ANALISIS PERBANDINGAN
KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM MENGIMPLEMENTASIKAN TEKNOLOGI AAA

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika

Disusun oleh:

Nama : Muh. Yasri Reza Afrizal
No. Mahasiswa : 05 523 378

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

HALAMAN JUDUL

ANALISIS PERBANDINGAN KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM MENGIMPLEMENTASIKAN TEKNOLOGI AAA

LAPORAN TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika

Disusun oleh:

Nama : Muh. Yasri Reza Afrizal
No. Mahasiswa : 05 523 378

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

LEMBAR PENGESAHAN DOSEN PEMBIMBING

ANALISIS PERBANDINGAN
KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM
MENGIMPLEMENTASIKAN TEKNOLOGI AAA

TUGAS AKHIR

Disusun oleh:

Nama : Muh. Yasri Reza Afrizal

No. Mahasiswa : 05 523 378

Yogyakarta, 8 Maret 2011

Pembimbing,

Yudi Prayudi, S.Si, M.Kom

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Muh. Yasri Reza Afrizal

No. Mahasiswa : 05 523 378

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

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 2011

Muh. Yasri Reza Afrizal

LEMBAR PENGESAHAN DOSEN PENGUJI

ANALISIS PERBANDINGAN KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM MENGIMPLEMENTASIKAN TEKNOLOGI AAA

TUGAS AKHIR

Disusun oleh:

Nama : Muh. Yasri Reza Afrizal

No. Mahasiswa : 05 523 378

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

Yogyakarta,

Tim Penguji,

Yudi Prayudi, S.Si., M.Kom. _____

Ketua

Syarif Hidayat, S.Kom., MIT. _____

Anggota I

Ari Sujarwo, S.Kom. _____

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

HALAMAN PERSEMBAHAN

Allah Subhanahu wata'ala

Tuhan Pemilik langit dan bumi serta yang ada diantaranya yang memberikan segala kenikmatan kepada hamba yang lemah yang mengharapkan pahala dan surga, sebagai tempat berserah diri dan pasrah, dan sebagai satu-satunya yang berhak disembah, satu-satunya tempat meminta pertolongan, satu-satunya tujuan doa-doa, tak ada yang dapat dipersekutukan denganNya.

Rasulullah dan NabiNya Muhammad
Shalallaahu 'Alayhi Wasallam

Suri tauladan, patokan dalam beramal sholeh, yang telah berjuang dengan sepenuh hati agar islam sampai kepada kita sehingga kita bisa mengenal dengan sebenar-benarnya kepahaman kepada Allah sang pencipta.

Keluarga

Ayah dan ibu dua orang manusia yang harus kita taati setelah Allah dan Rasulullah, dua orang yang menjadi sebab lahirnya kita didunia, Istriku semoga menjadi wanita sholehah dan ibu dari anak-anak yang sholeh/sholehah, Adik-adikku semoga Allah senantiasa menjaga dan memberi kemudahaan kepada kalian.

Almamater

Jurusan Teknik Informatika, Universitas Islam Indonesia

HALAMAN MOTTO

“Jika meninggal seorang anak Adam, maka terputuslah semua ‘amalannya, kecuali 3 perkara : sedeqah jariyyahnya atau ‘*ilmu yang bermanfaat*’ atau anak sholeh yang senantiasa mendo’akannya”.

(HR. Abu Hurairah *radhiallahu ‘anhu*)

“Wahai putra-putriku, tuntutlah ilmu dan curahkan seluruh tenagamu untuknya. Karena, kalaupun hari ini kalian menjadi kaum yang kerdil, kelak dengan ilmu tersebut Allah menjadikan kalian sebagai pembesar kaum.” Lalu beliau melanjutkan, “Sungguh menyedihkan, adakah di dunia ini yang lebih buruk daripada seorang tua yang bodoh?”

(Urwah bin Zubair)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Assalamu'alaikum Warohmatullahi Wabarokatuh

Alhamdulillah, puji syukur kepada Allah 'Azza wa Jalla atas limpahan hidayah, taufiq serta rahmat-Nya, sehingga laporan Tugas Akhir yang berjudul "**ANALISIS PERBANDINGAN KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM MENGIMPLEMENTASIKAN TEKNOLOGI AAA**" ini dapat terselesaikan dengan baik atas izinNya. Shalawat dan salam semoga senantiasa tercurahkan atas Nabi kita Rasulullah Muhammad Sholallahu 'alayhi wasallam, beserta keluarga, sahabat, tabi'in, tabi'ut tabi'in serta kepada pengikutnya yang senantiasa berdiri di atas Al-Qur'an dan As-Sunnah dan yang setia dalam menjaga amalan agama yang sebenar-benarnya hingga akhir zaman.

Laporan Tugas Akhir ini disusun sebagai salah satu syarat guna memperoleh gelar sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Penulis menyampaikan ucapan terimakasih atas bantuan, bimbingan, dukungan dan do'a dari berbagai pihak yang ikut membantu demi kelancaran pelaksanaan Tugas Akhir ini. Untuk itu dengan segala kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. ALLAH Subhanahu wata'ala, Rabb dilangit dan bumi, tidak ada Tuhan selain ALLAH yang berkah disembah, segala puji bagi ALLAH yang telah memudahkan segala urusan hambanya.
2. Nabi Muhammad Shalallahu 'alayhi wasallam sebagai panutan, teladan kaum muslimin dan patokan dalam beramal sholeh dan dalam keseharian.

3. Mama Sri Muljati dan papa Muhammad Ilyas yang selalu mendoakan dan selalu memikirkan kebahagian anak-anaknya, “Ya Allah jagalah kedua orang tuaku sebagaimana mereka menjagaku ketika aku masih kecil”.
4. Jeanny Adriana Paath satu-satunya wanita yang telah halal bagi penulis terimakasih atas pelayanan dan bantuan serta dukungan dan sebagai istri yang senantiasa menyediakan makanan sebagai sumber tenaga dalam mengerjakan Tugas Akhir ☺. Semoga menjadi istri yang sholehah.
5. Ibunda Trimurti, ibunda Susilawati, ayahanda Riffai, seluruh keluarga besar Dg Matorang atas dukungan moral yang diberikan. Ayah dan ibu mertua dan juga kepada paman, bibi dan Saudara(i) di Makassar.
6. Bapak Yudi Prayudi, S.Si., M.Kom, selaku Ketua Jurusan Teknik Informatika dan selaku dosen pembimbing terimakasih atas pengarahan, bimbingan dalam pelaksanaan tugas akhir dan penulisan laporan.
7. Bapak Arwan Akhmad Kh, S.Kom, M.Cs, selaku dosen pembimbing, terimakasih atas pengarahan, bimbingan dalam pelaksanaan tugas akhir dan penulisan laporan.
8. Seluruh dosen dan staf pengajar Jurusan Teknik Informatika, terimakasih atas ilmu dan pengetahuan yang telah diberikan semoga menjadi amal jariyyah.
9. Aan Kurniawan, Anis Asrory, Bagus Aji Saputra, Baytiomo Mawarto, Dodi Prastyo, Ikhsan ND, Mas Dewa atas sedikit banyak bantuannya dalam pembuatan Tugas Akhir dan buat anak-anak ALIEN’05 terimakasih atas pertemanan dan persaudaraannya serta buat seluruh teman-teman Informatika.
10. Semua pihak yang tidak dapat penulis sebutkan satu persatu dalam membantu sejak pengumpulan data sampai penyusunan Tugas Akhir ini.

Semoga amal ibadah dan kebaikan diterima dan mendapat ganjaran yang setimpal dari Allah ‘Azza wa Jalla.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, oleh sebab itu penulis sangat mengharapkan kritik serta saran yang bersifat membangun untuk perbaikan di masa mendatang.

Akhirnya, semoga Tugas Akhir ini dapat berguna dan memberikan manfaat yang besar bagi penulis serta semua pembaca, Amin.

Wassalamu'alaikum Warohmatullahi Wabarokatuh.

Yogyakarta, 8 Maret 2011

Muh. Yasri Reza Afrizal

ABSTRAK

Bagi administrator suatu jaringan komputer keamanan merupakan salahsatu hal yang perlu diperhatikan agar bisa menimbulkan kepercayaan dalam mengakses jaringan bagi user yang berada pada jaringan tersebut. Salahsatu bagian dari keamanan yaitu seorang administrastor harus mengetahui siapa saja yang bisa mengakses jaringan dan apa saja yang bisa diakses di dalam jaringan. Oleh kerena itu diciptakan suatu teknologi yang terkait yaitu AAA.

Teknologi AAA (*Authentication, Authorization* and *Accounting*) adalah suatu teknologi yang memungkinkan administrator mengatur otentikasi atau pengenalan dan pengidentifikasian pengguna, otorisasi atau kontrol bagi pengguna dalam mengakses jaringan, *Accounting* atau proses mengumpulkan informasi terkait apa yang dilakukan oleh pengguna. Ada dua protokol yang dapat mengimplementasikan teknologi ini yaitu RADIUS dan TACACS+. Kedua protokol memiliki beberapa perbedaan dalam mengimplementasikan teknologi AAA sehingga analisis terhadap kedua protokol ini layak didiskusikan.

Kata kunci : AAA (*Authentication, Authorization* and *Accounting*), RADIUS, TACACS+, Server AAA, Klien AAA

TAKARIR

Authentication	:	(otentikasi) pengidentifikasian user
Authorization	:	(otorisasi) kontrol akses user
Accounting	:	pengumpulan informasi user
Mobile internet device	:	alat untuk koneksi internet secara mobile
Method list	:	daftar metode otentikasi
Interface	:	antarmuka
Network Access Server	:	penyedia akses ke jaringan
Pass response	:	paket yang berarti user diizinkan
Fail response	:	paket yang berarti akses user ditolak
Shared secret (secret key)	:	kunci yang digunakan server dan klien AAA
Sniffer	:	suatu kegiatan yang dilakukan untuk mengintai paket yang lewat
Protocol analyzer	:	tool yang digunakan untuk menganalisa paket-paket yang ada pada suatu jaringan
Chipper	:	algoritma untuk menampilkan enkripsi dan dekripsi
Brute force	:	teknik yang digunakan untuk menebak password
Resource	:	sumber
Access Control List	:	daftar untuk mengontrol akses user
Privilege levels	:	tingkatan akses user
IOS Command	:	perintah-perintah yang terdapat pada cisco
Security server	:	server AAA
Attribute Value (AV)	:	attribute yang terdapat pada protokol AAA
Billing	:	informasi penggunaan *resource* pada jaringan

Access Control Server	:	server AAA
Acknowledgement (ACK)	:	pesan pemberitahuan bahwa paket telah sampai ke tujuan
TCP windowing	:	teknik pada TCP untuk mencegah kemacetan pertukaran paket yang terjadi
TCP reset	:	bagian dari paket TCP yang digunakan untuk mengagalkan koneksi TCP
Debugging	:	proses menghilangkan Bug dari suatu program
Clear text	:	text asli tanpa enkripsi

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

LEMBAR PENGESAHAN DOSEN PEMBIMBING

ANALISIS PERBANDINGAN
KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM
MENGIMPLEMENTASIKAN TEKNOLOGI AAA

TUGAS AKHIR

Disusun oleh:

Nama : Muh. Yasri Reza Afrizal

No. Mahasiswa : 05 523 378

Yogyakarta, 8 Maret 2011

Pembimbing,

Yudi Prayudi, S.Si, M.Kom

LEMBAR PENGESAHAN DOSEN PENGUJI

ANALISIS PERBANDINGAN KINERJA PROTOKOL RADIUS DAN TACACS+ DALAM MENGIMPLEMENTASIKAN TEKNOLOGI AAA

TUGAS AKHIR

Disusun oleh:

Nama : **Muh. Yasri Reza Afrizal**

No. Mahasiswa : **05 523 378**

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

Yogyakarta,

Tim Penguji,

Yudi Prayudi, S.Si., M.Kom.

Ketua

Syarif Hidayat, S.Kom., MIT.

Anggota I

Ari Sujarwo, S.Kom.

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Keamanan menjadi hal yang paling penting dalam suatu sistem, dan merupakan pekerjaan yang sulit dihadapi oleh administrator jaringan, salah satunya adalah bagaimana membatasi akses dalam suatu jaringan. Router atau Firewall biasanya hanya memfilter berdasarkan alamat IP sumber dan alamat IP tujuan, juga port untuk mengontrol akses layanan, itu berarti pembatasan hanya kepada perangkat bukan kepada individu-individu. Sebagai contoh apabila kita mengizinkan trafik dari 10.0.0.1 untuk mengakses sebuah server web, maka secara otomatis siapa saja yang berada di depan perangkat dengan alamat 10.0.0.1 dapat mengakses server web ini.

Metode filter yang lebih aman dan fleksibel yaitu memberikan akses lebih spesifik kepada individu yang berwenang, dengan kata lain hanya orang yang memiliki kombinasi username dan password yang benar yang bisa mengakses layanan, tetapi sangat tidak efisien ketika kita menggunakan database pada setiap device yang akan dibatasi aksesnya, sehingga diciptakanlah Access Control Server (ACS) yang menempatkan pusat database user. Protokol yang dapat digunakan untuk mengerjakan tugas ini adalah RADIUS (Remote Authentication Dial In User Service) dan TACACS+ (Terminal Access Controller Access-Control System plus) [SAN02].

RADIUS dikembangkan oleh Livingstone Enterprise,inc. sebagai suatu protokol pada jaringan yang terpusat pada teknologi AAA (Authetication Authorization Acounting). RADIUS adalah client/server protokol yang bekerja di level aplikasi dengan menggunakan UDP (User Datagram Protocol) sebagai protokol pada transportasi paket data, sedangkan TACACS+ dikembangkan oleh Cisco setelah melakukan evaluasi terhadap RADIUS, beberapa fitur dimasukan kedalam TACACS+. TACACS+ mengunakan TCP (Transmission Control Protocol) yang berbeda dengan RADIUS. Selain itu terdapat beberapa perbedaan

lain antara RADIUS dan TACACS+, sehingga analisa terhadap dua protokol ini layak didiskusikan [CIS08].

1.2 Rumusan Masalah

Dari latar belakang di atas dapat ditarik rumusan masalah sebagai berikut :

1. Bagaimana membangun Access Control Server dengan menggunakan protokol RADIUS dan TACACS+.
2. Bagaimana protokol RADIUS dan TACACS+ mengimplementasikan keamanan pada proses AAA.
3. Apa yang menjadi penyebab sehingga fleksiblitas, efektifitas dan efisiensi pada protokol TACACS+ dan RADIUS berbeda.

1.3 Batasan Masalah

a) Membangun server RADIUS dan TACACS+ pada Local Area Network.
b) Implementasi protokol RADIUS dan TACACS+ pada Local Area Network.
c) Analisa dilakukan terhadap protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk memberikan analisis kinerja protokol RADIUS dan TACACS+ agar pengguna dapat menemukan solusi yang tepat sesuai kebutuhan keamanan dalam sebuah jaringan. Penelitian ini berisi diskusi mengenai perbedaan antara RADIUS dan TACACS+ dalam menerapkan teknologi AAA, agar pilihan penggunaan protokol RADIUS atau TACACS+ berdasarkan informasi yang jelas.

1.5 Manfaat Penelitian

a) Memberikan informasi kepada penguna khususnya administrator jaringan mengenai kelebihan dan kekurangan dari kedua protokol tersebut.

b) Membantu para administrator jaringan dalam memilih sebuah protokol (TACACS+ dan RADIUS) sehingga protokol yang digunakan bisa maksimal dalam memenuhi kebutuhan keamanan dan infrastruktur jaringan.

1.6 Sistematika Penulisan

Adapun sistematika laporan tugas akhir ini terdiri atas 5 bab yang secara garis besar adalah sebagai berikut :

BAB I PENDAHULUAN

Berisikan pendahuluan yang menjelaskan latar belakang dari tugas akhir, gambaran umum permasalahan beserta batasan masalah yang menjadi tolok ukur penulisan dalam melakukan penelitian, tujuan penelitian, manfaat penelitian, hipotesis serta sistematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Berisikan tentang landasan teori, yang merupakan pembahasan tentang teori-teori yang digunakan dan relevan dengan topik tugas akhir, yaitu konsep teknologi AAA serta gambaran dari protokol TACACS+ dan RADIUS.

BAB III METODOLOGI

Merupakan bagian yang membahas dan menguraikan langkah-langkah yang digunakan untuk menyelesaikan masalah pada penelitian ”Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA” antara lain bagaimana membangun AAA server dengan menggunakan protokol RADIUS dan TACACS+ serta apa perbedaan yang ada pada protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA.

BAB IV HASIL DAN PEMBAHASAN

Berisi hasil dan pembahasan dari penelitian ”Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA” meliputi hasil pengujian hipotesis yang terkait dengan masalah efektifitas, fleksibilitas, efisiensi serta keamanan pada protokol tersebut.

BAB V KESIMPULAN DAN SARAN

Memuat rangkuman yang didapat dari penjabaran pada bab-bab sebelumnya baik dari segi konsep maupun dari sisi implementasi dan kinerja sistem dan juga memuat saran-saran yang didapatkan dari kekurangan-kekurangan sistem agar kedepannya dapat memaksimalkan kinerja sistem berdasarkan pengujian yang telah dilakukan sebelumnya.

BAB II
LANDASAN TEORI

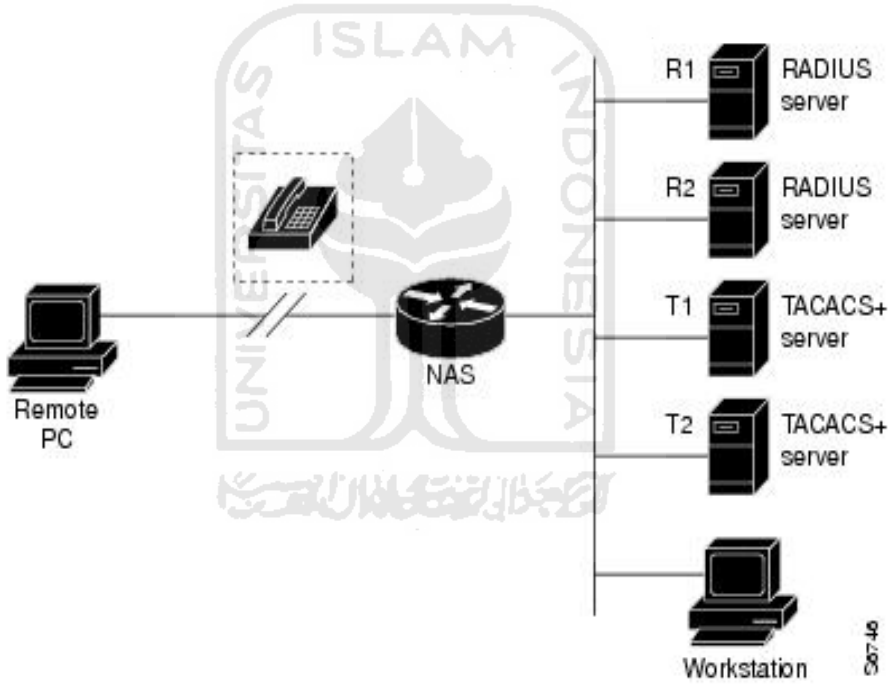
2.1 Teknologi AAA (*Authentication,Authorization Accounting*)

AAA (*Authentication Authorization Accounting*) adalah tiga layanan yang menyediakan keamanan pada sebuah jaringan komputer yang mencatat akivitas user dengan mengidentifikasi siapakah yang sedang mengaksess layanan, layanan apa saja yang bisa diakses dan juga layanan apa yang mereka gunakan ketika melakukan koneksi terhadap server AAA. Contohnya apakah mereka menggunakan akses internet *dial up* atau dengan *mobile internet devices* dll. Bagian otentikasi akan memberitahukan siapa yang menggunakan layanan atau siapa yang mengakses server, sedangkan bagian otorisasi menentukan apa saja yang diizinkan kepada user ketika telah mengakses server, dan *accounting* memantau apa yang dilakukan oleh user ketika berada dalam sistem.

AAA memiliki filosofi bahwa AAA diciptakan untuk memungkinkan seseorang melakukan konfigurasi secara dinamis terhadap tipe otentikasi dan otorisasi yang diinginkan pada tiap-tiap user atau yang berbasis pada layanan, yaitu teknologi AAA bisa melakukan kontrol akses kepada tiap-tiap user atau melakukan kontrol akses berbasis layanan atau service pada suatu jaringan dengan mudah. Otentikasi dan otorisasi dalam penerapannya dibutuhkan pendefinisian awal dari tipe otentikasi dan otorisasi yang dibutuhkan dengan membuat *method lists*, kemudian aplikasikan *method lists* tersebut kepada layanan yang lebih spesifik atau kepada *interface* yang ada.

