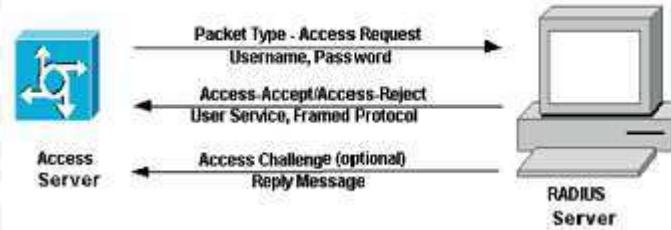
RADIUS melakukan otentikasi user melalui serangkaian komunikasi antara client dan server. Bila user telah berhasil melakukan otentikasi, maka user tersebut dapat menggunakan layanan yang disediakan oleh jaringan [ARI-07:22].

Protokol RADIUS merupakan protokol *connectionless* berbasis UDP yang tidak menggunakan koneksi langsung. Satu paket RADIUS ditandai dengan field UDP yang menggunakan port 1812. Beberapa pertimbangan RADIUS menggunakan lapisan transport UDP yaitu: [ARI-07:25]

1. Jika permintaan otentikasi pertama gagal, maka permintaan kedua harus dipertimbangkan.
2. Bersifat *stateless* yang menyederhanakan protokol pada penggunaan UDP.
3. UDP menyederhanakan implementasi dari sisi server.

2.3.3 Mekanisme Akses RADIUS

Protokol Lalu lintas pesan pada RADIUS menggunakan metode permintaan dan tanggapan (*client/server*) yang dapat dilihat pada gambar di bawah [ARI-07:27].

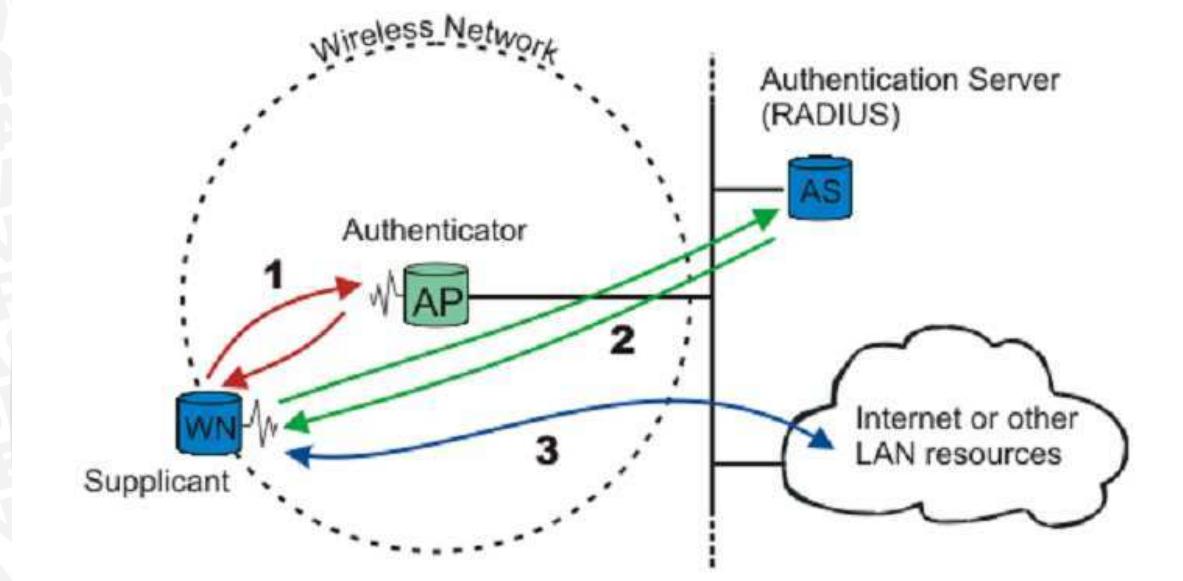


Gambar 2.6 Traffic pesan client NAS dengan server RADIUS
Sumber : [ARI-07:27]

- User melakukan dial-in pada NAS. NAS akan meminta user memasukkan nama dan password jika koneksi tersebut berhasil dibangun. [ARI-07:25].
- NAS akan membangun paket data berupa informasi, yang dinamakan *access request*. Informasi diberikan oleh NAS pada server RADIUS berisi informasi spesifik dari NAS itu sendiri yang meminta *access request*, port yang digunakan untuk koneksi modem serta nama dan *password*. Untuk proteksi dari *hacker*, NAS yang bertindak sebagai RADIUS *client*, melakukan enkripsi *password* sebelum dikirimkan pada RADIUS server. *Access request* ini dikirimkan pada jaringan dari RADIUS *client* ke RADIUS *server*. Jika RADIUS *server* tidak dapat dijangkau, RADIUS *client* dapat melakukan pemindahan rute pada server alternatif jika didefinisikan pada konfigurasi NAS [ARI-07:25].
- Ketika *access request* diterima, server otentikasi akan memvalidasi permintaan tersebut dan melakukan dekripsi paket data untuk memperoleh informasi nama dan *password*. Jika nama dan *password* sesuai dengan basis data pada *server*, *server* akan mengirimkan *access accept* yang berisi informasi kebutuhan sistem *network* yang harus disediakan oleh *user*. Selain itu *access accept* ini dapat berisi informasi untuk membatasi akses *user* pada jaringan. Jika proses *login* tidak menemui kesesuaian, maka RADIUS *server* akan mengirimkan *access reject* pada NAS dan *user* tidak dapat mengakses jaringan [ARI-07:26].
- Untuk menjamin permintaan *user* benar-benar diberikan pada pihak yang benar, RADIUS *server* mengirimkan *authentication key* atau *signature* yang menandakan keberadaannya pada RADIUS *client* [ARI-07:26].

2.3.4 Arsitektur Radius

Arsitektur sistem ini terdiri dari tiga bagian, yaitu wireless node (supplicant), access point (autentikator), otentikasi server. Otentikasi server yang digunakan adalah Remote Authentication Dial-In Service (RADIUS) server dan digunakan untuk autentikasi pengguna yang akan mengakses wireless LAN. EAP adalah protokol layer 2 yang menggantikan PAP dan CHAP [SET-05:5].



Gambar 2.7 Gambar 2.3.3. Arsitektur Mekanisme Autentikasi menggunakan RADIUS server

Sumber : [SET-05:5]

Penjelasan Gambar :

1. Wireless Node (WN)/Supplicant meminta akses ke wireless network, Access Point (AP) akan menanyakan identitas Supplicant. Tidak ada trafik data selain EAP yang diperbolehkan sebelum Supplicant terautentikasi. Access point bukanlah sebuah autentikator, tetapi access point berisi autentikator [SET-05:5].
2. Setelah nama-pengguna dan password dikirim, proses autentikasi dimulai. Protokol yang digunakan antara Supplicant dan Autentikator adalah EAP, atau EAP Protocol over LAN (EAPoL). Autentikator mengenkapsulasi kembali pesan EAP kedalam format RADIUS, dan mengirimnya ke RADIUS server. Selama proses autentikasi, autentikator hanya menyampaikan paket antara Supplicant dan RADIUS server.

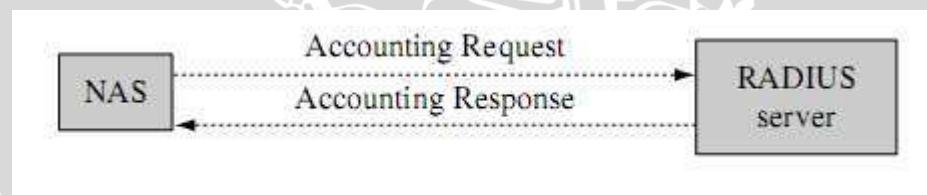
Setelah proses autentikasi selesai, RADIUS server mengirimkan pesan sukses (atau gagal, apabila proses autentikasinya gagal) [SET-05:6].

3. Apabila proses autentikasi sukses, Suplicant diperbolehkan untuk mengakses wireless LAN dan/atau Internet [SET-05:6].

2.3.5 RADIUS Accounting

Dokumen dasar standart RADIUS (RADIUS2865) tidak menyediakan spesifikasi untuk dukungan *accounting*. Akantetapi, *RADIUS accounting* didefinisikan di informasi RFC bagian yang lain (RADACC2866). Prosedur *accounting* juga berdasarkan model *client-server* dimana *client* (NAS) melewatkkan informasi *accounting* tentang *user* kepada *RADIUS server*, yang merupakan *host* mesin dari *RADIUS accounting* [NAK-05:139].

Accounting RADIUS menggunakan dua tipe *message*: *Accounting Request* dan *Accounting Response*, keduanya juga dikirimkan melalui UDP. *Accounting Request* selalu dikirimkan dari *RADIUS client* ke *RADIUS server*, ketika *Accounting Response* di-generate oleh *RADIUS server* saat menerima dan memproses *Accounting Requests* (Gambar 7.4.3). Bagaimanapun, seperti yang akan kita lihat pada skenario *roaming proxy server* mungkin berperan pada pertukaran *Accounting Request-Response* [NAK-05:139].



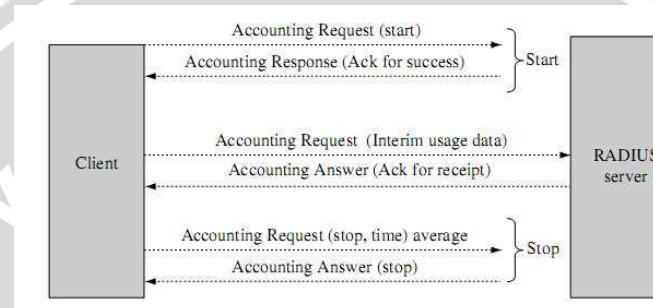
Gambar 2.8 Traffic pesan client NAS dengan server RADIUS

Sumber : [NAK-05:139]

2.3.5.1 Mekanisme Accounting

NAS mampu mendukung mekanisme *accounting* RADIUS untuk meng-generate *Accounting Request* “Start” pada saat operasi *start* dan mengirimkannya ke *RADIUS accounting server*. Paket ini menentukan, antara lain, tipe service yang dikirim, dan user yang dikirim service tersebut. Saat penerimaan *accounting request* yang valid, server menambahkan *accounting record* pada *log* dan menjawab *request* dengan meng-generate *Accounting Response* untuk mengidikasikan bahwa paket telah diterima [NAK-05:139].

Pada akhir pengiriman *service*, *client* akan meng-generate paket *Accounting Stop* yang menjelaskan tipe servis telah dikirim dan dan statistik seperti durasi *session*, alasan *disconnect*, jumlah oktet *input* dan *output*.itu semua akan dikirim ke *RADIUS accounting server*, yang mana akan mengirim kembali pernyataan bahwa paket telah diterima. Tentusaja, jika *RADIUS accounting server* tidak dapat dengan sukses merekam, paket *accounting* tidak akan mengirim pernyataan *Accounting-Response* kepada *client* [NAK-05:140].



Gambar 2.9 Pertukaran Message pada saat accounting session

Sumber : [NAK-05:140]

2.4 MySQL

Menyimpan data dalam file biasa memiliki banyak keterbatasan. Semakin besar ukuran file, pencarian data menjadi lebih sulit. File biasa juga tidak memiliki kemampuan untuk mengolah data, misalnya menghitung total nilai, rata-rata, dan lain sebagainya [PRA-06:18].

MySQL adalah sebuah program database server yang mampu menerima dan mengirimkan datanya dengan sangat cepat, multi user serta menggunakan perintah standart SQL (*Structured Query Language*).

MySQL merupakan sebuah database yang free, artinya kita bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. Selain sebagai database server, MySQL juga merupakan program yang dapat mengakses suatu database MySQL yang berposisi sebagai server. Pada saat itu berarti program kita berposisi sebagai Client. Jadi MySQL adalah sebuah database yang dapat digunakan baik sebagai Client maupun Server.

Database MySQL, biasa kita baca mai-es-ki-el atau bisa juga mai-se-kuel, merupakan suatu perangkat lunak database yang berbentuk database relasional atau dalam bahasa basisdata sering kita sebut dengan Relation Database Management System (RDBMS) yang menggunakan bahasa permintaan bernama SQL.

MySQL adalah sebuah program database, sedangkan SQL adalah bahasa perintah (Query) dalam program MySQL. Selain MySQL ada beberapa database server lain yang menggunakan standar query berupa SQL antara lain adalah Oracle, PostgreSQL, MySQL front, MsSQL, SQL Server, dll.

2.4.1 Kelebihan MySQL

MySQL adalah sebuah database server, dapat juga berperan sebagai client sehingga sering disebut database client/server, yang open source dengan kemampuan dapat berjalan baik di OS (Operating Sistem) manapun, dengan Platform Windows maupun Linux. Selain itu database ini memiliki beberapa kelebihan disbanding database lain, di antaranya adalah:

- MySQL sebagai Database Management System (DBMS)
- MySQL sebagai Relational Database Management System (RDBMS)
- MySQL adalah sebuah Software database yang OpenSource, artinya program ini bersifat free atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
- MySQL merupakan database server, jadi dengan menggunakan database ini dapat diconnect ke media internet sehingga dapat diakses dari jauh
- MySQL merupakan sebuah database client. Selain menjadi server yang melayani permintaan, MySQL juga dapat melakukan query yang mengakses database pada Server. Jadi MySQL dapat juga berperan sebagai Client.
- MySQL mampu menerima query yang bertumpuk dalam satu permintaan atau yang disebut Multi Threading.
- MySQL merupakan sebuah database yang mampu menyimpan data berkapasitas sangat besar hingga berukuran Gigabyte sekalipun.

- MySQL didukung oleh driver ODBC, artinya database MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
- MySQL adalah database menggunakan enkripsi password. Jadi database ini cukup aman karena memiliki password untuk mengaksesnya.
- MySQL merupakan Serer database yang multi user, artinya database ini tidak hanya digunakan oleh sepihak orang akan tetapi merupakan database yang dapat digunakan oleh banyak pengguna.
- MySQL dapat menciptakan lebih dari 16 kunci per tabel, dan dalam satu kunci memungkinkan berisi belasan Field (kolom).
- MySQL mendukung field yang dijadikan sebagai kunci primer dan kunci Uniq (atau Unique).
- MySQL didukung oleh sebuah component C dan perl API, sehingga Database MySQL dapat diakses melalui sebuah program aplikasi yang berada di bawah protocol internet berupa Web. Biasanya aplikasi yang sering digunakan adalah PHP dan Perl.
- MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-update-an tabel.
- MySQL menggunakan suatu bahasa permintaan standar yang bernama SQL (*Structured Query Language*) yaitu sebuah bahasa permintaan yang distandardkan pada beberapa database server seperti Oracle, PosgreSQL, dll.

Nah dengan beberapa kelebihan yang dimiliki diatas MySQL menjadisebuah program database yang sangat popular digunakan. Pada umumnya MySQL digunakan sebagai database yang sangat popular digunakan. Pada umumnya MySQL digunakan sebagai database yang diakses melalui web.

BAB III

METODOLOGI

Penyusunan dan penelitian ini dilakukan dengan metode-metode sebagai berikut :

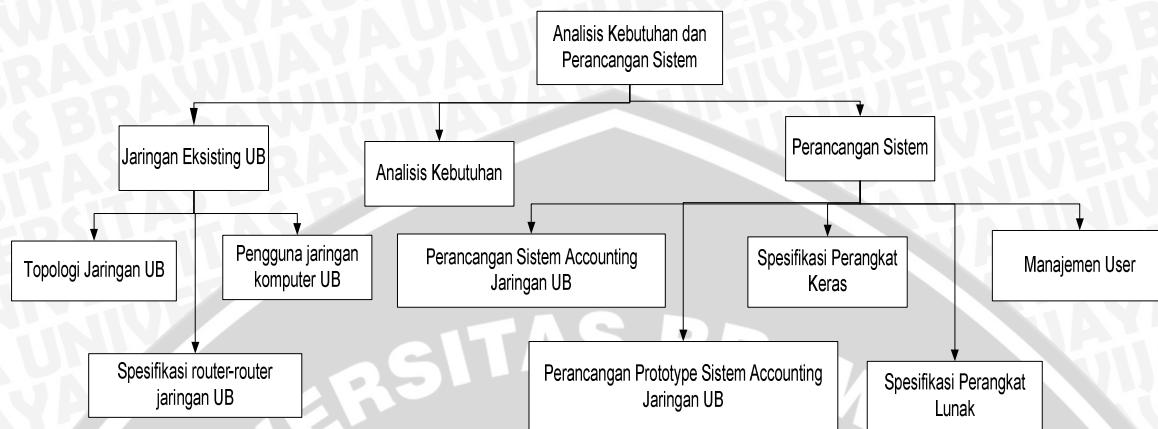
3.1 Studi Literatur

Berupa pencarian literatur tentang *AAA server* menggunakan protokol *RADIUS* khususnya pada proses *accounting* (pencatatan) dan hal-hal yang mendukung penelitian tentang *RADIUS* seperti *NAS*, *database* server yang menyimpan data *RADIUS accounting*, serta sistem operasi yang di gunakan serta tentang konfigurasi wireless router *access point*. Literatur dapat berupa *text book*, *whitepaper*, *RFC*, tugas akhir, tesis, maupun sumber bacaan *softcopy* yang didapatkan dari *internet*.

3.2 Analisa Kebutuhan dan Perancangan Sistem

Pada tahapan ini langkah awalnya adalah mempelajari dan menganalisa jaringan komputer eksisting UB meliputi topologi jaringan komputer UB, mengetahui spesifikasi perangkat keras *router*, serta siapa saja pengguna jaringan komputer UB. Menganalisa topologi yang ada di UB bertujuan untuk mengetahui dimana saja letak dan berapa jumlah dari *access point*, *router*, dan *node-node* lain yang ada, serta untuk mengetahui bagaimana pengaturan ip dan aturan-aturan yang berlaku. Hal ini bertujuan agar peneliti dapat melakukan analisa kebutuhan dan perancangan sistem yang akan diterapkan diantaranya yaitu menentukan dimana letak dan rancangan konfigurasi server *RADIUS* nantinya, letak dan rancangan konfigurasi *access point* dan *router* yang nantinya akan dijadikan *NAS* serta letak dan konfigurasi komputer yang akan digunakan pada tahap pengujian. Langkah selanjutnya yaitu menentukan aplikasi apa yang digunakan untuk server *RADIUS*, server basis data, dan *NAS* serta bagaimana konfigurasinya. Selain itu dibuat aplikasi data koneksi *user* untuk mempermudah seorang administrator jaringan komputer melihat data *accounting user*. Selain itu aplikasi ini juga dapat digunakan untuk melakukan pencarian *user* dengan berdasarkan alamat ip dan/atau waktu user melakukan koneksi.

Selanjutnya akan dilakukan perancangan dan dibuat *prototype* untuk keperluan analisis dan pengujian. Untuk lebih jelasnya dapat dilihat seperti pada gambar di bawah.



Gambar 3.1 Alur Proses Analisis dan Perancangan sistem

3.3 Implementasi dan Analisis Hasil

Pada tahap ini hal pertama yang dilakukan adalah merealisasi dan mengimplementasikan hasil analisa kebutuhan dan perancangan *prototype* pada satu *segmen* tertentu di jaringan komputer Universitas Brawijaya sesuai dengan perancangan yang telah dibuat. Selanjutnya yaitu melakukan instalasi server *RADIUS* dan mengkonfigurasinya. Konfigurasi dilakukan dengan tujuan untuk mendapatkan sistem yang paling optimum pada server *RADIUS* serta untuk keamanan sistem autentikasi. Setelah server *RADIUS* selesai, maka diikuti instalasi *database server* dan konfigurasinya, selanjutnya yaitu membuat database yang di butuhkan oleh server *RADIUS* dan mengkonfigurasi ulang server *RADIUS* untuk disinkronisasikan dengan database yang telah dibuat. Pada server basis data ini juga dibuat aplikasi data koneksi *user*. Sampai tahap ini implementasi untuk server *RADIUS* dan database server telah selesai.

Tahap berikutnya adalah konfigurasi *access point*, *router* serta *NAS* yang merupakan client *RADIUS*. Untuk pembuatan *NAS* kita perlu menginstall dan mengkonfigurasi router sebagai client *RADIUS*. Konfigurasi tersebut di maksudkan untuk proses *request* ke *RADIUS* dan agar *NAS* dapat berkomunikasi secara langsung dengan server *RADIUS* untuk proses AAA yang akan terhubung ke jaringan.

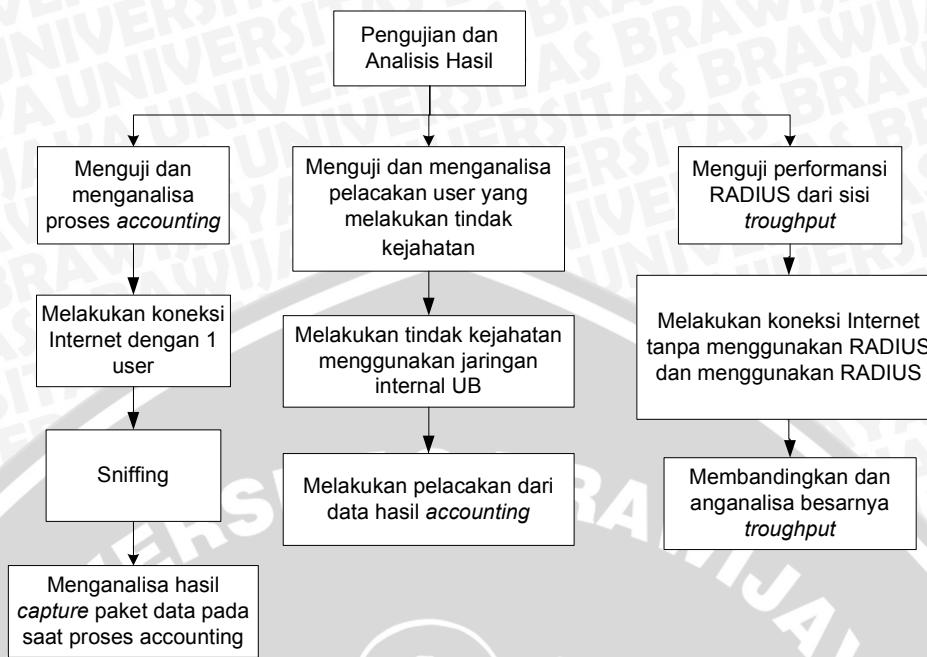


Gambar 3.2 Alur Proses Implementasi

Setelah beberapa tahap diatas maka kebutuhan implementasi untuk penelitian telah selesai dilaksanakan dan hal yang perlu dilakukan adalah mempersiapkan komputer untuk melakukan pengujian terhadap sistem yang akan dibangun yaitu pengujian pada jaringan eksternal untuk pengujian keamanan dan pengujian pada jaringan internal untuk melakukan *sniffing* agar peneliti dapat melakukan analisa terhadap paket data yang lewat pada saat proses pencatatan (*accounting*) berlangsung. Selanjutnya yaitu mengolah hasil *accounting* untuk keperluan *tracking* (siapa, apa, kapan dan dimana permintaan berasal dan kemana respon akan dikirimkan) jika terjadi suatu tindak kejahatan yang menggunakan akses jaringan komputer.

