

BAB 2 LANDASAN KEPUSTAKAAN

Di dalam bab ini, diambil beberapa kajian pustaka dan dasar teori yang dijadikan sebagai landasan kepastakaan dalam pengerjaan implementasi autentikasi mode multi-auth pada jaringan *Local Area Network* berbasis kabel menggunakan protokol IEEE 802.1X dan radius server, analisis dan perancangan serta pengujian yang telah ada pada BAB 1.

2.1 Kajian Pustaka

Tabel 2.1 di bawah ini merupakan beberapa kajian pustaka dari hasil penelitian sebelumnya yang terkait dengan penelitian yang dilakukan oleh penulis saat ini.

Tabel 2.1 Kajian Pustaka

No	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian terdahulu	Rencana penelitian
1.	Kothaluru, T.R. dan Mecca, M.Y.S., (2012). <i>Evaluation of EAP Authentication Methods in Wired and Wireless Networks.</i>	Melakukan implementasi standar protokol IEEE 802.1X dan radius server di jaringan kabel serta melakukan pengujian perbandingan waktu autentikasi menggunakan salah satu metode EAP di jaringan kabel	Melakukan pengujian perbandingan waktu autentikasi menggunakan metode EAP yang berbeda pada jaringan kabel dan nirkabel	Melakukan pengujian perbandingan waktu autentikasi pada mode single-host dan multi-auth dengan menggunakan metode EAP-PEAP
2.	Loos, J., (2014). <i>Implementing IEEE 802.1X for Wired Networks</i>	Melakukan implementasi standar protokol IEEE 802.1X dan radius server dengan mode single-host di jaringan kabel	Melakukan implementasi standar protokol IEEE 802.1X dan radius server dengan mode single-host di jaringan kabel tanpa ada kesimpulan dari hasil autentikasi	Melakukan implementasi standar protokol IEEE 802.1X dan radius server menggunakan mode multi-auth di jaringan kabel

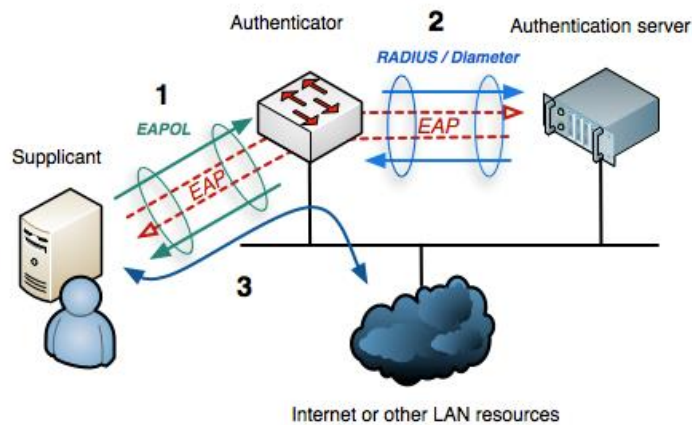
3.	Kovacic, S., Đulić, E. & Sehidic, A., (2017). <i>Improving the Security of Access to Network Resources Using the 802.1X Standard in Wired and Wireless Environments</i>	Melakukan implementasi standar protokol IEEE 802.1X dan radius server di jaringan kabel	Melakukan implementasi standar protokol IEEE 802.1X dan radius server di jaringan kabel dan nirkabel tanpa menggunakan active directory sebagai proses autentikasi	Melakukan implementasi standar protokol IEEE 802.1X dan radius server menggunakan mode multi-auth di jaringan kabel dan menggunakan active directory untuk melakukan autentikasi
----	---	---	--	--

Sumber: (Kothaluru, T.R. dan Mecca, M.Y.S., 2012) (Loos, J., 2014) (Kovacic, S., Đulić, E. & Sehidic, A., 2017)

2.2 IEEE 802.1X

Standar IEEE 802.1X menawarkan visibilitas yang belum pernah ada sebelumnya dan akses kontrol berbasis identitas yang aman di jaringan. Dengan desain dan komponen, dapat memenuhi kebutuhan kebijakan dalam keamanan, meminimalkan dampak penyerangan terhadap infrastruktur dan pengguna. Konsultan, kontraktor dan tamu sekarang memerlukan akses ke sumber daya jaringan melalui koneksi LAN yang sama dengan karyawan biasa, yang mungkin membawa perangkat yang tidak terkelola ke tempat kerja. Karena koneksi jaringan menjadi sangat diperlukan dalam operasi bisnis sehari-hari, kemungkinan orang atau perangkat yang tidak teridentifikasi akan dapat mengakses informasi data rahasia yang ada di perusahaan atau organisasi tersebut. Solusi terbaik dan paling aman untuk kerentanan pada sisi akses adalah memanfaatkan teknologi keamanan jaringan.

IEEE 802.1X adalah sebuah standar IEEE untuk Port-based Network Access Control (PNAC), yang dirancang untuk lingkungan jaringan berbasis kabel yang menyediakan kemampuan untuk mengizinkan atau menolak konektivitas jaringan berdasarkan identitas pengguna atau perangkat yang ingin tersambung ke sebuah *Local Area Network* (LAN). Port dalam jaringan kabel diartikan sebagai port fisik pada switch Ethernet. Standarnya mendefinisikan metodologi enkapsulasi untuk pengangkutan EAP melalui LAN (EAPOL) dan menyediakan kerangka kerja autentikasi yang kuat di mana setiap protokol autentikasi menyediakan tingkat keamanan yang tinggi (CISCO, 2011).



