

# Membangun server dengan Open Source

I Made Wiryana SSi, SKom, MSc\* dan I Wayan S Wicaksana SSi MEng<sup>†</sup>

14 Oktober 2002

## Ringkasan

Tulisan ini akan mencoba memberikan gambaran mengenai aplikasi-aplikasi Open Source. Istilah Open Source digunakan ketimbang Freeware (yang mengacu pada GPL), karena memiliki cakupan lisensi yang lebih luas. Dalam tulisan ini tidak dilakukan penjabaran detail ataupun konfigurasi yang perlu dilakukan untuk setiap program aplikasi. Tulisan ini hanya bertujuan untuk memperkenalkan beragam aplikasi Open Source yang memberikan kemungkinan untuk dimanfaatkan dalam upaya membangun suatu server. Istilah server telah berkembang dari yang sekedar menangani Web atau mail kini telah berkembang menjadi berbagai fungsi. Open Source dapat digunakan untuk kebutuhan membangun server tersebut.

## 1 Pendahuluan

Pada saat ini teknologi informasi (TI) telah mulai bermetamorfosa menjadi suatu tahapan teknologi yang pervasif. Artinya teknologi tersebut akan terasa keberadaannya ketika tidak ada (atau tak bekerja). Dengan kata lain kita sudah mulai tergantung dengan bantuan teknologi informasi. Kemajuan perkembangan Internet dan World Wide Web (WWW) telah menunjukkan suatu langkah ke arah ini. Konsekuensi dari sistem informasi yang menjadi pervasif adalah timbulnya dampak yang besar pada masyarakat secara luas. Akan banyak industri yang berubah atau digantikan sama sekali. Atau juga akan banyak tumbuh industri baru sesuai dengan kebutuhan perkembangan teknologi informasi itu.

Informasi dapat dipandang sebagai media pertukaran murni. Walaupun ada biaya untuk mengakses, mendistribusikan, ataupun menyimpan informasi, informasi itu dianggap tak ada biayanya. Pada organisasi modern, dan dalam bahasan ekonomis secara luas, informasi telah menjadi komoditas yang sangat berharga, dan telah berubah dan dianggap sebagai sumber daya habis terpakai, bukannya barang bebas. Dalam suatu organisasi perlu dipertimbangkan bahwa informasi memiliki karakter yang multivalued, dan multidimensi. Dari sisi pandangan teori sistem, informasi memungkinkan kebebasan beraksi, mengendalikan pengeluaran, mengefisienkan pengalokasian sumber daya dan waktu. Sirkulasi informasi yang terbuka dan bebas merupakan kondisi yang optimal untuk pemanfaatan informasi.

Di samping iming-iming keuntungan dari pemanfaatan teknologi informasi, sangatlah tidak realistis bila mengasumsikan bahwa teknologi informasi tidak menimbulkan permasalahan dalam penerapannya. Informasi jelas dapat disalah-gunakan. Polusi informasi, yaitu propagasi informasi yang salah, dan pemanfaatan informasi (baik benar atau salah) untuk mengendalikan hidup manusia tanpa atau dengan disadari merupakan suatu akibat dari penyalah-gunaan ini. Misinformasi akan terakumulasi dan menyebabkan permasalahan pada masyarakat. Semakin tua suatu masyarakat, semakin besar pula kemungkinan mengakumulasi beragam misinformasi ini, dan mulailah mengalami berbagai dampak buruk.

Pergeseran fokus perhatian ke sisi manusia membuat kita harus merevisi perhatian kita pada perkembangan TI yang telah ditempuh selama ini. Di tambah lagi di tengah situasi kesulitan

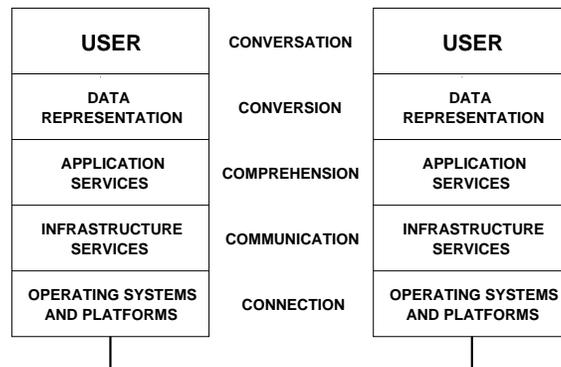
---

\*Dosen tetap Universitas Gunadarma, sedang melanjutkan studi doctoral di RVS Arbeitsgruppe, Universitas Bielefeld Jerman. Kolumnis majalah Infolinux.

<sup>†</sup>Kepala Jurusan Teknik Komputer Universitas Gunadarma.

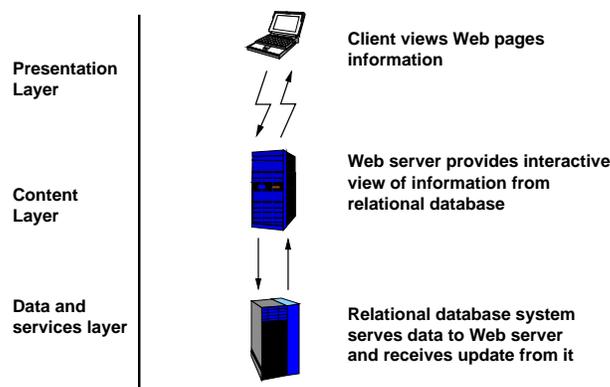
ekonomi seperti sekarang ini. Sehingga sudah seyogyanya perhatian kita bukan saja mengikuti trend yang menghabiskan dana untuk pembelian perangkat lunak/keras, akan tetapi kita harus mempertimbangkan pengkayaan kemampuan SDM (*brainware*) yang dimiliki.

