BAB II
LANDASAN TEORI

2.1 Autentikasi User Secara Hierarki

Pada bab ini menjelaskan definisi dari autentikasi, metode keamanan yang digunakan serta menjelaskan tentang *multi-factor authentication*.

2.1.1 Definisi Autentikasi

Autentikasi adalah suatu metode untuk menentukan atau memastikan bahwa seseorang (atau sesuatu) adalah asli atau benar. Adapun proses *validasi user* pada saat memasuki sistem yaitu nama dan *password* dari *user* melalui proses pengecekan *user* pada suatu database yang diregistrasi sebelumnya oleh *user* itu sendiri. Pada sistem komputer, autentikasi biasanya terjadi pada saat *login* atau permintaan akses.

Selain itu autentikasi juga merupakan salah satu dari banyak metode yang digunakan untuk membuktikan bahwa dokumen tertentu yang diterima secara elektronik asli datang dari orang yang bersangkutan dan tidak berubah keasliannya, dengan cara mengirimkan suatu kode tertentu melaui *e-mail* kemudian pemilik *e-mail* membalas *e-mail* tersebut.

Autentikasi server berfungsi untuk mengenali *user* yang berintegrasi ke jaringan dan memuat semua informasi dari *user* tersebut. Dalam praktek biasanya autentikasi server mempunyai database dengan fungsi untuk menjaga server jika suatu saat ada masalah, segala informasi di dalamnya tidak akan terganggu.

Dalam aplikasi Web dibutuhkan mekanisme yang dapat melindungi data dari para *hacker*, misalnya sebuah situs Web berisikan foto-foto keluarga yang hanya dapat diakses sesama anggota keluarga itu sendiri. Mekanisme ini dapat diimplementasikan dalam bentuk sebuah proses login yang terdiri dari tiga buah tahapan yaitu identifikasi, autentikasi dan otorisasi.

Proses autentikasi pada prinsipnya berfungsi sebagai kesempatan pengguna dan pemberi layanan dalam proses pengaksesan *resource*. Pihak pengguna harus bersedia memberikan segala informasi yang dibutuhkan pemberi layanan agar

berhak mendapatkan *resource*-nya. Sedangkan pihak pemberi layanan harus menjamin bahwa pihak yang tidak berhak tidak dapat mengakses *resource* tersebut (Pramartha, 2013). Ada beberapa metode autentikasi yang digunakan untuk memverifikasi data pengguna, yaitu akan dijelaskan pada subbab 2.1.2 berikut ini.

2.1.2 Metode-Metode Autentikasi

Metode autentikasi bisa dilihat dalam 4 kategori metode (Pramartha, 2013):

a. *Something you know*

Metode ini adalah metode yang paling lazim karena menggunakan kerahasiaan informasi, contohnya adalah password dan PIN (*Personal Identification Number*). Cara ini berasumsi bahwa tidak ada yang mengetahui rahasia dari informasi tersebut kecuali si pemilik sendiri.

b. *Something you have*

Metode ini merupakan faktor tambahan untuk membuat autentikasi menjadi lebih aman dengan menggunakan barang contohnya *ID card*, kartu kredit, telepon seluler, perangkat token dan sebagainya. Hanya pemilik dari barang tersebut yang berasumsi bahwa segala rahasia yang dimilikinya pasti aman.

c. *Something you are*

Metode yang paling jarang digunakan karena faktor teknologi dan manusia. Cara ini menggunakan bagian tubuh yang tidak mungkin sama dengan bagian tubuh orang lain seperti sidik jari, DNA, suara, pola retina, atau aspek biometrik lain.

d. *Something you do*

Berasumsi bahwa setiap *user* pasti berbeda dalam melakukan sesuatu hal. Contoh : Penggunaan analisis suara (*voice recognation*), dan analisis tulisan tangan.