Gambar 2.1 Arsitektur 802.1X

Sumber: CISCO (2011)

2.2.1 Komponen IEEE 802.1X

Standar IEEE 802.1X menyediakan akses kontrol Layer data-link (L2) menggunakan validasi kredensial data pengguna atau perangkat yang mencoba mengakses port fisik. Menurut Kovacic, Đulić dan Sehidic (2017) IEEE 802.1X memiliki tiga komponen utama yaitu:

1. *Supplicant* (pengguna)

Perangkat yang mampu mendukung protokol IEEE 802.1X (misalnya ponsel, PC dll) dapat digunakan untuk mendapatkan hak autentikasi untuk mendapatkan akses melalui jaringan. Proses yang dilakukan adalah *supplicant* mengirimkan kredensial yang diperlukan ke authenticator dan selanjutnya dikirim ke server autentikasi untuk mendapatkan akses melalui jaringan. Komunikasi antara *supplicant* dan authenticator dibuat menggunakan EAPOL dan beroperasi di layer data-link. Karena operasi berlangsung di layer data-link, tidak diperlukan alamat IP untuk memulai proses autentikasi.

2. *Authenticator*

Perangkat seperti switch, router atau access point. Authenticator bertindak sebagai perantara antara *supplicant* dan server autentikasi untuk mengendalikan akses di antara keduanya. Kredensial data yang dikonfirmasi atau ditolak oleh server autentikasi dilewatkan melalui authenticator. Umumnya, authenticator mengatur port-portnya baik terbuka maupun tertutup dari respon yang diterima oleh server autentikasi atas permintaan yang diberikan oleh *supplicant*. Bergantung pada respon yang diberikan oleh server autentikasi, authenticator memutuskan apakah *supplicant* harus diberi hak akses atau tidak.

3. *Server Autentikasi*

Server autentikasi adalah server yang mendukung protokol RADIUS dan EAP. Server autentikasi bertugas untuk memeriksa dan memvalidasi identitas *supplicant* yang dikirimkan oleh *supplicant* melalui authenticator.



Gambar 2.2 Komponen 802.1X

Sumber: CISCO (2011)

2.2.2 Protokol IEEE 802.1X

Menurut CISCO (2011) IEEE 802.1X menggunakan protokol berikut:

1. Extensible Authentication Protokol (EAP)

EAP adalah sebuah kerangka autentikasi hasil pengembangan IEEE yang berfungsi secara fleksibel. EAP sendiri tidaklah mekanisme autentikasi secara spesifik, EAP hanya menyediakan fungsi transport untuk membawa informasi autentikasi yang disediakan oleh metode EAP. Dengan begitu, jika ada mekanisme autentikasi (metode EAP) baru, tidak perlu melakukan *upgrade* pada semua peralatan jaringan. Saat ini terdapat banyak metode EAP, namun yang memenuhi standar untuk bekerja di jaringan kabel sebagaimana yang dijelaskan pada *Request For Comments* (RFC) 4017 hanya ada tujuh metode, termasuk di antaranya adalah MD5, EAP-TLS, EAP-PEAP dan EAP-TTLS.

2. EAP over LAN (EAPoL)

Enkapsulasi yang didefinisikan oleh IEEE 802.1X untuk awal proses pengiriman EAP dari *supplicant* ke *authenticator* melalui jaringan IEEE 802.1X. EAPoL adalah protokol layer data-link.

3. RADIUS (Remote Authentication Dial in User Service)

Merupakan protokol yang diterapkan secara luas yang digunakan untuk membawa informasi autentikasi, otorisasi dan *accounting* antara server autentikasi dan authenticator. Protokol jaringan yang mengimplementasikan manajemen AAA secara terpusat bagi komputer atau pengguna yang ingin terkoneksi dan menggunakan layanan dalam suatu jaringan. RADIUS secara default menggunakan port 1812 dan UDP (User Datagram Protocol) sebagai protokol transport. Walaupun IEEE 802.1X tidak secara spesifik disebutkan merupakan jenis server autentikasi yang digunakan, namun RADIUS secara “*de facto*” adalah server autentikasi untuk 802.1X.

2.2.3 Metode EAP

EAP adalah protokol yang umum digunakan untuk autentikasi pengguna di jaringan berbasis IEEE 802.1X. Namun, EAP sebenarnya lebih tepat untuk dikatakan sebagai kerangka komunikasi antara *supplicant* dan server serta protokol yang digunakan dalam komunikasi dapat dipilih dari protokol yang tersedia dalam kerangka EAP itu sendiri. *Supplicant* dan server harus menggunakan protokol yang sama untuk autentikasi dan komunikasi (Hermaduanti dan Riadi, 2016). Beberapa protokol yang ada dalam kerangka EAP, juga dikenal sebagai metode EAP yang

biasa digunakan untuk berkomunikasi dalam jaringan berbasis IEEE 802.1X yang diantaranya adalah sebagai berikut:

1. EAP-MD5

Protokol EAP-MD5 dijelaskan dalam RFC 2284. Ini serupa dengan protokol PPP-CHAP. Ini adalah protokol *challenge response handshake*. Menggunakan id dan kata sandi agar pengguna dikenali. Database autentikasi menyimpan semua ID pengguna dan kata sandi. Karena MD5 adalah protokol *challenge*, server RADIUS mengirimkan *challenge* secara acak kepada *supplicant*. *Supplicant* membuat hash MD5 dari kata sandi pengguna, *supplicant* mengembalikan hash ke server, server akan memeriksa hash di database. Server yang menerima hash MD5 dari *supplicant* akan dibandingkan dengan nilai hash MD5 yang telah tersimpan di server. Perbandingan akan menentukan apakah *supplicant* valid atau tidak (Hermaduanti dan Riadi, 2016).