Method lists adalah daftar yang mendefinisikan metode otentikasi yang digunakan untuk mengotentikasi user. *Method lists* memungkinkan seorang administrator untuk menunjuk satu atau lebih protokol keamanan yang digunakan untuk otentikasi, kemudian memastikan sistem backup untuk otentikasi jika sekiranya metode sebelumnya gagal. Perangkat lunak Cisco IOS (Internetwork Operating System) menggunakan metode yang pertama yang digunakan untuk mengotentikasi para user, jika metode tersebut tidak memberikan tanggapan maka Cisco IOS akan memilih metode otentikasi selanjutnya yang terdapat pada *method*

list. Proses ini akan terus berjalan sampai adanya sebuah komunikasi yang sukses dengan sebuah metode otentikasi yang terdaftar atau tidak adanya komunikasi yang sukses dari semua metode otentikasi, dalam hal ini berarti otentikasi gagal. Perangkat lunak Cisco IOS berusaha melakukan sebuah komunikasi kepada metode otentikasi selanjutnya hanya apabila tidak mendapatkan tanggapan dari metode sebelumnya. Jika otentikasi gagal dalam siklus ini atau dengan arti bahwa *security server* atau database username lokal menanggapi dengan menolak akses dari user, maka proses otentikasi berhenti dan tidak lagi melakukan komunikasi lanjut kepada metode otentikasi yang lain.



Gambar 2.1 Konfigurasi jaringan AAA.

Dari gambar diatas, dimisalkan administrator sistem telah mendefinisikan sebuah *method list* dimana koneksi atau komunikasi awal akan dilakukan kepada R1 untuk informasi otentikasi, kemudian R2, T1, T2 dan terakhir adalah database username lokal yang terdapat pada *Network Access Server* (NAS). Ketika seorang user (Remote PC) melakukan koneksi terhadap sebuah jaringan, *Network Access Server* (NAS) mula-mula menanyakan R1 untuk informasi otentikasi. Jika R1

mengotentikasi user, berarti ada sebuah *PASS response* kepada NAS dan user diizinkan untuk mengakses jaringan. Jika R1 memberikan sebuah *FAIL response*, berarti user tidak diizinkan atau ditolak untuk mengakses jaringan dan sesi tersebut diakhiri, atau proses berhenti. Tetapi apabila R1 tidak melakukan tanggapan sama sekali, maka NAS memprosesnya sebagai sebuah ERROR dan kemudian menanyakan ke R2 untuk informasi otentikasi. Pola ini terus berlanjut kepada sisa metode yang ditentukan sebelumnya terus menerus sampai user terotentikasi atau tertolak. Jika semua metode otentikasi memberikan tanggapan ERROR maka NAS akan memproses sesi tersebut sebagai sebuah kegagalan komunikasi, dan sesi tersebut akan diakhiri.

Perlu diketahui bahwa sebuah *FAIL response* punya perbedaan yang berarti dengan sebuah ERROR. Sebuah FAIL berarti bahwa seorang user yang melakukan komunikasi dengan server atau yang mengakses server tidak memiliki kriteria yang terdapat pada database yang digunakan untuk diotentikasi dengan sukses. Proses otentikasi berakhir dengan *FAIL response*. Sebuah ERROR berarti server belum memberikan tanggapan kepada sebuah permintaan otentikasi. Hanya ketika adanya sebuah ERROR maka AAA server akan memilih metode otentikasi selanjutnya yang sebelumnya telah didefinisikan pada otentikasi *method list* [CIS10].

2.2 *Authentication Authorization dan Accounting*

AAA memiliki tiga fungsi dalam keamanan dan masing-masing memiliki peranan yang penting, yaitu :

2.2.1 Otentikasi

Otentikasi menyediakan metode untuk melakukan identifikasi bagi para user, termasuk user dan password dialog, kemudian enkripsi (tergantung kepada protokol keamanan yang digunakan). Otentikasi dapat berlangsung dalam suatu proses individu atau dapat dikombinasikan dengan otorisasi dan *accounting* [CAR04].

Otentikasi adalah cara mengidentifikasi seorang user terlebih dahulu sebelum diizinkan untuk mengakses jaringan dan layanan yang terdapat pada suatu jaringan. AAA dikonfigurasi dengan mendefinisikan nama dari daftar metode otentikasi, kemudian daftar tersebut diaplikasikan ke beberapa interface. *Method list* mendefinisikan tipe-tipe otentikasi yang dilakukan dan urutan dibagian (interface) yang mana mereka dijalankan. *Method list* ini harus diaplikasikan kepada interface tertentu sebelum metode otentikasi yang lain akan dijalankan. Salah satu pengecualian adalah *method list* default (yang diberi nama ”default”). Metode ini secara otomatis akan diaplikasikan kepada seluruh interface yang terkoneksi apabila tidak ada *method list* lain yang didefinisikan [CIS10].

Dalam melakukan konfigurasi untuk otentikasi pada perangkat Cisco, diperlukan memenuhi beberapa langkah yaitu:

Langkah 1. Mengaktifkan proses AAA.
Meskipun AAA adalah sebuah protokol yang umum yang bisa ditemukan pada kebanyakan perusahaan jaringan, protokol tersebut secara default tidak diaktifkan.

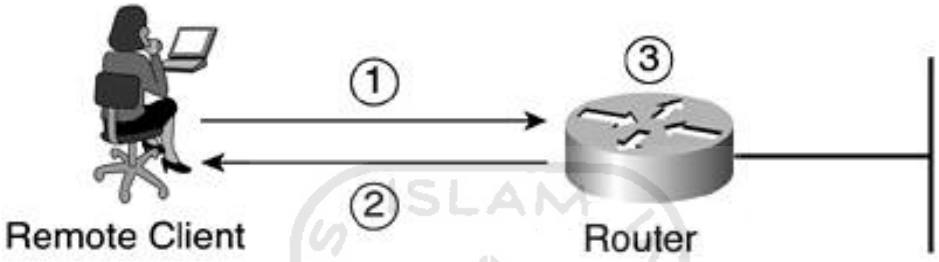
Langkah 2. Mendefinisikan lokasi, protokol, dan *secret key* untuk komunikasi server.

Langkah 3. Definisikan *method list* untuk otentikasi [CAR04].

Otentikasi dilakukan untuk memastikan siapa user jaringan sebenarnya. Ini menjadi penting karena tidak ada yang menginginkan seseorang yang tidak diharapkan melakukan akses ke jaringan. Otentikasi mengizinkan administrator untuk mengidentifikasi siapa yang dapat terkoneksi ke peranti jaringan atau internet dengan memasukkan username dan password.

Saat user mengoneksi ke router dari jarak jauh dengan telnet, user harus memberikan password agar dapat mengakses router. Cara ini berfungsi, tetapi tidak aman karena jika router dikoneksikan ke internet, penyerang akan berusaha untuk terkoneksi, dan bisa saja seorang administrator tidak mengetahui hal ini. Yang dilakukan oleh penyerang adalah menebak sebuah kata kunci untuk mengakses router.

Saat seseorang log in ke piranti jaringan dan membuat perubahan, bagaimana orang tersebut dapat diketahui, dan apa saja yang dikerjakan. Maka dengan otentikasi, kapanpun user ingin mengakses, user harus mengisi username dan password (yang telah ditentukan oleh administrator jaringan) [THO05]. Sehingga dapat diketahui siapa yang mengakses jaringan dan perubahan apa yang dilakukan.



Gambar 2.2 Contoh otentikasi sederhana.

Pada gambar di atas diilustrasikan seorang user melakukan akses kepada router Cisco dengan telnet. Router Cisco dikonfigurasi untuk meminta otentikasi dari siapa saja yang mencoba mengakses router via telnet. Seperti password yang dimasukkan oleh user, password dikirim ke router sebagai *clear text*. Kemudian router mengambil username dan password yang dimasukkan dan meletakkan username dan password tersebut kedalam sebuah paket yang dikirim ke server AAA atau dibandingkan dengan username dan password lokal yang dikonfigurasikan.

Proses pada gambar di atas adalah sebagai berikut :

Langkah 1. User (remote client) membuat koneksi kepada router.

Langkah 2. Router menganjurkan user untuk memberikan username dan password.

Langkah 3. Router mengotentikasi username dan password dalam database lokal. User kemudian diotorisasi berdasarkan informasi yang ada pada database lokal.

Tentu saja contoh diatas bukan tipe terbaik dari otentikasi untuk dilakukan karena siapa saja yang mempunyai akses ke jaringan dan ke jalur yang diambil administrator lokal ke router dapat dengan mudah melihat username dan password tersebut dengan menggunakan beberapa tipe aplikasi *sniffer* dan *protocol analyzer*. Faktanya, kebanyakan protokol tidak melakukan enkripsi password, saat yang lain menggunakan *chiper* yang lemah dan dapat menjadi rentan terhadap serangan *brute force* [CAR04].

2.2.2 Otorisasi

Otorisasi menyediakan metode untuk mengontrol akses seorang user, termasuk otorisasi tiap-tiap layanan, berdasarkan daftar catatan dan profil user, user group support, dsb [CIS10].

Dimisalkan dalam sebuah perjalanan dimana ketika sebuah pesawat komersial yang akan terbang sampai ketempat tujuan perjalanan. Dalam pesawat tersebut terdapat beberapa bagian tempat duduk, bagian di depan terdapat tempat duduk yang nyaman, lebar, empuk dan memiliki tambahan fasilitas yang banyak, dan semua penumpang lebih memilih duduk di tempat tersebut daripada tempat duduk yang berada di bagian belakang karena tempatnya kurang nyaman, tidak ada ruang yang cukup untuk kaki dsb. Sayangnya jika seorang penumpang membayar untuk kelas ekonomi maka tidak akan bisa duduk di tempat duduk kelas eksekutif yang terdapat di bagian depan. Mirip seperti proses ini yaitu fungsi otorisasi pada AAA. Jika penumpang tersebut hanya memiliki otorisasi untuk tiket kelas ekonomi, maka tidak akan bisa mengakses sumber dari kelas eksekutif. Seluruh informasi ini disimpan didalam sistem komputer perusahan penerbangan dan dapat dengan mudah diverifikasi dengan melihat nama para penumpang di sistem komputer dan mereferensikan tempat duduk yang telah ditetapkan [CAR04].

Yang terkait dengan otentikasi adalah otorisasi, sesudah pengguna diotentikasi, ada sebuah cara untuk memastikan bahwa user diotorisasikan untuk mengerjakan sesuatu yang dibutuhkan. Misalnya, seorang user umum tidak

memiliki izin untuk mengakses semua file yang ada didalam sistem file. Gunakan ACL (*Access Control List*) atau policy yang menyediakan otorisasi.

Otorisasi mengizinkan administrator untuk mengontrol tingkatan akses user sesudah mereka mendapatkan akses ke router. Perangkat lunak Cisco IOS mengizinkan tingkatan akses tertentu (*privilege levels*) untuk mengontrol *IOS command* mana yang dapat digunakan oleh user. Misalnya, user dengan *privilege* 0 tidak dapat menjalankan semua *IOS command* sedangkan yang memiliki *privilege* 15 dapat mengerjakan semua *IOS command*. Otorisasi juga dapat melakukan filter terhadap layanan atau protokol yang diizinkan dalam jaringan, seperti hanya mengizinkan FTP, telnet atau HTTP [THO05].

Otorisasi bekerja dengan mengumpulkan sekelompok atribut yang mendeskripsikan apa yang diototorisasi bagi user untuk dilakukan. Atribut ini dibandingkan dengan informasi yang terdapat pada database dan hasilnya dikembalikan ke AAA untuk ditetapkan kapabilitas dan batas-batas bagi user. Database dapat diletakkan secara local pada NAS atau pada router atau dapat juga ditempatkan secara remote pada *security server* RADIUS atau TACACS+. Server keamanan seperti RADIUS dan TACACS+ meng-otentikasi user agar mendapatkan hak-hak tertentu dengan menghubungkan pasangan-pasangan *attribute value* (AV), yang mana mendefinisikan hak-hak tersebut pada user yang tepat. Semua metode otorisasi harus didefinisikan lewat AAA [CIS10].

Otorisasi adalah sebuah metode dari menyediakan beberapa hak istimewa kepada user mengenai layanan yang direquest oleh user. Beberapa protokol yang support terhadap otorisasi adalah IP, Internet Packet Exchange (IPX), AppleTalk Remote Access (ARA) dan telnet. Otorisasi dapat dikonfigurasi berdasarkan grup user, seorang user adalah anggota dari grup tersebut atau bisa juga berdasarkan individu user tersebut. Otorisasi dapat dikonfigurasi secara local pada beberapa kasus atau tetap dikonfigurasi pada remote server.

Ada beberapa langkah yang harus dipenuhi dalam melakukan konfigurasi otorisasi.

Langkah 1 AAA mengumpulkan atribut-atribut berdasarkan pada layanan-layanan yang diminta user untuk di lakukan pada proses otorisasi.

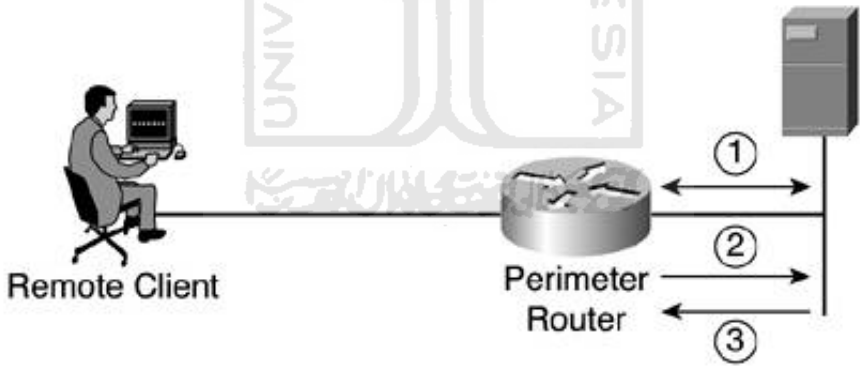
Langkah 2 Atribut-atribut ini dibandingkan dengan sebuah database yang berisi informasi tentang izin yang diberikan kepada user dalam melakukan akses.

Langkah 3 Setelah user diverifikasi, walaupun hasilnya user tersebut diotorisasi atau tidak, hasil dari pengujian tersebut dikembalikan ke proses AAA.

Langkah 4 Setelah rangkaian langkah-langkah sebelumnya, proses AAA kemudian dapat menentukan batas yang tepat kepada user.

Langkah 5 Jika proses otorisasi user ditempatkan pada remote server maka, maka proses otorisasi user dibandingkan kepada *attribute value* (AV).

Sebuah *Method list* mengonfigurasi otentikasi, sebuah *method list* juga dikonfigurasi untuk menetapkan metode otorisasi, perlu untuk mengotentikasi user sebelum menetapkan apa yang terotorisasi bagi user tersebut untuk dilakukan, oleh karena itu otorisasi membutuhkan otentikasi.



Gambar 2.3 Contoh otorisasi dasar FTP

Berikut adalah langkah-langkah pada gambar 2.3 :

Langkah 1 Untuk melakukan otorisasi, diperlukan sebuah sesi yang ditetapkan dengan server AAA.

Langkah 2 Router meminta otorisasi terhadap layanan yang diminta oleh user kepada server AAA.

Langkah 3 Server AAA memberikan PASS/FAIL untuk otorisasi.

Gambar 2.3 menunjukan proses otorisasi dasar yang dapat dibuat, sebagai tambahan dari proses otentikasi yang ditampilkan pada contoh sebelumnya. Satu perbedaan yang perlu diketahui adalah bahwa contoh otentikasi sebelumnya hanya otentikasi local yang didiskusikan. Pada contoh otorisasi ini server AAA ditambahkan [CAR04].

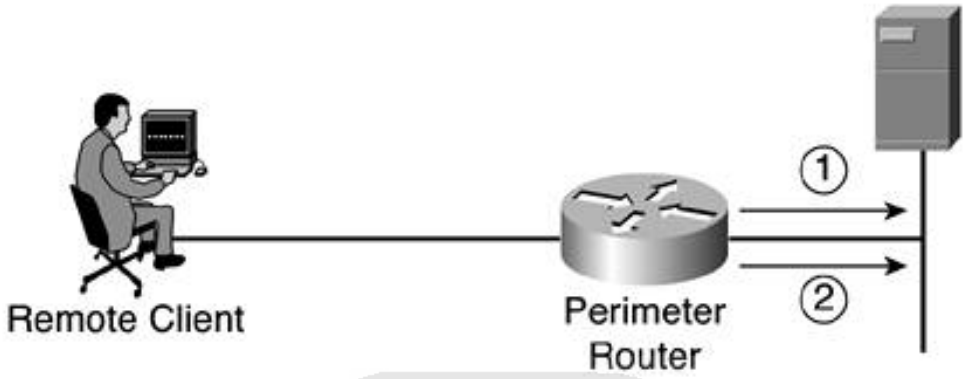
2.2.3 *Accounting*

Accounting menyediakan metode untuk mengumpulkan dan mengirimkan informasi *security server* yang digunakan untuk *billing,* mengaudit dan memberikan laporan, seperti identitas user, waktu mulai dan berhenti mengakses, perintah-perintah yang dieksekusi (seperti Point to Point Protocol), jumlah paket data jumlah byte.

Accounting mengizinkan administrator untuk melacak baik layanan yang sedang diakses oleh user maupun jumlah *resource* jaringan yang digunakan oleh user. Ketika AAA *accounting* diaktifkan, NAS melaporkan aktivitas user kepada server RADIUS atau TACACS+ (tergantung dimana seorang administrator mengimplementasikan metode keamanannya) dalam bentuk catatan-catatan *accounting*. Tiap catatan *accounting* terdiri atas pasangan-pasangan AV *accounting* dan ditempatkan pada ACS (*Access Control Server*). Data ini kemudian dapat dianalisa untuk kepentingan manajemen jaringan, *billing* klien, dan/atau untuk melakukan audit. Semua metode *accounting* harus didefinisikan lewat AAA. Demikian pula dengan otentikasi dan otorisasi. *Accounting* dikonfigurasi dengan mendefiniskan sebuah list yang diberi nama dari metode *accounting*, dan aplikasikan list tersebut ke berbagai *interface* [CIS10].

Accounting terjadi sesudah langkah-langkah otentikasi dan otorisasi terpenuhi. Informasi yang dikumpulkan melalui *accounting* dapat menyediakan bukti forensik dari sebuah perilaku atau hacking karena mereka memiliki petunjuk mengenai waktu/tanggal dan kegiatan user. Khususnya, administrator dapat melacak user mana yang log ke suatu router, perintah IOS mana yang digunnakan oleh user, dan berapa banyak byte yang ditransfer user. Misalnya, dengan

accounting administrator dapat memonitor router yang konfigurasinya telah dirubah. Router atau *security server* dapat mengumpulkan informasi *accounting*.



Gambar 2.4 Contoh *accounting* dasar

Perlu dicatat bahwa proses *accounting* didahului proses otentikasi. Berikut adalah penjelasan dari gambar 2.4 :

Langkah 1 Ketika user telah diotentikasi, proses *accounting* meng-generate sebuah pesan (start message) untuk memulai proses *accounting*.

Langkah 2 Ketika user selesai, sebuah pesan (stop message) merekam dan proses *accounting* berakhir.

AAA adalah sebuah framework untuk otentikasi, otorisasi dan *accounting*. Untuk menjalankan proses-proses tersebut, sebuah device Cisco mengunakan sebuah *method list*, bersama dengan konfigurasi yang lain untuk menetapkan server dan protokol. Pada poin ini seorang administrator seharusnya memiliki sebuah dasar pemikiran mengenai apa itu AAA, apa yang disediakannya terhadap sebuah jaringan, dan sebahagian besar proses konfigurasi dasar [CAR04].

2.2.4 Memulai konfigurasi AAA

Pada awalnya seorang administrator harus memutuskan jenis solusi keamanan seperti apa yang diinginkan untuk diimplementasi. Perlu ada taksiran mengenai resiko keamanan pada jaringan khusus dan memutuskan cara tepat untuk mencegah serangan atau akses masuk yang tidak sah. Cisco

merekomendasikan untuk menggunakan AAA, tanpa melihat seberapa kecil kebutuhan keamanan yang mungkin ada.

Melakukan konfigurasi AAA relatif mudah setelah proses dasar yang bersangkutan dimengerti. Untuk melaukan konfigurasi keamanan pada sebuah router Cisco atau sebuah NAS menggunakan AAA, terdapat beberapa proses yang perlu diikuti.