Pengujian sistem dilakukan untuk mengetahui proses serta keberhasilan skenario dan konfigurasi sistem *accounting* yang terjadi ketika *user* melakukan koneksi ke jaringan komputer. Selain itu pengujian dilakukan untuk mengetahui proses apa saja yang terjadi ketika proses *accounting* dengan cara menganalisa paket data apa saja yang lewat antara *NAS* dan server *RADIUS* pada saat koneksi dengan cara melakukan *sniffing*. Selanjutnya melakukan uji performansi berupa *throughput* dengan menggunakan aplikasi *iperf*. Terakhir menganalisa hasil *accounting* untuk keperluan *tracking* (siapa, apa, kapan dan dimana permintaan berasal dan kemana respon akan dikirimkan) jika terjadi suatu tindak kejahatan yang menggunakan akses dari jaringan komputer internal UB.

Setelah hasil pengujian didapatkan, selanjutnya dilakukan analisa terhadap setiap hasil pengujian yang didapatkan.



Gambar 3.3 Alur Proses Pengujian dan Analisis hasil

3.4 Pengambilan Kesimpulan dan Saran

Setelah mendapatkan hasil dan analisis dari tahap pengujian, maka langkah berikutnya adalah penarikan kesimpulan dari hasil pengujian dan memberi saran untuk pengembangan lebih lanjut.

BAB IV

PERANCANGAN

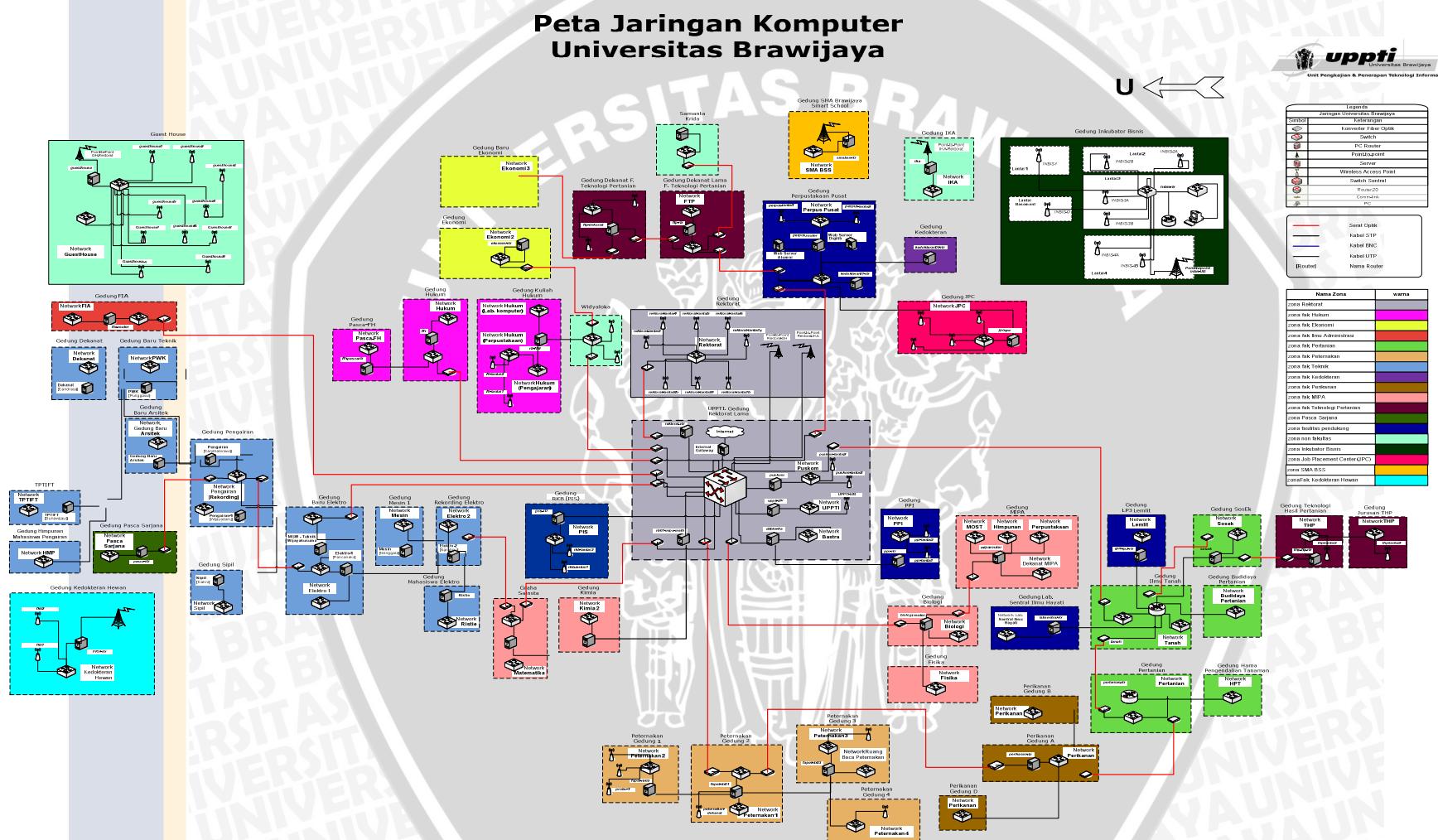
Pada bab ini akan dijelaskan mekanisme perancangan implementasi sistem *accounting user* jaringan komputer secara terpusat menggunakan protokol RADIUS di Universitas Brawijaya. Sebelum melakukan perancangan penulis harus mengetahui terlebih dahulu bagaimana jaringan komputer eksisting di UB saat ini. Selanjutnya dapat dilakukan analisis kebutuhan dan perancangan sistem.

4.1 Jaringan Komputer Eksisting Universitas Brawijaya

Subbab ini akan menjelaskan bagaimana topologi jaringan komputer Universitas Brawijaya dan seperti apa spesifikasi router-router yang ada pada sistem jaringan komputer Universitas Brawijaya saat ini. Hal ini dibutuhkan untuk analisis dimana peletakan NAS dan *server* serta sejauh mana sistem ini dapat diterapkan di Universitas Brawijaya nantinya berkaitan dengan spesifikasi perangkat keras dan perangkat lunak yang telah ada.

Pada saat ini, terdapat dua macam tipe koneksi di lokal jaringan komputer (*Local Area Network [LAN]*) Universitas Brawijaya yaitu melalui kabel (*wired*) dan tanpa kabel (*wireless*). Untuk jaringan komputer yang menggunakan kabel terdapat jaringan utama (*backbone*) yang tersebar di setiap fakultas kemudian dari tiap fakultas dibagi lagi ke tiap jurusan. Setelah itu pada tingkat jurusan terdapat beberapa titik *access point* yang disediakan sebagai fasilitas *hotspot area* untuk mahasiswa yang ingin melakukan koneksi ke jaringan komputer secara *wireless*.

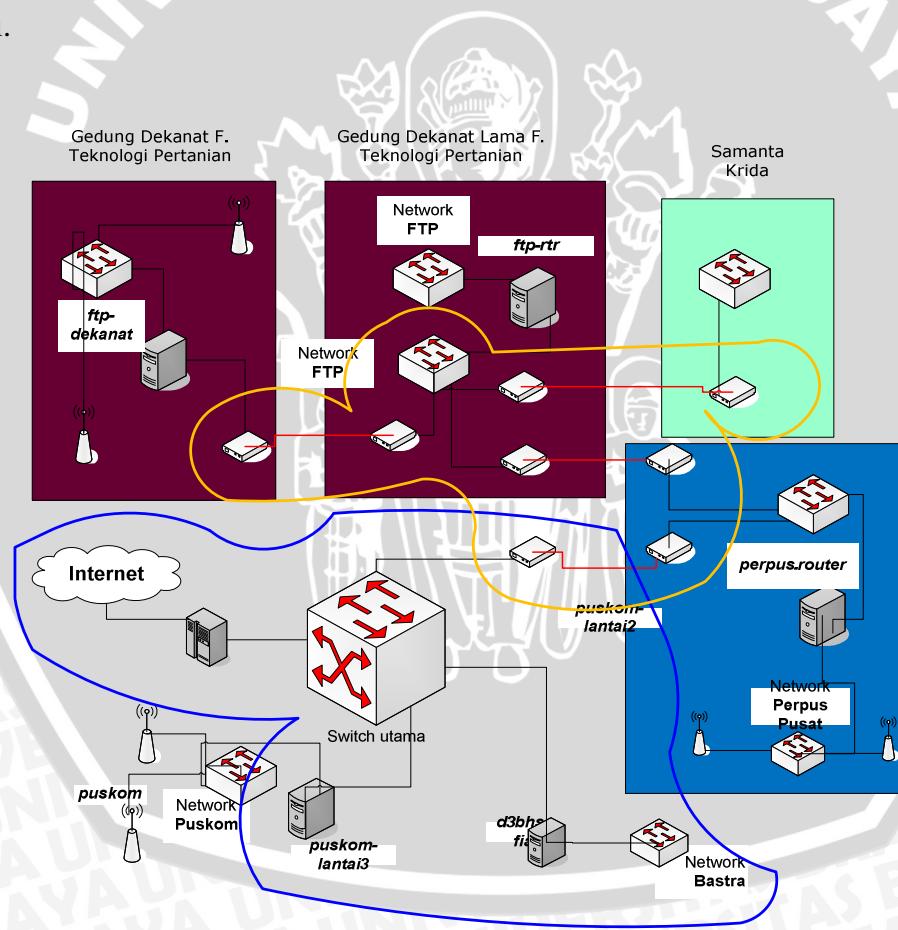
4.1.1 Topologi Jaringan Komputer Universitas Brawijaya



Gambar 4.1 Topologi Komputer Jaringan Universitas Brawijaya

Sumber : PPTI - UB

Dari gambar di atas dapat kita ketahui bahwa tipe topologi jaringan komputer di UB adalah tipe jaringan pohon. Hal ini dapat dilihat dari struktur jaringannya yang memadukan karakteristik dari jaringan linier dan jaringan bintang. Jaringan komputer di UB terdiri dari sekumpulan *workstation* berkonfigurasi dengan struktur bintang yang terhubung dengan jaringan *backbone* yang merupakan jaringan utama. Jaringan *backbone* atau jaringan utama UB berbentuk bintang yang mana simpul pusatnya adalah *internal gateway* jaringan komputer UB sedangkan untuk simpul *client*-nya merupakan merupakan sebuah *gateway* dari *zona* jaringan komputer di setiap fakultas dan *non fakultas*. Sedangkan untuk *gateway* setiap jurusan dihubungkan secara linier yang terhubung ke jaringan *backbone* yang berada di *zona* fakultas masing-masing. Selanjutnya untuk jaringan di jurusan menggunakan jaringan bintang yang terhubung ke *gateway* jurusan. Untuk lebih jelasnya kita lihat gambar berikut ini.



Gambar 4.2 Potongan Topologi Komputer Jaringan Universitas Brawijaya

Dari gambar di atas dapat kita lihat bahwa pada jaringan utama yang berada dalam area garis berwarna biru berbentuk jaringan bintang dengan *switch* utama sebagai *node* pusat. Sedangkan untuk simpul *client* mengarah ke fakultas-fakultas salah satunya ke jaringan komputer Fakultas Teknologi Pertanian (FTP). Selanjutnya dapat kita lihat pula jaringan komputer yang berada di dalam area garis oranye merupakan jaringan linier. Jaringan tersebut merupakan jaringan yang menghubungkan antara jaringan komputer di gedung FTP lain dan gedung Samantha Krida dengan jaringan utama. Pola jaringan komputer di FTP seperti di atas juga diterapkan pada fakultas-fakultas dan gedung lain di UB. Dari analisis diatas dapat kita simpulkan bahwa jaringan komputer di UB merupakan bentuk jaringan pohon.

4.1.2 Spesifikasi router-router jaringan UB

Berikut adalah spesifikasi router-router jaringan komputer yang ada Universitas Brawijaya.

Tabel 4.1 Spesifikasi Router Jaringan internal Universitas Brawijaya

No.	Hostname	Prosesor	Hardisk (GB)	RAM (MB)	Interface (Jumlah)	Sistem Operasi
1	rtr-fh2	intel pentium 4 2.80 Ghz	40	256	2	Centos
2	pasca-hukum-rtr.brawijaya.ac.id	Intel Pentium (R) 4 CPU 3.06 GHz	40	256	5	Centos
3	fh.router.brawijaya.ac.id	Intel Pentium (R) 4 CPU 3.06 GHz	40	256	5	Centos
4	fapet-rtr01.brawijaya.ac.id	Pentium III (Katmai)	4	64	6	Redhat
5	fapet2-rtr	Pentium III (Katmai)	20	128	3	Centos
6	fapet3-rtr	Intel Celeron 2.40 GHz	80	256	3	Centos
8	ftp-rtr	Intel ® Pentium® 4 CPU 2.26 GHz	16	128	2	Centos
9	Fakultas Teknologi Pertanian THP	Pentium III (Katmai)	8	128	2	Redhat
11	FTP-Dekanat-rtr	AMD Sempron(tm) Processor 2800+	10	64	2	Centos
12	perpus.router.brawijaya.a.c.id	Intel (R) Pentium (R) 4 CPU 2.26 GHz	8	256	4	Centos
13	jpc-gw	Intel (R)Pentium (R) Dual Cpu E2140 @ 1.60 Ghz	160	512	2	Centos
14	sosek.router.brawijaya.ac.id	Pentium II (Deschutes)	4	80	2	Fedora Core
15	IKA-GW	Pentium III (Coppermine)	4	128	2	Centos

Sumber : PPTI - UB

Dari tabel di atas dapat dilihat bahwa router-router di UB masih menggunakan pc router dengan linux sebagai sistem operasinya. Beberapa masih menggunakan menggunakan Pentium III bahkan ada satu yang masih menggunakan pentium II sedangkan untuk yang lain 50% sudah menggunakan pentium IV dan diatasnya. Untuk sistem operasi yang digunakan adalah linux dengan distro redhat, fedora core, dan centos yang semuanya merupakan turunan dari redhat. Di setiap router juga terdapat server dhcp yang dapat memberikan user jaringan komputer konfigurasi alamat IP secara otomatis. Selain itu juga terdapat firewall menggunakan iptables untuk keamanan jaringan komputer di tingkat *router*.

4.1.3 Pengguna Jaringan Komputer Universitas Brawijaya

Pada dasarnya pengguna yang diizinkan untuk menggunakan fasilitas jaringan komputer UB adalah dosen, karyawan, serta mahasiswa UB. Namun seiring dengan makin banyaknya *access point* yang menyediakan akses internet gratis secara *wireless* di UB saat ini, kita tidak dapat mengetahui dengan pasti siapa *user* yang sedang melakukan koneksi ke jaringan komputer UB. Hal ini dikarenakan setiap orang bisa dengan bebas melakukan koneksi ke jaringan komputer UB melalui *access point* tanpa adanya sistem keamanan yang memadai sehingga sangat memungkinkan bagi pengguna yang tidak berhak (*illegal*) dapat masuk ke dalam sistem jaringan komputer yang terhubung dan melakukan aktifitas yang merugikan.

4.2 Analisis Kebutuhan

Setelah melihat jaringan komputer eksisting UB dapat dirumuskan beberapa kebutuhan sebagai berikut :

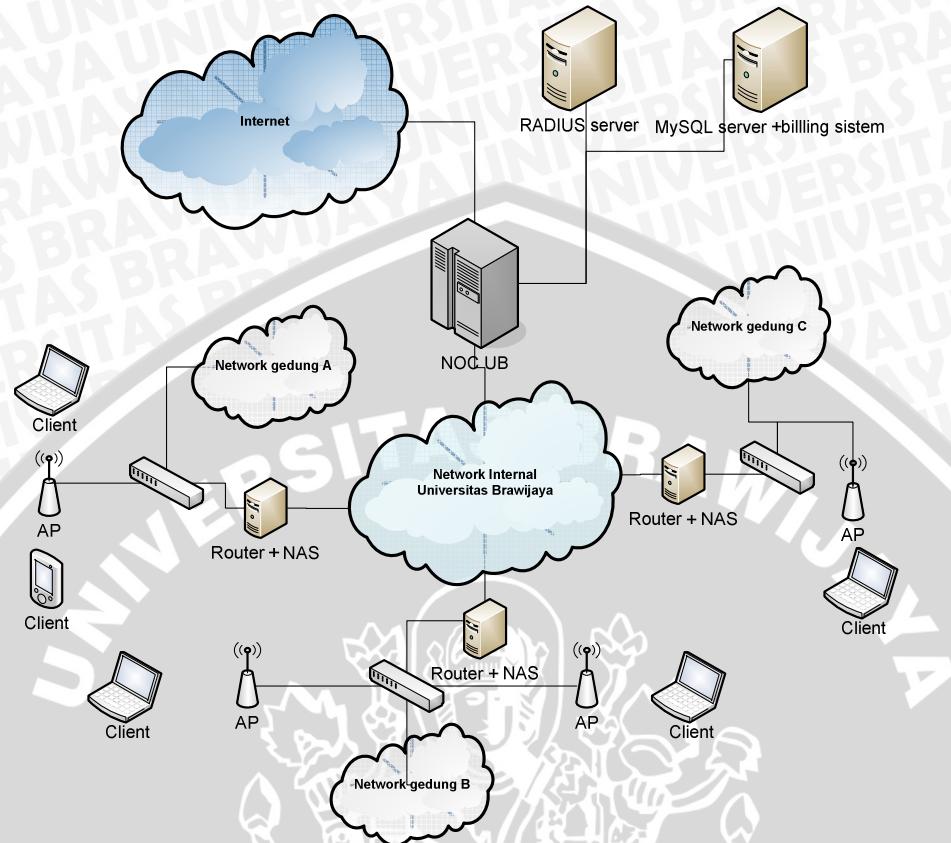
- Suatu sistem yang dapat melakukan proses pencatatan siapa saja user yang yang terkoneksi
- Suatu sistem yang dapat melakukan pencatatan aktifitas user yang terkoneksi dan sumber daya yang telah digunakan.
- Suatu aplikasi yang dapat digunakan mempermudah seorang administrator jaringan komputer melihat data *accounting user*.
- Sistem yang mudah diimplementasikan dengan jaringan komputer eksisting UB

4.3 Perancangan Sistem

Perancangan Implementasi Sistem *Accounting User* Jaringan Komputer Secara Terpusat Menggunakan Protokol RADIUS skripsi ini dibuat berdasarkan analisis kebutuhan yang sudah dijabarkan diatas. Pearancangan sistem meliputi bagaimana topologi dan prinsip kerja sistem itu nantinya Selanjutnya untuk keperluan pengujian sistem, maka dibuat sebuah prototipe dari sistem yang kemudian diimplementasikan untuk dianalisa bagaimana performansi dari sistem yang telah dirancang tersebut.

4.3.1 Perancangan Sistem Accounting Jaringan UB

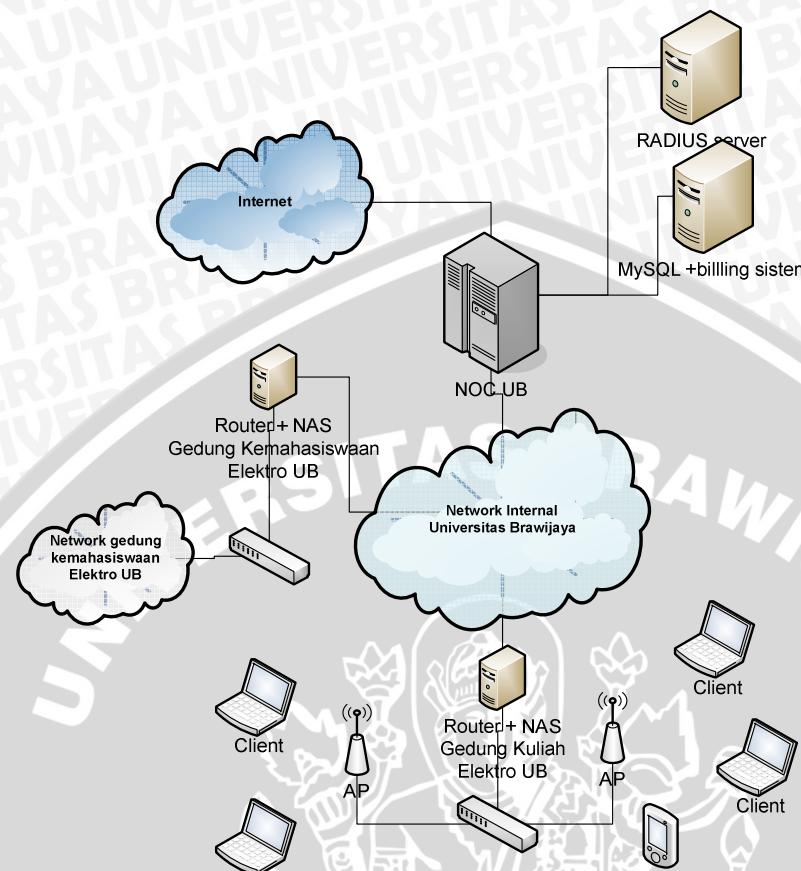
Sistem *accounting* jaringan UB dirancang secara terpusat, yaitu semua NAS atau *client radius* yang ada disetiap *end-router* akan melakukan proses autentikasi, otorisasi dan *accounting* pada satu *server RADIUS*. Hal ini bertujuan agar setiap user dapat menggunakan fasilitas penggunaan *resource* jaringan komputer UB di manapun dia berada dalam kawasan kampus. *Server RADIUS*, *server* basis data, dan billing sistem ditempatkan pada *NOC (Network Operation Center)* - UB yang merupakan pusat operasi jaringan komputer di Universitas Brawijaya. Selanjutnya untuk setiap *end-router* yang ada di UB digunakan sebagai NAS yang berfungsi sebagai gateway dan autentikator user jaringan komputer UB baik yang terkoneksi menggunakan kabel maupun tanpa kabel. Sedangkan untuk *access point* akan difungsikan hanya pada layer 2 saja, hal ini bertujuan agar mudah dalam pengimplementasian nantinya karena tidak membatasi pada merek dan jenis *access point* tertentu. Untuk lebih jelasnya dapat kita lihat pada gambar topologi berikut.



