

BAB III

LANDASAN TEORI

3.1 LDAP

3.1.1 Pengertian LDAP

Menurut Cartealy (2013, p75), *Lightweight Directory Access Protocol* (LDAP) merupakan protokol yang mendefinisikan bagaimana data *directory* dapat diakses melalui jaringan. LDAP biasa digunakan untuk menyimpan berbagai informasi terpusat yang dapat diakses oleh berbagai macam mesin atau aplikasi dari jaringan. Penggunaan LDAP di dalam sistem akan membuat pencarian informasi menjadi terintegrasi dan sangat mudah. sebagai contoh, LDAP seringkali digunakan untuk menyimpan nama pengguna dan sandi yang terdapat di dalam sistem secara terpusat.

Seperti disebutkan sebelumnya, LDAP merupakan protokol yang hanya mengatur bagaimana caranya mengakses informasi yang tersimpan sebagai direktori melalui jaringan, bukan bagaimana menyimpan informasi. Semua aplikasi *server* LDAP memiliki protokol pengaksesan direktori yang sama, walaupun mungkin memiliki sistem penyimpanan direktori yang berbeda. Hal ini menyebabkan banyak *vendor* membuat produk yang mendukung LDAP, karena *vendor* tersebut hanya perlu membuat produk yang dapat mengerti bagaimana caranya mengakses informasi LDAP. *Vendor* tidak perlu khawatir sistem apa yang digunakan sebagai penyimpanan direktori. Oleh karena itu, LDAP bisa dikatakan terbebas dari *platform* yang digunakan.

LDAP pada dasarnya menyimpan informasi sama halnya seperti yang dilakukan oleh aplikasi *database* relasional seperti MYSQL. LDAP juga memproses permintaan dan perubahan terhadap data atau informasi seperti yang dilakukan oleh berbagai aplikasi RDBMS sehingga LDAP sebenarnya merupakan aplikasi *database*. Namun LDAP bukanlah merupakan aplikasi *database* relasional. LDAP tidak dioptimasi untuk memproses perubahan data dalam skala besar dan cepat secara nyata. LDAP dioptimasi untuk mencari informasi secara cepat.

Jika aplikasi RDBMS sangat mementingkan keakuratan data secara *real-time* tidak halnya dengan LDAP. LDAP tidak terlalu mementingkan keakuratan data. LDAP lebih mementingkan perform *read* yang sangat cepat. Ketidakakuratan data selama beberapa saat sebagai akibat perubahan informasi, bukanlah masalah yang sangat penting bagi LDAP. Oleh karena itu LDAP tidak tepat jika digunakan sebagai tulang punggung penyimpanan informasi aplikasi e-commerce yang sangat sibuk, tetapi LDAP akan sangat cocok untuk digunakan untuk menyimpan informasi yang sangat jarang berubah atau tidak pernah berubah sama sekali.

Server LDAP dapat mereplikasi informasi ke beberapa *server* LDAP lain dengan cara menarik (*pull*) informasi dari *server* utama ataupun sebaliknya. *Server* utama yang mendorong (*push*) informasi ke *server* lainnya sehingga didalam satu sistem mungkin terdapat beberapa *server* LDAP yang saling terkoneksi. Selain itu, LDAP juga dapat mendelegasikan beberapa informasi tertentu ke *server* lain seperti halnya yang dilakukan oleh DNS. Kemampuan –

kemampuan tersebut merupakan keunggulan LDAP dibandingkan aplikasi RDBMS karena semua kemampuan tersebut merupakan protokol LDAP *standart*.

3.1.2 Terminologi LDAP

Menurut Cartealy (2013, p76), LDAP merupakan salah satu protokol yang memiliki terminologi yang sangat banyak dan sangat membingungkan. Terminologi tersebut harus dipahami untuk dapat mengerti bagaimana LDAP bekerja namun jangan khawatir karena hanya beberapa terminologi saja yang sangat berguna dan sudah cukup untuk dapat membuat LDAP berfungsi.

