

KOMUNIKASI MODEM SERIAL DENGAN MENGGUNAKAN POINT TO POINT PROTOCOL PADA SISTEM OPERASI LINUX

Silvia Rostianingsih

Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra
e-mail : silvia@peter.petra.ac.id

Rudy Adipranata

Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra
e-mail : rudy@petra.ac.id

ABSTRAK: Sistem operasi Linux yang berkembang dengan pesat akhir-akhir ini mulai menarik minat sebagian orang, terutama kalangan akademisi dan profesional yang berkecimpung di dunia komputer. Beberapa hal yang membuat Linux semakin diminati adalah sifat Linux yang open source, artinya banyak aplikasi bersifat gratis, tidak diharuskan untuk membayar. Bahkan source codenya bisa didapatkan dengan mudah, serta kestabilan Linux sebagai sistem operasi mulai diakui oleh banyak kalangan.

Sistem operasi Linux juga dapat melakukan hubungan antar modem, bahkan dapat melakukan koneksi ke Internet. Sistem operasi Linux juga mendukung sistem *client-server*. Jadi di dalam suatu sistem Local Area Network dapat menggunakan hanya dengan satu modem saja untuk dapat koneksi ke Internet dengan menggunakan banyak komputer.

Point to Point Protocol merupakan salah satu protocol yang banyak sekali dipakai untuk melakukan koneksi ke Internet maupun ke LAN, karena beberapa kemudahan yang dimiliki, terutama proses instalasi dan penggunaannya.

Kata kunci: *point to point protocol, linuc, client-server.*

ABSTRACT: *Linux Operating System which growing rapidly this day start to attract many people, especially academic and proffessionals in computer's world. Many things which make Linux more attractive is the open source's characteristic, which mean its application is freely available to everyone. In fact, we can get the source code easily, and the stabilize as an operating system getting admission by many society.*

Linux also can make a connection between modem, even internet. Linux also support the client-server system. Therefore, Local Area Network system can use only one modem as the connection for many computers to go through the internet.

Point to point protocol is one of the protocols which used to do the connection to the internet or LAN, because of a lot of things its provided easily, especially the installation and the used of it.

Keywords: *point to point protocol, linuc, client-server.*

1. PENDAHULUAN

PPP (*the Point to Point Protocol*) adalah suatu mekanisme (cara kerja) untuk membuat dan menjalankan IP (*the Internet Protocol*) serta jaringan protokol yang lain melalui hubungan serial, yang bisa juga diartikan sebagai hubungan serial secara langsung (tanpa menggunakan modem) dengan melalui telnet ataupun hubungan serial dengan menggunakan modem dan saluran telepon, bahkan dengan menggunakan saluran digital seperti ISDN (*Integrated System Digital Network*).

Point to Point Protocol memberikan suatu metode untuk komunikasi melalui serial. PPP terdiri dari tiga bagian yaitu : metode untuk pengkapsulan datagram, perluasan dari *Link Control Protocol* (LCP), dan anggota dari *Network Control Protocol* (NCP) untuk membentuk dan mengkonfigurasi protokol pada *network layer* yang berbeda.

Proses pengkapsulan tersebut diberikan oleh *driver* pada kernel. Di samping itu terdapat istilah yang dinamakan *pppd*, yaitu *Point to Point Protocol daemon*. *pppd* adalah dasar dari LCP, yang dapat mem-

buktikan keaslian data, dan juga NCP untuk membentuk dan mengkonfigurasi *Internet Protocol* (IP), yang sering juga disebut *IP Control Protocol*.

Dengan menggunakan PPP, maka komputer Linux dapat dihubungkan dengan PPP *server* (komputer yang dijadikan sebagai *server*) dan dapat melakukan akses ke jaringan dimana *server* tersebut terhubung seperti halnya jika melakukan hubungan secara langsung ke jaringan. Komputer Linux ini sendiri juga dapat dijadikan sebagai PPP *server*, sehingga komputer lain dapat melakukan hubungan (*dial*) ke komputer Linux tersebut dan dapat melakukan akses pada semua fasilitas yang tersedia pada komputer Linux maupun pada jaringan.

Perbedaan mendasar di dalam melakukan hubungan dengan PPP dan Ethernet card adalah dari segi kecepatannya. Secara teori Ethernet card standard dapat terhubung pada kecepatan 10 Mbps (Mega bits per second). Sedangkan pada modem analog hanya dapat beroperasi pada kecepatan maksimal 56 Kbps (kilo bits per second). Selain itu, jenis hubungan dengan menggunakan PPP terdapat batasan-batasan di dalam penggunaan aplikasi dan pelayanan.

1.1 Distribusi Linux

Sesuai dengan sifat sistem operasi Linux yang *open source*, maka terdapat bermacam-macam perusahaan yang mendistribusikan sistem operasi Linux dengan masing-masing mempunyai cara yang berbeda di dalam melakukan segala sesuatunya. Beberapa contoh perusahaan distribusi Linux yang terkenal adalah: Slackware, Red Hat, Debian.

Secara umum, terdapat dua macam sistem inialisasi yaitu BSD system initialisation dan System V system initialitation.

