# PERANCANGAN DAN IMPLEMENTASI SISTEM AUTENTIKASI DAN OTORISASI DENGAN SERVER AAA (*AUTHENTICATION, AUTHORIZATION, ACCOUNTING*) BERBASIS PROTOKOL RADIUS

**SKRIPSI**

Diajukan untuk memenuhi sebagian prasyarat
memperoleh gelar Sarjana Teknik


Disusun oleh:

**WINDA SEPTARINI**
**NIM. 0410630093**

**DEPARTEMEN PENDIDIKAN NASIONAL**
**UNIVERSITAS BRAWIJAYA**
**FAKULTAS TEKNIK**
**JURUSAN ELEKTRO**
**MALANG**
**2010**

# PERANCANGAN DAN IMPLEMENTASI SISTEM AUTENTIKASI DAN OTORISASI DENGAN SERVER AAA (*AUTHENTICATION, AUTHORIZATION, ACCOUNTING*) BERBASIS PROTOKOL RADIUS

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Teknik


Disusun oleh :

**WINDA SEPTARINI**
**NIM. 0410630093**


Dosen Pembimbing:

| | |
|---|---|
| **Raden Arif Setyawan ST., MT** | **Arief Andy Soebroto, ST., M.Kom.** |
| **NIP. 19750819 199903 001** | **NIP. 19720425 199903 1 002** |

# PENGANTAR

Berkat rahmat Allah SWT beserta kasih sayangnya, penulis bersyukur telah dapat menyelesaikan tugas akhir dengan judul "**Perancangan dan Implementasi Sistem Autentikasi dan Otorisasi Dengan Server AAA (*Authentication*, *Authorization*, *Accounting*) Berbasis Protokol Radius**". Tugas akhir ini disusun untuk memenuhi sebagian persyaratan untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Program Studi Teknik Informatika dan Komputer Fakultas Teknik Universitas Brawijaya Malang.

Penulis mengucapkan banyak terimakasih atas dukungan berbagai pihak yang memberikan dukungan dan bimbingan sehingga tugas akhir ini dapat terselesaikan dengan lancar tanpa halangan yang berarti. Pada kesempatan ini penulis mengucapkan banyak terimakasih kepada:

1. Bapak Rudy Yuwono, ST., M.Sc. dan Bapak Muhammad Aziz Muslim, ST, MT. selaku Ketua dan Sekretaris Jurusan Teknik Elektro Universitas Brawijaya serta segenap Bapak/Ibu Dosen, Staff Administrasi dan Perpustakaan Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.
2. Bapak R.Arief Setyawan, ST., MT. selaku Dosen Pembimbing I yang telah memberikan bimbingan, inspirasi dan motivasi pada saat penyusuna tugas akhir ini.
3. Bapak Arief Andy Soebroto, ST., M.Kom. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan kritik yang membangun dalam penyusunan tugas akhir ini.
4. Kedua orang tua penulis yang tidak lelah memberikan dorongan dan doa demi terselesaikannya tugas akhir ini.
5. Segenap staf PPTI Universitas Brawijaya yang telah memberikan kesempatan bagi penulis untuk mengembangkan diri dan memberikan sarana yang lebih dari cukup demi terselesaikannya tugas akhir ini.

6. Teman-teman mahasiswa Teknik Elektro Universitas Brawijaya, khususnya angkatan 2004 yang telah saling mendukung satu sama lain agar tetap kompak dan saling membantu dalam penyelesaian tugas akhir ini. Begitu juga dengan para anggota RisTIE, para asisten Lab Telekomunikasi yang telah memberikan tempat dan kesempatan pada penulis untuk menggunakan fasilitas jaringannya.
7. Semua pihak yang tidak bisa disebutkan satu per satu baik terlibat secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis berharap dan berdoa Allah SWT akan melimpahkan balasan kebaikan kepada pihak-pihak tersebut.

Penulis menyadari bahwa tugas akhir ini masih memiliki kekurangan. Namun hal tersebut tidak menyurutkan niat penulis untuk dapat bermanfaat bagi pembacanya maupun pihak lain. Semoga Allah SWT selalu melimpahkan kita dengan hidayah dan rahmat. Amin.

Malang, 18 Januari 2010

Penulis

## ABSTRAK

Winda Septarini. 2009. *Perancangan Dan Implementasi Sistem Autentikasi Dan Otorisasi Dengan Server AAA (Authentication, Authorization, Accounting) Berbasis Protokol Radius*. Dosen Pembimbing : Raden Arief Setyawan, ST., MT. dan Arief Andi Soebroto, ST., MKom.

Universitas Brawijaya (UB) sejak beberapa tahun terakhir telah membangun dan mengembangkan jaringan komputer di tiap jurusan, fakultas, maupun institusinya. Namun kemudahan mendapatkan akses internet dan bertambahnya user di jaringan UB menimbulkan suatu permasalahan. Permasalahan tersebut misalnya adanya user-user yang tidak berhak menggunakan jaringan internal UB dikarenakan tersebarnya access point di lingkungan kampus UB. Hal tersebut merupakan latar belakang dilakukannya penelitian penggunaan Radius pada jaringan Universitas Brawijaya.

Pada skripsi ini dirancang arsitektur sistem AAA (Radius) yang sesuai dengan jaringan yang ada kemudian diukur performansi sistemnya. Performansi sistem yang diukur adalah Quality of service (QoS), yaitu perbandingan transfer rate pada saat sebelum dan sesudah sistem diterapkan dan delay autentikasi pada beberapa variasi jumlah user yang melakukan autentikasi bersamaan. Dalam skripsi ini juga membahas tentang mekanisme handoff pada jaringan yang telah diterapkan sistem ini. Dari pengujian didapatkan hasil bahwa user pada jaringan prototype diharuskan melakukan autentikasi sebelum melakukan koneksi internet.

Dari penelitian yang dilakukan dapat disimpulkan bahwa sistem AAA dengan protokol Radius yang diujikan bekerja dengan baik pada sistem dengan server Radius dan server basis data terletak zona yang bukan merupakan jaringan internal. NAS pada rancangan ini juga berfungsi sebagai end router pada jaringan internal UB yang terdistribusi pada tiap-tiap segmen jaringan pada jaringan internal UB. Namun terjadi penurunan kualitas layanan jaringan (transfer rate) pada jaringan yang diterapkan Radius dibandingkan dengan jaringan yang belum diterapkan Radius yaitu sebesar 2,65%. Sistem mampu menangani beberapa autentikasi yang terjadi bersamaan. Namun performanya berkurang seiring dengan peningkatan jumlah autentikasi yang terjadi bersamaan. Pada jumlah autentikasi bersamaan lebih dari 500, semakin besar jumlah autentikasi yang terjadi bersamaan, rata-rata delay autentikasi cenderung tetap. Pada peristiwa handoff dengan user Radius, user diharuskan melakukan autentikasi ulang Radius pada saat setelah terkoneksi dengan access point tujuan untuk mendapatkan layanan jaringan karena Radius menganggap perpindahan access point sebagai terputusnya koneksi fisik, sehingga session dianggap berakhir oleh NAS. Karena proses autentikasi Radius pada handoff dilakukan manual, maka penghitungan latency tidak bisa dilakukan.

*Kata Kunci : AAA, Radius, NAS, authentication, authorization, handoff*

## DAFTAR ISI

## DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR LAMPIRAN

## DAFTAR ISTILAH

| | |
|---|---|
| ***Server*** | Aplikasi yang memberikan pelayanan kepada user. *Server* dapat menerima permintaan (*request*), melakukan pelayanan yang diminta, kemudian mengembalikan sebagai reply. *Server* dapat melayani multi request bersamaan. [DHO-07]. |
| ***Client*** | Pihak yang meminta pelayanan [DHO-07]. |
| ***Router*** | Perangkat yang menghubungkan jaringan pada layer IP (dalam protokol TCP/IP) dan mengarahkan jalur paket data. Perangkat ini mampu memilih jalur yang terbaik untuk pengiriman data, karena memiliki *routing* [DHO-07]. |
| ***Access point*** | Sebuah perangkat jaringan yang bekerja seperti *bridge* yang menghubungkan jaringan wireless 802.11 dengan jaringan kabel [GAS-05]. |
| **Analisis Paket** | Proses penangkapan dan menampilkan paket yang lewat pada suatu *interface* jaringan [STI-05]. |
| **Atribut** | Salah satu bagian pada paket Radius yang membawa informasi yang spesifik mengenai autentikasi, otorisasi, dan detail konfigurasi untuk *request* dan *reply* [RFC-286]. |
| **Sesi** | Layanan yang diberikan NAS kepada user sejak user berhasil terautentikasi hingga user *logoff*. |
| **Log in** | Proses yang bertujuan untuk mendapatkan akses pribadi oleh user yang ditandai dengan identifikasi user [WIK-10]. |
| **Log out** | Proses berhenti dari sebuah akses setelah user melakukan proses log in sebelumnya [WIK-10]. |

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Akses internet merupakan hal yang penting untuk menunjang kebutuhan akan informasi pada dunia pendidikan, terutama civitas kampus. Universitas Brawijaya (UB) sejak beberapa tahun terakhir telah membangun dan mengembangkan jaringan komputer di tiap jurusan, fakultas, maupun institusinya. Hal ini dilakukan untuk memenuhi kebutuhan civitas kampus akan akses internet. Jaringan komputer tersebut dibangun dengan media kabel (*wired*) maupun nirkabel (*wireless*).

Pada jaringan kabel di UB digunakan media kabel UTP (*Unshielded Twisted Pair*) dan serat optik. Jaringan kabel ini kebanyakan diterapkan pada gedung maupun koneksi antar gedung. Sedangkan pada jaringan *wireless* terdapat beberapa *access point* di gedung maupun ruang terbuka. Banyaknya *access point* ini memungkinkan terjadinya peristiwa *handoff* (perpindahan layanan dari satu *access point* ke *access point* yang lain) bagi user yang bergerak.

Kemudahan mendapatkan akses internet dan bertambahnya user di jaringan UB menimbulkan suatu permasalahan. Permasalahan tersebut misalnya adanya user-user yang tidak berhak menggunakan jaringan internal UB dikarenakan tersebarnya *access point* di lingkungan kampus UB. Identitas user yang melakukan aktifitas *hacking* tidak akan terlacak jika user-user yang menggunakan jaringan UB tidak dikendalikan. Hal ini akan mengganggu keamanan jaringan UB [GAS-05].

Untuk mengatasi permasalahan tersebut diperlukan sistem pencatatan user untuk mengendalikan user dan aktifitasnya. Mekanisme yang mendukung sistem tersebut adalah *Authentication*, *Authorization*, *Accounting* (AAA) [NAK-05]. AAA adalah serangkaian mekanisme yang terdiri dari proses autentikasi, otorisasi, dan *accounting*. Autentikasi adalah proses pengidentifikasian user. Otorisasi adalah

proses pengecekan user apakah boleh mengakses sumber daya tertentu. Sedangkan *accounting* adalah proses pencatatan sumber daya jaringan yang telah dipakai oleh user [NAK-05].

Beberapa protokol yang menerapkan mekanisme AAA adalah Radius dan Diameter. Radius adalah protokol yang paling banyak dipakai. Hal ini dikarenakan Radius mudah diimplementasikan dan didukung oleh berbagai perangkat jaringan, meski memiliki beberapa kelemahan sekuritas [NAK-05]. Sedangkan Diameter merupakan protokol yang memperbarui Radius. Diameter dirancang untuk memperbaiki kekurangan-kekurangan protokol Radius. Namun Diameter masih sulit untuk diimplementasikan karena masih ada beberapa fitur yang belum bisa diimplementasikan oleh aplikasi yang menerapkan protokol Diameter [DJU-07].

Dengan pertimbangan kelebihan dan kelemahan pada protokol Radius dan Diameter, maka pada skripsi ini dipilih protokol Radius untuk sistem pencatatan user jaringan komputer UB sebagai solusi atas permasalahan yang telah dipaparkan.

## 1.2 Rumusan Masalah

Dengan latar belakang tersebut, maka diperoleh rumusan masalah sebagai berikut.

- Merancang dan membangun sistem AAA sesuai dengan analisis kebutuhan dengan Radius sebagai protokolnya.
- Menganalisis proses authentikasi dan otorisasi yang terjadi.
- Membandingkan *transfer rate* pada saat menggunakan *resource* jaringan pada saat sebelum melakukan autentikasi dan otorisasi dengan sesudah melakukan autentikasi dan otorisasi.
- Membandingkan *delay* proses autentikasi pada beberapa variasi jumlah proses autentikasi yang terjadi.
- Menganalisis proses *handoff* layer 2 yang terjadi
- Mengukur *latency* (delay waktu antara data dikirim hingga data dihadirkan pada user) proses *handoff* layer 2 yang terjadi.

## 1.3 Batasan Masalah

Untuk mempertegas pembahasan skripsi ini, diterapkan batasan-batasan masalah sebagai berikut.

- Pembahasan difokuskan pada proses autentikasi, otorisasi user, dan *handoff*.
- Pembahasan difokuskan pada mekanisme autentikasi *handoff* yang terjadi pada Radius dan mengukur besar *latency*-nya.
- Implementasi sistem ini dilakukan di jaringan UB dengan studi kasus di Jurusan Teknik Elektro.
- Digunakan *software* Freeradius sebagai *software* yang mengaplikasikan protocol Radius pada server Radius.
- Tidak digunakan mekanisme NAT (*Network Address Translation*) dalam pembagian IP untuk user *wireless* maupun *wired.*

## 1.4 Tujuan

Tujuan dilakukannya penelitian ini adalah untuk menerapkan sistem yang mampu menangani proses autentikasi dan otorisasi dengan protokol Radius dan untuk mengamati kualitas layanan pada jaringan yang telah diterapkan sistem AAA di dalamnya.

## 1.5 Manfaat

Pengimplementasian Radius sebagai protokol sistem otorisasi di sini akan memberikan manfaat yaitu meningkatkan sekuritas pada jaringan yang diimplementasikan sistem ini.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan ini akan menjelaskan tahapan-tahapan yang akan dilakukan dalam tugas akhir ini. Berikut adalah sistematika pembahasannya.

**Bab I Pendahuluan**

Pada bab ini akan diulas mengenai latar belakang yang mendasari penelitian ini, rumusan masalah yang akan dibahas, batasan masalah, tujuan dan manfaat penyusunan tugas akhir ini.

**Bab II Tinjauan Pustaka**

Pada bab tinjauan pustaka akan dibahas mengenai tinjauan pustaka yang digunakan sebagai acuan dalam penyusunan tugas akhir ini. Tinjauan pustaka tersebut meliputi :

1. Konsep dasar jaringan komputer
2. WiFi
3. Keamanan Jaringan
4. Radius

**Bab III Metode Penelitian**

Dalam bab ini akan dibahas mengenai metode penelitian yang diterapkan dalam penelitian ini. Antara lain adalah studi literatur, analisis kebutuhan dan perancangan sistem, implementasi, pengujian dan analisis hasil, dan yang terakhir adalah pengambilan kesimpulan.

**Bab IV Analisis Kebutuhan dan Perancangan Sistem**

Bab ini memuat perancangan untuk sistem yang akan dibangun, meliputi analisis dan perancangan sistem, sebagai dasar untuk melakukan implementasi.

**Bab V Implementasi dan Pengujian**

Bab ini memuat implementasi dan pengujian sistem yang telah dirancang. Pengujian yang dilakukan adalah mengenai hal-hal yang terdapat dalam rumusan masalah. Bab ini juga memuat analisis terhadap hasil pengujian tersebut.

**Bab VI Penutup**

Pada bab ini akan dibahas mengenai kesimpulan, yaitu hasil dari proses penelitian yang menjawab permasalahan-permasalahan yang terdapat dalam rumusan masalah. Selain itu juga akan dibahas saran dari peneliti sebagai rekomendasi untuk penelitian selanjutnya.

# BAB II

# TINJAUAN PUSTAKA

Bab ini menjelaskan tentang kajian pustaka dan dasar teori yang digunakan untuk menunjang penulisan skripsi ini. Kajian pustaka diperlukan untuk melakukan kajian terhadap karya ilmiah yang berkaitan dengan sksipsi ini. Kajian pustaka yang dilakukan adalah tentang penerapan protokol Radius sebagai protokol yang menerapkan sistem keamanan AAA pada jaringan komputer. Beberapa dasar teori yang diperlukan berdasar kajian pustaka adalah mengenai konsep dasar jaringan komputer, teori mengenai WiFi, keamanan jaringan, dan mengenai protokol Radius.

## 2.1 Penerapan Sistem Keamanan Jaringan Berbasis Protokol Radius

Dalam suatu jaringan *Local Area Network* (LAN) kampus yang memiliki banyak user diperlukan sistem pemantauan dan manajemen user. Hal ini bertujuan agar *administrator* jaringan dapat memantau dan mengontrol user pada jaringan [ERZ-05]. Sistem pemantauan dan manajemen user tersebut berupa sistem autentikasi user berbasis protokol Radius. Salah satu keuntungan penerapan sistem ini pada *Wireless Local Area Network* (WLAN) adalah kemudahan user untuk terkoneksi pada *access point* tanpa harus memasukkan *network security key* yang dimiliki oleh seorang user dan kemudahan *adimistrator* jaringan untuk melakukan aktifitas manajemen user [ERZ-05].

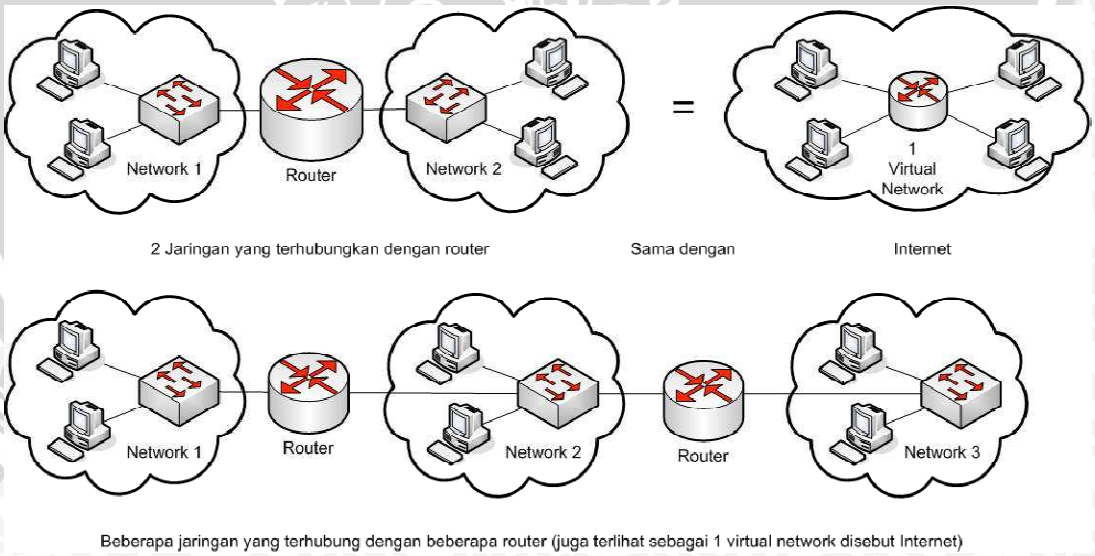
Infrastruktur jaringan komputer di Universitas Brawijaya (UB) telah dibangun sejak tahun 2002. Hingga sekarang pengembangan jaringan komputer di UB masih terus dilakukan. Di UB terdapat 55 buah router di bawahnya yang terkoneksi dengan internet. Router-router tersebut membagi jaringan dalam segmen-segmen untuk tiap-tiap fakultas maupun jurusannya. Jumlah user yang terkoneksi rata-rata sebesar 3000 user (dihitung dari IP address yang *up*) yang terkoneksi baik melalui kabel maupun nirkabel (*wireless).*

Namun dengan jumlah client rata-rata sebesar 3000 tersebut di UB masih belum terdapat sistem pencatatan user dan manajemen user. Sehingga identitas user yang terkoneksi tidak diketahui. Hal ini akan menyebabkan terganggunya keamanan jaringan di UB karena user bebas terkoneksi, baik user yang merupakan civitas UB maupun bukan.

Oleh karena itu dengan jaringan internal UB yang terus berkembang dan jumlah user yang terus bertambah, maka keberadaan server AAA di jaringan internal UB sebagai sistem manajemen user sangat penting. Sehingga judul yang diambil adalah "Perancangan dan Implementasi Sistem Autentikasi dan Otorisasi dengan Server AAA Berbasis Protokol Radius".

## 2.2 Konsep Dasar Jaringan Komputer

Jaringan komputer adalah himpunan sejumlah komputer *autonomous* yang saling terkoneksi. Dua buah komputer dikatakan saling terhubung bila keduanya dapat saling bertukar informasi. Kedua komputer yang tersambung dapat saling bertukar informasi dikarenakan keduanya menggunakan protokol yang sama.




Gambar 2.1. Router dan Internet
Sumber: [DHO-07]

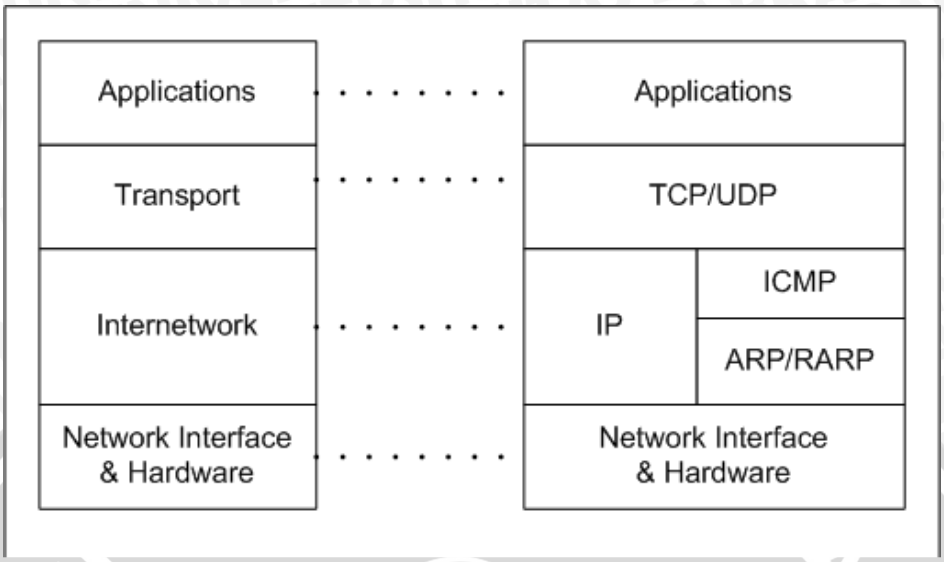
Jika terdapat dua jaringan yang dikoneksikan, maka perlu dibangun router diantara keduanya sebagai penjembatan antara kedua jaringan tersebut. Internet merupakan gabungan dari banyak jaringan yang saling terkoneksi [LAM-04].

Protokol adalah serangkaian aturan dan standar untuk saling berkomunikasi. Komunikasi tersebut adalah komunikasi yang terjadi antar komuter saat mengirimkan data dan seterusnya. Keduanya, pengirim dan penerima yang terlibat dalam pentransferan data harus mengenali dan mematuhi protokol yang sama [BLA-04].

Terdapat dua model referensi yang penting. Antara lain adalah model referensi OSI dan model referensi TCP/IP (Protokol TCP/IP) [LAM-04]. Berikut ini dijelaskan mengenai protokol TCP/IP sebagai acuan yang digunakan pada skripsi ini.

### 2.2.1 Protokol TCP/IP

Jaringan TCP/IP adalah fondasi teknis dari Internet dan LAN paling modern. Jaringan ini membentuk mekanisme komunikasi antar komputer dan untuk memindahkan informasi. TCP (*Transmission Control Protocol*) dan IP (*Internet Protocol*) adalah dua protokol yang penting yang digunkan di Internet [MAN-04]. Pada protokol TCP/IP terdapat layer-layer yang pada prinsipnya sama dengan OSI Layer, namun disini dipersimpel menjadi 4 layer.




Gambar 2.2. Ilustrasi Protokol TCP/IP
Sumber: [DHO-04]

Penjelasan detil tentang tiap layernya akan dibahas pada sub bab berikut ini. Pada penjelasan di bawah ini akan lebih diperdalam mengenai layer fisik, karena perlu dijelaskan mengenai *wireless* lebih dalam.

#### 2.2.1.1 Layer Network Interface dan Hardware

Layer ini biasa disebut dengan Layer Datalink. Layer ini bertanggung jawab untuk meletakkan frame-frame jaringan di atas media jaringan yang digunakan. TCP/IP dapat bekerja dengan banyak teknologi transport, mulai dari teknologi transport dalam LAN (seperti halnya Ethernet dan Token Ring), MAN dan WAN (seperti halnya *dial-up* modem yang berjalan di atas *Public Switched Telephone Network* (PSTN), *Integrated Services Digital Network* (ISDN), serta *Asynchronous Transfer Mode* (ATM) [WIK-08]. Terdapat beberapa jenis layer fisik. Antara lain kabel dan nirkabel (wireless). Mengenai jaringan dengan medium wireless akan dibahas kemudian.

#### 2.2.1.2 Layer Network

Protokol lapisan internetwork: bertanggung jawab untuk melakukan pemetaan (routing) dan enkapsulasi paket-paket data jaringan menjadi paket-paket IP.