1. Aktifkan AAA dengan menggunakan perintah konfigurasi global **aaa new-model**.
2. Jika menginginkan untuk menggunakan *security server* yang berbeda, maka lakukan konfigurasi parameter protokol keamanan seperti RADIUS, TACACS+, atau Kerberos.
3. Definisikan *method lists* untuk otentikasi dengan menggunakan perintah **aaa *authentication***.
4. Aplikasikan *method lists* tersebut ke sebuah *interaface* atau jalur khusus, jika diperlukan.
5. Konfigurasikan otorisasi menggunakan perintah **aaa *authorization***.
6. Konfigurasikan *accounting* menggunakan perintah **aaa *accounting***.

Pada proses diatas, melakukan konfigurasi untuk otorisasi atau *accounting* adalah optional, atau bisa digunakan atau diabaikan tergantung pada keinginan administrator dan kebutuhan jaringan.

Sebelum layanan-layanan pada AAA dapat digunakan, AAA harus diaktifkan terlebih dahulu, jika mengaktifkan AAA menggunakan **aaa new-model,** maka untuk me-nonaktifkan AAA menggunakan perintah **no aaa new-model** [CIS10].

2.3 TACACS+ (Terminal Access Controller Access Control System plus)

TACACS+ adalah protokol terbaru yang menyediakan detail informasi *accounting* dan kontrol administratif yang fleksibel pada proses-proses otentikasi dan otorisasi. TACACS+ difasilitasi melalui AAA dan dapat dibolehkan hanya melalui perintah-perintah AAA. TACACS+ adalah hasil dari evolusi TACACS dan extended TACACS (XTACACS). Cisco IOS menyokong ketiga protokol ini.

TACACS adalah protokol paling tua dan tidak cocok (tidak kompatibel) dengan protokol terbaru (TACACS+). Protokol ini menyediakan cek password dan otentikasi, dan pemberitahuan mengenai tindakan user untuk keamanan dan tujuan *accounting*. TACACS menggunakan User Datagram Protokol (UDP) sebagai protokol komunikasinya.

XTACACS adalah sebuah perpanjangan dari protokol TACACS lama, memenuhi tambahan kemampuan kepada TACACS. XTACACS menyediakan informasi mengenai penterjemah protokol dan kegunaan router. Informasi ini digunakan dalam jejak auditing dan file-file *accounting*. XTACACS tidak cocok (tidak kompatibel) dengan TACACS+ dan XTACACS juga menggunakan UDP.

Pada sebuah situasi dimana TACACS+ digunakan, sebuah server menjalankan daemon dan menggunakan ini untuk berkomunikasi dan membuat paket yang ditujukan untuk klien AAA. TACACS+ menggunakan protokol TCP untuk menyediakan pengiriman yang terpercaya. Sebuah *shared secret key* juga digunakan diantara klien AAA dan server AAA yang menjalankan protokol TACACS+. Tiap bagian AAA dijalankan secara terpisah pada TACACS+. Tiap bagian dari layanan, otentikasi, otorisasi, atau *accounting*, dapat diikat kepada databasenya sendiri pada AAA server untuk mendapatkan keuntungan dari layanan lain yang tersedia pada server tersebut atau pada jaringan, tergantung kepada kapabilitas dari daemon.

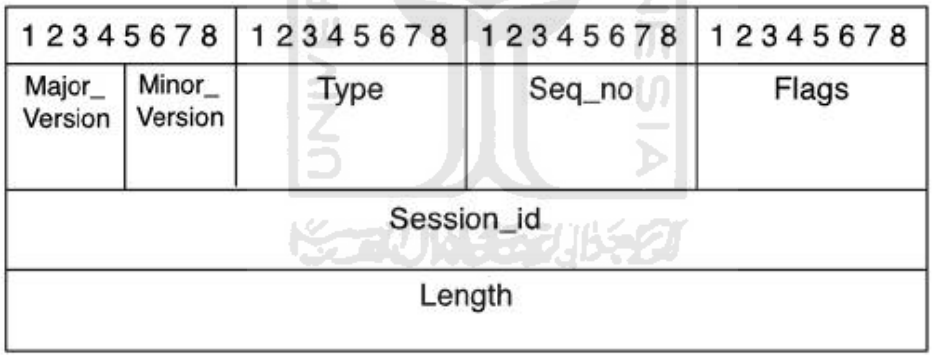
A. Mekanisme komunikasi TACACS+

Komunikasi TACACS+ antara *Network Access Server* (NAS) dan klien AAA berdasarkan protokol TCP dan menyediakan mekanisme pengiriman yang terpercaya pada proses pengiriman pesan teknologi AAA. TACACS+ menggunakan TCP port 49 dan membuat suatu sesi untuk memfasilitasi proses pengiriman pesan tersebut. Terdapat beberapa keuntungan dalam menggunakan TCP pada TACACS+. Diantara keuntungan-keuntungan ini faktanya adalah bahwa TACACS+ menggunakan TCP untuk menyediakan sebuah *acknowledgement* dari request yang dibuat oleh NAS atau klien AAA.

Sebagai tambahan terhadap *acknowledgement* yang ditambahkan dalam TCP, TACACS+ juga mempunyai kemampuan, melalui kemampuan yang melekat dari protokol TCP, menyesuaikan terhadap kemacetan dan bandwidth. Sebuah contoh dari kemampuan atau fungsionalitas TCP adalah pemanfaatan dari *TCP windowing.* TACACS+ juga memiliki kemampuan untuk dengan seketika menentukan ketika sebuah server AAA tidak tersedia dengan menggunakan *TCP reset* untuk memperingatankan klien AAA mengenai permasalahan komunikasi. Kemampuan ini tidak dapat disediakan pada protokol RADIUS karena protokol RADIUS menggunakan UDP pada bagian pegiriman data.

B. Format dan nilai header TACACS+

ID TACACS+ mendefinisikan sebuah header dengan besar 12 byte yang ada pada seluruh paket TACACS+. Header ini selalu dikirimkan dalam format *clear text*.



Gambar 2.5 Format header pada TACACS+

- **Major_version** ini adalah versi utama dari protokol TACACS+. Nilainya tampak pada header sebagai TAC_PLUS_MAJOR_VER=0xc.
- **Minor_version** memiliki nilai yang tampak pada header TACACS+ sebagai TAC_PLUS_MINOR_VER_DEFAULT=0x0 dan TAC_PLUS_MINOR_VER_ONE=0x1.
- **Type** ini membedakan jenis paket. Hanya beberapa tipe yang sah yaitu:

- TAC_PLUS_AUTHEN=0x01, jenis paket ini menandakan otentikasi.
- TAC_PLUS_AUTHOR=0x01, jenis paket ini menandakan otorisasi.
- TAC_PLUS_ACCT=0x03, jenis paket ini menandakan *accounting*.

Arti dari jenis-jenis paket tersebut adalah bahwa TACACS+ memiliki kemampuan untuk menjalankan otentikasi, otorisasi, dan *accounting* sebagai fungsi-fungsi yang terpisah, yang mana RADIUS tidak memiliki kapabilitas seperti ini.

- **Seq_no** ini menetapkan sequence number (nomor urutan) dari sesi yang sedang berjalan. TACACS+ mempunyai kemampuan untuk menjalankan beberapa sesi atau satu sesi pada setiap klien AAA. Paket permulaan dari sebuah sesi diidentifikasi dengan sequence number 1. Paket-paket berikutnya adalah tambahan dari nomor sebelumnya. Karena klien AAA mengirim paket pertamanya ke server AAA yang sedang menjalankan TACACS+ selalu bernomor 1, dan paket-paket setelahnya dari klien AAA diidentifikasi dengan sequence number atau nomor urutan ganjil. Sebagai tambahan dari pola pengurutan ini, bahwa sequence number tertinggi yang dapat dicapai adalah 2^8-1. Setelah nilai ini tercapai, sesi yang dibuat antara klien AAA dan server AAA diputar ulang, dan sesi baru dimulai. Ketika sesi dimulai kembali, sekali lagi sesi tersebut dimulai dengan sequence number 1.
- **Flags** pada bagian flag akan terdapat beberapa flag. Flag jenis TAC_PLUS_UNENCRYPTED_FLAG menetapkan kalau enkripsi sedang dijalankan pada paket TACACS+ tersebut. Jika flag ini diatur, berarti bahwa nilainya 1, sebaliknya enkripsi tidak sedang dijalankan apabila nilai dari flag ini adalah 0. Kemampuan untuk men-disable enkripsi pada TACACS+ semata-mata digunakan untuk tujuan *debugging.* Kemampuan ini baiknya ketika perlu untuk melihat seluruh informasi yang terdapat pada paket data. Perlu diingat bahwa header selalu dikirim sebagai *clear text.* TAC_PLUS_SINGLE_CONNECT _FLAG menetapkan apakah *multiplexing* pada beberapa sesi TACACS+ per satu sesi TCP di-support atau tidak.

Penetapan ini ditetapkan pada dua paket pesan dari satu sesi. Ketika telah ditetapkan maka tidak berubah.

- **Session_id** ini adalah sebuah nilai acak yang menunjuk kepada sesi yang sedang berjalan antara klien AAA dan server AAA yang menjalankan daemon TACACS+.
- **Length** ini menerangkan atau menjelaskan tentang panjang secara keseluruhan dari paket TACACS+.

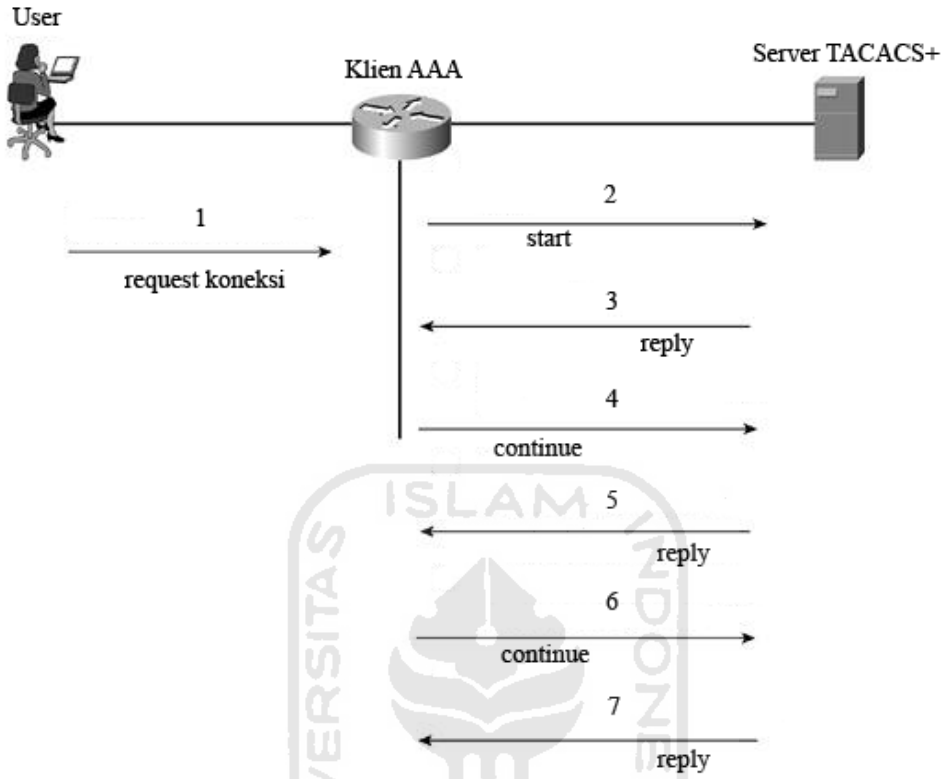
2.3.1 Otentikasi,Otorisasi dan *Accounting* pada TACACS+

Ada tiga kemungkinan aktivitas yang dapat dilakukan selama manjalankan TACACS+. Operasi pertama yang dilakukan adalah otentikasi. Operasi ini selesai dengan teridentifikasinya user secara jelas. Operasi kedua adalah otorisasi dan hanya mungkin setelah user diidentifikasi. Oleh karena itu harus terotentikasi lebih dahulu sebelum diotorisasi. Operasi ketiga adalah *accounting*. Proses *accounting* menyimpan bekas aksi atau catatan-catatan dari segala sesuatu yang dilakukan user. Ketiga proses ini berdiri sendiri-sendiri.

A. TACACS+ dan Otentikasi

Ketika otentikasi dilakukan pada TACACS+, pertukaran tiga paket berbeda berlangsung. Ketiga paket tersebut adalah :

- **Start** ini digunakan pada awalnya ketika user melakukan koneksi.
- **Reply** dikirim oleh server AAA selama proses otentikasi.
- **Continue** digunakan oleh klien AAA untuk mengembalikan username dan password kepada server AAA.



Gambar 2.6 Proses otentikasi pada TACACS+

Pada gambar 2.6, seorang user memulai atau meminta sebuah koneksi kepada klien AAA. Berikut adalah proses yang terjadi pada saat otentikasi :

Langkah 1 Klien AAA menerima permintaan koneksi dari user.

Langkah 2 Tipe paket pertama, START, dikirim kepada server AAA yang menjalankan daemon TACACS+. START ini berisi informasi tentang tipe otentikasi.

Langkah 3 Server TACACS+ kemudian mengirim balik paket REPLY ke klien AAA, pada posisi ini, server AAA meminta username.

Langkah 4 AAA klien mengirim sebuah paket CONTINUE kepada server TACACS+ dengan username yang disediakan user.

Langkah 5 Server TACACS+ kemudian mengirim kembali paket REPLY ke klien AAA untuk meminta kepada klien untuk mendapatkan password dari user.

Langkah 6 Klien AAA mengirim paket CONTINUE ke server TACACS+ dengan password yang disediakan oleh user.

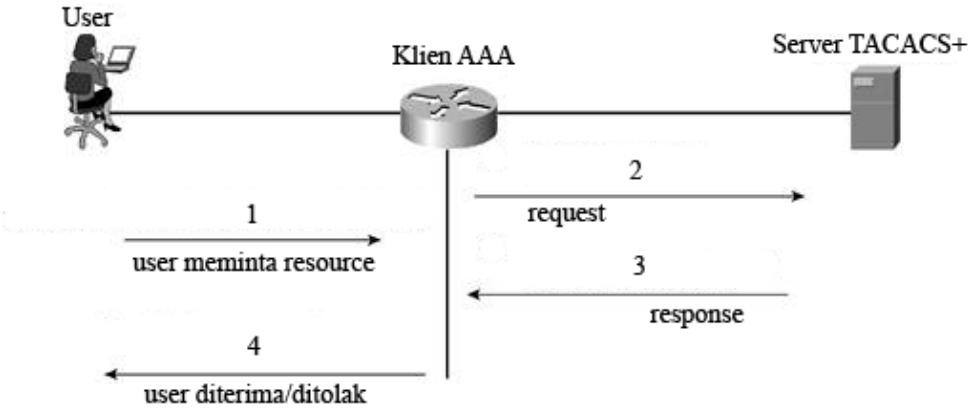
Langkah 7 Server TACACS+ kemudian mengirim kembali paket REPLY ke klien AAA untuk menyatakan apakah user mendapatkan *PASS/FAIL response* dari otentikasi. Kemungkinan nilai yang dikembalikan setelah proses tersebut dapat berupa : ACCEPT, REJECT, ERROR atau CONTINUE.

- ACCEPT dikembalikan ketika user diotentikasi dan layanan bagi user dapat dimulai. Jika NAS dikonfigurasi untuk mengharuskan otorisasi, maka proses otorisasi dimulai.
- REJECT berarti user telah gagal untuk diotentikasi. User dapat ditolak untuk melakukan akses lebih jauh atau disarankan untuk mengulang urutan login tergantung pada daemon TACACS+.
- ERROR terjadi dibeberapa saat selama proses otentikasi. Ini dapat terjadi pada daemon atau pada koneksi jaringan antara daemon dan NAS. Jika sebuah *ERROR response* diterima maka NAS mencoba untuk menggunakan sebuah metode alternative untuk meng-otentikasi user tersebut.
- CONTINUE berarti user disarankan pada informasi otentikasi tambahan.

Paket START dan CONTINUE selalu dikirim oleh klien AAA dan paket REPLY selalu dikirim oleh server TACACS+.

B. TACACS+ dan Otorisasi

Untuk memfasilitasi otorisasi pada TACACS+, ada dua tipe pesan yang digunakan. Pesan pertama adalah REQUEST, yang kedua adalah RESPONSE. Sumber pesan REQUST dari klien AAA dan sumber RESPONSE dari server AAA.



Gambar 2.7 Otorisasi sederhana TACACS+

Berikut adalah langkah-langkah yang terjadi pada saat proses otorisasi :

Langkah 1 Klien AAA menerima permintaan dari user untuk mengakses *resource*. Ini diasumsikan bahwa otentikasi telah berlangsung.

Langkah 2 REQUEST dikirim ke server AAA untuk meminta layanan.

Langkah 3 RESPONSE dikembalikan ke klien AAA yang mengindikasikan *PASS/FAIL response*.

Langkah 4 Klien AAA memberikan dua kemungkinan kepada user, apakah user tersebut ditolak atau diterima untuk mengakses.

RESPONSE berisi salah satu dari balasan dibawah ini :

- *FAIL response* dari server menunjukan bahwa layanan yang diminta untuk otorisasi tidak dibolehkan.
- Jika server menanggapi dengan sebuah PASS_ADD berarti bahwa permintaan tersebut diotorisasi, dan informasi yang dikembalikan dalam RESPONSE digunakan sebagai tambahan bagi informasi yang diminta. Jika tidak ada tambahan argumen yang dikembalikan dari AAA server dalam pesan RESPONSE, maka permintaan diotorisasi.
- Pada beberapa kasus, sebuah PASS_REPL mungkin dikembalikan ke klien AAA. Pada kasus ini, server memilih mengabaikan REQUEST dan menggantinya dengan informasi yang dikembalikan dalam RESPONSE.

- Jika statusnya diatur untuk FOLLOW, ini menunjukan bahwa server AAA yang mengirim RESPONSE menginginkan agar otorisasi terjadi pada server yang lain, dan informasi server ini didaftar dalam paket RESPONSE. Klien AAA memiliki pilihan menggunakan server ini atau dengan mudah dapat memperlakukan proses ini sebagai sebuah FAIL.
- Jika status yang dikembalikan ERROR, ini menunjukan sebuah error pada server AAA. Ini biasanya karena ada ketidak cocokan terhadap *preshared key*, bagaimanapun juga itu dapat menjadi sebuah persoalan dan perbaikan harus dilakukan.

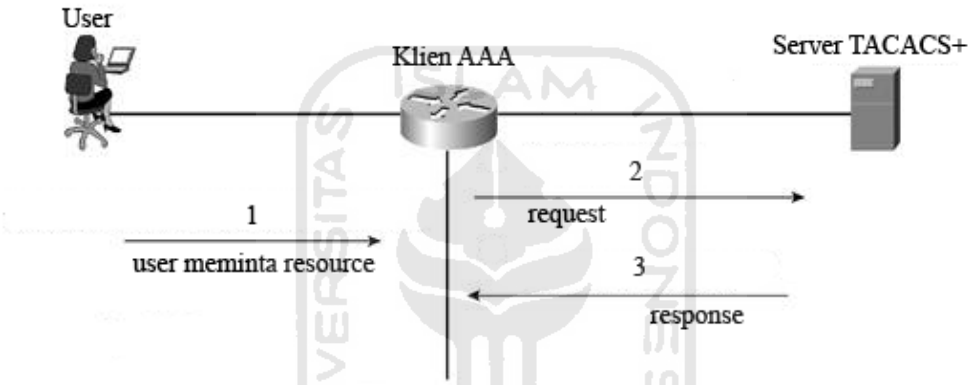
C. TACACS+ dan *Accounting*

Fungsionalitas dari *accounting* dalam TACACS+ mirip dengan otorisasi. *Accounting* berlangsung dengan mengirimkan sebuah record kepada server AAA. Tiap-tiap record terdiri dari satu pasang AV untuk *accounting*. Tiga tipe record dapat dikirim ke server AAA. Tipe-tipe record tersebut adalah :

- **Start record** menunjukan mulainya sebuah layanan dan berisi informasi yang terdapat dalam proses otentikasi, seperti halnya informasi spesifik pada laporannya.
- **Stop record** menunjukan akan berhentinya sebuah layanan dan berisi informasi yang terdapat dalam proses otentikasi, seperti halnya informasi spesifik pada laporannya.
- **Continue record** dikirim ketika layanan masih dalam progress dan mengijinkan klien AAA menyediakan informasi terbaru untuk server AAA. Seperti yang terlihat pada tipe record sebelumnya, record tipe ini juga terdapat informasi yang ada pada proses otorisasi, seperti halnya informasi spesifik pada laporannya.