1.2 Nomor IP

Setiap komputer yang terhubung ke Internet pasti mempunyai nomor IP (*the Internet Protocol*) yang unik (tidak ada yang sama) dan telah ditetapkan pada masing-masing negara. Jika suatu komputer akan melakukan hubungan ke *Local Area Network* (LAN), maka perlu menggunakan nomor IP yang diberikan pada masing-

masing komputer pada LAN tersebut. Untuk melakukan hubungan ke LAN yang lain, tidak boleh sembarangan menggunakan nomor IP milik komputer lain, karena hal ini bisa menyebabkan terjadinya gangguan pada komunikasi data dari komputer lain yang nomor IP nya telah dipakai tersebut.

Nomor IP ada yang dikenal dengan sebutan ‘unconnected network’, artinya jaringan yang tidak terhubung, yaitu nomor IP yang digunakan untuk jaringan yang tidak terhubung ke Internet, biasanya nomor IP tersebut dipakai hanya untuk melakukan hubungan antar komputer pada suatu LAN.

Berikut ini adalah contoh nomor IP yang secara spesifik hanya dipakai untuk LAN yang tidak terhubung ke Internet. Urutannya adalah sebagai berikut:

- 1 alamat jaringan kelas A
10.0.0.0 (netmask 255.0.0.0)
- 16 alamat jaringan kelas B
172.16.0.0 - 172.31.0.0 (netmask 255.255.0.0)
- 256 alamat jaringan kelas C
192.168.0.0 - 192.168.255.0
(netmask 255.255.255.0)

Apabila mempunyai suatu LAN yang tidak mempunyai nomor IP yang unik, maka dapat menggunakan salah satu dari alamat jaringan tersebut di atas untuk masing-masing komputer. Hal yang perlu diperhatikan adalah bahwa nomor IP tersebut di atas tidak akan pernah dipergunakan di Internet, sehingga tidak akan dapat digunakan jika akan terhubung ke Internet.

Akan tetapi, nomor IP tersebut masih dapat dipergunakan pada *Local Ethernet* pada komputer yang telah terhubung ke Internet. Hal ini disebabkan nomor IP yang sesungguhnya itu hanya untuk melakukan hubungan ke jaringan, bukan ke komputer pada Internet. Sebagai contoh, misalnya jaringan menggunakan nomor IP 10.0.0.1, maka pada saat melakukan hubungan ke Internet menggunakan PPP, maka PPP akan memberikan nomor IP yang valid yang diberikan oleh *server*. Dengan demikian maka pada saat yang sama hanya satu komputer saja dalam LAN yang dapat melakukan hubungan ke Internet.

Bagi para pemakai komputer yang akan melakukan hubungan ke *Internet Service Provider* melalui PPP, tidak perlu meng-

gunakan nomor IP. Jika ingin menghubungkan LAN dalam skala yang kecil ke Internet, maka bisa meminta Internet Service Provider untuk memberikan semacam subnet, yaitu sederetan IP address yang tidak terpakai, atau dengan cara lain menggunakan nomor IP samaran.

Sedangkan bagi pemakai yang hanya menggunakan satu komputer untuk melakukan hubungan ke Internet melalui Internet Service Provider, maka Internet Service Provider akan memberikan nomor IP dynamic.

Dengan menggunakan nomor IP yang dynamic, maka setiap melakukan hubungan tidak akan menggunakan nomor IP yang sama. Hal ini mempunyai implikasi di dalam berbagai jenis aplikasi yang dilayani oleh server pada komputer Linux seperti misalnya pengiriman mail, ftp, http dan lain sebagainya. Berbagai jenis pelayanan inilah yang kemudian menjadi bahan pemikiran para pakar komputer di dunia bahwa komputer pada masa yang akan datang haruslah memiliki kemampuan melakukan akses berbagai macam pelayanan tersebut dengan menggunakan nomor IP yang sama. Karena jika menggunakan nomor IP yang berbeda-beda tersebut akan menyebabkan kesulitan di dalam pengaturan nomor IP dan ada kemungkinan nomor IP habis untuk dipakai semuanya.

1.3 Client dan Server

PPP mempunyai sifat *peer to peer protocol*, artinya protokol yang mempunyai persamaan antara satu dengan lainnya, jadi secara teknis tidak ada perbedaan antara komputer yang melakukan hubungan dan komputer yang akan dihubungi.

Untuk lebih jelasnya, bahwa pada saat komputer akan melakukan hubungan (dial) ke komputer lain dengan menggunakan PPP, maka komputer tersebut disebut sebagai *client*. Sedangkan komputer yang akan dihubungi itu disebut sebagai *server*. Jika komputer Linux dibuat supaya dapat digunakan untuk menerima dan menangani *dial in* pada hubungan PPP, maka komputer tersebut dinamakan PPP server.

Setiap komputer Linux dapat berfungsi sebagai PPP *server* maupun *client*, bahkan bisa dilakukan secara bergantian (simultan),

jika mempunyai lebih dari satu *serial port* (atau modem jika dibutuhkan). Jadi pada kenyataannya tidak ada perbedaan yang mendasar antara *client* dan *server* pada saat suatu hubungan PPP terbentuk.

Kegunaan dari PPP sebagai client adalah dapat melakukan hubungan pada satu komputer atau lebih pada suatu lokasi di Internet, dan kebanyakan orang lebih suka menggunakan komputer Linux mereka sebagai *client*, dibanding menggunakan komputer Linux mereka sebagai *server*.

2. POINT TO POINT PROTOCOL PADA CLIENT

Terdapat urutan langkah yang perlu dilakukan untuk membuat supaya komputer Linux berfungsi sebagai *client* sehingga dapat terhubung ke *server*.