Protokol yang bekerja dalam lapisan ini adalah *Internet Protocol* (IP), *Address Resolution Protocol* (ARP), *Internet Control Message Protocol* (ICMP), dan *Internet Group Management Protocol* (IGMP) [WIK-08].

#### 2.2.1.3 Layer Transport

Layer ini berguna untuk membuat komunikasi menggunakan sesi koneksi yang bersifat connection-oriented atau broadcast yang bersifat connectionless. Protokol dalam lapisan ini adalah *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) [WIK-08].

#### 2.2.1.4 Layer Aplikasi

Layer ini bertanggung jawab untuk menyediakan akses kepada aplikasi terhadap layanan jaringan TCP/IP. Protokol ini mencakup protokol *Dynamic Host Configuration Protocol* (DHCP), *Domain Name System* (DNS), *Hypertext Transfer Protocol* (HTTP), *File Transfer Protocol* (FTP), Telnet, *Simple Mail Transfer Protocol* (SMTP), *Simple Network Management Protocol* (SNMP), dan lain-lain [WIK-06].

### 2.2.2 Quality of Service (QoS)

QoS adalah parameter pengukuran untuk ketersediaan *service* sistem dan kualitas transmisi. Ketersediaan *service* adalah hal yang sangat penting pada elemen QoS. Sebelum QoS dapat diimplementasikan dengan baik, infrastruktur jaringan harus didesain untuk ketersediaan yang tinggi. Sedangkan kualitas transmisi ditentukan oleh beberapa faktor berikut ini [SZI-05].

- *Delay*

  *Delay* adalah jumlah waktu yang diperlukan sebuah paket untuk mencapai *endpoint* penerima setelah ditransmisikan dari *endpoint* pengirim. Satuan d*elay* adalah detik (s).

- *Transfer Rate* Data

  *Transfer rate* data adalah banyaknya data yang diterima pada tiap satuan waktu. *Transfer rate* data ini biasanya berbentuk satuan bit per detik (bps).

- *Latency*

  *Latency* adalah *delay* waktu ketika stream data dikirim dan ketika stream itu dihadirkan pada end user. *Latency* dapat disebut juga sebagai *round-trip time*. Satuan *latency* adalah detik (s). *Latency* merupakan masalah utama dalam WiFi *handoff*. Penerapan *handoff* pada jaringan WiFi mengakibatkan *latency* yang besar sehingga pengaplikasian *mobile node* sulit dilakukan. Namun usaha untuk mengurangi *latency* pada WiFi *handoff* masih terus dilakukan [SZI-05].

## 2.3 Wireless Fidelity (WiFi)

Jaringan *wireless* adalah pelengkap sempurna bagi jaringan kabel, tapi jaringan wireless tetaplah bukan pengganti jaringan kabel. Seiring telepon genggam menjadi pelengkap telepon kabel, wireless LAN juga menyediakan kelebihan mobilitas pada penggunanya. Berikut ini adalah hal-hal yang membuat jaringan wireless berbeda :

- Kurangnya batasan fisik. Hal ini dapat menjadi kelemahan jaringan wireless. Karena saat jaringan tradisional dimana datanya melintas pada jalur tertentu (kabel). Sedangkan pada jaringan wireless, sinyal di dikirimkan dan diterima oleh setiap orang yang memiliki penerima gelombang radio. Inilah yang membuat jaringan wireless tidak aman [GAS-05].
- Media fisik yang dinamik. Saat jaringan kabel ditempatkan pada suatu tempat yang permanen, media fisik jaringan *wireless* lebih dinamis. Gelombang radio memantulkan pada objek dan menembus tembok [GAS-05].

- Keamanan. Beberapa jaringan *wireless* berbasis pada gelombang radio, dimana gelombang radio ini pada dasarnya terbuka bagi adanya interupsi. 802.11 tidak membantu banyak hal terkait dengan protokol keamanan. Jaringan *wireless* harus disertai dengan autentikasi untuk mencegah user yang tidak berhak dan koneksi yang terautentikasi harus terenkripsi untuk menghindari gangguan trafik oleh pihak yang tidak berhak [GAS-05].

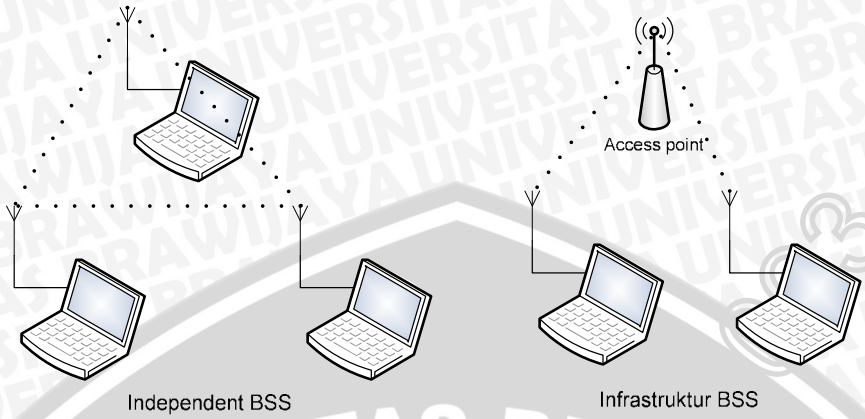
IEEE 802.11 adalah sekumpulan standar untuk komunikasi *wireless* LAN (WiFi) pada komputer. 802.11 adalah anggota dari keluarga IEEE 802. Keluarga IEEE 802 adalah beberapa seri spesifikasi untuk teknologi LAN. IEEE 802 fokus pada dua layer terbawah, yaitu fisik dan datalink. 802.11 didesain untuk menjadi sama seperti link layer-link layer yang lain bagi layer di atasnya. Administrator jaringan yang akrab dengan Ethernet (802.2) akan merasa nyaman dengan 802.11. Kadang 802.11 diartikan sebagai "*wireless* Ethernet".

### 2.3.1 Beberapa Jenis Infrastruktur WiFi

Dasar pondasi jaringan 802.11 adalah *Basic Service Set* (BSS), yang berarti kumpulan perangkat yang berkomunikasi satu sama lain. Komunikasi tersebut terjadi di suatu tempat yang disebut dengan basic service area, yang ditandai olek karakteristik propagasi pada medium wireless [GAS-05]. Terdapat beberapa macam jaringan *wireless* 802.11. Antara lain:

- Jaringan wireless independen

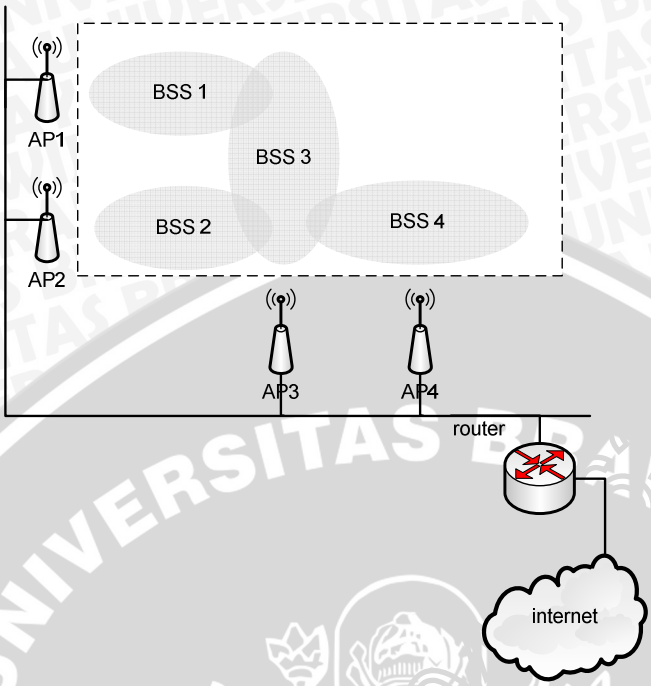
Atau biasa disebut dengan IBSS (*Independent* BSS). Pada jenis ini, setiap perangkat yang berkomunikasi secara IBSS satu sama lain secara langsung. Minimal dua perangkat yang terkoneksi langsung dapat disebut IBSS. Komunikasi IBSS ini sering disebut dengan jaringan ad hoc. Gambar 4 sebelah kiri menunjukkan beberapa laptop yang terkoneksi secara ad hoc [GAS-05].




Gambar 2.3. Independen dan infrastruktur BSS
Sumber: [GAS-05]

- Jaringan wireless infrastruktur

Gambar 2.3 sebelah kanan menunjukkan contoh jaringan wirekess infrastruktur. Jenis ini dibedakan dengan adanya access point di dalamnya. Jika sebuah perangkat ingin berkomunikasi dengan perangkat lain yang sama-sama terkoneksi dengan access point tersebut, maka dua hop yang harus dilalui. Yaitu mengiriman data ke access point dari perangkat pertama, lalu ke perangkat yang kedua, perangkat tujuan [GAS-05].




Gambar 2.4. Extended service area
Sumber: [GAS-05]

- Extended service area

BSS dapat mencangkup sebuah kantor yang kecil atau rumah. Namun tidak bisa mencangkup area yang lebih luas. Tipe jaringan wireless yang satu ini dapat menjadi solusi dengan menyambungkan beberapa BSS pada suatu extended service set (ESS). Hal ini dapat dilakukan dengan menyamakan SSID (Service Set Identifier), dimana user biasa menggunakan sebagai nama jaringan [GAS-05].

### 2.3.2 Handoff

Pengguna jaringan wireless berkembang, sehingga pengguna wireless tidak hanya host yang menetap, tapi juga host yang *mobile*. Untuk memenuhi kebutuhan ini, terdapat mekanisme *handoff*, yaitu saat sebuah *Mobile Node* (MN) berpindah layanan dari satu *Basic Service Set* (BSS) ke BSS yang lain karena, misal, perpindahan lokasi. Peristiwa *handoff* ini sangat memungkinkan terjadi pada sebuah MN yang terkoneksi pada sebuah jaringan wireless, dalam hal ini WLAN (802.11) yang terbatas jangkauan spektrumnya [ZHA-08].

#### 2.3.2.1 Tahapan-tahapan *handoff*

Terdapat beberapa tahap terjadi saat sebuah MN melakukan proses *handoff*.

- *Scanning*

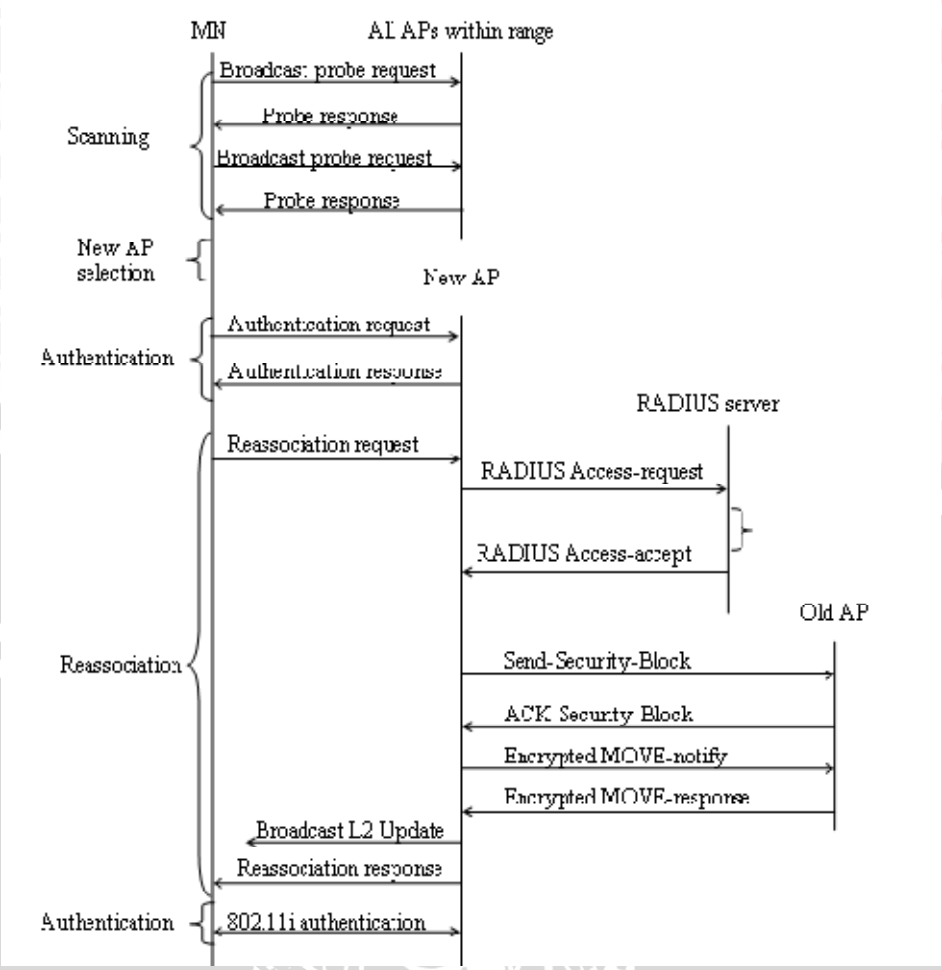
  Saat pada tahap scanning, MN mencari dan menentukan BSS yang terdapat pada jangkauannya sebagai BSS tujuan.

- *Authentication*

  Setelah memilih BSS tujuan, MN melakukan autentikasi ke AP baru. Disini AP berusaha mencari tahu apakah MN yang bersangkutan berhak terkoneksi pada AP tersebut. Dalam hal ini menyangkut mengenai *security key* seperti WEP, WAP, dan sebagainya.

- *Reassociation*

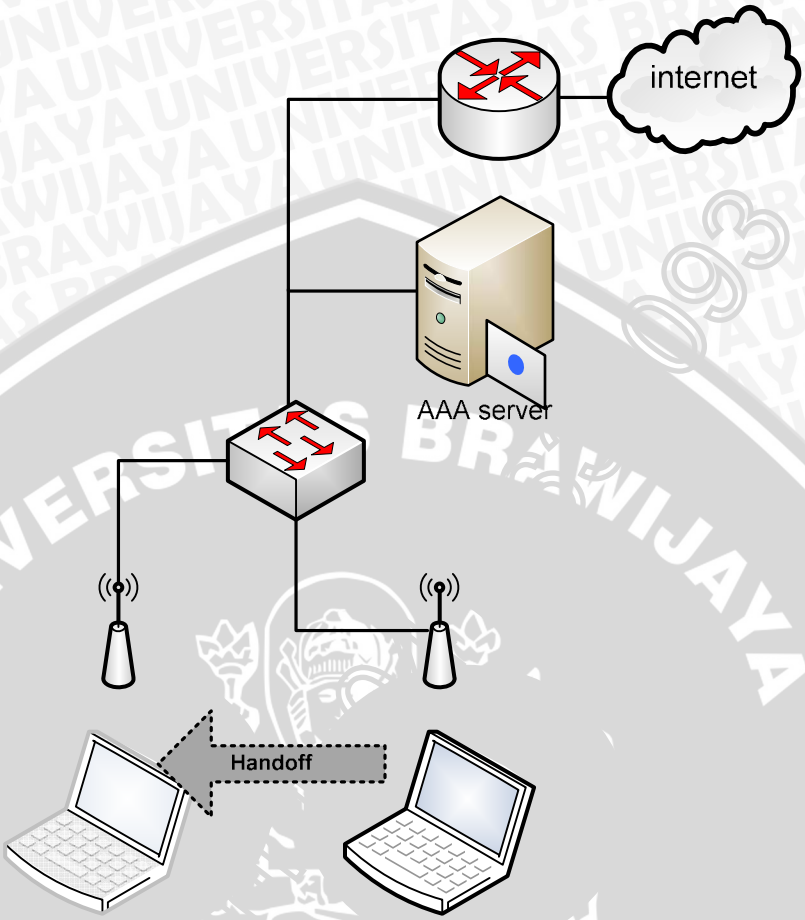
  Dalam tahap ini, MN melakukan autentikasi pada AAA server, dalam hal ini server Radius [ZHA-08].




Gambar 2.5. Proses *handoff* dan diagram aliran *message*
Sumber: [ZHA-08]

### 2.3.2.2 *Handoff* layer 2

*Handoff* yang dimaksud dalam skripsi ini adalah *handoff* layer 2. Yang dimaksud *handoff* layer 2 adalah peristiwa *handoff* yang terjadi pada AP-AP yang masih berada pada satu *Extended Service Set* (ESS). Sehingga perpindahan koneksi hanya terjadi pada layer 2 (MAC). Berikut ini adalah gambar yang mengilustrasikan peristiwa *handoff* layer 2.




Gambar 2.6. *Handoff* Layer 2
Sumber: [MAR-07]

## 2.4 Keamanan Jaringan

Telah banyak didengar mengenai kejahatan di dunia komputer, khususnya jaringan internet. Virus, Trojan, Web *deface*, sampai dengan masalah pencurian kartu kredit. Hal ini merupakan tindakan yang sangat merugikan. Buruknya keamanan jaringan sebuah sistem akan mengakibatkan kerugiann materiil dan memburuknya reputasi [STI-05].

Terdapat beberapa aspek keamanan jaringan komputer antara lain:

- *Privacy*/*Confidentality*

  *Privacy* atau *confidentality* adalah usaha untuk menjaga informasi dari orang

yang tidak berhak mengakses. *Privacy* lebih ke arah data-data yang bersifat pribadi, sedangkan *confidentality* berhubungan dengan datan yang diberikat kepada pihak lain untuk keperluan tertentu.

- *Integrity*

  Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa ijin dari pemilik informasi. Informasi harus sesuai dan sama persis dengan saat informasi dikirimkan.

- Authentication

  Aspek ini berhubungan dengan metode atau cara untuk menyatakan bahwa informasi benar-benar asli, pihak yang mengakses atau memberikan informasi adalah benar-benar orang yang dimaksud. Metode yang paling sering digunakan misalnya metode *password*.

- Availability

  *Availability* adalah ketersediaan data dan informasi. Data dan informasi yang berada dalam suatu sistem tersedia dan dapat dimanfaatkan oleh pihak yang berhak saat dibutuhkan.

- *Access Control*

  Aspek ini berhubungan dengan cara pengaturan akses pada informasi. Hal ini biasanya berhubungan dengan klarifikasi data (*public*, *private*, *top secret*) dan *user* (*guest*, *admin*), mekanisme autentikasi dan juga *privacy*. Contoh sederhana penerapan *access control* adalah mekanisme username dan *password*. Dengan demikian user akan dibatasi otoritasnya sesuai dengan kebutuhannya dan haknya [STI-05].

### 2.4.1 AAA (Authentication, Authorization, Accounting)

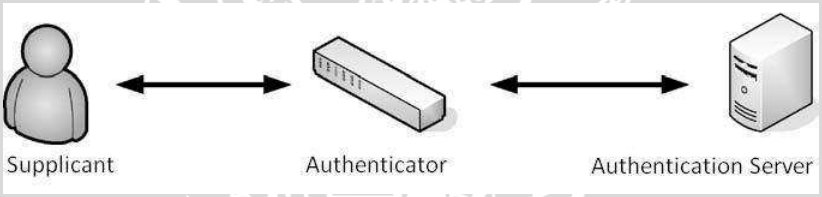
Sistem keamanan ini pada intinya menangani proses autentikasi dari banyak user dan membandingkannya dengan database pada server. Server ini dapat menghubungkan ke sebuah switch suatu jaringan yang akan mengatur akses masuk ke jaringan tersebut [STI-05:105].

#### 2.4.1.1 Autentikasi (Authentication)

Aspek ini berhubungan dengan metode atau cara untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, server yang kita hubungi adalah server yang asli. Biasanya metode ini dikenal dengan permintaan username dan password pada layanannya [STI-05:48]. Beberapa protokol autentikasi antara lain PAP (*Password Autenthication Protocol*), CHAP (*Challenge Handshake Authentication Protocol*), dan EAP (*Extensible Authentication Protocol*) dijelaskan kemudian.

802.1X adalah sekumpulan protokol-protokol untuk mensupport metodologi untuk menjawab pertanyaan: Apakah ini adalah benar-benar user atau perangkat yang diperbolehkan terkoneksi ke jaringan? 802.1 bekerja di layer datalink (layer 2).

Terdapat 3 pihak yang terlibat pada 802.1X. Yang pertama adalah supplicant, *authenticator*, dan authentication server. Gambar 6 menunjukkan komunikasi antara ketiganya secara logikal [BRO-07].




Gambar 2.7. Dasar komunikasi 802.1X
Sumber: [BRO-07]

Saat *supplicant* (dapat berupa perangkat wireless, misal laptop) meminta akses ke LAN *resource, Authenicator* (dapat berupa access point) meminta identitas supplicant. Pada saat ini hanya paket-paket autentikasi yang diperbolehkan lewat (*authenticator* menutup port). Setelah identitas dikirim, proses autentikasi dimulai.

Selama proses autentikasi, *authenticator* menyampaikan paket antara *supplicant* dan server. Ketika proses autentikasi selesai, server mengirim pesan sukses atau gagal. Jika sukses, kemudian *authenticator* membuka port untuk *supplicant* [BRO-07].

#### 2.4.1.2 Otorisasi (Authorization)

Otorisasi dapat berarti memperbolehkan beberapa tipe hak akses (privilege) pada sebuah entitas atau user berdasarkan pada autentikasi mereka, privilege apa yang mereka gunakan dan kondisi sistem saat itu. Kebanyakan memperbolehkan hak akses itu berupa memberi batasan memberikan keleluasaan untuk menggunakan suatu servis [WIK-08].

#### 2.4.1.3 Accounting

Accounting dapat diartikan sebagai melacak konsumsi penggunaan layanan/*resource*. Accounting ini dapat dimanfaatkan sebagai billing, planning, dan lain-lain [WIK-08].

## 2.5 Radius (Remote Access Dial-In User Service)

Protokol Radius merupakan protocol AAA yang banyak digunakan di pasaran. Hal ini dikarenakan Radius adalah protocol yang mudah diterapkan, *vendor independent*, dan sederhana.

Protokol Radius dapat menggunakan mekanisme *client-server*, dimana NAS (*Network Access Server*) bertindak sebagai *client* Radius. Dalam hal ini perlu dibedakan antara *client* Radius dengan *end client* dalam skenario komunikasi. Dalam konteks Radius, *client* adalah pihak yang berperan sebagai *client* dalam pengiriman

pesan Radius. Sedangkan *end client* adalah perangkat yang melakukan autentikasi ke jaringan melalui NAS. *End client* ini dalam konteks arsitekrur sistem AAA disebut sebagai *supplicant* [NAK-05].

NAS atau *client* Radius yang digunakan pada penelitian ini berupa *captive portal*. *Captive portal* adalah *software* yang menyediakan fasilitas antarmuka berupa web untuk mendukung *end client* atau user untuk mengisikan *username* dan *password* pada web tersebut. Untuk keamanan data antara *end client* dan NAS *captive portal* menyediakan fasilitas HTTPS (*Hypertext Transfer Protocol Secure*) sehingga username dan password yang dikirim dari end client ke NAS menggunakan koneksi *Transport Layer Security* (TLS).

Selama proses autentikasi, *client* Radius bertanggungjawab meneruskan informasi user dalam bentuk paket Access-Request pada server Radius dan menunggu respon dari server. Respon yang terjadi dapat berupa Access-Accept yang berarti autentikasi berhasil, maupun Access-Reject yang berarti autentikasi gagal. Jika paket yang diterima adalah Access-Accept, NAS akan membuka port-portnya sehingga end client dapat menggunakan *resource* jaringan. Sedangkan server Radius bertanggungjawab sebagai pemroses request, mengauntentikasi user, dan mengembalikan informasi yang diperlukan *client* untuk memberikan layanan pada user [NAK-05].

Berikut ini adalah beberapa ciri protokol Radius:

- Menggunakan UDP untuk protokol transportnya
  UDP dianggap lebih cocok untuk digunakan oleh Radius. Hal ini dikarenakan TCP dianggap membutuhkan waktu terlalu lama untuk menyelenggarakan session.
- Menggunakan MD5 *hash function* sebagai *attribute*/*password hiding*.
- Menggunakan *shared secret* (radius secret) untuk keamanan berkomunikasi antara NAS dan server.
- Mendukung autentikasi CHAP dan PAP

PAP (*Password Authentication Protocol*) dan CHAP (*Challenge Handshake Authentication Protocol*) adalah protokol dasar untuk autentikasi yang digunakan oleh Radius. Namun kedua protokol ini tidaklah mutlak yang bias digunakan oleh Radius. Radius juga mendukung protokol lain yang tergantung vendor. Nanti akan dijelaskan lebih lanjut mengenai kedua protokol autentikasi ini.

- Menyediakan lebih dari 50 AVP (*Attribute-Value Pairs*) dengan kemampuan untuk membuat atribut sesuai kebutuhan.

### 2.5.1 Beberapa Tipe Paket Radius

Berikut ini adalah tabel yang menunjukkan beberapa tipe paket Radius yang berkaitan dengan autentikasi beserta penjelasannya.