Sesuai namanya, LDAP tersusun atas direktori. Direktori merupakan kumpulan dari objek dengan atribut yang tersusun secara hirarki. Secara gampangnya, direktori merupakan kumpulan dari informasi yang tersusun secara hirarki.

Data atau informasi di LDAP disimpan di dalam struktur berjenjang seperti pohon dan setiap objek tersebut disebut *entri*. Struktur berjenjang berbentuk pohon tersebut secara definisi disebut dengan *Data Information Tree* (DIT). Di level teratas, terdapat objek yang disebut *root* atau *sufik*, dan objek lainnya disimpan didalam objek *root* tersebut.

Setiap entri yang terdapat di struktur pohon mempunyai hanya satu entri atau objek induk. Setiap entri tersebut dapat tidak memiliki entri turunan (*child entry*) ataupun dapat memiliki satu atau lebih entri turunan. Setiap entri yang memiliki induk yang sama merupakan entri saudara (*sibling*).

Setiap entri merupakan contoh dari satu atau lebih kelas objek (*object class*). Kelas objek terdiri dari nol atau lebih atribut. Atribut memiliki nama dan nilai.

Ada tiga definisi yang sangat penting di LDAP, yaitu skema, kelas objek dan atribut. Ketiganya saling terkait dan menjadi tulang punggung LDAP.

3.1.3 Skema LDAP

Menurut Cartealy (2013, p77), skema bisa diibaratkan seperti sistem pemaketan untuk kelas objek dan atribut. Setiap kelas objek dan atribut harus didefinisikan di dalam skema, dan skema tersebut harus di deklarasikan di dalam berkas konfigurasi daemon slapd, **slapd.conf**.

Skema menentukan informasi yang dapat disimpan di dalam LDAP, sehingga tidak sembarang informasi dapat disimpan di LDAP. Semua kelas objek dan atribut harus didefinisikan di dalam skema, termasuk hubungan antara kelas objek dan atribut, ataupun sifat objek kelas atau atribut tersebut. Semua informasi tersebut harus didefinisikan di dalam skema. Setiap skema hanya mengakomodasikan objek kelas dan atribut untuk tujuan tertentu, sebagai contoh skema samba ditujukan untuk mengakomodasi informasi yang dibutuhkan oleh samba.

Jika membutuhkan suatu informasi tertentu untuk disimpan di dalam LDAP, maka harus menggunakan skema yang sesuai. Sebagai contoh, jika ingin menyimpan informasi nama pengguna dan sandinya dengan menggunakan atribut **uid** dan **userPassword**, maka harus menggunakan skema *core*.

Kini banyak sekali jenis skema yang beredar di dunia maya, sehingga tidak perlu membuat skema untuk mengakomodasi kebutuhan. Yang perlu dilakukan hanyalah mencari skema yang tepat, dan menggunakannya. Tetapi tentu saja ada kemungkinan jika mempunyai kebutuhan khusus, maka terpaksa harus membuat skema yang dibutuhkan.

Secara default OpenLDAP akan mengikutsertakan skema sebagai berikut :

- core.schema Merupakan skema inti OpenLDAP.
- cosine.schema Merupakan skema untuk Cosine dan X.500.
- nis.schema Merupakan skema untuk NIS.
- inetorgperson.schema Merupakan skema untuk InetOrgPerson.

3.1.4 Kelas Objek LDAP

Menurut Cartealy (2013, p78), kelas objek merupakan *container* yang berfungsi untuk mengelompokan atribut. Kelas objek akan menentukan apakah suatu atribut harus ada, atau bersifat pilihan.

Kelas objek bersifat hirarki dan dapat mewarisi atribut dari kelas objek induknya. Sebagai contoh, kelas objek InetOrgPerson merupakan turunan dari kelas objek organizationalPerson dan mewarisi atribut yang dimiliki oleh kelas objek organizationalPerson.

3.1.5 Atribut LDAP

Menurut Cartealy (2013, p80), atribut LDAP merupakan struktur terkecil dari skema dan merupakan anggota kelas objek. Atribut memiliki nama dan juga memiliki nilai dan setiap atribut dapat memiliki lebih dari satu nilai.