2.1 Pengenalan Perangkat Lunak

Pada komputer Linux yang akan dijadikan sebagai client harus sudah ter-*install* perangkat lunak yang dibutuhkan yaitu perangkat lunak untuk PPP. Jika belum ada maka perangkat lunak tersebut bisa didapatkan dari Linux PPP daemon.

2.2 Compile PPP Pada Kernel

Untuk dapat menggunakan PPP, maka kernel pada Linux harus di-*compile* terlebih dahulu. Source code untuk kernel pada Linux bisa didapatkan pada direktori `/usr/src/linux` pada sistem file standar Linux. Kebanyakan distribusi Linux sudah menyediakannya yaitu berupa file dan sub direktorinya sebagai suatu bagian dari proses instalasi.

2.3 Instalasi Linux Kernel

Untuk proses install dan *compile* kernel pada Linux maka hal yang paling penting adalah *log in* sebagai *root*, karena jika *log in* sebagai *user* biasa tidak akan bisa melakukan proses *compile* tersebut. Adapun langkah-langkah setelah itu adalah :

- Ganti direktori menjadi direktori `/usr/src`.
- Melakukan pengecekan pada direktori `/usr/src/linux` untuk memastikan bahwa *source* sudah ada.

- Jika belum ada *source* bisa didapatkan dari direktori Linux kernel *source*.
- Memilih kernel yang tepat, biasanya yang sering muncul pada saat pertama kali. Setelah itu, letakkan *file source* tersebut pada direktori `/usr/src`. Perlu diperhatikan bahwa 'tar' adalah file kompres, jadi bisa saja terdiri dari beberapa file dalam beberapa direktori, sama dengan file zip pada DOS.
- Jika Linux *source* sudah di instal tetapi ingin melakukan *upgrade* ke kernel yang baru, maka kernel yang lama harus di hapus dulu dengan menggunakan perintah : **rm -rf /usr/src/linux**.
- Setelah itu melakukan proses *uncompress* dan *extract* dengan perintah :
tar xzf linux-2.0.XX.tar.gz.

2.4 Pengenalan Perangkat Keras

Untuk melakukan proses *compile* ulang kernel, adalah penting untuk mengetahui peralatan apa yang terdapat pada komputer, dan juga hal lain yang perlu diketahui adalah mengenai *set up* IRQ, I/O address dan sebagainya.

2.5 Konfigurasi Kernel yang Umum Pada PPP

Jika suatu komputer Linux akan dijadikan sebagai PPP *server*, maka harus dilakukan proses *compile* pada IP *forwarding*. Hal ini perlu juga dilakukan jika ingin menggunakan Linux untuk dapat terhubung ke LAN dan secara bersamaan LAN terhubung ke Internet. Jika terhubung ke Internet melalui suatu LAN, hal yang harus diperhatikan adalah sistem keamanannya, misalnya dengan menggunakan IP fire wall pada kernel. Setelah proses instalasi dan *boot up* kernel selesai dilakukan, maka selanjutnya dapat menjalankan dan melakukan test dengan menggunakan hubungan PPP.

2.6 Mendapatkan Informasi yang Dibutuhkan Tentang PPP Server

Sebelum melakukan hubungan PPP ke *server*, perlu untuk mengetahui informasi-informasi dari *sysadmin* pada PPP server, yaitu :

- Nomor telepon untuk dial
Jika menggunakan PABX, maka perlu juga menggunakan kode nomor PABX untuk *dial* keluar, biasanya menggunakan digit 0 atau 9.
- Jika *server* menggunakan nomor IP STATIC, maka perlu diketahui nomor IP untuk hubungan PPP. Sedangkan jika Internet Service Provider memberikan subnet dari nomor IP yang valid, maka perlu diketahui pula nomor IP yang dapat digunakan dan *netmask*-nya. Akan tetapi kebanyakan Internet Service Provider menggunakan nomor IP DYNAMIC. Pada kenyataannya, meskipun menggunakan nomor IP STATIC, kebanyakan PPP *server* tidak akan pernah mengijinkan *client* untuk memilih nomor IP nya sendiri dengan alasan keamanan.
- Biasanya nomor IP dari Internet Service Provider Domain Name Server terdapat dua meskipun satu saja sudah cukup.
- Pada MS Windows 95 PPP setup mengijinkan alamat DNS diabaikan dengan catatan *client* adalah bagian dari proses koneksi, sehingga tidak diperlukan lagi alamat IP dari DNS *server*. Akan tetapi untuk Linux, diperlukan setidaknya satu DNS. Implementasi PPP pada Linux tidak mengijinkan *setting* nomor IP DNS secara *dynamic* pada waktu hubungan berlangsung. Yang perlu diperhatikan adalah ketika komputer Linux sebagai PPP *client* tidak dapat menerima alamat DNS dari *server*,
- Jika *server* menggunakan PAP/CHAP maka perlu untuk mengetahui *user name* dan *password* pada Internet Service Provider.
- *Server* yang digunakan untuk menjalankan PPP apakah secara otomatis menjalankan PPP pada waktu *login* atau perlu memberikan suatu perintah dan perintah apa yang diperlukan tersebut.
- Perlu diketahui juga apakah *server* menggunakan Microsoft Windows NT system dan apakah juga menggunakan MS PAP/CHAP system. Sering orang mengatakan bahwa dengan menggunakan Windows NT adalah suatu suatu cara untuk meningkatkan keamanan.