Dari keempat metode diatas yang paling lazim adalah menggunakan *password*. Metode autentikasi dengan menggunakan *password* statis adalah yang lebih sering digunakan. Tetapi jika setiap *user* menggunakan *password* statis atau

dengan *password* yang sama masuk ke dalam suatu sistem berulang kali maka *password* tersebut akan sangat berbahaya terhadap para *hacker*. Namun ada sebuah sistem dibuat untuk mengatasi serangan tersebut yaitu dengan sistem autentikasi *One Time Password* (OTP) .

Sedangkan, beberapa metode autentikasi lain yang tidak sering digunakan antara lain:

- Berbasis pengenalan (*recognition*) atau autentikasi *cognometric*, yaitu sesuatu yang dikenal oleh *user*. Contohnya *user* harus mengenali dari beberapa wajah yang dirahasiakan.
- Berbasis *cybermetric*, yaitu sesuai yang ada pada komputer. Contohnya adalah membatasi suatu akses hanya dari komputer yang memiliki kombinasi unik antara *hardware* dan *software* tertentu saja.
- Berbasis lokasi, contohnya adalah penggunaan ATM atau kartu kredit hanya untuk cabang tertentu, membatasi *login root* hanya dari terminal tertentu.
- Berbasis waktu. Contohnya adalah membatasi penggunaan sebuah *account* hanya pada waktu tertentu saja, misalnya jam kerja.
- Berbasis ukuran. Contohnya adalah membatasi terjadinya transaksi hanya untuk sejumlah tertentu saja.

Para *hacker* akan lebih sulit mengakses sistem dan mendapatkan informasi yang bersifat rahasia jika metode autentikasi lebih ditingkatkan. Pengguna harus membuktikan bahwa pengguna tersebut memiliki akses secara fisik (contoh: paspor, token, kartu autentikasi) atau yang bersifat unik ditubuh pengguna (contoh: sidik jari, retina, bentuk wajah) (Pramartha, 2013). Terdapat faktor yang mengkombinasi dua atau lebih metode keamanan dalam proses autentikasi, kemudian akan dijelaskan pada subbab 2.1.3.

2.1.3 Multi-Factor Authentication

Autentikasi *user* menggunakan dua atau lebih metode dengan tujuan mampu meningkatkan keamanan disebut *multi-factor authentication*. *Multi-factor authentication* menyediakan metode keamanan tambahan dalam proses autentikasi.

Metode tambahan ini mengurangi peluang para *hacker* untuk mencoba memasuki sistem komputer. Selain mendapatkan keuntungan, mekanisme ini juga memiliki kelemahan sebagai berikut (Pramartha, 2013);

a. Kelebihan multi faktor autentikasi

- Autentikasi terjamin keamanannya karena paket data terlindungi oleh sistem keamanan berlapis.
- Keaslian pesan (*data integrity*) tetap terjaga tanpa mengalami perubahan atau modifikasi.

b. Kelemahannya:

- Situasi diluar perkiraan misalnya di mana seorang *user* ternyata tidak dapat melakukan autentikasi untuk dirinya sendiri ke server sistem komputer karena telah kehilangan *smart card.*
- *Multi-factor authentication* juga menyebabkan pengeluaran biaya perawatan yang lebih pada sistem komputer. Hal ini terjadi karena lebih banyak *hardware* yang dibutuhkan untuk mengimplementasikan proses autentikasi.

Hierarchical Design Model diperlukan dalam autentikasi *user* secara hierarki agar dapat membantu meringkas perancangan sebuah *internetworks*. Subbab 2.2 akan menjelaskan garis besar dari *Hierarchical Design Model.*

2.2 Hierarchical Design Model

Menurut CCNA Study Guide (Lammle, 2000), model berbentuk hierarki (*Hierarchical model*) memungkinkan untuk merancang sebuah *internetworks*. Sebuah *network* yang besar akan menjadi sangat rumit dengan protokol yang banyak, konfigurasi yang detail dan teknologi yang beragam. Dengan model hierarki kerumitan itu menjadi sebuah model yang mudah dimengerti, sehingga mempermudah dalam mendesain dan membangun jaringan terskala. Seperti menggunakan *tool access list* pada level tertentu di sebuah *network* yang hierarki dan tidak menggunakannya di level lain.