2. EAP-TLS

EAP-TLS adalah standar IETF yang didefinisikan dalam RFC 2715. EAP-TLS menangani sejumlah kelemahan dalam protokol EAP lainnya untuk autentikasi yang aman. Namun, dalam mengatasi kelemahan ini, EAP-TLS meningkatkan kompleksitas penyebaran. Tidak seperti EAP-PEAP (yang hanya memerlukan sertifikat di sisi server), EAP-TLS memerlukan sertifikat di sisi *supplicant* dan di sisi server untuk saling autentikasi. Dalam IEEE 802.1X, pertukaran pesan EAP-TLS saling autentikasi, negosiasi metode enkripsi, dan menentukan kunci enkripsi antara *supplicant* dan server autentikasi (CISCO,2011).

EAP-TLS dijelaskan dalam RFC 2716, menggunakan sertifikat *public key infrastructure* (PKI) untuk *supplicant* dan server autentikasi untuk saling memberikan autentikasi antara keduanya. Sertifikat PKI akan berisi informasi tentang nama server atau informasi pengguna. Hal ini adalah salah satu metode yang aman yang digunakan, karena *tunnel* TLS dibuat saat pertukaran sertifikat antara *supplicant* dan server autentikasi. Hal lain yang perlu dicatat di sini adalah meskipun sebuah *tunnel* dibuat untuk melindungi pesan EAP, identitas pengguna dikirim dengan teks biasa sebelum proses pertukaran sertifikat dimulai (Kothaluru dan Mecca, 2012).

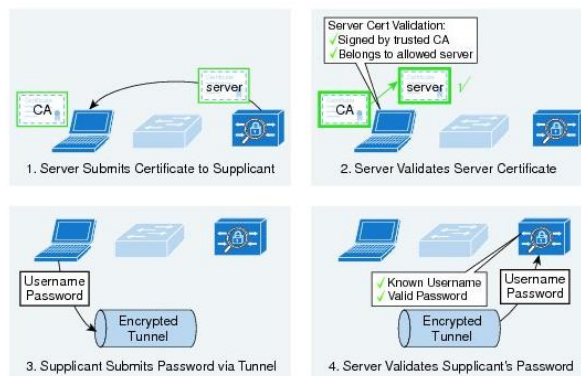
3. EAP-TTLS

EAP-TTLS dijelaskan dalam RFC 5281, perpanjangan dari EAP-TLS, dibuat untuk mengurangi kompleksitas penerapan yang dihadapi saat menerapkan TLS yaitu untuk menghilangkan sertifikat digital PKI. Setelah pembuatan server autentikasi, TTLS saja perlu mengotentikasi dirinya sendiri kepada si *supplicant*. *Supplicant* secara opsional dapat mengautentikasi dirinya ke server. Oleh karena itu metode autentikasi ada satu atau dua cara. EAP-TTLS mendukung banyak *inner* protokol seperti PAP, CHAP, MSCHAP dan MSCHAPv2 untuk autentikasi *supplicant*. Proses autentikasi berlangsung di dalam *tunnel* yang aman. Ada dua versi TTLS yaitu TTLSv1 dan TTLSv2 (Kothaluru dan Mecca, 2012).

4. EAP-PEAP

EAP-PEAP dikembangkan oleh Cisco Systems, Microsoft Corporation, dan RSA Security, Inc. PEAP adalah tipe EAP yang membahas masalah keamanan dengan terlebih dahulu membuat *tunnel* yang aman serta dienkripsi untuk melindungi integritas data dengan TLS. *Tunnel* ini dibuat dengan menggunakan sertifikat server yang valid yang dikirim oleh server autentikasi ke *supplicant* pada awal negosiasi PEAP. Di dalam *tunnel* yang aman ini, negosiasi EAP baru terjadi untuk mengautentikasi *supplicant*. Karena *tunnel* TLS melindungi negosiasi EAP dan autentikasi untuk upaya akses jaringan. Autentikasi berbasis kata sandi yang biasanya rentan terhadap serangan dapat digunakan dengan aman untuk autentikasi. Dengan membungkus pesan EAP dalam TLS, metode EAP yang berjalan di dalam PEAP dilengkapi dengan dukungan *built-in* untuk pertukaran kunci, pembukaan kembali *session* dan pemasangan kembali. Selanjutnya, karena PEAP memerlukan sertifikat hanya di server autentikasi, memungkinkan untuk mengautentikasi *supplicant* LAN dengan aman tanpa meminta setiap *supplicant* untuk memiliki sertifikat sendiri (Kothaluru dan Mecca, 2012).

MSCHAPv2 biasanya digunakan sebagai tipe EAP kedua di dalam *tunnel* PEAP. MS-CHAPv2 adalah protokol autentikasi *mutual-response* berbasis kata kunci, *challenge-response*, yang menggunakan Message-Digest Algorithm (MD4) dan Data Encryption Standard (DES) untuk mengenkripsi *response*. Authenticator meminta koneksi *supplicant* dan *supplicant* dapat meminta koneksi server autentikasi. Jika salah satu permintaan tidak dijawab dengan benar, koneksi bisa ditolak. Meskipun MSCHAPv2 memberikan perlindungan yang lebih baik daripada protokol autentikasi *challenge-response* sebelumnya, namun masih rentan terhadap serangan Dictionary offline. Pengguna jahat dapat menangkap pertukaran MSCHAPv2 dan menebak kata sandi sampai benar. Digunakan dalam kombinasi dengan PEAP, bagaimanapun, pertukaran MSCHAPv2 dilindungi dengan keamanan yang kuat dari *tunnel* TLS. PEAP-MSCHAPv2 digunakan terutama di lingkungan Microsoft Active Directory (CISCO,2011).