2.7 Konfigurasi Modem dan Serial Port

Hal yang penting untuk diperhatikan adalah bahwa modem telah di set up dengan benar dan juga letak *serial port* dari modem. Terdapat sedikit perbedaan antara DOS dan Linux di dalam memberikan nama serial, yaitu :

- DOS com1 = Linux /dev/cua0 (and /dev/ttyS0)
- DOS com2 = Linux /dev/cua1 (and /dev/ttyS1)
- dan seterusnya.

Jika komputer mempunyai 4 *serial ports*, maka pada komputer standard com1 dan com3 menggunakan IRQ4 sedangkan com2 dan com4 menggunakan IRQ3. Jika pada komputer terdapat suatu peralatan pada serial port dimana IRQ-nya sama dengan modem, maka kemungkinan besar terdapat masalah. Jadi yang perlu diperhatikan adalah bahwa *serial port* modem IRQ nya harus unik. Kebanyakan *card* serial yang modern yang biasanya terdapat pada Mother board yang baru, mengijinkan untuk memindah IRQ dari *serial port*.

Perlu diperhatikan bahwa jika ingin bermain-main dengan IRQ, bukan hanya membuka komputer kemudian melepaskan card dan mengganti *jumper*, tetapi juga perlu diketahui peralatan apa yang menggunakan IRQ tersebut. Hal lain yang perlu diperhatikan adalah jika komputer melakukan booting dengan sistem operasi yang lain, pemindahan IRQ dapat menyebabkan proses booting tidak dapat berjalan dengan baik. Ataupun jika mengganti *serial port* ke non-standard IRQ, maka Linux perlu diberitahu masing-masing IRQ yang digunakan.

2.7.1 Serial Port dan Kecepatannya

Jika menggunakan modem eksternal dengan kecepatan tinggi (14.400 atau lebih) maka *serial port* harus mampu untuk mengatasi kecepatan modem, seperti misalnya kemampuan modem untuk melakukan kegiatan kompres data. Untuk masalah ini membutuhkan *serial port* yang menggunakan modern UART (Universal Asynchronous Receiver Transmitter) misalnya 16550 (A). Jika menggunakan komputer yang lama (atau *serial card* yang lama), besar kemung-

kinan bahwa *serial port* hanya mempunyai 8250 UART, dimana hal ini akan menimbulkan masalah jika menggunakan modem dengan kecepatan tinggi.

2.7.2 Nama Serial Port

Sebelumnya, Linux menggunakan cuaX untuk melakukan dial out dan ttySx untuk dial in. Kode kernel yang diperlukan telah diperbaharui pada kernel versi 2.0.x dan sekarang dapat menggunakan ttySx untuk dial in dan dial out. Mungkin untuk masa yang akan datang cuaX tidak akan dipakai lagi.

2.7.3 Pengontrolan Aliran Data

Pada saat data dikirimkan melalui jalur komunikasi serial, ada kemungkinan bahwa data datang lebih cepat daripada kecepatan komputer untuk menanganinya (mungkin komputer sibuk untuk melakukan proses yang lain, perlu diingat bahwa Linux adalah sistem operasi *multi user* dan *multi tasking*). Untuk memastikan bahwa data tidak hilang, beberapa metode untuk mengontrol aliran data perlu diketahui.

Ada dua cara untuk melakukan hal tersebut yaitu :

- Menggunakan sinyal Hardware (Clear To Send/Request to Send - CTS/RTS)
- Menggunakan sinyal software (control S dan control Q, juga XON/XOFF).

Walaupun cara yang kedua tersebut mungkin baik untuk hubungan teks, data pada hubungan PPP semuanya menggunakan 8 bit dan ada kemungkinan bahwa pada suatu tempat pada data terdapat sejumlah byte data yang diartikan sebagai control S dan control Q. Jadi, jika suatu modem di set up dengan menggunakan *software flow control*, maka terkadang menjadi cepat kacau.

Untuk hubungan kecepatan tinggi dengan menggunakan PPP (dimana menggunakan 8 bit data) *hardware flow control* sangat penting dan itulah sebabnya mengapa harus menggunakan *hardware flow control*.

2.7.4 Test Modem untuk Dial Out

Setelah melakukan proses pengecekan pada *serial port* dan melakukan set up pada

modem, langkah berikutnya adalah mencoba untuk melakukan koneksi ke Internet Service Provider apakah bisa terhubung atau tidak.

Berikutnya adalah melakukan *set up* yang dibutuhkan untuk inialisasi modem pada PPP dan melakukan *dial* ke PPP *server* yang diinginkan. Dengan catatan pada saat ini tidak sedang melakukan hubungan PPP, tetapi hanya meyakinkan apakah nomor telepon yang dituju sudah benar dan untuk mengecek apakah *server* sudah benar, untuk berikutnya melakukan *login* pada *server*.

Selama proses ini, segala sesuatu yang berhubungan dengan pencatatan semua proses login menunggu *server* memberitahu kapan waktu untuk memasukkan *user name* dan *password*-nya, dan perintah-perintah lainnya yang diperlukan untuk melakukan koneksi dengan PPP.