Mekanisme Internet awalnya sangat sederhana, karena dikembangkan hanya untuk menyediakan data-data atau publikasi ilmiah antar rekan-rekan ilmuwan. Kini Internet telah berkembang dan menyediakan berbagai layanan. Sehingga dikenalnya konsep "Service" yang lebih kompleks daripada sekedar menyediakan dokumen di suatu situs Internet. Konsep service ini lebih umum dibanding konsep lapisan jaringan yang sangat teknis



Gambar 1: Standard DoD

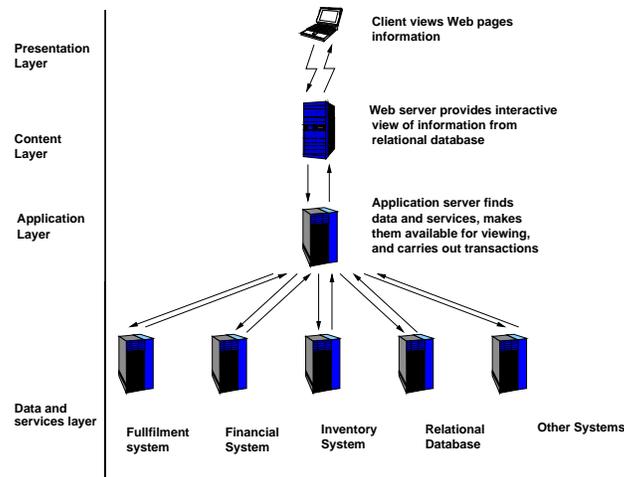
Internet, awalnya dikembangkan untuk menghubungkan antar pihak yang saling dipercaya untuk tujuan saling bertukar menukar informasi. Walau merupakan proyek Departemen Pertahanan Amerika, Internet digunakan dan dikembangkan untuk tujuan kolaborasi dunia akademi yang serba terbuka. Perkembangan Internet yang pesat dan kini telah menjadi suatu jaringan raksasa yang saling menghubungkan berbagai jaringan. Pemanfaatannya di bidang bisnis menjadikan terjadinya pergeseran model. Dari bentukan komunitas pengguna Internet yang cenderung berupa suatu *Gemainschaft* dengan norma internal dan tradisi yang diatur berdasarkan status dan didorong oleh kecintaan, kewajiban serta kesamaan pemahaman dan tujuan, Sekarang telah bergeser dan cenderung menjadi suatu *Gesselschaft* yang terdiri dari individu (organisasi) yang memiliki interest masing-masing yang saling berkompetisi untuk kepentingan material sehingga berbentuk pasar bebas. Pada bentuk pertama bisa dikatakan tak ada batasan antara privat dan publik, sedang pada yang kedua terjadi perbedaan secara jelas.



Gambar 2: Generasi 1 Internet

Perubahan fungsi, serta komunitas pengguna Internet tampaknya belum diikuti dengan perubahan drastis teknologi jaringan yang mendasarinya. Teknologi yang digunakan relatif masih memanfaatkan TCP/IP yang serba terbuka. Terbuka di sini bukan berarti source code atau stan-

dardnya diketahui banyak orang, tetapi dalam mekanismenya yang masih membuka alamat tujuan dan pengirimnya. Ketertutupan informasi yang berkaitan dengan suatu protokol bukan merupakan suatu jaminan bahwa protokol itu akan lebih aman. Seperti diketahui algoritma atau mekanisme kriptografi yang menjadi sandaran usaha penyusunan jalur komunikasi aman pun menggunakan algoritma yang mekanismenya diketahui oleh orang banyak. Beberapa protocol telah dikembangkan untuk mengatasi kekurangan protocol TCP/IP, seperti QoS, IPSEC (IP Secure), IP-NG (IP New Generation).



Gambar 3: Generasi ke 2

## 2 Pertimbangan pemilihan

Dalam membangun server yang kritis maka masalah biaya perlu dipertimbangkan dengan seksama, dalam hal ini termasuk :

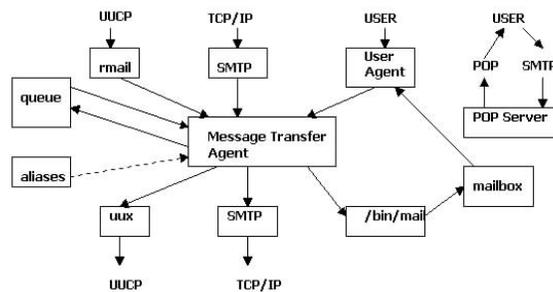
- **Biaya perangkat keras**, termasuk perangkat keras tambahan yang dibutuhkan karena merubah sistem operasi. Ataupun biaya upgrade perangkat keras yang disebabkan perangkat lunak mengalami up-grade. Perangkat lunak Open Source dapat beroperasi dengan menggunakan hardware yang lebih rendah, karena memungkinkan diubah, dikompilasi ulang, dan dioptimaskan sesuai dengan perangkat keras yang ada. Perlu dipertimbangkan biaya perangkat keras ini akan meluruh dengan cepat, mengingat perkembangan dunia komputer yang cepat.
- **Biaya perangkat lunak** baik pembelian ataupun lisensi. Termasuk biaya yang harus dikeluarkan ketika melakukan upgrade. Dalam hal ini harus sudah mulai berbicara dalam kerangka legalitas, program bajakan sebaiknya tidak digunakan lagi. Sehingga perhitungan biaya harus termasuk perhitungan pembelian perangkat lunak sistem operasi, dan juga perangkat lunak aplikasi.
- **Biaya perawatan**, termasuk perjanjian dukungan teknis, biaya yang hilang ketika sistem *down*. Dalam menghitung biaya perawatan, sebaiknya bukan saja bergantung pada 'seberapa mudah interaksi perawatant tersebut, tetapi juga termasuk seberapa sering perangkat tersebut tak bekerja (*down*). Sebab setiap kerusakan akan memakan biaya perawatan, baik tenaga maupun waktu. Dengan kata lain berbicara masalah mengadministrasi sistem, tidak boleh dilupakan dengan mengaitkannya dengan kehandalan sistem (reliabilitas). Tentu saja biaya untuk personal yang menangani sistem administrasi perlu sangat dipertimbangkan