Gambar 2.3 fungsi High Level PEAP-MSCHAPv2

Sumber: CISCO (2011)

2.2.4 Urutan Operasi Dalam 802.1X

Menurut CISCO (2011), urutan tahapan operasi dalam IEEE 802.1X adalah sebagai berikut:

1. Session Initiation

Autentikasi IEEE 802.1X dapat diinisiasi oleh switch authenticator atau *supplicant*. Dari perspektif switch, *session* autentikasi dimulai saat switch mendeteksi adanya *link up* ke port. Switch memulai autentikasi dengan mengirimkan pesan *EAP-Request-Identity* ke *supplicant*. Jika switch tidak menerima *response*, switch mentransmisikan kembali permintaan pada interval periodik tertentu.

Supplicant dapat melakukan autentikasi dengan mengirimkan frame EAPoL-Start. Pesan EAPoL-Start memungkinkan *supplicant* untuk mempercepat proses autentikasi tanpa menunggu *EAP-Request-Identity* berikutnya dari switch. Pesan EAPoL-Start diperlukan dalam situasi di mana *supplicant* tidak siap memproses Permintaan EAP dari switch (misalnya, karena sistem operasi masih melakukan *booting*) atau bila tidak ada perubahan *physical link state* pada switch (misalnya, karena *supplicant* secara tidak langsung terhubung melalui IP telepon atau hub).

2. Session Authentication

Selama tahap ini, switch me-*relay* pesan EAP antara *supplicant* dan server autentikasi, menyalin pesan EAP dalam *frame* EAPoL ke paket RADIUS dan sebaliknya. Pada bagian pertama pertukaran, *supplicant* dan server autentikasi menyetujui metode EAP yang ingin digunakan.

Sisa pertukaran ditentukan oleh metode EAP yang digunakan. Metode EAP mendefinisikan jenis kredensial yang digunakan untuk memvalidasi identitas *supplicant* dan bagaimana kredensial diajukan. Bergantung pada metode tersebut, *supplicant* dapat mengajukan sandi, sertifikat, token, atau kredensial lainnya. Kredensial itu kemudian dapat dilewatkan ke dalam *tunnel* terenkripsi TLS, seperti hash atau beberapa bentuk keamanan lainnya.

3. Session Authorization

Jika *supplicant* mengajukan kredensial yang valid, server autentikasi mengembalikan pesan RADIUS *Access-Accept* dengan pesan *EAP-Success* yang dikapsulasi. Hal ini menunjukkan kepada switch bahwa *supplicant* harus diizinkan mengakses port. Opsional, server autentikasi mungkin menyertakan kebijakan akses jaringan dinamis (misalnya, VLAN dinamis atau ACL) dalam pesan *Access-Accept*. Dengan tidak adanya instruksi kebijakan yang dinamis, switch hanya membuka port.

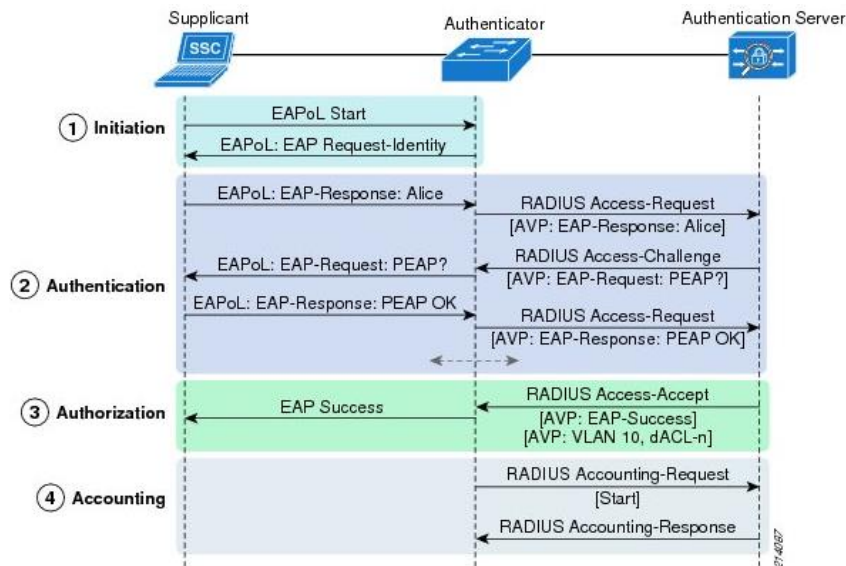
Jika *supplicant* mengajukan kredensial yang tidak sah atau tidak diizinkan mengakses jaringan karena alasan kebijakan, server autentikasi mengembalikan pesan RADIUS *Access-Reject* dengan pesan *EAP-Failure* yang dikapsulasi. Hal ini mengindikasikan bahwa *supplicant* tidak diizinkan mengakses port. Bergantung pada bagaimana switch dikonfigurasi, mungkin mencoba lagi autentikasi, gunakan port ke VLAN Auth-Fail, atau coba metode autentikasi alternatif.