Manfaat menggunakan model hierarki untuk desain jaringan meliputi;

- Penghematan biaya.
- Memudahkan pemahaman perancangan.
- Mengembangkan jaringan dengan mudah karena berbasis modulasi.
- Memudahkan perawatan (*maintenance*) saat perbaikan jaringan.

Adapun lapisan - lapisan yang membagi hierarki jaringan computer (Lammle, 2000) yaitu :

1. *Core Layer* (Lapisan Inti)

Lapisan inti adalah lapisan yang bertanggung jawab untuk memindahkan lalu lintas data yang besar secara cepat, karena menunjukkan karakteristik jaringan bagi sebuah perusahaan.

Ketika jaringan menggunakan perangkat router, jumlah lompatan (hop) antara router yang satu dengan yang lainnya disebut diameter. Sebagai catatan, sangat dibutuhkan sebuah diameter yang konsisten diantara hierarki jaringan. Perjalanan dari satu *node* ke *node* yang lain diantara *backbone*, harus memiliki jumlah hop yang sama. Jarak dari *node* akhir ke server dalam *backbone* juga harus konsisten.

2. *Distribution Layer* (Lapisan Distribusi)

Lapisan distribusi di dalam jaringan adalah titik komunikasi diantara jaringan lapisan akses dan lapisan inti. Fungsi utama dari lapisan distribusi adalah menyediakan *routing*, *filtering*, dan akses WAN. Lapisan distribusi harus menentukan cara terbaik untuk menangani permintaan layanan jaringan, sebagai contoh bagaimana permintaan untuk sebuah *file* diteruskan ke sebuah server. Setelah lapisan distribusi menentukan lintasan terbaik, kemudian akan meneruskan permintaan tersebut lapisan inti.

3. *Access Layer* (Lapisan Akses)

Lapisan akses mengendalikan akses *user* dan *workgroup* ke sumber daya *internetwork*. Lapisan akses disebut juga sebagai lapisan desktop. Sumber daya jaringan yang diperlukan *user* akan tersedia secara lokal.

2.3 Komponen dan Jenis Keamanan Autentikasi

Dibawah ini adalah proses untuk menjelaskan peran setiap komponen yang digunakan selama proses autentikasi dan jenis keamanan autentikasi.

2.3.1 Wi-Fi Protected Access (WPA)

Wi-Fi Alliance pada tahun 2003 membuat metode keamanan baru untuk melengkapi dari standard sebelumnya, yaitu *Wired Equivalent Privacy* (WEP) yang membantu mengurangi kelemahan keamanan pada jaringan. WPA menyediakan enkripsi data yang lebih canggih dari WEP dan juga menyediakan autentikasi pengguna berdasarkan 802.1x dan protokol *Extensible Authentication Protocol* (EAP). Metode enkripsi WPA ini menggunakan *Temporal Key Integrity* Protocol (TKIP). Untuk mendapatkan akses ke suatu jaringan yang dilindungi, *user* harus memasukkan kata sandi atau *password* sesuai pada *wireless* yang telah diatur. Setelah itu semua komunikasi antara perangkat *wireless* dan perangkat *wireless* yang dimiliki *user* terenkripsi dengan baik (S'to, 2015).

WPA memiliki 2 versi, salah satunya adalah tipe perusahaan yang memerlukan autentikasi terhadap radius dengan menggunakan mekanisme *username* dan *password*. Satu tahun setelah WPA dirilis, protokol WPA2 menjadi lebih kuat untuk metode keamanan *wireless*. Pengamanan jaringan nirkabel dengan metode WPA ini, minimal ada tiga pilihan yang harus diisi administrator jaringan agar jaringan tersebut dapat beroperasi. Ketiga menu yang harus diisi tersebut adalah (Muhammad, 2015):

1. Server

Komputer server yang mengarah ke *access point* kemudian memberi autentikasi kepada *client*. Beberapa perangkat lunak yang sering digunakan antara lain *freeRADIUS, openRADIUS* dan lain-lain.