- **Biaya yang harus dikeluarkan ketika data hilang** karena kesalahan program di sistem operasi atau perangkat keras yang dipersyaratkan oleh sistem operasi. Masalah hilangnya data, atau downnya sistem, serta pekerjaan tambahan akibat virus harus juga dipertimbangkan sebagai biaya yang harus dikeluarkan.
- **Biaya pelatihan**, harus dipertimbangkan biaya pelatihan awal dan biaya pelatihan ulang setiap ada perubahan versi sistem. Termasuk apakah pengetahuan yang diperoleh pada pelatihan tersebut merupakan pengetahuan yang bersifat 'generic' atau hanyalah pengetahuan pengoperasian suatu produk belaka. Sepintas lalu penggunaan Open Source akan meminta biaya investasi pelatihan yang cukup tinggi. Tetapi hal itu akan terbayar dengan dasar pengetahuan yang mereka dapatkan tidak hanya sekedar sebagai pengguna, tapi bisa berkembang sebagai penyedia kebutuhan teknologi Informasi perusahaan itu di masa mendatang. Juga perlu difikirkan investasi pelatihan ulang (retraining) yang harus dilakukan.
- **Biaya migrasi**, termasuk biaya ekstensibilitas, dan skalabilitas untuk perkembangan di masa mendatang.

Dari sisi biaya perangkat keras dan lunak sudah barang tentu GNU/Linux dan aplikasi Open Source menawarkan suatu solusi yang sangat menarik sekali. Di samping rendahnya biaya investasi perangkat lunak, dengan perangkat keras yang sama maka kecepatan yang ditawarkan oleh GNU/Linux lebih tinggi dari OS komersial. Permasalahan utama dari akseptibilitas bisnis untuk GNU/Linux ini adalah persepsi para manajer Teknologi Informasi. Sebagian besar mereka belum mendengar mengenai GNU/Linux ini secara jelas.

### 3 Layanan email

Mail merupakan salah satu aplikasi kritis dan utama bagi suatu server Internet. Sehingga sangat penting suatu server Internet mempertimbangkan keberadaan aplikasi mail server.



Gambar 4: Mekanisme mail di server Internet

MTA (Mail Transfer Agent) atau lebih dikenal dengan mail server merupakan salah satu komponen penting pada server Internet. Memilih berbagai MTA sangat sulit, karena setiap orang memiliki kebutuhan dan pertimbangan yang berbeda. Sangat perlu dipahami pertimbangan dan kebiasaan sistem administrator dalam menggunakan jenis MTA-nya, karena email merupakan suatu layanan yang sangat penting pada server internet. Membangun server mail biasanya dilakukan dengan pertimbangan yang lebih teliti ketimbang web server sebab setiap situs harus mendaftarkan mail exchanger yang digunakannya pada global DNS.

Berikut ini dijabarkan beberapa contoh MTA yang tersedia bebas yang memiliki fitur hampir sama tetapi dengan beberapa kekhasannya. Semua MTA tersebut dapat menangani beban lalu lintas email yang besar, dapat berinteraksi dengan database dalam berbagai format, tidak mudah dieksploit, tersedia source code, dan juga ada dokumentasi pihak ke-3 yang tersedia. Tersedia source code secara bebas. Memang aplikasi seperti MS Exchange atau Lotus Notes sepintas lalu bisa dimanfaatkan sebagai mail server, tetapi sebetulnya aplikasi ini belum cukup mampu

untuk menanggapi beban email yang banyak seperti halnya MTA berikut ini. Kebutuhan pengguna akan jenis MTA yang digunakan ini beragam. Berbagai kriteria biasanya digunakan untuk pertimbangan misal :

- Kemudahan administrasi termasuk dari sisi kebiasaan administrator sistem.
- Kinerja
- Sekuriti
- Kebutuhan jangka panjang

Pada dasarnya tidak ada satu MTA yang dapat memuaskan semua pengguna. Setiap pengguna memiliki kebutuhan dan keterbatasannya masing-masing.

### 3.1 sendmail

Sendmail [<http://www.sendmail.org>] merupakan MTA yang paling banyak digunakan, diperkirakan sekitar 50% dari server email di Internet menggunakannya. Sangat berpengaruh pada perkembangan Internet itu sendiri. Kelemahan sendmail sebagian besar disebabkan karena usia, dan perubahan dasar design (keadaan Internet masa awal didisainnya). Mungkin bila dari awal Eric Allman (disainer Sendmail) mengira bahwa sendmail akan sesukses ini, maka sendmail tentunya akan didisain lain.

