BAB II
LANDASAN TEORI

2.1 Education Roaming (*eduroam*)

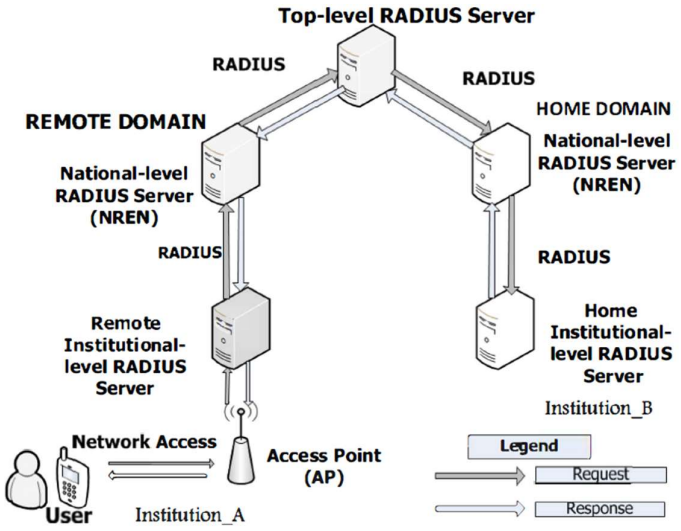
Education Roaming (*eduroam*) adalah layanan *roaming* WLAN antar lembaga akademis dan lembaga penelitian di seluruh dunia. Eduroam memberikan akses layanan internet yang aman kepada pengguna dari sebuah institusi yang berpartisipasi di dalam layanan eduroam ketika pengguna tersebut berkunjung ke institusi lain yang juga berpartisipasi di dalam layanan eduroam (Durnford, 2013). Pengguna hanya perlu menggunakan kredensial dari institusi asalnya untuk proses otentikasi. Untuk proses otorisasi dilakukan oleh institusi tempat pengguna tersebut berkunjung. Setelah mendapatkan izin, maka pengguna tersebut akan mendapatkan ip dan dapat melakukan akses internet melalui *firewall* dan *server proxy* dari institusi tempat ia berkunjung (Tekeni, Thomson, & Botha, 2014).

Inisiatif eduroam awalnya dimulai pada tahun 2002. TF-Mobility Klaas Wierenga dari SURFnet yang merupakan bagian dari Trans European Research and Education Nerworking Association (TERENA), yang saat ini berganti nama menjadi GÉANT, menyatukan sebuah infrastruktur berbasis RADIUS dengan protokol IEEE 802.1x untuk akses internet *roaming* antar institusi di Eropa. Tujuan dari berdirinya TERENA adalah untuk mempromosikan dan berpartisipasi dalam pengembangan infrastruktur informasi dan telekomunikasi internasional berkualitas tinggi untuk kepentingan penelitian dan pendidikan. Langkah apapun yang diperlukan akan diambil untuk menunjukkan bahwa infrastuktur yang akan di kembangkan didasarkan pada standar yang terbuka dan menggunakan teknologi paling canggih yang tersedia (Olesen, 2003). Selama masa pengembangannya, pada tahun 2003 sudah mulai banyak institusi di Eropa yang mulai bergabung. Hingga pada tahun 2004, Australia menjadi negara non-Eropa pertama yang terhubung di dalam layanan eduroam (Tekeni et al., 2014). Sampai saat ini, sudah terdapat 89 operator *roaming* yang tersebar di seluruh dunia yang memberikan akses layanan eduroam.

Infrastruktur eduroam dibuat berdasarkan hierarki RADIUS *proxy server* yang teroganisir dan protokol IEEE 802.1x. Inisiatif ini menghasilkan penggunaan tiga tingkatan level dari RADIUS *proxy server* yang dinamakan: *Top-level server*, *National-level server, Institutional-level server*. *Top-level server* bertugas sebagai jembatan antar *National-level*

server untuk komunikasi global, sementara *National-level server* bertanggung jawab untuk menghubungkan antar institusi dalam negeri. Institusi yang ingin bergabung ke dalam layanan eduroam harus menghubungkan layanannya ke *National-level server* dan mendedikasikan sebuah *server* untuk eduroam. Setiap institusi penyedia layanan eduroam memiliki 2 peranan penting yakni sebagai *Identity Provider* (IdP) atau nama lainnya institusi asal, dan *Service Provider* (SP) atau nama lainnya institusi yang dikunjungi. IdP berperan pada proses otentikasi dan SP berperan pada proses otorisasi (Olesen, 2003).