Accounting juga menggunakan dua tipe pesan yang sama yang digunakan oleh otorisasi, pesan REQUEST dan RESPONSE. Server AAA memiliki kemampuan untuk mengirim tiga jenis balasan pada sebuah pesan RESPONSE yaitu :

- SUCCESS menunjukan bahwa server telah menerima record yang dikirim oleh klien AAA.
- ERROR menunjukan bahwa server gagal memasukkan record tersebut ke database.
- FOLLOW ini serupa dengan FOLLOW dalam otorisasi. Ini menunjukan bahwa server berharap klien AAA untuk mengirim record ke server AAA yang lain, dan informasi mengenai server AAA terdapat dalam paket RESPONSE.



Gambar 2.8 *Accounting* dasar

Gambar 2.8 menunjukan contoh dasar dari prosess *accounting* antara klien AAA dan server AAA. Berikut adalah langkah-langkah yang terjadi pada saat proses *accounting* :

Langkah 1 User meminta *resource* dari klien AAA.

Langkah 2 Klien AAA mengirimkan paket REQUEST kepada server AAA. Paket tersebut dapat berisi record yang ada, yaitu : START, STOP, atau CONTINUE.

Langkah 3 RESPONSE dikembalikan kepada klien AAA, paket ini dapat berisi SUCCESS, ERROR, atau FOLLOW [CAR04].

2.4 RADIUS (Remote Authentication Dial In User Service)

RADIUS juga merupakan protokol yang support terhadap tiga bagian AAA. Protokol otentikasi RADIUS didokumentasikan secara terpisah dari protocol *accounting*, meski demikian kedua-duanya dapat digunakan bersama.

RADIUS pada awalnya dkembangkan oleh Livingstone Enterprises, Inc. RADIUS dicakup dalam RFC 2865, sebagai rival protokol TACACS+ yang diimplementasikan Cisco. RADIUS adalah sebuah protokol yang menggunakan UDP sebagai protokol untuk transmisi data, sebuah klien dan sebuah server. Server mengembalikan sebuah hasil pada informasi yang diminta oleh klien. Informasi yang dikembalikan dari server ke klien dapat ditempatkan pada server RADIUS atau pada perangkat external yang secara langsung berkomunikasi dengan RADIUS. Jika kasusnya seperti ini, klien tidak mempunyai pengetahuan apapun mengenai ini. Tidak seperti TACACS+, RADIUS menjalankan otentikasi dan otorisasi pada waktu yang sama dan *accounting* secara terpisah.

RADIUS adalah sebuah standar Internet Engineering Task Force (IETF) yang digunakan untuk AAA. RADIUS juga merupakan sebuah model client/server. Yang berarti klien AAA mengirim informasi user kepada server AAA, pada kasus ini informasi dikirim via protocol RADIUS, dan server RADIUS menanggapi dengan semua informasi yang dibutuhkan oleh klien AAA untuk menyediakan koneksi dan layanan bagi user.

Untuk otentikasi, sebuah *shared secret key* meng-otentikasi pesan antara server AAA/RADIUS dan klien AAA. *Shared secret key* tersebut tidak pernah benar-benar dikirim melewati kabel sehingga integritas dari kunci tersebut terawat.

Ketika RADIUS meng-otentikasi user, banyak metode otentikasi yang dapat digunakan. RADIUS support otentikasi via PPP CHAP (Point-to-Point Protocol Challenge Handshake Authentication Protocol) dan PAP (PPP Password Authentication Protocol), seperti halnya yang lain. RADIUS juga adalah protocol yang mengizinkan vendor kemampuan untuk menambah *attribute value* baru tanpa menciptakan sebuah masalah terhadap *attribute value* yang ada.

Perbedaan besar antara TACACS+ dan RADIUS adalah bahwa RADIUS tidak mengerjakan otentikasi dan otorisasi secara terpisah. RADIUS juga menyediakan *accounting* yang lebih baik.

RADIUS menggunakan protocol UDP dan menggunakan port 1645 dan 1812 untuk otentikasi dan 1646 dan 1813 untuk *accounting*. Port 1812 dan 1813 dapat dilihat pada implementasi RADIUS terbaru.

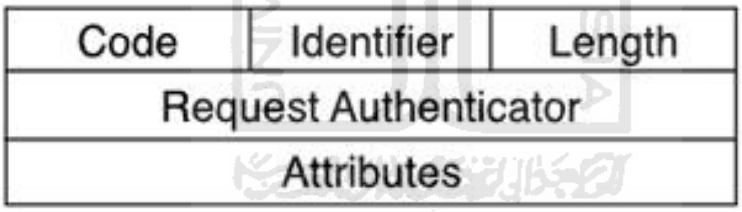
A. Operasi RADIUS

Dibawah ini adalah proses yang digunakan dalam sebuah login RADIUS :

Langkah 1 User melakukan login dan membuat klien AAA mengirimkan sebuah query (Access-Request) ke server RADIUS.

Langkah 2 Response yang sesuai (Access-Accept atau Access-Reject) dikembalikan oleh server ke klien AAA.

Paket Access-Request berisi user name, password yang dienkripsi, IP address dari klien AAA, dan port yang digunakan.



Gambar 2.9 Format paket RADIUS

Tiap paket RADIUS berisi informasi-informasi dibawah ini :

- **Code** sebesar satu octet, dan mengidentifikasikan salah satu tipe dari paket RADIUS berikut ini :
    - Access-Request (1)
    - Access-Accept (2)
    - Access-Reject (3)
    - Accounting-Request (4)
    - Accounting-Response (5)

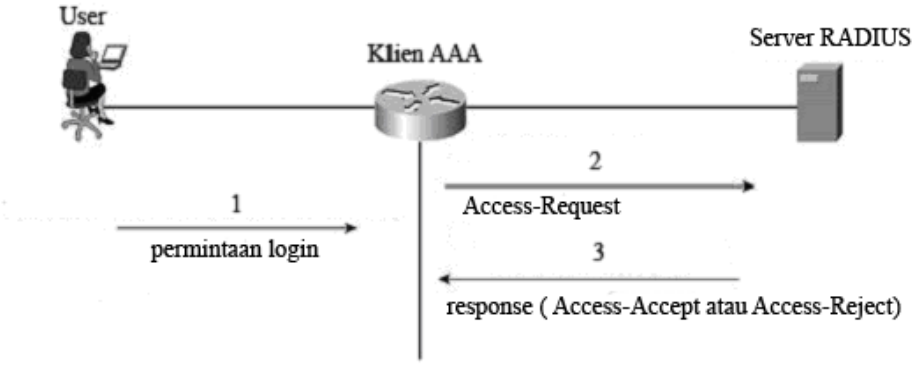
- Access-Challenge (11)
- Status-Server (12)
- Status-Client (13)
- Reserved (255)

- **Identifier** memiliki panjang sebesar satu octet, dan membantu RADIUS server mencocokan request dan response dan mendeteksi duplikat request.
- **Length** sebesar dua octet ini menspesifikasikan panjang keseluruhan paket.
- **Request Authenticator** memiliki luas sebesar 16 octet. Ini meng-otentikasi balasan dari RADIUS server. Ada dua tipe dari authenticator ini yaitu :
  - **Request-Authenticator** tersedia dalam paket Access-Request dan Accounting-Request.
  - **Response-Authenticator** tersedia dalam paket Access-Accept, Access-Reject, Access-Challenge, dan Accounting-Response.

B. Otentikasi dan Otorisasi RADIUS

Ketika sebuah server AAA yang menjalankan RADIUS menerima Access-Request dari klien AAA, server tersebut mencari database untuk username yang terdaftar. Jika username tidak berada dalam database, maka ada dua pilihan, apakah server RADIUS akan me-load sebuah default profile atau seketika mengirim sebuah pesan Access-Reject. Access-Reject ini dapat ditemani oleh pesan teks pilihan, yang mana menunjukan alasan dari penolakan akses ke server RADIUS/AAA.

Jika username ditemukan dan password yang diberikan benar atau cocok, maka server RADIUS mengembalikan sebuah Access-Accept, termasuk daftar dari *attribute- value* yang mengambarkan parameter-parameter untuk digunakan pada sesi ini. Ciri khas parameter tersebut termasuk, tipe protocol, IP address untuk memastikan user tersebut (IP address static atau dynamic), dsb. Informasi konfigurasi pada server RADIUS menetapkan apa yang dipasang pada klien AAA. Sebagai pilihan, server AAA dapat mengirim sebuah Access-Challenge ke klien AAA untuk meminta password baru.



Gambar 2.10 Pertukaran paket pada RADIUS

Proses yang terjadi pada gambar 2.10 adalah sebagai berikut :

Langkah 1 User mengirim sebuah permintaan login ke klien AAA.

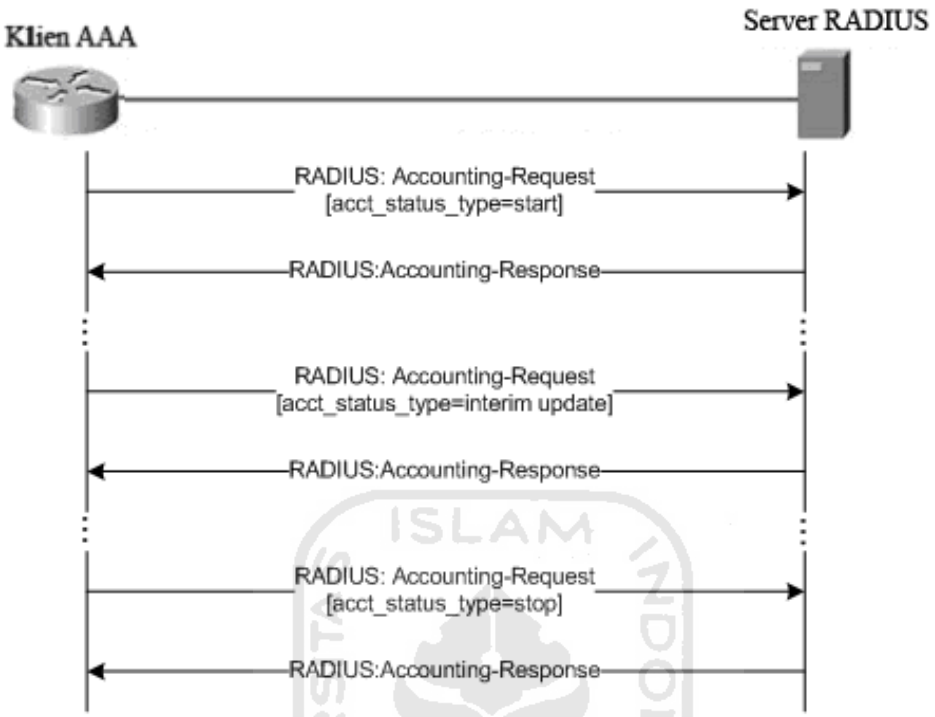
Langkah 2 Klien AAA mengirim Access-Request ke server AAA.

Langkah 3 Server AAA mengembalikan respose yang dapat berupa Access-Accept atau Access-Reject.

Otorisasi dalam RADIUS selesai bersamaan dengan otentikasi. Pesan Access-Accept yang dikembalikan, termasuk juga daftar dari pasangan AV yang ada untuk mengotorisasi user [CAR04].

C. *Accounting* RADIUS

Accounting pada RADIUS dijalankan dengan mengirim pesan pada mulainya atau berhentinya sebuah sesi *accounting*. Pesan-pesan ini termasuk informasi mengenai sesi tersebut. Informasi yang mungkin dimasukkan termasuk informasi waktu, paket-paket, jumlah byte, dll. Pesan-pesan ini dikirim menggunakan UDP pada port 1813. proses *accounting* untuk RADIUS dapat dilihat dalam RFC 2866. pesan yang dikirim antara server AAA dan klien AAA adalah Accounting-Request dan Acconting-Response.



Gambar 2.11 Aliran data pada *accounting* RADIUS

Aliran data pada gambar 2.11 adalah ketika akses ke jaringan diizinkan kepada user oleh klien AAA, sebuah Accounting-Request yang berisi salah satu pasangan *attribute value* (Acct-Status-Type=start) dikirim ke server RADIUS untuk memberi tanda bahwa user telah mulai mengakses jaringan. Kemudian Accounting-Response dikembalikan oleh server sebagai tanggapan dari paket Accounting-Request.

Kemudian pada waktu tertentu dikirim Acct-Status-Type=interim update, yaitu pasangan *attribute value* yang bertujuan untuk memperbaharui informasi user, mengenai jumlah resourse yang telah digunakan, dsb.

Akhirnya ketika akses jaringan user ditutup, klien AAA/RADIUS mengirim Accounting-Request yang berisi jenis *attribute value* Acct-Status-Type dengan nilai “stop” ke server RADIUS, yang menyediakan informasi penggunaan akhir dari waktu, paket dan data yang ditransfer, dan informasi lain yang berhubungan dengan akses user ke jaringan [RIG00].

BAB III

METODOLOGI

3.1 Metode Analisis

Metode analisis yang digunakan untuk mendapatkan data yang diperlukan dalam melakukan perbandingan protokol RADIUS dan TACACS+ ini menggunakan metode komparasi yaitu dengan melakukan perbandingan antara kedua sistem protokol RADIUS dan TACACS+, yang mana akan dilakukan sebuah eksplorasi kepada kedua protokol RADIUS dan TACACS+, eksplorasi yang berkaitan dengan keamanan, fleksibilitas, efektifitas dan efisiensi serta hal-hal lain yang mungkin ditemukan pada protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA.

3.1.1 Metode Analisis kinerja protokol

Analisa dilakukan terhadap client dan server serta komunikasi data dan proses komputasi yang dihasilkan ketika menggunakan protocol RADIUS dan TACACS+, kemudian dilakukan perbandingan dalam hal keamanan, fleksibilitas kedua protocol tersebut dalam memberikan pelayanan dan kenyamanan terhadap user atau administrator jaringan yang menggunakan protokol ini. Analisa juga dilakukan dengan hasil tujuan adalah mendapatkan atau bisa memberikan definisi mengenai kapan, bagaimana masing-masing protokol ini efektif serta effisien digunakan. Metode perbandingan yang akan dilakukan adalah sebagai berikut :

3.2 Analisis Masalah

Analisis sistem adalah prosedur yang dilakukan untuk membuat spesifikasi perbandingan protokol RADIUS dan TACACS+ yang akan dilakukan. Analisis sistem dilakukan dengan tujuan untuk mendapatkan dan mengidentifikasi kekurangan dan kelebihan yang terdapat pada protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA. Setelah diidentifikasi maka dapat

dilakukan perbandingan antara RADIUS dan TACACS+, sehingga dapat dirinci lebih dalam hasil yang telah didapat dari segmen sebelumnya. Parameter perbandingan yang ditemukan dari analisis masalah adalah sebagai berikut :

a. Analisa keamanan

Pengujian keamanan pada protokol RADIUS dan TACACS+ dilakukan untuk mengetahui perbedaan yang terdapat pada kedua protokol terkait dengan isu-isu keamanan. Eksplorasi dilakukan dengan cara mengamati metode komunikasi paket data, kemudian mendefinisikan kelebihan dan kekurangan dari metode tersebut.

b. Analisa Fleksibilitas protokol

Hasil dari analisa ini akan ditemukan dengan melakukan eksplorasi terhadap fitur-fitur yang disediakan oleh kedua protokol, yaitu fitur-fitur yang berkaitan dengan teknologi-teknologi yang disediakan oleh masing-masing protokol.

c. Analisa efektifitas dan efisiensi penggunaan protokol RADIUS dan TACACS+

Hasil dari analisa efektifitas dan efisiensi penggunaan protokol RADIUS dan TACACS+, bisa didapatkan apabila sebelumnya telah melakukan analisis pada kedua masalah sebelumnya, setelah eksplorasi keamanan dan fitur-fitur yang terdapat pada protokol RADIUS dan TACACS+ dilakukan, maka dapat didefinisikan perbedaan kedua protokol tersebut dalam hal efektifitas dan efisiensi kegunaan protokol tersebut.

Penggunaan parameter-paremeter tersebut adalah berdasarkan pada salah satu artikel yang dengan judul “TACACS+ and RADIUS comparison” yang dirilis oleh Cisco. Setelah membaca dan mengamati isi dari artikel tersebut, maka dapat disimpulkan beberapa parameter seperti yang telah disebutkan sebelumnya. Parameter-parameter tersebut dipilih dikarenakan adanya kemungkinan untuk melakukan pembuktian dengan jaringan sederhana dan juga dengan memperhitungkan sumber daya yang dimiliki.

3.3 Rancangan Sistem

Protokol otentikasi atau dalam hal ini RADIUS dan TACACS+ melakukan proses otentikasi pertukaran data melalui server dan client yang mana server melakukan sebuah proses komputasi yang diminta oleh client. Untuk mengetahui kinerja dari tiap-tiap protocol, maka akan dibangun sebuah bentuk jaringan yang sama untuk protokol RADIUS dan TACACS+ sehingga perbandingan bisa dilakukan dengan baik. Selain itu dibutuhkan juga adanya sokongan dari perangkat lunak maupun perangkat keras guna memenuhi kebutuhan analisis pada protocol RADIUS+ dan TACACS+.

3.3.1 Klien pada server AAA

Klien dari server RADIUS atau TACACS+ adalah sebuah NAS yang menyediakan akses ke sebuah jaringan bagi client dari NAS itu sendiri. NAS tersebut harus support terhadap dua jenis protokol transport yaitu TCP dan UDP selain itu, meskipun AAA adalah protokol yang umum digunakan pada perusahaan jaringan, pada perangkat Cisco, protokol tersebut secara default tidak diaktifkan, sehingga diperlukan untuk mengaktifkan proses AAA pada NAS (pada kasus Router Cisco bertindak sebagai NAS itu sendiri).

3.3.2 Klien output

Adapun keluaran yang dihasilkan berupa semua informasi tentang keadaan mesin client (Remote client/client dari NAS). Yang mana informasi tersebut dikirimkan lagi ke server AAA untuk diotentikasi, dan kemudian hasilnya yaitu sebuah *PASS/FAIL response* dikembalikan kepada user.

3.3.3 Server AAA

Setelah NAS menerima permintaan untuk koneksi maka NAS akan meminta syarat yaitu username dan password yang kemudian akan dikirimkan kepada server AAA. Server AAA menerima username dan password dan kemudian mengotentikasinya pada database yang telah ditentukan, user yang diizinkan (*PASS response*) akan diotorisasi berdasarkan informasi pada database.

NAS akan menerima *FAIL response* dan mengembalikannya kepada user ketika username dan password yang diberikan ditolak oleh server AAA.

3.3.4 Fungsi Server

Server pada Protokol keamanan seperti RADIUS dan TACACS+, atau secara keseluruhan disebut server AAA memiliki seluruh informasi mengenai user dan tingkatan akses yang diizinkan bagi tiap-tiap user, informasi tersebut disimpan didalam database yang digunakan pada proses otentikasi dan otorisasi user.

Hasil dari analisis ini adalah mengetahui perbedaan antara RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA.

3.4 Perangkat lunak yang dibutuhkan

Perangkat keras komputer tidak akan berarti tanpa adanya perangkat lunak begitu juga sebaliknya. Jadi kedua perangkat tersebut saling mendukung satu sama lain. Perangkat keras hanya dapat berfungsi jika diberikan perintah-perintah kepadanya, perintah tersebut diberikan oleh perangkat lunak. Perangkat lunak yang dibutuhkan untuk implementasi analisis protokol RADIUS dan TACACS+ ini adalah sebagai berikut :

1. Sistem operasi Windows Server 2003
2. Sistem operasi Windows XP SP 3
3. ClearBox RADIUS Server pada server RADIUS
4. ClearBox TACACS+ Server pada server TACACS+

3.5 Perangkat keras yang dibutuhkan

Penggunaan sistem komputer sebagai alat bantu dalam menyelesaikan tugas-tugas atau pekerjaan banyak digunakan pada saat ini. Hal tersebut merupakan suatu alternatif pemecahan masalah karena banyaknya kemudahan-kemudahan yang dapat diperoleh.

