

# Otentifikasi Menggunakan Directory Service OpenLDAP

Saat ini, penggunaan *directory service* sebagai *back end* untuk otentikasi sudah kian populer di berbagai *platform*. Kita akan membahas contoh penggunaan *directory service* OpenLDAP sebagai back end untuk otentifikasi sistem.

Kalau dipikir-pikir, saat ini pengguna komputer sebenarnya cukup kerepotan juga. Terutama dalam soal mengingat *password*. Mulai dari masuk komputer, seseorang harus mengingat *password account* sistemnya. Setelah itu, pengguna mungkin akan mengawali rutinitas satu hari ini dengan memeriksa daftar tugas dan pesan yang tersimpan dalam aplikasi ERP yang digunakan oleh perusahaan. Untuk masuk ke sistem ERP tersebut, pengguna tersebut juga harus mengisikan *password*.

Belum lagi ketika ada waktu senggang dan pengguna tersebut ingin berkomunikasi dengan Yahoo! Messenger, *password* kembali harus dimasukkan. Atau ketika ingin memeriksa rekening memanfaatkan Internet banking. Atau menggunakan webmail. Atau mem-*posting* pesan di forum. Atau mencari teman di Friendster, dan lain-lain. Pada akhirnya, semakin banyak servis yang digunakan, semakin banyak pula *password* yang harus diingat. Hal ini memiliki efek cukup mengerikan. Sebagian besar user kemudian dapat saja menempelkan kombinasi *username* dan *password*-nya di layar komputer. Bahwa tindakan tersebut pada akhirnya dapat menyebabkan masalah keamanan, kita tidak dapat menyalahkan user tersebut.

Di dalam sistem komputer saja, di luar semua servis yang kita bicarakan tersebut, *password* banyak sekali kita temukan. *Account* sistem, *database* sistem, *account* sistem tambahan (seperti terkoneksi ke jaringan windows atau UNIX lain), *messaging* sistem, *back-up* server, ERP, dan lain sebagainya. Minta ampun.

Yang paling kerepotan, ada kalanya sistem otentikasi pada suatu sistem tidak

dapat bekerja. Dan oleh karena itu, kita pun tidak dapat memasuki sistem tersebut. Belum lagi, terlalu banyak *username* dan *password* yang harus dikelola oleh admin. Masalah-masalah tambahan seperti susahnya melakukan pengubahan *password* atau sinkronisasi *password* akan timbul. Belum lagi ditambah dengan ilustrasi sebelumnya, bahwa user akan sangat kerepotan.

Oleh sebab itu, sudah sejak lama pendekatan *single sign on* (SSO) dilakukan pada berbagai sistem. Untuk melakukan SSO, kita akan membutuhkan sebuah sistem yang khusus menyimpan informasi login dan servis, beserta atribut tambahan seperti *group* dan lain sebagainya. Sistem tersebut biasanya memanfaatkan *directory service*.

*Directory service* adalah hal yang luar biasa. Kita mengenal berbagai protokol dan implementasi akan hal tersebut. Berbagai vendor juga tidak mau kalah dengan *directory service* mereka. Novell sempat sangat terkenal dengan NDSnya, Microsoft berusaha mencaplok pasar Novell dengan Active Directorynya, dan komunitas yang sibuk berteriak dengan OpenLDAP-nya.

Secara sederhana, suatu *directory service* atau *directory server*, umumnya adalah suatu *database system* sederhana yang menekankan pada layanan direktori. Menyimpan berbagai informasi sesuai dengan servis yang didukung. Sangat menekankan pada performa *read* daripada *write*.

Sistem ini dikenali terpisah dari konsep *database relational* atau *Object Oriented* pada umumnya, dan kita seharusnya juga tidak beranggapan bahwa *directory service* dapat digunakan sekalian untuk menyimpan data dan berfungsi sebagai *database*

*relational*.

Di Linux, *directory service* yang populer adalah OpenLDAP. LDAP adalah singkatan dari *Lightweigh Directory Access Protocol*. Artinya, protokol kelas ringan untuk akses pada *directory service*. Dengan memanfaatkan OpenLDAP, kita dapat menyimpan informasi login sistem user ke LDAP server. Kita dapat juga menyimpan informasi login email user, samba, jabber, squid, *database* seperti PostgreSQL, dan lain sebagainya. Semua servis tersebut umumnya mendefinisikan apa yang diperlukan ke dalam sebuah atau lebih skema. Sebagai contoh, skema untuk login ke sistem akan membutuhkan *username*, *password*, *uid*, *gid*, *shell*, dan lain sebagainya. Setiap service umumnya berbeda. Namun, untuk hampir sebagian besar service, kita dapat mempergunakan skema LDAP yang telah tersedia.