Tabel 2.1. Beberapa tipe paket Radius

| Nama Pesan | Deskripsi |
|---|---|
| Access-Request | Pesan ini dihasilkan oleh NAS (*client* Radius) yang ditujukan kepada server untuk meneruskan permintaan user. |
| Access-Challenge | Pesan ini dikirim dari server Radius pada NAS dan umumnya digunakan untuk menanyakan tentang sesuatu atau melakukan negosiasi. |
| Access-Accept | Pesan ini dikirim dari server Radius menuju NAS untuk mengabarkan *request* yang dikabulkan. |
| Access-Reject | Pesan ini dihasilkan oleh server Radius untuk menolak suatu *request*. |

Sumber: [NAK-05]

### 2.5.2 Format Paket Radius

Radius berkomunikasi dengan port 1812, dan 1813 untuk port *accounting*. Berikut ini adalah penggambaran format paket Radius.


Gambar 2.8. Format Paket Radius
Sumber: [NAK-05]

Berikut ini penjelasan untuk header paketnya.

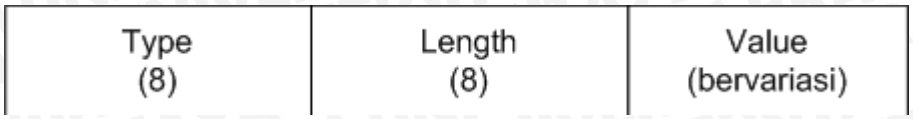
Tabel 2.2. Bagian-bagian pada header paket Radius

| **Nama sub-field** | **Penjelasan** |
|---|---|
| Code | Panjangnya 1 oktet dan bertujuan untuk menentukan tipe paket Radius. Code bernilai 1 untuk Access-Request, 2 untuk Access-Accept, 3 untuk Access-Reject, dan sebagainya. |
| ID | Panjangnya 1 oktet, bertujuan untuk mencocokkan *request* dan *response* nya. |
| Length | Panjangnya 2 oktet, bertujuan untuk menunjukkan panjang seluruh message. |
| *Authenticator* | Panjangnya 16 oktet, berisi hasil penghitungan MD5 *hash*, dan terdiri dari dua jenis: *request authenticator* dan *response authenticator*. |
| *Attributes* dan *values* | Panjangnya bervariasi tergantung pada parameter yang dikenakan pada user tertentu. Penjelasan lebih detail akan dipaparkan pada sub bab tersendiri. |

Sumber: [NAK-05]

### 2.5.3 Atribut Radius

Atribut Radius membawa informasi autentikasi dan otorisasi. Format atribut pada paket data dijelaskan pada gambar berikut. Atribut Radius disertakan pada informasi user yang tersimpan baik pada basis data maupun *file*.




Gambar 2.9 Format Atribut Radius pada paket data
Sumber: [NAK-05]

*Field Type* sepanjang satu oktet berisi kode angka yang mengindikasikan suatu atribut tertentu. *Field Lenght* sepanjang satu oktet berisi informasi tentang panjang atribut tersebut yang di dalamnya sudah termasuk *Type*, *Lenght*, dan *Value*. *Value* berisi informasi tentang atribut terkait. Panjang *value* bervariasi.

Berikut ini adalah sebagian contoh dari atribut Radius beserta kode *Type*nya.

Tabel 2.3 Tabel Atribut Radius

| *Type* | Atribut | Kegunaan |
|---|---|---|
| 1 | User-Name | Menyimpan nama *user* |
| 2 | User-Password | Menyimpan password *user* |
| 3 | CHAP-Password | Menyimpan password CHAP |
| 4 | NAS-IP-Address | Alamat IP NAS |
| 5 | NAS-Port | Nomor port NAS |
| 6 | Service-Type | Servis yang digunakan *user* untuk login |
| 27 | Session-Timeout | Batas waktu sesi |
| 28 | Idle-Timeout | Batas waktu sesi dalam keadaan tidak ada aktifitas |
| 60 | CHAP-Challenge | Berisi CHAP challenge yang diberikan oleh NAS pada autentikasi CHAP |

Sumber: [NAK-05]

### 2.5.4 Beberapa Teknologi Autentikasi

Untuk dapat melakukan komunikasi antara *client* dan *server* untuk pertama kali, diterapkan sebuah protokol autentikasi. Berikut ini dijelaskan mengenai beberapa teknologi autentikasi yang disupport oleh Radius.

#### 2.5.4.1 PAP (Password Authentication Protocol)

PAP adalah protokol autentikasi dengan menggunakan username dan password. Pada awalnya PAP digunakan pada PPP (*Point to Point Protocol*). Mekanisme protokol ini adalah sebagai berikut.

- Ketika user mencoba membuat koneksi PPP dengan NAS, NAS akan meminta username dan password.
- Saat menerima username dan password dari user, NAS membuat sebuah pesan *Access Request* untuk server sebagai berikut: NAS membuat sebuah *Request Authenticator* (RA) dan menggunakan RA dan *shared secret* yang di*share* dengan Radius server untuk menyembunyikan password user dan menghasilkan *User-Password Attribute*. NAS kemudian menunggu respon dari server. NAS menyertakan RA pada bagian *Authenticator* pada pesan *Access Request*. Daftar atribut untuk pesan ini ditampilkan pada tabel 1.

Tabel 2.4. Daftar atribut pada pesan *Access-Request* untuk PAP

| **Nama Atribut** | **Deskripsi** |
|---|---|
| User name | PAP ID untuk user |
| User password | MD5 hash dari *shared secret*, RA, dan password user. |
| NAS IP | Alamat IP untuk NAS (atribut ini bisa digantikan oleh NAS ID) |
| NAS Port | Port dimana user terkoneksi pada NAS, bukan port UDP |

Sumber: [NAK-05]

- Jika NAS tidak menerima respon pesan *Access Request* dari server selama tenggang waktu yang terspesifikasikan, NAS dapat mengirim ulang *request* pada server yang sama, atau mencoba mengirimkannya pada server alternatif (mungkin pada model *round robin*). Namun hal ini tidak distandarkan.
- Selama menerima pesan *Access Request*, server Radius melakukan pengecekan atas kelengkapan pesan *Access Request*, salah satunya mengecek apakah server memiliki *shared secret* dengan NAS. Kemudian mencocokan username dan password yang diterima dengan yang ada di database.
- Jika terjadi kecocokan username dan password, server Radius mengirimkan pesan *Access Accept* pada NAS.
- Jika jika beberapa kondisi tidak dipenuhi seperti: user tidak tersedia di database server, penghitungan password *hash* tidak berhasil, maka server menolak *request* dengan mengirimkan pesan *Access Reject* pada NAS [NAK-05].

#### 2.5.4.2 CHAP (Challenge Handshake Authentication Protocol)

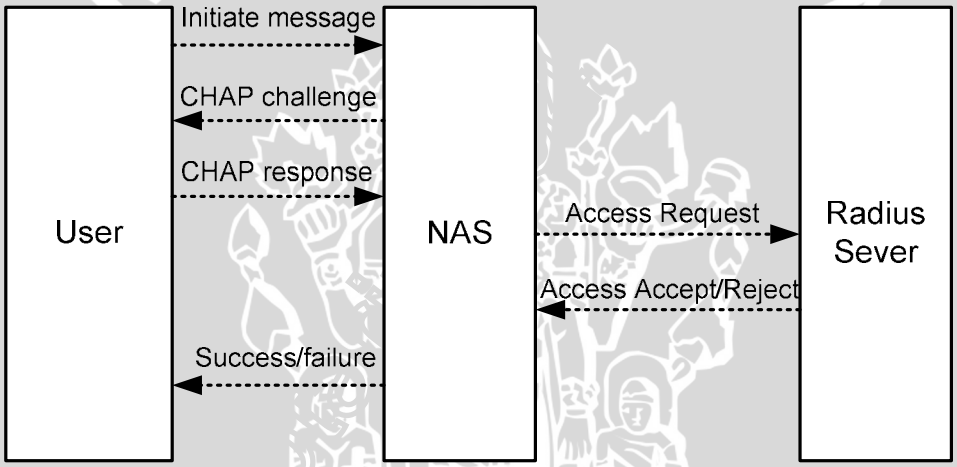
CHAP (*Challenge Handshake Authentication Protocol*) adalah salah satu dari protokol autentikasi dasar yang disupport oleh Radius. Berikut ini adalah mekanismenya:

- User yang telah dikonfigurasi untuk melakukan autentikasi dengan CHAP melakukan *request* kepada NAS untuk dapat terkoneksi pada jaringan.
- NAS membuat *challenge* 16 oktet dan mengirimkannya pada user melalui pesan CHAP *challenge*. Dalam pesan ini terdapat CHAP ID di dalamnya.
- User merespon *challenge* mengunakan pesan respon yang didalamnya termasuk CHAP ID yang sama dengan yang digunakan pada pesan *challenge*, username CHAP, dan responnya atas *challenge* tersebut. Respon ini kemudian dihitung dengan menggunakan MD5 *hash*, dengan *secret* user sebagai key:

  MD5 (ID, secret, challenge)
- NAS kemudian membuat sebuah *Access Request*. NAS memasukkan

username CHAP dari user pada atribut *User-Name*, serta CHAP ID dan CHAP password pada atribut *User-Password* dan mengirim atribut-atribut ini bersama dengan *Access Request* kepada server Radius.

- Server Radius kemudian mengecek password pada *User-Name* dan menghitung sebuah *hash* dengan cara yang sama seperti yang dilakukan user. Jika atribut *CHAP-Challenge* tidak tersedia pada pesan, server menggunakan nilai pada bagian *Request Authenticator* pada pesan *Access Request*. Server membandingkan hasil dari *hash* dengan nilai yang terdapat pada *CHAP pasword*. Jika cocok, server mengirim pesan *Access Accept* kepada NAS. Jika tidak, server mengirimkan pesan *Access Reject* [NAK-05].




Gambar 2.10. Radius Messaging untuk autentikasi dengan protokol CHAP
Sumber: [NAK-05]

Berikut ini adalah tabel atribut yang terdapat pada pesan *Access Request* untuk protokol CHAP.

Tabel 2.5. Beberapa atribut yang terdapat pada pesan *access-request* pada protokol CHAP

| **Nama atribut** | **Deskripsi** |
|---|---|
| User name CHAP | User name |
| CHAP password | Atribut ini seperti pada TLV (*type lenght value form*), nilai di dalamnya termasuk CHAP ID dan respon CHAP dari user kepada *challenge* yang hasilkan oleh NAS dan pada pesan *CHAP response* kepada NAS. |
| CHAP challenge | *CHAP challenge* dimana NAS memberikan kepada user (jika nilainya lebih pendek dari 16 oktet, dapat ditempatkan pada bagian *request authenticator*). |
| NAS IP | Alamat IP NAS. |
| NAS port | Port dimana user terkoneksi, bukan port UDP. |

Sumber: [NAK-05]

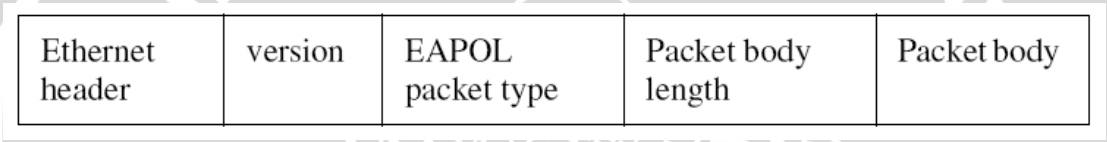
#### 2.5.4.3 Extensible Authentication Protocol (EAP)

Pada dasarnya, EAP hanya menyelenggarakan proses negosisasi antara user dan server Radius (dengan NAS yang berada di tengah-tengahnya). EAP lebih tepatnya merupakan *framework* autentikasi, bukan merupakan protokol autentikasi. EAP pada dasarnya bekerja pada layer data link seperti PPP, tanpa membutuhkan IP. EAP bertugas menyalurkan pertukaran pesan antara NAS dan server.

Keuntungan utama EAP adalah keleluasaannya untuk menyelenggarakan model autentikasi *three-party* pada berbagai infrastruktur. Inilah sebabnya protokol ini dinamakan *extensible*. Seperti yang telah disebutkan, EAP awalnya didesain untuk PPP. Namun telah banyak pengembangan dari protokol ini. Salah satunya EAPoL (EAP over LAN).

EAP pada awalnya didesain untuk mendukung akses jaringan dan mekanisme autentikasi pada environment yang tidak tersedia IP messaging. Karena alasan ini makan EAP didesain secara spesifik untuk bekerja pada layer datalink, seperti tipe link PPP dan Ethernet. Untuk mendukung model autentikasi tiga pihak yang terdapat NAS dan server AAA, dukungan untuk pensinyalan EAP telah ditambahkan pada protokol RADIUS dan Diameter [NAK-05].

EAP *over* LAN (EAPoL) didesain untuk IEEE 802.1X dan spesifikasinya untuk menyediakan jalan untuk menempatkan EAP pada protokol LAN. Karena IEEE 802 LAN menyediakan layanan framing, konsep protokol EAPoL sangat simpel : mengenkapsulasi pesan EAP didalam frame EAPoL [NAK-05].



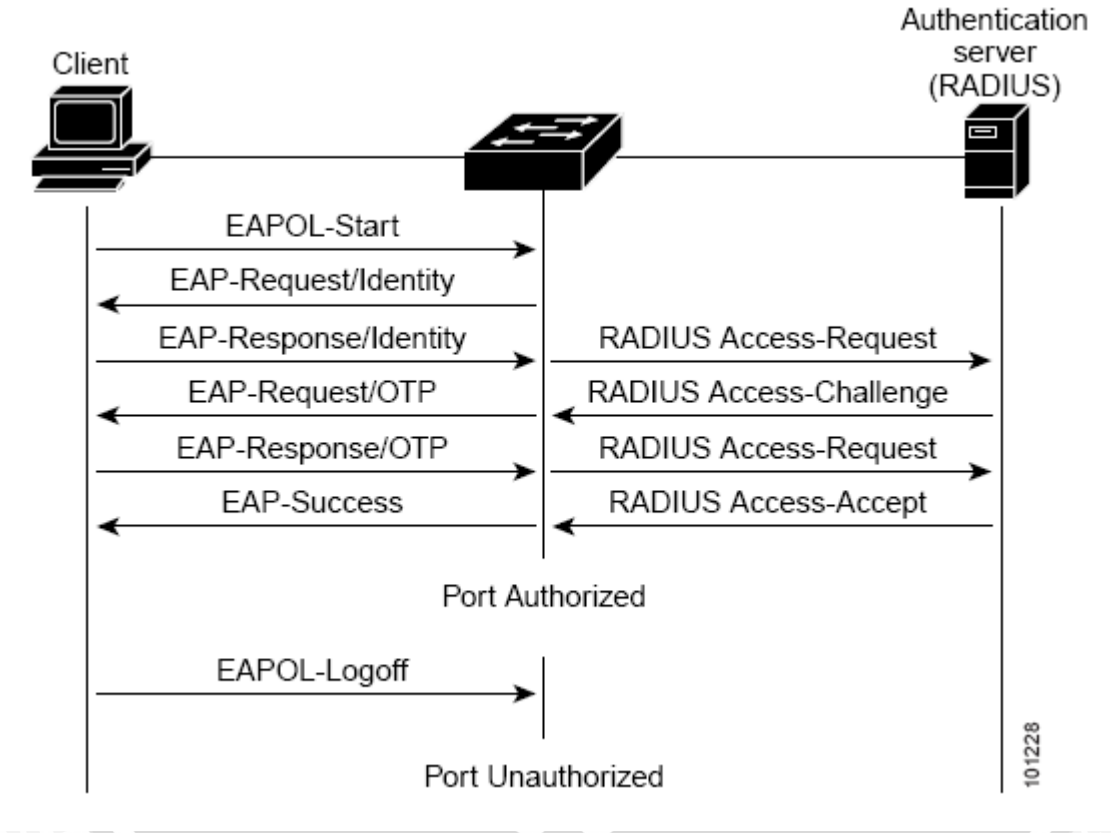

Gambar 2.11. Format paket EAPoL
Sumber: [NAK-05]

Meski dapat diwakili dengan satu tipe frame dengan nama EAPoL, pada 802.1X ditambahkan tipe pesan EAPoL untuk melaksanakan beberapa tugas selain hanya mengenkapsulasi pesan EAP yang dapat dihasilkan dimanapun. Pesan-pesan EAPoL tersebut adalah [NAK-05]:

- EAPOL-Start
- EAPOL-Key
- EAPOL-Packet
- EAPOL-Logoff
- EAPOL-Encapsulated-ASF-Alert

Peranan NAS (*authenticator*) dalam mensinyalan sangat simpel. Secara umum, NAS melanjutkan *request* dari server pada user dan *response* dari user ke server tanpa mengerti banyak mengenai pesan EAP. Namun NAS didesain untuk mengerti pesan *success* atau *failure* dengan cara menunggu pesan tersebut untuk mengetahui bahwa autentikasi telah selesai dilakukan atau tidak.

EAP dapat bekerja langsung pada layer datalink tanpa memerlukan protokol layer network. Peristiwa ini terjadi pada saat komunikasi antara user dan *authenticator*. Dalam hal ini *authenticator* yang terletak pada sub jaringan melanjutkan paket pada server AAA. Pada komunikasi antara *authenticator* dan AAA server, pesan EAP terenkapsulasi pada paket pesan AAA. Gambar di bawah ini menunjukkan model perjalanan pesan-pesan EAPOL pada jalur dari user menuju server AAA dan sebaliknya (model autentikasi *three party*).




Gambar 2.12. EAP pada model autentikasi *three party*
Sumber: [NAK-05]

# BAB III

# METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai berbagai metode yang digunakan untuk menjawab rumusan masalah yang terdapat pada bab I. Hal-hal yang dijelaskan pada bab metode penelitian ini antara lain adalah studi literatur, analisis kebutuhan dan perancangan sistem, implementasi dan pengujian, dan yang terakhir adalah pengambilan kesimpulan.

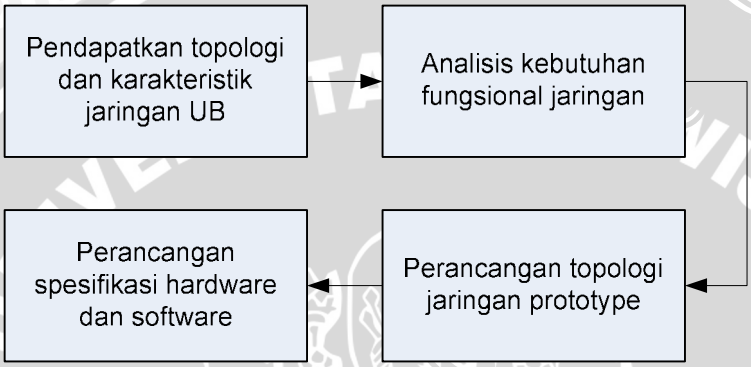
## 3.1 Studi Literatur

Studi literatur yang dilakukan berupa pencarian literatur baik berupa *textbook, whitepaper*, jurnal, *Request For Comment* (RFC), maupun penelitian-penelitian yang pernah dilakukan, dan dokumentasi-dokumentasi *project* yang dipublikasikan di situs web tentang Radius, WiFi *handoff, captive portal* dan *wireless router access point* dan pensettingannya.

## 3.2 Analisis Kebutuhan dan Perancangan Sistem

Pada tahap awal analisis dilakukan dengan mendapatkan topologi jaringan dan karakteristik jaringan yang ada di Universitas Brawijaya (UB). Dengan demikian akan diketahui apa-apa yang dibutuhkan, menyesuaikan dengan kondisi jaringan yang ada. Setelah mendapatkan topologi tersebut, dilakukan analisis kebutuhan fungsional jaringan. Dalam hasil analisis ini nantinya akan terdapat kebutuhan-kebutuhan pembagian otoritas dari beberapa tipetipe user.

Tahap selanjutnya adalah perancangan sistem. Perancangan dilakukan dengan menggunakan hasil analisis sebagai dasar untuk menentukan posisi server dan topologi yang memungkinkan pada jaringan *prototype*. Perancangan juga dilakukan dengan menentukan spesifikasi alat yang akan digunakan beserta *tool-tool* aplikasi yang akan digunakan. Tahap ini juga menjelaskan tentang rancangan otoritas beberapa tipe *user* yang dibutuhkan.




Gambar 3.1. Alur kerja analisis dan perancangan

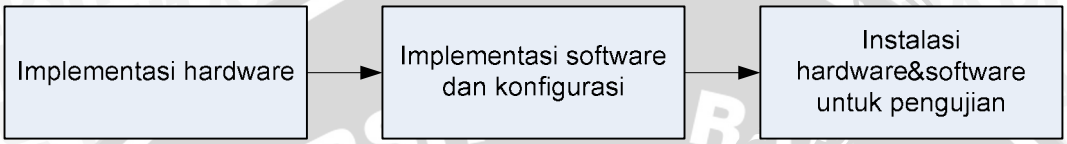
## 3.3 Implementasi dan Pengujian

Tahap awal implementasi adalah implementasi *hardware* yang dilakukan dengan memasang semua alat-alat (*hardware*) sesuai dengan rancangan hasil proses analisis dan perancangan yang telah dilakukan sebelumnya. Langkah selanjutnya adalah implementasi *software* yang dilakukan dengan menginstall *software-software* dan tool yang diperlukan kemudian mensetting konfigurasi yang diperlukan berdasarkan dengan perancangan dan analisis yang dilakukan sebelumnya.

Tahap selanjutnya adalah mengkondisikan jaringan yang telah dibangun sesuai dengan perancangan itu agar siap dilakukan pengujian. Kondisi yang dimaksud adalah:

- Menyiapkan komputer yang digunakan sebagai user/*client* yang akan mengakses jaringan baik secara *wireless* maupun *wired*.

- Pada kedua cangkupan jaringan *wireless* terdapat perpotongan yang di area tersebut mencangkup dua BSS.
- Menyiapkan dan memasang sebuah komputer yang akan digunakan sebagai host yang melakukan *sniffing* untuk menangkap paket sehingga dapat dianalisis kemudian.




Gambar 3.2. Alur kerja implementasi

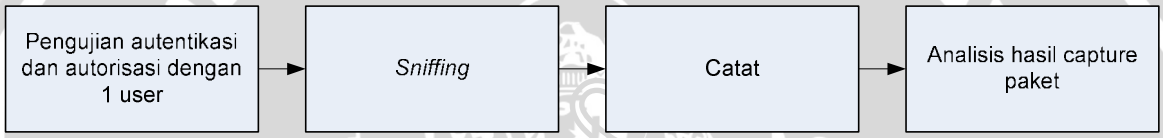
Tahap selanjutnya yang dilakukan adalah pengujian dan analisis hasil mengenai hal-hal yang menjadi poin pada rumusan masalah. Pengujian pertama, yaitu keberhasilan autentikasi dan otorisasi user dilakukan dengan menggunakan sebuah komputer untuk melakukan koneksi ke jaringan fakultas kemudian dilakukan aktifitas autentikasinya (berupa pengisian *username* dan *password*). Pada saat proses autentikasi tersebut, dilakukan proses sniffing paket untuk mengetahui pertukaran paket data yang terjadi. Sniffing ini dilakukan di posisi yang memungkinkan, kemudian hasilnya dicatat.

Pengujian selanjutnya dilakukan dengan mengukur *transfer rate* koneksi pada saat sebelum autentikasi dan sesudah autentikasi kemudian membandingkannya. Pengujian dilakukan pada jaringan yang terisolasi sehingga tidak ada faktor lain yang mempengaruhi pengukuran *transfer rate* ini. Pengukuran *transfer rate* dilakukan dengan bantuan *tool* pengukur *QoS*, yaitu Iperf.

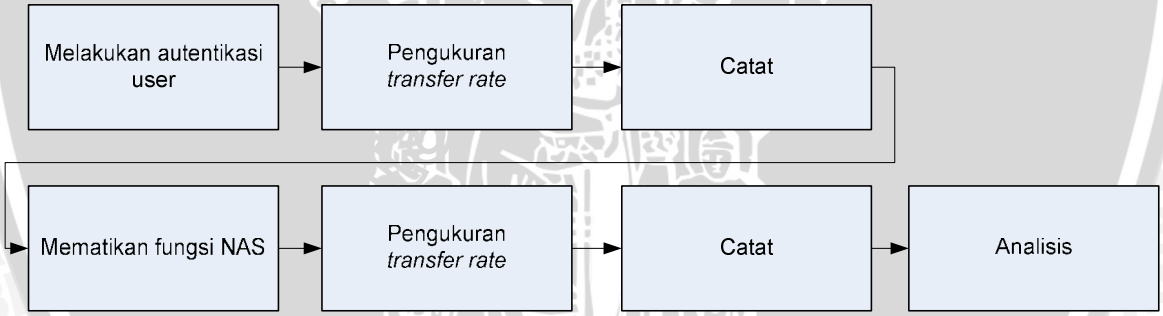
Pengujian ketiga dilakukan dengan mensimulasikan beberapa autentikasi secara bersamaan dengan jumlah yang bervariasi kemudian dihitung *delay* per autentikasinya. Mendapatkan besar *delay* ini dilakukan dengan menggunakan *tool* penganalisis paket. Pengujian ini dilakukan pada NAS atau bisa disebut juga sebagai Radius *client*.

Pengujian keempat, yaitu analisis proses *handoff* dilakukan dengan melakukan koneksi pada suatu *access point* dan melakukan autentikasi user Radius dengan menggunakan sebuah perangkat *mobile*. Kemudian dilakukan *handoff*, yaitu berpindah dari satu BSS ke BSS yang lain dalam satu jaringan. Pada saat sudah terkoneksi dengan access point tujuan, dilakukan pengamatan terhadap koneksi perangkat *mobile* dengan jaringan/internet dan pengamatan terhadap *session* user Radius.