MTA ini diimplementasikan sebagai suatu program monolitik besar yang mengerjakan segalanya. Memang suatu program besar memudahkan penggunaan data antara bagian sistem, di program monolitik tersebut. Sayangnya suatu program besar ini memiliki kencerungan kesalahan fatal. Hal ini yang sering menimbulkan permasalahan sekuriti pada sendmail.

Sendmail memiliki konfigurasi yang "aneh", beberapa masalah sekuriti dan disain yang berlandaskan model Unix di tahun 80-an. Memang banyak beberapa masalah remote exploit pada sendmail. Tetapi saat ini sendmail paling banyak digunakan. Walau sendmail ini kurang efisien, tetapi sendmail ini tetap paling banyak digunakan, karena banyak sysadm telah terbiasa dan mengetahui cara menanganinya untuk menghindari masalahnya. Tetapi pada tahun 1999 dan 2000 sendmail telah mengalami perbaikan baik dari sisi fitur, kinerja maupun sekuriti. Pada tahun 2001 sendmail telah memiliki fitur yang kayak dan secara default sendmail telah terinstal pada beberapa Unix. Dan untuk kebutuhan sederhana atau pada umumnya setting bawaan telah bekerja tanpa perlu dilakukan modifikasi.

Konfigurasi yang memanfaatkan perangkat bantu M4 sangat disukai oleh banyak pengguna yang telah bertahun-tahun terbiasa menggunakannya. Memang saat ini penggunaan sendmail secara prosentase cenderung menurun, tetapi perlu diingat, ketika awal Internet dapat dikatakan 100% mail server menggunakan sendmail. Baru ketika tahun 1996-an menurut menjadi 80% dan makin menurun. Pada survei 1999 sekitar 60% mail server menggunakan sendmail.

### 3.2 qmail

Tujuan utama dari disain qmail [<http://www.qmail.org>] adalah menggantikan fungsi sendmail, serta memberikan keamanan yang lebih baik dan kinerja proses. Qmail menggunakan hierarki yang ketat dari program untuk menjalankan program dengan urutan yang tertentu dan menghilangkan komponen yang tak perlu setelah digunakan. Pendekatan ini memiliki keuntungan karena memberikan isolasi antar proses dengan baik, tetapi memiliki *overhead* yaitu beban ketika membuat proses baru dan komunikasi antar proses. Untuk menekan ini maka pembagian tugas antar proses harus dilakukan secara tepat. MTA qmail ini tidaklah kompatibel dengan sendmail dalam hal konfigurasi ataupun format penyimpanan berkasnya, sehingga tak dapat menggantikan sendmail begitu saja. qmail telah banyak menarik minat orang yang memperhatikan sekuriti dan beberapa fiturnya.

### 3.3 postfix

Postfix [<http://www.postfix.org>] merupakan MTA yang didisain agar lebih aman dan memiliki model kerja yang terdiri dari proses-proses kecil yang terpisah yang memiliki tugas khusus tanpa relasi *parent-child*. Dengan demikian diharapkan sistem memiliki isolasi komponen yang lebih baik ketimbang menggunakan satu program besar seperti sendmail. Di samping itu, pendekatan yang digunakan postfix ini memudahkan layanan untuk penulisan ulang alamat (*addres rewrite*) pada setiap komponen tanpa harus

Postfix diimplemetasikan sebagai suatu master server yang menjalankan daemon postfix untuk proses lainnya sesuai kebutuhan, misal daemon untuk menerima atau mengirim mail, daemon untuk memproses penyaluran mail lokal dan lain sebagainya. Proses ini dibuat dengan jumlah sesuai dengan konfigurasi yang digunakan, dan akan digunakan ulang. Dihilangkan setelah menganggur (*idle*) dalam interval waktu tertentu. Pendekatan ini menurunkan *overhead* ketika membuat proses (seperti yang terjadi pada qmail) dan masih tetap memberikan isolasi yang baik pada proses yang terpisah.

Postfix seperti halnya qmail, ditulis dengan pertimbangan sekuriti oleh Wietsa Venema (yang membuat tcpwrap). Postfix dapat dikatakan memenuhi kebutuhan antara qmail dan exim. Terdiri dari beberapa komponen program (tetapi lebih sedikit dari qmail), dan memiliki berkas konfigurasi monolitik. Postfix memiliki penekanan pada sekuriti, tetapi tanpa menggunakan proses pengelolaan yang khusus dan berbeda dengan pola pengelolaan Unix biasa.

Berkas konfigurasi qmail cukup fleksible walau belum sefleksible Exim. Postfix memiliki kinerja sangat cepat, dan juga dapat menggantikan sendmail secara langsung. Karena didisain agar kompatibel dan dapat langsung menggantikan sendmail tanpa perubahan yang berarti bagi program lainnya yang menggunakan MTA sendmail (misal mailing list seperti majordomo).

### 3.4 exim

Exim [<http://www.exim.org>] sebuah MTA yang dikembangkan oleh Universitas Cambridge untuk digunakan pada sistem Unix-nya yang terhubung ke Internet. Penulis exim terinspirasi kerjanya ketika mengerjakan smail, dan beberapa permasalahan yang diperolehnya ketika menggunakan sendmail. Exim dapat digunakan sebagai mailer serba guna pada mesin Unix. Exim tidak didisain dengan pemikiran yang berbeda total seperti halnya qmail, sehingga Exim dapat langsung menggantikan sendmail. Exim beroperasi dan tampak seperti halnya daemon biasa, dengan berkas konfigurasi yang monolitik, dengan daemon yang monolitik dan beberapa berkas log dan spool.