2. *Authentication Port*

Port diasumsikan sebagai pintu *service* atau dapat diartikan sebagai sebuah nilai yang ditetapkan untuk mengindentifikasi sebuah layanan. Nomor *port* yang digunakan dalam autentikasi hierarki pada penelitian ini adalah 1812.

3. *Shared Secret*

Shared Secret adalah kunci yang akan dibagikan ke komputer dan juga kepada *client* secara transparan.

2.3.2 Wi-Fi Protected Access versi 2 (WPA2)

WPA2 merupakan jenis keamanan yang ditingkatkan dari WPA untuk semua *hardware* Wi-Fi bersertifikat sejak tahun 2006. WPA2 memberikan keamanan dan enkripsi untuk transmisi data dan konektivitas komputer pada umumnya. Selain TKIP, WPA2 memiliki metode enkripsi yang jauh lebih berkembang yaitu *Advanced Encryption Standard* (AES). Pada perangkat *access point*, apabila memilih metode keamanan WPA2 maka secara otomatis enkripsi yang digunakan adalah AES. Seperti WPA, WPA2 dirilis dalam dua versi yaitu pribadi dan perusahaan (S'to, 2015).

Menimbang dari semua fitur yang ada, WPA2 adalah pilihan terbaik untuk pengaturan keamanan *access point* dalam proses autentikasi di radius server. Pengguna (*supplicant*) dan protokol autentikasi server 802.1X bekerja sama untuk meminta *user* memberikan informasi *credentials*.

Konsep WPA/WPA2-*Enterprise* ada 3 bagian yang terlibat (Wi-Fi Alliance, 2006) yaitu :

1. *Supplicant*

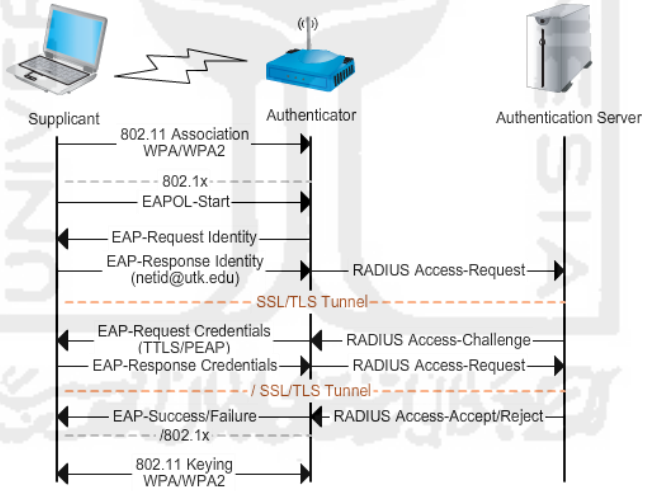
Supplicant adalah perangkat lunak yang terdapat di dalam perangkat komputasi *user* dengan berdasar pada protokol IEEE 802.1X dan menggunakan protokol EAP untuk mengirim informasi autentikasi. *Supplicant* sering dibangun ke dalam sistem operasi, tetapi juga menjadi program yang terpisah. Perangkat lunak *supplicant* pada perangkat *user* harus terlebih dahulu dikonfigurasi sebelum melakukan seluruh proses autentikasi.

2. *Authenticator*

Authenticator dalam hal ini adalah *device* yang memproses *supplicant* dapat mengakses jaringan atau tidak. *Device* yang dimaksud adalah *access point* dengan berdasar pada protokol 802.1X. Protokol adalah suatu kumpulan dari banyak aturan yang berhubungan dengan komunikasi data antara alat-alat komunikasi dari data

tersebut supaya komunikasi data dilakukan dengan benar. Jabatan tangan merupakan contoh dari protokol antara dua manusia yang akan berkomunikasi. Dalam istilah komputer, jabatan tangan atau disebut *handshaking* menunjukkan suatu protokol dari adanya komunikasi data bila dua buah alat dihubungkan satu dengan yang lainnya. Tujuannya adalah untuk menentukan bahwa keduanya telah kompatibel (Jogiyanto, 1999).