Dengan pendekatan seperti ini, user hanya perlu mengingat satu user dan satu *password* untuk masuk ke semua service. *Single sign on*.

Beberapa dari kita mungkin sempat berpikir bahwa cara ini memiliki *single point of failure*. Bagaimana kalau LDAP server mogok? Berarti, semua service bahkan tidak dapat digunakan oleh user. Ini lebih berbahaya daripada cara konvensional yang walaupun saah satu atau beberapa mogok, user masih bisa memanfaatkan servis lainnya.

Untunglah. Kita mengenal adanya mekanisme replikasi untuk LDAP server. Urusan replikasi ini menjadi topik tersendiri yang menarik dan berada di luar cakupan artikel ini.

Dalam tulisan kita kali ini, kita akan memfokuskan terlebih dahulu otentikasi

untuk sistem. Harap diperhatikan, walaupun otentikasi menggunakan LDAP terdengar ideal pada akhirnya, banyak kendala yang mungkin harus kita jumpai. Tidak semua servis peduli terhadap otentikasi dengan LDAP sebagai backend-nya. Banyak sekali yang hanya memanfaatkan sebuah file password. Beberapa waktu yang lalu, SAM-BA bahkan tidak *official* mendukung LDAP. Saat ini, tentu saja urusan ini sudah selesai.

Tutorial ini mengasumsikan penggunaan distro SUSE 9.1 Namun, diusahakan pembahasan juga mencakup hal-hal yang tidak distro-dependent sehingga tetap bisa diikuti oleh pengguna distro lain. Tutorial ini tidak membahas OpenLDAP secara kompleks dan komplis, namun lebih bertujuan untuk memungkinkan otentikasi sistem.

## Instalasi paket yang diperlukan

Pertama-tama, kita perlu menginstal beberapa paket. Bukalah modul instalasi program YaST dan pilihlah paket `openldap2`. Anda juga mungkin ingin menginstal paket `yast2-ldap` dan `yast2-ldap-client`. Paket `openldap2-client` umumnya telah terinstal untuk digunakan oleh berbagai paket lain. Apabila karena sesuatu hal paket ini belum terinstall, maka Anda mungkin ingin menginstalnya juga.

SUSE melakukan sedikit penyesuaian terhadap paket OpenLDAP. Oleh karena itu, bagi Anda yang pernah bekerja dengan OpenLDAP di distro lain, Anda mungkin akan merasakan beberapa perbedaan.

Paket `openldap2` ini telah berikan server `ldap`, `replication` daemon, skema-skema,

tool-tool administrasi, dan lain sebagainya.

Setelah semua paket ini diinstal, maka kita telah siap melanjutkan ke langkah berikutnya.

## Konfigurasi OpenLDAP

OpenLDAP menyimpan semua konfigurasinya ke dalam `/etc/openldap`. Di dalam direktori ini, paling tidak Anda akan menemukan dua file konfigurasi, yaitu `ldap.conf` dan `slapd.conf`.

Harap diperhatikan, server untuk OpenLDAP dinamakan `slapd`, dan oleh karena itu, konfigurasi untuk server OpenLDAP adalah `slapd.conf`. Sementara, `ldap.conf` lebih merupakan konfigurasi global dan umum untuk `ldap` client. Anda bisa mengedit file ini untuk menyesuaikan pengaturan LDAP client. SUSE telah menyiapkan pengaturan client LDAP untuk penggunaan YaST. Anda bisa mengatur `ldap.conf` sesuai apa yang Anda isikan pada YaST. Pengaturan `ldap.conf` ini akan berguna untuk aplikasi yang tidak diatur oleh YaST.

Pertama-tama, kita akan mempergunakan YaST untuk konfigurasi LDAP Client. Kita memang belum mengatur LDAP server, namun, pemahaman LDAP dapat pula dimulai dengan memahami bagaimana cara terhubung ke server LDAP dan beberapa istilah yang umum ditemukan pada penggunaan LDAP. Oleh karena itu, bukalah YaST dan akseslah modul LDAP Client di Network Services|LDAP Client. Sebuah dialog akan ditampilkan.