4. Session Accounting

Jika switch berhasil menerapkan kebijakan otorisasi, switch dapat mengirim pesan *RADIUS Accounting-Request* ke server autentikasi dengan rincian tentang *session* yang berwenang. Pesan *Accounting-Request* dikirim untuk *session* yang diberi wewenang secara dinamis dan juga *session* resmi lokal misalnya *Guest VLAN* dan *Auth-Fail VLAN*.

5. Session Termination

Pemutusan *session* merupakan bagian penting dari proses autentikasi IEEE 802.1X. Untuk memastikan integritas *session* yang diautentikasi, *session* harus dihapus saat *endpoint* yang terautentikasi terputus dari jaringan. *Session* yang tidak segera diakhiri dapat menimbulkan celah yang dapat ditembus dari bagian keamanan yang ada. Idealnya, penghentian *session* terjadi setelah *endpoint* dicabut secara fisik, tapi hal ini tidak selalu terjadi, mungkin jika *endpoint* terhubung secara tidak langsung misalnya melalui IP phone atau hub.



Gambar 2.4 Urutan Operasi High-Level 802.1X

Sumber: CISCO (2011)

2.2.5 Host-Mode

Menurut CISCO (2011) Secara *default*, IEEE-802.1X-enabled port hanya mengizinkan satu perangkat per port fisik. Setiap alamat MAC tambahan yang terhubung di port menyebabkan pelanggaran dalam keamanan. Seringkali, batasan satu perangkat per port tidak sesuai dengan semua persyaratan jaringan. Switch Cisco Catalyst memungkinkan untuk menangani beberapa kasus penggunaan dengan memodifikasi perilaku default. Host-mode pada port fisik menentukan jumlah dan jenis perangkat yang diperbolehkan pada port. Adapun host-mode dan pengaplikasiannya adalah sebagai berikut:

1. Mode single-host

Dalam mode single-host, hanya satu MAC atau alamat IP yang dapat diautentikasi dengan metode apapun pada port. Jika alamat MAC yang berbeda terdeteksi pada port setelah perangkat telah dikonfirmasi dengan 802.1X, MAB, atau Web Authentication, maka terjadi pelanggaran keamanan yang dilakukan pada port tersebut. Ini adalah perilaku default dari 802.1X.

2. Mode multi-domain-authentication (MDA)

MDA dirancang khusus untuk memenuhi persyaratan IP telepon di lingkungan 802.1X. Bila MDA dikonfigurasi, dua perangkat diperbolehkan mengakses port: satu di *voice* VLAN, dan satu di data VLAN. Alamat MAC tambahan memicu pelanggaran sistem keamanan.

3. Mode multi-auth

Jika port dikonfigurasi untuk mode multi-auth, beberapa perangkat dapat diautentikasi dalam data Virtual Local Area Network atau VLAN. Setiap alamat MAC baru yang muncul di port diautentikasi secara terpisah. Memungkinkan beberapa alamat MAC dalam satu port yang sama, namun semua perangkat harus diautentikasi untuk mengakses sumber daya jaringan. Multi-auth dapat digunakan untuk menjembatani lingkungan virtual atau mendukung dengan perpanjangan alat yaitu hub.

4. Mode multi-host

Tidak seperti mode multi-auth, yang mengautentikasi setiap alamat MAC, mode multi-host mengautentikasi alamat MAC pertama dan kemudian mengizinkan sejumlah alamat MAC lainnya secara tidak terbatas. Dalam mode multi-host, dapat melampirkan beberapa host ke satu port *802.1X-enabled port*. Dalam mode ini, hanya satu dari *supplicant* yang harus diberi wewenang agar semua *supplicant* diberi akses jaringan. Jika port menjadi tidak terautentikasi (kesalahan reautentikasi atau pesan logoff-EAPOL diterima), switch menolak akses jaringan dari semua *supplicant* yang terlampir. Dalam hal ini, titik akses jaringan kabel bertanggung jawab untuk mengautentikasi *supplicant* yang menyertainya, dan ini juga bertindak sebagai *supplicant* untuk beralih. Dengan mode multi-host yang diaktifkan, dapat menggunakan autentikasi 802.1X untuk mengautentikasi port dan dapat menggunakan keamanan port untuk mengelola akses jaringan untuk semua alamat MAC, termasuk alamat MAC *supplicant* yang memakai port tersebut.

2.3 Switch

Switch adalah perangkat yang digunakan untuk menghubungkan beberapa perangkat lain dalam jaringan dan mengaturnya sedemikian rupa agar perangkat-perangkat tersebut dapat berkomunikasi secara baik dan efisien.

Switch terdiri dari dua jenis yaitu:

1. Switch unmanageable

Switch unmanageable adalah switch yang tidak didesain untuk dikonfigurasi sehingga kemampuan menghubungkannya terbatas. Biasa digunakan dalam jaringan berskala kecil yang hanya membutuhkan sedikit hubungan perangkat.

2. Switch manageable

Switch manageable adalah switch yang didesain untuk bisa di konfigurasi sehingga dapat mengatur koneksi dalam jumlah yang banyak dan menawarkan kontrol atas koneksi yang lebih luas. Biasa digunakan di jaringan dengan jumlah koneksi yang banyak seperti perusahaan.