Perangkat keras komputer yang digunakan adalah yang dapat mendukung perangkat lunak sehingga bisa berjalan secara optimal. Perangkat keras yang akan digunakan adalah sebagai berikut :

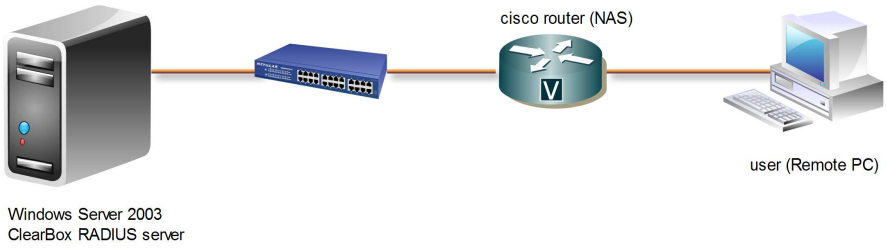
Komputer untuk tiap-tiap server RADIUS dan TACACS+:

1. Intel 1.66 GHz 1 CPU.
2. 512 MB RAM.
3. Harddisk 4 GB.
4. 1 LANcard.

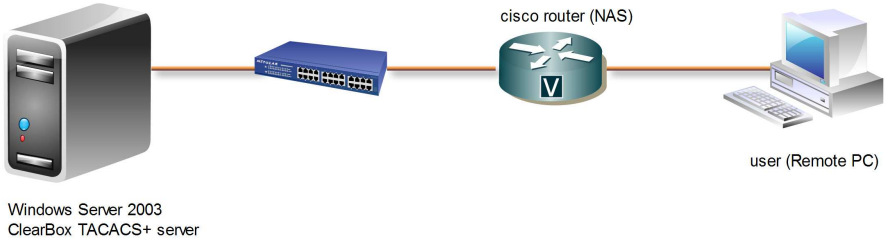
Untuk Remote PC yang diperlukan adalah perangkat jaringan umum yaitu LAN card untuk koneksi berbasis kabel atau Wi-Fi untuk koneksi nirkabel. Kemudian untuk Router dengan IOS yang support terhadap protokol RADIUS dan TACACS+.

3.6 Perancangan Network Diagram

Network diagram adalah sebuah pemetaan jaringan yang dilakukan seorang administrator untuk merepresentasikan kegiatan-kegiatan yang akan dilakukan. Network diagram sangat berguna ketika seorang administrator akan menyusun sebuah jaringan komputer, karena semua aktifitas akan ditampilkan didalam network diagram. Berikut merupakan gambaran umum dari Network diagram yang akan dibangun :



Gambar 3.1 Network Diagram dengan RADIUS

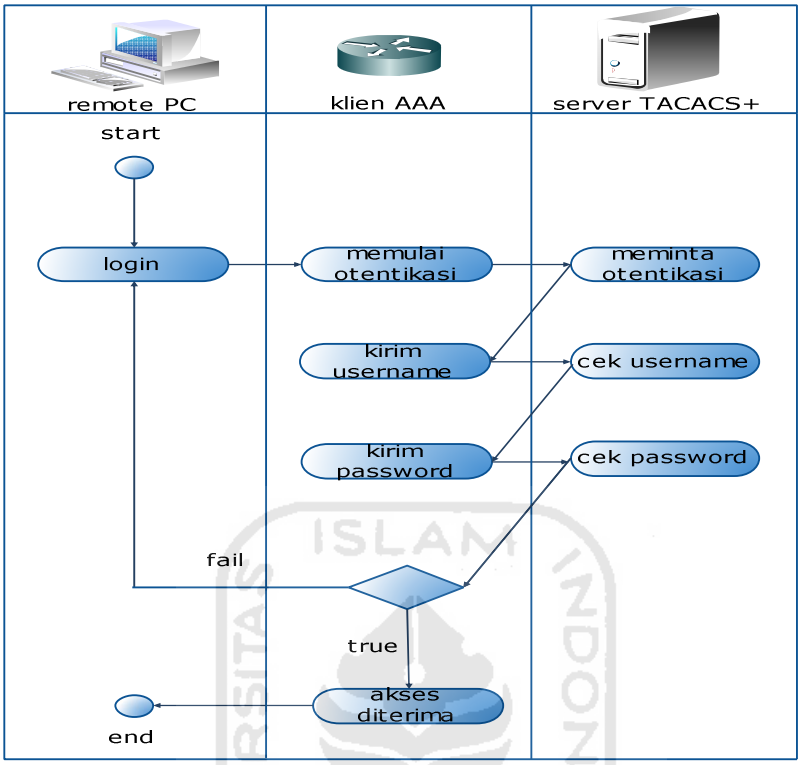


Gambar 3.2 Network Diagram dengan TACACS+

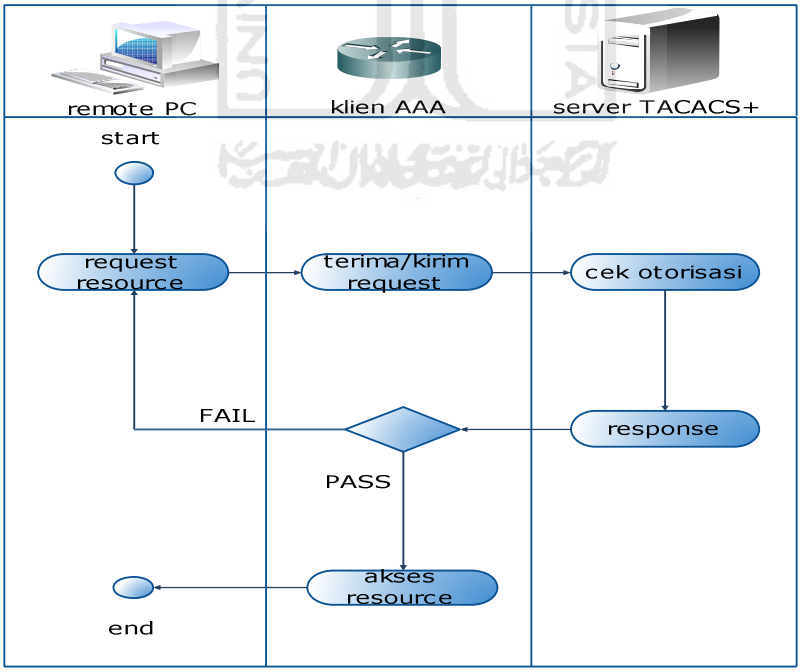
Dari gambar 3.1 dan gambar 3.2, dapat dilihat bahwa analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan taknologi AAA akan melibatkan 2 buah router cisco sebagai 2 buah PC dan juga 2 buah server yaitu : server RADIUS dan server TACACS+ yang berada pada jaringan yang berbeda. Seperti yang telah disebutkan sebelumnya protokol RADIUS dan TACACS+ menggunakan tipe protokol transport yang berbeda, RADIUS menggunakan UDP sedangkan TACACS+ menggunakan TCP. Sedangkan router yang bertindak sebagai NAS sebagai perantara antara user dan server.

3.7 Perancangan Activity Diagram

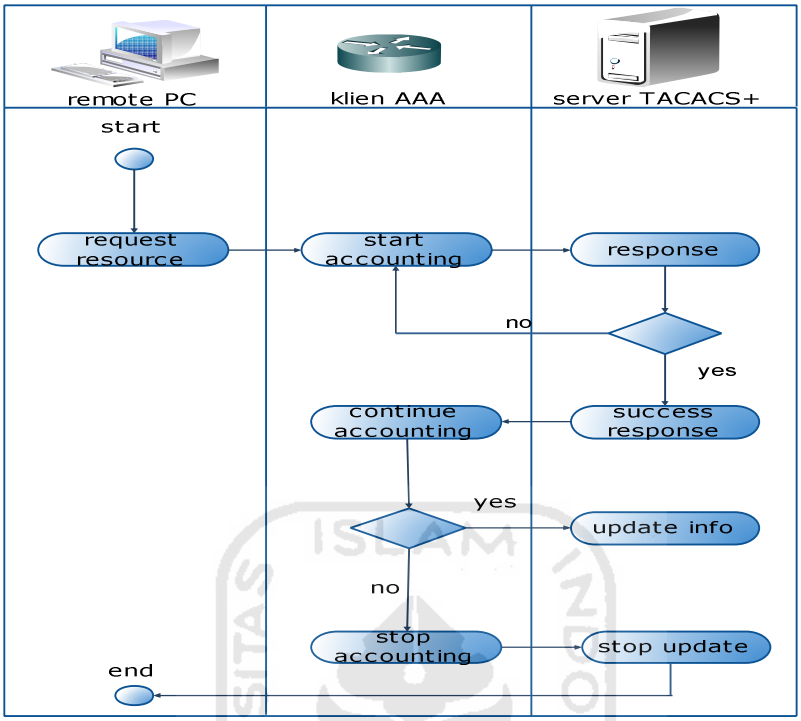
Berikut merupakan gambaran umum dari Activity diagram dari proses pada protokol RADIUS dan TACACS+:



Gambar 3.3 Activity Diagram proses otentikasi pada TACACS+

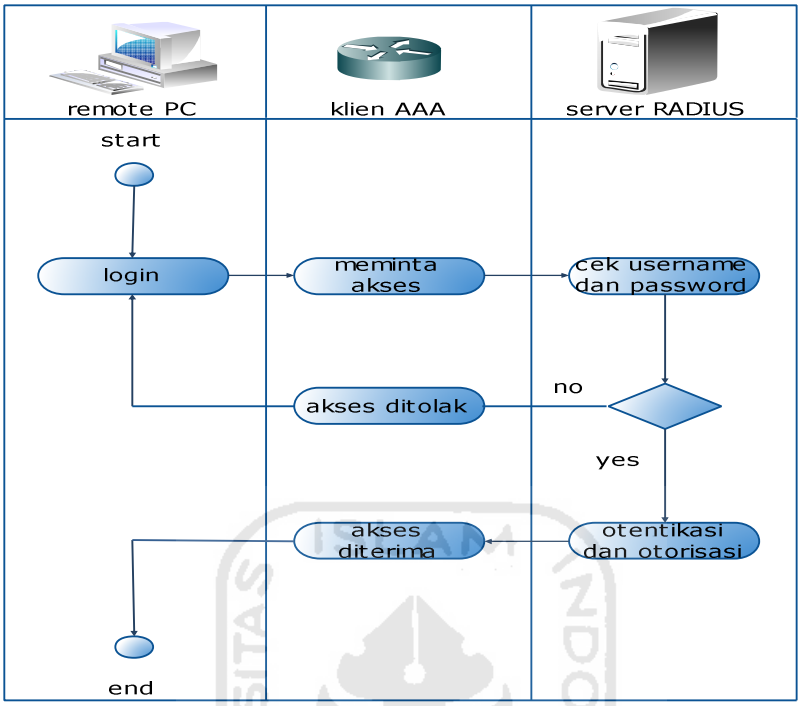


Gambar 3.4 Activity Diagram proses otorisasi pada TACACS+

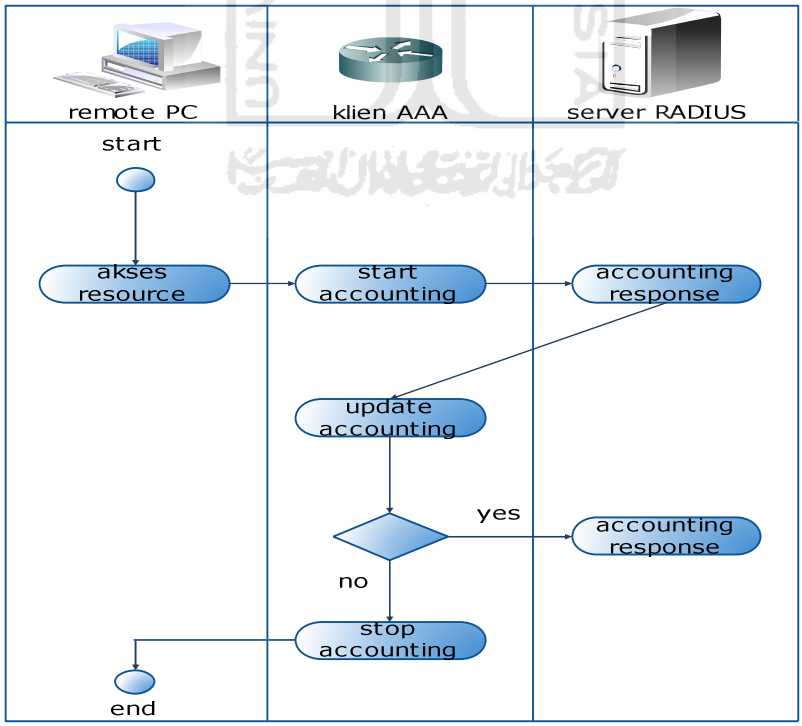


Gambar 3.5 Activity Diagram proses accounting pada TACACS+

Pada Gambar diatas dapat dilihat bahwa proses AAA pada TACACS+ adalah tiga proses yang berdiri sendiri-sendiri. Pada activity diagram proses otentikasi, setelah user melakukan login, maka klien dari TACACS+ akan mengirim username dan password secara terpisah sesuai permintaan dari server TACACS+. Kemudian pada proses selanjutnya, otorisasi berlangsung ketika ada permintaan dari user untuk mengakses resource yang sediakan. Selanjutnya proses *accounting* dimulai dengan pesan start yang mengindikasikan dimulainya layanan, kemudian pesan continue yang berisi info terbaru dari user, yang digunakan untuk mengupdate informasi, kemudian pesan stop yang menunjukan berhentinya sebuah layanan.



Gambar 3.6 Activity Diagram proses otentikasi dan otorisasi pada RADIUS



Gambar 3.7 Activity Diagram Proses *accounting* pada RADIUS

Pada Gambar diatas dapat dilihat bahwa proses otentikasi dan otorisasi bersamaan pada satu proses berjalan, otorisasi pada RADIUS selesai bersamaan dengan otentikasi, pesan yang mengindikasikan bahwa user terotentikasi atau diizinkan untuk mengakses server/layanan/resource, termasuk juga daftar pasangan AV yang ada untuk mengotorisasi user. Kemudian *accounting* pada RADIUS mirip seperti pada proses *accounting* pada TACACS+, yang mana terdiri dari pesan start untuk mulainya sebuah *accounting*, update informasi dan berhentinya sebuah sesi *accounting*.

BAB IV

HASIL DAN PEMBAHASAN

Hasil dan pembahasan Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA (*Authentication*, *Authorization*, *Accounting*) dilakukan agar dapat diketahui bagaimana teknologi AAA dikerjakan pada protokol RADIUS dan TACACS+, dan berisi Implementasi protokol RADIUS dan TACACS+ pada dua topologi yang serupa.

4.1. Batasan Implementasi

Sistem yang dibangun adalah simulasi dengan topologi yang sama pada kedua jaringan, yang mana tiap-tiap infrastruktur memiliki sebuah server AAA, sebuah Router sebagai klien dari server AAA dan juga bertindak sebagai NAS (Network Access Server), dan sebuah PC client yang nantinya akan melakukan simulasi akses kepada NAS dengan menggunakan salah satu aplikasi remote login yaitu Telnet, yang secara default sudah didukung oleh sistem operasi yang digunakan pada simulasi ini. Dan juga paket-paket yang dianalisis adalah pertukaran paket-paket yang terjadi antara server AAA dan klien AAA.

4.2 Tahapan Analisis Perbandingan protokol RADIUS dan TACACS+

Dalam Analisis perbandingan kinerja protokol RADIUS dan TACACS+, terdapat beberapa tahapan yang yaitu :

1. Tahap Pembuatan dan Konfigurasi server RADIUS dan server TACACS+.

Pada tahap ini yang dilakukan adalah membangun server AAA yaitu server yang akan melakukan komputasi fitur-fitur dari protokol RADIUS dan TACACS+. Untuk menjalankan server AAA diperlukan adanya sebuah aplikasi, dan aplikasi yang digunakan adalah ClearBox TACACS+ RADIUS server yang berjalan diatas sistem operasi Windows.

2. Tahap Konfigurasi NAS.

Pada tahap ini dilakukan konfigurasi router yang bertindak sebagai NAS atau klien dari server AAA (RADIUS/TACACS+), yang mana konfigurasi bertujuan untuk mengaktifkan fungsi dari server AAA dalam melakukan Otentikasi, Otorisasi dan *Accounting*.

3. Tahap Eksplorasi.

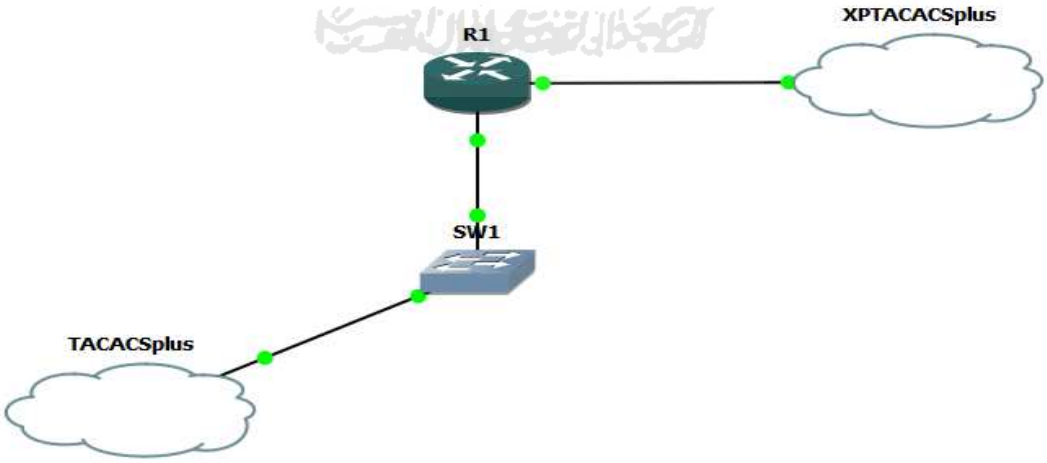
Pada tahap ini dilakukan eksplorasi yang berkaitan dengan fungsi protokol AAA yaitu RADIUS dan TACACS+, sehingga akan ditemukan beberapa perbedaan dari protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA.

4. Tahap Analisa perbandingan.

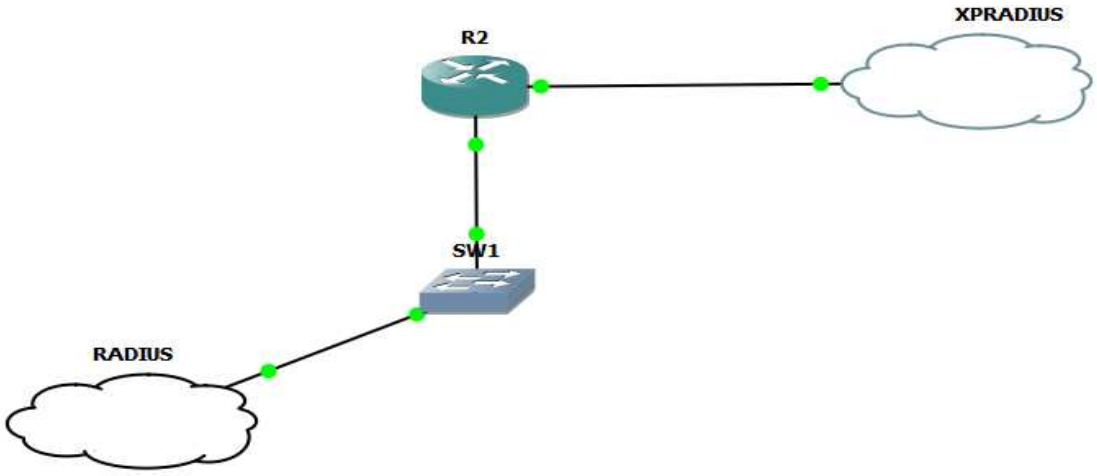
Pada tahap ini dilakukan analisa terhadap kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA, setelah sebelumnya melakukan eksplorasi terkait dengan parameter pembanding yaitu dari sisi keamanan, fleksibilitas serta efektivitas dan efisiensi dari protokol RADIUS dan TACACS+.

4.3 Implementasi perbandingan protokol RADIUS dan TACACS+

Seperti yang telah disampaikan sebelumnya bahwa pada implementasi perbandingan protokol RADIUS dan TACACS+ terdapat dua jaringan masing-masing yaitu jaringan yang menggunakan protokol RADIUS dan jaringan yang menggunakan protokol TACACS+. Berikut adalah gambar topologi untuk simulasi kedua jaringan tersebut.



Gambar 4.1 Topologi Simulasi TACACS+



Gambar 4.2 Topologi simulasi RADIUS

Secara sederhana pada tiap-tiap gambar diatas memiliki dua buah cloud (gambar awan) yang menghubungkan ke mesin virtual yang telah ditentukan. Untuk cloud dengan nama TACACSplus dan RADIUS adalah cloud yang terhubung dengan VMware yang menjalankan server TACACS+ dan RADIUS sedangkan untuk cloud dengan nama XPTACACSplus dan XPRADIUS adalah cloud yang terhubung dengan VirtualBox yang mana cloud tersebut bertindak sebagai user yang akan melakukan percobaan akses ke router. Kedua topologi diatas dibuat pada sebuah aplikasi untuk simulasi router yaitu GNS3.

4.3.1 Langkah-langkah Instalasi Server

Proses instalasi server dilakukan guna memenuhi proses perbandingan protokol RADIUS dan TACACS+. Sistem operasi yang digunakan yaitu Windows server 2003 dengan aplikasi ClearBox TACACS+ RADIUS server yang bertindak sebagai server AAA (RADIUS/TACACS+).