Jika *server* menggunakan PAP, maka tidak akan terlihat *prompt* untuk *login*, tetapi malahan terlihat semacam text yang kacau yang disebabkan oleh hubungan antara *control protocol* (yang terlihat acak-acakan seperti sampah) memenuhi layar.

Beberapa hal yang perlu diperhatikan adalah sebagai berikut:

- Beberapa server ada yang cerdas, bisa melakukan *login* dengan menggunakan teks yang berdasarkan pada *user name* dan *password* atau PAP. Jadi jika Internet Service Provider menggunakan PAP, tetapi tidak akan kelihatan seperti sampah, dan hal ini belum tentu salah.
- Beberapa *server* meminta untuk memasukkan beberapa text untuk inisial baru kemudian menjalankan urutan dari PAP.
- Beberapa *server* PPP bersikap pasif, artinya mereka hanya menunggu sampai *client* yang melakukan *dial* mengirimkan paket *lcp* yang berlaku. Jika *server* PPP yang akan dihubungi bersikap pasif, maka pada layar tidak akan nampak seperti sampah.
- Beberapa *server* tidak akan menjalankan PPP sampai ada penekanan tombol ENTER.
- Ada *prompt* yang meminta untuk memasukkan *user name*.
- Ada *prompt* yang meminta untuk memasukkan *password*.

Jika *server* tidak secara otomatis menjalankan PPP maka perlu untuk menunggu *server* menampilkan *prompt* yang memberitahu bahwa sudah bisa memasukkan perintah untuk menjalankan PPP. Akan tetapi jika *server* secara otomatis menjalankan PPP, berarti sekali melakukan *login*, maka akan dapat langsung melihat tampilan teks seperti sampah pada layar, ini berarti PPP *server* mengirimkan informasi kepada komputer untuk melakukan *start up* dan mengkonfigurasi hubungan PPP.

Biasanya pada layar tampilannya seperti di bawah ini:

```
~y}#.!!}!} }8}!}$%U"}&} } } }%}&
...}"}{"} .~y}
```

Pada beberapa sistem PPP harus dijalankan pertama kali pada *server*. Hal ini biasanya disebabkan karena *server* sudah mengizinkan untuk *login* PPP dan *login* dengan menggunakan *user name/password* yang sama. Jika terjadi kasus seperti ini, maka berikan perintah lagi untuk *login*. Maka, sekali lagi akan terlihat tampilan seperti sampah pada layar. Hal ini menandakan bahwa *server* sudah siap untuk melakukan koneksi PPP.

2.8 Setting Name to Address Resolution

Manusia suka memberi nama sesuatu dengan kata-kata, akan tetapi lain halnya dengan komputer. Komputer memberi nama sesuatu dengan angka-angka. Pada jaringan TCP/IP, masing-masing komputer diberi nama yang unik, dan dikelompokkan menjadi grup-grup. Sebagai contoh, untuk Linux *workstation* diberi nama *archenland* dan terletak pada grup *interweft.com.au*, sedangkan manusia akan membaca alamat tersebut sebagai *archenland.interweft.com.au* (atau sering disebut sebagai FQDN - *fully qualified domain name*). Bagaimanapun, supaya komputer ini dapat berkomunikasi dengan komputer lain pada Internet, maka yang perlu dimengerti adalah nomor IP. Proses menerjemahkan nama komputer dan grupnya menjadi angka-angka yang digunakan pada Internet dinamakan Domain Name Service. Adapun yang terjadi adalah sebagai berikut :

- Pertama-tama harus mengetahui IP *address* dari komputer. Suatu aplikasi

membutuhkan informasi ini untuk memecahkan kode dengan menggunakan komputer Linux.

- Lalu dimasukkan ke dalam *local host file* (/etc/hosts dan/atau *domain name servers* yang mengetahui (biasanya terdapat di dalam /etc/host.conf);
- Jika terdapat pada *host file*, maka akan diberikan jawaban kembali.
- Jika domain name server telah ditetapkan, maka komputer baru dapat melakukan komunikasi dengan komputer yang diinginkan.
- Jika DNS komputer sudah mengetahui nomor IP dari komputer yang akan melakukan komunikasi, akan diberitahu. Jika belum tahu, maka akan mencari lagi ke server yang lain di Internet untuk mendapatkan informasi tersebut. Lalu server tadi memberikan informasi kembali kepada aplikasi yang memerlukan tersebut.

Pada saat melakukan koneksi dengan PPP, sistem operasi Linux perlu diberi tahu dimana bisa mendapatkan nama host untuk di terjemahkan menjadi nomor IP sehingga pada akhirnya dapat digunakan untuk menerjemahkan nama komputer menjadi nomor IP yang dibutuhkan. Salah satu cara yang dapat dilakukan yaitu dengan memasukkan semua nama *host* ke dalam file /etc/hosts, yang pada kenyataannya tidak mungkin dilakukan jika sedang melakukan koneksi ke Internet. Tetapi cara terbaik adalah dengan melakukan *set up* Linux supaya dapat melakukan penggantian dari nama yang didapatkan menjadi informasi berupa angka-angka secara otomatis. Pelayanan ini diberikan oleh *Domain Name Server* (DNS) system. Semua yang dibutuhkan hanyalah memasukkan nomor IP untuk *domain name server* ke dalam file /etc/resolv.conf.

2.9 Menggunakan PPP dan Hak Istimewa Pada Root

Dikarenakan PPP perlu untuk melakukan set up peralatan jaringan, maka orang yang mempunyai wewenang pada *root* perlu untuk mengganti kernel *routing table* dan sebagainya.