Gambar 4.3 Perancangan Topologi Sistem accounting Jaringan Komputer UB

4.3.2 Perancangan Prototipe Sistem *Accounting* Jaringan Komputer UB

Perancangan prototipe sistem *accounting* jaringan komputer UB dibuat berdasarkan analisis yang telah dibahas sebelumnya. Perancangan prototipe ini akan digunakan untuk keperluan pengujian sistem yang telah dirancang. Hasil perancangan prototipe kemudian akan diimplementasikan di jaringan komputer Jurusan Teknik Elektro Universitas Brawijaya agar penulis dapat memperoleh data *real* untuk analisis hasil pengujian. Pada perancangan prototipe ini topologi yang digunakan adalah bentuk jaringan pohon, hal ini disesuaikan dengan topologi jaringan komputer UB. Pada prototipe ini peletakan *server RADIUS*, *server MySQL* serta *billing* sistem disesuaikan dengan perancangan yang dibuat sebelumnya yaitu di zona *NOC-UB*. Selanjutnya digunakan dua buah *end-router* yang difungsikan sebagai *NAS* untuk pengujian dua jenis jaringan komputer yaitu jaringan kabel dan jaringan tanpa kabel. Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 4.4 Perancangan Prototipe Sistem accounting Jaringan Komputer UB

Adapun alokasi IP dapat dilihat pada tabel berikut:

Tabel 4.2 Alokasi IP Prototipe Sistem accounting Jaringan Komputer UB

Nama	Hostname	Interface	Alamat IP	Gateway
Server RADIUS		eth 0	172.18.3.31/24	172.18.3.1/24
Server basis data		eth 0	172.18.3.32/24	172.18.3.1/24
NAS-1		eth 0 (<i>uplink</i>)	172.17.67.62/	172.17.67.41/
		tun 0 (<i>downlink</i>)	172.17..91.1/25	
NAS-2		eth 0 (<i>uplink</i>)	172.17.67.67/	172.17.67.65/
		tun 0 (<i>downlink</i>)	172.17..91.0/25	

4.3.3 Perancangan Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yaitu berupa deskripsi tentang perangkat atau peralatan yang digunakan dalam tugas akhir ini. Berisi spesifikasi prosesor, memory, hard disk dan peralatan lainnya. Deskripsi perangkat juga bisa menyebutkan merk, type, nama, kecepatan, kapasitas dan hal-hal lain yang cukup mendetail dari perangkat tersebut.

Perangkat keras yang digunakan dalam penelitian ini yaitu berupa 4 buah *personal computer* yang digunakan sebagai *server RADIUS*, 2 *client RADIUS*, dan *server basis data* yang akan disusun sebagai *prototipe* jaringan komputer UB untuk keperluan pada saat pengujian. Selain itu, diperlukan 2 *wireless access point* untuk pengujian agar *supplicant* dapat terkoneksi ke jaringan komputer baik menggunakan kabel maupun *wireless*. Berikut adalah spesifikasi dari perangkat keras yang digunakan dalam penelitian kali ini.

Server Radius:

- Prosessor : Intel(R) Pentium(R) 4 CPU 2.00GHz
- RAM : 128 MB
- Hardisk : 40 GB
- LAN Card : D-Link System Inc RTL8139 Ethernet

Server basis data dan manajemen user:

- Prosessor : Intel(R) Pentium(R) III CPU family 1266MHz
- RAM : 640 MB
- Hardisk : 18 GB
- LAN Card : Intel Corporation 82557/8/9/0/1 Ethernet Pro 100

NAS-1:

- Prosessor : AMD Athlon(tm) XP 2500+
- RAM : 256 MB
- Hardisk : 40 GB
- LAN Card :
 - Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+
 - VIA Technologies, Inc. VT6102 [Rhine-II]

NAS-2

- Prosessor : Intel(R) Pentium(R) 4 CPU 2.40GHz
- RAM : 512 MB
- Hardisk : 40 GB
- LAN Card :
 - VIA Technologies, Inc. VT6102 [Rhine-II]
 - Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+

Access point-1

- Produk/seri : Linksys WRT54GL 802.11 b/g
- Data Rate : 54 Mbps
- Fungsi : Bridge

Access point-2

- Produk/seri : 3com 3crwe876075 802.11 a/b/g
- Data Rate : 108Mbps
- Fungsi : Bridge

4.3.4 Perancangan Spesifikasi Perangkat Lunak

Pada sistem *accounting user* menggunakan RADIUS ini perangkat lunak yang digunakan yaitu

4.3.4.1 Freeradius

Freeradius merupakan aplikasi RADIUS *server* yang digunakan sebagai *server* otentikasi, otorisasi dan *accounting*. Berikut spesifikasi perancangannya

- Freeradius menggunakan versi 2.1.6-0
- Berapa pada pusat jaringan *internal* UB.
- *Server* memiliki dukungan terhadap *server* MySQL sebagai basis data yang menyimpan data *accounting user*.
- *Server* mampu membatasi user agar tidak bisa melakukan koneksi dengan *account* yang sama pada saat yang bersamaan.

- *Server* mampu melakukan pembatasan akses user berdasarkan lama waktu atau besarnya volume pemakaian.
- Freeradius mendaftar alamat IP setiap NAS yang ada agar dapat berkomunikasi dengan *server*. Untuk keamanan koneksi antara NAS dan *server*, keduanya memiliki sebuah kata kunci yang sama disebut *radiussecret*.

4.3.4.2 Mysql

MySQL merupakan DBMS (*Data Base Management System*) yang berfungsi sebagai *server* basis data untuk penyimpanan data *accounting* dari *server* RADIUS. Berikut spesifikasi perancangannya

- MySQL menggunakan versi 5.0
- *Server* basis data terpisah dengan *server* RADIUS dan berada satu segmen alamat IP agar dapat saling berkomunikasi.
- Berada pada pusat jaringan *internal* UB.
- Basis data memiliki skema tabel easyhotspot yang merupakan skema tabel Radius yang dimodifikasi dengan tambahan tabel untuk manajemen user.

4.3.4.3 Chillispot

Chillispot adalah aplikasi yang digunakan sebagai NAS yang merupakan *client* RADIUS. Berikut spesifikasi perancangannya

- Menggunakan versi Chillispot-1.1.0-1
- Alamat IP NAS didaftarkan pada *server* RADIUS.
- Berfungsi sebagai *router* dan *server* DHCP yang dapat memberikan alamat IP secara otomatis kepada user.
- Memiliki alamat IP dan segmentasi jaringan yang sama dengan jaringan eksisting UB.
- *Firewall* menggunakan *rule* eksisting jaringan UB dengan tambahan rule yang mendukung fungsi NAS yaitu me-*redirect* semua *packet forwarding* ke port 3990 sebelum user melakukan otentifikasi.

- Untuk mengamankan komunikasi data antara NAS dan *user* maka diterapkan fungsi https pada halaman login sehingga informasi berharga seperti *username* dan *password* dalam keadaan terenkripsi.

4.3.4.4 Linux Ubuntu

Linux Ubuntu sebagai Sistem Operasi yang digunakan pada komputer server RADIUS. Alasan mengapa menggunakan sistem operasi ini adalah karena dengan sistem operasi ini lebih mudah dalam tahap implementasi, selain itu juga sistem operasi mendukung aplikasi freeradius versi terbaru. Versi yang digunakan adalah Linux Ubuntu 8.04.

4.3.4.5 Linux Centos

Linux Centos sebagai Sistem Operasi yang digunakan pada komputer NAS. Alasan mengapa menggunakan sistem operasi ini adalah karena dengan sistem operasi merupakan sistem operasi yang banyak digunakan pada *end-router* yang ada di UB. versi yang digunakan adalah Centos 5.3.

4.3.4.6 Aplikasi Data Koneksi User

Aplikasi ini dibuat untuk mempermudah seorang administrator jaringan komputer melihat data *accounting user*. Selain itu aplikasi ini juga dapat digunakan untuk melakukan pencarian *user* dengan berdasarkan alamat ip dan/atau waktu user melakukan koneksi. Berikut spesifikasi perancangannya

- Mempunyai halaman login untuk administrator
- Aplikasi dapat menampilkan data *accounting user*.
- Aplikasi dapat melakukan pencarian *user* dengan berdasarkan alamat ip dan/atau waktu user melakukan koneksi.

4.3.5 Perancangan Manajemen User

Sesuai dengan analisis kebutuhan di atas, maka manajemen user jaringan komputer UB dibagi sebagai berikut

- Mahasiswa
 - Username menggunakan NIM
 - Koneksi akan putus jika user tidak melakukan aktifitas selama 5 menit
 - User tidak bisa melakukan *login* lebih dari satu kali pada saat yang bersamaan.
- Dosen dan Pegawai
 - Username menggunakan NIP/NIK
 - Koneksi akan putus jika user tidak melakukan aktifitas selama 5 menit
 - User tidak bisa melakukan *login* lebih dari satu kali pada saat yang bersamaan.

BAB V

IMPLEMENTASI DAN ANALISIS HASIL

Prototipe sistem *accounting* user jaringan komputer UB yang sudah dirancang pada Bab IV, akan diimplementasi dan diujikan pada tahap ini. Ada tiga tahap utama yang dilakukan pada bab ini, yaitu implementasi, pengujian dan analisis hasil prototipe sistem *accounting* user jaringan komputer UB.

5.1 Implementasi

Implementasi dilakukan dengan melakukan instalasi perangkat keras dan perangkat lunak. Setelah itu, menyusun sesuai dengan prototipe topologi jaringan komputer yang telah dibuat pada tahap perancangan. Selanjutnya dilakukan konfigurasi pada setiap server sebagai berikut.

5.1.1 Server RADIUS

Konfigurasi pada komputer server RADIUS dilakukan setelah Sistem Operasi Linux Ubuntu ter-*install*. Dalam penginstalan, digunakan mode minimum, yaitu hanya menginstal paket dan software yang dibutuhkan, untuk alasan keamanan dan keringkasan. Yang dipilih dalam penginstalan antara lain :

- Freeradius, aplikasi server RADIUS.
- Freeradius-mysql, modul tambahan agar freeradius dapat menggunakan MySQL sebagai server basis data.

Konfigurasi yang perlu dilakukan agar server RADIUS dapat berfungsi dengan baik adalah sebagai berikut

- a. Mengkonfigurasi alamat IP dan *default gateway* server RADIUS

Alamat ip di *setting ip static* sesuai dengan perancangan yang telah ada. Konfigurasi dilakukan dengan perubahan pada *file interface* seperti berikut :

```
#vim /etc/network/interface
```

```
iface eth0
inet static address 172.18.3.31
netmask 255.255.255.0
gateway 172.18.3.1
```

- b. Melakukan konfigurasi freeradius agar dapat berkomunikasi dengan *server* basis data MySQL

Konfigurasi ini dilakukan dengan merubah nilai pada parameter database, server, login, password, radius_db dan usergroup_table yang terdapat dalam file sql.conf sesuai dengan konfigurasi database yang digunakan. Konfigurasi dilakukan seperti dibawah ini :

```
#vim /etc/freeradius/sql.conf
```

```
database = "mysql"
driver = "rlm_sql_${database}"
server = "172.18.3.32"
login = "koko"
password = "sayangwinda"
radius_db = "easyhotspot"
usergroup_table = "usergroup"
```

- c. Melakukan konfigurasi freeradius agar membaca dan menyimpan data *accounting* user dari server basis data

Konfigurasi ini dilakukan dengan menambahkan parameter *sql* pada fungsi *accounting* di file radius.conf seperti di bawah ini :

```
#vim /etc/freeradius/sites-enabled/default
```

```
accounting {
    sql
}
```

- d. Melakukan konfigurasi freeradius untuk membatasi user, agar tidak bisa melakukan koneksi dengan *account* yang sama pada saat yang bersamaan

Pembatasan user dilakukan dengan menambahkan baris perintah pada file dialup.conf. Dengan baris perintah tersebut, freeradius akan melakukan *query* pada basis data untuk memeriksa apakah user sedang melakukan koneksi dengan *account* yang sama pada saat *user login*.

```
#vim /etc/freeradius/sql/mysql/dialup.conf
```

```
simul_count_query = "SELECT COUNT(*) \
    FROM ${acct_table1} \
    WHERE username = '%{SQL-User-Name}' \
    AND acctstoptime IS NULL"

simul_verify_query = "SELECT radacctid, acctsessionid, username, \
    nasipaddress, nasportid, framedipaddress, \
    callingstationid, framedprotocol \
    FROM ${acct_table1} \
    WHERE username = '%{SQL-User-Name}' \
    AND acctstoptime IS NULL"
```

- e. Melakukan konfigurasi freeradius agar mampu melakukan pembatasan akses user berdasarkan lama waktu atau besarnya volume pemakaian

```
#vim /etc/freeradius/sql/mysql/counter.conf
```



```
sqlcounter noresetcounter {  
    counter-name = Max-All-Session-Time  
    check-name = Max-All-Session  
    reply-name = Session-Timeout  
    sqlmod-inst = sql  
    key = User-Name  
    reset = never  
    query = "SELECT SUM(AcctSessionTime) FROM radacct  
    WHERE UserName='%{%k}'"  
}  
  
sqlcounter octetslimit {  
    counter-name = Max-All-MB  
    check-name = Max-All-MB  
    reply-name = Chillispot-Max-Total-Octets  
    key = User-Name  
    reset = never  
    query = "SELECT SUM(acctinputoctets+acctoutputoctets)  
    from radacct WHERE UserName='%{%k}'"  
    sqlmod-inst = sql  
}
```

```
#vim /etc/freeradius/sites-enabled/default
```

```
authorize {  
    sql  
    noresetcounter  
    octetslimit  
}
```

- f. Mendata informasi NAS yang diperbolehkan melakukan komunikasi dengan server RADIUS

```
# vim /etc/freeradius/client.conf
```

```
client 172.17.67.62 {  
    secret      = testing123-1  
    shortname   = NAS-1  
}  
  
client 172.17.67.67 {  
    secret      = testing123-2  
    shortname   = NAS-2  
}
```

5.1.2 Server basis data dan manajemen user

Konfigurasi pada komputer server basis data juga dilakukan setelah Sistem Operasi Linux Ubuntu ter-*install*. Dalam penginstalan, digunakan mode minimum, yaitu hanya menginstal paket dan software yang dibutuhkan, untuk alasan keamanan dan keringkasan. Yang dipilih dalam penginstalan antara lain :

- MySQL-server, aplikasi server basis data.
- Apache, sebagai server web untuk aplikasi Easyhotspot karena merupakan aplikasi berbasis web.
- php5, modul tambahan server web agar dapat menterjemahkan bahasa pemrograman php
- php5-mysql, modul tambahan server web agar aplikasi yang berbasis bahasa pemrograman php dapat melakukan koneksi dengan server database MySQL.
- Easyhotspot, sebagai *web based management user* dan sistem *billing*.

Konfigurasi yang perlu dilakukan agar server basis data dapat berfungsi dengan baik adalah sebagai berikut

- a. Mengkonfigurasi alamat IP dan *default gateway* server basis data

Alamat ip di *setting ip static* sesuai dengan perancangan yang telah ada. Konfigurasi dilakukan dengan perubahan pada *file interface* seperti berikut :

```
#vim /etc/network/interface
```

```
iface eth0
inet static address 172.18.3.32
netmask 255.255.255.0
gateway 172.18.3.1
```

- b. Membuat data base untuk RADIUS

```
#mysql -u root -p
```

```
>CREATE DATABASE easyhotspot;
>quit;
```

- c. Menambah user baru pada server basis data agar server RADIUS dapat mengakses basis data easyhotspot

```
#mysql -u root -p
```

```
> CREATE USER koko@172.18.3.31 IDENTIFIED BY 'sayangwinda';
>GRANT ALL PRIVILEGES ON easyhotspot.* to 'koko'@'172.18.3.31'
IDENTIFIED BY 'sayangwinda';
>FLUSH PRIVILEGES;
>quit;
```

- d. Menambahkan skema tabel ke basis data easyhotspot.

```
#mysql -u root -p easyhotspot < /var/www/easyhotspot/easyhotspot.sql
```

- e. Melakukan konfigurasi easyhotspot agar dapat berkomunikasi dengan *server* basis data MySQL

```
#vim /var/www/easyhotspot/system/application/config/database.php
```

```
$db['default']['hostname'] = "localhost";
$db['default']['username'] = "koko";
$db['default']['password'] = "sayangwinda";
$db['default']['database'] = "easyhotspot";
```

5.1.3 NAS-1

Konfigurasi pada komputer NAS-1 dilakukan setelah Sistem Operasi Linux Centos ter-install. Dalam penginstalan, juga digunakan mode minimum, yaitu hanya menginstal paket dan software yang dibutuhkan, untuk alasan keamanan dan keringkasan. Yang dipilih dalam penginstalan antara lain :

- Chillispot, aplikasi server basis data.
- Apache, sebagai server web untuk halaman login
- IPTABLES, sebagai firewall
- Open-ssl, modul untuk keamanan data web

Konfigurasi yang perlu dilakukan agar NAS dapat berfungsi dengan baik adalah sebagai berikut

- a. Mengkonfigurasi alamat IP dan *default gateway* NAS-1

Alamat ip di *setting ip static* sesuai dengan perancangan yang telah ada. Konfigurasi dilakukan dengan perubahan pada *file interface* seperti berikut :

```
#vi /etc/sysconfig/network-script/ifcfg-eth0
```

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
DHCP_HOSTNAME=NAS-1
IPADDR=172.17.67.62
NETMASK=255.255.255.224
```

```
#vi /etc/sysconfig/network-script/ifcfg-eth1
```

```
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
DHCP_HOSTNAME=NAS-1
```

- b. Mengaktifkan fungsi untuk meneruskan paket data

Konfigurasi ini dilakukan dengan menghilangkan tanda # di depan parameter net.ipv4.ip_forward yang terdapat dalam file sysctl.conf

```
#vi /etc/sysctl.conf
```

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

- c. Konfigurasi jaringan chillispot

Konfigurasi ini dilakukan untuk memberikan alamat IP NAS dan menentukan interface yang terkoneksi langsung dengan user. Selain itu menentukan DNS serta range IP otomatis yang diberikan kepada user.