Hingga saat ini belum memiliki reputasi keamanan yang buruk, dan dapat menangani beban berat serta memiliki fasilitas integrasi yang baik. Exim dapat dikembangkan dengan berbagai cara.

Yang menarik dari MTA selain sendmail ini adalah kesamaan idea dan kode yang digunakan. Postfix menggunakan pustakan Perl Compatible Regular Expression yang dikembangkan untuk Exim. Exim memahami Constant Database format yang dikembangkan untuk qmail dan format berkas mail Maildir yang dikembangkan oleh qmail. Postfix dapat menggunakan Constant Database Format. Keterbukaan pengembangan ini menjadikan MTA tersebut dapat saling berinteroperasi.

### 3.5 Kebutuhan pendukung lainnya

Di samping kebutuhan MTA, maka dibutuhkan juga layanan yang menandai transport email dari server ke client agar email dapat dibaca. Dua metoda utama yang sering digunakan yaitu POP (Post Office Protocol) dan IMAP (Internet Message Access Protocol). Metoda yang telah lama ada seperti UUCP (Unix to Unix Copy Protocol) saat ini jarang digunakan, padahal lebih efisien ketimbang IMAP atau POP untuk situasi seperti di Indonesia.

Untuk menyediakan layanan POP/IMAP tersebut maka dapat digunakan :

- **Qpopper**

Qpopper [<http://www.eudora.com/qpopper/>] merupakan suatu server untuk melayani pro-

tokol POP3. Biasanya digunakan dengan MTA seperti sendmail atau smail. Qpopper disain dengan tujuan keamanan, kestabilan, dan kinerja. Qpopper mencegah terjadinya kerusakan spool. Menyediakan fitur seperti bulletin, dukungan perangkat wireless, APOP dan integrasi dengan Kerberos, drac dan sebagainya.

- **UW-IMAP**

UW-IMAP [<http://www.washington.edu/imap/>] merupakan server IMAP yang dikembangkan oleh Universitas Washington.

- **Cyrus IMAP**

Cyrus IMAP [<http://asg.web.cmu.edu/cyrus/imapd/>] berbeda dengan implementasi IMAP lainnya, karena didisain untuk dijalankan pada suatu server yang terisolir, dimana user biasa tak diperbolehkan login. Database mailbox disimpan sebagai bagian dari filesystem yang bersifat privat ke sistem Cyrus IMAP. Pengguna mengakses email melalui protokol IMAP, POP3. Disain database privat menyediakan keuntungan seperti efisiensi, skalabilitas dan kemudahan pengelolaan. Koneksi baca/tulis secara bersamaan pada mailbox yang sama dimungkinkan. Server IMAP ini mendukung Access Control List (ACL) untuk mailbox dan kuota.

- **Courier IMAPD**

Courier-IMAPD [<http://www.inter7.com/courierimap/>] menyediakan akses IMAP ke format Maildir. Server IMAP ini tidak menangani bekas format **mailbox** (`/var/spool/mail`). Tetapi khusus ditujukan untuk Maildir. Maildir adalah format berbasis direktori yang awalnya dikenalkan oleh Qmail, dan juga diadopsi sebagai model penyimpanan berkas email alternatif oleh Exim dan postfix.

## 4 Layanan milis (*mailing list*)

Milis adalah suatu fasilitas yang lazim digunakan untuk melakukan diskusi via email. Biasanya digunakan program yang berfungsi sebagai pengelola milis, yang akan menyebarkan 1 email ke seluruh anggota milis. Penggunaan program manajer milis (*mailing list manager*) ini tidak saja untuk diskusi virtual, tetapi juga dapat digunakan untuk mengirimkan newsletter ke para kustomer, mendistribusikan perangkat lunak ke para kustomer, memberikan berita sekuriti dan berbagai kegiatan untuk berkomunikasi berdasarkan suatu kepentingan yang sama.

Banyak cara untuk menerapkan milis ini. Cara tersederhana, menggunakan satu alamat tunggal yang akan memforwardkan email ke beberapa orang yang merupakan anggota milis tersebut. Cara sederhana ini tentu saja memiliki beberapa kelemahan, misal pengguna tak dapat menambahkan namanya sendiri atau tak dapat menghapus namanya sendiri. Juga tak ada cara untuk membatasi siapa yang boleh mengirimkan email ke milis tersebut, dan beberapa hal lainnya. Sehingga sebagian besar orang memilih menggunakan perangkat lunak yang sering disebut *mailing list manager* (MLM).

Banyak perangkat lunak Open Source tersedia untuk melakukan fungsi MLM ini, rata-rata memiliki fitur seperti pengguna dapat berlangganan atau berhenti berlangganan, kemungkinan mengelola pengguna, kemungkinan membatasi pengiriman email. Beberapa MLM juga memberikan fitur tambahan seperti kemungkinan memfilter spam email.

### 4.1 Majordomo

Majordomo [<http://www.greatcircle.com/majordomo/>] ini merupakan salah satu program manajer milis yang banyak digunakan. Ditulis menggunakan Perl, sehingga untuk menggunakannya, Perl harus terinstal dengan benar. Setelah program ini terinstal, maka operasi untuk mengelola milis dapat dilakukan melalui email, tanpa perlu campur tangan postmaster dari site tersebut.