Jika user lain selain *root* melakukan *set up* koneksi PPP, maka *pppd* program akan menjadi *setuid root*. Jika /usr/sbin/pppd tidak melakukan *set up* dengan cara ini, maka sebagai *root* perlu menggunakan perintah: **chmod u+s /usr/sbin/pppd**

Apa yang dilakukan di sini adalah dengan menjalankan *pppd* dengan hak istimewa dari *root* walaupun masih sedang dijalankan *user* biasa. Hal ini mengizinkan *user* biasa untuk menjalankan *pppd* dengan wewenang yang diperlukan untuk melakukan *set up interface* jaringan dan kernel *routing table*.

Program yang menjalankan 'set uid root' sangat berpotensi untuk menciptakan hole pada sistem keamanan dan diharapkan untuk lebih berhati-hati di dalam membuat program 'suid root'. Beberapa program termasuk *pppd* biasanya ditulis dengan meminimalkan bahaya di dalam penggunaan 'suid root', sehingga dapat lebih terjamin keamanannya.

Apabila sebagai *root* menginginkan agar user lain pada sistem dapat melakukan koneksi dengan PPP, maka hal yang perlu dilakukan adalah membuat *ppp-on/off scripts* dapat dibaca/dieksekusi.

Seandainya tidak menginginkan ada orang lain dapat menggunakan koneksi PPP, meskipun punya *login* pada komputer Linux, contohnya anak-anak supaya tidak bisa melakukan koneksi ke Internet tanpa pengawasan, maka perlu dibuat suatu PPP grup, caranya *login* sebagai *root* dan *edit /etc/group* dan :

- Buat *pppd* *suid root*, yang hanya dimiliki oleh *user root* dan PPP grup, dengan tidak memberi ijin yang lain kepada file tersebut.
- Buat *ppp-on/off scripts* hanya dimiliki oleh *user root* and PPP grup
- Buat *ppp-on/off scripts* bisa dibaca/dieksekusi oleh group PPP
- Hapus hak akses yang lain untuk *ppp-on/off*.
- Dilakukan pencatatan jika ada user yang akan masuk ke PPP ke dalam PPP grup pada /etc/group.

Dengan melakukan demikian, maka *user* biasa tidak dapat melakukan *shut down* secara *software*, tetapi dengan paksaan. Hanya *user* sebagai *root* saja yang dapat

menjalankan *ppp-off script*. User lain hanya dapat mematikan modem, atau memutuskan saluran telepon dari internal modem.

Alternatif lain, atau mungkin cara yang lebih baik untuk melakukan *set up* seperti ini adalah menggunakan program *sudo*. Program ini menawarkan keamanan yang lebih canggih dan dapat di *set up* supaya user lain hanya dapat mengaktifkan/menonaktifkan hubungan dengan menggunakan *script*. Dengan menggunakan *sudo* akan membuat user untuk mengaktifkan/menonaktifkan koneksi PPP dengan benar dan aman.

2.10 Setting File-File yang Diperlukan Untuk PPP

Diperlukan *login* sebagai *root* untuk membuat direktori dan melakukan proses edit suatu file pada saat melakukan *set up* PPP, supaya PPP dapat diakses oleh semua user. PPP menggunakan sekumpulan file untuk melakukan koneksi dan *set up* koneksinya.

Pada Red Hat Linux lokasi instalasi terletak pada `/usr/doc/ppp-2.2.0f-2/scripts`. Apabila direktori tidak ditemukan, berarti direktori tersebut belum ada dan perlu dibuat dengan hak dan ijin sama seperti di atas. Tetapi jika direktori tersebut sudah ada, maka akan berisi suatu file *template option* yang bernama `options.tpl`. File ini berisi penjelasan-penjelasan tentang PPP option, selain yang terdapat pada manual *pppd*. Akan tetapi jika bisa menggunakan file ini sebagai basis dari file `/etc/ppp/options`, maka akan lebih baik untuk membuat file *option* sendiri yang tidak berisi semua penjelasan seperti yang ada pada *template*, sehingga ukuran file akan menjadi lebih pendek dan lebih mudah untuk dipelajari.

Jika mempunyai *multiple serial line/modem* (khususnya untuk PPP server), maka pembuatan file `/etc/ppp/options` yang berisi option tentang semua *serial port* yang mendukung *dial in/out* dan *set up* individual *file option* pada masing-masing line pada koneksi PPP dengan *individual setting* diperlukan pada masing-masing *port*.

File *option* yang spesifik tersebut mempunyai nama `options.ttyx1`, `options.ttyx2` and sebagainya (dimana x merupakan

nomor dari *serial port*). Bagaimanapun, jika hanya untuk melakukan koneksi *single PPP*, maka cukup menggunakan file `/etc/ppp/options`. Alternatif yang lain, dapat meletakkan semua option sebagai argumen pada perintah *pppd* itu sendiri.

Akan lebih memudahkan untuk proses pemeliharaan di waktu yang akan datang jika setup menggunakan file-file `/etc/ppp/options.ttySx`. Jika ingin menggunakan PPP untuk melakukan koneksi ke berbagai nomor site yang berbeda, maka dapat dibuat suatu file *option* untuk masing-masing *site* pada `/etc/ppp/options.site` dan menspesifikasikan file *option* sebagai parameter untuk perintah PPP yang dihubungi.