```
#vi /etc/chilli.conf
```

```
net 172.17.91.0/25
dynip 172.17.91.0/24
dns1 202.162.208.99
dns2 202.162.208.100
domain NAS-1
dhcpif eth1
```

- d. Konfigurasi chillispot agar dapat berkomunikasi dengan *server RADIUS*

```
#vi /etc/chilli.conf
```

```
radiuslisten 127.0.0.1
radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
radiussecret theradiussecret
uamserver https://172.17.91.1/cgi-bin/hotspotlogin.cgi
uamsecret theuamsecret
uamhomepage http://172.17.91.1/login/
uamlisten 172.17.91.1
uamallowed 172.17.91.1
```

5.1.4 NAS-2

Konfigurasi pada komputer NAS-2 dilakukan setelah Sistem Operasi Linux Centos terinstall. Dalam penginstalan, juga digunakan mode minimum, yaitu hanya menginstal paket dan software yang dibutuhkan, untuk alasan keamanan dan keringkasan. Yang dipilih dalam penginstalan antara lain :

- Chillispot, aplikasi server basis data.
- Apache, sebagai server web untuk halaman login
- IPTABLES, sebagai firewall
- Open-ssl, modul untuk keamanan data web

Konfigurasi yang perlu dilakukan agar NAS dapat berfungsi dengan baik adalah sebagai berikut

- a. Mengkonfigurasi alamat IP dan *default gateway* NAS-2

Alamat ip di *setting ip static* sesuai dengan perancangan yang telah ada. Konfigurasi dilakukan dengan perubahan pada *file* interface seperti berikut :

```
#vi /etc/sysconfig/network-script/ifcfg-eth0
```

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
DHCP_HOSTNAME=NAS-2
IPADDR=172.17.67.67
NETMASK=255.255.255.224
```

```
#vi /etc/sysconfig/network-script/ifcfg-eth1
```

```
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
DHCP_HOSTNAME=NAS-2
```

- b. Mengaktifkan fungsi untuk meneruskan paket data

Konfigurasi ini dilakukan dengan menghilangkan tanda # di depan parameter net.ipv4.ip_forward yang terdapat dalam *file* sysctl.conf

```
#vi /etc/sysctl.conf
```

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

- c. Konfigurasi jaringan chillispot

Konfigurasi ini dilakukan untuk memberikan alamat IP NAS dan menentukan *interface* yang terkoneksi langsung dengan user. Selain itu menentukan DNS serta range IP otomatis yang diberikan kepada user.

```
#vi /etc/chilli.conf
```

```
net 172.17.91.0/25
dynip 172.17.91.0/24
dns1 202.162.208.99
dns2 202.162.208.100
domain NAS-!
dhcpif eth1
```

- d. Konfigurasi chillispot agar dapat berkomunikasi dengan *server RADIUS*

```
#vi /etc/chilli.conf
```

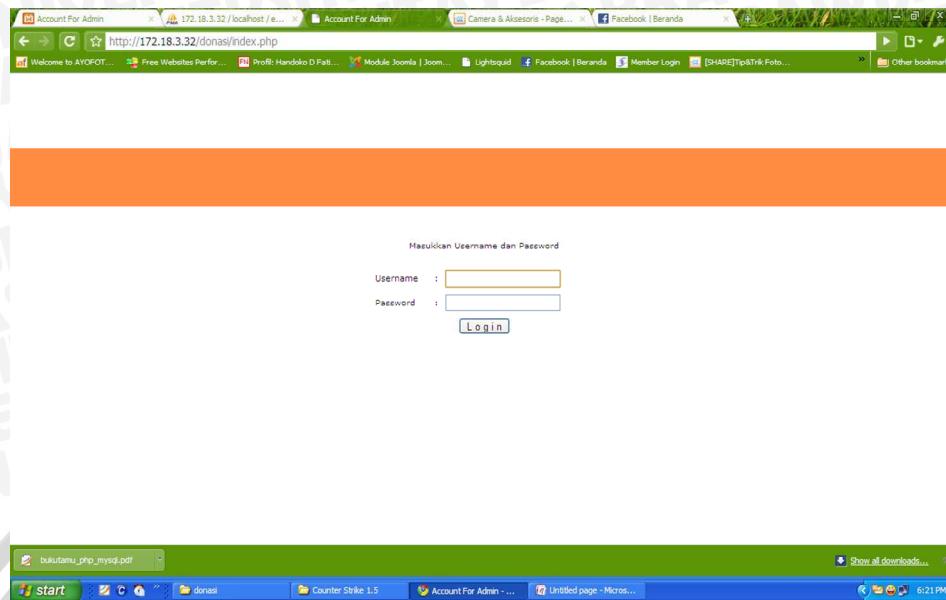
```
radiuslisten 127.0.0.1
radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
radiussecret theradiussecret
uamserver https://172.17.91.1/cgi-bin/hotspotlogin.cgi
uamsecret theuamsecret
uamhomepage http://172.17.91.1/login/
uamlisten 172.17.91.1
uamallowed 172.17.91.1
```

5.1.5 Aplikasi Data Koneksi User

Pembuatan aplikasi data koneksi *user* ini terdiri dari 3 bagian antara lain :

- a. Halaman *login* administrator

Halaman ini digunakan administrator untuk melakukan *login* ke aplikasi sehingga tidak semua orang dapat melihat data *accounting user*.

Gambar 5.1 Halaman *login*

b. Halaman data *accounting user*

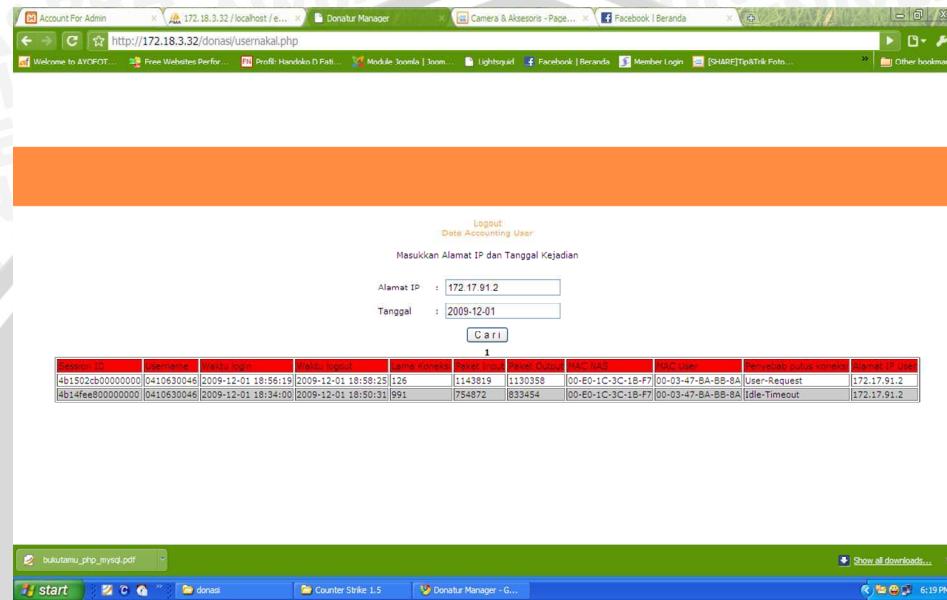
Halaman ini digunakan administrator untuk melihat data *accounting user*.

Laporan Pencarian User Nakal											
No	ID	Nama	Jenis Kelamin	Tanggal Lahir	Alamat	Kota	Kode Pos	No Telepon	Email	Status	Tanggal Masuk
Bscc06e00000000	pasihheru	2010-01-28 12:49:01	2010-01-28 13:30:20	2475	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.30
Bscc06f00000000	[REDACTED]	2010-01-28 10:57:06	2010-01-28 12:05:21	4153	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.10
Bscc06g0000000001	[REDACTED]	2010-01-28 13:55:35	21028	956434	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.21
Bscc06h0000000003	pasihheru	2010-01-28 10:29:20	2010-01-28 11:39:17	4197	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Toreout	172.17.91.23
Bscc06i0000000002	[REDACTED]	2010-01-28 10:20:27	2010-01-28 10:29:33	535	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.12
Bscc06j0000000001	[REDACTED]	2010-01-28 09:51:52	2010-01-28 10:27:49	672	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.21
Bscc06k0000000005	[REDACTED]	2010-01-28 09:56:00	2010-01-28 09:49:55	775	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.17
Bscc06l0000000002	[REDACTED]	2010-01-28 09:56:45	2010-01-28 10:18:18	4593	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.9
Bscc06m0000000004	[REDACTED]	2010-01-28 08:01:42	2010-01-28 10:29:02	5077	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.10
Bscc06n0000000006	[REDACTED]	2010-01-28 07:46:45	2010-01-28 08:05:07	13398	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Carrier	172.17.91.8
Bscc06o0000000008	[REDACTED]	2010-01-27 22:12:01	2010-01-27 22:16:35	241	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.7
Bscc06p0000000001	[REDACTED]	2010-01-27 22:12:01	2010-01-27 22:13:01	772	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Carrier	172.17.91.2
Bscc06q0000000003	[REDACTED]	2010-01-27 22:13:02	2010-01-27 22:20:03	773	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.13
Bscc06r0000000004	[REDACTED]	2010-01-27 22:14:02	2010-01-27 22:21:21	5033	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	User+Request	172.17.91.130
Bscc06s0000000005	[REDACTED]	2010-01-27 22:15:26	2010-01-27 22:24:26	6333	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	NAS+Reboot	172.17.91.130
Bscc06t0000000006	[REDACTED]	2010-01-27 22:16:45	2010-01-27 22:24:14	6533	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	NAS+Reboot	172.17.91.130
Bscc06u0000000007	[REDACTED]	2010-01-27 22:17:45	2010-01-27 22:35:27	2074	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	NAS+Reboot	172.17.91.130
Bscc06v0000000001	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:23:05	8076	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.152
Bscc06w0000000002	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:20:45	9960	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.160
Bscc06x0000000004	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:21:06	10660	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.156
Bscc06y0000000006	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:25:07	5659	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.154
Bscc06z0000000008	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:33:45	1876	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.130
Bscc06a0000000009	[REDACTED]	2010-01-28 01:31:12:52	2010-02-01 02:13:49:09	61120	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	[REDACTED]	Lost+Carrier	172.17.91.130

Gambar 5.2 Halaman data *accounting user*

c. Halaman pencarian *user* nakal

Halaman ini digunakan administrator untuk mencari *user* yang melakukan tindak kejahatan menggunakan jaringan komputer *internal* UB.



Gambar 5.3 Halaman pencarian *user* nakal

5.2 Pengujian dan Analisa Hasil

Pengujian dan analisis hasil dilakukan untuk mengetahui apakah sistem yang telah dirancang dan diimplementasikan telah bekerja sesuai dengan tujuan pembuatan sistem. Pengujian dan analisis dilakukan secara bertahap sebagaimana urutan permasalahan yang ada di bagian rumusan masalah.

5.2.1 Proses Accounting

Pengujian proses *accounting* bertujuan untuk mengetahui proses *accounting* yang terjadi pada sistem. Selain itu pengujian ini juga bertujuan untuk membandingkan teori dengan hasil pengujian yang didapatkan.

5.2.1.1 Prosedur Pengujian

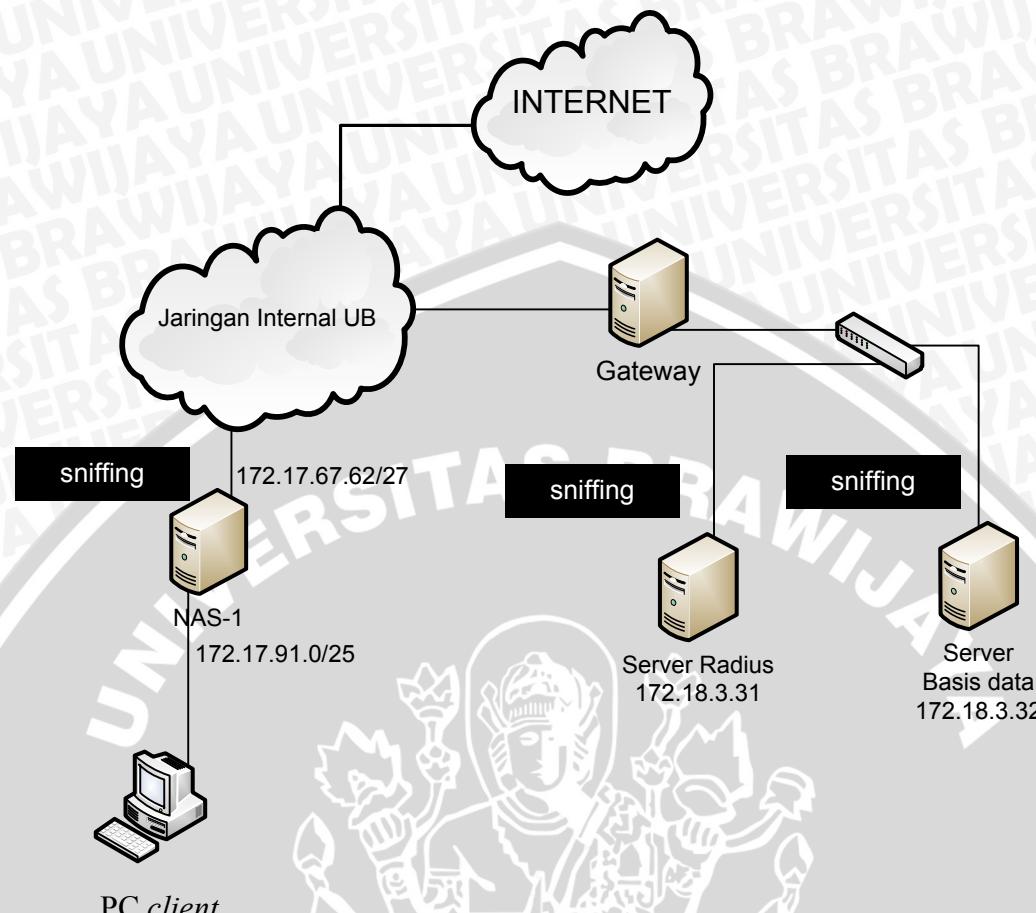
Pada tahap ini pengujian ini dilakukan dua kali, yaitu pada saat user melakukan *login* dan pada saat user melakukan *logout*. Pengujian pada saat user melakukan *login* dilakukan dengan:

- Menjalankan Freeradius pada *server RADIUS* dijalankan pada *debug mode*, sehingga dapat menampilkan proses-proses yang terjadi pada Radius.
- Menjalankan *server basis data*
- Mengkoneksikan sebuah komputer PC *client* pada jaringan di bawah NAS-1. Kemudian PC tersebut akan mendapatkan alamat IP yang diberikan oleh NAS-1 yang juga berfungsi sebagai DHCP server.
- Selanjutnya membuka browser pada PC *client* akan untuk melakukan *login* ke jaringan komputer. Pada browser terdapat form untuk mengisikan *username* dan *password user*. Kemudian PC *client* melakukan *login* ke jaringan komputer dengan mengisikan *username* dan *password*.
- Pada saat yang sama, dilakukan *sniffing* pada NAS-1, *server RADIUS* dan *server basis data*. Dengan *sniffing* ini kita dapat menganalisis paket dengan menggunakan aplikasi *tcpdump*. Aplikasi ini akan menampilkan paket-paket yang masuk dan keluar melalui interface tertentu.

Setelah melakukan pengujian pada saat user melakukan *login*, dilakukan pengujian *accounting* pada saat user melakukan *logout* dengan prosedur:

- Melakukan proses *logout* (keluar) pada PC *client*.
- Pada saat yang sama, dilakukan *sniffing* pada NAS-1, *server RADIUS* dan *server basis data*. Dengan *sniffing* ini kita dapat menganalisis paket dengan menggunakan aplikasi *tcpdump*. Aplikasi ini akan menampilkan paket-paket yang masuk dan keluar melalui interface tertentu.

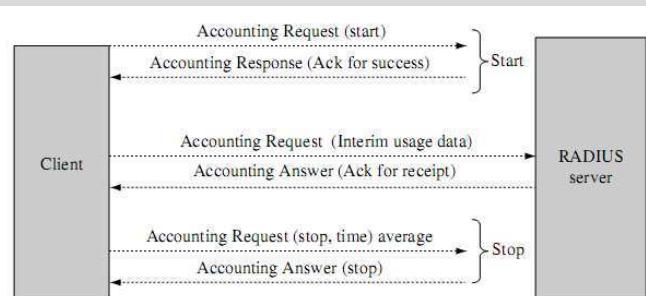
Gambar di bawah ini menjelaskan prosedur pengujian untuk mengetahui proses *accounting* yang terjadi.



Gambar 5.4 Prosedur pengujian untuk mengetahui proses accounting

5.2.1.2 Hasil yang diharapkan

Pada pengujian ini diharapkan proses *accounting* yang terjadi pada sistem yang diterapkan sesuai dengan dasar teori yang ada yaitu sebagai berikut:



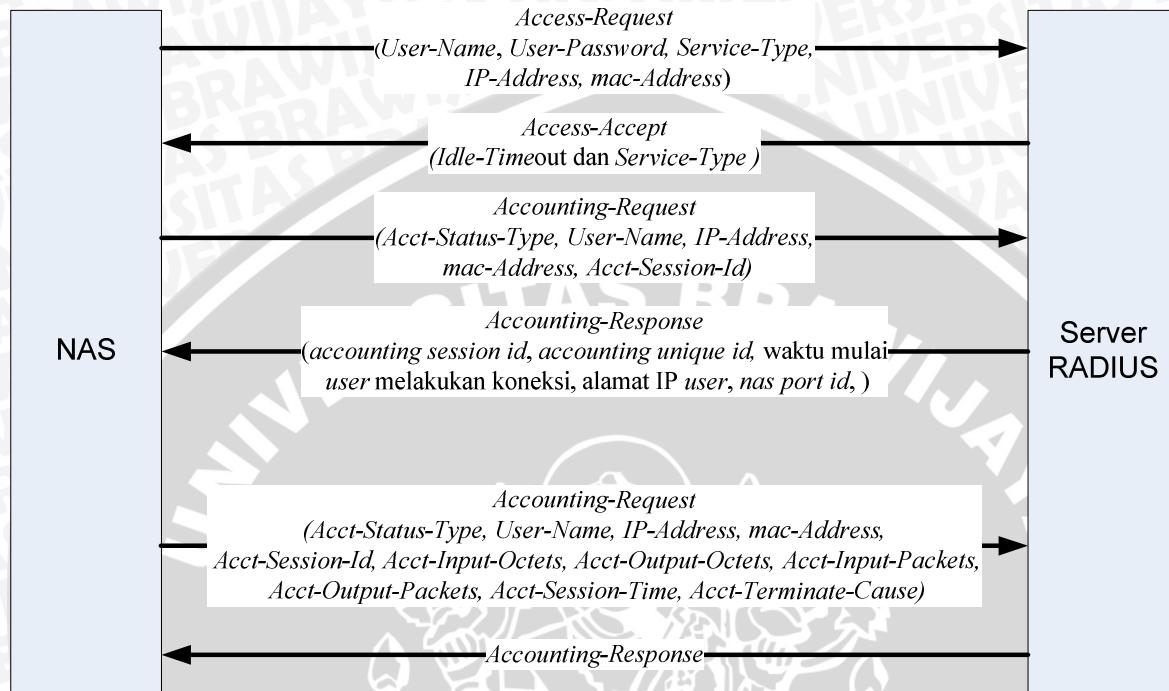
Gambar 5.5 Proses accounting yang diharapkan

5.2.1.3 Hasil Pengujian dan Analisis Hasil

Pada pengujian ini dapat disimpulkan bahwa proses yang terjadi adalah sebagai berikut.

- NAS melakukan autentikasi user lalu mengirimkan paket *Access-Request* kepada *server RADIUS*. Paket tersebut berisi *User-Name*, *User-Password*, *Service-Type*, *IP-Address*, *mac-Address*, beserta atribut lain yang menyertainya.
- *Server RADIUS* kemudian menerima paket tersebut dan melakukan otentikasi user dengan mencari data yang sesuai dengan atribut-atribut di paket *Access-Request* pada *server basis data*.
- Setelah *record* ditemukan, *server RADIUS* memeriksa bagaimana otorisasi user lalu mengirimkan paket *Access-Accept* yang berisi atribut otorisasi seperti *Idle-Timeout* dan *Service-Type* kepada NAS. Paket yang dikirim dan diterima NAS tampak pada gambar 5.2. Sedangkan paket yang diterima dan dikirim oleh server Radius tampak pada gambar 5.3.
- Setelah itu NAS mengirimkan paket *Accounting-Request* kepada *server RADIUS*. Paket tersebut berisi *Acct-Status-Type*, *User-Name*, *IP-Address*, *mac-Address*, *Acct-Session-Id* beserta atribut lain yang menyertainya.
- *Server RADIUS* kemudian menerima paket tersebut dan melakukan penambahan data status user berupa username yang sedang melakukan koneksi, *accounting session id*, *accounting unique id*, waktu mulai *user* melakukan koneksi, alamat IP *user*, *nas port id*, dan lain sebagainya pada tabel raddact di *server basis data*.
- Setelah itu, *server RADIUS* mengirimkan paket *Accounting-Response* dan *user* dapat melakukan koneksi ke jaringan komputer.
- Pada saat *user* memutuskan koneksi NAS mengirimkan paket *Accounting-Request* kepada *server RADIUS*. Paket tersebut berisi *Acct-Status-Type*, *User-Name*, *IP-Address*, *mac-Address*, *Acct-Session-Id*, *Acct-Input-Octets*, *Acct-Output-Octets*, *Acct-Input-Packets*, *Acct-Output-Packets*, *Acct-Session-Time*, *Acct-Terminate-Cause*, beserta atribut lain yang menyertainya.
- *Server RADIUS* kemudian menerima paket tersebut dan melakukan update data status user sesuai dengan atribut-atribut paket *Accounting-Request* pada tabel raddact di *server basis data*.

- Setelah itu, *server RADIUS* mengirimkan paket *Accounting-Response* yang menandakan bahwa data accounting user telah berhasil di simpan dan NAS dapat menghapus *session user* tersebut.



Gambar 5.6 Proses accounting pada pengujian

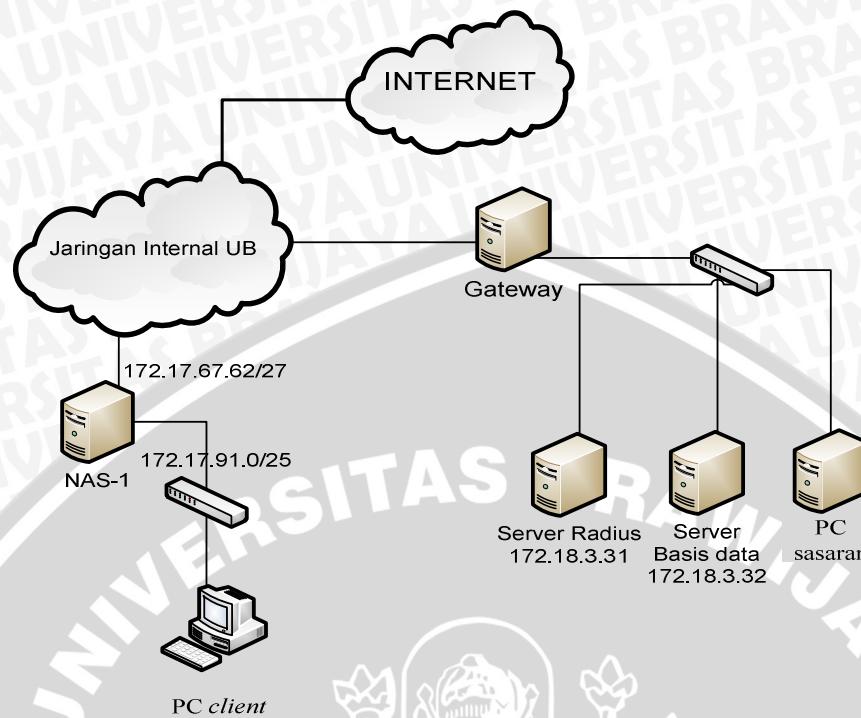
Dari hasil diatas dapat disimpulkan bahwa proses yang terjadi sistem ini sesuai dengan teori yang ada pada dasar teori.

5.2.2 Pelacakan (*tracking*)

Tujuan pengujian ini adalah untuk melakukan pelacakan (*tracking*) siapa dan dimana tempat user yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya. Pada pengujian ini dilakukan simulasi tindak kejahatan sederhana yaitu melakukan DoS berupa ping *flood*.