Pada gambar 2.1 menjelaskan tentang pengangkutan permintaan otentikasi pengguna dari SP menuju IdP dan sebaliknya, dengan begitu maka sebuah sistem *server* RADIUS di seluruh dunia dibuat. Biasanya setiap IdP memasang sebuah *server* RADIUS yang terhubung ke database lokal pengguna. *Server* RADIUS ini terhubung ke *server* RADIUS tingkat nasional dan juga terhubung ke *server* RADIUS lain secara dinamis. Dengan adanya format penamaan "user@realm", *realm* diartikan sebagai alamat asal pengguna. Informasi *realm* tersebut digunakan oleh *server* RADIUS untuk merutekan permintaan ke *server* RADIUS berikutnya.



Gambar 2.1 Proses Otentikasi dan Otorisasi Layanan Eduroam

Sumber: (Tekeni et al., 2014)

Untuk mentransfer informasi otentikasi pengguna secara aman di seluruh infrastruktur RADIUS ke IdP, dan untuk mencegah pengguna lain melakukan pencurian informasi setelah proses otentikasi berhasil, maka SP harus memasang protokol standar IEEE 802.1X yang mencakup penggunaan *Extensible Authentication Protocol* (EAP) pada *access point* atau

switch. EAP adalah wadah yang membawa data otentikasi selama perjalanan melewati *server* RADIUS. Terdapat banyak tipe EAP yang dapat dipilih oleh sebuah IdP, antara lain (Durnford, 2013):

a. *Protected* EAP (PEAP) adalah sebuah protokol Microsoft yang menetapkan sebuah TLS *tunnel*, dan mengirim kredensial dalam bentuk MS-CHAPv2
b. *Tunneled* TLS (TTLS) adalah sebuah protokol IETF yang menetapkan sebuah TLS *tunnel*, dan mengirimkan kredensial dalam bentuk format konfigurasi yang beragam
c. TLS (*Transport Layer Security*) adalah sebuah IETF protokol yang mengotentikasi pengguna dan IdP dengan 2 sertifikat X.509
d. *Flexible Authentication via Secure Tunneling* (FAST) adalah sebuah protokol Cisco yang menetapkan sebuah TLS *tunnel*, dan mengirimkan kredensial dengan cara yang khusus.

RADIUS mengangkut nama pengguna ke dalam atribut User-Name, yang terlihat dalam bentuk *cleartext* ke semua perangkat penghubung di dalam perjalanannya. Beberapa metode EAP memungkinkan untuk menempatkan atribut User-Name yang berbeda ke dalam paket RADIUS. Dalam hal ini, istilah-istilah berikut digunakan:

a. *Outer identity*

   Atribut User-Name dalam paket RADIUS yang dapat dilihat oleh semua perangkat penghubung
b. *Inner identity*

   Kredensial pengguna yang sebenarnya. Atribut ini hanya terlihat oleh pengguna dan IdP

Hampir semua jenis EAP mendukung penggunaan *anonymous outer identity.* Penggunaan utama dari *outer identity* adalah untuk perlindungan privasi yang lebih baik bagi pengguna. Pengguna yang dikonfigurasi menggunakan *outer identity* dapat menyembunyikan identitas aslinya dari SP. Sebagai gantinya, identitas pengguna akan diganti dengan *dummy value*.