Pertama-tama, Anda harus memilih Use LDAP. Dan, setelah itu, isikanlah Base

DN pencarian server LDAP Anda. DN adalah singkatan dari *Distinguished Name*. Dalam tulisan ini, kita nantinya akan mengatur LDAP client dengan base DN adalah `dc=company, dc=com`. Kurang lebih, ini mewakili penulisan domain `company.com`.

Setelah itu, isikanlah nama server LDAP Anda. Dalam tulisan ini, tentu saja namanya adalah `127.0.0.1`.

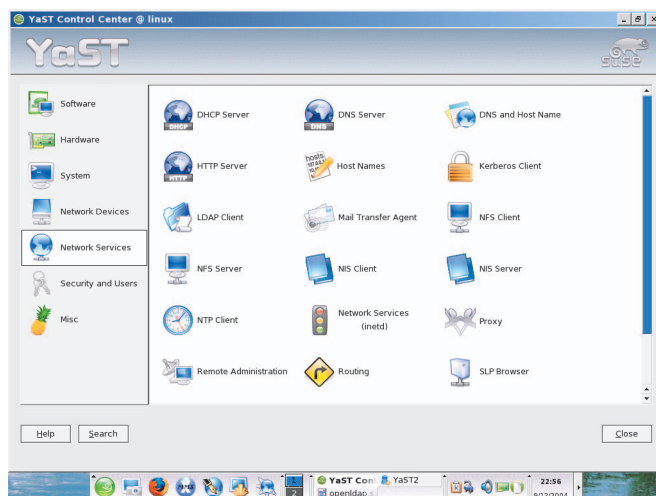
Hanya kedua hal tersebutlah yang perlu diatur untuk saat ini. Pembahasan mengenai automounter akan dibahas pada akhir artikel. Sementara, opsi LDAP TLS/SSL adalah pilihan untuk berkomunikasi dengan server LDAP yang mendukung keamanan SSL. Dan, pilihan LDAP Version 2 adalah pilihan untuk berkomunikasi dengan LDAP versi 2. Saat ini, kita menggunakan versi 3 walaupun nama paketnya adalah `openldap2`. Memang cukup membingungkan.

Dengan mengisi kedua hal ini, kita telah menentukan LDAP server yang digunakan adalah `127.0.0.1` dan pencarian akan dilakukan pada Base DN `dc=company, dc=com`.

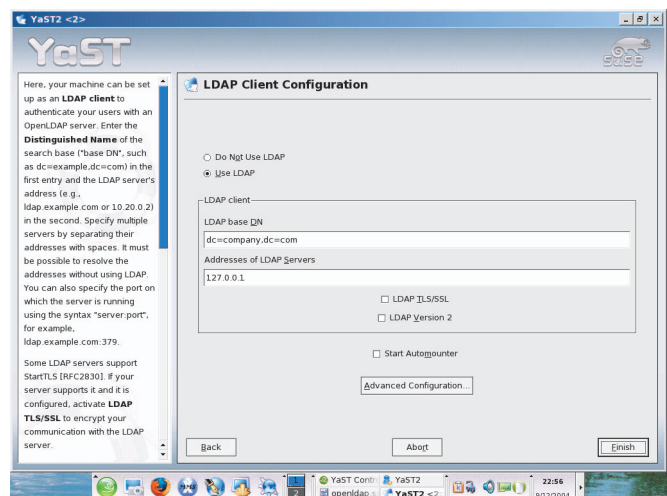
Selanjutnya, kliklah tombol Finish. Kita telah selesai dengan pengaturan client untuk saat ini. Kita akan melanjutkan pada konfigurasi server. Kembalilah ke `/etc/openldap`. Kita akan bersiap-siap untuk mengedit `slapd.conf`.