5.2.2.1 Prosedur Pengujian

Untuk menguji bagaimana pelacakan (*tracking*) yang dihasilkan dari pengimplementasian sistem, dilakukan tahap-tahap sebagai berikut:



Gambar 5.7 Prosedur pengujian untuk pelacakan (*tracking*)

- Mengaktifkan fungsi router sebagai NAS dengan cara mengaktifkan *firewall* dan *service* Chillispot.
- Alamat IP pada ketiga *client* diset dinamis atau sebagai *client* DHCP kemudian mengkoneksikan komputer uji (PC *client*) pada *router*.
- Dalam pengujian ini PC *client* melakukan suatu serangan DoS berupa ping *flood* ke PC sasaran.
- Pada PC sasaran dipasang sebuah *firewall* yang dapat menyimpan *log*.

5.2.2.2 Hasil yang diharapkan

Pada pengujian ini diharapkan kita menemukan secara pasti siapa dan dimana tempat user yang melakukan tindak kejahatan melalui jaringan internal UB pada simulasi yang dilakukan dengan cara melakukan pelacakan (*tracking*).

5.2.2.3 Hasil Pengujian dan Analisis Hasil

Data mengenai *accounting user* di *server RADIUS* disimpan pada *server basis data* yaitu pada tabel radacct. Dari pengamatan data yang telah masuk pada basis data dapat disimpulkan fungsi dari masing-masing kolom di tabel radacct adalah sebagai berikut :

Tabel 5.1 Fungsi tiap kolom pada tabel radacct

Kolom	Fungsi
Radacctid	menyimpan data penomoran data <i>accounting</i>
acctsessionid	menyimpan data <i>session id user</i> ketika melakukan koneksi
acctuniqueid	menyimpan data <i>unique id user</i> ketika melakukan koneksi
Username	menyimpan data <i>username</i> yang melakukan koneksi
Nasportid	menyimpan data nas <i>port id</i> yang digunakan <i>user</i> ketika melakukan koneksi
nasporttype	menyimpan data nas <i>port type</i> yang digunakan <i>user</i> ketika melakukan koneksi
acctstarttime	menyimpan data waktu mulai <i>user</i> melakukan koneksi
acctstoptime	menyimpan data waktu selesai <i>user</i> melakukan koneksi
acctsessiontime	menyimpan data lama waktu <i>user</i> melakukan koneksi
acctinbytes	menyimpan data besarnya data <i>input</i> yang dipakai <i>user</i> ketika melakukan koneksi
acctoutbytes	menyimpan data besarnya data <i>output</i> yang dipakai <i>user</i> ketika melakukan koneksi
calledstationid	menyimpan data lama alamat mac nas yang menjadi <i>gateway user</i>
callingstationid	menyimpan data lama alamat mac <i>user</i> yang melakukan koneksi
acctterminatecause	menyimpan data sebab putusnya koneksi <i>user</i>
framedipaddress	menyimpan data alamat ip <i>user</i> ketika melakukan koneksi

Sedangkan dari serangan yang dilakukan didapat hasil *log firewall* sebagai berikut berikut :

```
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12966
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12967
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12968
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12969
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
```

```
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12970
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12971
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12972
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12973
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12974
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12975
Dec  1 18:57:28 localhost kernel: IN=eth0 OUT=
MAC=00:1a:64:70:00:a4:00:02:1b:fd:03:53:08:00 SRC=172.17.91.2
DST=172.18.3.184 LEN=84 TOS=0x00 PREC=0x00 TTL=54 ID=0 DF PROTO=ICMP
TYPE=8 CODE=0 ID=27401 SEQ=12976
```

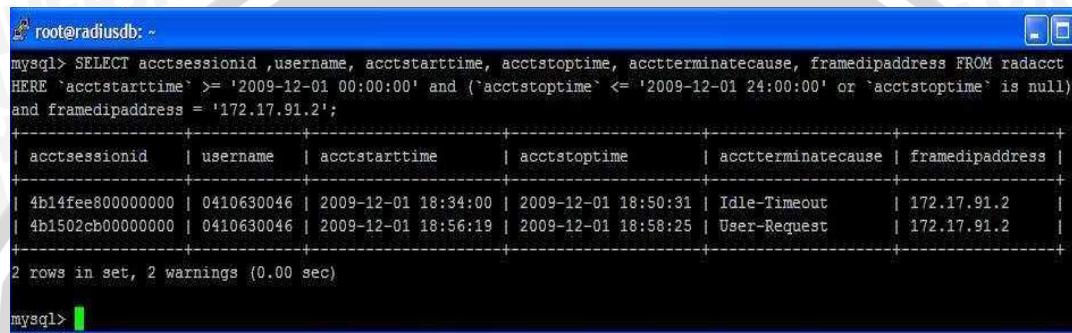
Pada hasil *log firewall* diatas dapat dilihat bahwa telah ada yang melakukan DoS berupa ping *flood* ke PC sasaran yang berasal dari alamat IP 172.17.91.2 dan serangan itu dilakukan pada pada 1 Desember 2009 dan sekitar pukul 18:57. Dari hasil log tersebut, kita akan mencoba melakukan *tracking* siapa, kapan dan dimana tempat orang yang melakukan kegiatan tersebut.

Dari tabel diatas, dapat kita simpulkan bahwa untuk mengetahui siapa dan dimana tempat orang yang melakukan DoS dapat melihat dari username dan alamat mac baik nas maupun user yang bersangkutan. Pada studi kasus diatas diketahui ada user yang telah melakukan DoS berupa ping *flood* ke PC sasaran yang berasal dari alamat IP 172.17.91.2 dan serangan itu dilakukan pada pada 1 Desember 2009 dan sekitar pukul 18:57. Dari data tersebut maka kita dapat melakukan *tracking* dengan cara melakukan query ke basis data dengan cara berikut

- Melakukan pencarian siapa user yang melakukan koneksi pada tanggal Desember 2009 dan menggunakan alamat IP 172.17.91.2 dengan cara query ke basis data radius dan pada kolom radacct sebagai berikut :

```
mysql> SELECT acctsessionid ,username, acctstarttime, acctstoptime, acctterminatecause, framedipaddress FROM radacct WHERE `acctstarttime` >= '2009-12-01 00:00:00' and (`acctstoptime` <= '2009-12-01 24:00:00' or `acctstoptime` is null) and framedipaddress = '172.17.91.2';
```

Dari query diatas maka akan ditampilkan data seperti di bawah :



The screenshot shows a terminal window with a blue header bar containing the text "root@radiusdb: ~". Below the header, the MySQL command is entered:

```
mysql> SELECT acctsessionid ,username, acctstarttime, acctstoptime, acctterminatecause, framedipaddress FROM radacct WHERE `acctstarttime` >= '2009-12-01 00:00:00' and (`acctstoptime` <= '2009-12-01 24:00:00' or `acctstoptime` is null) and framedipaddress = '172.17.91.2';
```

The output displays two rows of data:

acctsessionid	username	acctstarttime	acctstoptime	acctterminatecause	framedipaddress
4b14fee800000000	0410630046	2009-12-01 18:34:00	2009-12-01 18:50:31	Idle-Timeout	172.17.91.2
4b1502cb00000000	0410630046	2009-12-01 18:56:19	2009-12-01 18:58:25	User-Request	172.17.91.2

Below the table, the message "2 rows in set, 2 warnings (0.00 sec)" is displayed.

Gambar 5.8 Hasil query pengujian

Tabel berikut adalah salinan dari hasil di atas.

Tabel 5.2 Tabel hasil pelacakan

acctsessionid	Username	Acctstarttim e	acctstoptime	Accttermin atecause	framedipaddress
4b14fee800000000	0410630046	2009-12-01 18:34:00	2009-12-01 18:50:31	Idle- Timeout	172.17.91.2
4b1502cb00000000	0410630046	2009-12-01 18:56:19	2009-12-01 18:58:25	User- Request	172.17.91.2

- Berdasar hasil *query* kita telah mendapat 2 user yang melakukan koneksi pada tanggal 1 Desember 2009 dengan alamat IP 172.17.91.2. Untuk mengetahui user mana yang telah melakukan serangan kita tinggal melihat berdasarkan waktu mulai dan waktu terputusnya koneksi user tersebut, mana yang paling mendekati pukul 18:57. Dari tabel diatas dapat kita lihat data yang paling mendekati adalah pada data baris ke-2. Jadi user yang telah melakukan serangan adalah user dengan username 0410430046.

- Selain dengan menggunakan query seperti di atas dapat pul melakukan pencarian dengan aplikasi koneksi user yang telah dibuat. Dengan aplikasi ini dapat mempermudah kita dalam pencarian user yang melakukan tindak kejahatan.

The screenshot shows a search interface for accounting users. At the top, there are buttons for 'Logout' and 'Data Accounting User'. Below them is a text input field labeled 'Masukkan Alamat IP dan Tanggal Kejadian'. Underneath are two input fields: 'Alamat IP' containing '172.17.91.2' and 'Tanggal' containing '2009-12-01'. A 'Cari' button is located below these fields. The main area displays a table with one row of data. The table has columns for Session ID, Username, Andi_Agus, Waktu_login, Waktu_logout, Jumlah_Koneksi, Sesi_Id_User, Paket_Download, PAC_NAS, PAC_User, Penyebab_Jadi_Koneksi, and Alamat_IP_User. The data row is:

Session ID	username	Andi_Agus	Waktu_login	Waktu_logout	Jumlah_Koneksi	Sesi_Id_User	Paket_Download	PAC_NAS	PAC_User	Penyebab_Jadi_Koneksi	Alamat_IP_User
4b1502cb00000000	0410630046	2009-12-01 18:56:19	2009-12-01 18:58:25	126	1143819	1130358	00-60-1C-3C-1B-F7	00-03-47-BA-BB-8A	User-Request	172.17.91.2	
#b1/fee8000000000	0410630046	2009-12-01 18:34:00	2009-12-01 18:50:31	991	754872	833454	00-60-1C-3C-1B-F7	00-03-47-BA-BB-8A	Idle-TIMEOUT	172.17.91.2	

Gambar 5.9 Hasil pencarian menggunakan aplikasi

Dari langkah-langkah diatas maka kita dapat mengetahui bahwa pelakunya adalah orang yang dengan username 0410430046 yaitu mahasiswa yang memiliki Nomor Induk Mahasiswa 0410430046. User tersebut melakukan koneksi mulai pukul 18:56:19 sampai dengan pukul 18:58:25 pada tanggal 1 Desember 2009. Serangan tersebut dilakukan di network 172.17.91.0/25 yaitu network gedung kemahasiswaan teknik Elektro.

Dari hasil diatas dapat disimpulkan bahwa pelacakan (*tracking*) siapa dan dimana tempat *user* yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya dilakukan dengan syarat alamat IP dan waktu kejadian diketahui.

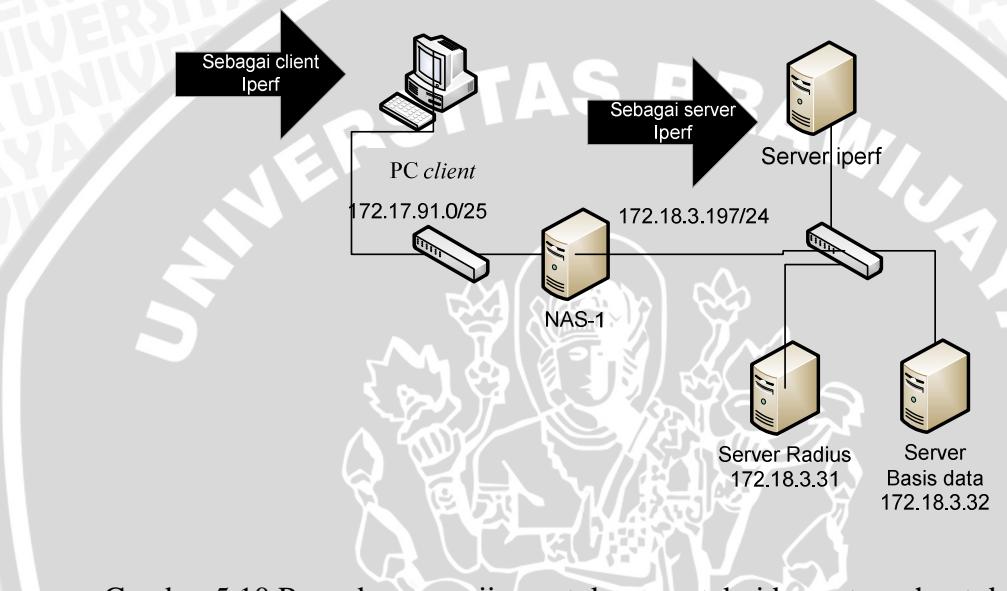
5.2.3 Troughput

Tujuan pengujian ini adalah untuk membandingkan besar *throughput* pada saat sebelum diterapkan sistem dan saat sesudah diterapkan sistem.

5.2.3.1 Prosedur Pengujian

Untuk menguji besar *throughput* pada saat sebelum diterapkan sistem dilakukan tahap-tahap sebagai berikut:

- Mematikan fungsi NAS pada *router*. Hal ini dilakukan dengan mematikan *firewall* pada *router* dan menghentikan *service* Chillispot pada *router*.
- Alamat IP client, dalam hal ini PC *client*, diset statis menjadi alamat IP yang masih pada segmen jaringan tersebut. Ketiganya kemudian dikoneksikan ke *router* dengan kabel UTP. Gambar berikut ini menjelaskan prosedur pengujian untuk mendapatkan besar *throughput* data.



Gambar 5.10 Prosedur pengujian untuk mengetahui besar throughput data

- Pada pengujian ini digunakan *tool* Iperf untuk mengukur besar *throughput* data. Server Radius diperlakukan sebagai *server* Iperf. Kemudian PC(A) diperlakukan sebagai *client* Iperf.
- Pengambilan data dilakukan sebanyak 10 kali secara berurutan untuk mendapatkan hasil yang bervariasi.

Untuk menguji besar *throughput* pada saat user sudah terautentikasi dilakukan tahap-tahap sebagai berikut:

- Mengaktifkan fungsi router sebagai NAS dengan cara mengaktifkan *firewall* dan *service* Chillispot.
- Alamat IP pada ketiga *client* diset dinamis atau sebagai *client* DHCP kemudian mengkoneksikan komputer uji (PC *client*) pada *router*.

- Dalam pengujian ini juga digunakan *tool* Iperf. *Server RADIUS* diperlakukan sebagai *server* Iperf. Sedangkan PC *client* diperlakukan sebagai *client* Iperf.
- Pengambilan data dilakukan sebanyak 10 kali secara berurutan untuk mendapatkan hasil yang bervariasi.

5.2.3.2 Hasil yang diharapkan

Pada pengujian ini diharapkan besar *troughput* data pada saat sesudah diterapkan sistem hampir sama dengan saat sebelum diterapkan sistem.

5.2.3.3 Hasil Pengujian dan Analisis Hasil

Dari pengujian besarnya *troughput* pada saat sebelum dan sesudah diterapkan sistem, didapat hasil sebagai berikut:

Tabel 5.3 Hasil pengujian besarnya *troughput* sebelum diterapkan sistem

No,	Troughput (Mbps)
1	89
2	82.7
3	85.3
4	83.3
5	85.5
6	58.9
7	77.5
8	78.7
9	84.9
10	74.3
Jumlah	800.1
Rata-rata	80.01
N. Tertinggi	89.00
N. Terendah	58.90
Standar Deviasi	8.61

Tabel 5.4 Hasil pengujian besarnya throughput sesudah diterapkan sistem

No.	Troughput (Mbps)
1	84
2	89.4
3	79
4	84.8
5	89.3
6	79.3
7	77.7
8	73
9	74
10	69
Jumlah	799.5
Rata-rata	79.95
N. Tertinggi	89.40
N. Terendah	69.00
Standar Deviasi	6.89

Karena besarnya rata-rata *throughput* sebelum diterapkan sistem lebih besar daripada sesudah diterapkan sistem, namun selisihnya sangat sedikit sekali.

Besarnya prosentase penurunan *throughput* adalah sebagai berikut :

$$\begin{aligned}\text{prosentase penurunan } & \text{throughput} = ((\text{Sebelum} - \text{Sesudah}) / \text{sebelum}) \times 100\% \\ & = ((80.01 - 79.95) / 80.01) \times 100\% \\ & = 0.07\%\end{aligned}$$

Karena besarnya penurunan *throughput* sangat kecil sekali mendekati 0% maka dapat disimpulkan bahwa dengan sistem ini, besarnya nilai *throughput* sesudah diterapkan sistem hampir sama dengan besarnya nilai *throughput* sebelum diterapkan sistem.

BAB VI

PENUTUP

6.1 Kesimpulan

Dari hasil pengujian dan analisis perancangan implementasi sistem *accounting user* jaringan komputer secara terpusat menggunakan protokol RADIUS, dapat disimpulkan bahwa :

1. Proses yang terjadi pada prototipe sistem *accounting* jaringan kumputer UB dari hasil analisis sesuai dengan teori yang ada pada dasar teori.
2. Dari hasil pengujian diperoleh data-data *accounting* antara lain data penomoran data *accounting*, *session id user*, *unique id user*, *username* yang melakukan koneksi, nas *port id* yang digunakan *user*, nas *port type* yang digunakan *user*, waktu mulai *user* melakukan koneksi, waktu selesai *user* melakukan koneksi, lama waktu *user* melakukan koneksi, besarnya data *input* yang dipakai *user*, besarnya data *output* yang dipakai *user*, alamat mac nas yang menjadi *gateway user*, alamat mac *user* yang melakukan koneksi, sebab putusnya koneksi *user*, dan dataalamat ip user ketika melakukan koneksi.
3. Pelacakan (*tracking*) siapa dan dimana tempat *user* yang melakukan tindak kejahatan melalui jaringan internal Universitas Brawijaya dilakukan dengan syarat alamat IP dan waktu kejadian diketahui.
4. Besar nilai *troughput* untuk satu *user* sesudah diterapkan sistem hampir sama dengan besarnya nilai *troughput* sebelum diterapkan sistem.

6.2 Saran

Saran yang dapat diberikan pada pengembangan sistem ini yaitu agar sistem *accounting* jaringan UB dirancang secara terpusat, yaitu semua NAS atau *client radius* harus berada pada setiap *end-router* yang melakukan proses autentikasi, otorisasi dan *accounting* pada satu *server RADIUS*. Hal ini bertujuan agar setiap user dapat menggunakan fasilitas penggunaan *resource* jaringan komputer UB di manapun dia berada dalam kawasan kampus. NAS atau *client radius* harus berada pada setiap *end-router* bertujuan agar NAS lebih mudah dalam menangani user yang berada pada jaringan dibawahnya.

Selain itu, dapat dibuat suatu sistem aplikasi yang dapat digunakan untuk memudahkan admin dalam konfigurasi server, management user, analisis data hasil *accounting* dan sebagai penghitung biaya akses user.

Hal lain yang bisa dilakukan adalah untuk diterapkan di Universitas Brawijaya dapat juga disenergikan dengan aplikasi lain yang telah ada seperti siakad atau e-mail brawijaya sehingga dengan satu id user saja user dapat melakukan akses internet sekaligus melakukan akses pada aplikasi lain tersebut.

DAFTAR PUSTAKA

[ARD-04]	Ardiyansah, Dian. 2004. Teknologi Jaringan Komputer. IlmuKomputer.Com.
[ARI-07]	Arif, Teuku Yuliar; Syahrial; dan Zulkiram. 2007. Studi Protokol Autentikasi pada Layanan Internet Service Provider (ISP). Jurnal Rekayasa Elektrika: Jurusan Teknik Elektro - Fakultas Teknik Universitas Syiah Kuala.
[ARY-08]	Aryanto, Joseph Rich. 2008. Graf dan Topologi Jaringan. Program Studi Teknik Informatika Institut Teknologi Bandung.
[BLA-04]	Blank, Andrew G. 2004. TCP/IP Foundations. Sybex: United States of America.
[DEW-03]	Dewo, E Setio. 2003. Bandwidth dan Throughput, IlmuKomputer.Com.
[DHO-07]	Dhoto. 2007. Jaringan Komputer. Politeknik Elektronika Negeri Surabaya - Institut Teknologi Sepuluh Nopember (PENS-ITS): Surabaya.
[HUS-04]	Husni. 2004. Implementasi Jaringan Komputer dengan Linux Redhat 9. Penerbit ANDI Yogyakarta: Yogyakarta.

[KIZ-05]	Kizza, Joseph Migga. 2005. Computer Network Security. Springer: United States of America.
[NAK-05]	Nakhjiri, Madjid and Nakhjiri, Mahsa. 2005. AAA And Network Security for Mobile Access (RADIUS, DIAMETER, EAP,PKI, and IP Mobility. John Willey & Sons Ltd: England.
[PRA-06]	Pradhana, Harindra Wisnu. 2006. Dasar-Dasar Web Design dengan PHP dan MySQL. Teknik Elektro Universitas Diponegoro.
[RFC-286]	Rigney, Carl. 2004. http://tools.ietf.org/html/rfc2866 , tanggal akses 4 Desember 2009.
[SET-05]	Setiawan, Agung W. 2005. Remote Authentication Dial In User Service (RADIUS) untuk Autentikasi Pengguna Wireless LAN. Departemen Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Bandung.
[STI-05]	Stiawan, D. 2005. Sistem Keamanan Komputer. PT. Elex Media Komputindo. Jakarta.
[SUP-07]	Supriyadi, Andi; dan Gartina, Dhani. 2007. Memilih Topologi Jaringan dan Hardware dalam Desain Sebuah Jaringan Komputer. Jurnal Informatika Pertanian Volume 16 No. 2, 2007.