Informasi pada *inner identity* harus sepenuhnya akurat, karena akan digunakan untuk mengotentikasi pengguna. Tidak harus mengandung simbol @ karena nama pengguna itu bersifat lokal dan hanya dapat dilihat oleh IdP. *Outer identity* dapat ditulis dalam bentuk “anonymous@restena.lu” dan *inner identity* dapat ditulis dalam bentuk “stefan.winter” (Durnford, 2013). Untuk permintaan *routing* eduroam, bagian “@restena.lu” dari *outer*

identity digunakan untuk mengarahkan permintaan ke *realm* restena.lu dan membangun sebuah *tunnel* yang aman, sedangkan identitas asli di dalam *tunnel* yang dicari dalam database pengguna adalah “stefan.winter”.

Penggunaan EAP pada layanan eduroam harus memungkinkan otentikasi timbal balik yaitu pengguna dapat memverifikasi bahwa dia terhubung ke IdP-nya di mana pun pengguna berada. Selain itu layanan eduroam juga harus memungkinkan proses enkripsi kredensial yang digunakan (contoh: hanya pengguna dan IdP-nya yang dapat melihat pertukaran kredensial).

Pemilihan metode EAP dalam penerapannya pada layanan eduroam bergantung pada beberapa faktor, yaitu: kemampuan *backend* manajemen identitas pengguna, dan jenis perangkat apa yang ingin didukung. Metode EAP-TLS dapat dipilih jika *backend* manajemen identitas mendukung sertifikat klien X.509. Selain itu, metode EAP yang lain seperti EAP-TTLS, PEAP, EAP-FAST, EAP-PWD dapat digunakan apabila *backend* manajemen identitas menyimpan *password* dalam bentuk *cleartext* atau enkripsi NT-Hash.

Komparasi mengenai tipe metode EAP akan ditampilkan pada tabel 2.1 Komparasi tersebut akan dibandingkan berdasarkan fitur dan manfaat dari masing-masing metode EAP.

Tabel 2.1 Tabel Komparasi Tipe Metode EAP

Fitur/Manfaat	**TLS**	**TTLS**	**PEAP**	**FAST**
Kebutuhan sertifikat pada sisi klien	ya	tidak	tidak	tidak
Kebutuhan sertifikat pada sisi *server*	ya	tidak	ya	tidak
Manajemen WEP *key*	ya	ya	ya	ya
Atribut otentikasi	timbal balik	timbal balik	timbal balik	timbal balik
Wi-Fi Security	Sangat baik	baik	baik	baik
Tingkat kesulitan pemasangan	Sulit (pemasangan sertifikat)	menengah	menengah	Menengah

Metode EAP-TLS dianggap sebagai metode otentikasi yang paling aman. Hal ini dibuktikan dengan adanya pemasangan sertifikat pada sisi *server* dan klien. Pemasangan sertifikat ini bertujuan untuk proses otentikasi yang dilakukan pada perangkat yang

digunakan oleh pengguna terhadap layanan eduroam. Untuk dapat meneruskan permintaan otentikasi, maka pengguna harus meminta konfigurasi *sertifikat* pada perangkat yang akan digunakan untuk terhubung ke layanan eduroam. Institusi yang ingin menggunakan metode ini harus menyediakan *Public Key Infrastructure* (PKI) untuk dapat mengotentikasi antara klien dan *server*.

Metode EAP lainnya seperti TTLS, PEAP dan FAST meskipun tidak sebaik TLS namun metode ini juga menjanjikan keamanan untuk klien. Penggunaan 3 metode diatas merupakan solusi untuk menghindari penggunaan PKI, karena hal tersebut dapat mengurangi biaya implementasi secara keseluruhan dan memudahkan proses manajemen jaringan maupun permasalahan administrasi (penerbitan sertifikat).

Layanan eduroam memberikan akademisi maupun peneliti akses internet aman tanpa biaya. Jaringan layanan eduroam menggunakan metode enkripsi *end-to-end* untuk mencegah bocornya informasi *traffic* yang dilakukan oleh pengguna. Selain itu, eduroam dengan protokol 802.1x juga menjanjikan keamanan kredensial pengguna ketika melakukan koneksi di institusi lain. Layanan eduroam menghilangkan kebutuhan untuk penyediaan akun sementara pengunjung, sehingga dapat mengurangi beban administrasi terhadap pengguna yang berkunjung.