Yang perlu diperhatikan, ketika menggunakan LDAP harus menggunakan sebisa mungkin minimal satu atribut yang bersifat unik. Hal ini untuk mempermudah pencarian ataupun perubahan informasi ketika terdapat lebih dari satu nilai atribut yang bersifat sama. Sebagai perumpamaan, sebuah buku telepon bisa berisikan informasi nama, alamat dan nomer telepon. Informasi nama dan alamat mungkin tidak unik, namun nomer telepon merupakan informasi unik yang membedakan suatu entri dengan yang lainnya.

Hal yang sama berlaku ketika menggunakan LDAP untuk menyimpan informasi mengenai pengguna di sistem. Ada baiknya menerapkan aturan ID pengguna tidak boleh sama untuk setiap orang sehingga entri ini dapat digunakan untuk membedakan.

3.2 RADIUS

Menurut Cartealy (2013, p123), RADIUS (*Remote Access Dial in User*) merupakan suatu protokol yang digunakan secara luas untuk otentikasi pengguna jaringan. Secara sederhana, RADIUS menyediakan fungsi otentikasi, otorisasi dan akunting, atau dalam lebih dikenal dengan AAA (*Authhentication, Authorization, and Accounting*) untuk menggunakan layanan jaringan.

RADIUS seringkali digunakan sebagai sentralisasi otentikasi dan akunting untuk pengguna di dalam jaringan. RADIUS pada dasarnya akan

melakukan otentikasi terhadap *client* dan memberikan akses jaringan ke *client* sesuai dengan hak akses *client* tersebut.

RADIUS sendiri tidak menyimpan informasi mengenai *client*, tetapi RADIUS dapat menggunakan berbagai macam backend seperti MYSQL dan LDAP *server* untuk menyimpan informasi tersebut. Yang terjadi ketika RADIUS melakukan otentikasi terhadap *client* adalah RADIUS akan menanyakan informasi seperti sandi ke backend tersebut. RADIUS disini hanya akan melakukan fungsi AAA saja, tetapi informasi yang dibutuhkan oleh RADIUS untuk melakukan fungsi tersebut didapatkan dari berbagai macam *backend* seperti LDAP *server*.

Selain itu, RADIUS juga biasanya bekerja dengan aplikasi *client* RADIUS. Aplikasi inilah yang sebenarnya akan mengirimkan permintaan dari pengguna ke RADIUS. RADIUS-lah yang sebenarnya melakukan fungsi AAA. *Client* RADIUS hanyalah mengirimkan ataupun meneruskan permintaan dari pengguna ke *server* RADIUS.

Dalam membangun *captive portal* RADIUS disini digunakan untuk melakukan otentikasi terhadap pengguna jaringan di jaringan lokal. RADIUS akan menerima permintaan dari *server* untuk melakukan otentikasi pengguna jaringan dan mencari informasi pengguna dan sandinya di *database*.

Salah satu aplikasi RADIUS yang bersifat bebas dan kode sumber terbuka adalah **FreeRADIUS**. Sebenarnya FreeRADIUS bukanlah merupakan aplikasi *server* saja, tetapi merupakan kumpulan aplikasi *server* dan *client* RADIUS yang dikembangkan oleh FreeRADIUS *Project*. FreeRADIUS memiliki banyak kemampuan, salah satunya yaitu menghubungkan *server* dengan *database*.

3.3 LINUX

3.3.1 Sejarah

Menurut Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003 (2003, p206), Linux sangat mirip dengan sistem-sistem UNIX, hal ini dikarenakan kompatibilitas dengan UNIX merupakan tujuan utama desain dari proyek Linux. Perkembangan Linux dimulai pada tahun 1991, ketika mahasiswa Finlandia bernama Linus Torvalds menulis Linux, sebuah kernel untuk prosesor 80386, prosesor 32-bit pertama dalam kumpulan CPU intel yang cocok untuk PC.