[WAR-04]	Warsito. 2004. Sistem Keamanan Jaringan Multi-Domain Menggunakan Protokol DIAMETER. Program Magister Teknik Elektro Program Khusus Teknologi Informasi – Dikmenjur Institut Teknologi Bandung
[WIC-06]	Wicaksono, Sulistyo Unggul. 2006. Kajian Sistem Keamanan Jaringan CDMA. Program Studi Teknik Informatika, Institut Teknologi Bandung
[WKI-08]	Anonymous. 2008. http://id.wikipedia.org/wiki/OSI_Reference Model , Tanggal akses : 25 September 2008.
[WKI-08]	Anonymous. 2008. http://id.wikipedia.org/wiki/Protokol_(komputer) , Tanggal akses : 25 September 2008.

LAMPIRAN 1**SCRIPT KONFIGURASI SERVER RADIUS****/etc/freeradius/sql.conf**

```
## sql.conf -- SQL modules
sql {
    database = "mysql"
    driver = "rlm_sql_${database}"
    server = "172.18.3.197"
    login = "root"
    password = "joaneh2"
    radius_db = "easyhotspot"
    acct_table1 = "radacct"
    # Allow for storing data after authentication
    postauth_table = "radpostauth"
    authcheck_table = "radcheck"
    authreply_table = "radreply"
    groupcheck_table = "radgroupcheck"
    groupreply_table = "radgroupreply"
    # Table to keep group info
    usergroup_table = "usergroup"
    # If set to 'yes' (default) we read the group tables
    # If set to 'no' the user MUST have Fall-Through = Yes in the radreply
table
    read_groups = yes
    # Print all SQL statements when in debug mode (-x)
    sqltrace = no
    sqltracefile = ${logdir}/sqltrace.sql
    # number of sql connections to make to server
    num_sql_socks = 5
    # number of seconds to delay retrying on a failed database
    # connection (per_socket)
    connect_failure_retry_delay = 60
    # lifetime of an SQL socket. If you are having network issues
    # such as TCP sessions expiring, you may need to set the socket
    # lifetime. If set to non-zero, any open connections will be
```

```
# closed "lifetime" seconds after they were first opened.  
lifetime = 0  
  
# Maximum number of queries used by an SQL socket. If you are  
# having issues with SQL sockets lasting "too long", you can  
# limit the number of queries performed over one socket. After  
# "max_queries", the socket will be closed. Use 0 for "no limit".  
max_queries = 0  
  
# Table to keep radius client info  
nas_table = "nas"  
  
# Read driver-specific configuration  
$INCLUDE sql/${database}/dialup.conf  
}
```

/etc/freeradius/sites-enabled/default

```
authorize {  
  
    # The chap module will set 'Auth-Type := CHAP' if we are  
    # handling a CHAP request and Auth-Type has not already been set  
    chap  
  
    #  
    # If the users are logging in with an MS-CHAP-Challenge  
    # attribute for authentication, the mschap module will find  
    # the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'  
    # to the request, which will cause the server to then use  
    # the mschap module for authentication.  
    mschap  
  
    #  
    # If you are using multiple kinds of realms, you probably  
    # want to set "ignore_null = yes" for all of them.  
    # Otherwise, when the first style of realm doesn't match,  
    # the other styles won't be checked.  
    #  
    suffix  
        # The example below uses module failover to avoid querying all  
        # of the following modules if the EAP module returns "ok".  
        # Therefore, your LDAP and/or SQL servers will not be queried  
        # for the many packets that go back and forth to set up TTLS  
        # or PEAP. The load on those servers will therefore be reduced.
```

```
#  
eap {  
    ok = return  
}  
  
# Pull crypt'd passwords from /etc/passwd or /etc/shadow,  
# using the system API's to get the password. If you want  
# to read /etc/passwd or /etc/shadow directly, see the  
# passwd module in radiusd.conf.  
#  
unix  
  
    # See "Authorization Queries" in sql.conf  
sql  
noresetcounter  
octetslimit  
  
expiration  
logintime  
  
    # This module should be listed last, so that the other modules  
    # get a chance to set Auth-Type for themselves.  
#  
pap  
  
# Authentication.  
  
authenticate {  
    #  
    # PAP authentication, when a back-end database listed  
    # in the 'authorize' section supplies a password. The  
    # password can be clear-text, or encrypted.  
Auth-Type PAP {  
    pap  
}  
  
    #  
    # Most people want CHAP authentication
```

```
# A back-end database listed in the 'authorize' section
# MUST supply a CLEAR TEXT password. Encrypted passwords
# won't work.

Auth-Type CHAP {
    chap
}

#
# MSCHAP authentication.

Auth-Type MS-CHAP {
    mschap
}

#
# See 'man getpwent' for information on how the 'unix'
# module checks the users password. Note that packets
# containing CHAP-Password attributes CANNOT be authenticated
# against /etc/passwd! See the FAQ for details.

#
unix

# Allow EAP authentication.

eap
}

#
# Pre-accounting. Decide which accounting type to use.

#
preacct {
    preprocess
    suffix

    # Read the 'acct_users' file
    files
}

# Accounting. Log the accounting data.

#
```

```
accounting {  
    #  
    # Create a 'detail'ed log of the packets.  
    # Note that accounting requests which are proxied  
    # are also logged in the detail file.  
    detail  
        # If you don't use "radlast", you can delete this line.  
    unix  
        #  
        # For Simultaneous-Use tracking.  
        #  
        # Due to packet losses in the network, the data here  
        # may be incorrect. There is little we can do about it.  
    radutmp  
        #  
        # See "Accounting queries" in sql.conf  
    sql  
        # Filter attributes from the accounting response.  
        attr_filter.accounting_response  
}  
  
# Session database, used for checking Simultaneous-Use. Either the radutmp  
# or rlm_sql module can handle this.  
# The rlm_sql module is *much* faster  
session {  
    sql  
}  
# Post-Authentication  
# Once we KNOW that the user has been authenticated, there are  
# additional steps we can take.  
post-auth {  
  
    # See "Authentication Logging Queries" in sql.conf  
    sql  
    exec  
        # Add the ldap module name (or instance) if you have set  
        # 'edir_account_policy_check = yes' in the ldap module configuration  
        #  
    Post-Auth-Type REJECT {  
        attr_filter.access_reject
```

```
        }

# When the server receives a reply to a request it proxied
# to a home server, the request may be massaged here, in the
# post-proxy stage.

#
post-proxy {

    # You MUST also use the 'nostrip' option in the 'realm'
    # configuration. Otherwise, the User-Name attribute
    # in the proxied request will not match the user name
    # hidden inside of the EAP packet, and the end server will
    # reject the EAP request.

    #
eap
}
```

/etc/freeradius/sql/mysql/dialup.conf

```
## dialup.conf -- MySQL configuration for default schema (schema.sql)
sql_user_name = "%{User-Name}"
nas_query      = "SELECT id, nasname, shortname, type, secret FROM
${nas_table}"
authorize_check_query = "SELECT id, username, attribute, value, op \
    FROM ${authcheck_table} \
    WHERE username = '%{SQL-User-Name}' \
    ORDER BY id"
authorize_reply_query = "SELECT id, username, attribute, value, op \
    FROM ${authreply_table} \
    WHERE username = '%{SQL-User-Name}' \
    ORDER BY id"
group_membership_query = "SELECT groupname \
    FROM ${usergroup_table} \
    WHERE username = '%{SQL-User-Name}' \
    ORDER BY priority"
authorize_group_check_query = "SELECT id, groupname, attribute, \
    Value, op \
    FROM ${groupcheck_table} \
    WHERE groupname = '%{Sql-Group}' \\"
```

```
        ORDER BY id"

authorize_group_reply_query = "SELECT id, groupname, attribute, \
    value, op \
FROM ${groupreply_table} \
WHERE groupname = '%{Sql-Group}' \
ORDER BY id"

accounting_onoff_query = "\
UPDATE ${acct_table1} \
SET \
    acctstoptime      = '%S', \
    acctsessiontime   = unix_timestamp('%S') - \
                        unix_timestamp(acctstarttime), \
    acctterminatecause = '%{Acct-Terminate-Cause}', \
    acctstopdelay      = '%{Acct-Delay-Time}:0' \
WHERE acctstoptime IS NULL \
AND nasipaddress     = '%{NAS-IP-Address}' \
AND acctstarttime    <= '%S'"

accounting_update_query = "\
UPDATE ${acct_table1} \
SET \
    framedipaddress = '%{Framed-IP-Address}', \
    acctsessiontime  = '%{Acct-Session-Time}', \
    acctinputoctets  = '%{Acct-Input-Gigawords}:0' << 32 | \
                      '%{Acct-Input-Octets}:0', \
    acctoutputoctets = '%{Acct-Output-Gigawords}:0' << 32 | \
                      '%{Acct-Output-Octets}:0' \
WHERE acctsessionid = '%{Acct-Session-Id}' \
AND username         = '%{SQL-User-Name}' \
AND nasipaddress     = '%{NAS-IP-Address}'"

accounting_update_query_alt = "\
INSERT INTO ${acct_table1} \
(acctsessionid,      acctuniqueid,      username, \
realm,               nasipaddress,      nasportid, \
nasporttype,          acctstarttime,     acctsessiontime, \
acctauthentic,        connectinfo_start, acctinputoctets, \
acctoutputoctets,     calledstationid,   callingstationid, \
servicetype,          framedprotocol,   framedipaddress, \
acctstartdelay,       xascendsessionsrvkey) \
```

```
VALUES \
(''{Acct-Session-Id}'', ''{Acct-Unique-Session-Id}'', \
'{SQL-User-Name}'', \
'{Realm}'', ''{NAS-IP-Address}'', ''{NAS-Port}'', \
'{NAS-Port-Type}'', \
DATE_SUB(''S'', \
        INTERVAL (%%{Acct-Session-Time}:-0} + \
%%{Acct-Delay-Time}:-0)) SECOND), \
'{Acct-Session-Time}'', \
'{Acct-Authentic}'', '', \
'{%{Acct-Input-Gigawords}:-0}' << 32 | \
'{%{Acct-Input-Octets}:-0}'', \
'{%{Acct-Output-Gigawords}:-0}' << 32 | \
'{%{Acct-Output-Octets}:-0}'', \
'{Called-Station-Id}'', ''{Calling-Station-Id}'', \
'{Service-Type}'', ''{Framed-Protocol}'', \
'{Framed-IP-Address}'', \
'0', ''{X-Ascend-Session-Svr-Key}'')"

accounting_start_query = " \
INSERT INTO ${acct_table1} \
(acctsessionid,      acctuniqueid,      username, \
realm,               nasipaddress,      nasportid, \
nasporttype,          acctstarttime,     acctstoptime, \
acctsessiontime,     acctauthentic,     connectinfo_start, \
connectinfo_stop,    acctinputoctets,   acctoutputoctets, \
calledstationid,     callingstationid,  acctterminatecause, \
servicetype,          framedprotocol,   framedipaddress, \
acctstartdelay,      acctstopdelay,    xascendsessionsvrkey)\

VALUES \
(''{Acct-Session-Id}'', ''{Acct-Unique-Session-Id}'', \
'{SQL-User-Name}'', \
'{Realm}'', ''{NAS-IP-Address}'', ''{NAS-Port}'', \
'{NAS-Port-Type}'', ''S'', NULL, \
'0', ''{Acct-Authentic}'', ''{Connect-Info}'', \
'', '0', '0', \
'{Called-Station-Id}'', ''{Calling-Station-Id}'', '', \
'{Service-Type}'', ''{Framed-Protocol}'', ''{Framed-IP-Address}'', \
'{%{Acct-Delay-Time}:-0}'', '0', ''{X-Ascend-Session-Svr-Key}''")
```

```
accounting_start_query_alt = " \
    UPDATE ${acct_table1} SET \
        acctstarttime      = '%S', \
        acctstartdelay     = '%{{Acct-Delay-Time}:-0}', \
        connectinfo_start = '%{Connect-Info}' \
    WHERE acctsessionid = '%{Acct-Session-Id}' \
    AND username        = '%{SQL-User-Name}' \
    AND nasipaddress    = '%{NAS-IP-Address}'"

accounting_stop_query = " \
    UPDATE ${acct_table2} SET \
        acctstoptime      = '%S', \
        acctsessiontime   = '%{Acct-Session-Time}', \
        acctinprotoctets  = '%{{Acct-Input-Gigawords}:-0}' << 32 | \
                           '%{{Acct-Input-Octets}:-0}', \
        acctoutprotoctets = '%{{Acct-Output-Gigawords}:-0}' << 32 | \
                           '%{{Acct-Output-Octets}:-0}', \
        acctterminatecause = '%{Acct-Terminate-Cause}', \
        acctstopdelay      = '%{{Acct-Delay-Time}:-0}', \
        connectinfo_stop   = '%{Connect-Info}' \
    WHERE acctsessionid = '%{Acct-Session-Id}' \
    AND username        = '%{SQL-User-Name}' \
    AND nasipaddress    = '%{NAS-IP-Address}'"

accounting_stop_query_alt = " \
    INSERT INTO ${acct_table2} \
        (acctsessionid, acctuniqueid, username, \
         realm, nasipaddress, nasportid, \
         nasporttype, acctstarttime, acctstoptime, \
         acctsessiontime, acctauthentic, connectinfo_start, \
         connectinfo_stop, acctinprotoctets, acctoutprotoctets, \
         calledstationid, callingstationid, acctterminatecause, \
         servicetype, framedprotocol, framedipaddress, \
         acctstartdelay, acctstopdelay) \
    VALUES \
        ('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}', \
         '%{SQL-User-Name}', \
         '%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}', \
         '%{NAS-Port-Type}', \
         DATE_SUB('%S', \
                 INTERVAL (%{{Acct-Session-Time}:-0} + \
                           '%{{Acct-Delay-Time}:-0}) SECOND), \
         '%{Acct-Session-Time}', \
         '%{Acct-Delay-Time}', \
         '%{Connect-Info}', \
         '%{Acct-Input-Octets}', \
         '%{Acct-Output-Octets}', \
         '%{Acct-Input-Gigawords}', \
         '%{Acct-Output-Gigawords}', \
         '%{Acct-Terminate-Cause}', \
         '%{Acct-Delay-Time}', \
         '%{Connect-Info}', \
         '%{Acct-Session-Id}', \
         '%{SQL-User-Name}', \
         '%{NAS-IP-Address}')
```

```
'%S', '%{Acct-Session-Time}', '%{Acct-Authentic}', '', \  
'%{Connect-Info}', \  
'%{Acct-Input-Gigawords}:-0}' << 32 | \  
'%{Acct-Input-Octets}:-0}', \  
'%{Acct-Output-Gigawords}:-0}' << 32 | \  
'%{Acct-Output-Octets}:-0}', \  
'%{Called-Station-Id}', '%{Calling-Station-Id}', \  
'%{Acct-Terminate-Cause}', \  
'%{Service-Type}', '%{Framed-Protocol}', '%{Framed-IP-Address}', \  
'0', '%{Acct-Delay-Time}:-0})'"  
  
#####  
# Simultaneous Use Checking Queries  
#####  
# simul_count_query - query for the number of current connections  
# - If this is not defined, no simultaneous use checking  
# - will be performed by this module instance  
# simul_verify_query - query to return details of current connections for  
verification  
# - Leave blank or commented out to disable verification step  
# - Note that the returned field order should not be changed.  
#####  
  
simul_count_query = "SELECT COUNT(*) \  
    #FROM ${acct_table1} \  
    #WHERE username = '%{SQL-User-Name}' \  
    #AND acctstoptime IS NULL"  
simul_verify_query = "SELECT radacctid, acctsessionid, username, \  
    nasipaddress, nasportid, framedipaddress, \  
    callingstationid, framedprotocol \  
    FROM ${acct_table1} \  
    WHERE username = '%{SQL-User-Name}' \  
    AND acctstoptime IS NULL"  
  
#####  
# Authentication Logging Queries  
#####  
# postauth_query      - Insert some info after authentication  
#####  
  
postauth_query = "INSERT INTO ${postauth_table} \  
    (username, pass, reply, authdate) \  
VALUES (%s, %s, %s, %s)
```

```
        VALUES ( \
        '%{User-Name}', \
        '%{User-Password}:-%{Chap-Password}}}', \
        '%{reply:Packet-Type}', '%S'))
```

/etc/freeradius/sql/mysql/counter.conf

```
## counter.conf -- PostgreSQL queries for rlm_sqlcounter
##
sqlcounter dailycounter {
    counter-name = Daily-Session-Time
    check-name = Max-Daily-Session
    reply-name = Session-Timeout
    sqlmod-inst = sql
    key = User-Name
    reset = daily
    # This query properly handles calls that span from the
    # previous reset period into the current period but
    # involves more work for the SQL server than those
    # below
    query = "SELECT SUM(acctsessiontime - \
               GREATEST((%b - UNIX_TIMESTAMP(acctstarttime)), 0)) \
              FROM radacct WHERE username = '%{k}' AND \
              UNIX_TIMESTAMP(acctstarttime) + acctsessiontime > '%b'"
}
sqlcounter monthlycounter {
    counter-name = Monthly-Session-Time
    check-name = Max-Monthly-Session
    reply-name = Session-Timeout
    sqlmod-inst = sql
    key = User-Name
    reset = monthly
    # This query properly handles calls that span from the
    # previous reset period into the current period but
    # involves more work for the SQL server than those
    # below
    query = "SELECT SUM(acctsessiontime - \
               GREATEST((%b - UNIX_TIMESTAMP(acctstarttime)), 0)) \
              FROM radacct WHERE username=' %{k}' AND \
              UNIX_TIMESTAMP(acctstarttime) + acctsessiontime > '%b'"
```

```
}

sqlcounter noresetcounter {
    counter-name = Max-All-Session-Time
    check-name = Max-All-Session
    sqlmod-inst = sql
    key = User-Name
    reply-name = Session-Timeout
    reset = never
    query = "SELECT      SUM(AcctSessionTime)      FROM      radacct      WHERE
UserName= '%{<u>'"
}
sqlcounter octetslimit {
    counter-name = Max-All-MB
    check-name = Max-All-MB
    reply-name = Max-Total-Octets
    key = User-Name
    sqlmod-inst = sql
    reset = never
    query = "SELECT      SUM(acctinbytes+acctoutbytes)      FROM      radacct
WHERE UserName= '%{<u>'"
}
/etc/freeradius/clients.conf

## clients.conf -- client configuration directives
client localhost {
    ipaddr = 127.0.0.1
    secret = radiussecret
    require_message_authenticator = no
    nastype = other # localhost isn't usually a NAS...
}
-----INI CLIENT KAMI-----
client 172.17.67.62 {
    secret = radiussecret
    shortname = RISTIE
}

client 172.17.67.67 {
    secret = radiussecret
    shortname = gedungkuliah
}
```

LAMPIRAN 2

SCRIPT KONFIGURASI NAS

/etc/chilli.conf

```
net          172.17.91.128/25
dynip        172.17.91.128/25
dns1         202.162.208.99
dns2         202.162.208.100
domain       skripsiTEUB.brawijaya.ac.id
#radiuslisten 172.18.3.31
Radiusserver 1 172.18.3.31
radiusserver2 172.18.3.31
radiussecret   radiussecret
dhcpif        eth2
uamserver     https://172.17.91.129/cgi-bin/hotspotlogin.cgi
uamsecret      uamsecret
uamhomepage   http://172.17.91.129/login/
uamlisten     172.17.91.129
uamallowed    172.17.91.129
```