Dengan adanya keterkaitan pengembangan layanan eduroam dengan protokol dan hiearki RADIUS, maka subbab selanjutnya akan menjelaskan tentang *server* RADIUS secara detail.

2.2 RADIUS

Remote Authentication Dial in User Service (RADIUS) adalah sebuah protokol *Authentication, Authorization* dan *Accounting* (AAA) untuk aplikasi seperti akses jaringan atau mobilitas IP. Protokol ini biasanya digunakan untuk perangkat jaringan tertanam seperti *router, modem server, switch,* dll. RADIUS diakui sebagai standar dalam penerapan *remote authentication* dan *accounting* (Feng, 2009). Adapun penjelasan mengenai protokol AAA yaitu:

a. *Authentication*

   Merupakan proses pencocokan informasi (nama atau kata sandi) yang dimasukkan oleh pengguna ke dalam sebuah layanan jaringan yang telah dikonfigurasi di server RADIUS. Jika informasi yang dimasukkan cocok, maka pengguna diautentikasi dan memperoleh akses ke jaringan.

b. *Authorization*

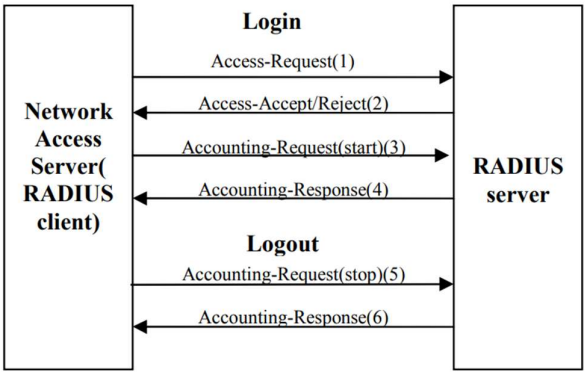
Merupakan proses pemberian izin terhadap pengguna yang telah mendapatkan hak akses di dalam jaringan. *Network Access Server* (NAS) akan mengirimkan informasi tentang pengguna yang telah dimasukkan sebelumnya sebagai paket *request* kepada *server* RADIUS. *Server* RADIUS akan memberikan validasi kemudian memberikan alamat IP kepada NAS untuk diteruskan ke pengguna. Dalam tahap ini, NAS bisa saja menolak alamat IP yang diberikan oleh *server* RADIUS sehingga koneksi akan terputus.

c. *Accounting*

Merupakan proses pencatatan informasi tentang sumber daya yang digunakan pengguna saat berada di dalam jaringan. Informasi dapat berupa waktu penggunaan sesi, jumlah paket data yang dikirim dan diterima, dan beberapa informasi yang diperlukan. Selama sesi di dalam jaringan, NAS akan mengirimkan informasi secara berkala. Semua informasi yang dicatat, akan disimpan sebagai data untuk membantu administrator dalam memelihara jaringan. Data tersebut digunakan untuk mengatur kebijakan penggunaan *bandwidth* maupun batasan dalam pengaksesan internet.

Gambar 2.2 menjelaskan tentang komunikasi antar RADIUS. Proses pertama yang dilakukan adalah NAS akan mengirimkan paket *access-request* kepada *server* RADIUS, kemudian *server* RADIUS akan mengirimkan kembali paket kepada NAS. Paket tersebut bisa berupa *access-accept/reject*. Jika yang dikirim adalah *access-accept*, maka NAS akan mengirimkan paket selanjutnya yaitu status *start* pada paket *accounting-request*. NAS akan meminta informasi *accounting* yang telah dikonfigurasi pada sisi *serve* RADIUS. Kemudian *server* RADIUS akan mengirimkan respon dari permintaan tersebut. Setelah pengguna selesai menggunakan layanan, maka NAS akan mengirimkan status *stop* pada paket *accounting-request* kepada *server* RADIUS. Setelah RADIUS menerima permintaan status *stop* dari NAS, maka *server* RADIUS akan mengirimkan respon berupa informasi *accounting* yang menandakan bahwa pengguna telah selesai menggunakan layanan.