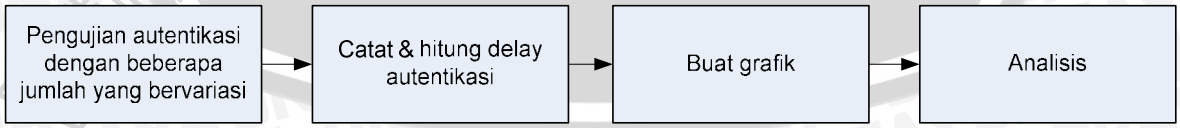
Analisis hasil dilakukan dengan melakukan pencatatan hasil pengujian yang telah dilakukan, Kemudian hasil parameter-parameter yang diukur ditampilkan dalam tampilan yang representatif.



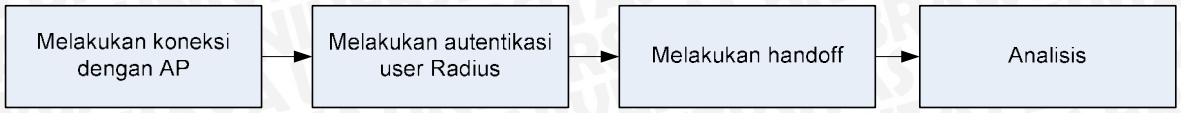

Gambar 3.3. Alur kerja pengamatan proses autentikasi dan otorisasi dan analisis hasilnya




Gambar 3.4. Alur kerja pengujian dan mengukuran transfer rate dan analisisnya




Gambar 3.5. Alur kerja pengujian autentikasi dengan jumlah user yang bervariasi dan analisisnya




Gambar 3.6. Alur kerja pengujian analisis proses *handoff*

## 3.4 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah mendapatkan hasil dari pengujian yang telah dilakukan. Kesimpulan yang didapat merupakan jawaban atas poin-poin yang terdapat dalam rumusan masalah.

# BAB IV
# ANALISIS KEBUTUHAN DAN PERANCANGAN SISTEM

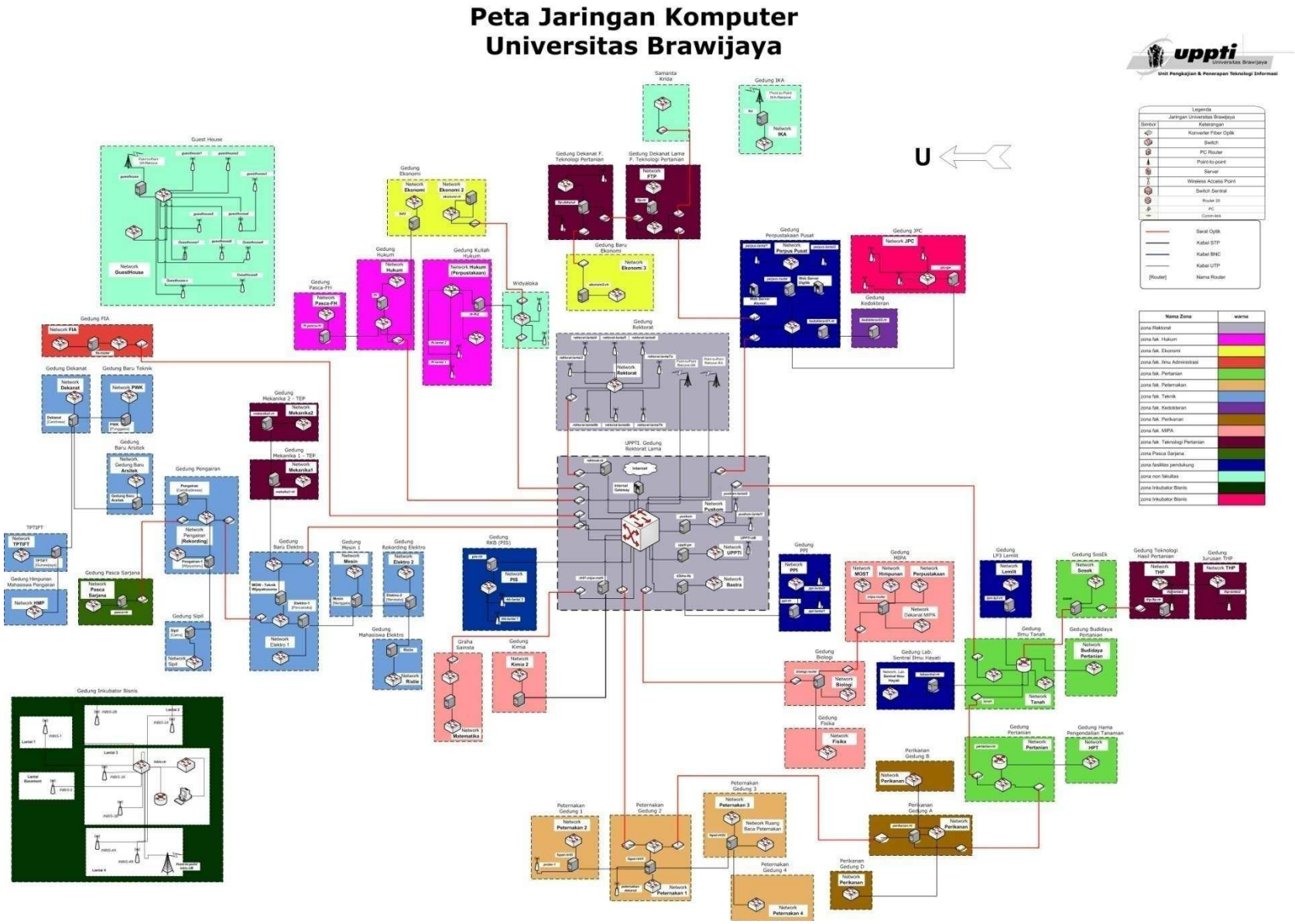
Bab ini membahas mengenai perancangan sistem yang akan dibangun. Perancangan ini meliputi analisis kebutuhan sistem dan perancangan sistem.

## 4.1 Jaringan Komputer Universitas Brawijaya (UB)

Jaringan komputer UB dibagi dalam zona-zona yang rata-rata tiap zonanya dibagi menurut fakultas maupun lembaga. Pada tiap zona terdapat satu *router* yang pada beberapa zona tersebut terdapat beberapa *router* di bawah *router*. *Router-router* tersebut antara lain berfungsi sebagai:

- Pembagi jaringan. Tiap-tiap zona di UB memiliki alokasi alamat IP khusus. Pembagian jaringan ini ditandai dengan pengalokasian alamat IP untuk tiap-tiap zona.
- DHCP *server*. DHCP *server* digunakan sebagai pendistribusi alamat IP dinamik.
- Firewall. Firewall pada jaringan digunakan untuk mengatur paket yang masuk dan keluar jaringan. Tiap-tiap zona memiliki kebutuhan aturan yang beragam.

Gambar 4.1 mengilutrasikan topologi jaringan komputer UB.




Gambar 4.1 Peta Jaringan Komputer UB
Sumber: Dokumentasi PPTI UB

Pada jaringan komputer UB terdapat *Network Operating Center* (NOC) sebagai pusat jaringan komputer UB yang di dalamnya terdapat berbagai *server* dan *gateway* yang mendukung berlangsungnya layanan jaringan komputer UB. Alamat-alamat IP yang terdistribusi pada *user* jaringan UB merupakan alamat IP untuk IP *private.*

Alamat-alamat IP tersebut tergolong alamat IP kelas B. Untuk terkoneksi dengan internet, digunakan mekanisme *Network Address Translation* (NAT) sehingga semua *user* yang menggunakan layanan jaringan UB akan dikenali sebagai beberapa alamat IP yang sama, yaitu alamat IP *public* milik UB.

## 4.2 *User* Jaringan UB

*User* dapat terkoneksi baik melalui jaringan kabel maupun wireless. Infrastruktur jaringan kabel, baik UTP maupun serat optik terdapat pada gedung. Koneksi antar gedung dibangun dengan kabel UTP maupun serat optik. Koneksi ke komputer *user* dibangun dengan kabel UTP maupun wireless.

*User* jaringan di UB memiliki sekitar 3000 *user* yang setiap harinya menggunakan layanan akses internet. *User-user* tersebut dapat digolongkan menjadi beberapa tipe antara lain:

- Dosen dan Pegawai

  Setiap dosen UB memiliki hak menggunakan jaringan komputer UB. Status dosen ini dikenakan pada seseorang hingga pegawai tersebut berhenti atau pensiun.
- Mahasiswa

  Setiap user yang terdaftar sebagai mahasiswa memiliki hak menggunakan jaringan komputer UB. Status mahasiswa ini berlaku selama 1 semester (dengan asumsi satu semester sama dengan 6 bulan) dan akan diperbarui di awal semester.

- Tamu

  Tamu merupakan user yang bersifat sementara yang juga memiliki hak menggunakan layanan jaringan komputer UB jika diperlukan.

Pihak-pihak yang memiliki hak untuk menggunakan layanan jaringan UB adalah yang tergolong pada tipe-tipe *user* yang telah disebutkan di atas. Jika terdapat pihak yang tidak tergolong pada tipe *user* tersebut menggunakan layanan jaringan UB, maka pihak tersebut tidak seharusnya menggunakan layanan jaringan UB. Dengan demikian diperlukan sistem keamanan yang mendata *user* dan hanya memperbolehkan *user-user* yang berhak saja untuk dapat menggunakan layanan jaringan UB. Jika tidak terdapat sistem tersebut maka setiap orang dapat menggunakan layanan akses jaringan UB tanpa diketahui identitasnya. Hal ini sangat mungkin terjadi terutama dikarenakan tersebarnya *access point-access point* secara bebas di lingkungan UB.

## 4.3 Analisis Kebutuhan Fungsional Sistem

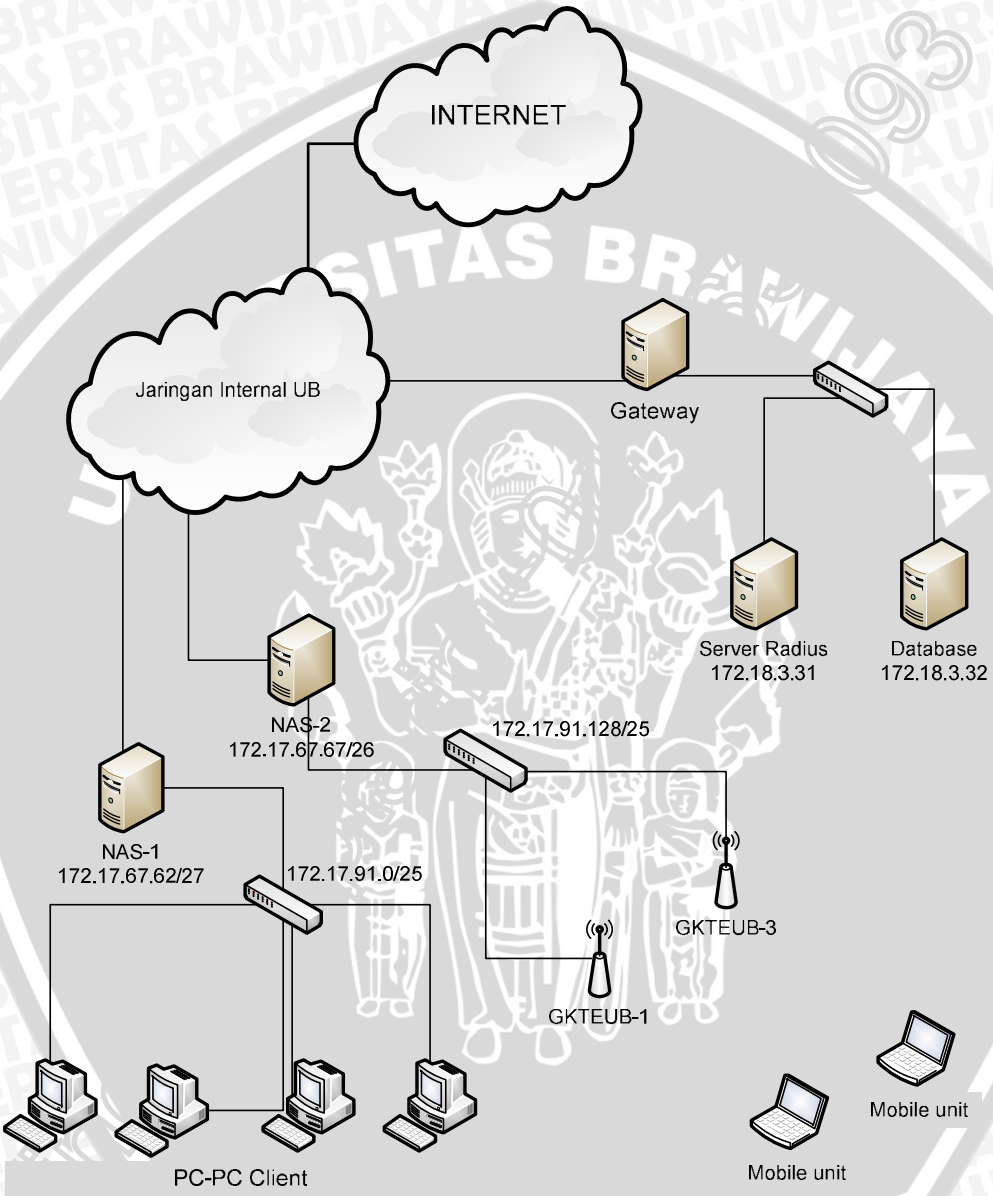
Berdasarkan paparan analisis di atas, maka dapat dirumuskan bahwa:

- Diperlukan sistem autentikasi dan otorisasi *user* pada jaringan komputer UB.
- Diperlukan sistem autentikasi pada jaringan komputer UB yang dapat mengakomodasi peristiwa *handoff*.
- Diperlukan sistem pendataan *user* pada jaringan komputer UB yang dapat membagi jenis-jenis *user* dan otoritasnya.
- Sistem yang akan diterapkan mudah disesuaikan dengan jaringan komputer UB yang telah ada.

## 4.4 Topologi Jaringan *Prototype*

Berikut ini adalah topologi jaringan *prototype* yang akan digunakan dalam penelitian ini. Jaringan *prototype* ini diterapkan di bawah jaringan Fakultas Teknik yang lebih tepatnya di bawah jaringan gedung kemahasiswaan elektro dan di bawah

jaringan gedung kuliah elektro. *Server* Radius dan *server* basis data terdapat pada zona tersendiri terpisah dengan zona NAS dan user. Gambar 4.2 mengilustrasikan peta jaringan *prototype*.




Gambar 4.2 Perancangan jaringan *prototype*
Sumber : Perancangan

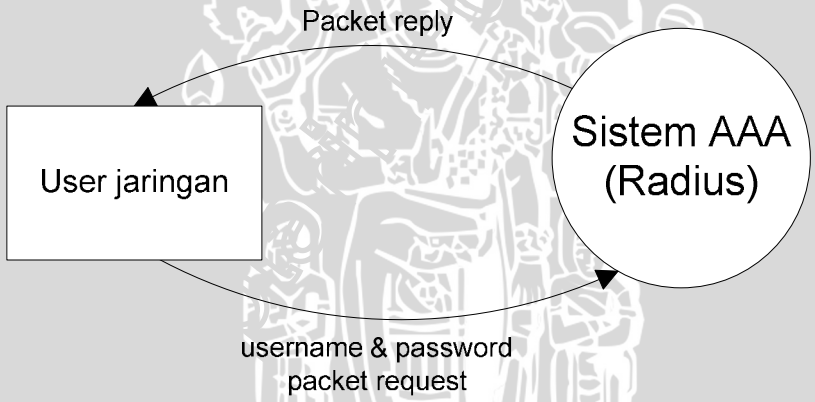
## 4.5 Analisis Data Flow Diagram (DFD)

Data Flow Diagram adalah metode pemodelan analisis untuk menampilkan perpindahan data yang melalui sistem dan menampilkan fungsi dan subfungsi yang mentransformasikan aliran data [PRE-04].

Analisis sistem pada skripsi ini disajikan dengan Context Diagram (DFD level 0) dan DFD level 1.

### 4.5.1 DFD level 0 (Context Diagram)

DFD level 0 merupakan diagram yang menampilkan masukan proses, proses dan keluaran proses dari sistem secara umum yang menggambarkan keterkaitan sistem dengan entitas di luar sistem. DFD level 0 merupakan DFD yang pertama kali disusun. Berikut ini adalah gambar DFD level 0 sistem:




Gambar 4.3 DFD Level 0 (*Context Diagram*) sistem
Sumber: Perancangan

Gambar 4.3 adalah gambar *context diagram* yang menjelaskan gambaran sistem secara umum. Sistem AAA dengan protokol Radius merupakan sistem yang diterapkan untuk memproses autentikasi dan otorisasi user. Sedangkan user jaringan adalah pihak pengguna sistem yang menggunakan sistem dengan cara memberikan *username* dan *password* dan menggunakan layanan jaringan. Dari *context diagram* di atas, Sistem AAA memiliki masukan dan keluaran sebagi berikut.

- *Username* dan *password packet request*

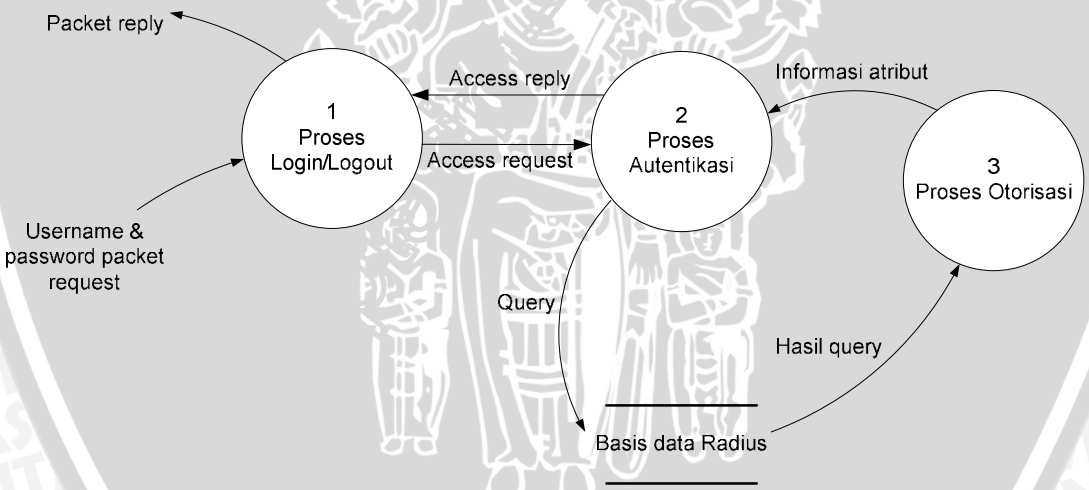
  *Username* dan *password packet request* adalah paket data yang ditujukan kepada sistem AAA yang berisikan informasi *username* dan *password* untuk diautentikasi.

- *Packet reply*

  *Packet reply* yang dikirim dari sistem AAA menuju user berupa packet data yang menginformasikan bahwa autentikasi berhasil maupun autentikasi gagal.

### 4.5.2 DFD level 1 Client ke Server Radius

DFD level 1 merupakan diagram yang menjelaskan sistem secara detail berdasarkan subfungsi sistem. Berikut ini adalah DFD level 1 *Client* ke *Server* Radius dari *context diagram* sebelumnya.




Gambar 4.4 DFD Level 1 Sistem AAA
Sumber: Perancangan

Dari DFD level 1 tersebut dijabarkan proses-prosesnya sebagai berikut.

- Proses login/logout menerima masukan dari user jaringan berupa *username* dan *password.* Dari proses login/logoff tersebut dihasilkan *access request*.

- *Access request* ini sebagai masukan proses autentikasi user.
- Proses autentikasi yang berarti mengecek keaslian user mengolah *access request* kemudian mencari keberadaan user dengan mengirimkan *query* pada basis data Radius. Hasil *query* kemudian menjadi masukan bagi proses otorisasi yang kemudian informasi atribut otorisasi sebagai masukan proses autentikasi sebagai hasil proses otorisasi.
- Hasil autentikasi berupa a*ccess reply* sebagai masukan proses login/logout untuk memberikan informasi pada user jaringan bahwa autentikasi berhasil atau gagal.

## 4.6 Perancangan Sistem

Sesuai dengan analisis kebutuhan sistem, maka diperlukan sistem AAA yang mengakomodasi sistem autentikasi dan otorisasi *user* sebagai solusi permasalahan di atas. Sistem AAA tersebut terdiri dari beberapa komponen, antara lain *server* Radius, *server* basis data, dan NAS. Berikut ini adalah dan software dan spesifikasi hardware minimum yang digunakan sebagai acuan implementasi untuk diterapkan pada sistem.

### 4.6.1 Software

Spesifikasi software yang digunakan adalah sebagai berikut:

Untuk *server* Radius:

- Sistem operasi : Linux Ubuntu 8.10
- Aplikasi Radius : Freeradius 2.1.6-0

Untuk *server* basis data:

- Sistem operasi : Linux Ubuntu 8.04
- Aplikasi basis data : Mysql 5.0.51a-3ubuntu5.4

Untuk NAS-1 dan NAS-2:

- Sistem Operasi : Linux Centos 5.3
- Captive Portal : Chillispot-1.1.0-1

### 4.6.2 Hardware

Spesifikasi hardware minimum yang digunakan *server-server* dan NAS adalah sebagai berikut:

Untuk *server* Radius:

- Server Radius tidak memiliki spesifikasi hardware minimum.
- Alokasi memori pada hardisk : 10 MB

*Server* basis data pada:

- RAM : 1 MB
- Alokasi memori pada hardisk : 4 MB

NAS :

- Prosesor : Pentium 233 MHz
- LAN Card : 2 buah
- RAM : 64 MB
- Alokasi memori pada hardisk : 10 MB

### 4.6.3 Perancangan Jenis User dan Otoritasnya

Sesuai dengan analisis kebutuhan di atas, maka jenis-jenis user dapat dibagi menjadi sebagai berikut.

- Mahasiswa

  Jenis user mahasiswa ini memiliki otoritas sebagai berikut.
  - Berlaku hingga 6 bulan setelah mendaftar
  - Sesi akan putus jika dibiarkan tanpa aktifitas selama 5 menit
  - User akan hanya dapat memiliki satu sesi pada saat yang sama

- Dosen dan Pegawai

  Jenis user ini memiliki otoritas sebagai berikut.
  - Tidak ada batasan masa penggunaan
  - Sesi akan putus jika dibiarkan tanpa aktifitas selama 5 menit
  - User akan hanya dapat memiliki satu sesi pada saat yang sama

- Tamu

  Jenis user ini memiliki otoritas sebagai berikut.
  - Pengunaan digunakan sewaktu-waktu
  - Masa penggunaan terbatas tergantung kebutuhan
  - User akan hanya memiliki satu sesi pada saat yang sama

### 4.6.4 Perancangan Konfigurasi Teknis

Perancangan konfigurasi teknis merupakan acuan untuk tahap implementasi yang akan dilakukan. Secara umum, sistem terdiri dari *Server* Radius, *Server* basis data, NAS, dan *access point*. Pada setiap entitas tersebut dirancang konfigurasi teknis sebagai berikut.

#### 4.6.4.1 *Server* Radius

Pada *server* Radius, dirancang parameter-parameter berikut.

- *Server* mendaftar alamat IP NAS, yang berarti hanya membolehkan IP NAS yang terdaftar untuk dapat ditindaklanjuti paket datanya. Untuk keamanan koneksi antara NAS dan *server*, keduanya memiliki sebuah kata kunci yang sama, atau disebut juga *radiussecret*.
- *Server* memiliki dukungan terhadap DBMS (*Data Base Management System*) yang digunakan pada *server* basis data dalam hal ini digunakan mysql.
- *Server* Radius dikonfigurasi agar dapat berkomunikasi dengan *server* basis data yang memiliki alamat IP yang berbeda dengan *server* Radius namun masih pada satu segmen jaringan yang sama.
- *Firewall server* Radius dikondisikan port 1812 terbuka karena port yang digunakan untuk autentikasi dan otorisasi Radius adalah port 1812.

#### 4.6.4.2 *Server* Basis Data

Pada *server* basis data, direncanakan parameter-parameter berikut.

- *Server* basis data dapat berkomunikasi dengan *server* Radius yang memiliki alamat IP yang berbeda namun masih pada satu segmen jaringan yang sama.
- *Server* basis data memiliki skema tabel standar Radius yang terdapat pada mysql.
- *Firewall server* basis data dikondisikan port 3306 terbuka karena mysql berkomunikasi dengan port 3306.

#### 4.6.4.3 NAS

Fungsi NAS dirancang untuk dapat menggantikan fungsi router. Dengan demikian NAS dirancang untuk dapat memiliki fungsi sebagai berikut:

- Client Radius

  NAS akan berfungsi sebagai autentikator yang menjembatani proses

autentikasi user. Server Radius mendaftar alamat-alamat IP NAS-NAS yang berfungsi sebagai clientnya.

- Pembagi jaringan

  Seperto halnya router, NAS juga berlaku sebagai pembagi jaringan. NAS hanya bekerja pada satu segmen jaringan tertentu.