Umumnya, Anda tidak perlu banyak mengutak-atik pengaturan bagian atas. Apa yang perlu Anda perhatikan adalah bagian bawah konfigurasi ini. Berikut ini adalah contoh milik penulis:

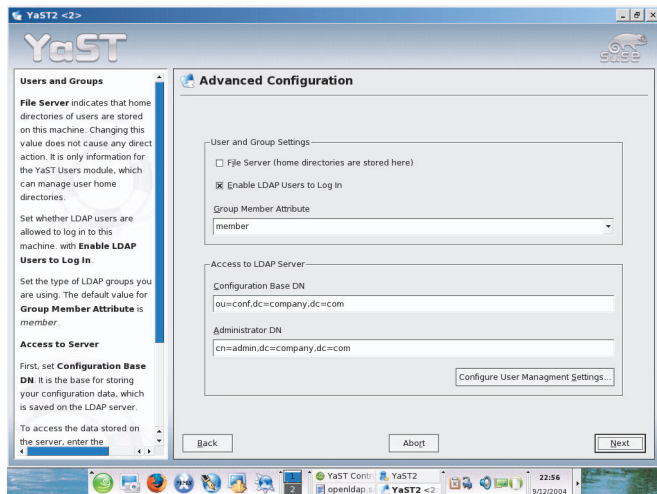
|            |        |
|------------|--------|
| database   | bdb    |
| checkpoint | 1024 5 |



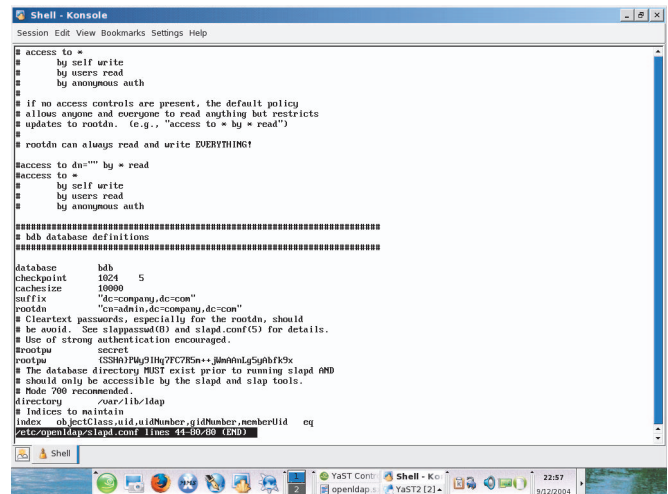
YaST.



Modul LDAP Client YaST.



Advanced LDAP Client configuration.



Konfigurasi file slapd.conf.

```

cachesize 10000
suffix "dc=company,dc=com"
rootdn "cn=admin,dc=company,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoid. See slapasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
#rootpw secret
rootpw {SSHA}PWY9IHq7FC7R5m++jWmAAnLg5yAbfk9x
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index objectClass,uid,uidNumber,gidNumber,memberUid eq

```

Perhatikanlah bagian suffix, rootdn, rootpw, dan index. Untuk saat ini, rootdn dapat dianggap sebagai user admin untuk LDAP server kita. Sementara, rootpw adalah password untuk admin. Anda mungkin akan mempergunakan program slapasswd untuk menghasilkan password dalam keadaan terenkripsi. Password cleartext sangat tidak disarankan.

## Menjalankan service OpenLDAP

Anda mungkin ingin menjalankan service ldap setiap kali booting. Oleh karena itu,

daftarkanlah service ldap ini dalam proses booting Anda. Bukalah YaST dan akseslah *System|Runlevel Editor*. Enable-kanlah servis ldap. Anda akan melihat pesan sukses apabila semuanya berjalan lancar.

Kini, LDAP server Anda telah berjalan. Sekarang, kembalilah kepada modul LDAP Client YaST. Kliklah tombol Advanced Configuration. Sebuah dialog baru akan ditampilkan. Aktifkanlah pilihan Enable LDAP users to Log in. Dengan mengaktifkan pilihan ini, Anda telah menyelesaikan banyak hal. Apabila Anda harus melakukannya secara manual, banyak hal yang perlu diatur. Banyak file konfigurasi yang perlu diatur. Terimakasih kepada YaST.