Protokol RADIUS dibentuk berdasarkan model dari *client/server.* Sebuah RADIUS *client*, khususnya NAS (*network access server*), mengirimkan informasi kredensial pengguna dalam bentuk pesan RADIUS ke RADIUS *server.* RADIUS *server* mengotentikasi dan mengotorisasi permintaan dari RADIUS *client*, kemudian mengirimkan kembali sebuah pesan respons RADIUS. RADIUS *client* juga mengirimkan pesan akuntansi RADIUS ke RADIUS *server.* Pesan RADIUS dikirimkan dalam bentuk *User Datagram Protocol* (UDP).



Gambar 2.2 Diagram Komunikasi Antar RADIUS

Sumber: *Analysis, Implementation and Extensions of RADIUS Protocol* (Feng,2009)

Pada tabel 2.2 akan ditampilkan beberapa komponen sistem RADIUS yang dilengkapi dengan fungsi beserta contoh

Tabel 2.2 Tabel Komponen Sistem RADIUS

Nama Komponen	Fungsi	Contoh
Pengguna/Perangkat	Meminta akses ke jaringan	• Laptop • Modem ADSL • Telepon VOIP
NAS	Menyediakan akses ke jaringan untuk pengguna/perangkat	• Switch • Wireless AP • DSLAM • VPN Terminator
Authentication Server	• Menerima permintaan otentikasi dari NAS • Mengembalikan hasil otentikasi ke NAS • Secara opsional meminta informasi pengguna dan konfigurasi dari database • Dapat mengembalikan parameter konfigurasi ke NAS	• FreeRADIUS • Radiator • IAS • NPS • ACS

Nama Komponen	Fungsi	Contoh
	• Menerima informasi akuntansi dai NAS	
Data Store	• Opsional database atau direktori dengan informasi otentikasi dan otorisasi pengguna. *Server* RADIUS berkomunikasi dengan Data Store yang menggunakan DB API atau LDAP	• SQL Database • Kerberos Service Server • LDAP Directory

Pada penerapan layanan eduroam di UII, *server* RADIUS yang digunakan adalah FreeRADIUS. FreeRADIUS berbasis *open-source* sehingga lebih mudah dikembangkan dan diterapkan. Untuk penjelasan lebih detail mengenai FreeRadius, dapat dilihat pada subbab selanjutnya.

2.3 FreeRADIUS

FreeRADIUS adalah *server* RADIUS berbasis *open-source* yang paling popular dan paling banyak disebarkan di dunia. FreeRADIUS berfungsi sebagai dasar dalam beberapa penawaran komersial, dan menyediakan kebutuhan AAA dari banyak perusahaan dan ISP tingkat 1. FreeRADIUS juga banyak digunakan oleh komunitas akademik (contohnya: eduroam) (Anonymous, *THE FREERADIUS TECHNICAL GUIDE*, 2014).

FreeRADIUS dikembangkan oleh Alan DeKok dan Miquel van Smoorenburg pada bulan Agustus tahun 1999. Miquel sebelumnya telah merilis aplikasi *server* Cistron RADIUS, yang secara luas digunakan ketika *server* Livingston (*server* RADIUS pertama yang menjadi induk dari semua *server* RADIUS di masa mendatang) tidak lagi beroperasi. FreeRADIUS dikembangkan menggunakan desain modular, untuk meningkatkan keaktifan komunitas yang terlibat (Anonymous, *THE FREERADIUS TECHNICAL GUIDE*, 2014).