- Server DHCP

  Dengan adanya DHCP server akan memudahkan user untuk mendapatkan alamat IP secara otomatis.

- *Firewall*

  *Firewall* yang diterapkan pada NAS sedikit berbeda dengan *firewall* yang diterapkan pada router. Pada NAS, diberlakukan firewall yang mendukung fungsi NAS itu sendiri, antara lain adalah fungsi pembatasan akses oleh user saat user tersebut belum melakukan autentikasi. Pada *firewall* juga dihilangkan fungsi NAT.

- HTTPS

  Untuk mengamankan komunikasi data antara NAS dan komputer *user* maka diterapkan fungsi https pada halaman login pada NAS sehingga informasi berharga seperti *username* dan *password* tidak mudah di*sniffing*.

#### 4.6.4.4 *Access Point*

Agar dapat diterapkan proses *handoff*, maka pada jaringan *prototype* ini digunakan dua buah *access point* yang terkoneksi pada jaringan di bawah NAS-2. Kedua buah *access point* ini difungsikan pada lapisan 2 (lapisanMAC) sehingga user yang terkoneksi melalui media *wireless* akan langsung tersambung dengan jaringan di bawah NAS-2. Kedua *access point* difungsikan dengan SSID yang terlihat.

# BAB V
# IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dijelaskan tentang implementasi dan pengujian yang dilakukan pada penelitian ini. Implementasi dan pengujian yang dilakukan berdasarkan analisis dan perancangan yang dilakukan sebelumnya.

## 5.1 Implementasi

Implementasi dilakukan dengan melakukan instalasi baik *hardware* maupun *software*. Setelah dilakukan implementasi *hardware* dan *software,* dilakukan proses manajemen user. Proses manajemen user yang dimaksud adalah penambahan user dan grup usernya beserta atribut otorisasi yang berkaitan. Untuk dapat dilakukan pengujian pada sistem setelah user dan otorisasinya telah ditambahkan, dilakukan persiapan pengujian dengan cara instalasi *software* maupun *hardware* untuk keperluan pengujian.

### 5.1.1 Implementasi Hardware

Instalasi *hardware* dilakukan sesuai dengan topologi jaringan *prototype* yang telah disusun (pada sub bab 4.4) dan spesifikasi *hardware* pada perancangan (pada sub bab 4.6.2). Dalam rangka implementasi *hardware*, *server* Radius dan *Server* basis data diletakkan di gedung Rektorat Lama lantai 2 UB. Sedangkan NAS-1 di Gedung Kemahasiswaan Teknik Elektro UB (TEUB) Lantai 2 dan NAS-2 pada Gedung Kuliah TEUB lantai 1. Peletakan *access point* GKTEUB-3 di Gedung Kuliah TEUB lantai 1 dan *access point* GKTEUB-1 di Gedung Kuliah TEUB lantai 2.

### 5.1.2 Implementasi *Software*

Instalasi *software* dilakukan sesuai dengan spesfikasi *software* pada perancangan (sub bab 4.6.1) dan perancangan konfigurasi teknis (sub bab 4.6.4). Implementasi *software* dilakukan antara lain dengan instalasi dan konfigurasi.

Instalasi dan konfigurasi yang dilakukan antara lain pada *server* Radius, *server* basis data, dan NAS.

#### 5.1.2.1 *Server* Radius

Tahap awal yang dilakukan pada *server* Radius adalah instalasi software yang telah dirancang pada perancangan sebelumnya, antara lain:

- *Operating System* Linux Ubuntu 8.10
  Dalam penginstalan, digunakan mode minimum, yaitu hanya menginstal paket dan software dasar dengan tujuan keamanan dan keringkasan.
- Aplikasi Radius Freeradius 2.1.6-0
  Freeradius yang diinstal merupakan Freeradius versi terbaru pada saat implementasi dilakukan. Paket yang dipilih dalam penginstalan ini antara lain :
  - Freeradius-2.1.6-0, aplikasi dasar server Radius.
  - Freeradius-mysql-2.1.6-0, modul tambahan agar freeradius dapat menggunakan MySQL sebagai *server* basis data.

Konfigurasi yang dilakukan pada server Radius setelah proses instalasi adalah sebagai berikut.

- Konfigurasi alamat IP
  Alamat IP *server* Radius adalah 172.18.3.31 dengan *subnet mask* 255.255.255.0. Untuk menerapkan konfigurasi ini dilakukan penambahan baris pada file /etc/network/interfaces:

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

- Konfigurasi acuan data user

  Pada skripsi ini data user mengacu pada basis data SQL. Hal ini perlu didefinisikan pada file /etc/freeradius/sites-enabled/default dengan menambahkan baris sebagai berikut:

```
accounting {
sql
}
```

- Koneksi dengan basis data MySQL

  Agar dapat terkoneksi dengan basis data MySQL yang tepisah dengan server Radius dilakukan pengeditan parameter-parameter database, server, login, password, dan radius_db yang terdapat dalam file /etc/freeradius/sql.conf:

```
database = "mysql"
driver = "rlm_sql_${database}"
server = "172.18.3.32"
login = "root"
password = "radiuspassword"
radius_db = "radius"
```

- Pencatatan NAS

  NAS yang berfungsi sebagai *client* Radius harus terdaftar pada file /etc/freeradius/clients.conf dengan menambahkan baris berikut pada file tersebut.

  Untuk NAS-1:

```
client 172.17.67.62 {
        secret          = radiussecret
        shortname  = NAS-1
}
```

Untuk NAS-2:

```
client 172.17.67.67 {
        secret          = radiussecret
        shortname  = NAS-2
}
```

Berikut ini adalah *screenshot* proses inisialisasi *server* Radius hingga *server* siap menerima *request.*


Gambar 5.1 Tampilan proses inisialisasi pada server Radius
(Sumber: Implementasi)

### 5.1.2.2 Server Basis Data

Tahap awal implementasi pada *server* basis data adalah instalasi software antara lain:

- *Operating System* Linux Ubuntu 8.04

  Dalam penginstalan, digunakan mode minimum, yaitu hanya menginstal paket dan software dasar dengan tujuan keamanan dan keringkasan.

- Aplikasi DBMS MySQL

  Versi yang diinstal adalah MySQL 5.0.51a-3ubuntu5.4 yang merupakan MySQL versi bawaan dari Ubuntu 8.04.

- Phpmyadmin

  Phpmyadmin merupakan aplikasi berbasis web dengan bahasa pemrograman PHP yang memudahkan untuk mengelola basis data MySQL.

Konfigurasi yang dilakukan setelah melakukan penginstalan software antara lain:

- Konfigurasi alamat IP

  Alamat IP *server* basis data adalah 172.18.3.32 dengan *subnet mask* 255.255.255.0. Untuk menerapkan konfigurasi ini dilakukan penambahan baris pada file /etc/network/interfaces:

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

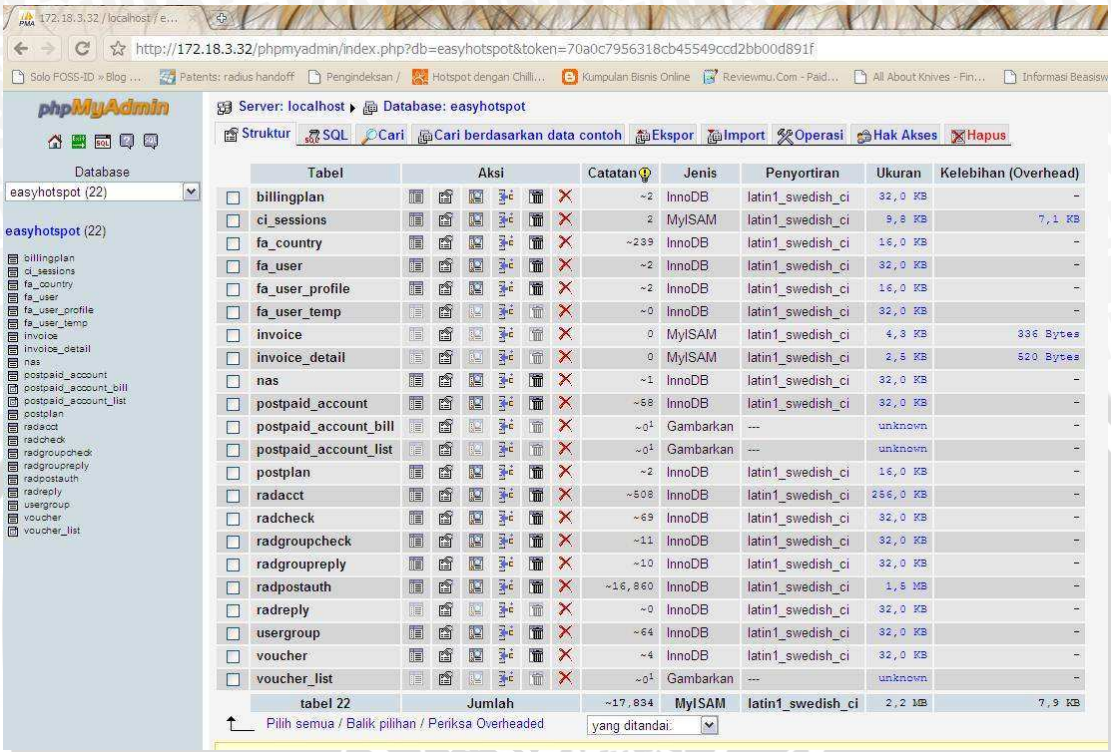
- Instalasi skema basis data Radius

  Skema basis data Radius telah disediakan oleh aplikasi Freeradius. Penerapan skema basis data dilakukan dengan:

```
#mysql –u root –p
>CREATE DATABASE radius;
>quit;
#mysql -u root -p radius <
/etc/freeradius/sql/mysql/schema.sql
```

- Pengkoneksian basis data Radius dengan *server* Radius

Berikut ini adalah *screenshot* hasil implementasi pada *server* Radius. Tampilan basis data disajikan dalam tampilan Phpmyadmin.




Gambar 5.2 Tampilan basis data Radius pada *server* basis data
(Sumber: Implementasi)

### 5.1.2.3 NAS

Tahap-tahap implementasi NAS berlaku untuk NAS-1 dan NAS-2. Sebelum dilakukan konfigurasi, dilakukan instalasi software antara lain:

- *Operating System* Linux Centos 5.3

  Dalam penginstalan, digunakan mode minimum, yaitu hanya menginstal paket dan software dasar dengan tujuan keamanan dan keringkasan.

- Aplikasi *Captive Portal* Chillispot-1.1.0-1

  Chillispot merupakan aplikasi yang menjalankan fungsi NAS pada router. Instalasi dilakukan dengan men*download* langsung dari situs http://www.chillispot.info.

- Aplikasi web *server* Apache

  Karena user mengirimkan *username* dan *password*nya melalui halaman web, maka NAS juga berfungsi sebagai *server* web. Dengan demikian dilakukan instalasi aplikasi *server* web Apache untuk dapat mendukung fungsi NAS.

- Modul Openssl

  Openssl diperlukan untuk menyediakan fasilitas HTTPS (*Hyper Text Transfer Protocol Secure*) pada halaman login user.

Konfigurasi yang dilakukan pada NAS adalah sebagai berikut.

- Konfigurasi alamat IP

  Alamat IP *uplink* pada *server* NAS-1 adalah 172.17.67.62 dengan *subnet mask* 255.255.255.224. Sedangkan pada NAS-2 172.17.67.67 dengan *subnet mask* 255.255.255.224. Untuk menerapkan konfigurasi ini dilakukan penambahan baris pada file /etc/sysconfig/network-script/ifcfg-eth0 pada NAS-1:

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

  dan pada NAS-2:

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

Sedangkan alamat IP *downlink* diset untuk mendapatkan alamat IP secara otomatis melalui DHCP. Pada NAS-1 pengaturan ini ditambahkan pada file /etc/sysconfig/network-script/ifcfg-eth1:

```
DEVICE=eth1
BOOTPROTO=dhcp
ONBOOT=yes
```

Sedangkan *downlink* pada NAS-2 ditambahkan pada file /etc/sysconfig/network-script/ifcfg-eth2:

```
DEVICE=eth2
BOOTPROTO=dhcp
ONBOOT=yes
```

- Fungsi meneruskan paket

  Ip_forward adalah fungsi dalam Linux untuk meneruskan paket yang diterima dari satu antarmuka jaringan ke antarmuka jaringan yang lain. Fungsi ini merupakan fungsi dasar sebuah *router*. Untuk mengaktifkan fungsi ini dilakukan pengeditan pada file /etc/sysctl.conf dengan mengedit bagian ip_forwarding menjadi:

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

- Konfigurasi Chillispot

  Konfigurasi utama Chillispot terletak pada file /etc/chilli.conf. Pada NAS-1, file tersebut diedit dengan parameter-parameter berikut.

```
net             172.17.91.0/25
dynip           172.17.91.0/25
dns1            202.162.208.99
domain          skripsiTEUB.brawijaya.ac.id
radiusserver1   172.18.3.31
radiussecret    radiussecret
dhcpif          eth2
uamserver                  https://172.17.91.1/cgi-
bin/hotspotlogin.cgi
uamsecret       uamsecret
uamhomepage     http://172.17.91.1/login/
uamlisten       172.17.91.1
uamallowed       172.17.91.1
```

  Pada NAS-2, file /etc/chilli.conf diedit dengan parameter-parameter sebagai berikut.

```
net             172.17.91.128/25
dynip           172.17.91.128/25
dns1            202.162.208.99
domain          skripsiTEUB.brawijaya.ac.id
radiusserver1   172.18.3.31
radiussecret    radiussecret
dhcpif          eth2
uamserver                 https://172.17.91.129/cgi-
bin/hotspotlogin.cgi
uamsecret       uamsecret
uamhomepage     http://172.17.91.129/login/
uamlisten       172.17.91.129
uamallowed       172.17.91.129
```

- Firewall pada NAS

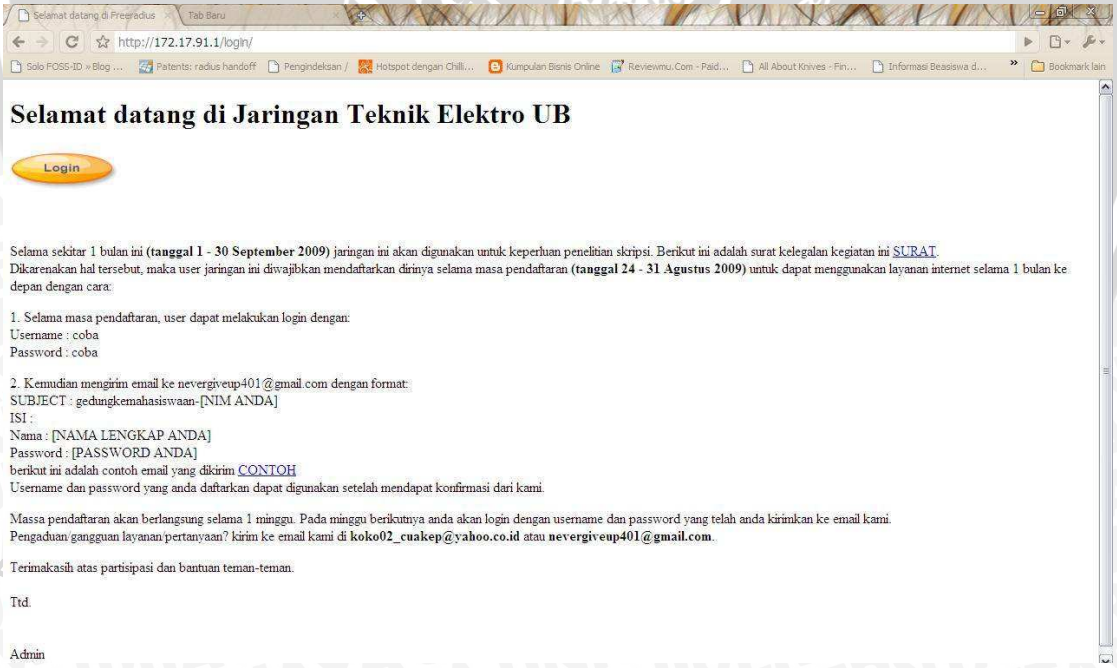
  Chillispot telah menyediakan serangkaian rule firewall yang diterapkan pada pada NAS. Untuk mengaktifkan firewall tersebut diaktifkan dengan:

```
#sh /usr/share/doc/chillispot/firewall.iptables
```

- Halaman pembuka

  Halaman pembuka digunakan sebagai sarana untuk menampung pengumuman-pengumuman yang ditujukan pada user sebelum memasuki halaman login dan melakukan autentikasi. Halaman pembuka akan selalu tampil pada browser user setiap user terkoneksi dengan Radius. Halaman pembuka pada skripsi ini dibuat dengan bahasa html.

  Berikut ini adalah *screenshot* halaman pembuka sebelum memasuki halaman login.




Gambar 5.3 Tampilan halaman pembuka
(Sumber: Implementasi)

Berikut ini adalah *screenshot* halaman login yang tampil pada setiap *browser* user.


Gambar 5.4 Tampilan halaman login
(Sumber: Implementasi)

### 5.1.3 Manajemen User

Penambahan user dilakukan dengan melakukan input pada basis data Radius dengan bantuan aplikasi Phpmyadmin. Satu jenis user diwujudkan dalam bentuk satu grup user yang tiap grup user tersebut merupakan kumpulan dari user-user yang memiliki otorisasi yang sama. Penambahan user dilakukan dengan melakukan input pada basis data Radius dengan bantuan aplikasi Phpmyadmin. Berikut ini adalah grup user dan user-user contoh yang ditambahkan pada basis data beserta otorisasinya. Tiap otorisasi diwakili oleh satu atribut user.

Tabel 5.1 Pembagian grup user beserta otoritasnya

| Nama Grup User | Otorisasi | Atribut | Contoh nama user |
|---|---|---|---|
| dosen | *Session* tidak bisa diduplikat | Simultaneous-Use := 1 | dosen_1 |
| | Setelah waktu tertentu tidak ada aktifitas, dianggap logout | Idle-Timeout:=500 | |
| mahasiswa | *Session* tidak bisa diduplikat | Simultaneous-Use := 1 | mahasiswa_1 |
| | Setelah waktu tertentu tidak ada aktifitas, dianggap logout | Idle-Timeout:=500 | |
| | Masa penggunaan 6 bulan | Max-All-Session:=15552000 | |
| tamu | *Session* tidak bisa diduplikat | Simultaneous-Use := 1 | tamu_1 |
| | Setelah waktu tertentu tidak ada aktifitas, dianggap logout | Idle-Timeout:=500 | |
| | Masa penggunaan satu minggu | Max-All-Session:= 604800 | |

(Sumber: Implementasi)

### 5.1.4 Implementasi untuk Pengujian

Tahap berikutnya yang dilakukan pada implementasi adalah persiapan pengujian. Untuk melakukan pengujian digunakan sebuah komputer uji. Spesifikasi komputer uji tersebut adalah sebagai berikut.

- Prosesor : Intel Core 2 Duo T5300 @1.73 GHz
- RAM : 2 GB
- LAN Card : Realtek RTL8168/8111 PCI-E Gigabit Ethernet NIC
- Wireless Card : Intel PRO/Wireless 3945ABG

Pada pengujian akan dibutuhkan beberapa *tool* berupa *software* yang akan membantu dalam melakukan pengujian. *Software-software* tersebut antara lain:

- Tcpdump

  Tcpdump adalah program yang digunakan untuk analisis paket. Tcpdump menampilkan paket yang melewati *interface* jaringan pada sebuah komputer yang menjalankan program ini. Informasi yang ditampilkan oleh tcpdump dapat diatur sehingga sesuai dengan keinginan. Program ini diinstal pada NAS dan *server*.
- Wireshark

  Wireshark adalah *software* yang digunakan untuk analisis paket. Paket yang dianalisis adalah paket yang melewati *interface* jaringan yang terdapat Wireshark di dalamnya. Perbedaan Wireshark dengan Tcpdump adalah wireshark berbasis GUI. Wireshark dilengkapi dengan fasilitas filter paket dan pengurutan dan menampilkan paket secara detail. *Software* ini diinstal baik pada PC uji, NAS, maupun pada *server*.
- Iperf

  Iperf adalah *tool* yang digunakan untuk mengukur *througput* dari sebuah network. Iperf bekerja dengan sistem *server-client* kemudian Iperf membangkitkann paket TCP dan UDP dari *server* menuju *client*. Port

default iperf adalah TCP 5001. *Tool* ini tersedia dalam berbagai versi, baik Windows XP maupun Linux. *Software* ini diinstal pada PC uji sebagai komputer *user* dan server Radius.

- Radclient

  Radclienr adalah program untuk radius client. Program ini bekerja dengan mengirim paket pada Server Radius kemudian menampilkan paket balasannya. Program ini biasa digunakan untuk pengetesan konfigurasi pada Server Radius atau juga dapat digunakan untuk memonitor apakah Server Radius telah bekerja. Radclient dapat mengirim paket kepada server sesuai dengan atribut yang dapat disesuaikan [MAN-09].

- Ping

  Ping adalah *tool* yang digunakan untuk mengecek konektivitas dari satu *host* ke *host* lain. Ping juga dapat digunakan untuk mengukur QoS pada suatu jaringan. QoS yang dapat diukur dari tool ini adalah *latency* (*round-trip time*) dan *packet loss*. *Tool* ini dimiliki oleh hampir semua sistem operasi, termasuk yang digunakan di penelitian ini yaitu Linux dan Windows XP [WIK-09].

## 5.2 Pengujian

Pengujian yang dilakukan bertujuan untuk mengetahui hal-hal sebagai berikut.

- Proses autentikasi dan otorisasi yang terjadi.
- Perbedaan *transfer rate* pada penggunaan saat sebelum melakukan autentikasi dan otorisasi dengan sesudah melakukan autentikasi dan otorisasi.
- Perbedaan *delay* proses autentikasi pada beberapa variasi jumlah proses autentikasi yang terjadi.
- Proses *handoff* layer 2 yang terjadi.
- Besar *latency* pada proses *handoff* layer 2 yang terjadi.

Untuk itu pengujian dilakukan beberapa tahap. Tahap-tahap tersebut dijelaskan pada sub bab berikut.

### 5.2.1 Mengetahui Proses Autentikasi dan Otorisasi

Pengujian proses autentikasi dan otorisasi bertujuan untuk mengetahui keberhasilan proses autentikasi dan otorisasi yang terjadi pada sistem. Selain itu pengujian ini juga bertujuan untuk membandingkan teori dengan hasil pengujian yang didapatkan.

#### 5.2.1.1 Prosedur Pengujian

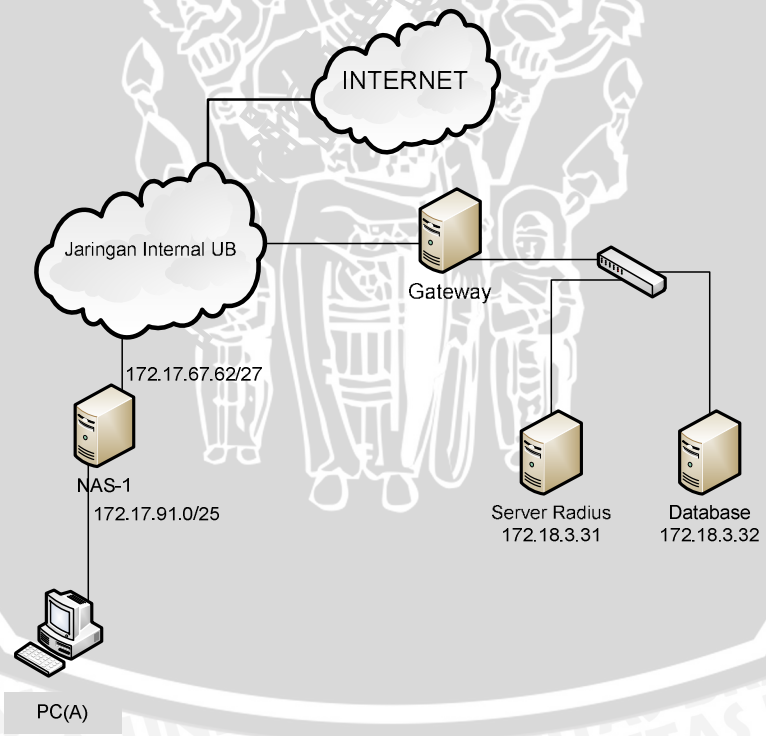
Pengujian ini dilakukan dua kali. Yaitu percobaan autentikasi berhasil kemudian percobaan autentikasi gagal. Pengujian autentikasi berhasil dilakukan dengan:

- Mengaktifkan *server* Radius dan *server* basis datanya. Program Freeradius pada *server* dijalankan pada *debug* mode, sehingga dapat menampilkan proses-proses yang terjadi pada Radius.
- Mengkoneksikan sebuah komputer uji (MN-1) pada jaringan di bawah NAS-1. Kemudian komputer tersebut akan mendapatkan alamat IP yang diberikan oleh NAS-1 yang juga merupakan DHCP server.
- Saat membuka browser pada MN-1, akan tampil halaman login yang di dalamnya terdapat form untuk mengisikan *username* dan *password user*. Kemudian dilakukan autentikasi pada MN-1 dengan mengisikan *username* dan *password*.
- Pada saat yang sama, dilakukan *sniffing* pada NAS-1 dan *server* Radius. Dengan *sniffing* ini kita dapat menganalisis paket dengan menggunakan *software* tcpdump. *Software* ini akan menampilkan paket-paket yang masuk dan keluar melalui interface tertentu.