2.4 Radius Server

Radius mempunyai standar IEEE 802.1X, yang berguna untuk menghasilkan akses kontrol dan Konsep AAA (Autentikasi, authorization dan *accounting*) untuk jaringan kabel berbasis Protokol UDP. Apabila pengguna melakukan koneksi ke suatu jaringan menggunakan kabel, maka radius akan bekerja dengan ketiga konsep metodenya tersebut. Dengan metode autentikasi, yaitu memastikan apakah *supplicant* tersebut benar telah terdaftar pada sebuah jaringan berbasis kabel, autentikasi merupakan sebuah integritas, dapat dilihat keaslian identitas *supplicant* dengan menggunakan *username* dan kata sandi. Terdapat port autentikasi di radius, yaitu port 1812. Terdapat vendor – vendor hardware dan software yang mengimplementasikan radius sebagai solusi autentikasi pengguna, jadi integritas dalam suatu network dapat benar – benar terjaga apabila menggunakan radius (Microsoft, 2017).

Otorisasi yaitu untuk mengetahui hak akses dia sebagai apa, apakah hanya sebagai *supplicant*, administrator (yang bekerja untuk meng-insert, update, delete), atau sebagai pimpinan. Hal ini digunakan agar keamanan jaringan lebih terkontrol karena telah di bagi tingkat aksesnya masing – masing. Konsep lainnya adalah *accounting*, *accounting* ini merupakan pencatatan kegiatan yang dilakukan *supplicant* dimulai dari awal mengakses jaringan hingga selesai menggunakan jaringan. radius *accounting* menggunakan port 1813 namun ada juga vendor yang menggunakan port 1645/1646 (Kothaluru dan Mecca, 2012). Menurut Kothaluru dan Mecca (2012) fitur utama radius adalah:

- Bertanggung jawab untuk menyampaikan informasi pengguna.
- Menunggu sampai respon dikembalikan.
- Bertanggung jawab atas permintaan koneksi pengguna, melakukan otentikasi kepada pengguna dan menyediakan semua informasi konfigurasi yang diperlukan untuk menyampaikan informasi dari server ke pengguna.

Pada hal ini, penelitian akan memakai radius server milik Windows Server yaitu Network Policy Server (NPS). NPS adalah implementasi Microsoft server

Remote-Authentication Dial-In User Service (RADIUS). Sebagai server radius, NPS melakukan autentikasi, otorisasi dan *accounting* secara terpusat untuk berbagai jenis akses jaringan, termasuk akses remote nirkabel, autentikasi,jaringan kabel, akses dial-up dan virtual private network (VPN) dan koneksi router-ke-router(Microsoft, 2017).

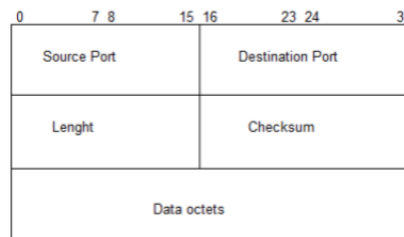
NPS memungkinkan penggunaan perangkat nirkabel, switch, remote access atau peralatan VPN yang heterogen, dapat menggunakan NPS dengan layanan Routing dan Remote Access, yang tersedia di Microsoft Windows Server.

Bila server NPS adalah anggota domain Active Directory, NPS menggunakan layanan direktori sebagai basis data akun pengguna dan merupakan bagian dari solusi untuk masuk kedalam jaringan. Kumpulan kredensial data yang sama digunakan untuk akses kontrol jaringan (mengautentikasi dan memberi otorisasi akses ke jaringan) dan masuk ke domain Active Directory (Microsoft, 2017).

Server radius memiliki akses ke informasi akun pengguna dan dapat memeriksa kredensial data autentikasi akses jaringan. Jika kredensial data pengguna itu asli dan upaya penyambungannya diotorisasi, server radius memberi otorisasi akses pengguna berdasarkan kondisi yang ditentukan dan mencatat koneksi akses jaringan di log *accounting*. Penggunaan radius memungkinkan akses jaringan autentikasi pengguna, otorisasi, dan data *accounting* dikumpulkan dan dijaga di lokasi sentral, bukan pada setiap server akses. (Microsoft, 2017)

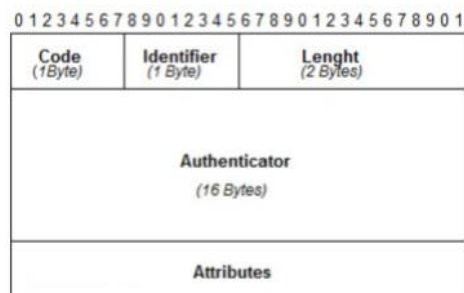
2.4.1 Format Paket

Menurut Kothaluru dan Mecca (2012) menyatakan bahwa “Setiap paket di dalam server radius dienkapsulasi dalam *field* data UDP. Port tujuan UDP menunjukkan nomor port radius. Port yang ditugaskan untuk autentikasi radius adalah 1812 dan untuk *accounting* adalah 1813.



Gambar 2.5 Struktur paket UDP

Format frame pada radius sebagai berikut:



Gambar 2.6 Format frame radius

2.4.1.1 Code

Bidang Code adalah satu byte. Code diidentifikasi sebagai jenis paket radius. Radius menerima paket untuk memeriksa bidang code dan jika code yang diterima tidak valid, maka paket tersebut akan dibuang. Membedakan tipe pesan RADIUS yang dikirimkan berdasarkan daftar *code* pada Gambar 2.7.

Code RADIUS (desimal) adalah sebagai berikut:

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

Gambar 2.7 Code radius dan operasinya

2.4.1.2 Identifier

Panjang identifier adalah satu byte. Berfungsi untuk memeriksa *request* dan *response* dari frame yang dikirim dan diterima.

2.4.1.3 Length

Bidang *length* digunakan untuk memeriksa total byte yang dikirim dalam paket termasuk *identifier*, *length*, *authenticator*, *attribute* dan *code*. Jika paket berisi beberapa byte tambahan maka byte tambahan dianggap sebagai *padding* dan data diabaikan.