Pada awal perkembangannya, *source code* Linux disediakan secara bebas melalui *Internet*. Hasilnya, sejarah Linux merupakan kolaborasi banyak user dari seluruh dunia, semuanya dilakukan secara eksklusif melalui *Internet*. Dari kernel awal yang hanya mengimplementasikan subset kecil dari sistem UNIX, sistem Linux telah bertumbuh dimana sudah mampu memasukkan banyak fungsi UNIX.

Kernel Linux perlu dibedakan dari sebuah sistem Linux: kernel Linux merupakan sebuah perangkat lunak orisinal yang dibuat oleh komunitas Linux sedangkan sistem Linux, yang diketahui saat ini, mengandung banyak komponen yang dibuat sendiri atau dipinjam dari proyek lain.

3.3.2 Distribusi LINUX

Menurut Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003 (2003, p206), siapa pun dapat menginstall sistem Linux, ia hanya perlu mengambil revisi terakhir dari komponen sistem yang diperlukan melalui situs ftp

lalu di-*compile*. Pada awal keberadaan Linux, operasi seperti diatas persis seperti yang dilaksanakan oleh pengguna Linux. Namun, dengan semakin dewasanya Linux, berbagai individu dan kelompok berusaha membuat pekerjaan tersebut lebih mudah dengan cara menyediakan sebuah set bingkisan yang standar dan sudah di-*compile* terlebih dahulu supaya dapat diinstall secara mudah.

Koleksi atau distribusi ini, mengandung lebih dari sistem Linux dasar. Mereka mengandung instalasi sistem ekstra dan utilitas manajemen, juga paket yang sudah di *compile* dan siap diinstall dari banyak alat UNIX yang biasa, seperti *news servers*, *web browsers*, *text-processing* dan alat mengedit, termasuk juga *games*.

Distribusi pertama mengatur paket-paket ini secara sederhana menyediakan sebuah sarana untuk memindahkan seluruh file ke tempat yang sesuai. Salah satu kontribusi yang penting dari distribusi modern adalah manajemen atau pengaturan paket-paket yang lebih baik. Distribusi Linux pada saat sekarang ini melibatkan *database packet tracking* yang memperbolehkan suatu paket agar dapat diinstal, di *upgrade*, atau dihilangkan tanpa susah payah.

Distribusi SLS (*Soft Landing System*) adalah koleksi pertama dari bingkisan Linux yang dikenal sebagai distribusi komplit. Walau pun SLS dapat diinstall sebagai entitas tersendiri, dia tidak memiliki alat-alat manajemen bingkisan yang sekarang diharapkan dari distribusi Linux. Distribusi Slackware adalah peningkatan yang besar dalam kualitas keseluruhan (walau pun masih memiliki manajemen bingkisan yang buruk); Slackware masih merupakan salah satu distribusi yang paling sering diinstall dalam komunitas Linux.

Sejak dirilisnya Slackware, sejumlah besar distribusi komersil dan non-komersil Linux telah tersedia. Red Hat dan Debian adalah distribusi yang terkenal dari perusahaan pendukung Linux komersil dan perangkat lunak bebas komunitas Linux. Pendukung Linux komersil lainnya termasuk distribusi dari Caldera, Craftworks, dan Work- Group Solutions. Contoh distribusi lain adalah SuSE dan Unifix yang berasal dari German.

3.3.3 LINUX Saat Ini

Menurut Gabungan Kelompok Kerja 21–28 IKI-20230 Semester Genap 2002/2003 (2003, p206), Linux merupakan salah satu sistem operasi yang perkembangannya paling cepat. Kehadiran sejumlah kelompok pengembang, tersebar di seluruh dunia, yang selalu memperbaiki segala fiturnya, ikut membantu kemajuan sistem operasi Linux. Bersamaan dengan itu, banyak pengembang yang sedang bekerja untuk memindahkan berbagai aplikasi ke Linux (dapat berjalan di Linux).

Masalah utama yang dihadapi Linux dahulu adalah *interface* yang berupa teks (*text based interface*). Ini membuat orang awam tidak tertarik menggunakan Linux karena harus dipelajari terlebih dahulu untuk dapat dimengerti cara penggunaannya (tidak *user-friendly*). Tetapi keadaan ini sudah mulai berubah dengan kehadiran KDE dan GNOME. Keduanya memiliki tampilan desktop yang menarik sehingga mengubah persepsi dunia tentang Linux.