FreeRADIUS dapat digunakan di berbagai *server* RADIUS dan berdasarkan desain protokol *feature-rich, modular* dan *scalable*, yang memberikan manfaat dan keuntungan kepada administrator jaringan antara lain:

a. *Feature-rich*

FreeRADIUS mendukung lebih banyak tipe otentikasi dibandingkan *server open-source* lainnya. Sebagai contoh, FreeRADIUS merupakan satu-satunya *server* RADIUS *open-source* yang mendukung EAP dan *virtual server*. Penggunaan *virtual server* dapat mengurangi biaya pemeliharaan *server* dan juga dapat menyederhanakan implementasi yang kompleks. Dengan demikian, kemampuan *virtual server* FreeRADIUS memberikannya keuntungan besar dalam persaingan antar *server open-source*

b. *Modular*

Protokol desain modular membuat FreeRADIUS lebih mudah dimengerti. Antarmuka modular juga menyederhanakan dalam penambahan atau penghapusan modul. Misalnya, jika fitur tidak diperlukan untuk konfigurasi maka modul tersebut dapat dihapus. Modul yang telah dihapus tidak akan mempengaruhi kinerja *server*, penggunaan memori dan keamanan. Fleksibilitas ini memungkinkan *server* untuk berjalan di *platform* mulai dari sistem yang tertanam hingga mesin *multi-core* yang memiliki kapasitas RAM *gigabyte*.

c. *Scalable*

Sebuah *server* RADIUS dapat dengan mudah bertransisi dari menangani satu permintaan setiap detik hingga menangani ribuan permintaan setiap detik, cukup dengan mengkonfigurasi ulang beberapa pengaturan *default*. Banyak organisasi besar bergantung pada FreeRADIUS untuk kebutuhan AAA mereka. Cukup dengan menggunakan satu *server* FreeRADIUS dapat memenuhi kebutuhan suatu organisasi yang besar.

Dengan FreeRADIUS sebagai *server* yang menyediakan kebutuhan AAA, maka setiap institusi dapat menganalisis layanan eduroam dengan mengolah informasi yang tersimpan pada *logfile.* Informasi pada *logfile* dapat diolah dengan metode *parsing* sehingga data lebih mudah dibaca dan dianalisis. Penjelasan detail mengenai *parsing* dapat dilihat pada subbab selanjutnya.

2.4 *Parsing*

Parsing menurut bahasa artinya adalah mengurai atau mencacah. Menurut ilmu komputer *parsing* atau *syntax analysis* adalah proses menganalisis urutan token untuk menentukan struktur gramatik dengan memperhatikan tata bahasa formal tertentu (Manu, 2016). Proses *parsing* akan membagi kalimat menjadi kata-kata yang kemudian hasil pembagian tersebut digunakan ke proses selanjutnya.

Untuk melakukan proses *parsing,* dibutuhkan suatu program yang dinamakan *parser*. *Parser* adalah *compiler* atau *interpreter* yang memecah data menjadi elemen yang lebih kecil untuk mempermudah terjemahan ke bahasa lain. *Parser* mengambil bentuk urutan token atau instruksi program dan biasanya membangun struktur data dalam bentuk pohon parsing. Pada penelitian ini, peneliti menggunakan *parser tool* yang tersedia pada distro linux yakni AWK. Penjelasan mengenai AWK akan dijelaskan pada subbab selanjutnya.

2.5 AWK

AWK adalah sebuah perintah atau *tool* yang tersedia di semua distro Linux atau Unix untuk melakukan penyaringan teks, manipulasi dll. Alat ini digunakan terutama untuk memproses file teks dan pelaporan. AWK dapat diperlakukan sebagai bahasa pemrograman karena kemampuannya seperti operasi aritmatika, binari, kondisi, *looping*, fungsi dll (Anne, 2013).

Saat digunakan sebagai alat penanganan teks (*text-handling tool*), AWK memungkinkan pengguna untuk menentukan pola kata atau kalimat untuk dicocokkan. Dengan menggunakan ekspresi regular (*regular expression/regex/regexp*), pola kata atau kalimat dapat ditentukan. File input diuraikan dalam baris demi baris. Ketika kata atau kalimat cocok dengan pola yang ditentukan, maka perintah AWK akan dieksekusi, baik dengan mengganti kata atau kalimat atau menulis ulang seluruh baris (Cheng & Lin, 2008). Pada penelitian ini, AWK akan digunakan untuk proses parsing *logfile* menjadi data yang lebih sederhana yang kemudian diubah ke dalam bentuk .csv (*comma separated value*).