2.4.1.4 Authenticator

Authenticator memiliki panjang 16 byte. Hal ini digunakan untuk *request* dan *response* dari authenticator.

2.4.1.5 Attributes

Attributes berisikan informasi yang dibawa pesan RADIUS. Setiap pesan dapat membawa satu atau lebih atribut. Contoh *attribute* RADIUS adalah *username, password, CHAP-password, alamat IP dan pesan balasan*”.

2.5 Transport Layer Security

Transport layer security atau TLS adalah sebuah protokol yang menjamin privasi dan integritas data antara client atau server yang berkomunikasi. TLS mempunyai kapabilitas untuk melayani servis autentikasi antara client dan server dengan membuat koneksi yang dienkripsi antara keduanya sehingga memberikan garansi integritas dan kerahasiaan pengiriman data. TLS merupakan protokol yang independen terhadap protokol aplikasi, sehingga protokol pada tingkat lebih tinggi dapat secara transparan dijalankan di atas protokol TLS. Protokol TLS memberikan tiga fungsi security yaitu data confidentiality, data integrity dan juga autentikasi (Loos, 2014).

2.6 Algoritma RSA

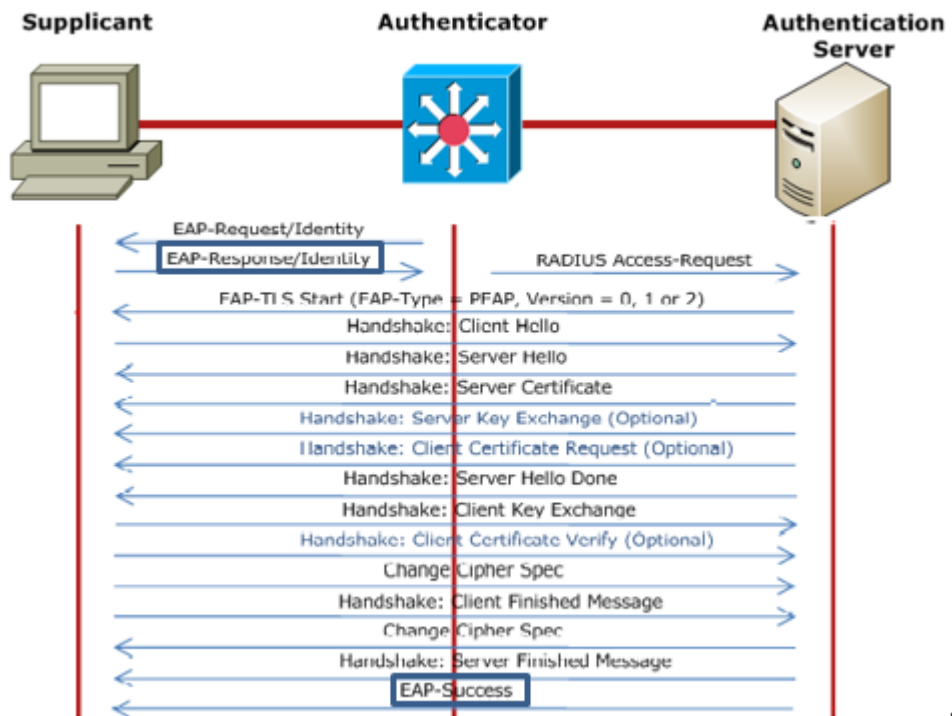
Algoritma ini dinamakan sesuai dengan nama penemunya, Ron Rivest, Adi Shamir dan Adleman (Rivest-Shamir-Adleman) yang dipublikasikan pada tahun 1977 di MIT, menjawab tantangan yang diberikan algoritma pertukaran kunci Diffie Hellman.

RSA merupakan algoritma kriptografi asimetri, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Kunci yang digunakan untuk mengenkripsi disebut dengan kunci publik, dan yang digunakan untuk mendekripsi disebut dengan kunci privat. RSA adalah salah satu algoritma kriptografi yang menggunakan konsep kriptografi kunci publik. RSA membutuhkan tiga langkah dalam prosesnya, yaitu pembangkitan kunci, enkripsi, dan dekripsi. Proses enkripsi dan dekripsi merupakan proses yang hampir sama. Jika bilangan acak yang dibangkitkan kuat, maka akan lebih sulit untuk melakukan cracking terhadap pesan. Parameter kuat tidaknya suatu kunci terdapat pada besarnya bilangan acak yang digunakan. Skema RSA sendiri mengadopsi dari skema block cipher, dimana sebelum dilakukan enkripsi, plaintext yang ada dibagi – bagi menjadi blok – blok dengan panjang yang sama, dimana plaintext dan ciphertextnya berupa integer (bilangan bulat) antara 1 hingga n , dimana n berukuran biasanya sebesar 1024 bit (Zhou dan Tang, 2011).

2.7 Waktu Autentikasi

2.7.1 Waktu Autentikasi Berhasil

Untuk menghitung waktu autentikasi berhasil, pesan EAP dilihat pada *supplicant* dengan menggunakan *tools* Wireshark. Gambar 2.8 merupakan diagram percakapan EAP-PEAP dimana digunakan sebagai ilustrasi untuk proses perhitungan waktu autentikasi berhasil.



Gambar 2.8 Diagram percakapan EAP-PEAP

Untuk contoh hasil *capture* waktu dari *tool* Wireshark, yang nantinya akan digunakan sebagai contoh untuk perhitungan waktu autentikasi dapat dilihat pada Gambar 2.9.