Linux di negara-negara berkembang mengalami kemajuan yang sangat pesat. Harga perangkat lunak (misalkan sebuah sistem operasi) bisa mencapai US

\$100 atau lebih. Di negara yang rata-rata penghasilan per tahun adalah US \$200-300, US \$100 sangatlah besar. Dengan adanya Linux, semua berubah. Karena Linux dapat digunakan pada komputer yang kuno, dia menjadi alternatif cocok bagi komputer beranggaran kecil. Di negara-negara Asia, Afrika, dan Amerika Latin, Linux adalah jalan keluar bagi penggemar komputer.

Pemanfaatan Linux juga sudah diterapkan pada super komputer.

Diberikan beberapa contoh:

- The Tetragrid, sebuah mega computer dari Amerika yang dapat menghitung lebih dari 13 trilyun kalkulasi per detik (13.6 TeraFLOPS). Tetragrid dapat dimanfaatkan untuk mencari solusi dari masalah matematika kompleks dan simulasi, dari astronomi dan riset kanker hingga ramalan cuaca.
- Evolocity, juga dari Amerika, dapat berjalan dengan kecepatan maksimum 9.2 TeraFLOPS (Floating Operations Per Second), menjadikannya sebagai salah satu dari lima supercomputer tercepat di dunia.

Jika melihat ke depan, kemungkinan Linux akan menjadi sistem operasi yang paling dominan bukanlah suatu hal yang mustahil. Karena semua kelebihan yang dimilikinya, setiap hari semakin banyak orang di dunia yang mulai berpaling ke Linux.

3.3.4 *Server* LINUX

Menurut Cartealy (2013, pXV), salah satu OS (Operating System) yang banyak digunakan oleh *server* adalah Linux. Linux merupakan OS yang bersifat bebas (*free*) dan kode sumber terbuka (*open source*). Salah satu sifat Linux yang lain adalah pengguna tidak perlu membayar lisensi untuk memakai Linux, dan tentu saja ini tidak kalah pentingnya terutama sebagai pertimbangan biaya.

Menurut laporan Netcraft (2011), Linux digunakan oleh 13 dari 20 daftar teratas perusahaan hosting terhandal di dunia.

3.4 **Basis Data** (*Database*)

Menurut Ramez (2000), Data merupakan fakta mengenai suatu objek seperti manusia, benda, peristiwa, konsep, keadaan dan sebagainya yang dapat dicatat dan mempunyai arti secara implisit. Data dapat dinyatakan dalam bentuk angka, karakter atau simbol, sehingga bila data dikumpulkan dan saling berhubungan maka dikenal dengan istilah basis data (*database*). Sedangkan menurut Abdul (1999) basis data merupakan kumpulan informasi bermanfaat yang diorganisasikan ke dalam aturan yang khusus. Informasi ini adalah data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang. Menurut Encyclopedia of Computer Science and Engineer, para ilmuwan di bidang informasi menerima definisi standar informasi yaitu data yang digunakan dalam pengambilan keputusan.

Definisi lain dari basis data menurut Fabbri dan Schwab adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan duplikasi data.

Menurut Ramez Elmasri mendefinisikan basis data lebih dibatasi pada arti implisit yang khusus, yaitu:

- a. Basis data merupakan penyajian suatu aspek dari dunia nyata (*real world*).
- b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit. Sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
- c. Basis data perlu dirancang, dibangun dan data dikumpulkan untuk suatu tujuan. Basis data dapat digunakan oleh beberapa user dan beberapa aplikasi yang sesuai dengan kepentingan user.

Menurut Waliyanto (2000) dapat dikatakan bahwa basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa user untuk berbagai kepentingan

3.5 *Captive Portal*

Menurut Cartaly (2013, p131), *captive portal* merupakan *server* yang seringkali digunakan untuk memaksa pemakai jaringan untuk melakukan otentikasi. Teknik *captive portal* akan mencegat semua koneksi dari *client* dengan tujuan port apapun dan membloknnya sehingga *client* tersebut tidak dapat menggunakan sumber daya jaringan.