3. PEMAKAIAN NOMOR IP DYNAMIC PADA INTERNET SERVICE

Kebanyakan Internet Service Provider memberikan nomor IP *dynamic* bagi para *user*-nya. Pertama, terdapat pelayanan yang bersifat keluar yang dapat bekerja dengan baik. Itulah sebabnya dapat melakukan pengiriman e-mail dengan menggunakan *sendmail*, *ftp file* dari *remote site*, *finger user* pada komputer yang lain, dan melakukan *browser* dengan menggunakan *web browser*, dan sebagainya.

Secara khusus, *user* dapat membalas e-mail yang diterima dan disimpan pada komputer saat *offline*. *Mail* disimpan pada antrian yang akan dikirimkan pada saat *dial* kembali ke Internet Service Provider.

Pada kenyataannya, komputer tidak akan melakukan koneksi ke Internet 24 jam terus menerus per hari, dan tidak mungkin mempunyai nomor IP yang sama setiap kali melakukan koneksi. Jadi sangat tidak mungkin jika menerima e-mail yang ditujukan langsung ke komputer, dan sangat sulit untuk *set up* suatu web atau ftp server bagi orang lain untuk akses. Se jauh ini Internet tidak memberikan suatu komputer dengan nomor yang unik, komputer yang berhubungan secara permanen tidak mempunyai nomor IP yang unik. Perlu diingat bahwa komputer yang lain bisa saja menggunakan nomor IP tersebut pada saat dialokasikan pada *dial in*.

Jika seseorang melakukan *set up* WWW, maka tidak akan diketahui oleh user yang lain sampai komputer milik orang tersebut terkoneksi dan ada nomor Ipnnya. Terdapat beberapa cara untuk mendapatkan informasi seperti ini yaitu dengan menelepon mereka, mengirim email untuk memberitahukan atau menggunakan file “.plan” pada *shell* Internet Service Provider dengan anggapan bahwa provider mengizinkan akses *shell* dan *finger*.

Untuk kebanyakan user, hal tersebut tidak menjadi masalah. Kebanyakan orang menginginkan untuk mengirim dan menerima email dengan menggunakan *account* pada Internet Service Provider dan melakukan koneksi keluar ke WWW, ftp dan *server* yang lain pada Internet. Jika menginginkan koneksi yang bersifat ke dalam, misalnya koneksi pada *server*, maka perlu mendapatkan nomor IP *static*.

Dengan menggunakan nomor IP *dynamic*, maka dapat mengkonfigurasi *sendmail* pada komputer untuk mengirimkan email secara lokal. Konfigurasi dari *sendmail* tersebut sangat sulit dan tak jelas.

4. HUBUNGAN DUA LOCAL AREA NETWORK DENGAN MENGGUNAKAN PPP

Pada dasarnya tidak ada perbedaan antara menghubungkan satu Linux PC pada PPP server dan menghubungkan dua LAN dengan menggunakan PPP pada suatu komputer pada masing-masing LAN. Perlu diingat bahwa PPP adalah *peer to peer protocol*.

Yang perlu diperhatikan di sini adalah perlunya mengerti mengenai bagaimana *routing* itu dibuat. Untuk menghubungkan dua LAN, maka perlu menggunakan nomor IP jaringan yang berbeda. Dan perlu untuk menggunakan nomor IP *static* atau IP *masquerade*.

4.1 Pemilihan Nomor IP

Nomor IP digunakan untuk interface PPP. Jika menggunakan nomor IP *static*, maka perlu melakukan *dial in* pada nomor telepon yang spesifik. Untuk mendapatkan spesifikasi modem dan nomor *port* untuk koneksi perlu mengedit file */etc/ppp/options*

[.ttyXX]. Mungkin perlu untuk pergantian pada file */etc/ppp/options* dan membuat file *option.ttyXX* yang tepat untuk koneksi yang lain.

4.2 Pemilihan Rute yang Akan Digunakan

Perlu diketahui bahwa untuk menyusun suatu paket yang akan dikirimkan melalui LAN melalui route yang sudah disediakan oleh hubungan PPP. Proses ini terbagi dalam dua tahap.

Pertama, perlu membentuk suatu route dari komputer yang menjalankan hubungan PPP ke jaringan yang akan dituju. Jika hubungan ke Internet, maka hal ini dapat diatasi dengan *default route* yang dibentuk oleh pppd itu sendiri pada koneksi dengan menggunakan option 'defaultroute' pada pppd. Jika hubungan yang terjadi hanya menghubungkan dua LAN, maka spesifikasi jaringan untuk *route* harus ditambahkan pada masing-masing jaringan yang terhubung. Caranya yaitu dengan menggunakan perintah 'route' untuk masing-masing jaringan pada script */etc/ppp/ip-up*.

Kedua, yang diperlukan adalah memberitahu pada komputer yang lain pada LAN bahwa Linux komputer ini sesungguhnya adalah 'gateway' dari jaringan sejauh hubungan menggunakan ppp. Tentu saja, administrator jaringan pada hubungan yang lain menggunakan metode ini juga. Sesungguhnya yang diperlukan di dalam melakukan pengiriman paket yaitu spesifikasi rute jaringan, bukan *default route*.

4.3 Keamanan Jaringan

Jika menghubungkan LAN ke Internet dengan menggunakan PPP, maka diperlukan sistem untuk menjaga keamanannya agar terhindar dari orang-orang yang tidak berhak masuk ke dalam sistem dan melakukan pengacauan.