Juga tersedia interface dengan Web untuk melakukan hal ini yang bernama **MajorCool**. MajorDomo ini mengendalikan daftar alamat yang diberikan kepada MTA (seperti sendmail, postfix,

smail). Program ini sendiri tidak melakukan pengiriman mail (walau memiliki skrip untuk memformat dan mengarsip pesan).

Majordomo mendukung berbagai jenis milis seperti milis termoderasi, milis bebas, pengelolaan dapat dilakukan melalui mail. Memiliki dukungan *digest* (ringkasan email harian atau bulanan). Memiliki disain yang modular, sehingga bisa ditambahkan fungsinya dengan program Perl. Memiliki dukungan ftpmail juga konfirmasi untuk berlangganan, untuk mencegah permintaan langganan palsu. Proses konfirmasi ini dilakukan berdasarkan header atau ekspresi beraturan pada body (regular expression)

## 4.2 Listar

Listar [<http://www.listar.org/>] ini memiliki fitur yang menarik seperti fitur anti-spam, dapat menghilangkan pesan MIME dan menghapus attachment serta dukungan virtual host. Listar tidak seperti majordomo, tidak membutuhkan preivilege (hak) khusus dalam operasinya. Majordomo atau manajer milis lainnya membutuhkan tingkat root atau daemon agar dapat beroperasi. Listar memiliki pustaka client SMPT yang dapat berkomunikasi langsung dengan mailserver lokal. Listar dijalankan dg suid ke user biasa, tetapi user ini tak perlu memiliki preivilege khusus. Suid dilakukan hanya untuk mengelola hak akses file (ownership). Pengelolaan jarak jauh secara aman, memungkinkan pengadministrasian Listar melalui email secara aman (mencegah spoofing pada alamat administrasi). Tidak perlu menggunakan password untuk diingat, cukup mengirim pesan "**admin**" dan akan menerima formulir pengelolaan administrasi secara aman, hanya tinggal diisi dan dikirim kembali.

Listar ini mendukung berbagai sistem operasi bebas seperti GNU/Linux dan FreeBSD, dapat dikembangkan secara mudah tanpa perlu menghentikan pengoperasian, dan memiliki opsi untuk dikompilasi dengan modus "*dynamic module*". Dengan cara ini memungkinkan fungsi baru dikembangkan, dikompilasi dan diletakkan di direktori "*modules*" tanpa perlu menghentikan Listar yang sedang bekerja. Listar menggunakan definisi *runtime* untuk semua operasinya misal: user flag, command, processing hook. sehingga memungkinkan suatu modul baru dipasang dan menghasilkan fungsi seperti yang diharapkan. Pelanggan dapat menentukan jenis langganan misal module Digest mendefinisikan flag user "DIGEST" sebagai kata yang dapat digunakan pengguna untuk menentukan bahwa dia akan menerima dalam bentuk ringkasan. Fitur lainnya dari listar misal kemampuan untuk menjejaki dan mengelola alamat pelanggan yang buruk. Kemampuan mengubah pesan MIME ke teks biasa, dan menghapus attachment MIME, serta kemungkinan dijalankan pada Virtual Host.

## 4.3 ezmlm

Ezmlm (**EZ Mailing List Manager**) [<http://www.ezmlm.org>] ini merupakan sistem mailing list yang mudah dikonfigurasi. Program mailing list ini dioperasikan menggunakan **qmail**. Memberikan banyak fasilitas, dan digunakan secara luas. Program ezmlm ini ditulis oleh Dan J. Bernstein menggunakan C dengan pertimbangan kecepatan dan keandalan. Kerjanya digabungkan dengan mekanisme pengiriman email dari qmail sehingga terkait dengan qmail. Mekanisme yang digunakan ezmlm sangat inovatif dan masih jarang digunakan oleh manajer milis lainnya. Sebagai contoh alamat email disimpan dalam hash sehingga proses pencarian akan lebih cepat. Ezmlm memiliki pengaliran pesan tambahan, pesan disimpan 100 pesan per direktori sehingga memungkinkan penyimpanan dan pembacaan lebih efisien. Berkas penting ditulis dengan nama baru dan hanya akan dipindahkan ke tempat baru bila telah disimpan secara aman.

Ada beberapa inovasi pada ezmlm ini, misal kemampuan mendeteksi alamat email yang mental, dan secara otomatis memberikan peringatan dan juga mengirim email pengujian. Bila pengujian ini mental juga maka alamat tersebut akan dihapus dari daftar langganan. Sehingga sistem tidak akan menghapus email yang mental sementara, dibutuhkan waktu sekitar 12 hari sejak email yang mental pertama sebelum dikirimkan peringatan, dan 12 hari lagi bila tidak ada jawaban maka akan dihapus.

Hal lainnya yang menarik dari ezmlm ini adalah penggunaan cookies dengan kriptografi pada timestamp, alamat dan aksi. Mekanisme ini digunakan untuk menjamin bahwa pengguna yang mengirimkan berlangganan/berhenti berlangganan bukanlah orang lain. Juga digunakan untuk mencegah pemalsuan pesan pengujian yang dikirimkan untuk mengetahui alamat yang mental.