802.1X adalah standar *Institute of Electrical and Electronics Engineers* (IEEE) untuk jaringan lokal dan metropolitan. IEEE adalah sebuah *port* berdasarkan pada protokol jaringan *access control*. Awalnya dirancang untuk *port Ethernet*, lalu diperluas menjadi jaringan *wireless*. Standard 802.1X mendefinisikan penggunaan EAP (*Extensible Authentication Protocol*) untuk keperluan autentikasi dalam komunikasi jaringan *wireless* atau *point-to-point* karena memungkinkan mekanisme autentikasi yang berbeda. **Gambar 2.1** dibawah ini menunjukkan langkah autentikasi pada 802.1X.



Gambar 2.1 Langkah Autentikasi Pada 802.1x (Hagley, 2011)

Tahapan dari proses autentikasi dari **Gambar 2.1** adalah sebagai berikut;

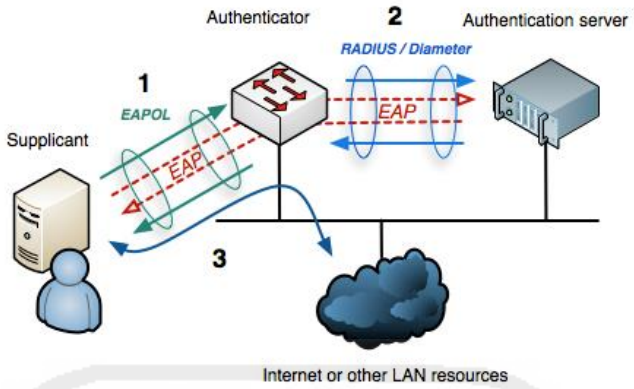
1. Ketika *authenticator* mendeteksi adanya *supplicant*, *authenticator* meminta identitas dari *supplicant*.
2. Setelah *supplicant* menanggapi permintaan *authenticator* untuk mengirim identitas, disinilah proses autentikasi dimulai. Protokol yang digunakan antara *supplicant* dan *authenticator* adalah EAP *Over Local Area Network* (EAPOL).

3. *Authenticator* kembali merangkum pesan EAP ke dalam format radius kemudian meneruskannya ke *authentication server* yaitu radius.
4. Server radius melakukan evaluasi *Access-Request*, apabila server menerima identitas yang berbeda dari *user* atau membutuhkan informasi lebih lanjut seperti meminta *credentials user*, server akan mengeluarkan paket *access-challenge.*
5. Setelah paket *access-challenge* dikeluarkan, *authenticator* meminta *supplicant* untuk memberikan informasi lebih lanjut atau *credentials user*.
6. *Supplicant* menanggapi permintaan tersebut dan kemudian tugas *authenticator* melempar informasi yang diberikan ke radius server untuk dilakukan proses autentikasi.
7. Ketika autentikasi telah selesai, server mengirim kembali ke *authenticator* apakah autentikasi berstatus *access-acccept* (autentikasi berhasil) atau *access-reject* (autentikasi gagal) ke *supplicant*.
8. Jika autentikasi sukses, *supplicant* diijinkan mengakses ke sumber jaringan yang dilindungi. Selama proses autentikasi, *authenticator* hanya sebagai perantara untuk menyampaikan paket antara *supplicant* dan radius server.

Terdapat dua jenis protokol untuk metode autentikasi, yaitu:

- EAP-*Tunneled Transport Layer Security* (EAP-TTLS)

EAP-TTLS adalah sebuah protokol EAP yang sama kuat dengan protokol EAP-*Transport Layer Security* (EAP-TLS) dikembangkan oleh *Funk Software dan Certicom* dan keseluruhan didukung oleh semua perangkat lunak. Protokol ini digunakan oleh radius server untuk melakukan autentikasi *client*. Keuntungan dari jenis metode autentikasi ini adalah jalur akses selama autentikasi dipastikan aman dan mencegah terbongkarnya *credentials user* (Rahman, 2011).