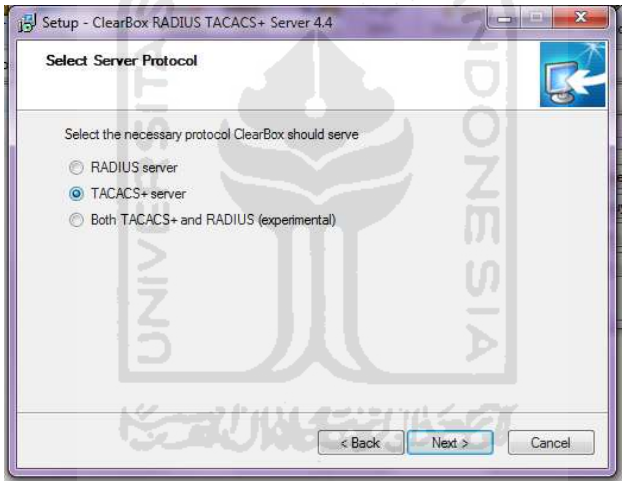
4.3.1.1 Instalasi Windows server

Analisis perbandingan protokol RADIUS dan TACACS+ mengunakan aplikasi berbasis windows, sehingga instalasi system operasi Windows server 2003 perlu dilakukan terlebih dahulu.

4.3.1.2 Instalasi server AAA (RADIUS/TACACS+)

Dalam menggunakan protokol AAA diperlukan adanya sebuah server yang mana bertujuan untuk melakukan proses komputasi terhadap paket-paket yang dikirim oleh NAS, berupa request proses-proses yang diinginkan (otentikasi, otorisasi, dan *accounting*).

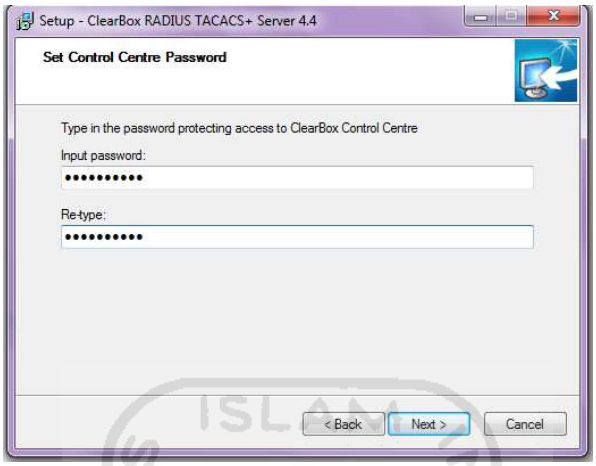
1. Tahap instalasi server ini meliputi instalasi RADIUS server dan instalasi TACACS+ server, yang keduanya menggunakan aplikasi ClearBox TACACS+ RADIUS server version 4.4.



Gambar 4.3 Instalasi server AAA

Pada proses instalasi terdapat pilihan server protokol, yang sesuai dengan protokol yang akan dijalankan. Aplikasi ini dipasang di atas Windows server 2003, dan juga aplikasi ini dipasang didua tempat yang berbeda untuk kebutuhan analisis, yaitu pada jaringan yang menggunakan protokol RADIUS dan pada jaringan yang menggunakan protokol TACACS+.

2. Gambar 4.4 adalah halaman untuk pengaturan password untuk memproteksi akses terhadap ClearBox.



Gambar 4.4 pengaturan password untuk akses ClearBox Control Center

4.3.1.3 Wireshark

Wireshark adalah salah satu tools yang digunakan untuk menganalisa jaringan. Tools digunakan untuk mempermudah analisa paket-paket data dari proses yang dijalankan oleh protokol RADIUS dan TACACS+. Tools ini dipasang pada suatu tempat yang terdapat diantara klien dan server AAA sehingga pertukaran paket dari kedua titik tersebut dapat ditangkap untuk kebutuhan analisis.

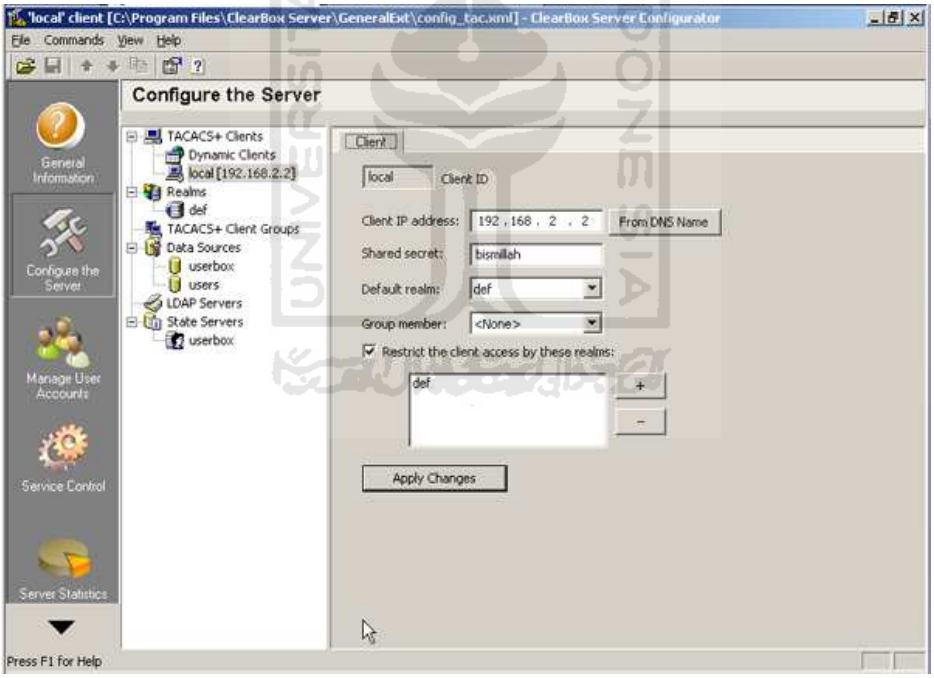
4.3.2 Konfigurasi server AAA

Konfigurasi server AAA diperlukan sebelum layanan dari server AAA dijalankan, yang berguna untuk mengidentifikasi klien beserta proses-proses yang dibutuhkan pada sebuah jaringan. Server AAA dikonfigurasi pertama dengan menentukan alamat IP dari klien AAA kemudian mengonfigurasi otentikasi, otorisasi, dan *accounting* yang digunakan.

4.3.2.1 Konfigurasi server TACACS+.

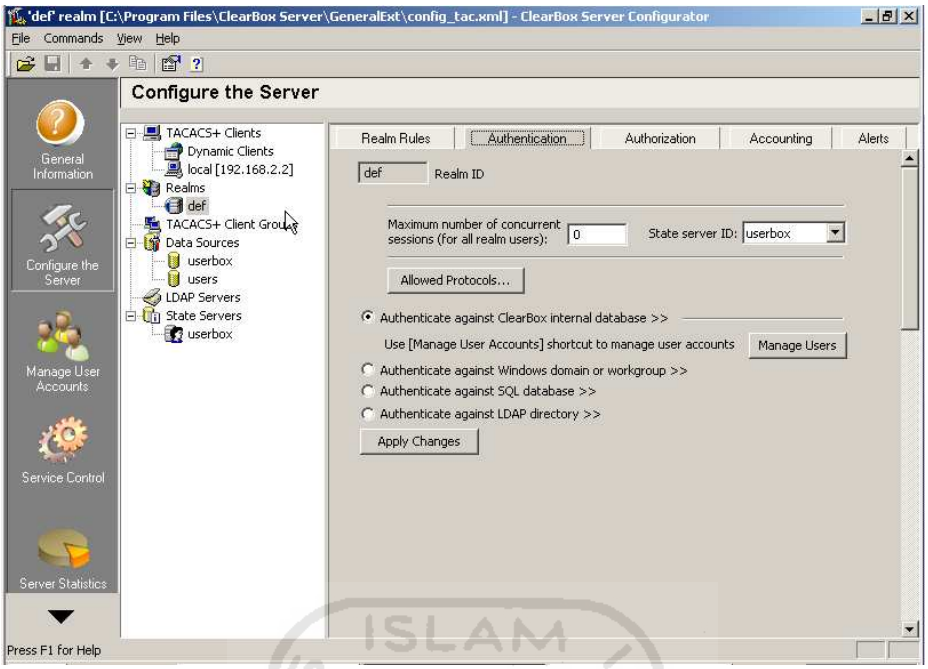
Konfigurasi yang dilakukan pada server TACACS+ yaitu menentukan alamat klien,menentukan *shared secret* (secret key), membuat user dsb. Dibawah ini adalah tahapan konfigurasi server TACACS+.

1. Pada gambar 4.5 dilakukan penentuan alamat IP dari klien TACACS+ atau Router yang bertindak sebagai NAS, juga menentukan *shared secret* yang digunakan oleh server dan klien dalam proses komunikasi, juga mendefinisikan realm yang digunakan, realm adalah satu kumpulan peraturan-peraturan yang digunakan untuk mendefinisikan bagaimana memproses permintaan yang masuk untuk otentikasi, otorisasi maupun *accounting*.



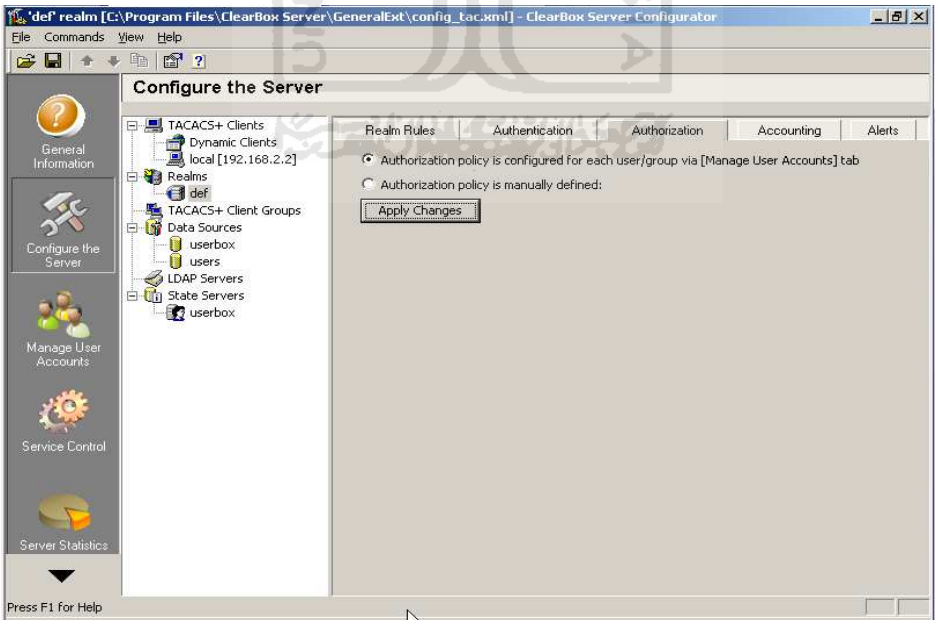
Gambar 4.5 Konfigurasi TACACS+

2. Pada gambar 4.6 otentikasi diatur berdasarkan pilihan bagaimana otentikasi dilakukan pada analisis ini yang digunakan adalah otentikasi dilakukan dengan melakukan cek kepada database internal dari aplikasi ClearBox.



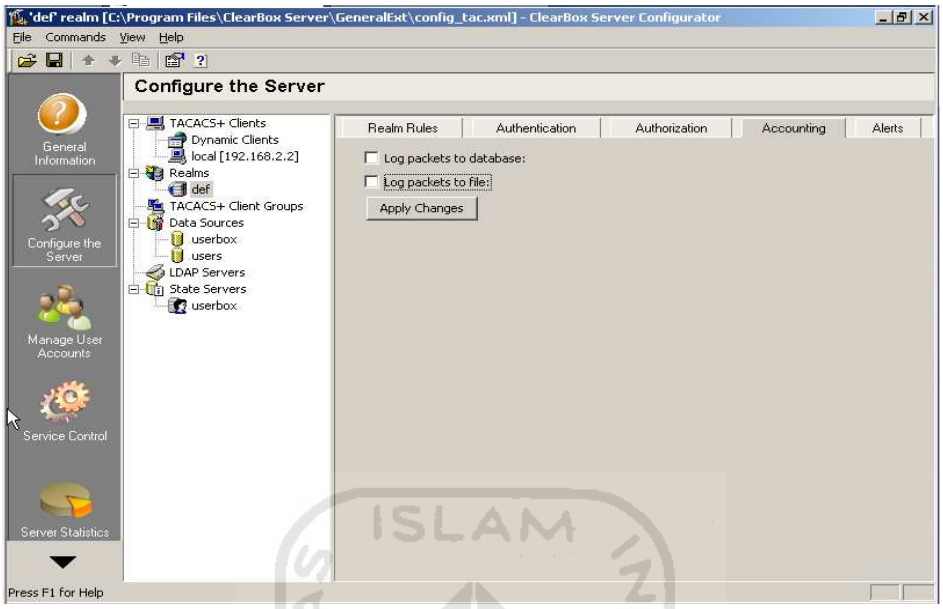
Gambar 4.6 Tab otentikasi

3. Pada gambar 4.7 otorisasi bisa dilakukan dengan dua pilihan yaitu konfigurasi otorisasi untuk tiap-tiap user atau tiap-tiap group atau konfigurasi secara manual dengan perintah-perintah SQL.



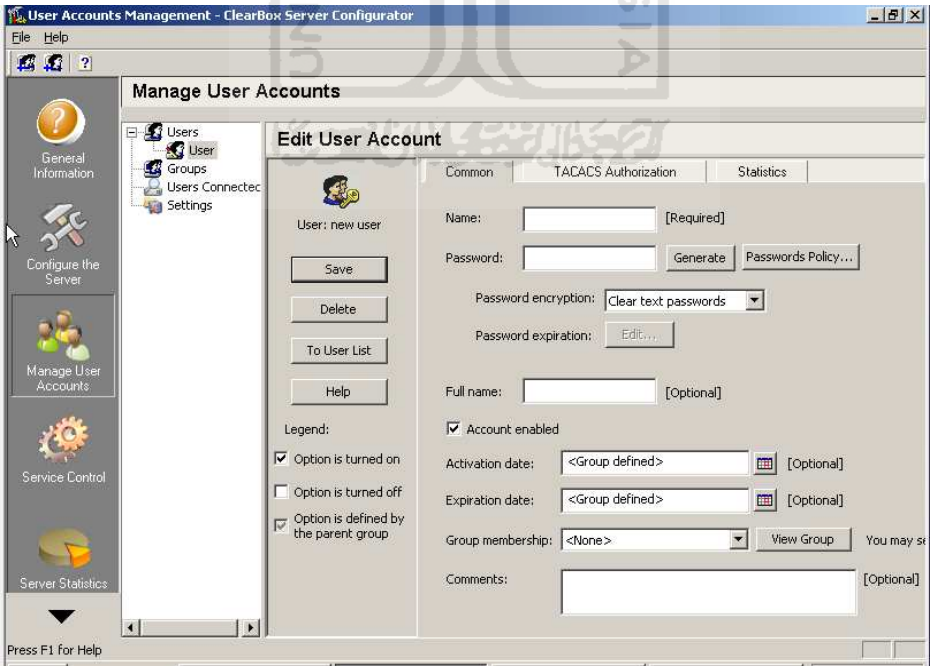
Gambar 4.7 Tab otorisasi

4. Pada gambar 4.8 pada tab *accounting* ada dua pilihan apakah data *accounting* dimasukan ke dalam database atau ke sebuah file yang ditentukan.



Gambar 4.8 Tab *accounting*

5. Pada gambar 4.9 merupakan tahap konfigurasi user, yang mana administrator bisa membuat user, password, group, menentukan otorisasi dari tiap-tiap group atau user.

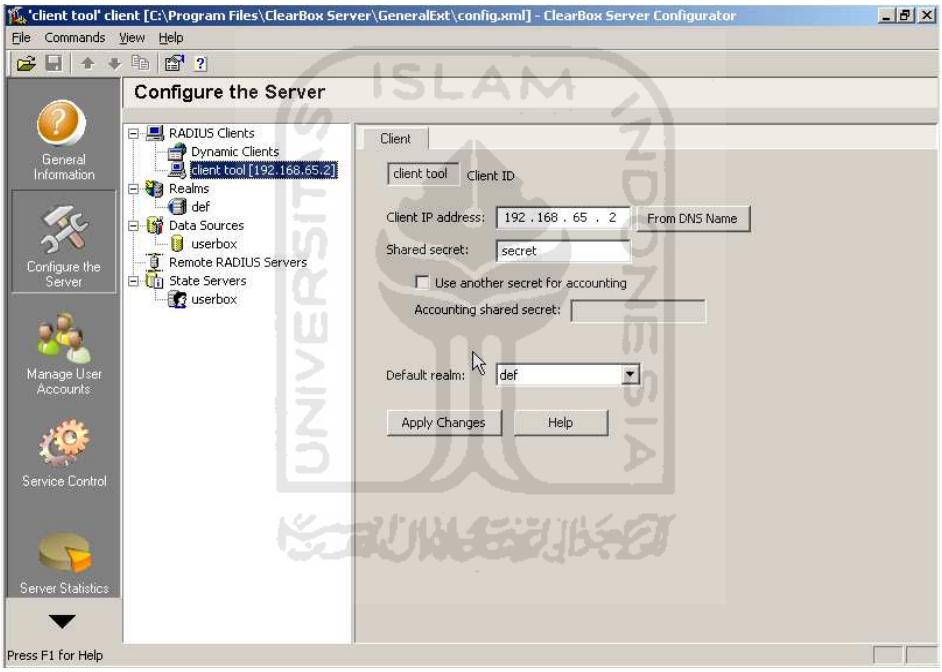


Gambar 4.9 Halaman konfigurasi user

4.3.2.2 Konfigurasi server RADIUS.

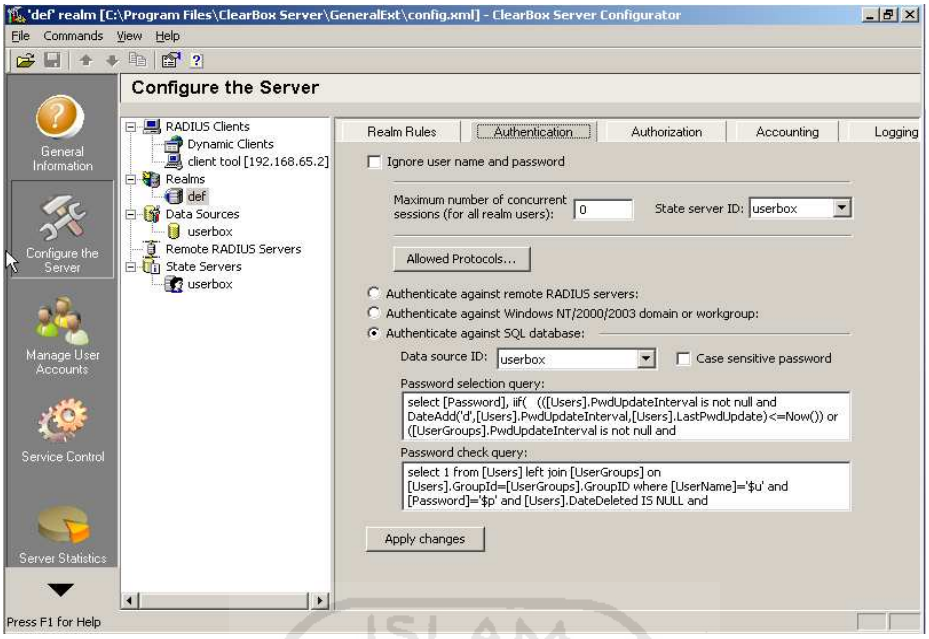
Konfigurasi yang dilakukan pada server RADIUS yaitu menentukan alamat klien,menentukan *shared secret* (secret key), membuat user dsb. Dibawah ini adalah tahapan konfigurasi server RADIUS.

1. Pada gambar 4.10 adalah halaman untuk melakukan konfigurasi terhadap RADIUS, seperti pada konfigurasi TACACS+, seperti penentuan alamat IP dari klien atau Router yang bertindak sebagai NAS, menentukan *shared secret*, mendefinisikan realm yang digunakan.