Hal ini dilakukan sampai *client* tersebut membuka browser dan mencoba mengakses *internet*. Akses tersebut kemudian diblokkan ke halaman web yang biasanya merupakan laman login untuk melakukan otentikasi. Ketika *client* dapat

memberikan otentikasinya, maka *client* tersebut akan diijinkan untuk menggunakan sumber daya jaringan.

Salah satu aplikasi *captive portal* di Linux adalah *CoovaChilli*. *CoovaChilli* merupakan pengembangan dari *ChilliSpot* yang kini terhenti. Untuk membangun *captive portal*, *CoovaChilli* akan berguna untuk menghubungi RADIUS dengan menggunakan NAS dan sandi yang telah ditentukan di konfigurasi *FreeRadius*.

Yang perlu diperhatikan ketika menggunakan *CoovaChilli* adalah *server* akan membutuhkan minimal dua buah kartu jaringan. Satu akan berhadapan dengan jaringan *client local*, yang lainnya akan menghadap ke RADIUS. *CoovaChilli* akan mencegat semua trafik yang berasal dari dalam jaringan lokal sehingga penempatan kartu jaringan yang menghadap jaringan lokal diperlukan. *CoovaChilli* juga perlu memiliki kartu jaringan yang menghadap RADIUS untuk meneruskan permintaan ke RADIUS.

3.6 Area Hotspot

3.6.1 Area

Menurut Poerwodarminto (2003, p58), area adalah bagian permukaan bumi atau daerah, sedangkan menurut Kamus Besar Bahasa Indonesia cet.7 (1996 p55), area adalah bagian dari permukaan bumi atau daerah atau wilayah geografis yang digunakan untuk keperluan khusus.

Sehingga dari definisi di atas dapat disimpulkan bahwa pengertian area adalah bagian atau daerah atau wilayah permukaan bumi yang digunakan untuk keperluan khusus, atau dapat pula diartikan wilayah untuk keperluan *hotspot*.

3.6.2 Hotspot (Wi-Fi)

Menurut Priyambodo (2005, p1), *hotspot* (Wi-Fi) adalah satu standar *Wireless Networking* tanpa kabel, hanya dengan komponen yang sesuai dapat terkoneksi ke jaringan.

Komponen Utama jaringan Wi-Fi :

- *Acces Point*
- *Wireless LAN Device*
- *Mobile/ Desktop PC*
- *Ethernet LAN*

Selanjutnya menurut Mulyanta (2008: 52) Wi-Fi merupakan merek dagang *wireless LAN* yang diperkenalkan dan distandarisasi oleh *Wi-Fi Alliance*.

Sedangkan *hotspot* (Wi-Fi) yang akan dibahas dalam penelitian ini adalah sarana terkoneksi jaringan *internet* tanpa kabel, dengan menggunakan standar *wireless LAN*, namun demikian dalam menjalankan *hotspot* diperlukan sarana lain, seperti *Notebook/laptop/PDA* yang memiliki fasilitas *wireless LAN*.

3.6.3 Pengertian Area Hotspot (Wi-Fi)

Area hotspot (Wi-fi) adalah bagian atau daerah atau wilayah yang terkoneksi jaringan *internet* tanpa kabel. Wi-Fi (*Wireless Fidelity*) adalah istilah populer untuk jaringan *wireless* (tanpa kabel) dengan frekuensi tinggi.

Keunggulan dan Kelemahan Jaringan *Hotspot* (Wi-Fi) menurut Priyambodo (2005, p5) adalah sebagai berikut :

Keunggulan :

- Biaya Pemeliharaan murah
- Infrastruktur berdemensi kecil
- Pembangunannya cepat
- Mudah dan murah untuk direlokasi
- Mendukung Portabilitas

Kelemahan :

- Biaya Peralatan mahal
- Delay yang sangat besar
- Mudah untuk terinterferensi
- Kapasitas jaringan kecil
- Keamanan/kerahasiaan data kurang terjamin