Gambar 2.2 Proses Metode Autentikasi EAP-TTLS (Cudbard, 2010)

- Protokol EAP *Microsoft Challenge Handshake Authentication* (PEAP-MSCHAP)

Protected Extensible Authentication Protocol (PEAP) adalah protokol yang menggabungkan komunikasi EAP dalam jalur akses TLS yang dienkripsi. PEAP dirilis untuk lebih meningkatkan perlindungan dari komunikasi EAP dan dikembangkan oleh *Cisco Systems*, *Microsoft* dan *RSA Security*.

Meskipun EAP digunakan dalam autentikasi, seluruh percakapan EAP dikirim sebagai teks yang tidak terenkripsi. Kondisi seperti ini tidak menjamin karena *user* yang tidak memiliki hak akses dapat masuk ke media transmisi dan menangkap pesan EAP dari autentikasi sukses.

Untuk mengatasi masalah keamanan ini, PEAP pertama menciptakan sebuah *channel* pengamanan yang mana keduanya dienkripsi dan dilindungi dalam hal integritas data yaitu dengan *Transport Layer Security* (TLS). Hal ini untuk mencegah dirusaknya paket percakapan EAP.

Dengan PEAP *tunnel*, berbagai jenis metode EAP digunakan salah satunya adalah *Microsoft Challenge Handshake Authentication Protocol* (MS-CHAP).

3. *Authentication Server / RADIUS*

Authentication Server untuk autentikasi secara hierarki menggunakan Protokol *Remote Authentication Dial In User Service Server* (RADIUS). Radius adalah protokol jaringan yang menyediakan *Authentication, Authorization,*

*Accounting (*AAA) yaitu protokol keamanan untuk memungkinkan *client* memiliki akses ke sumber jaringan yang dilindungi. Pertama kali dikembangkan oleh *Livingstone Enterprises Inc*. kemudian menjadi standar *Internet Engineering Task Force* (IETF) (Cisco, 2006).

Radius adalah layanan *client* server yang menggunakan transportasi *User Datagram Protocol* (UDP) dan berjalan pada *application layer.* UDP memiliki tiga fungsi utama, yaitu:

- *Authentication user* : Sebelum memberikan akses ke jaringan.
- *Authorization user* : Dikonfirmasi untuk memiliki akses ke sumber jaringan tertentu.
- *Accounting User* : Menunjukkan jumlah sumber daya (seperti waktu, paket, *byte*, dan sebagainya) yang digunakan selama pemakaian koneksi internet.

2.3.3 Protokol Keamanan AAA

Menurut Hassel (2002) konsep kerja autentikasi server dikenal dengan AAA (*Authentication, Authorization, and Accounting*) yang terdiri dari autentikasi, otorisasi, dan pendaftaran akun pengguna. Pada konsep AAA ada tiga aspek dalam mengontrol akses *user*, masing-masing memiliki fungsi sebagai berikut;

a. Authentication

Autentikasi adalah proses verifikasi untuk menyatakan kebenaran suatu identitas. Mekanisme untuk melakukan autentikasi menggunakan kombinasi login ID/*username* dan *Password.* Jika kombinasi keduanya benar maka *user* dapat mengakses ke sumber daya jaringan yang dilindungi. Proses autentikasi dapat dianalogikan seperti seorang tamu yang datang ke rumah seorang *user*, sebelum tamu tersebut diperbolehkan masuk tentu *user* harus mengetahui tamu itu terlebih dahulu, jika *user* kenal dengan tamu tersebut maka tamu dipersilakan masuk. Namun, tamu akan ditolak jika *user* tidak mengenalinya.

b. Authorization

Proses *authorization* merupakan lanjutan dari proses *authentication*. Otorisasi mengidentifikasi aturan-aturan yang berlaku untuk memutuskan aktifitas apa saja yang diijinkan dalam sistem atau sumber daya jaringan tertentu pada pengguna yang terautentikasi. Analogi proses otorisasi adalah jika *user* sudah mengijinkan tamu untuk masuk ke rumah *user*, tentu *user* mempunyai aturan-aturan yang berlaku di dalam rumah. Misalnya tamu hanya boleh masuk sampai ruang tamu saja. Dengan aturan seperti ini tentu akan memudahkan seseorang untuk mengontrol terhadap sumber daya jaringan tertentu.