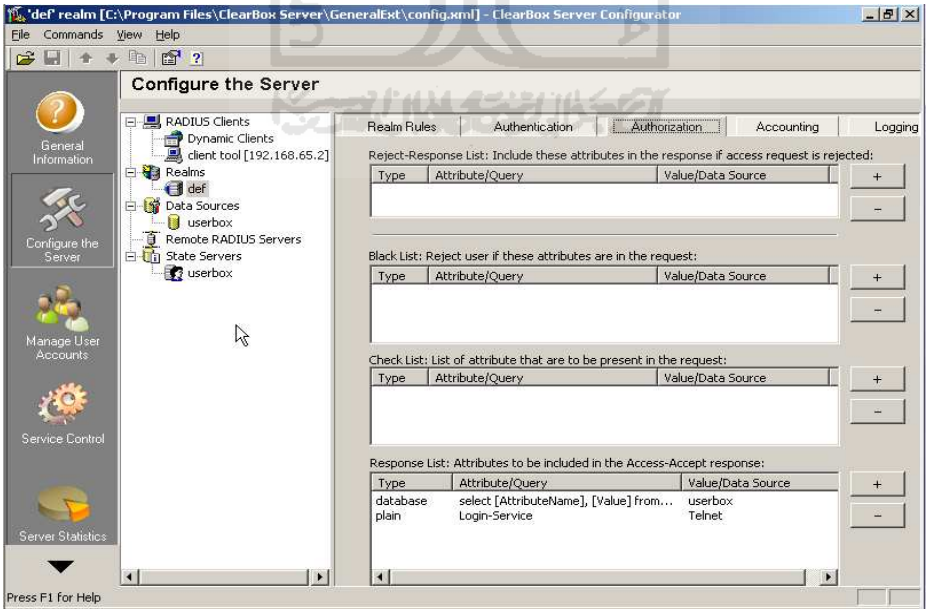
Gambar 4.10 Konfigurasi RADIUS

2. Pada gambar 4.11 yaitu menentukan otentikasi yang digunakan, dan memiliki fungsi yang sama dengan tab otentikasi pada server TACACS+.



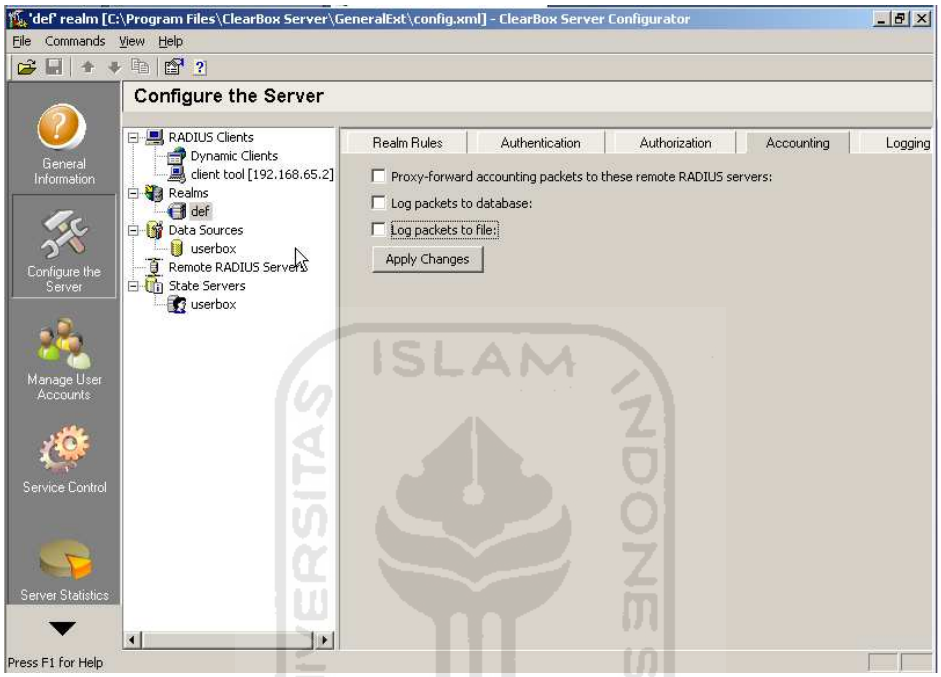
Gambar 4.11 Tab otentikasi

3. Pada gambar 4.12 Pada tab otorisasi terdapat beberapa kolom yang masing-masin memiliki fungsi yang berbeda diantaranya adalah kolom black-list yang bertujuan untuk menolak paket permintaan yang mempunyai atribut dan bernilai sama dengan atribut yang didefinisikan kedalam kolom black-list.



Gambar 4.12 Tab otorisasi

4. Pada gambar 4.13 merupakan tab untuk *accounting*, salah satu perbedaan dari tab *accounting* pada server TACACS+ adalah, adanya sebuah pilihan yang mana membolehkan server untuk mengirim data *accounting* ke server RADIUS yang berbeda, tetapi pada analisis ini pilihan tersebut tidak digunakan dikarenakan keterbatasan sumber daya untuk melakukan percobaan.



Gambar 4.13 Tab *accounting*

4.3.3 Konfigurasi Router (NAS)

Konfigurasi Router bertujuan untuk mengaktifkan proses AAA pada suatu jaringan. Router tersebut bertindak sebagai NAS yang memberikan akses kepada user berdasarkan hasil yang diterima dari server AAA. Sebelum melakukan konfigurasi klien dari server AAA, terlebih dahulu yang dilakukan adalah melakukan konfigurasi interface pada masing-masing router atau klien RADIUS dan TACACS+.

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#aaa new-model
Router(config)#aaa authentication login default group tacacs+ local
Router(config)#aaa authorization exec default group tacacs+ local
Router(config)#aaa accounting exec default start-stop group tacacs+
Router(config)#tacacs-server host 192.168. 2 .128
Router(config)#tacacs-server key bismillah
```

Gambar 4.14 Konfigurasi klien TACACS+

Penjelasan mengenai konfigurasi pada gambar 4.14 yaitu perintah *conf t* untuk masuk ke bagian terminal konfigurasi, kemudian perintah *aaa new-model* digunakan untuk mengaktifkan layanan AAA, kemudian *aaa authentication login default group TACACS+ local* memberi perintah kepada router untuk melakukan otentikasi dengan metode TACACS+ pada database server local ketika ada permintaan akses ke router, kemudian *aaa authorization exec default group TACACS+ local* memberi perintah kepada router untuk memulai proses otorisasi pada database server local jika ada permintaan, kemudian *aaa accounting exec default start-stop group TACACS+* untuk memulai dan memberhentikan proses *accounting* secara otomatis, perintah *tacacs-server host 192.168.65.128* memberitahukan router lokasi dari server TACACS+, *tacacs-server key bismillah* memberitahukan kepada router mengenai kunci (*shared secret/secret key*) yang digunakan dalam pertukaran data. konfigurasi klien TACACS+ dilakukan setelah melakukan konfigurasi *interface* yang berhubungan dengan server maupun dengan user.

```
interface FastEthernet0/0
 ip address 192.168.2.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.18.2 255.255.255.0
 duplex auto
 speed auto
```

Gambar 4.15 Konfigurasi *interface* pada klien TACACS+

Gambar 4.15 tampilan hasil konfigurasi *interface* pada klien TACACS+ yaitu pada *interface* fa0/0 klien TACACS+ diberikan alamat IP 192.168.2.2 yang berhubungan langsung dengan IP server TACACS+ yaitu 192.168.2.128, kemudian pada *interface* fa0/1 dengan IP 192.168.18.2, router berhubungan dengan user dengan alamat 192.168.18.1.

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#aaa new-model
Router(config)#aaa authentication login default group radius local
Router(config)#aaa authorization exec default group radius local
Router(config)#aaa accounting exec default start-stop radius
Router(config)#radius-server host 192.168.65.129 auth-port 1812 acct-port 1813
Router(config)#radius-server key secret
```

Gambar 4.16 Konfigurasi klien RADIUS

Konfigurasi klien RADIUS tidak begitu berbeda dengan konfigurasi klien TACACS+. Pada konfigurasi klien RADIUS ditambahkan pendefinisian port yang digunakan untuk otentikasi dan otorisasi.

```
interface FastEthernet0/0
 ip address 192.168.65.2 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.96.2 255.255.255.0
 duplex auto
 speed auto
```

Gambar 4.17 Konfigurasi *interface* pada klien RADIUS

Gambar 4.17 adalah tampilan hasil konfigurasi *interface* pada klien RADIUS yaitu pada *interface* fa0/0 klien RADIUS diberikan alamat IP 192.168.65.2 yang berhubungan langsung dengan IP server RADIUS yaitu 192.168.65.129, kemudian pada *interface* fa0/1 dengan IP 192.168.96.2, router berhubungan dengan user dengan alamat 192.168.96.1.

4.3.4 Analisis Protokol RADIUS dan TACACS+

Analisis yang akan dilakukan pada kedua jaringan yang telah mengimplementasikan teknologi AAA terdiri dari beberapa paramenter pembanding yaitu keamanan, fleksibilitas serta efektivitas dan efisiensi pada protokol RADIUS dan TACACS+. Paket dianalisa dengan menggunakan wireshark selain itu eksplorasi dilakukan agar bisa membandingkan fitur-fitur pada protokol RADIUS dan TACACS+ yang terkait dengan teknologi AAA.

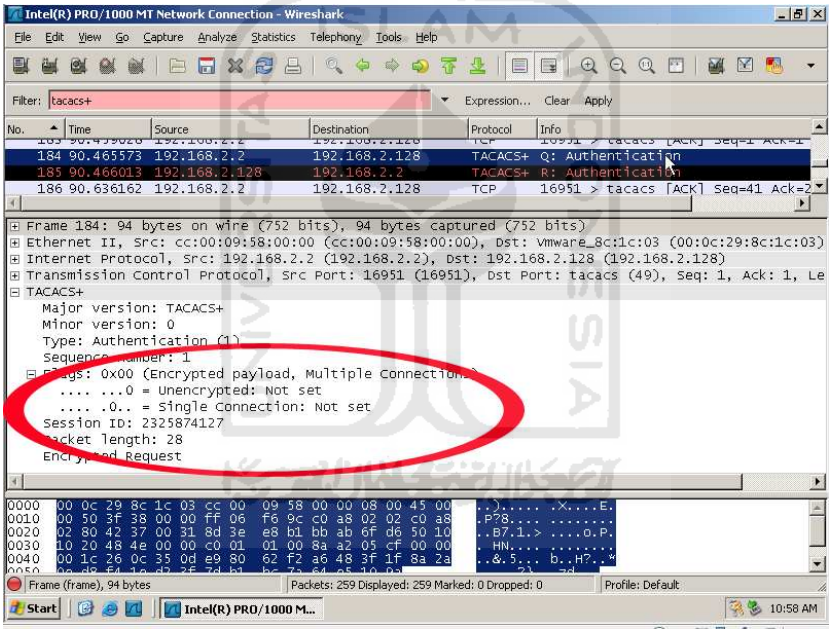
Pengujian akses dari user menggunakan aplikasi telnet, dikarenakan telnet telah ada secara default pada sistem yang digunakan pada analisis ini, berikut adalah hasil analisis dari protokol RADIUS dan TACACS+ berdasarkan parameter yang telah disebutkan :

4.3.4.1 Keamanan Protokol

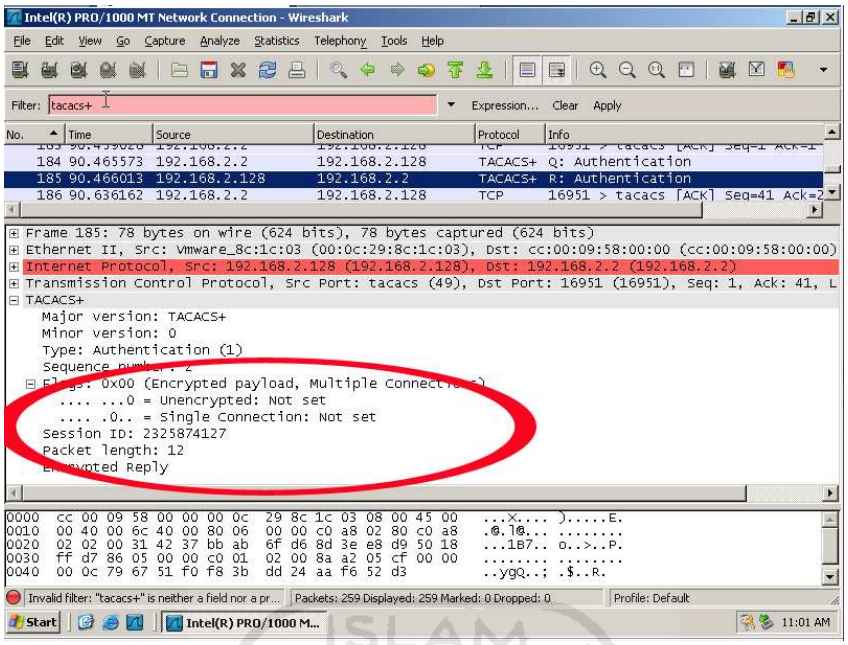
Pada analisa keamanan protokol paket yang tertangkap dari proses otentikasi dari kedua protokol akan dibandingan untuk menilai. Awalnya user akan melakukan akses dengan menggunakan telnet ke router atau klien server AAA, proses yang sama dilakukan pada kedua jaringan, jaringan yang menggunakan protokol RADIUS dan yang menggunakan TACACS+.

Adapun paket yang tertangkap pada proses pengintaian menggunakan wireshark pada kedua jaringan tersebut adalah sebagai berikut :

1. Pada gambar 4.18 dapat dilihat contoh paket yang tertangkap pada permulaan proses otentikasi, router atau NAS meminta enkripsi pada server TACACS+ untuk mengenkripsi paket berikutnya yang akan dikirim dari router ke server TACACS+. Begitu seterusnya sehingga seluruh paket permintaan maupun paket balasan yang berhubungan dengan protokol TACACS+ yaitu paket otentikasi pada gambar 4.19, paket otorisasi serta paket *accounting* terenkripsi sehingga tidak ada informasi nilai-nilai dari paket tersebut yang terlihat.

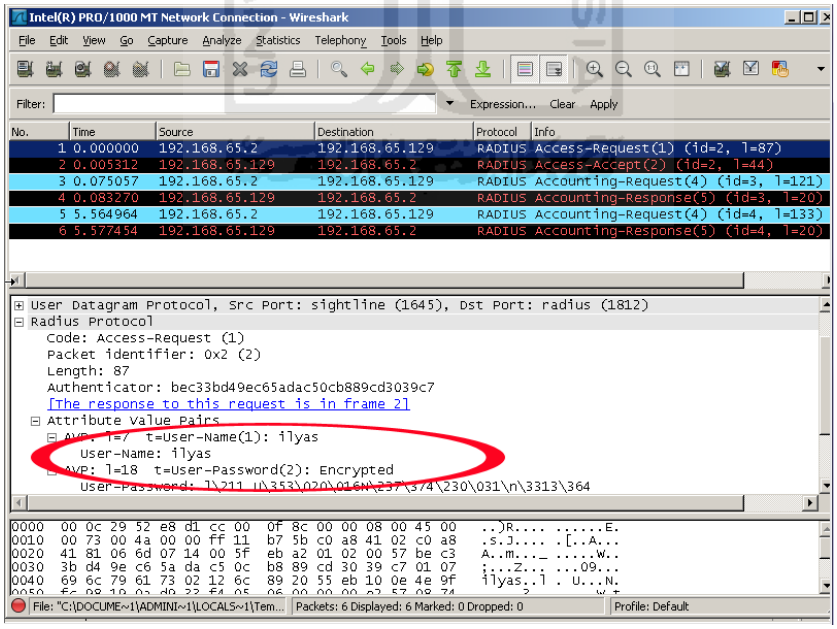


Gambar 4.18 Paket permintaan akses pada TACACS+

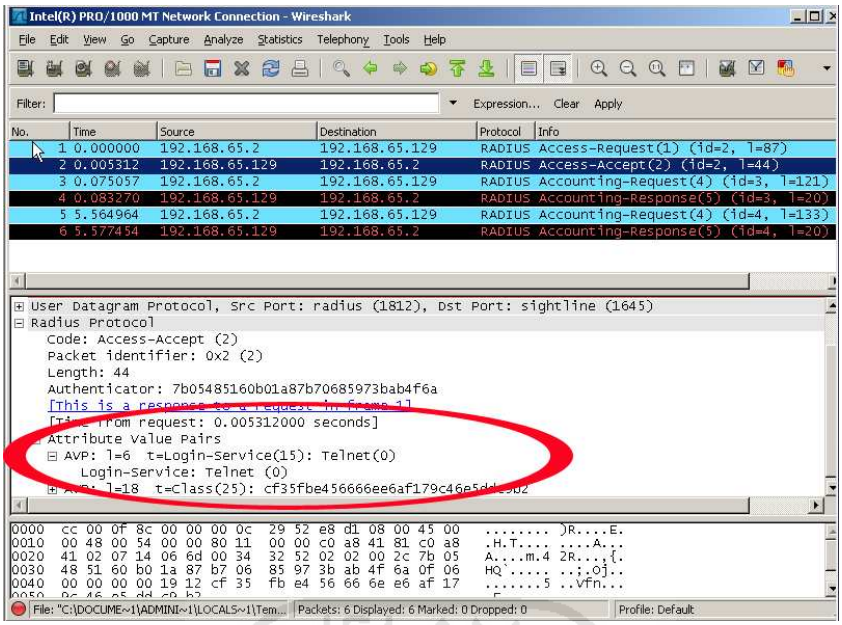


Gambar 4.19 Paket balasan permintaan pada TACACS+

2. Pada gambar 4.20 dan 4.21 dibawah dapat dilihat bahwa paket yang tertangkap dari protokol RADIUS hanya melakukan enkripsi terhadap password, sedangkan username dan attribute dari otorisasi tidak dienkripsi.



Gambar 4.20 Paket permintaan akses pada RADIUS

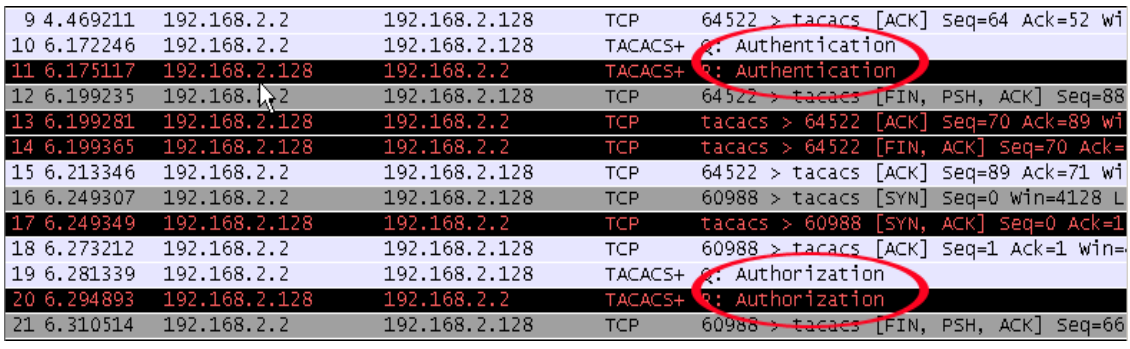


Gambar 4.21 Paket balasan permintaan pada RADIUS

4.3.4.2 Fleksibilitas Protokol

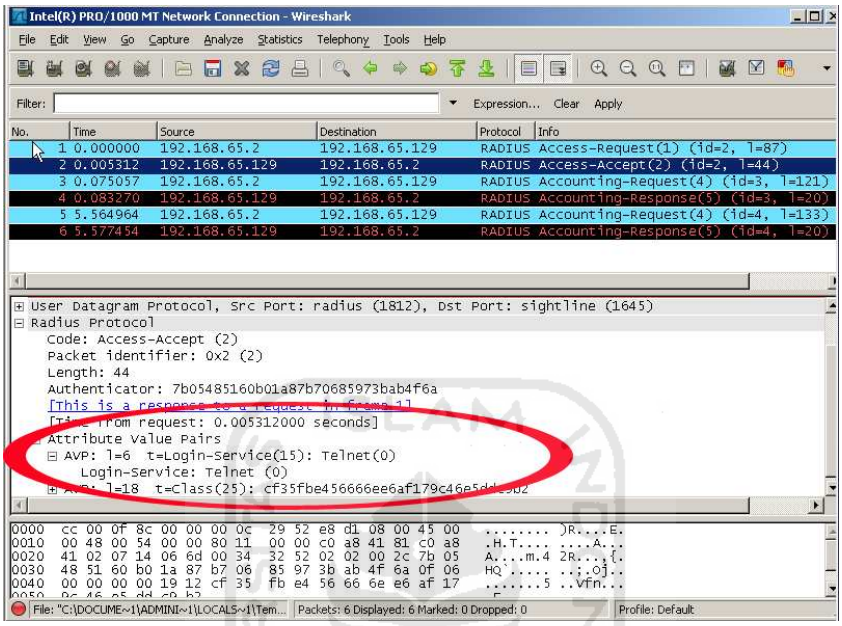
Fleksibilitas dalam bahasa adalah sesuatu yang mudah berganti atau luwes, sedangkan pada protokol keamanan, fleksibilitas berarti kemampuan protokol untuk beradaptasi dengan kebutuhan keamanan saat ini. Pada analisis fleksibilitas protokol selain analisa paket yang tertangkap dari pada proses AAA, dilakukan explorasi sederhana yang terkait. Hasil analisa fleksibilitas protokol adalah sebagai berikut :

1. Pada gambar 4.22 dapat dilihat pemisahan proses otentikasi dan otorisasi pada TACACS+, dimana otorisasi akan berjalan ketika ada permintaan khusus dari klien.