Setelah itu, pilihlah juga mmberr sebagai *Group Member Attribute*. Isikan juga Configuration Base DN, yang merupakan basis untuk menyimpan semua file konfigurasi di LDAP server Anda. Masukkanlah nilai yang masuk akal seperti:

```
ou=conf, dc= company, dc=com.
```

Pada bagian ini, ou merupakan singkatan dari *organizational unit*. Karena ini adalah kali pertama kita mengatur LDAP server, maka Configuration Base DN tersebut pasti tidak ditemukan di LDAP server. Kita perlu membuatnya terlebih dahulu. Buatlah sebuah file dengan isi sebagai berikut ini, dan simpan sebagai /etc/openldap/base.ldif:

```

dn: dc=company,dc=com
dc: company
objectClass: top
objectClass: domain

```

```

dn: ou=conf,dc=company,dc=com
ou: conf
objectClass: top
objectClass: OrganizationalUnit

```

Kemudian, aktiflah di console di direktori /etc/openldap. Berikanlah perintah berikut ini:

```
$ ldapadd -f base.ldif -x -
Dcn=admin,dc=company,dc=com -W
```

Masukkanlah password dan Anda akan melihat pembuatan DN ou=conf,dc=company,dc=com. Tutuplah console Anda, dan kembalilah ke modul YaST LDAP Client.

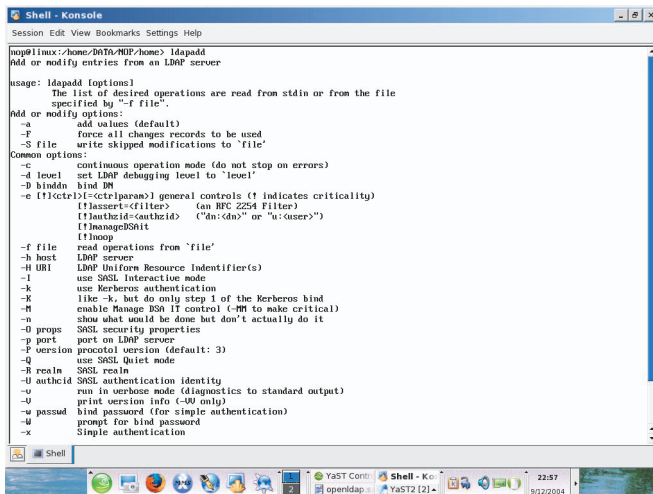
Pada bagian Administrator DN, masukkanlah entri rootdn yang Anda spesifikasi-kan di slapd.conf. Setelah itu, kliklah tombol Next dan Finish.

## Pembuatan user

Selanjutnya, urusan LDAP server untuk saat ini boleh diasumsikan beres. Kita akan membuat user yang akan disimpan dalam LDAP server. Jalankanlah modul Security and Users|Edit and Create Users. Sebuah dialog akan ditampilkan.

Pada bagian kanan bawah, klik dan tahanlah tombol Set Filter. Kemudian, pilihlah LDAP Users. Sebuah inputbox password akan ditampilkan. Masukkanlah password untuk admin LDAP Anda.

Setelah itu, Anda bisa menambahkan user seperti biasa. Dalam contoh kali ini, penulis akan membuat user nopldap. Setelah itu, Anda sudah bisa login menggunakan user tersebut seperti biasa.



Penggunaan ldapadd.

Walaupun password disimpan di LDAP server, user tetap bisa mempergunakan program passwd untuk mengganti password-nya. Program passwd siap membantu Anda seperti contoh keluaran berikut:

```
$ passwd
Changing password for noplldap.
Enter login(LDAP) password:
```

Sampai di sini, authentication server menggunakan LDAP sebagai backend telah selesai kita set.

Bagi Anda yang tidak menggunakan distro SUSE, beberapa hal seperti pembuatan user akan menjadi hal yang tidak trivial. Oleh karena itu, Anda bisa memilih beberapa cara. Cara pertama adalah dengan cara manual menggunakan ldapadd dan file .ldif. Cara ini sangat tidak direkomendasikan. Cara kedua adalah mempergunakan program directory\_ administrator. Program ini cukup populer dan berguna. Cara ketiga adalah dengan memanfaatkan berbagai tool pihak ketiga. Harap diperhatikan, beberapa tool seperti tool yang berjalan di web sepertinya begitu menjanjikan akan mempermudah. Sayangnya, terkadang, instalasinya saja sudah merepotkan dan membutuhkan banyak paket.