LAMPIRAN 3**DATA PENGUJIAN PROSES ACCOUNTING****RADIUS *debug***

```
root@radiusserver:~# freeradius -X > handokop1.debug

FreeRADIUS Version 2.1.6, for host i486-pc-linux-gnu, built on Jul 28 2009 at
22:15:13

Copyright (C) 1999-2009 The FreeRADIUS server project and contributors.
There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE.

You may redistribute copies of FreeRADIUS under the terms of the
GNU General Public License v2.

Starting - reading configuration files ...
including configuration file /etc/freeradius/radiusd.conf
including configuration file /etc/freeradius/clients.conf
including files in directory /etc/freeradius/modules/*
including configuration file /etc/freeradius/eap.conf
including configuration file /etc/freeradius/sql.conf
including configuration file /etc/freeradius/sql/mysql/dialup.conf
including configuration file /etc/freeradius/sql/mysql/counter.conf
including configuration file /etc/freeradius/policy.conf
including files in directory /etc/freeradius/sites-enabled/
including configuration file /etc/freeradius/sites-enabled/control-socket
including configuration file /etc/freeradius/sites-enabled/default
including configuration file /etc/freeradius/sites-enabled/inner-tunnel
group = freerad
user = freerad
including dictionary file /etc/freeradius/dictionary
main {
    prefix = "/usr"
    localstatedir = "/var"
    logdir = "/var/log/freeradius"
    libdir = "/usr/lib/freeradius"
    radacctdir = "/var/log/freeradius/radacct"
    hostname_lookups = no
```

```
max_request_time = 30
cleanup_delay = 5
max_requests = 1024
allow_core_dumps = no
pidfile = "/var/run/freeradius/freeradius.pid"
checkrad = "/usr/sbin/checkrad"
debug_level = 0
proxy_requests = yes

log {
    stripped_names = no
    auth = no
    auth_badpass = no
    auth_goodpass = no
}

security {
    max_attributes = 200
    reject_delay = 1
    status_server = yes
}
}

radiusd: ##### Loading Realms and Home Servers #####
home_server localhost {
    ipaddr = 127.0.0.1
    port = 1812
    type = "auth"
    secret = "testing123"
    response_window = 20
    max_outstanding = 65536
    require_message_authenticator = no
    zombie_period = 40
    status_check = "status-server"
    ping_interval = 30
    check_interval = 30
    num_answers_to_alive = 3
    num_pings_to_alive = 3
    revive_interval = 120
    status_check_timeout = 4
    irt = 2
    mrt = 16
    mrc = 5
```

```
mrd = 30
}

home_server_pool my_auth_failover {
    type = fail-over
    home_server = localhost
}
realm example.com {
    auth_pool = my_auth_failover
}
realm LOCAL {

}

radiusd: ##### Loading Clients #####
client localhost {
    ipaddr = 127.0.0.1
    require_message_authenticator = no
    secret = "radiussecret"
    nastype = "other"
}

client 172.17.67.62 {
    require_message_authenticator = no
    secret = "radiussecret"
    shortname = "RisTIE"
}

client 172.17.67.67 {
    require_message_authenticator = no
    secret = "radiussecret"
    shortname = "gedungkuliah"
}

radiusd: ##### Instantiating modules #####
instantiate {
Module: Linked to module rlm_exec
Module: Instantiating exec
exec {
    wait = no
    input_pairs = "request"
    shell_escape = yes
}
Module: Linked to module rlm_expr
Module: Instantiating expr
```

```
Module: Linked to module rlm_expiration
Module: Instantiating expiration
    expiration {
        reply-message = "Password Has Expired"
    }
Module: Linked to module rlm_logintime
Module: Instantiating logintime
    logintime {
        reply-message = "You are calling outside your allowed timespan"
        minimum-timeout = 60
    }
}

radiusd: ##### Loading Virtual Servers #####
server inner-tunnel {
    modules {
        Module: Checking authenticate {...} for more modules to load
        Module: Linked to module rlm_pap
        Module: Instantiating pap
            pap {
                encryption_scheme = "auto"
                auto_header = no
            }
        Module: Linked to module rlm_chap
        Module: Instantiating chap
        Module: Linked to module rlm_mschap
        Module: Instantiating mschap
            mschap {
                use_mppe = yes
                require_encryption = no
                require_strong = no
                with_ntdomain_hack = no
            }
        Module: Linked to module rlm_unix
        Module: Instantiating unix
            unix {
                radwtmp = "/var/log/freeradius/radwtmp"
            }
        Module: Linked to module rlm_eap
        Module: Instantiating eap
            eap {
```

```
    default_eap_type = "md5"
    timer_expire = 60
    ignore_unknown_eap_types = no
    cisco_accounting_username_bug = no
    max_sessions = 2048
}

Module: Linked to sub-module rlm_eap_md5
Module: Instantiating eap-md5
Module: Linked to sub-module rlm_eap_leap
Module: Instantiating eap-leap
Module: Linked to sub-module rlm_eap_gtc
Module: Instantiating eap-gtc
    gtc {
        challenge = "Password: "
        auth_type = "PAP"
    }
Module: Linked to sub-module rlm_eap_tls
Module: Instantiating eap-tls
    tls {
        rsa_key_exchange = no
        dh_key_exchange = yes
        rsa_key_length = 512
        dh_key_length = 512
        verify_depth = 0
        pem_file_type = yes
        private_key_file = "/etc/freeradius/certs/server.pem"
        certificate_file = "/etc/freeradius/certs/server.pem"
        CA_file = "/etc/freeradius/certs/ca.pem"
        private_key_password = "whatever"
        dh_file = "/etc/freeradius/certs/dh"
        random_file = "/etc/freeradius/certs/random"
        fragment_size = 1024
        include_length = yes
        check_crl = no
        cipher_list = "DEFAULT"
        make_cert_command = "/etc/freeradius/certs/bootstrap"
    cache {
        enable = no
        lifetime = 24
        max_entries = 255
    }
}
```

```
Module: Linked to sub-module rlm_eap_ttls
Module: Instantiating eap-ttls
ttls {
    default_eap_type = "md5"
    copy_request_to_tunnel = no
    use_tunneled_reply = no
    virtual_server = "inner-tunnel"
    include_length = yes
}
Module: Linked to sub-module rlm_eap_peap
Module: Instantiating eap-peap
peap {
    default_eap_type = "mschapv2"
    copy_request_to_tunnel = no
    use_tunneled_reply = no
    proxy_tunneled_request_as_eap = yes
    virtual_server = "inner-tunnel"
}
Module: Linked to sub-module rlm_eap_mschapv2
Module: Instantiating eap-mschapv2
mschapv2 {
    with_ntdomain_hack = no
}
Module: Checking authorize {...} for more modules to load
Module: Linked to module rlm_realm
Module: Instantiating suffix
realm suffix {
    format = "suffix"
    delimiter = "@"
    ignore_default = no
    ignore_null = no
}
Module: Linked to module rlm_files
Module: Instantiating files
files {
    usersfile = "/etc/freeradius/users"
    acctusersfile = "/etc/freeradius/acct_users"
    preproxy_usersfile = "/etc/freeradius/preproxy_users"
    compat = "no"
}
Module: Checking session {...} for more modules to load
Module: Linked to module rlm_radutmp
```

```
Module: Instantiating radutmp
radutmp {
    filename = "/var/log/freeradius/radutmp"
    username = "%{User-Name}"
    case_sensitive = yes
    check_with_nas = yes
    perm = 384
    callerid = yes
}

Module: Checking post-proxy {...} for more modules to load
Module: Checking post-auth {...} for more modules to load
Module: Linked to module rlm_attr_filter
Module: Instantiating attr_filter.access_reject
attr_filter attr_filter.access_reject {
    attrsfile = "/etc/freeradius/attrs.access_reject"
    key = "%{User-Name}"
}
} # modules
} # server
server {
modules {
Module: Checking authenticate {...} for more modules to load
Module: Checking authorize {...} for more modules to load
Module: Linked to module rlm_preprocess
Module: Instantiating preprocess
preprocess {
    huntgroups = "/etc/freeradius/huntgroups"
    hints = "/etc/freeradius/hints"
    with_ascend_hack = no
    ascend_channels_per_line = 23
    with_ntdomain_hack = no
    with_specialix_jetstream_hack = no
    with_cisco_vsa_hack = no
    with_alvarion_vsa_hack = no
}
Module: Linked to module rlm_sql
Module: Instantiating sql
```

```
sql {
    driver = "rlm_sql_mysql"
    server = "172.18.3.32"
    port =
    login = "koko"
    password = "sayangwinda"
    radius_db = "easyhotspot"
    read_groups = yes
    sqltrace = no
    sqltracefile = "/var/log/freeradius/sqltrace.sql"
    readclients = no

    deletestalesessions = yes
    num_sql_socks = 5
    lifetime = 0
    max_queries = 0
    sql_user_name = "%{User-Name}"
    default_user_profile = ""

nas_query = "SELECT id, nasname, shortname, type, secret FROM nas"
    authorize_check_query = "SELECT id, username, attribute, value, op
FROM radcheck
    WHERE username = '%{SQL-User-Name}' ORDER BY
id"

authorize_reply_query = "SELECT id, username, attribute, value, op
FROM radreply
    WHERE username = '%{SQL-User-Name}' ORDER BY
id"

authorize_group_check_query = "SELECT id, groupname, attribute,
Value, op
FROM radgroupcheck
    WHERE groupname = '%{Sql-
Group}' ORDER BY id"

authorize_group_reply_query = "SELECT id, groupname, attribute,
value, op
FROM radgroupreply
    WHERE groupname = '%{Sql-
Group}' ORDER BY id"

accounting_onoff_query = "
UPDATE radacct
SET
acctstoptime = '%S',
acctsessiontime =
unix_timestamp('%S')
acctterminatecause = '%{Acct-
Terminate-Cause}', acctstopdelay = %'{%{Acct-Delay-Time}:-0}
WHERE acctstoptime IS NULL
        AND nasipaddress = '%{NAS-IP-
```

```
Address}' AND acctstarttime <= '%S'"  
  
accounting_update_query = " UPDATE radacct SET framedipaddress = '%{Framed-IP-Address}', acctsessiontime = '%{Acct-Session-Time}',  
acctinbytes = '%{Acct-Input-Octets}', acctoutbytes = '%{Acct-Output-Octets}' << 32 |  
'%{Acct-Input-Gigawords}:-0}' << 32 |  
'%{Acct-Output-Gigawords}:-0}' << 32 |  
WHERE acctsessionid = '%{Acct-Session-Id}'  
AND username = '%{SQL-User-Name}' AND nasipaddress =  
'%{NAS-IP-Address}'"  
  
accounting_update_query_alt = " INSERT INTO radacct  
(acctsessionid,  
acctuniqueid,  
username,  
realm,  
nasportid,  
nasporttype,  
acctstarttime,  
acctauthentic,  
connectinfo_start,  
acctoutbytes,  
calledstationid,  
servicetype,  
framedprotocol,  
acctstartdelay,  
xascendsessionsvrkey)  
VALUES  
('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}',  
'%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}',  
'%{Service-Type}',  
DATE_SUB('%S',  
INTERVAL (%{Acct-Session-Time}:-0)  
SECOND),  
'%{Acct-Session-Time}',  
'%{Acct-Authentic}', '',  
'%{Acct-Input-Octets}',  
'%{Acct-Output-Octets}',  
32 |  
'%{Acct-Input-Gigawords}:-0}' << 32 |  
'%{Acct-Output-Gigawords}:-0}' <<  
'%{Called-Station-Id}', '%{Calling-Station-Id}',  
'%{Service-Type}',  
'%{Framed-Protocol}',  
'%{Framed-IP-Address}',  
'%{X-Ascend-Session-Svr-Key}'")  
  
accounting_start_query = "  
(acctsessionid, acctuniqueid,  
nasipaddress, nasportid,  
acctstoptime, acctstarttime,  
connectinfo_start, connectinfo_stop,  
acctoutbytes, acctinputbytes,  
acctterminatecause, servicetype,  
framedipaddress, framedprotocol,  
xascendsessionsvrkey)  
VALUES  
('%{Acct-Session-Id}', '%{Acct-Unique-Session-Id}',  
'%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}',  
'%{Service-Type}', '%S', NULL,  
'0', '%{Acct-Authentic}', '%{Connect-Info}',
```

```
'', '0', '0',          '%{Called-Station-Id}', '%{Calling-Station-Id}', '',
'%{Service-Type}',      '%{Framed-Protocol}',      '%{Framed-IP-Address}',
'%{{%{Acct-Delay-Time}:-0}}', '0', '%{X-Ascend-Session-Svr-Key}' )"
```



```
accounting_start_query_alt = "                                UPDATE radacct SET
acctstarttime      = '%S',                      acctstartdelay = '%{{%{Acct-Delay-
Time}:-0}}',
connectinfo_start = '%{Connect-Info}'           WHERE
acctsessionid     = '%{Acct-Session-Id}'        AND username      = '%{SQL-
User-Name}'          AND nasipaddress   = '%{NAS-IP-Address}''"
```



```
accounting_stop_query = "                                UPDATE radacct SET
acctstoptime       = '%S',                      acctsessiontime = '%{{Acct-Session-
Time}}',
acctinputoctets    = '%{{%{Acct-Input-Gigawords}:-0}}' << 32
|
acctoutputoctets  = '%{{%{Acct-Output-Gigawords}:-0}}' << 32 |
'%{{%{Acct-Output-Octets}:-0}}',
acctterminatecause = '%{{Acct-
Terminate-Cause}}',                      acctstopdelay = '%{{%{Acct-Delay-Time}:-0}}',
connectinfo_stop   = '%{Connect-Info}'           WHERE
acctsessionid     = '%{Acct-Session-Id}'        AND username      =
'%{SQL-User-Name}'          AND nasipaddress   = '%{NAS-IP-Address}''"
```



```
accounting_stop_query_alt = "                                INSERT INTO radacct
(acctsessionid, acctuniqueid, username,
nasportid,                  nasporttype, acctstarttime, acctstoptime,
acctsessiontime,             acctauthentic, connectinfo_start,
connectinfo_stop,            acctinputoctets, acctoutputoctets,
calledstationid,             callingstationid, acctterminatecause,
servicetype, framedprotocol, framedipaddress,
acctstopdelay)               VALUES ('%{Acct-Session-Id}', '%{Acct-
Unique-Session-Id}', '%{SQL-User-Name}', '%{Realm}', '%{NAS-IP-Address}', '%{NAS-Port}', '%{NAS-Port-Type}', DATE_SUB('%S', INTERVAL (%{{%{Acct-Session-Time}:-0}} +
'%{{%{Acct-Delay-Time}:-0}}) SECOND), '%S', '%{Acct-Session-Time}', '%{Acct-Authentic}', '', '%{Connect-Info}', '%{{%{Acct-Input-Gigawords}:-0}}' << 32 | '%{{%{Acct-Output-Gigawords}:-0}}' << 32 |
'%{{%{Acct-Output-Octets}:-0}}', '%{Called-Station-Id}', '%{Calling-
Station-Id}', '%{Acct-Terminate-Cause}', '%{Service-
Type}', '%{Framed-Protocol}', '%{Framed-IP-Address}', '0',
'%{{%{Acct-Delay-Time}:-0}}' )"
```

```
group_membership_query = "SELECT groupname FROM usergroup
WHERE username = '%{SQL-User-Name}' ORDER BY priority"
    connect_failure_retry_delay = 60
    simul_count_query = "SELECT COUNT(*) #FROM
radacct #WHERE username = '%{SQL-User-Name}'"
#AND acctstoptime IS NULL"

simul_verify_query = "SELECT radacctid, acctsessionid, username,
nasipaddress, nasportid, framedipaddress,
callingstationid, framedprotocol
FROM radacct
WHERE username = '%{SQL-User-Name}' AND
acctstoptime IS NULL"

postauth_query = "INSERT INTO radpostauth
(username, VALUES (
'%{User-Name}', '%{User-Password}:-%{Chap-
Password}}', '%{reply:Packet-Type}', '%S')"
safe-characters =
"@abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.-_:*"
}

rlm_sql (sql): Driver rlm_sql_mysql (module rlm_sql_mysql) loaded and linked
rlm_sql (sql): Attempting to connect to koko@172.18.3.32:/easyhotspot
rlm_sql (sql): starting 0
rlm_sql (sql): Attempting to connect rlm_sql_mysql #0
rlm_sql_mysql: Starting connect to MySQL server for #0
rlm_sql (sql): Connected new DB handle, #0
rlm_sql (sql): starting 1
rlm_sql (sql): Attempting to connect rlm_sql_mysql #1
rlm_sql_mysql: Starting connect to MySQL server for #1
rlm_sql (sql): Connected new DB handle, #1
rlm_sql (sql): starting 2
rlm_sql (sql): Attempting to connect rlm_sql_mysql #2
rlm_sql_mysql: Starting connect to MySQL server for #2
rlm_sql (sql): Connected new DB handle, #2
rlm_sql (sql): starting 3
rlm_sql (sql): Attempting to connect rlm_sql_mysql #3
rlm_sql_mysql: Starting connect to MySQL server for #3
rlm_sql (sql): Connected new DB handle, #3
rlm_sql (sql): starting 4
rlm_sql (sql): Attempting to connect rlm_sql_mysql #4
rlm_sql_mysql: Starting connect to MySQL server for #4
```

```
rlm_sql (sql): Connected new DB handle, #4
Module: Linked to module rlm_sqlcounter
Module: Instantiating noresetcounter

sqlcounter noresetcounter {
    counter-name = "Max-All-Session-Time"
    check-name = "Max-All-Session"
    reply-name = "Session-Timeout"
    key = "User-Name"
    sqlmod-inst = "sql"
    query = "SELECT SUM(AcctSessionTime) FROM radacct WHERE UserName='%\{%k\}'"
    reset = "never"
    safe-characters
"@abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.-_:*"
}

rlm_sqlcounter: Reply attribute Session-Timeout is number 27
rlm_sqlcounter: Counter attribute Max-All-Session-Time is number 11273
rlm_sqlcounter: Check attribute Max-All-Session is number 11274
rlm_sqlcounter: Current Time: 1009934612 [2002-01-02 08:23:32], Next reset 0
[2002-01-02 08:00:00]
rlm_sqlcounter: Current Time: 1009934612 [2002-01-02 08:23:32], Prev reset 0
[2002-01-02 08:00:00]
Module: Instantiating octetslimit

sqlcounter octetslimit {
    counter-name = "Max-All-MB"
    check-name = "Max-All-MB"
    key = "User-Name"
    sqlmod-inst = "sql"
    query = "SELECT SUM(acctinutoctets+acctoutputoctets) FROM radacct WHERE
UserName='%\{%k\}'"
    reset = "never"
    safe-characters
"@abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789.-_:*"
}

rlm_sqlcounter: Reply attribute set to Session-Timeout.
rlm_sqlcounter: Counter attribute Max-All-MB is number 11275
rlm_sqlcounter: Check attribute Max-All-MB is number 11275
rlm_sqlcounter: Current Time: 1009934612 [2002-01-02 08:23:32], Next reset 0
[2002-01-02 08:00:00]
```

```
rlm_sqlcounter: Current Time: 1009934612 [2002-01-02 08:23:32], Prev reset 0
[2002-01-02 08:00:00]

Module: Checking preacct {...} for more modules to load
Module: Linked to module rlm_acct_unique
Module: Instantiating acct_unique

acct_unique {
    key = "User-Name, Acct-Session-Id, NAS-IP-Address, Client-IP-Address,
NAS-Port"
}

Module: Checking accounting {...} for more modules to load
Module: Linked to module rlm_detail
Module: Instantiating detail
detail {
    detailfile = "/var/log/freeradius/radacct/%{Client-IP-Address}/detail-
%Y%m%d"
    header = "%t"
    detailperm = 384
    dirperm = 493
    locking = no
    log_packet_header = no
}
Module: Instantiating attr_filter.accounting_response
attr_filter attr_filter.accounting_response {
    attrsfile = "/etc/freeradius/attrs.accounting_response"
    key = "%{User-Name}"
}
Module: Checking session {...} for more modules to load
Module: Checking post-proxy {...} for more modules to load
Module: Checking post-auth {...} for more modules to load
} # modules
} # server

radiusd: ##### Opening IP addresses and Ports #####
listen {
    type = "auth"
    ipaddr = *
    port = 0
}
```

```
listen {
    type = "acct"
    ipaddr = *
    port = 0
}

listen {
    type = "control"
}

listen {
    socket = "/var/run/freeradius/freeradius.sock"
}
}

Listening on authentication address * port 1812
Listening on accounting address * port 1813
Listening on command file /var/run/freeradius/freeradius.sock
Listening on proxy address * port 1814
Ready to process requests.

rad_recv: Access-Request packet from host 172.18.3.218 port 59277, id=0,
length=201
    User-Name = "0410630046"
    User-Password = "koko"
    NAS-IP-Address = 0.0.0.0
    Service-Type = Login-User
    Framed-IP-Address = 172.17.91.131
    Calling-Station-Id = "00-02-55-EE-6E-8E"
    Called-Station-Id = "00-13-46-3B-0C-51"
    NAS-Identifier = "nas01"
    Acct-Session-Id = "4afdf257f00000001"
    NAS-Port-Type = Wireless-802.11
    NAS-Port = 1
    Message-Authenticator = 0xadd5a2d71cb777501fd1df788752c498
    WISPr-Logoff-URL = "http://172.17.91.129:3990/logoff"

-- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
```

```
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[unix] returns notfound
[sql] expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
rlm_sql (sql): Reserving sql socket id: 4
[sql] expand: SELECT id, username, attribute, value, op          FROM radcheck
WHERE username = '%{SQL-User-Name}'                                ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radcheck
WHERE username
= '0410630046'          ORDER BY id
[sql] User found in radcheck table
[sql] expand: SELECT id, username, attribute, value, op          FROM radreply
WHERE username = '%{SQL-User-Name}'                                ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radreply
WHERE username
= '0410630046'          ORDER BY id
[sql] expand: SELECT groupname          FROM usergroup          WHERE
username = '%{SQL-User-Name}'          ORDER BY priority -> SELECT groupname
FROM usergroup          WHERE username = '0410630046'          ORDER BY
priority
[sql] expand: SELECT id, groupname, attribute,          Value, op
FROM radgroupcheck          WHERE groupname = '%{Sql-Group}'          ORDER
BY id -> SELECT id, groupname, attribute,          Value, op
FROM radgroupcheck          WHERE groupname = 'packet'          ORDER BY id
[sql] User found in group packet
[sql] expand: SELECT id, groupname, attribute,          value, op
FROM radgroupreply          WHERE groupname = '%{Sql-Group}'          ORDER
BY id -> SELECT id, groupname, attribute,          value, op
FROM radgroupreply          WHERE groupname = 'packet'          ORDER BY id
rlm_sql (sql): Released sql socket id: 4
++[sql] returns ok
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[noresetcounter] returns noop
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[octetslimit] returns noop
++[expiration] returns noop
++[logintime] returns noop
++[pap] returns updated
Found Auth-Type = PAP
```

```
!!!!!!!  
!!! Replacing User-Password in config items with Cleartext-Password. !!!  
!!!!!!!  
!!! Please update your configuration so that the "known good" !!!  
!!! clear text password is in Cleartext-Password, and not in User-Password. !!!  
!!!!!!!  
+- entering group PAP {...}  
[pap] login attempt with password "koko"  
[pap] Using clear text password "koko"  
[pap] User authenticated successfully  
++[pap] returns ok  
+- entering group session {...}  
[sql] expand: %{User-Name} -> 0410630046  
[sql] sql_set_user escaped user --> '0410630046'  
[sql] expand: SELECT COUNT(*)  
#WHERE username = '%{SQL-User-Name}'  
acctstoptime IS NULL -> SELECT COUNT(*)  
radacct  
#AND acctstoptime IS NULL  
#FROM  
#WHERE username = '0410630046'  
rlm_sql (sql): Reserving sql socket id: 3  
[sql] expand: SELECT radacctid, acctsessionid, username,  
nasipaddress, nasportid,  
callingstationid, framedprotocol  
WHERE username = '%{SQL-User-Name}'  
acctstoptime IS NULL -> SELECT radacctid, acctsessionid, username,  
nasipaddress, nasportid,  
callingstationid, framedprotocol  
FROM radacct  
AND acctstoptime  
WHERE username = '0410630046'  
IS NULL  
rlm_sql (sql): Released sql socket id: 3  
++[sql] returns ok  
+- entering group post-auth {...}  
[sql] expand: %{User-Name} -> 0410630046  
[sql] sql_set_user escaped user --> '0410630046'  
[sql] expand: %{User-Password} -> koko  
[sql] expand: INSERT INTO radpostauth  
pass, reply, authdate)  
(username,  
VALUES (  
'%{User-Name}',  
'%{User-Password}:-%{Chap-  
Password}}',  
'%{reply:Packet-Type}', '%S') -> INSERT  
INTO radpostauth  
VALUES (  
'koko',  
'Access-Accept', '2002-01-02 08:25:08')
```

```
rlm_sql (sql) in sql_postauth: query is INSERT INTO radpostauth
(username, pass, reply, authdate)
'0410630046',
'koko',
'Access-Accept', '2002-01-02 08:25:08')
rlm_sql (sql): Reserving sql socket id: 2
rlm_sql (sql): Released sql socket id: 2
++[sql] returns ok
++[exec] returns noop
Sending Access-Accept of id 0 to 172.18.3.218 port 59277
    Idle-Timeout := 300
    Service-Type = Login-User
Finished request 0.
Going to the next request
Waking up in 4.9 seconds.
rad_recv: Accounting-Request packet from host 172.18.3.218 port 34525, id=4,
length=135
    Acct-Status-Type = Start
    User-Name = "0410630046"
    Calling-Station-Id = "00-02-55-EE-6E-8E"
    Called-Station-Id = "00-13-46-3B-0C-51"
    NAS-Port-Type = Wireless-802.11
    NAS-Port = 1
    NAS-Port-Id = "00000001"
    NAS-IP-Address = 0.0.0.0
    NAS-Identifier = "nas01"
    Framed-IP-Address = 172.17.91.131
    Acct-Session-Id = "4afdf257f00000001"
++ entering group preacct {...}
++[preprocess] returns ok
[acct_unique] Hashing 'NAS-Port = 1,Client-IP-Address = 172.18.3.218,NAS-IP-
Address = 0.0.0.0,Acct-Session-Id = "4afdf257f00000001",User-Name =
"0410630046"'
[acct_unique] Acct-Unique-Session-ID = "b5cf75304b7f3484".
++[acct_unique] returns ok
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
++[files] returns noop
++ entering group accounting {...}
[detail] expand: /var/log/freeradius/radacct/%{Client-IP-Address}/detail-
%Y%m%d -> /var/log/freeradius/radacct/172.18.3.218/detail-20020102
[detail] /var/log/freeradius/radacct/%{Client-IP-Address}/detail-%Y%m%d expands
```

```
to /var/log/freeradius/radacct/172.18.3.218/detail-20020102
[detail]      expand: %t -> Wed Jan  2 08:25:08 2002
++[detail] returns ok
++[unix] returns ok
[radutmp]      expand: /var/log/freeradius/radutmp -> /var/log/freeradius/radutmp
[radutmp]      expand: %{User-Name} -> 0410630046
rlm_radutmp: Login entry for NAS 0.0.0.0 port 1 wrong order
++[radutmp] returns ok
[sql] expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
[sql] expand: %{Acct-Delay-Time} ->
[sql] expand:           INSERT INTO radacct          (acctsessionid,
acctuniqueid,      username,             realm,            nasipaddress,
nasportid,         nasporttype,        acctstarttime,   acctstoptime,
acctsessiontime,   acctauthentic,       connectinfo_start,
connectinfo_stop,  acctinoutoctets,    acctterminatecause,
calledstationid,   callingstationid,   framedprotocol,
servicetype,       framingtype,        framedipaddress,
acctstartdelay,   acctstopdelay,     xascendsessionsvrkey) VALUES
(' %{Acct-Session-Id}', ' %{Acct-Unique-Session-Id}',      '%{SQL-User-
Name}',           '%{Realm}',        '%{NAS-IP-Address}',  '%{NAS-Port}',
'%{NAS-Port-Type}', '%S',  NULL,           '0',           '%{Acct-Authentic}',
'%{Connect-Info}',  '',   '0',   '0',           '%{Called-Station-
Id}',           '%{Calling-Station-Id}',   '',           '%{Service-Type}', '%{Framed-
Protocol}',       '%{Framed-IP-Address}', '',
rlm_sql (sql): Reserving sql socket id: 1
rlm_sql (sql): Released sql socket id: 1
++[sql] returns ok
[attr_filter.accounting_response]      expand: %{User-Name} -> 0410630046
attr_filter: Matched entry DEFAULT at line 12
++[attr_filter.accounting_response] returns updated
Sending Accounting-Response of id 4 to 172.18.3.218 port 34525
Finished request 1.

Cleaning up request 1 ID 4 with timestamp +96
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 0 ID 0 with timestamp +96
Ready to process requests.

rad_recv: Accounting-Request packet from host 172.18.3.218 port 34525, id=5,
length=183
    Acct-Status-Type = Stop
    User-Name = "0410630046"
```

```
Calling-Station-Id = "00-02-55-EE-6E-8E"
Called-Station-Id = "00-13-46-3B-0C-51"
NAS-Port-Type = Wireless-802.11
NAS-Port = 1
NAS-Port-Id = "00000001"
NAS-IP-Address = 0.0.0.0
NAS-Identifier = "nas01"
Framed-IP-Address = 172.17.91.131
Acct-Session-Id = "4af257f00000001"
Acct-Input-Octets = 170901
Acct-Output-Octets = 196855
Acct-Input-Gigawords = 0
Acct-Output-Gigawords = 0
Acct-Input-Packets = 2165
Acct-Output-Packets = 2089
Acct-Session-Time = 27
Acct-Terminate-Cause = User-Request
++ entering group preacct {...}
++[preprocess] returns ok
[acct_unique] Hashing 'NAS-Port = 1,Client-IP-Address = 172.18.3.218,NAS-IP-
Address = 0.0.0.0,Acct-Session-Id = "4af257f00000001",User-Name =
"0410630046"'
[acct_unique] Acct-Unique-Session-ID = "b5cf75304b7f3484".
++[acct_unique] returns ok
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
++[files] returns noop
++ entering group accounting {...}
[detail] expand: /var/log/freeradius/radacct/%{Client-IP-Address}/detail-
%Y%m%d -> /var/log/freeradius/radacct/172.18.3.218/detail-20020102
[detail] /var/log/freeradius/radacct/%{Client-IP-Address}/detail-%Y%m%d expands
to /var/log/freeradius/radacct/172.18.3.218/detail-20020102
[detail] expand: %t -> Wed Jan 2 08:25:35 2002
++[detail] returns ok
++[unix] returns ok
[radutmp] expand: /var/log/freeradius/radutmp -> /var/log/freeradius/radutmp
[radutmp] expand: %{User-Name} -> 0410630046
rlm_radutmp: Logout for NAS 0.0.0.0 port 1, but no Login record
++[radutmp] returns ok
[sql] expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
```

```
[sql] expand: %{Acct-Input-Gigawords} -> 0
[sql] expand: %{Acct-Input-Octets} -> 170901
[sql] expand: %{Acct-Output-Gigawords} -> 0
[sql] expand: %{Acct-Output-Octets} -> 196855
[sql] expand: %{Acct-Delay-Time} ->
[sql] expand:          UPDATE radacct SET          acctstoptime      =
'%' ,          acctsessiontime      = '%{Acct-Session-Time}' ,
acctinputoctets      = '%{ %{Acct-Input-Gigawords}:-0}' << 32 |
'%{ %{Acct-Input-Octets}:-0}' ,          acctoutputoctets      = '%{ %{Acct-
Output-Gigawords}:-0}' << 32 |          acctterminatecause      = '%{Acct-Terminate-
Cause}' ,          acctstopdelay      = '%{ %{Acct-Delay-Time}:-0}' ,
connectinfo_stop      = '%{Connect-Info}'          WHERE acctsessionid      =
'%{Acct-Session-Id}'          AND username      = '%{SQL-User-Name}'          AND
nasipaddress      = '%{NAS-IP-Address}' ->
acctstoptime      = '2002-01-02 08:25:35' ,          acctsessiontime      =
'27' ,          acctinputoctets      = '0' << 32 |
'170901' ,          acctoutputoctets      = '0' << 32 |
rlm_sql (sql): Reserving sql socket id: 0
rlm_sql (sql): Released sql socket id: 0
++[sql] returns ok
[attr_filter.accounting_response] expand: %{User-Name} -> 0410630046
attr_filter: Matched entry DEFAULT at line 12
++[attr_filter.accounting_response] returns updated
Sending Accounting-Response of id 5 to 172.18.3.218 port 34525
Finished request 2.
Cleaning up request 2 ID 5 with timestamp +123
Going to the next request
Ready to process requests.
```