5. IMPLEMENTASI PPP SERVER

Seperti yang telah dijelaskan sebelumnya, ada bermacam-macam cara untuk melakukan hal ini. Akan tetapi yang akan dijelaskan adalah yang menggunakan Cyclades *multi-port serial card* dan *rotary dial in* pada saluran telepon.

5.1 Compile Kernel

Semua yang berkaitan dengan mengkompile kernel telah dijelaskan pada bab sebelumnya. Untuk PPP server, perlu dimasukkan IP forwarding pada kernel, Fire walls, accounting, dan sebagainya. Jika menggunakan multi-port *serial card*, maka perlu juga *driver*-nya pada kernel.

5.2 Analisa Sistem Server

Misalkan melakukan dial up account PPP dan shell account dengan menggunakan pasangan user name/password yang sama. Hal ini memiliki keuntungan bahwa seorang user hanya membutuhkan satu *account* untuk berbagai macam tipe koneksi.

Proses agar seorang user dapat melakukan hubungan PPP yaitu:

- *Dial* ke rotary dialer (nomor telepon tunggal yang menghubungkan ke sekumpulan modem untuk mencari modem yang tidak dipakai).
- *Login* dengan menggunakan *user name* dan *password* yang sah.
- Pada *shell prompt*, ketikkan perintah ppp pada *server*.
- Jalankan PPP pada masing-masing PC (bisa lewat windows, DOS, dan Linux).

Server menggunakan file individual */etc/ppp/options.ttyXX* untuk masing-masing *port dial* in dan mengeset *remote* IP untuk alokasi nomor IP dynamic. Pada saat user memutuskan hubungan, maka pppd mendeteksi hal ini dan memberitahukan pada modem untuk *hang up*, dan memutuskan hubungan PPP pada waktu yang sama.

5.3 Analisa Perangkat Lunak

Adapun perangkat lunak yang diperlukan adalah sebagai berikut :

- Sistem Operasi Linux.
- Pppd dengan versi yang sama pada kernel.
- Program yang dapat menangani komunikasi modem.

5.4 Membuat File Options Untuk PPP

Perlu untuk melakukan set up pada direktori */etc/ppp/options* dengan pilihan-pilihan yang diperlukan untuk semua port dial up. Pilihan-pilihan tersebut adalah sebagai berikut :

- *asynmap* 0
- *netmask* 255.255.254.0
- *proxyarp*
- *lock*
- *crtsets*
- *modem*

Adapun hal yang perlu diperhatikan adalah perlunya memberikan alokasi nomor IP *dynamic* bagi para pemakai yang melakukan *dial up*. Caranya yaitu dengan mengalokasikan nomor IP untuk masing-masing *port* pada komputer yang melakukan *dial up*. Buat */etc/ppp/options.ttyXX* untuk masing-masing *port dial up*. Secara sederhana, yaitu dengan meletakkan nomor IP lokal dan nomor IP yang akan digunakan pada masing-masing *port*.

6. KESIMPULAN

Untuk dapat melakukan komunikasi antar modem dapat menggunakan Point To Point Protocol. Point To Point Protocol ini bersifat *peer to peer*, yang artinya tidak bisa dibedakan penggunaannya pada *server* atau *client*. Karena PPP dapat digunakan pada *server* maupun *client*.

Jika suatu komputer melakukan *dial in* disebut sebagai *client*, dan komputer yang bertugas menerima/menjawab telepon, melakukan pengecekan seperti misalnya *user name* dan *password* itu disebut sebagai *server*. Proses pembuatan PPP pada *client* sangat mudah. Hal yang perlu diperhatikan di dalam proses pembuatan PPP pada *client* adalah pengenalan pada perangkat lunak, apakah sudah ter-*install* atau belum. Jika belum ter-*install* maka bisa didapatkan pada Internet. Sedangkan jika sudah ter-*install* maka proses selanjutnya adalah meng-*compile* pada kernel, agar PPP dapat dijalankan.

Untuk menjalankan PPP dapat dilakukan dengan menjalankan *ppp-on* dan *ppp-on dialer*, sedangkan untuk memutuskan hubungan dilakukan dengan mengetikkan *ppp-off*.

Sedangkan untuk pembuatan PPP pada *server* hal yang perlu diperhatikan adalah mengenai sistem keamanan, untuk menghindari *user* yang tidak diijinkan untuk masuk ke dalam sistem. Terlebih lagi jika akan terhubung ke Internet.

DAFTAR PUSTAKA

1. Linux Point to Point Protocol Howto. <http://www.tldp.org/docs.html#howto>
2. Purcell, Robinson. 1996. *The Linux Bible: The GNU Testament* Purcell, Yggdrasil Computing.
3. John Purcell. 1997. *Linux Man: The Essential Man Pages for Linux*. Linux System Labs, 1997.
4. Olaf Kirch. *Linux Network Administrator's Guide*. O'Reilly & Associates. January, 1995.
5. Jim Mohr. *Linux User's Resource*. Prentice Hall, 1997.
6. Andrew Anderson. The Point to Point Protocol. [<http://www.tldp.org/LDP/nag/node107.html>], 1996.
7. David Pitts, Bill Ball. *Red Hat Linux 6 Unleashed*. USA: SAMS, 1999.