c. Accounting

Proses *accounting* merupakan proses pencatatan berapa lama pemakaian akses internet seorang pengguna yang terkoneksi (waktu mulai/waktu stop). Data dan informasi ini sangat berguna baik untuk pengguna maupun administrator. *Accounting* bermanfaat untuk melakukan pemeriksaan, membuat laporan pemakaian, membaca karakteristik jaringan, dan membuat tagihan pembayaran. Dapat disimpulkan proses *accounting* berguna untuk mengetahui apa saja yang dilakukan oleh *client* dan *server*. Analogi sederhananya yaitu mesin absensi di kantor, mesin akan mencatat waktu datang dan waktu pulang. Dengan demikian petugas dapat mengawasi karyawan dengan mudah (Hassel, 2002).

2.3.4 *Realm*

Realm dalam istilah radius adalah suatu penamaan *username*@domain. Domain atau DNS (*Domain Name Server)* dalam hal ini menggunakan nama organisasi misalnya "ugm.ac.id" atau "uny.ac.id". Jadi format *username* untuk autentikasi radius yaitu "*username@realm*". Format tersebut bukan sebagai alamat *e-mail* namun format ini sangat penting untuk proses autentikasi hierarki. Oleh karena itu tanpa menggunakan format, permintaan tidak dapat diidentifikasi dan diteruskan ke universitas asal sesuai *realm*-nya (GEANT Association, 2015).

2.3.5 LDAP (Lightweight Directory Access Protocol)

LDAP (*Lightweight Directory Access Protocol*) adalah sebuah aplikasi protokol yang digunakan untuk melakukan *query* atau memodifikasi data layanan direktori dan diimplementasikan dalam bentuk IP (*Internet Protocol*). LDAP menyediakan layanan *Active Directory*, yang berfungsi menyimpan konfigurasi jaringan baik berupa *user*, *group*, komputer, *hardware*, serta berbagai aturan keamanan dalam satu database terpusat. Elemen dasar dari *Active Directory* adalah *Active Directory Object*. Sebuah *Active Directory Object* ini dapat berupa sebuah akun pengguna, komputer yang tergabung ke dalam sebuah domain LDAP Server, printer, aplikasi, folder, atau sumber daya lainnya di dalam jaringan. Setiap objek memiliki atributnya masing-masing berupa properti yang umumnya bersifat unik (tergantung jenis objek tersebut). Sebagai contoh, atribut yang dapat dimiliki oleh objek akun pengguna mencakup nama pertama, nama akhir, alamat *e-mail*, nomor mahasiswa, dan nomor telepon. Beberapa atribut lainnya memiliki nilai yang telah ditentukan oleh sistem dan atribut lainnya dapat didefinisikan secara manual (atau dapat dikosongkan). *Active Directory* juga memiliki peraturan-peraturan untuk menata objek mana saja yang dapat disimpan di dalam direktori dan atribut mana saja yang dapat dimiliki oleh objek tersebut. Peraturan-peraturan tersebut, dinamakan juga dengan *Active Directory Schema*.

Sebuah jenis objek khusus yang dapat disimpan di dalam *Active Directory* adalah sesuatu yang disebut sebagai *Organizational Unit* (OU). OU adalah sebuah jenis objek *Active Directory* yang dapat mengandung objek lainnya, seperti sebuah akun pengguna, komputer, aplikasi atau mengandung objek OU lainnya. Administrator juga dapat mengontrol akses atau perijinan pada setiap anak pohon dalam sebuah OU, agar hanya objek-objek tertentu saja yang dapat mengaksesnya. OU ditampung dalam sebuah *entry*, yang merupakan struktur dasar dari *Active Directory* (pada kenyataannya, *Active Directory* tidak akan berjalan tanpa adanya domain). Setiap objek di dalam *Active Directory* harus termasuk ke dalam sebuah domain yang sama (Zeilenga, 2006).