## 4.4 Mailman

Mailman [<http://www.gnu.org/software/mailman/>] ini ditulis menggunakan Python, sehingga interpreter Python perlu terinstal. Tidak seperti manajer milis lainnya, program ini menyediakan tiap mailing list sebuah halaman web, yang memungkinkan pengguna untuk berlangganan, berhenti berlangganan, membaca arsip, menentukan model langganan (digets atau biasa), menghentikan sementara langganan, menyembunyikan alamat emailnya dari pelanggan lainnya dan lain sebagainya melalui web. Bahkan si administrator milis dapat melakukan administrasi melalui web ini. Seperti mengubah konfigurasi, mengelola pelanggan dan juga melakukan persetujuan posting.

Mailman juga mengintegrasikan beberapa fungsi yang sering dibutuhkan orang untuk sebuah mailing list, termasuk pengarsipan, gateway mail-ke-news, integrasi penanganan mail mental, pencegahan spam, perintah administrasi melalui email, dan penyaluran SMTP secara langsung (dengan fast bulk mailing), mendukung domain virtual dan berbagai fitur lainnya. Memiliki sistem pengarsipan berbasis web dan dapat digabungkan dengan pengarsip lainnya misal MHonArc. Memiliki fitur jawaban otomatis (*auto-reply*). Perintah mirip dengan major domo. Pemilik milis dan moderator dapat lebih dari satu.

Mailman dapat dijalankan pada sistem Unix dan kompatibel dengan web server dan browser serta SMTP server pada umumnya. Membutuhkan Python 1.5 ke atas. Beberapa program file dibuat dengan C demi pertimbangan keamanan

## 5 Layanan database

Server database merupakan salah satu aplikasi yang sangat penting bagi suatu server baik untuk keperluan Intranet ataupun Internet. Tentu saja dalam memilih database server ini pertimbangan kebutuhan harus menjadi patokan utama bukan sekedar kekayaan fitur dari database tersebut. Di samping tersedia beberapa database server proprietary dan komersial untuk GNU/Linux seperti :

- Informix [<http://www-3.ibm.com/software/data/informix/>]
- DB2 [<http://www.ibm.com/db2/linux>]
- Oracle [<http://www.oracle.com/linux/>]
- Adabas [<http://www.softwareag.com/adabas/>]

Tersedia beberapa database server freeware yang dijelaskan pada bagian berikut ini.

### 5.1 PostgreSQL

PostgreSQL [<http://www.postgress.org>] merupakan suatu Object-Relational DBMS (ODBMS) yang mendukung hampir semua perintah SQL, termasuk subselect, transaction type dan fungsi yang didefinisikan pengguna. Merupakan database open source yang paling maju. Awalnya merupakan proyek riset untuk mengembangkan suatu prototype DBMS generasi berikutnya yang dilakukan oleh Universitas California Berkeley dibawah bimbingan Prof. Michael Stonebreaker. Awalnya bernama Postgres, ketika fungsi SQL ditambahkan pada tahun 1995, namanya berganti menjadi Postgres95 dan pada akhir tahun 1996 namanya berubah menjadi PostgreSQL.

Penulis PostgreSQL pertama adalah Andrew Yu dan Jolly Chen. PostgreSQL mendukung model dan jenis data yang beragam. Kini dikembangkan oleh komunitas Internet dibawah koordinator Marc G. Fournier. Pendukung dan penyedia layanan komersial untuk database ini juga tersedia. Cukup banyak digunakan dalam aplikasi database transaksi sehari-hari. Bahkan penggunaan dengan GIS cukup banyak juga.

## 5.2 MySQL

MySQL [<http://www.mysql.com>] merupakan server database open source yang digunakan terbanyak di dunia. Arsitekturnya yang khas menyebabkan MySQL ini sangat cepat dalam operasinya dan mudah disesuaikan dengan kebutuhan. Penggunaan ulang kode dan pendekatan yang minimalis menghasilkan fitur yang kaya fungsi dan menghasilkan DBMS yang kecepatan, kekompakan, kestabilan dan kemudahan dalam penerapannya sangat sulit untuk ditandingi.

Pemisahan kode antara server utama dan penanganan tabel memungkinkan MySQL dijalankan pada suatu kendali transaksi yang ketat, atau dengan akses disk tanpa transaksi yang sangat cepat. Sistem penanganan tabel InnoDB kini memberikan fitur tambahan server MySQL standard, termasuk dukungan transaksi secara penuh, row-level locking dan persyaratan foreign-key. MySQL 4.0 akan mendukung lalu lintas yang aman antara client server, sehingga menaikkan keamanan terhadap serangan dan akses tak sah.

Aplikasi Web merupakan suatu penggunaan utama dari MySQL memang dapat menggunakan SSL untuk melindungi lalu lintas data antara browser dan aplikasi web. Penggunaan SSL ini biasanya ditulis dengan PHP, Perl atau ASP atau lainnya. Tetapi lalu lintas antara aplikasi skrip dan database server tidak dilindungi, perlindungan yang diberikan hanya karena antara kedua "server" ini pada mesin yang sama, atau pada mesin yang sama-sama berada di belakang firewall. Pada MySQL 4.0, daemon mysqld dapat langsung menggunakan SSL untuk memudahkan koneksi dari aplikasi end-user. Misal pada aplikasi client di luar firewall.

## 5.3 SapDB

SapDB [<http://www.sapdb.org>] ini merupakan suatu database server modern, berbasis SQL, telah banyak digunakan sebagai DBMS Industri. Bersifat platform-independent sehingga dapat digunakan pada beragam sistem. Sangat skalable, mudah digunakan dan dirawat serta mendukung transaksi secara penuh. SAPDB ini telah banyak digunakan bersama-sama aplikasi SAP pada solusi bisnis. SAPDB juga menawarkan kemampuan Object Oriented dan juga kemampuan yang mendukung pengelolaan data tak terstruktur.