Setelah melakukan pengujian autentikasi berhasil, dilakukan pengujian autentikasi gagal dengan prosedur:

- Melakukan proses *logout* (keluar) pada MN-1.
- Melakukan autentikasi dengan mengisikan lagi *username* dan *password* pada form di halaman web dengan kombinasi *username* dan *password* yang salah sehingga autentikasi gagal.
- Pada saat yang sama, dilakukan *sniffing* pada NAS-1 dan *server* Radius. Dengan *sniffing* ini kita dapat menganalisis paket dengan menggunakan *software* tcpdump. *Software* ini akan menampilkan paket-paket yang masuk dan keluar melalui interface tertentu.

Gambar di bawah ini menjelaskan topologi jaringan untuk pengujian mengetahui proses autentikasi dan otorisasi yang terjadi.




Gambar 5.5 Topologi jaringan untuk pengujian mengetahui proses autentikasi dan otorisasi
(Sumber: Perancangan)

#### 5.2.1.2 Hasil Pengujian

Untuk percobaan autentikasi berhasil didapatkan hasil sebagai berikut.

Hasil analisis paket pada NAS-1:

```
[root@NAS-1 ~]# tcpdump -i eth0 port 1812 or port 1813
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:43:57.568605 IP 172.18.3.20.35462 > 172.18.3.31.radius: RADIUS, Access Request (1), id: 0x00 length: 199
16:43:57.603750 IP 172.18.3.31.radius > 172.18.3.20.35462: RADIUS, Access Accept (2), id: 0x00 length: 32
16:43:57.604635 IP 172.18.3.20.55067 > 172.18.3.31.radius-acct: RADIUS, Accounting Request (4), id: 0x16 length: 135
16:43:57.626183 IP 172.18.3.31.radius-acct > 172.18.3.20.55067: RADIUS, Accounting Response (5), id: 0x16 length: 20

4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

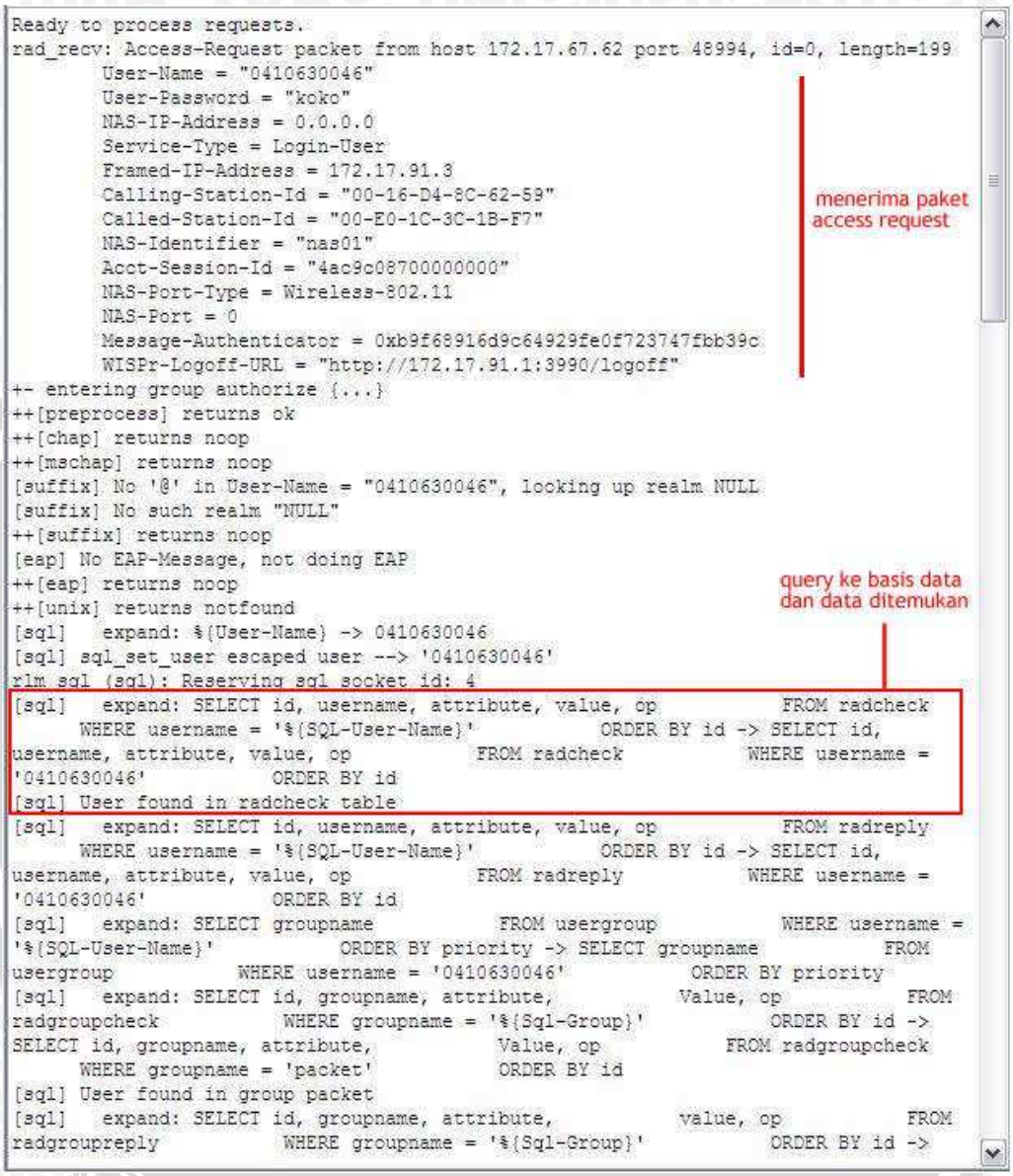
Gambar 5.6. Hasil analisa paket pada NAS-1 saat autentikasi berhasil
(Sumber: Pengujian)

Hasil analisis paket pada *server* Radius:

```
root@radiusserver:~# tcpdump -i eth1 port 1812 or port 1813
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
16:57:50.180869 IP 172.18.3.20.55067 > radiusserver.local.radius-acct: RADIUS, Accounting Request (4), id: 0x1b length: 183
16:57:50.209503 IP radiusserver.local.radius-acct > 172.18.3.20.55067: RADIUS, Accounting Response (5), id: 0x1b length: 20
16:57:58.905887 IP 172.18.3.20.60211 > radiusserver.local.radius: RADIUS, Access Request (1), id: 0x00 length: 199
16:57:58.947104 IP radiusserver.local.radius > 172.18.3.20.60211: RADIUS, Access Accept (2), id: 0x00 length: 32
16:57:58.947826 IP 172.18.3.20.55067 > radiusserver.local.radius-acct: RADIUS, Accounting Request (4), id: 0x1c length: 135
16:57:58.976112 IP radiusserver.local.radius-acct > 172.18.3.20.55067: RADIUS, Accounting Response (5), id: 0x1c length: 20
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

Gambar 5.7. Hasil analisa paket pada *server* Radius saat autentikasi berhasil
(Sumber: Pengujian)

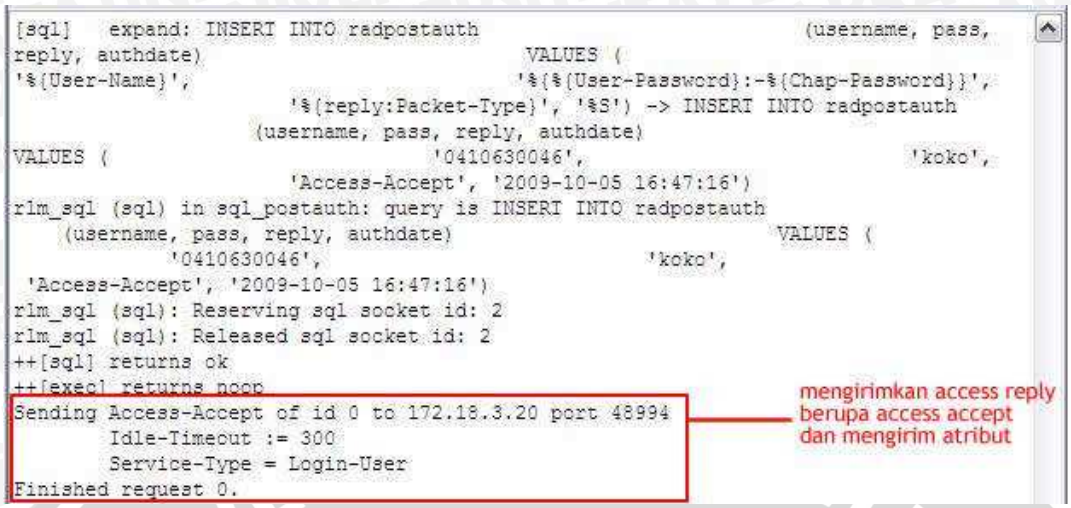
Berikut ini adalah cuplikan proses yang terjadi pada server Radius saat autentikasi berhasil.




```
Ready to process requests.
rad_recv: Access-Request packet from host 172.17.67.62 port 48994, id=0, length=199
        User-Name = "0410630046"
        User-Password = "koko"
        NAS-IP-Address = 0.0.0.0
        Service-Type = Login-User
        Framed-IP-Address = 172.17.91.3
        Calling-Station-Id = "00-16-D4-8C-62-59"
        Called-Station-Id = "00-E0-1C-3C-1B-F7"
        NAS-Identifier = "nas01"
        Acct-Session-Id = "4ac9c08700000000"
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 0
        Message-Authenticator = 0xb9f68916d9c64929fe0f723747fbb39c
        WISPr-Logoff-URL = "http://172.17.91.1:3990/logoff"
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[unix] returns notfound
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
rlm_sql (sql): Reserving sql socket id: 4
[sql]   expand: SELECT id, username, attribute, value, op          FROM radcheck
      WHERE username = '%{SQL-User-Name}'          ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radcheck          WHERE username =
'0410630046'          ORDER BY id
[sql] User found in radcheck table
[sql]   expand: SELECT id, username, attribute, value, op          FROM radreply
      WHERE username = '%{SQL-User-Name}'          ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radreply          WHERE username =
'0410630046'          ORDER BY id
[sql]   expand: SELECT groupname          FROM usergroup          WHERE username =
'%{SQL-User-Name}'          ORDER BY priority -> SELECT groupname          FROM
usergroup          WHERE username = '0410630046'          ORDER BY priority
[sql]   expand: SELECT id, groupname, attribute,          Value, op          FROM
radgroupcheck          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          Value, op          FROM radgroupcheck
      WHERE groupname = 'packet'          ORDER BY id
[sql] User found in group packet
[sql]   expand: SELECT id, groupname, attribute,          value, op          FROM
radgroupreply          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
```

Gambar 5.8 Tampilan proses autentikasi pada *server* Radius
(Sumber: Pengujian)


Gambar 5.9 Tampilan proses autentikasi pada server Radius (lanjutan)
(Sumber: Pengujian)




Gambar 5.10 Tampilan proses autentikasi pada server Radius (lanjutan)
(Sumber: Pengujian)

Untuk percobaan autentikasi gagal didapatkan hasil sebagai berikut.

Pada NAS-1:

```
[root@NAS-1 ~]# tcpdump -i eth0 port 1812 or port 1813
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:49:40.479658 IP 172.18.3.20.46592 > 172.18.3.31.radius: RADIUS, Access Request (1), id: 0x00 length: 199
16:49:41.484895 IP 172.18.3.31.radius > 172.18.3.20.46592: RADIUS, Access Reject (3), id: 0x00 length: 20

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

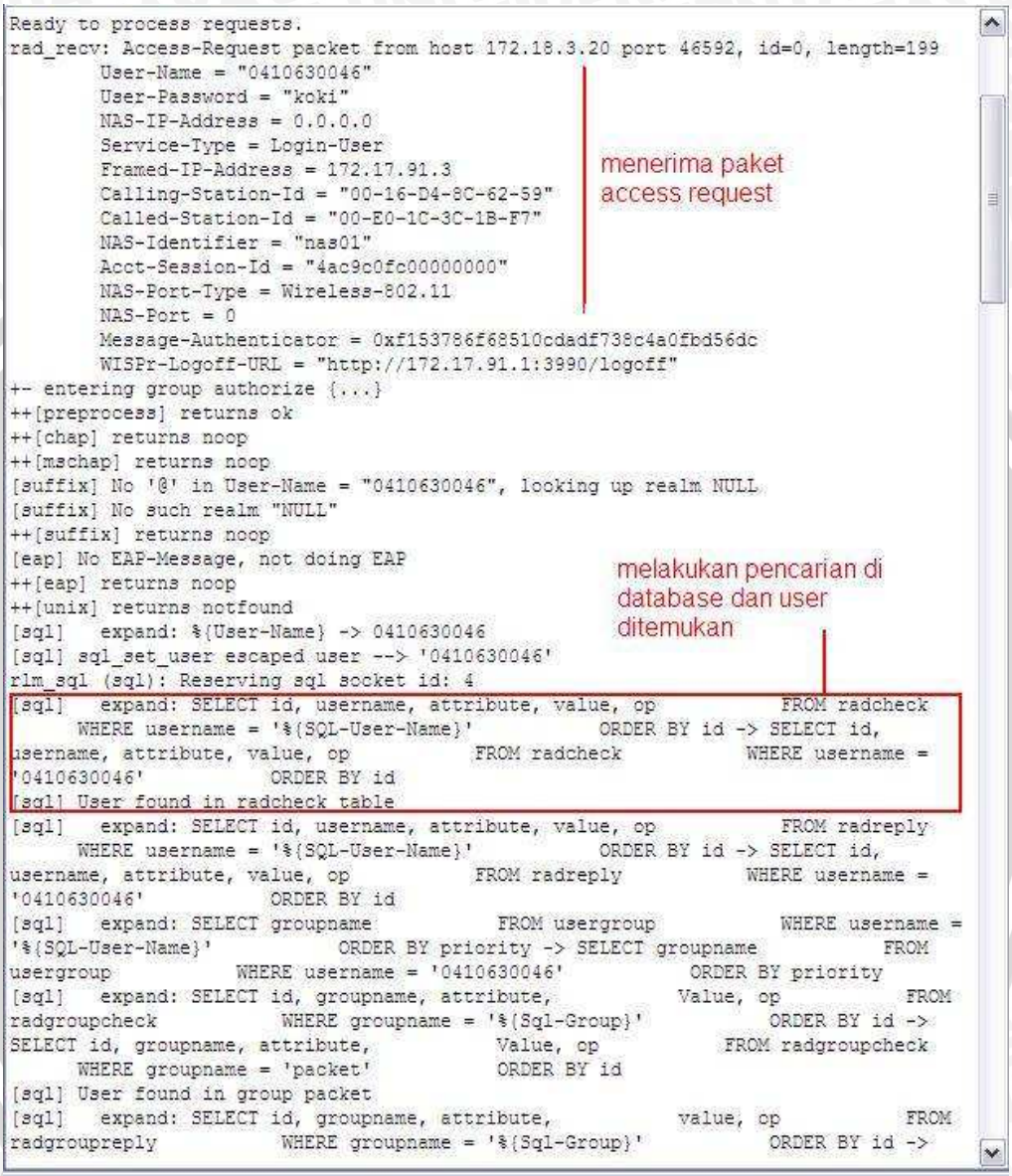
Gambar 5.11. Hasil analisa paket pada NAS-1 saat autentikasi gagal
(Sumber: Pengujian)

Pada *server* Radius:

```
root@radiusserver:~# tcpdump -i eth1 port 1812
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
16:49:47.397062 IP 172.18.3.20.46592 > radiusserver.local.radius: RADIUS, Access Request (1), id: 0x00 length: 199
16:49:48.402364 IP radiusserver.local.radius > 172.18.3.20.46592: RADIUS, Access Reject (3), id: 0x00 length: 20
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

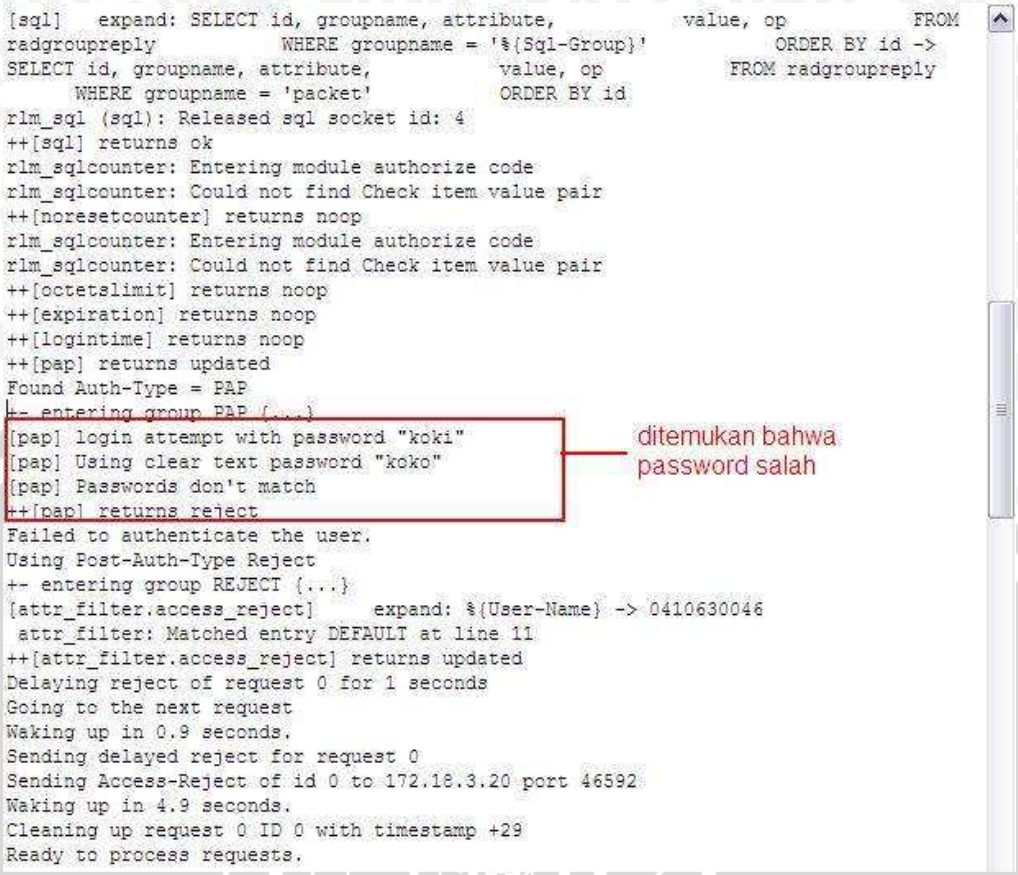
Gambar 5.12. Hasil analisa paket pada *server* Radius saat autentikasi gagal
(Sumber: Pengujian)

Berikut ini adalah cuplikan proses yang terjadi pada server Radius saat terjadi autentikasi yang gagal.




```
Ready to process requests.
rad_recv: Access-Request packet from host 172.18.3.20 port 46592, id=0, length=199
        User-Name = "0410630046"
        User-Password = "koki"
        NAS-IP-Address = 0.0.0.0
        Service-Type = Login-User
        Framed-IP-Address = 172.17.91.3
        Calling-Station-Id = "00-16-D4-8C-62-59"
        Called-Station-Id = "00-E0-1C-3C-1B-F7"
        NAS-Identifier = "nas01"
        Acct-Session-Id = "4ac9c0fc00000000"
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 0
        Message-Authenticator = 0xf153786f68510cdadf738c4a0fbd56dc
        WISPr-Logoff-URL = "http://172.17.91.1:3990/logoff"
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[unix] returns notfound
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
rlm_sql (sql): Reserving sql socket id: 4
[sql]   expand: SELECT id, username, attribute, value, op          FROM radcheck
      WHERE username = '%{SQL-User-Name}'           ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radcheck          WHERE username =
'0410630046'          ORDER BY id
[sql] User found in radcheck table
[sql]   expand: SELECT id, username, attribute, value, op          FROM radreply
      WHERE username = '%{SQL-User-Name}'           ORDER BY id -> SELECT id,
username, attribute, value, op          FROM radreply          WHERE username =
'0410630046'          ORDER BY id
[sql]   expand: SELECT groupname          FROM usergroup          WHERE username =
'%{SQL-User-Name}'          ORDER BY priority -> SELECT groupname          FROM
usergroup          WHERE username = '0410630046'          ORDER BY priority
[sql]   expand: SELECT id, groupname, attribute,          Value, op          FROM
radgroupcheck          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          Value, op          FROM radgroupcheck
      WHERE groupname = 'packet'          ORDER BY id
[sql] User found in group packet
[sql]   expand: SELECT id, groupname, attribute,          value, op          FROM
radgroupreply          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
```

Gambar 5.13 Tampilan proses autentikasi gagal pada *server* Radius
(Sumber: Pengujian)




Gambar 5.14 Tampilan proses autentikasi gagal pada server Radius (lanjutan)
(Sumber: Pengujian)

#### 5.2.1.3 Analisis Hasil

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

- NAS mengirimkan paket Access-Request kepada *server* Radius. Paket tersebut berisi *username* dan *password* beserta atribut lain yang menyertainya.
- *Server* Radius kemudian menerima paket tersebut dan mengolahnya dengan mencari *record* yang sesuai dengan atribut pada paket Access-Request pada *server* basis data.
- Setelah *record* ditemukan, sebagian atribut otorisasi diolah server untuk diterapkan sebelum paket Access-Accept dikirim ke NAS.

Sebagian atribut yang lain dikirim dari *server* ke NAS melalui paket Access-Accept. Paket yang dikirim dan diterima NAS tampak pada gambar 5.2. Sedangkan paket yang diterima dan dikirim oleh server Radius tampak pada gambar 5.3.

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

- NAS mengirimkan paket Access-Request kepada *server* Radius. Paket tersebut berisi *username* dan *password* beserta atribut lain yang menyertainya.
- *Server* Radius kemudian menerima paket tersebut dan mengolahnya dengan mencari *record* yang sesuai dengan atribut pada paket Access-Request pada *server* basis data.
- Karena *record* yang sesuai tidak ditemukan, maka *server* Radius mengirimkan paket Access-Reject pada NAS. Paket yang dikirim dan diterima NAS tampak pada gambar 5.4. Sedangkan paket yang diterima dan dikirim oleh *server* Radius tampak pada gambar 5.5.

### 5.2.2 Membandingkan Besar Transfer *Rate*

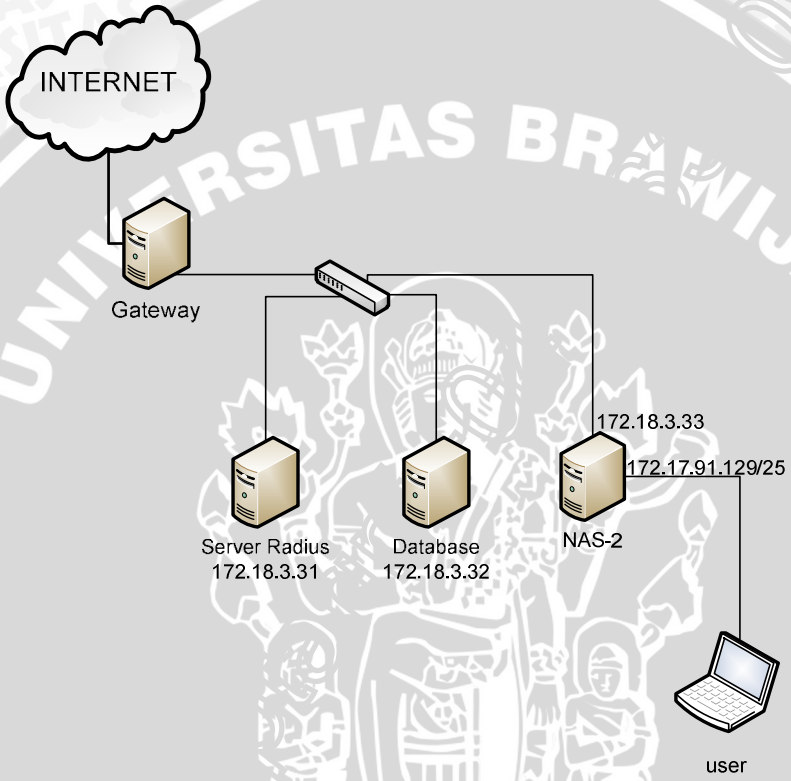
Tujuan pengujian ini adalah untuk membandingkan besar *transfer rate* pada pengaksesan suatu *resource* saat sebelum diterapkan sistem dan saat sesudah diterapkan sistem. Pengujian ini dilakukan dengan harapan agar setelah sistem diterapkan tidak terdapat penurunan kualitas yang siginifikan.

#### 5.2.2.1 Prosedur Pengujian

Untuk menguji besar *transfer rate* pada saat user belum terautentikasi dilakukan tahap-tahap sebagai berikut:

- Mematikan fungsi NAS pada *router*. Hal ini dilakukan dengan mematikan *firewall* pada *router* dan menghentikan *service* Chillispot pada *router*.