No.	Time	Source	Destination	Protocol	Length	Info
86	86.920368	Dell_06:7a:6a	Nearest	EAP	44	Response, Identity
87	87.020971	Cisco_b6:e2:08	Nearest	EAP	60	Request, Protected EAP (EAP-PEAP)
88	87.021440	Dell_06:7a:6a	Nearest	TLSv1	216	Client Hello
89	87.033442	Cisco_b6:e2:08	Nearest	TLSv1	173	Server Hello, Change Cipher Spec, Encrypted Handshake Message
90	87.036370	Dell_06:7a:6a	Nearest	TLSv1	87	Change Cipher Spec, Encrypted Handshake Message
91	87.046817	Cisco_b6:e2:08	Nearest	TLSv1	125	Application Data
92	87.047216	Dell_06:7a:6a	Nearest	TLSv1	61	Application Data
93	87.054426	Cisco_b6:e2:08	Nearest	TLSv1	61	Application Data
94	87.054705	Dell_06:7a:6a	Nearest	TLSv1	77	Application Data
95	87.063074	Cisco_b6:e2:08	Nearest	TLSv1	77	Application Data
96	87.063342	Dell_06:7a:6a	Nearest	TLSv1	77	Application Data
97	87.078971	Cisco_b6:e2:08	Nearest	TLSv1	93	Application Data
98	87.080061	Dell_06:7a:6a	Nearest	TLSv1	141	Application Data
99	87.089905	Cisco_b6:e2:08	Nearest	TLSv1	109	Application Data
100	87.090507	Dell_06:7a:6a	Nearest	TLSv1	61	Application Data
101	87.100376	Cisco_b6:e2:08	Nearest	TLSv1	125	Application Data
102	87.101249	Dell_06:7a:6a	Nearest	TLSv1	125	Application Data
105	88.150678	Cisco_b6:e2:08	Nearest	EAP	60	Success

Gambar 2.9 Capture protokol EAP-PEAP berhasil pada *tool* Wireshark

Waktu autentikasi berhasil yang dihitung berisi waktu yang dibutuhkan *supplicant* untuk melakukan autentikasi di jaringan. Menurut Kothaluru dan Mecca (2012) rumus yang digunakan untuk menghitung waktu autentikasi adalah:

$$T_{Waktu\ Autentikasi} = T_{Waktu\ EAP\ success} - T_{Respon\ Identity}$$

Dimana:

$T_{Waktu\ Autentikasi}$ = Total waktu autentikasi

$T_{Waktu\ EAP\ success}$ = Pesan EAP di terakhir

$T_{Respon\ Identity}$ = Pesan EAP disaat *supplicant* mulai mengirimkan kredensial autentikasi

Menurut Kothaluru dan Mecca (2012) untuk memvalidasi hasil dari waktu autentikasi berhasil, dilakukan percobaan sebanyak 15 percobaan autentikasi. Nilai rata – rata 15 percobaan tersebut dilakukan perhitungan. Nilai rata-rata diambil dan dijadikan nilai yang mewakili host-mode tersebut dengan menggunakan rumus:

$$T_{Rata-rata} = \frac{\text{jumlah dari total percobaan waktu autentikasi}}{N}$$

Dimana:

N = Jumlah percobaan yang digunakan

2.7.2 Waktu Autentikasi Tidak Berhasil

Untuk menghitung waktu autentikasi tidak berhasil, pesan EAP dilihat pada *supplicant* dengan menggunakan *tool* Wireshark. Untuk contoh hasil *capture* waktu dari *tool* Wireshark, yang nantinya akan digunakan sebagai contoh untuk perhitungan waktu autentikasi dapat dilihat pada Gambar 2.10.

No.	Time	Source	Destination	Protocol	Length	Info
48	16.581593	Elitegro_b7:5f:4d	Nearest	EAP	48	Response, Identity
49	16.588171	Cisco_b6:e2:02	Nearest	EAP	60	Failure

> Frame 49: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Cisco_b6:e2:02 (00:38:df:b6:e2:02), Dst: Nearest (01:80:c2:00:00:03)
> 802.1X Authentication
> Extensible Authentication Protocol

Gambar 2.10 Capture autentikasi tidak berhasil pada *tool* Wireshark

Waktu autentikasi tidak berhasil yang dihitung berisi waktu yang dibutuhkan *supplicant* untuk melakukan autentikasi di jaringan. Menurut Kothaluru dan Mecca (2012) rumus yang digunakan untuk menghitung waktu autentikasi tidak berhasil adalah:

$$T_{Waktu\ Autentikasi} = T_{Waktu\ EAP\ Failure} - T_{Respon\ Identity}$$

Dimana:

$T_{Waktu\ Autentikasi}$ = Total waktu autentikasi

$T_{Waktu\ EAP\ Failure}$ = Pesan EAP di terakhir

$T_{Respon\ Identity}$ = Pesan EAP disaat *supplicant* mulai mengirimkan kredensial autentikasi

Menurut Kothaluru dan Mecca (2012) untuk memvalidasi hasil dari waktu autentikasi tidak berhasil, dilakukan percobaan sebanyak 15 percobaan autentikasi. Nilai rata – rata 15 percobaan tersebut dilakukan perhitungan. Nilai rata-rata diambil dan dijadikan nilai yang mewakili host-mode tersebut dengan menggunakan rumus:

$$T_{Rata-rata} = \frac{\text{jumlah dari total percobaan waktu autentikasi}}{N}$$

Dimana:

N = Jumlah percobaan yang digunakan