Hasil sniffing server RADIUS

```
root@radiusserver:~# tcpdump -i eth1 port 1812 or port 1813 or port 3306
08:25:08.754718 IP 172.18.3.218.59277 > radiusserver.local.radius: RADIUS,
Access Request (1), id: 0x00 length: 201
08:25:08.755718 IP radiusserver.local.49357 > radiusdb.local.mysql: P
3414307517:3414307649(132) ack 2108568370 win 1460
08:25:08.757284 IP radiusdb.local.mysql > radiusserver.local.49357: P
1:381(380) ack 132 win 108
08:25:08.757372 IP radiusserver.local.49357 > radiusdb.local.mysql: . ack 381
win 1728
08:25:08.757667 IP radiusserver.local.49357 > radiusdb.local.mysql: P
132:264(132) ack 381 win 1728
08:25:08.757916 IP radiusdb.local.mysql > radiusserver.local.49357: P
381:721(340) ack 264 win 125
08:25:08.758049 IP radiusserver.local.49357 > radiusdb.local.mysql: P
264:378(114) ack 721 win 1996
08:25:08.758227 IP radiusdb.local.mysql > radiusserver.local.49357: P
721:828(107) ack 378 win 125
08:25:08.758373 IP radiusserver.local.49357 > radiusdb.local.mysql: P
378:523(145) ack 828 win 1996
08:25:08.758667 IP radiusdb.local.mysql > radiusserver.local.49357: P
828:1256(428) ack 523 win 142
08:25:08.758785 IP radiusserver.local.49357 > radiusdb.local.mysql: P
523:668(145) ack 1256 win 2264
08:25:08.759314 IP radiusdb.local.mysql > radiusserver.local.49357: P
1256:1722(466) ack 668 win 159
08:25:08.759747 IP radiusserver.local.49356 > radiusdb.local.mysql: P
3402347064:3402347242(178) ack 2099674108 win 1460
08:25:08.760131 IP radiusdb.local.mysql > radiusserver.local.49356: P 1:64(63)
ack 178 win 108
08:25:08.760167 IP radiusserver.local.49356 > radiusdb.local.mysql: . ack 64
win 1460
08:25:08.760254 IP radiusserver.local.49356 > radiusdb.local.mysql: P
178:523(345) ack 64 win 1460
08:25:08.762073 IP radiusdb.local.mysql > radiusserver.local.49356: P
64:687(623) ack 523 win 125
08:25:08.762403 IP radiusserver.local.49355 > radiusdb.local.mysql: P
3408667109:3408667372(263) ack 2100411519 win 1460
08:25:08.791769 IP radiusdb.local.mysql > radiusserver.local.49355: P 1:14(13)
```

```
ack 263 win 108
08:25:08.791845 IP radiusserver.local.49355 > radiusdb.local.mysql: . ack 14
win 1460
08:25:08.793417 IP radiusserver.local.49357 > radiusdb.local.mysql: . ack 1722
win 2532
08:25:08.793431 IP radiusserver.local.49356 > radiusdb.local.mysql: . ack 687
win 1772
08:25:08.793554 IP radiusserver.local.radius > 172.18.3.218.59277: RADIUS,
Access Accept (2), id: 0x00 length: 32
08:25:08.794098 IP 172.18.3.218.34525 > radiusserver.local.radius-acct: RADIUS,
Accounting Request (4), id: 0x04 length: 135
08:25:08.795369 IP radiusserver.local.49354 > radiusdb.local.mysql: P
3414651115:3414652041(926) ack 2107101503 win 1460
08:25:08.805163 IP radiusdb.local.mysql > radiusserver.local.49354: P 1:14(13)
ack 926 win 121
08:25:08.805211 IP radiusserver.local.49354 > radiusdb.local.mysql: . ack 14
win 1460
08:25:08.805369 IP radiusserver.local.radius-acct > 172.18.3.218.34525: RADIUS,
Accounting Response (5), id: 0x04 length: 20
08:25:35.826685 IP 172.18.3.218.34525 > radiusserver.local.radius-acct: RADIUS,
Accounting Request (4), id: 0x05 length: 183
08:25:35.828110 IP radiusserver.local.49353 > radiusdb.local.mysql: P
3409913989:3409914570(581) ack 2112876269 win 1460
08:25:35.852146 IP radiusdb.local.mysql > radiusserver.local.49353: P 1:53(52)
ack 581 win 110
08:25:35.852206 IP radiusserver.local.49353 > radiusdb.local.mysql: . ack 53
win 1460
08:25:35.852389 IP radiusserver.local.radius-acct > 172.18.3.218.34525: RADIUS,
Accounting Response (5), id: 0x05 length: 20

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
^C33 packets captured
33 packets received by filter
0 packets dropped by kernel
root@radiusserver:~#
```

Hasil sniffing NAS

```
[root@NAS-2 ~]# tcpdump -i eth0 port 1812 or port 1813
16:27:28.618651  IP  172.18.3.218.59277  >  172.18.3.31.radius:  RADIUS,  Access
Request (1), id: 0x00 length: 201
16:27:28.657599  IP  172.18.3.31.radius  >  172.18.3.218.59277:  RADIUS,  Access
Accept (2), id: 0x00 length: 32
16:27:28.658022  IP  172.18.3.218.34525  >  172.18.3.31.radius-acct:  RADIUS,
Accounting Request (4), id: 0x04 length: 135
16:27:28.669407  IP  172.18.3.31.radius-acct  >  172.18.3.218.34525:  RADIUS,
Accounting Response (5), id: 0x04 length: 20
16:27:55.687082  IP  172.18.3.218.34525  >  172.18.3.31.radius-acct:  RADIUS,
Accounting Request (4), id: 0x05 length: 183
16:27:55.712923  IP  172.18.3.31.radius-acct  >  172.18.3.218.34525:  RADIUS,
Accounting Response (5), id: 0x05 length: 20

tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

Hasil sniffing server basis data

```
root@radiusdb:~# tcpdump -i eth1 port 3306 > handokop1.dumpdb
16:29:30.112734 IP radiusserver.local.49357 > radiusdb.local.mysql: P
3414307517:3414307649(132) ack 2108568370 win 1460
16:29:30.114010 IP radiusdb.local.mysql > radiusserver.local.49357: P
1:381(380) ack 132 win 108
16:29:30.114344 IP radiusserver.local.49357 > radiusdb.local.mysql: . ack 381
win 1728
16:29:30.114673 IP radiusserver.local.49357 > radiusdb.local.mysql: P
132:264(132) ack 381 win 1728
16:29:30.114713 IP radiusdb.local.mysql > radiusserver.local.49357: P
381:721(340) ack 264 win 125
16:29:30.115050 IP radiusserver.local.49357 > radiusdb.local.mysql: P
264:378(114) ack 721 win 1996
16:29:30.115088 IP radiusdb.local.mysql > radiusserver.local.49357: P
721:828(107) ack 378 win 125
16:29:30.115381 IP radiusserver.local.49357 > radiusdb.local.mysql: P
378:523(145) ack 828 win 1996
16:29:30.115445 IP radiusdb.local.mysql > radiusserver.local.49357: P
828:1256(428) ack 523 win 142
16:29:30.115794 IP radiusserver.local.49357 > radiusdb.local.mysql: P
523:668(145) ack 1256 win 2264
16:29:30.116082 IP radiusdb.local.mysql > radiusserver.local.49357: P
1256:1722(466) ack 668 win 159
16:29:30.116763 IP radiusserver.local.49356 > radiusdb.local.mysql: P
3402347064:3402347242(178) ack 2099674108 win 1460
16:29:30.117002 IP radiusdb.local.mysql > radiusserver.local.49356: P 1:64(63)
ack 178 win 108
16:29:30.117139 IP radiusserver.local.49356 > radiusdb.local.mysql: . ack 64
win 1460
16:29:30.117310 IP radiusserver.local.49356 > radiusdb.local.mysql: P
178:523(345) ack 64 win 1460
16:29:30.118793 IP radiusdb.local.mysql > radiusserver.local.49356: P
64:687(623) ack 523 win 125
16:29:30.119447 IP radiusserver.local.49355 > radiusdb.local.mysql: P
3408667109:3408667372(263) ack 2100411519 win 1460
16:29:30.148640 IP radiusdb.local.mysql > radiusserver.local.49355: P 1:14(13)
ack 263 win 108
16:29:30.148820 IP radiusserver.local.49355 > radiusdb.local.mysql: . ack 14
```

```
win 1460
16:29:30.150395 IP radiusserver.local.49357 > radiusdb.local.mysql: . ack 1722
win 2532
16:29:30.150404 IP radiusserver.local.49356 > radiusdb.local.mysql: . ack 687
win 1772
16:29:30.152571 IP radiusserver.local.49354 > radiusdb.local.mysql: P
3414651115:3414652041(926) ack 2107101503 win 1460
16:29:30.162035 IP radiusdb.local.mysql > radiusserver.local.49354: P 1:14(13)
ack 926 win 121
16:29:30.162183 IP radiusserver.local.49354 > radiusdb.local.mysql: . ack 14
win 1460
16:29:57.184411 IP radiusserver.local.49353 > radiusdb.local.mysql: P
3409913989:3409914570(581) ack 2112876269 win 1460
16:29:57.208183 IP radiusdb.local.mysql > radiusserver.local.49353: P 1:53(52)
ack 581 win 110
16:29:57.208359 IP radiusserver.local.49353 > radiusdb.local.mysql: . ack 53
win 1460
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
27 packets captured
27 packets received by filter
0 packets dropped by kernel
```

LAMPIRAN 4**DATA PENGUJIAN TROUGHPUT****Hasil pengukuran *troughput* sebelum menggunakan sistem**

```
[root@rantsa ~]# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[  4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3131
[  4]  0.0-10.0 sec   106 MBytes  89.0 Mbits/sec
[  5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3132
[  5]  0.0-10.0 sec   98.7 MBytes  82.7 Mbits/sec
[  4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3133
[  4]  0.0-10.0 sec   102 MBytes  85.3 Mbits/sec
[  5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3134
[  5]  0.0-10.0 sec   99.3 MBytes  83.3 Mbits/sec
[  4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3135
[  4]  0.0-10.0 sec   102 MBytes  85.5 Mbits/sec
[  5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3136
[  5]  0.0-10.0 sec   70.3 MBytes  58.9 Mbits/sec
[  4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3137
[  4]  0.0-10.0 sec   92.4 MBytes  77.5 Mbits/sec
[  5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3138
[  5]  0.0-10.0 sec   93.8 MBytes  78.7 Mbits/sec
[  4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3139
[  4]  0.0-10.0 sec   101 MBytes  84.9 Mbits/sec
[  5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3140
[  5]  0.0-10.0 sec   88.6 MBytes  74.3 Mbits/sec
[root@rantsa ~]#
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3131 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 106 MBytes 88.9 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3132 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 98.7 MBytes 82.7 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3133 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 102 MBytes 85.2 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3134 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 99.3 MBytes 83.2 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
```

```
[1912] local 172.17.91.130 port 3135 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   102 MBytes  85.3 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3136 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   70.3 MBytes  58.9 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3137 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   92.4 MBytes  77.4 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3138 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   93.8 MBytes  78.6 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3139 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   101 MBytes  84.7 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
-----  
[1912] local 172.17.91.130 port 3140 connected with 172.18.3.197 port 5001  
[ ID] Interval Transfer Bandwidth  
[1912] 0.0-10.0 sec 88.6 MBytes 74.2 Mbits/sec
```



Hasil pengukuran throughput setelah menggunakan sistem

```
[root@rantsa ~]# iperf -s
-----
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
-----
[ 4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3081
[ 4] 0.0-10.0 sec   100 MBytes  84.0 Mbits/sec
[ 5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3082
[ 5] 0.0-10.0 sec   107 MBytes  89.4 Mbits/sec
[ 4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3085
[ 4] 0.0-10.0 sec   94.3 MBytes  79.0 Mbits/sec
[ 5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3086
[ 5] 0.0-10.0 sec   101 MBytes  84.8 Mbits/sec
[ 4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3087
[ 4] 0.0-10.0 sec   107 MBytes  89.3 Mbits/sec
[ 5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3088
[ 5] 0.0-10.0 sec   94.5 MBytes  79.3 Mbits/sec
[ 4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3089
[ 4] 0.0-10.2 sec   94.3 MBytes  77.7 Mbits/sec
[ 5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3090
[ 5] 0.0-10.0 sec   87.1 MBytes  73.0 Mbits/sec
[ 4] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3091
[ 4] 0.0-10.1 sec   88.7 MBytes  74.0 Mbits/sec
[ 5] local 172.18.3.197 port 5001 connected with 172.18.3.33 port 3092
[ 5] 0.0-10.2 sec   84.0 MBytes  69.4 Mbits/sec
[root@rantsa ~]#
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3081 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   100 MBytes  83.9 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3082 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 107 MBytes 89.4 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3085 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 94.3 MBytes 79.0 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3086 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 101 MBytes 84.6 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
[1912] local 172.17.91.130 port 3087 connected with 172.18.3.197 port 5001
[ ID] Interval Transfer Bandwidth
[1912] 0.0-10.0 sec 107 MBytes 89.3 Mbits/sec
```

```
C:\Documents and Settings\kokothehokage>iperf -c 172.18.3.197
-----
```

```
Client connecting to 172.18.3.197, TCP port 5001
TCP window size: 8.00 KByte (default)
-----
```

```
[1912] local 172.17.91.130 port 3088 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   94.5 MBytes  79.1 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3089 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.2 sec   94.3 MBytes  77.6 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3090 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.0 sec   87.1 MBytes  73.0 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3091 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.1 sec   88.7 MBytes  73.8 Mbits/sec
```

```
C:\Documents and Settings\kokothekokage>iperf -c 172.18.3.197
```

```
-----  
Client connecting to 172.18.3.197, TCP port 5001  
TCP window size: 8.00 KByte (default)
```

```
[1912] local 172.17.91.130 port 3092 connected with 172.18.3.197 port 5001
[ ID] Interval      Transfer     Bandwidth
[1912]  0.0-10.2 sec   84.0 MBytes  69.3 Mbits/sec
```