- Alamat IP client, dalam hal ini MN-1, diset statis menjadi alamat IP yang masih pada segmen jaringan tersebut. Ketiganya kemudian dikoneksikan ke *router* dengan kabel UTP. Gambar berikut ini menjelaskan topologi jaringan pengujian untuk mendapatkan besar *transfer rate* data.




Gambar 5.15 Topologi jaringan pengujian untuk mengetahui besar *transfer rate* data (Sumber: Perancangan)

- Pada pengujian ini digunakan *tool* Iperf untuk mengukur besar *transfer rate* data. *Server* Radius diperlakukan sebagai *server* Iperf. Kemudian MN-1 diperlakukan sebagai *client* Iperf.
- Pengambilan data dilakukan sebanyak 10 kali secara berurutan untuk mendapatkan hasil yang bervariasi.

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

- Mengaktifkan fungsi router sebagai NAS dengan cara mengaktifkan *firewall* dan *service* Chillispot.
- Alamat IP pada ketiga *client* diset dinamis atau sebagai *client* DHCP kemudian mengkoneksikan komputer uji (MN-1) pada *router*.
- Dalam pengujian ini juga digunakan *tool* Iperf. *Server* Radius diperlakukan sebagai *server* Iperf. Sedangkan MN-1 diperlakukan sebagai *client* Iperf.
- Pengambilan data dilakukan sebanyak 10 kali secara berurutan untuk mendapatkan hasil yang bervariasi.

#### 5.2.2.2 Hasil Pengujian

Pada pengujian besar *transfer rate* pada saat tidak menggunakan Radius.

Tabel 5.2 Hasil pengukuran menggunakan iperf pada saat tidak menggunakan Radius

| No. | Transfer rate (Mbits/s) |
|---|---|
| 1 | 91,7 |
| 2 | 92,5 |
| 3 | 92,3 |
| 4 | 92,3 |
| 5 | 91,9 |
| 6 | 92,1 |
| 7 | 92,2 |
| 8 | 91,1 |
| 9 | 92,6 |
| 10 | 92,2 |
| rata-rata | 92,09 |

(Sumber: Pengujian)

Pada pengujian besar *transfer rate* pada saat user berhasil terautentikasi

didapat hasil sebagai berikut.

Tabel 5.3 Hasil pengukuran pada saat user berhasil terautentikasi

| No. | Transfer rate (Mbits/s) |
|---|---|
| 1 | 87,3 |
| 2 | 86,9 |
| 3 | 86,9 |
| 4 | 87,7 |
| 5 | 86,5 |
| 6 | 86,4 |
| 7 | 86,4 |
| 8 | 86,4 |
| 9 | 86,4 |
| 10 | 89,7 |
| rata-rata | 87,06 |

(Sumber: Pengujian)

#### 5.2.2.3 Analisis Hasil

Dari sepuluh hasil pengujian *transfer rate* pada saat tidak diterapkan Radius didapat rata-rata *transfer rate* sebesar 92,09 Mbit/s dan dari sepuluh hasil pengujian *transfer rate* pada saat user telah berhasil terautentikasi didapat rata-rata *transfer rate* sebesar 87,06 Mbit/s.

Besar perbandingan *transfer rate* antara layanan jaringan sebelum diterapkan Radius dengan layanan jaringan pada saat sesudah diterapkan Radius (telah berhasil terautentikasi) adalah:

$$= \left[1 - \frac{\text{transfer rate sesudah autentikasi}}{transfer\ rate belum\ diterapkan\ sistem}\right] \text{ x } 100\ \%$$

$$= \left[1 - \frac{87{,}06\ Mbit/s}{92{,}09\ Mbit/s}\right] \text{ x } 100\ \%$$

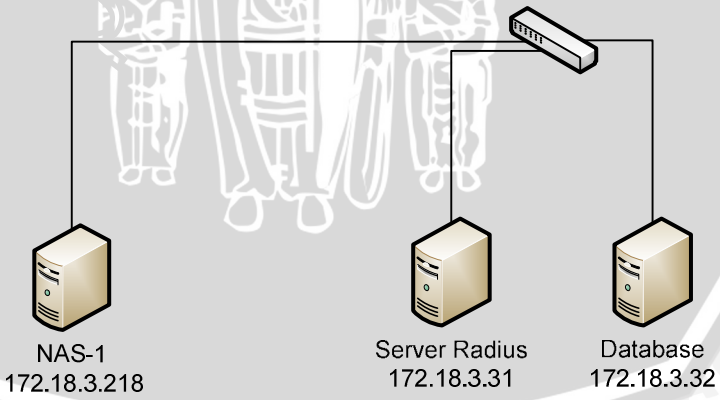
$$= 5{,}46\ \%$$

### 5.2.3 Mengetahui perbedaan *delay* autentikasi

Tujuan pengujian ini adalah untuk mengetahui kemampuan Server Radius menangani autentikasi. Pengujian dilakukan melakukan dengan beberapa variasi jumlah autentikasi yang terjadi bersamaan kemudian menghitung rata-rata *delay* autentikasi per satu user nya.

#### 5.2.3.1 Prosedur Pengujian

- Jaringan dikondisikan seperti Gambar 5.7 di bawah ini. Simulasi autentikasi dilakukan di NAS-1.




Gambar 5.16 Topologi jaringan pengujian untuk mengetahui variasi delay autentikasi (Sumber: Perancangan)

- Pada NAS-1 dilakukan simulasi autentikasi sebanyak 1 user. Simulasi autentikasi dilakukan dengan menggunakan program Radclient.
- Pada saat yang sama, pada NAS-1 dilakukan *sniffing* dengan menggunakan tcpdump. Dengan demikian akan tampak paket Radius yang keluar dari NAS-1 (paket *request*) dan paket Radius yang masuk ke NAS-1 (paket *response*) sekaligus waktu keluar dan masuknya paket tersebut. Dengan melihat perbedaan waktu ini dapat diambil waktu *delay* autentikasi.
- Prosedur ini juga dilakukan dengan jumlah autentikasi yang bervariasi. Variasi jumlah autentikasi tersebut adalah 1, 5, 20, 100, 200, 500, 1000, dan 2000.
- Pada tampilan hasil *tcpdump* didapatkan paket-paket *request* dan *response* yang keluar melalui uplink NAS-1 beserta pada waktunya. Dari data tersebut akan dihasilkan *delay* autentikasi yang merupakan selisih antara paket *response* dengan paket *request*. Pada percobaan ini, autentikasi yang terjadi adalah autentikasi yang berhasil.
- Untuk mendapatkan satu nilai *delay* autentikasi, dilakukan perhitungan rata-rata delay autentikasi tersebut.

### 5.2.3.2 Hasil Pengujian

Berikut ini adalah sampel hasil *capture* paket. Sampel hasil *capture* paket yang ditampilkan adalah simulasi 5 autentikasi bersamaan.

```
14:05:08.609303 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS,
Access Request (1), id: 0xd0 length: 50
14:05:08.609952 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS,
Access Request (1), id: 0xd1 length: 65
14:05:08.609986 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS,
Access Request (1), id: 0xd2 length: 66
14:05:08.610018 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS,
Access Request (1), id: 0xd3 length: 66
14:05:08.610050 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS,
Access Request (1), id: 0xd4 length: 66
14:05:08.634849 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS,
Access Accept (2), id: 0xd0 length: 20
14:05:08.645602 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS,
Access Accept (2), id: 0xd1 length: 20
14:05:08.657563 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS,
Access Accept (2), id: 0xd2 length: 20
14:05:08.663509 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS,
Access Accept (2), id: 0xd3 length: 20
14:05:08.669488 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS,
Access Accept (2), id: 0xd4 length: 20
```

Perhitungan delay dilakukan dengan rumus berikut.

$$\text{Delay} = (t_{response}) - (t_{request})$$

Dengan :

$t_{request}$ = waktu keluarnya paket reqest

$t_{response}$ = waktu datangnya paket response

Kemudian delay rata-rata pada autentikasi bersamaan dilakukan dengan rumus sebagai berikut.

$$\text{Rata-rata delay} = \frac{delay_1 + delay_2 + \cdots + delay_n}{n}$$

$$= \frac{(t_{response1} - t_{request1}) + (t_{response2} - t_{request2}) + \cdots + (t_{responsen} - t_{requestn})}{n}$$

$$= \frac{t_{response1} + t_{response2} + \ldots + t_{responsen}}{n} - \frac{t_{request1} + t_{request2} + \cdots + t_{requestn}}{n}$$

Dengan :

$delay_n$= delay pada autentikasi ke-n

$t_{requestn}$ = waktu keluarnya paket request ke-n

$t_{responsen}$ = waktu datangnya paket response ke-n

Berikut ini adalah contoh perhitungan pada 5 autentikasi bersamaan.

Rata-rata delay :

$$= \frac{t_{response1} + t_{response2} + \ldots + t_{responsen}}{n} - \frac{t_{request1} + t_{request2} + \cdots + t_{requestn}}{n}$$

Dengan n = 5, maka:

Rata-rata delay :

$$= \frac{(8{,}634849 + 8{,}645602 + 8{,}657563 + 8{,}663509 + 8{,}669488)}{5} - \frac{(8{,}609303 + 8{,}609952 + 8{,}609986 + 8{,}610018 + 8{,}61005)}{5}$$

= 0,0443404 s

Dengan perhitungan diatas,maka didapatkan rata-rata delay pada 5 autentikasi bersamaan adalah 0,0443404 s.

Dengan menerapkan rumus yang sama pada pengujian dengan jumlah lain, maka didapatkan hasil perhitungannya pada tabel berikut.

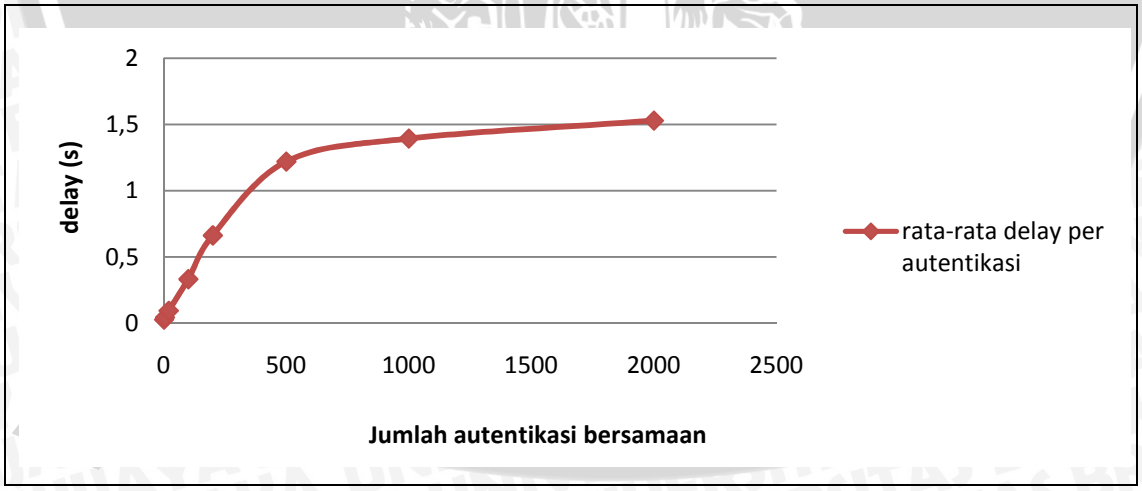
Tabel 5.4 Data hasil pengujian beberapa autentikasi bersamaan

| No. | Jumlah autentikasi | rata-rata delay per autentikasi (s) |
|---|---|---|
| 1 | 1 | 0,028722 |
| 2 | 5 | 0,443404 |
| 3 | 20 | 0,09577855 |
| 4 | 100 | 0,33304567 |
| 5 | 200 | 0,662980015 |
| 6 | 500 | 1,219661346 |
| 7 | 1000 | 1,39296607 |
| 8 | 2000 | 1,52774376 |

(Sumber: Pengujian)

### 5.2.3.3 Analisis Hasil

Dari Tabel 5.4 dapat dibuat grafik sebagai berikut.




Gambar 5.17. Grafik hubungan antara jumlah autentikasi bersamaan dengan rata-rata delay per autentikasi
(Sumber: pengujian dan perhitungan)

Grafik pada gambar 5.8 menunjukkan bahwa semakin besar jumlah autentikasi yang terjadi bersamaan, semakin besar rata-rata *delay* autentikasinya. Dengan kata lain semakin besar jumlah autentikasi yang terjadi bersamaan, semakin lambat proses autentikasinya. Pada jumlah autentikasi bersamaan lebih dari 500, semakin besar jumlah autentikasi yang terjadi bersamaan, rata-rata *delay* cenderung stabil.

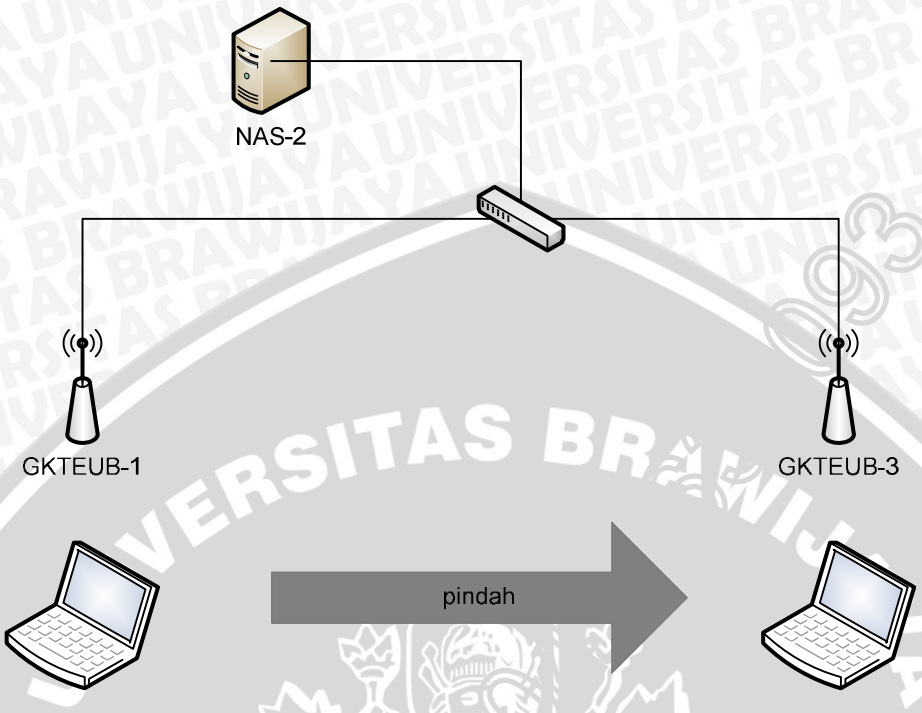
### 5.2.4 Menganalisa Proses *Handoff*

Tujuan pembahasan analisa proses *handoff* adalah untuk mengetahui bagaimana mekanisme *handoff* pada sistem yang telah diterapkan Radius. Proses *handoff* yang dianalisa pada pembahasan ini adalah proses handoff layer 2.

#### 5.2.4.1 Prosedur Pengujian

Prosedur pengujian analisa proses *handoff* adalah sebagai berikut:

- Sebuah perangkat *mobile* (MN)terkoneksi pada suatu *access point* (GKTEUB-1).
- Melakukan autentikasi Radius pada MN.
- Kemudian (MN) tersebut berpindah menjauhi jaungkauan (BSS) AP-1. MN berpindah menuju daerah jangkauan (BSS) *access point* lain (GKTEUB-3) yang memiliki BSS yang saling berpotongan dengan BSS GKTEUB-1 dan memiliki channel yang berbeda dengan BSS GKETUB-1 hingga MN meninggalkan BSS GKTUEB-3. Proses *handoff* tersebut diilustrasikan pada gambar 5.9.




Gambar 5.18 Ilustrasi proses *handoff*
(Sumber: Perancangan)

### 5.2.4.2 Hasil Pengujian

Dari pengujian yang telah dilakukan didapatkan hasil bahwa setelah MN berhasil terkoneksi dengan *access point* GKTEUB-3, user pada MN diharuskan melakukan autentikasi ulang untuk dapat tersambung dengan jaringan luar maupun internet.

### 5.2.4.3 Analisis Hasil

Handoff adalah proses perpindahan perangkat *mobile* koneksi dari suatu *access point* ke *access point* lain yang saling berpotongan daerah jangkauannya karena perpindahan posisi sebuah perangkat *mobile*. Pada saat perpindahan tersebut terdapat waktu dimana perangkat *mobile* tersebut kehilangan koneksi dengan *access point* awal Pada saat inilah terdapat tahapan-tahapan *handoff* seperti proses *scanning*, *authentication*, dan *reassociation*.

Setelah terkoneksi dengan *access point* tujuan (*authentication*), perangkat mobile tersebut harus melakukan *reassociation* atau autentikasi pada Radius karena *session* user pada Radius dianggap memutuskan koneksi fisik. Hal ini tercatat pada record *accounting* pada basis data Radius yang di dalamnya terdapat informasi bahwa user tersebut telah diberhentikan *session*nya dengan atribut Termination-Cause bernilai "Lost Carrier" dengan keterangan waktu berhentinya *session* sama dengan waktu dilakukannya proses *handoff*.

Berikut ini adalah penjelasan mengenai paket Accounting Request yang menghentikan *session* user (MN) yang sedang melakukan handoff tersebut.

- Paket Accounting-Request adalah paket yang dikirimkan oleh NAS kepada *server* Radius untuk memberikan informasi accounting pada server [RFC-286].
- Paket Accounting-Request tersebut memiliki atribut:
    - Acct-Status-Type dengan nilai "Stop" yang dapat diartikan permintaan pemberhentian proses *accounting* [RFC-358].
    - Termination-Cause dengan nilai "Lost-Carrier" yang berarti proses accounting dihentikan karena kehilangan koneksi fisik [RFC-358].

Karena proses reassociation dilakukan secara manual, maka perhitungan *latency* tidak dapat dilakukan karena *latency* dalam hal ini merupakan banyaknya waktu yang diperlukan dari proses *authentication* hingga *reassociation*.

# BAB VI PENUTUP

## 6.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan pada sistem, diambil kesimpulan sebagai berikut.

1. Sistem AAA dengan protokol Radius yang diujikan telah memenuhi kebutuhan fungsional sistem dengan *server* Radius dan *server* basis data terletak zona yang bukan merupakan jaringan internal. NAS pada rancangan ini juga berfungsi sebagai *end router* pada jaringan internal UB yang terdistribusi pada tiap-tiap segmen jaringan pada jaringan internal UB.
2. Proses autentikasi dan otorisasi pada Radius adalah sebagai berikut.
   - NAS mengirimkan paket Access-Request kepada *server* Radius. Paket tersebut berisi username dan password beserta atribut lain yang menyertainya.
   - *Server* Radius kemudian menerima paket tersebut dan mengolahnya dengan mencari *record* yang sesuai dengan atribut pada paket Access-Request pada *server* basis data.
   - Setelah *record* ditemukan, sebagian atribut otorisasi diolah *server* untuk diterapkan sebelum paket Access-Accept dikirim ke NAS. Sebagian atribut yang lain dikirim dari *server* ke NAS melalui paket Access-Accept.
3. Terjadi penurunan kualitas layanan jaringan (*transfer rate*) pada jaringan yang diterapkan Radius dibandingkan dengan jaringan yang belum diterapkan Radius yaitu sebesar 5,46%. Namun jumlah merupakan penurunan kualitas yang tidak signifikan sehingga dengan diterapkannya Radius dianggap tidak begitu mengganggu kualitas layanan jaringan yang telah ada.
4. Sistem mampu menangani beberapa autentikasi yang terjadi bersamaan. Namun performanya berkurang seiring dengan peningkatan jumlah autentikasi

yang terjadi bersamaan. Pada jumlah autentikasi bersamaan lebih dari 500, semakin besar jumlah autentikasi yang terjadi bersamaan, rata-rata delay autentikasi cenderung tetap.

5. Pada peristiwa *handoff* dengan *user* Radius, *user* diharuskan melakukan autentikasi ulang Radius pada saat setelah terkoneksi dengan *access point* tujuan untuk mendapatkan layanan jaringan karena Radius menganggap perpindahan *access point* sebagai terputusnya koneksi fisik, sehingga *session* dianggap berakhir oleh NAS. Karena proses autentikasi Radius pada *handoff* dilakukan manual, maka penghitungan *latency* tidak bisa dilakukan.

## 6.2 Saran

Dengan mencermati penelitian yang telah dilakukan, saran yang dapat diberikan untuk pengembangan lebih lanjut antara lain:

- Untuk mengeliminasi proses autentikasi ulang user secara manual pada peristiwa handoff, diperlukan IAPP (*Inter-Access Point Protocol*), protokol yang memungkinkan pertukaran informasi user *mobile* antar *access point*.
- Seiring dengan berkembangnya jaringan di Universitas Brawijaya, diperlukan penelitian mengenai performansi sistem AAA dengan protokol Radius dengan NAS berupa perangkat jaringan *dedicated* misal seperti Cisco.
- Pada skripsi ini digunakan protokol autentikasi PAP *(Password Authentication Protocol*). Jika menginginkan keamanan autentikasi yang lebih handal, dapat digunakan protokol autentikasi yang lebih aman seperti CHAP *(Challenge Handsake Authentication Protocol*) dan EAP-TLS *(Extensible Authentication Protocol-Transport Layer Security*).

# DAFTAR PUSTAKA


[ALL-06] Anonymous. 2006. 802.1X *White Paper*. Allied Telesyn.

[BLA-04] Blank, A. G. 2004. TCP/IP *Foundations*. Sybex. Alameda.

[BRO-07] Brown, E. L. 2007. 802.1X Port Based Authentication. Auerbach Publication. New York.

[DHO-07] Dhoto. 2007. Jaringan Komputer. PENS-ITS. Surabaya.

[DJU-07] Djuric, D. 2007. Opendiameter *Conformance Testing*. Research and Development Centre, Ericsson Nikola Tesla, Krapinska 45, HR-10002 Zagreb, Croatia.

[ERZ-05] Erzed, E., Adnan dan D. A. Ardy. 2005. Perancangan dan Implementasi Sistem Jaringan WLAN Berbasis Radius Server (Studi Kasus : WLAN STII I-TECH). Teknik Informatika STII I-Tech Jakarta. Jakarta.

[GAS-05] Gast, M. 2005. 802.11® *Wireless Networks The Definitive Guide*. O'Reilly. Sebastopol, CA.

[LAM-04] Lammle, T. 2005. CCNA™ : Cisco® *Certified Network Associate Study Guide*. Sybex. Alameda.

[MAN-04] Mansfield, N. 2004. Practical TCP/IP. Penerbit Andi. Yogyakarta.

[MAR-07] Martinovic, I. at all. 2007. *Measurement and Analysis of Handover Latencies in IEEE 802.11i Secured Networks*. Distributed Computer Systems Lab, University of Kaiserslautern. Germany.

[NAK-05] Nakhjiri, M. dan Madjid. 2005. *AAA and Network Security for Mobile Access*. Wiley. London.

[STI-05] Stiawan, D. 2005. Sistem Keamanan Komputer. PT. Elex Media Komputindo. Jakarta.

[STR-04] Strand, L. 2004. 802.1X Port−Based Authentication HOWTO. GNU Free Documentation License.

[WIK-08] Anonymous. 2008. http://en.wikipedia.org/wiki/AAA_protocol, tanggal akses 23 September 2008.

[WIK-08] Anonymous. 2008. http://id.wikipedia.org/wiki/TCP/IP, tanggal akses 25 September 2008.

[WIK-08] Anonymous. 2008. http://en.wikipedia.org/wiki/IEEE_802.11. tanggal akses 25 September 2008.

[ZHA-08] Zhang, L.J. dan S. Pierre. 2008. *Optimizing the Performance of Handoff Management in Wireless* LANs. IJCSNS, VOL. 8 No. 7 hal. 87-93.

[SZI-05] Szigeti, T. dan S. Hattingh. 2005. *End-to-End QoS Network Design*. Cisco Press. Indianapolis.

[MAN-09] Anonymous. 2009. http://linux.die.net/man/1/radclient, tanggal akses 1 Desember 2009.

[WIK-09] Anonymous. 2009. http://en.wikipedia.org/wiki/Ping, tanggal akses 1 Desember 2009.

[RFC-358] Congdon, Paul. 2005. http://tools.ietf.org/html/rfc3580, tanggal akses 4 Desember 2009.

[RFC-286] Rigney, Carl. 2004. http://tools.ietf.org/html/rfc2866, tanggal akses 4 Desember 2009.

[PRE-04] Pressman, Roger S. 2004. Software Engineering: A Practitioner's Approach. McGraw-Hill. New York.

[RFC-285] Rigney, Carl. 2004. http://tools.ietf.org/html/rfc2865, tanggal akses 15 Januari 2010.

[WIK-10] Anonymous. 2009. http://en.wikipedia.org/wiki/Login. tanggal akses 15 Januari 2010.