## LDAP + NFS

Dengan LDAP, sebenarnya Anda juga bisa membuat sistem komputasi yang terintegrasi. Semua home user akan disimpan dalam sebuah server file tersendiri. File server ini juga nantinya akan terintegrasi untuk user yang juga menggunakan Windows. Jadi, semacam gudang file. Semua diletakkan di

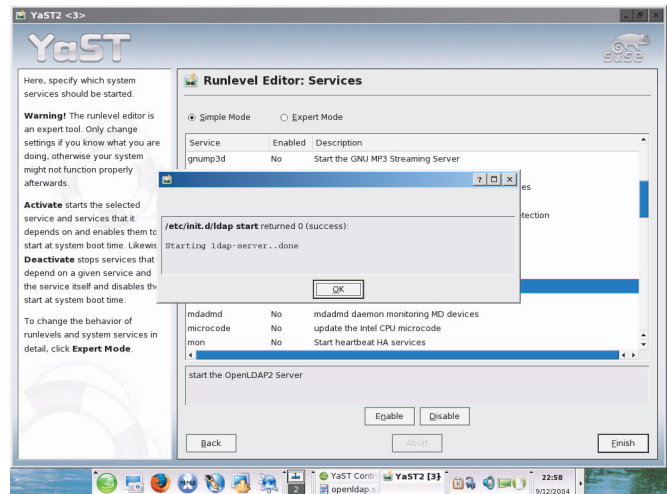
sini, dan, semuanya bahagia.

Untuk sharing file sesuai budaya UNIX, kita akan mempergunakan NFS. User root akan kita squash sehingga sharing file menjadi lebih aman. Kita kemudian mengeksport home direktori di server dan client akan melakukan mounting di direktori yang dibedakan dari home user lokal. Sebagai contoh, apabila home user lokal di simpan pada /home, maka katakanlah apa yang kita mount dari NFS akan diletakkan pada /nethome. Setiap kali booting, maka NFS akan dimount secara read write ke /nethome.

Authentication server LDAP disimpan pada server terpisah. Setelah user selesai login, maka dengan ID yang dimiliki, akan memiliki hak yang sama dan mampu mengakses /nethome/<username> sesuai hak yang mereka miliki. Tentu saja, pada saat pembuatan user, home directory user harus diset pada /nethome/<username>. Sebaiknya, di server, kita juga memiliki /nethome yang sebenarnya.

Bisa Anda bayangkan betapa serunya kondisi seperti ini apabila tercapai. Hal ini memang sekilas merupakan *single point of failure*. Namun, apabila kondisi penting ini bisa ditangani, maka selanjutnya, ke depan, kita akan lebih mudah mengatur file-file user. Penanganan agar risiko failure tidak fatal dapat melibatkan penggunaan backup system, replikasi home user, sinkronisasi oleh user, dan lain sebagainya.

Bicara soal sinkronisasi, user mungkin ingin melakukan sinkronisasi semua file yang dimiliki. Atau, paling tidak, user bisa men-download semua file yang ada dalam



Pengaturan runlevel.

home directory mereka di server ke dalam satu direktori lokal. Hal ini tentunya akan sangat berguna apabila terjadi kesalahan fatal yang menyebabkan downtime cukup lama.

## Service lain?

Setelah sistem, barangkali Anda ingin perlahan-lahan menggunakan LDAP sebagai *authentication backend* untuk service lain. Samba adalah yang paling menggiurkan. Dalam kasus SAMBA, SUSE 9.1 Pro agak sedikit kerepotan. Anda mungkin membutuhkan smbldap yang dibuat oleh idealx.org.

Apache juga siap mempergunakan LDAP sebagai backend authentication. Dengan demikian, diharapkan kita bisa menyediakan service WebDAV yang lebih baik lagi. Squid juga sangat mendukung penggunaan LDAP. Dan, apabila Anda mempergunakan Jabber, maka Anda akan dengan mudah dapat mempergunakan LDAP sebagai backend otentikasinya.

Kali pertama penulis mencoba LDAP dalam asuhan seorang senior, teman dan partner bisnis penulis saat ini, rasa-rasanya kepala pusing sekali dan tidak mengerti selama sehari-hari. LDAP adalah sistem yang kompleks. Dan, memang wajar apabila pemahamannya lebih memerlukan waktu. Dan, huruf L sendiri sebenarnya sudah menandakan bahwa LDAP sudah lebih sederhana dari directory service lain.

Setelah mengetahui LDAP, percayalah, rasa pusing Anda akan hilang. Ini penulis dapatkan sendiri. ☺

**Noprianto** (noprianto@infolinux.co.id)