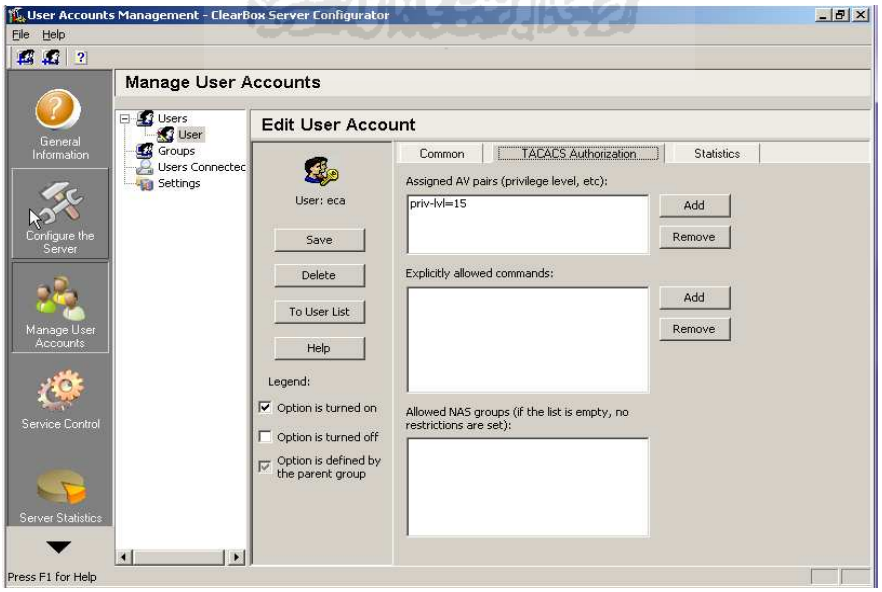
Gambar 4.22 Paket otentikasi dan otorisasi TACACS+

2. Pada gambar 4.23 dapat dilihat bahwa ketika server RADIUS memberikan balasan yang menyatakan bahwa otentikasi diterima, paket tersebut juga berisi otorisasi yang ada pada user tersebut, contohnya adalah paket dari nilai atribut yang ditandai, sehingga didapatkan informasi mengenai otorisasi yang digunakan terhadap user tersebut.



Gambar 4.23 Paket otentikasi sekaligus otorisasi RADIUS

3. Pada gambar 4.24 dapat dilihat bahwa pada protokol TACACS+ juga terdapat fitur yang membolehkan pengaturan tingkatan otorisasi untuk user ketika telah mengakses router. Fitur ini mendukung pengaturan per user ataupun per group.



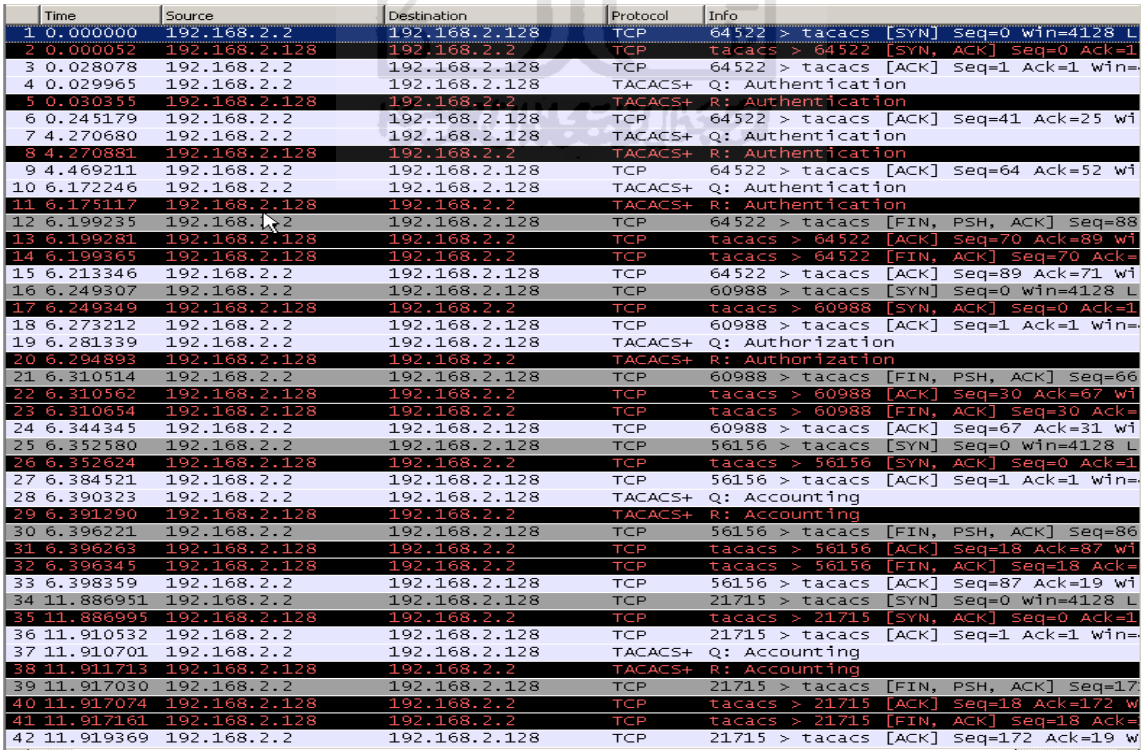
Gambar 4.24 Tab pengaturan user TACACS+

Fleksibilitas protokol TACACS+ salah satunya dikarenakan proses yang terpisah antara otentikasi dan otorisasi sehingga memungkinkan admin untuk mengkombinasikan TACACS+ dengan protokol otentikasi lainnya. Selain itu TACACS+ juga membolehkan pengaturan tingkatan-tingkatan otorisasi bagi user kepada router, contoh pada gambar 4.19 adalah percobaan pengaturan tingkatan otorisasi pada server ClearBox TACACS+.

4.3.4.3 Efektivitas dan Efisiensi Protokol

Pada analisis efektivitas dan efisiensi protokol analisa dilakukan setelah sebelumnya melakukan eksplorasi terkait otentikasi dan otorisasi pada protokol RADIUS dan TACACS+, kemudian melihat paket-paket yang terjadi dalam proses AAA pada protokol RADIUS dan TACACS+ sebagai salah satu bahan perbandingan dari sisi efektivitas dan efisiensi. Hasilnya adalah sebagai berikut :

1. Paket-paket pada Protokol TACACS+ lebih banyak karena TACACS+ menggunakan protokol TCP yang berorientasi pada konektivitas transport, sehingga berpengaruh pada jumlah keseluruhan paket yang ada pada satu sesi sederhana TACACS+.



	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [SYN] Seq=0 Win=4128 L
2	0.000052	192.168.2.128	192.168.2.2	TCP	tacacs > 64522 [SYN, ACK] Seq=0 Ack=1
3	0.028078	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [ACK] Seq=1 Ack=1 Win=
4	0.029965	192.168.2.2	192.168.2.128	TACACS+	Q: Authentication
5	0.030355	192.168.2.128	192.168.2.2	TACACS+	R: Authentication
6	0.245179	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [ACK] Seq=41 Ack=25 Wi
7	4.270680	192.168.2.2	192.168.2.128	TACACS+	Q: Authentication
8	4.270881	192.168.2.128	192.168.2.2	TACACS+	R: Authentication
9	4.469211	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [ACK] Seq=64 Ack=52 Wi
10	6.172246	192.168.2.2	192.168.2.128	TACACS+	Q: Authentication
11	6.175117	192.168.2.128	192.168.2.2	TACACS+	R: Authentication
12	6.199235	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [FIN, PSH, ACK] Seq=88
13	6.199281	192.168.2.128	192.168.2.2	TCP	tacacs > 64522 [ACK] Seq=70 Ack=89 Wi
14	6.199365	192.168.2.128	192.168.2.2	TCP	tacacs > 64522 [FIN, ACK] Seq=70 Ack=
15	6.213346	192.168.2.2	192.168.2.128	TCP	64522 > tacacs [ACK] Seq=89 Ack=71 Wi
16	6.249307	192.168.2.2	192.168.2.128	TCP	60988 > tacacs [SYN] Seq=0 Win=4128 L
17	6.249349	192.168.2.128	192.168.2.2	TCP	tacacs > 60988 [SYN, ACK] Seq=0 Ack=1
18	6.273212	192.168.2.2	192.168.2.128	TCP	60988 > tacacs [ACK] Seq=1 Ack=1 Win=
19	6.281339	192.168.2.2	192.168.2.128	TACACS+	Q: Authorization
20	6.294893	192.168.2.128	192.168.2.2	TACACS+	R: Authorization
21	6.310514	192.168.2.2	192.168.2.128	TCP	60988 > tacacs [FIN, PSH, ACK] Seq=66
22	6.310562	192.168.2.128	192.168.2.2	TCP	tacacs > 60988 [ACK] Seq=30 Ack=67 Wi
23	6.310654	192.168.2.128	192.168.2.2	TCP	tacacs > 60988 [FIN, ACK] Seq=30 Ack=
24	6.344345	192.168.2.2	192.168.2.128	TCP	60988 > tacacs [ACK] Seq=67 Ack=31 Wi
25	6.352580	192.168.2.2	192.168.2.128	TCP	56156 > tacacs [SYN] Seq=0 Win=4128 L
26	6.352624	192.168.2.128	192.168.2.2	TCP	tacacs > 56156 [SYN, ACK] Seq=0 Ack=1
27	6.384521	192.168.2.2	192.168.2.128	TCP	56156 > tacacs [ACK] Seq=1 Ack=1 Win=
28	6.390323	192.168.2.2	192.168.2.128	TACACS+	Q: Accounting
29	6.391290	192.168.2.128	192.168.2.2	TACACS+	R: Accounting
30	6.396221	192.168.2.2	192.168.2.128	TCP	56156 > tacacs [FIN, PSH, ACK] Seq=86
31	6.396263	192.168.2.128	192.168.2.2	TCP	tacacs > 56156 [ACK] Seq=18 Ack=87 Wi
32	6.396345	192.168.2.128	192.168.2.2	TCP	tacacs > 56156 [FIN, ACK] Seq=18 Ack=
33	6.398359	192.168.2.2	192.168.2.128	TCP	56156 > tacacs [ACK] Seq=87 Ack=19 Wi
34	11.886951	192.168.2.2	192.168.2.128	TCP	21715 > tacacs [SYN] Seq=0 Win=4128 L
35	11.886995	192.168.2.128	192.168.2.2	TCP	tacacs > 21715 [SYN, ACK] Seq=0 Ack=1
36	11.910532	192.168.2.2	192.168.2.128	TCP	21715 > tacacs [ACK] Seq=1 Ack=1 Win=
37	11.910701	192.168.2.2	192.168.2.128	TACACS+	Q: Accounting
38	11.911713	192.168.2.128	192.168.2.2	TACACS+	R: Accounting
39	11.917030	192.168.2.2	192.168.2.128	TCP	21715 > tacacs [FIN, PSH, ACK] Seq=17
40	11.917074	192.168.2.128	192.168.2.2	TCP	tacacs > 21715 [ACK] Seq=18 Ack=172 W
41	11.917161	192.168.2.128	192.168.2.2	TCP	tacacs > 21715 [FIN, ACK] Seq=18 Ack=
42	11.919369	192.168.2.2	192.168.2.128	TCP	21715 > tacacs [ACK] Seq=172 Ack=19 W

Traffic	Captured	Displayed	Marked
Avg. bytes/sec	245.902		
Avg. MBit/sec	0.002		
Avg. packet size	69.786 bytes		
Avg. packets/sec	3.524		
Between first and last packet	11.919 sec		
Bytes	2931		
Packets	42	42	0

Gambar 4.25 Paket dan jumlah Byte pada TACACS+

2. Dibandingkan dengan banyaknya paket pada satu sesi protokol TACACS+, paket dari protokol RADIUS jauh lebih sedikit, juga berpengaruh pada jumlah byte atau besaran paket keseluruhan yang terjadi pada proses otentikasi, otorisasi dan *accounting* pada RADIUS. Berikut hasil dari paket yang tertangkap pada satu sesi sederhana dari protokol RADIUS.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.65.2	192.168.65.129	RADIUS	Access-Request(1) (id=2, l=87)
2	0.005312	192.168.65.129	192.168.65.2	RADIUS	Access-Accept(2) (id=2, l=44)
3	0.075057	192.168.65.2	192.168.65.129	RADIUS	Accounting-Request(4) (id=3, l=121)
4	0.083270	192.168.65.129	192.168.65.2	RADIUS	Accounting-Response(5) (id=3, l=20)
5	5.564964	192.168.65.2	192.168.65.129	RADIUS	Accounting-Request(4) (id=4, l=133)
6	5.577454	192.168.65.129	192.168.65.2	RADIUS	Accounting-Response(5) (id=4, l=20)

Traffic	Captured	Displayed	Marked
Avg. bytes/sec	121.382		
Avg. MBit/sec	0.001		
Avg. packet size	112.833 bytes		
Avg. packets/sec	1.076		
Between first and last packet	5.577 sec		
Bytes	677		
Packets	6	6	0

Gambar 4.26 Paket dan jumlah Byte pada RADIUS

Dari gambar 4.25 dan 4.26 dapat kita lihat bahwa pada satu proses sederhana TACACS+ dibutuhkan 42 paket pada proses otentikasi, otorisasi dan *accounting* dengan besaran keseluruhan paket yaitu 2931 Bytes, sedangkan pada RADIUS hanya memerlukan 6 paket untuk melakukan otentikasi, otorisasi dan *accounting* dengan besaran keseluruhan paket lebih kecil yaitu 677 Bytes.

Setelah melakukan analisa terhadap parameter-paremeter diatas, hasil dari analisa dapat ditampilkan dalam bentuk sebuah tabel sederhana yang berisi poin-poin hasil analisa protokol RADIUS dan TACACS+.

Tabel 4.1 Hasil Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA

Parameter Pembanding	Protokol RADIUS	Protokol TACACS+
Keamanan	• Enkripsi password • Non enkripsi username • Non enkripsi paket otorisasi	• Enkripsi keseluruhan paket
Fleksibilitas	• Otentikasi dan otorisasi tidak dipisah	• Otentikasi dan otorisasi merupakan proses terpisah • Membolehkan kontrol otorisasi user pada router, per user/ per group
Efektivitas dan efisiensi	• Efektif digunakan pada jaringan dengan badwidth terbatas • Efisien waktu	• Efektif untuk pilihan keamanan maksimal dan fleksibitas • Tidak efisien pada jaringan dengan bandwidth terbatas

Pada tabel hasil analisis diatas dapat diketahui beberapa perbedaan kinerja protokol RADIUS dan TACACS+ dilihat dari sisi keamanan, fleksibilitas, serta efektivitas dan efisiensi dalam mengimplementasikan teknologi AAA. Berikut adalah penjelasan dari tabel diatas berdasarkan parameter yang ada.

a. Keamanan

Dari hasil analisis bahwa protokol TACACS+ memiliki tingkat keamanan yang cukup baik dimana seluruh paket TACACS+ yang terdapat didalam proses AAA terenkripsi, sedangkan pada protokol RADIUS hanya password dari username yang terenkripsi dan untuk paket berisi

otorisasi dan username begitu juga dengan *accounting* tidak dienkripsi, ini memungkinkan seorang punyusup bisa memperoleh informasi tambahan terkait.

b. Fleksibilitas

Dari hasil analisis bahwa protokol RADIUS kurang memiliki fleksibilitas yang bagus contohnya yaitu proses otentikasi dan otorisasi digabung sehingga tidak mudah untuk dipisahkan, sedangkan pada protokol TACACS+ proses otentikasi dan otorisasi terpisah sehingga memungkinkan TACACS+ digunakan bersama protokol otentikasi lainnya.

c. Efektifitas dan Efisiensi

Dari hasil analisis dapat dilihat bahwa Protokol TACACS+ efektif bagi suatu infrastruktur jaringan yang menginginkan keamanan dan fleksibilitas maksimal. Tetapi protokol TACACS+ tidak efisien jika digunakan pada jaringan dengan kuota bandwidth terbatas dikarenakan oleh karakteristik prokotol transfer yang digunakan TACACS+ yaitu TCP membutuhkan sumber daya bandwidth yang mencukupi. Sedangkan RADIUS efektif digunakan pada jaringan dengan bandwidth terbatas. Hal ini dikarenakan karakteristik protokol UDP yang digunakan salahsatunya untuk menghemat sumber daya dan juga lebih efisien terhadap waktu.

BAB IV
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA, maka dapat ditarik beberapa kesimpulan. Kesimpulan tersebut antara lain sebagai berikut :

a. Protokol RADIUS dan TACACS+ dapat disimulasikan secara sederhana dengan bantuan GNS3 yang bertindak sebagai simulator router cisco, VMware dan VirtualBox yang masing-masing bertindak sebagai server AAA dan PC user.

b. Server AAA tidak bisa dibangun diatas virtualbox, dikarenakan virtual LAN yang ada pada guest OS pada virtualbox tidak berhubungan langsung dengan router, terbatasi dengan semacam router virtual yang ada pada virtualbox sehingga proses pengidentifikasi alamat server AAA tidak dapat dijalankan.

c. Enkripsi paket secara keseluruhan pada protokol TACACS+ membuat keamanan protokol tersebut lebih terjamin, sehingga tidak ada informasi mengenai username, password, maupun paket-paket otorisasi dan *accounting* yang dapat diketahui oleh pihak ketiga.

d. Bahwa TACACS+ server memiliki beberapa kelebihan salahsatunya yaitu bisa dikombinasikan atau bekerja bersamaan dengan protokol otentikasi lainnya. Contohnya klien TACACS+ bisa melakukan otentikasi pada protokol otentikasi lain yang telah ditentukan sebelumnya, kemudian setelah user terotentikasi, klien TACACS+ baru akan melakukan otorisasi kepada server TACACS+

5.2 Saran

Berdasarkan kekurangan pada Analisis perbandingan kinerja protokol RADIUS dan TACACS+ dalam mengimplementasikan teknologi AAA , maka saran penyusun untuk pengembangan penelitian dimasa yang akan datang adalah sebagai berikut :

a. Diinginkan dimasa mendatang penelitian mengenai protokol AAA lebih lengkap dengan beberapa server AAA yang dikombinasi, juga bila dimungkinkan pengunaan teknik load balancing pada server AAA.
b. Untuk keamanan keseluruhan jaringan tidak hanya terpusat pada penggunaan protokol AAA seperti RADIUS dan TACACS+, pada topologi yang digunakan untuk keperluan analisa keamanan hanya ada pada paket yang ditukarkan dari klien AAA dan server AAA, sementara paket yang digunakan untuk percobaan akses ke router dari user tidak aman, dikarenakan router yang terhubung langsung dengan jaringan memudahkan pihak ketiga untuk melakukan pengintaian paket yang masuk ke router.
c. Untuk analisis selanjutnya diinginkan ke arah analisa berbagai atribut yang terdapat pada protokol RADIUS dan TACACS+ dimana perbedaan antara atribut pada RADIUS dan TACACS+ cukup banyak. Atribut pada protokol RADIUS dan TACACS+ digunakan antaralain sebagai penentuan otorisasi bagi user, penentuan proses yang dilakukan server AAA untuk *Accounting* dan sebagainya, sehingga perlu diketahui keseluruhan fungsi, kelebihan atau kekurangan dari keseluruhan atribut yang dimiliki RADIUS dan TACACS+.

DAFTAR PUSTAKA

[CAR04] Carroll, B. 2004. Cisco Access Control Security : AAA Administrative Service. Indianapolis, USA.

[CIS10] Cisco System, Inc. AAA Overview. (on-line) available at http://www.cisco.com/en/US/docs/ios/12_2/security/configuration/guide/scfaaa.html (diakses terakhir tanggal 3 Mei 2010).

[CIS08] Cisco System, Inc. 2008. TACACS+ and RADIUS Comparison. (on-line) available at http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml (diakses terakhir Februari 2010).

[RIG00] Rigney, C. 2000. RADIUS Accounting. (on-line) available at http://tools.ietf.org/html/rfc2866 (diakses terakhir tanggal 29 Mei 2010).

[SAN02] SANS Institute. 2002. Understanding and Implementing TACACS+. (on-line) available at http://www.sans.org/reading_room/whitepapers/networkdevs/understanding-implementing-tacacs-plus_117 (diakses terakhir Februari 2010).

[THO05] Thomas, T. 2005. Network Security First Step, Yogyakarta : ANDI.