# LAMPIRAN A

# DATA HASIL PENGUJIAN PROSES AUTENTIKASI DAN OTORISASI

## Pengujian Autentikasi Berhasil

*Capture* Paket Pada Server Radius

```
root@radiusserver:~# tcpdump -i eth1 port 1812
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
12:57:41.846876 IP 172.18.3.20.46312 > radiusserver.local.radius: RADIUS, Access
Request (1), id: 0x00 length: 199
12:57:41.871328 IP radiusserver.local.radius > 172.18.3.20.46312: RADIUS, Access
Accept (2), id: 0x00 length: 32

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

*Capture* Paket Pada NAS (interface *uplink*)

```
[root@NAS-1 ~]# tcpdump -i eth0 port 1812
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:57:41.208521 IP 172.18.3.20.46312 > 172.18.3.31.radius: RADIUS, Access Request
(1), id: 0x00 length: 199
12:57:41.233127 IP 172.18.3.31.radius > 172.18.3.20.46312: RADIUS, Access Accept
(2), id: 0x00 length: 32

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Keluaran *Debug* Freeradius

```
Ready to process requests.
rad_recv: Access-Request packet from host 172.18.3.20 port 48994, id=0, length=199
        User-Name = "0410630046"
        User-Password = "koko"
        NAS-IP-Address = 0.0.0.0
        Service-Type = Login-User
        Framed-IP-Address = 172.17.91.3
        Calling-Station-Id = "00-16-D4-8C-62-59"
        Called-Station-Id = "00-E0-1C-3C-1B-F7"
        NAS-Identifier = "nas01"
        Acct-Session-Id = "4ac9c08700000000"
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 0
        Message-Authenticator = 0xb9f68916d9c64929fe0f723747fbb39c
        WISPr-Logoff-URL = "http://172.17.91.1:3990/logoff"
```

```
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[unix] returns notfound
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
rlm_sql (sql): Reserving sql socket id: 4
[sql]   expand: SELECT id, username, attribute, value, op          FROM radcheck
WHERE username = '%{SQL-User-Name}'          ORDER BY id -> SELECT id, username,
attribute, value, op          FROM radcheck          WHERE username =
'0410630046'          ORDER BY id
[sql] User found in radcheck table
[sql]   expand: SELECT id, username, attribute, value, op          FROM radreply
WHERE username = '%{SQL-User-Name}'          ORDER BY id -> SELECT id, username,
attribute, value, op          FROM radreply          WHERE username =
'0410630046'          ORDER BY id
[sql]   expand: SELECT groupname          FROM usergroup          WHERE username
= '%{SQL-User-Name}'          ORDER BY priority -> SELECT groupname          FROM
usergroup          WHERE username = '0410630046'          ORDER BY priority
[sql]   expand: SELECT id, groupname, attribute,          Value, op          FROM
radgroupcheck          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          Value, op          FROM radgroupcheck
WHERE groupname = 'packet'          ORDER BY id
[sql] User found in group packet
[sql]   expand: SELECT id, groupname, attribute,          value, op          FROM
radgroupreply          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          value, op          FROM radgroupreply
WHERE groupname = 'packet'          ORDER BY id
rlm_sql (sql): Released sql socket id: 4
++[sql] returns ok
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[noresetcounter] returns noop
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[octetslimit] returns noop
++[expiration] returns noop
++[logintime] returns noop
++[pap] returns updated
Found Auth-Type = PAP
+- entering group PAP {...}
[pap] login attempt with password "koko"
[pap] Using clear text password "koko"
[pap] User authenticated successfully
++[pap] returns ok
+- entering group session {...}
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
```

```
[sql]   expand: SELECT COUNT(*)                                  #FROM radacct
#WHERE username = '%{SQL-User-Name}'                                  #AND acctstoptime
IS NULL -> SELECT COUNT(*)                                  #FROM radacct
#WHERE username = '0410630046'                                  #AND acctstoptime IS
NULL
rlm_sql (sql): Reserving sql socket id: 3
[sql]   expand: SELECT radacctid, acctsessionid, username,
nasipaddress, nasportid, framedipaddress,
callingstationid, framedprotocol                                  FROM radacct
WHERE username = '%{SQL-User-Name}'                                  AND acctstoptime
IS NULL -> SELECT radacctid, acctsessionid, username,
nasipaddress, nasportid, framedipaddress,
callingstationid, framedprotocol                                  FROM radacct
WHERE username = '0410630046'                                  AND acctstoptime IS
NULL
rlm_sql (sql): Released sql socket id: 3
++[sql] returns ok
+- entering group post-auth {...}
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
[sql]   expand: %{User-Password} -> koko
[sql]   expand: INSERT INTO radpostauth                                  (username, pass,
reply, authdate)                                  VALUES (
'%{User-Name}',                                  '%{%{User-Password}:-%{Chap-Password}}',
'%{reply:Packet-Type}', '%S') -> INSERT INTO radpostauth
(username, pass, reply, authdate)                                  VALUES (
'0410630046',                                  'koko',                                  'Access-
Accept', '2009-10-05 16:47:16')
rlm_sql (sql) in sql_postauth: query is INSERT INTO radpostauth
(username, pass, reply, authdate)                                  VALUES (
'0410630046',                                  'koko',                                  'Access-
Accept', '2009-10-05 16:47:16')
rlm_sql (sql): Reserving sql socket id: 2
rlm_sql (sql): Released sql socket id: 2
++[sql] returns ok
++[exec] returns noop
Sending Access-Accept of id 0 to 172.18.3.20 port 48994
        Idle-Timeout := 300
        Service-Type = Login-User
Finished request 0.
Going to the next request
Waking up in 4.9 seconds.
rad_recv: Accounting-Request packet from host 172.18.3.20 port 55067, id=24,
length=135
        Acct-Status-Type = Start
        User-Name = "0410630046"
        Calling-Station-Id = "00-16-D4-8C-62-59"
        Called-Station-Id = "00-E0-1C-3C-1B-F7"
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 0
        NAS-Port-Id = "00000000"
        NAS-IP-Address = 0.0.0.0
        NAS-Identifier = "nas01"
        Framed-IP-Address = 172.17.91.3
```

```
        Acct-Session-Id = "4ac9c08700000000"
+- entering group preacct {...}
++[preprocess] returns ok
[acct_unique] Hashing 'NAS-Port = 0,Client-IP-Address = 172.18.3.20,NAS-IP-Address
= 0.0.0.0,Acct-Session-Id = "4ac9c08700000000",User-Name = "0410630046"'
[acct_unique] Acct-Unique-Session-ID = "f5f9edfc3f9c9497".
++[acct_unique] returns ok
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
++[files] returns noop
+- entering group accounting {...}
[detail]        expand: /var/log/freeradius/radacct/%{Client-IP-Address}/detail-
%Y%m%d -> /var/log/freeradius/radacct/172.18.3.20/detail-20091005
[detail] /var/log/freeradius/radacct/%{Client-IP-Address}/detail-%Y%m%d expands to
/var/log/freeradius/radacct/172.18.3.20/detail-20091005
[detail]        expand: %t -> Mon Oct  5 16:47:16 2009
++[detail] returns ok
++[unix] returns ok
[radutmp]       expand: /var/log/freeradius/radutmp -> /var/log/freeradius/radutmp
[radutmp]       expand: %{User-Name} -> 0410630046
++[radutmp] returns ok
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
[sql]   expand: %{Acct-Delay-Time} ->
[sql]   expand:           INSERT INTO radacct          (acctsessionid,
acctuniqueid,     username,              realm,          nasipaddress,
nasportid,              nasporttype,      acctstarttime,    acctstoptime,
acctsessiontime,  acctauthentic,    connectinfo_start,
connectinfo_stop, acctinputoctets,  acctoutputoctets,             calledstationid,
callingstationid, acctterminatecause,           servicetype,
framedprotocol,   framedipaddress,          acctstartdelay,   acctstopdelay,
xascendsessionsvrkey)          VALUES          ('%{Acct-Session-Id}', '%{Acct-
Unique-Session-Id}',          '%{SQL-User-Name}',             '%{Realm}',
'%{NAS-IP-Address}', '%{NAS-Port}',             '%{NAS-Port-Type}', '%S', NULL,
'0', '%{Acct-Authentic}', '%{Connect-Info}',          '', '0', '0',
'%{Called-Station-Id}', '%{Calling-Station-Id}', '',              '%{Service-
Type}', '%{Framed-Protocol}', '%{Framed-IP-Address}',
rlm_sql (sql): Reserving sql socket id: 1
rlm_sql (sql): Released sql socket id: 1
++[sql] returns ok
[attr_filter.accounting_response]       expand: %{User-Name} -> 0410630046
 attr_filter: Matched entry DEFAULT at line 12
++[attr_filter.accounting_response] returns updated
Sending Accounting-Response of id 24 to 172.18.3.20 port 55067
Finished request 1.
Cleaning up request 1 ID 24 with timestamp +16
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 0 ID 0 with timestamp +16
Ready to process requests.
```

**Pengujian Autentikasi Gagal**

*Capture* Paket Pada Server Radius

```
root@radiusserver:~# tcpdump -i eth1 port 1812
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
16:49:47.397062 IP 172.18.3.20.46592 > radiusserver.local.radius: RADIUS, Access
Request (1), id: 0x00 length: 199
16:49:48.402364 IP radiusserver.local.radius > 172.18.3.20.46592: RADIUS, Access
Reject (3), id: 0x00 length: 20

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

*Capture* Paket Pada NAS (interface *uplink*)

```
[root@NAS-1 ~]# tcpdump -i eth0 port 1812 or port 1813
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
16:49:40.479658 IP 172.18.3.20.46592 > 172.18.3.31.radius: RADIUS, Access Request
(1), id: 0x00 length: 199
16:49:41.484895 IP 172.18.3.31.radius > 172.18.3.20.46592: RADIUS, Access Reject
(3), id: 0x00 length: 20

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Hasil Keluaran *Debug* Freeradius

```
Ready to process requests.
rad_recv: Access-Request packet from host 172.18.3.20 port 46592, id=0, length=199
        User-Name = "0410630046"
        User-Password = "koki"
        NAS-IP-Address = 0.0.0.0
        Service-Type = Login-User
        Framed-IP-Address = 172.17.91.3
        Calling-Station-Id = "00-16-D4-8C-62-59"
        Called-Station-Id = "00-E0-1C-3C-1B-F7"
        NAS-Identifier = "nas01"
        Acct-Session-Id = "4ac9c0fc00000000"
        NAS-Port-Type = Wireless-802.11
        NAS-Port = 0
        Message-Authenticator = 0xf153786f68510cdadf738c4a0fbd56dc
        WISPr-Logoff-URL = "http://172.17.91.1:3990/logoff"
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
[suffix] No '@' in User-Name = "0410630046", looking up realm NULL
```

```
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[unix] returns notfound
[sql]   expand: %{User-Name} -> 0410630046
[sql] sql_set_user escaped user --> '0410630046'
rlm_sql (sql): Reserving sql socket id: 4
[sql]   expand: SELECT id, username, attribute, value, op          FROM radcheck
WHERE username = '%{SQL-User-Name}'           ORDER BY id -> SELECT id, username,
attribute, value, op        FROM radcheck           WHERE username =
'0410630046'          ORDER BY id
[sql] User found in radcheck table
[sql]   expand: SELECT id, username, attribute, value, op           FROM radreply
WHERE username = '%{SQL-User-Name}'           ORDER BY id -> SELECT id, username,
attribute, value, op         FROM radreply           WHERE username =
'0410630046'          ORDER BY id
[sql]   expand: SELECT groupname           FROM usergroup          WHERE username
= '%{SQL-User-Name}'          ORDER BY priority -> SELECT groupname          FROM
usergroup         WHERE username = '0410630046'          ORDER BY priority
[sql]   expand: SELECT id, groupname, attribute,          Value, op          FROM
radgroupcheck          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          Value, op          FROM radgroupcheck
WHERE groupname = 'packet'          ORDER BY id
[sql] User found in group packet
[sql]   expand: SELECT id, groupname, attribute,          value, op          FROM
radgroupreply          WHERE groupname = '%{Sql-Group}'          ORDER BY id ->
SELECT id, groupname, attribute,          value, op          FROM radgroupreply
WHERE groupname = 'packet'          ORDER BY id
rlm_sql (sql): Released sql socket id: 4
++[sql] returns ok
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[noresetcounter] returns noop
rlm_sqlcounter: Entering module authorize code
rlm_sqlcounter: Could not find Check item value pair
++[octetslimit] returns noop
++[expiration] returns noop
++[logintime] returns noop
++[pap] returns updated
Found Auth-Type = PAP
+- entering group PAP {...}
[pap] login attempt with password "koki"
[pap] Using clear text password "koko"
[pap] Passwords don't match
++[pap] returns reject
Failed to authenticate the user.
Using Post-Auth-Type Reject
+- entering group REJECT {...}
[attr_filter.access_reject]     expand: %{User-Name} -> 0410630046
 attr_filter: Matched entry DEFAULT at line 11
++[attr_filter.access_reject] returns updated
Delaying reject of request 0 for 1 seconds
Going to the next request
```

```
Waking up in 0.9 seconds.
Sending delayed reject for request 0
Sending Access-Reject of id 0 to 172.18.3.20 port 46592
Waking up in 4.9 seconds.
Cleaning up request 0 ID 0 with timestamp +29
Ready to process requests.
```

## LAMPIRAN B

## DATA HASIL PENGUJIAN *TRANSFER RATE* SEBELUM DAN SESUDAH AUTENTIKASI

Hasil Keluaran Iperf Sebelum Autentikasi

```
C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1083 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  91.7 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1084 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.5 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1085 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.3 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1086 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.3 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1087 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
```

```
[1912]  0.0-10.0 sec   110 MBytes  91.9 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1088 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.1 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1089 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.2 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1090 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   109 MBytes  91.1 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1091 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   111 MBytes  92.6 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1093 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   110 MBytes  92.2 Mbits/sec
```

Hasil Keluaran Iperf Setelah Atutentikasi

```
C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
```

```
------------------------------------------------------------
[1912] local 172.17.91.130 port 1374 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   104 MBytes  87.3 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1375 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   104 MBytes  86.9 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1376 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   104 MBytes  86.9 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1377 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   105 MBytes  87.7 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1378 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   103 MBytes  86.5 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1381 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   105 MBytes  87.6 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
```

```
------------------------------------------------------------
[1912] local 172.17.91.130 port 1382 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   103 MBytes  86.5 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1383 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   103 MBytes  86.4 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1384 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   103 MBytes  86.4 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1385 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   103 MBytes  86.4 Mbits/sec

C:\Documents and Settings\Winda.MOVINGHOUSE>iperf -c 172.18.3.31
------------------------------------------------------------
Client connecting to 172.18.3.31, TCP port 5001
TCP window size: 8.00 KByte (default)
------------------------------------------------------------
[1912] local 172.17.91.130 port 1386 connected with 172.18.3.31 port 5001
[ ID] Interval       Transfer     Bandwidth
[1912]  0.0-10.0 sec   107 MBytes  89.7 Mbits/sec
```

# LAMPIRAN C

# DATA HASIL PENGUJIAN DELAY AUTENTIKASI PADA BEBERAPA VARIASI JUMLAH AUTENTIKASI BERSAMAAN

## 1 Autentikasi

Hasil Keluaran Radclient

```
Sending Access-Request of id 253 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=253, length=20
```

Hasil *Capture* Paket Pada Server Radius

```
10:49:21.814695 IP 172.18.3.218.42108 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xfd length: 50
10:49:21.844859 IP 172.18.3.31.radius > 172.18.3.218.42108: RADIUS, Access Accept (2), id:
0xfd length: 20
```

## 5 Autentikasi

Hasil Keluaran Radclient

```
Sending Access-Request of id 208 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
Sending Access-Request of id 209 to 172.18.3.31 port 1812
        User-Name = "pengujian"
        User-Password = "password pengujian"
Sending Access-Request of id 210 to 172.18.3.31 port 1812
        User-Name = "pengujian2"
        User-Password = "password pengujian2"
Sending Access-Request of id 211 to 172.18.3.31 port 1812
        User-Name = "pengujian3"
        User-Password = "password pengujian3"
Sending Access-Request of id 212 to 172.18.3.31 port 1812
        User-Name = "pengujian4"
        User-Password = "password pengujian4"
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=208, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=209, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=210, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=211, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=212, length=20
```

Hasil *Capture* Paket Pada Server Radius

```
14:05:08.609303 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xd0 length: 50
14:05:08.609952 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xd1 length: 65
14:05:08.609986 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xd2 length: 66
14:05:08.610018 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xd3 length: 66
14:05:08.610050 IP 172.18.3.218.52876 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
```

```
0xd4 length: 66
14:05:08.634849 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS, Access Accept (2), id:
0xd0 length: 20
14:05:08.645602 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS, Access Accept (2), id:
0xd1 length: 20
14:05:08.657563 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS, Access Accept (2), id:
0xd2 length: 20
14:05:08.663509 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS, Access Accept (2), id:
0xd3 length: 20
14:05:08.669488 IP 172.18.3.31.radius > 172.18.3.218.52876: RADIUS, Access Accept (2), id:
0xd4 length: 20
```

## 20 Autentikasi

Hasil Keluaran Radclient

```
Sending Access-Request of id 227 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
Sending Access-Request of id 228 to 172.18.3.31 port 1812
        User-Name = "pengujian"
        User-Password = "password pengujian"
Sending Access-Request of id 229 to 172.18.3.31 port 1812
        User-Name = "pengujian2"
        User-Password = "password pengujian2"
Sending Access-Request of id 230 to 172.18.3.31 port 1812
        User-Name = "pengujian3"
        User-Password = "password pengujian3"
Sending Access-Request of id 231 to 172.18.3.31 port 1812
        User-Name = "pengujian4"
        User-Password = "password pengujian4"
Sending Access-Request of id 232 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
Sending Access-Request of id 233 to 172.18.3.31 port 1812
        User-Name = "pengujian"
        User-Password = "password pengujian"
Sending Access-Request of id 234 to 172.18.3.31 port 1812
        User-Name = "pengujian2"
        User-Password = "password pengujian2"
Sending Access-Request of id 235 to 172.18.3.31 port 1812
        User-Name = "pengujian3"
        User-Password = "password pengujian3"
Sending Access-Request of id 236 to 172.18.3.31 port 1812
        User-Name = "pengujian4"
        User-Password = "password pengujian4"
Sending Access-Request of id 237 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
Sending Access-Request of id 238 to 172.18.3.31 port 1812
        User-Name = "pengujian"
        User-Password = "password pengujian"
Sending Access-Request of id 239 to 172.18.3.31 port 1812
        User-Name = "pengujian2"
        User-Password = "password pengujian2"
Sending Access-Request of id 240 to 172.18.3.31 port 1812
        User-Name = "pengujian3"
        User-Password = "password pengujian3"
Sending Access-Request of id 241 to 172.18.3.31 port 1812
        User-Name = "pengujian4"
        User-Password = "password pengujian4"
Sending Access-Request of id 242 to 172.18.3.31 port 1812
        User-Name = "0410630093"
        User-Password = "winda"
Sending Access-Request of id 243 to 172.18.3.31 port 1812
        User-Name = "pengujian"
        User-Password = "password pengujian"
```

```
Sending Access-Request of id 244 to 172.18.3.31 port 1812
        User-Name = "pengujian2"
        User-Password = "password pengujian2"
Sending Access-Request of id 245 to 172.18.3.31 port 1812
        User-Name = "pengujian3"
        User-Password = "password pengujian3"
Sending Access-Request of id 246 to 172.18.3.31 port 1812
        User-Name = "pengujian4"
        User-Password = "password pengujian4"
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=227, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=228, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=229, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=230, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=231, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=232, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=233, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=234, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=235, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=236, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=237, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=238, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=239, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=240, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=241, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=242, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=243, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=244, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=245, length=20
rad_recv: Access-Accept packet from host 172.18.3.31:1812, id=246, length=20
```

Hasil *Capture* Paket Pada Server Radius

```
10:59:36.570816 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe3 length: 50
10:59:36.573898 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe4 length: 65
10:59:36.573908 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe5 length: 66
10:59:36.573916 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe6 length: 66
10:59:36.573922 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe7 length: 66
10:59:36.573929 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe8 length: 50
10:59:36.573935 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xe9 length: 65
10:59:36.573942 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xea length: 66
10:59:36.573949 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xeb length: 66
10:59:36.573955 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xec length: 66
10:59:36.573962 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xed length: 50
10:59:36.573968 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xee length: 65
10:59:36.573975 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xef length: 66
10:59:36.573981 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf0 length: 66
10:59:36.573988 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf1 length: 66
10:59:36.573995 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf2 length: 50
10:59:36.574001 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf3 length: 65
10:59:36.574008 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf4 length: 66
```

```
10:59:36.574014 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf5 length: 66
10:59:36.574021 IP 172.18.3.218.52186 > 172.18.3.31.radius: RADIUS, Access Request (1), id:
0xf6 length: 66
10:59:36.604422 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe3 length: 20
10:59:36.619012 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe4 length: 20
10:59:36.624978 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe5 length: 20
10:59:36.630942 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe6 length: 20
10:59:36.637209 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe7 length: 20
10:59:36.643152 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe8 length: 20
10:59:36.649139 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xe9 length: 20
10:59:36.655116 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xea length: 20
10:59:36.661081 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xeb length: 20
10:59:36.667079 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xec length: 20
10:59:36.673033 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xed length: 20
10:59:36.679010 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xee length: 20
10:59:36.684987 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xef length: 20
10:59:36.690952 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf0 length: 20
10:59:36.697169 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf1 length: 20
10:59:36.702916 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf2 length: 20
10:59:36.708893 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf3 length: 20
10:59:36.714920 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf4 length: 20
10:59:36.720837 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf5 length: 20
10:59:36.726807 IP 172.18.3.31.radius > 172.18.3.218.52186: RADIUS, Access Accept (2), id:
0xf6 length: 20
```

# LAMPIRAN D

## KONFIGURASI ESENSIAL PADA SERVER RADIUS DAN NAS

### Konfigurasi Server Radius

/etc/freeradius/sites-enabled/default

```
authorize {
        chap
        mschap
        suffix
        eap {
                ok = return
        }
        unix
        sql
        noresetcounter
        octetslimit
        expiration
        logintime
        pap
}


authenticate {
                pap
        }
        Auth-Type CHAP {
                chap
        }
        Auth-Type MS-CHAP {
                mschap
        }
        unix
        eap
}

preacct {
        preprocess
        suffix
        files
}

accounting {
        detail
        radutmp
        sql
        attr_filter.accounting_response
```

```
}

session {
        sql
}
post-auth {
        sql
        exec
        Post-Auth-Type REJECT {
                attr_filter.access_reject
        }
}
post-proxy {
        eap
}
```

/etc/freeradius/clients.conf

```
client localhost {
        ipaddr = 127.0.0.1
        secret          = radiussecret
        require_message_authenticator = no
        nastype     = other
}
client 172.17.67.62 {
        secret          = radiussecret
        shortname       = RisTIE
}
client 172.17.67.67 {
        secret          = radiussecret
        shortname       = gedungkuliah
}
```

/etc/freeradius/sql.conf

```
sql {
        database = "mysql"
        driver = "rlm_sql_${database}"
        server = "172.18.3.197"
        login = "root"
        password = "joaneh2"
        radius_db = "easyhotspot"
        acct_table1 = "radacct"
        # Allow for storing data after authentication
        postauth_table = "radpostauth"
        authcheck_table = "radcheck"
        authreply_table = "radreply"
        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"
        read_groups = yes
        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql
```

```
        num_sql_socks = 5
        connect_failure_retry_delay = 60  lifetime = 0
        max_queries = 0
        nas_table = "nas"
        $INCLUDE sql/${database}/dialup.conf
}
```

**Konfigurasi NAS**

/etc/chilli.conf

```
net             172.17.91.128/25
dynip           172.17.91.128/25
dns1            202.162.208.99
dns2            202.162.208.100
domain          skripsiTEUB.brawijaya.ac.id
#radiuslisten 172.18.3.31
radiusserver1 172.18.3.31
radiusserver2 172.18.3.31
radiussecret  radiussecret
dhcpif        eth2
uamserver     https://172.17.91.129/cgi-bin/hotspotlogin.cgi
uamsecret     uamsecret
uamhomepage   http://172.17.91.129/login/
uamlisten     172.17.91.129
uamallowed    172.17.91.129
```

Script halaman sebelum login (/var/www/html/login/index.html)

```
<html>
<head>
</head>
<body>
      <a href=”http://192.168.1.1:3990/prelogin/”>login disini</a>
</body>
</html>
```

/var/www/cgi-bin/hotspotlogin.cgi

- Pada baris ke 27:

```
$uamsecret = uamsecret
```

- Pada baris ke 31:

```
$userpassword = 1
```

Firewall pada NAS

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1497:155415]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i eth0 -p tcp -m tcp --dport 22 --tcp-flags FIN,SYN,RST,ACK SYN -j ACCEPT
-A INPUT -i eth0 -j REJECT --reject-with icmp-port-unreachable
-A INPUT -i eth1 -j DROP
-A INPUT -i eth2 -j DROP
-A INPUT -p tcp -m tcp --dport 80 --tcp-flags FIN,SYN,RST,ACK SYN -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 --tcp-flags FIN,SYN,RST,ACK SYN -j ACCEPT
-A INPUT -p tcp -m tcp --dport 3990 --tcp-flags FIN,SYN,RST,ACK SYN -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A FORWARD -i eth1 -j DROP
-A FORWARD -o eth1 -j DROP
-A FORWARD -i eth2 -j DROP
-A FORWARD -o eth2 -j DROP
COMMIT
```