Awalnya pada tahun 1977-1980 dikembangkan di Technical University Berlin sebagai suatu proyek riset dengan judul "*Distributed database on minicomputer*". Sebagai partner dari industri adalah Nixdorf Computer AG. Lalu diproduksi secara komersial oleh Nixdorf Computer AG dengan nama DDB/4 pada platform Unix. Pada tahun 2000 SapDB dirilis menjadi Open Source dengan GPL dengan tujuan menaikkan kompetisi di pasar database serta untuk mendapatkan umpan balik dari komunitas Open Source untuk menaikkan kemampuan SapDB. Saat ini SapDB tetap juga ditawarkan dukungan komersial oleh SAP AG.

## 5.4 Firebird

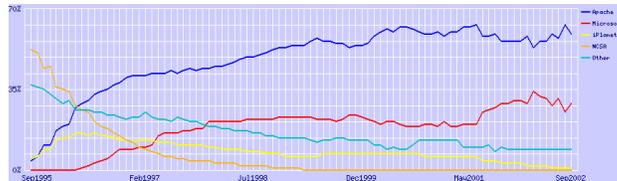
Firebird [<http://firebird.sourceforge.net/>] merupakan suatu RDBMS yang menawarkan fitur ANSI SQL-92 dan dapat dijalankan di GNU/Linux, MS Windows dan berbagai UNIX. Memberikan kemampuan konkurensi yang baik, kinerja tinggi serta dukungan bahasa yang mendukung store procedure dan trigger. Telah digunakan pada sistem nyata dengan berbagai nama sejak 1981. Firebird ini merupakan proyek bebas yang berbasis source code Interbase yang dirilis oleh Inprise Corp (kini dikenal dengan nama Borland). Tetapi bebas tidak terikat dengan kebijakan Borland.

## 6 Layanan Web

Web server merupakan salah satu tulang punggung penyedia layanan di Internet atau server di Internet. Tersedia banyak web server yang bersifat freeware. Antara lain

## 6.1 Web server serba guna

Apache [<http://www.apache.org>] merupakan web server yang terpopuler saat ini. Menurut survei netcraft lebih dari 50% situs di Internet menggunakan Apache sebagai Web server. Server Apache sangat luwes, memenuhi standar HTTP/1.1, mengimplementasikan protokol terbaru dan sangat luwes dikonfigurasi serta dapat ditambahkan modul lainnya melalui module Apache. Tersedia untuk berbagai sistem operasi.



Gambar 5: Penggunaan Apache hingga Oktober 2002

Apache menyediakan fasilitas yang kaya, yang sangat dibutuhkan suatu server serius, seperti autentikasi, pengaturan akses direktori, virtual host, kemampuan URL rewriting, dan juga alias. Kemampuan melakukan content negotiation membuat Apache mampu melayani beragam client secara otomatis, baik untuk berbagai browser yang memiliki kemampuan berbeda ataupun untuk divais akses yang berbeda. Fungsi log yang dimiliki oleh Apache dapat dikirim melalui proses piping, sehingga dapat dilakukan rotasi log, filter log, serta melakukan pemisahan log secara langsung.

Awalnya Apache dikembangkan berdasarkan keinginan untuk memperbaiki Web server yang saat itu populer (NCSA web server). Tetapi akhirnya mengalami perombakan dan penulisan ulang dan menjadi Web server yang berdiri sendiri dan berbeda dengan NCSA. Kini malah mengalahkan kepopuleran NCSA Web server. Pada tahun 1999 dibentuk Apache Software Foundation untuk mengurus perkembangan Apache ini. Apache telah membuktikan sebagai web server yang cepat, stabil dengan fitur yang paling kaya di antara web server lainnya. Saat ini proyek Apache telah berkembang dan tidak hanya sekedar Web server.

## 6.2 Web server cepat

Apache adalah suatu web server serba guna, tetapi pada kegunaan khusus terkadang kecepatan yang diberikan oleh Apache tidak mencukupi. Beberapa perangkat lunak freeware untuk web server dengan kecepatan tinggi tersedia. Sebagian besar web server tersebut hanya melayani halaman statis, sehingga ketika membutuhkan halaman dinamis, maka web server serba guna seperti Apache masih dibutuhkan. Web server cepat yang freeware tersebut antara lain :

- **Tux**

Tux [<http://people.redhat.com/mingo/TUX-patches/>], juga dikenal dengan nama komersial **Redhat Content Accelerator** [<http://www.redhat.com/docs/manuals/tux/>] merupakan suatu layer protokol HTTP dan object cache yang diintegrasikan ke kernel GNU/Linux. Merupakan suatu solusi ideal untuk menghasilkan solusi web server yang cepat dan terintegrasi dengan kernel. TUX singkata dari Threaded linUX http layer. Dibuat oleh Ingo Molnar (Red Hat). Merupakan suatu terobosan dalam pemanfaatan TCP dan server Web.

- **khttpd** `khttpd` [<http://www.fenrus.demon.nl/>] merupakan web server yang dijalankan pada lapisan kernel sebagai suatu modul (*device driver*). Hanya menangani halaman statis, dan melwatkan permohonan informasi non-statis kepada web server di lapisan user seperti Apache atau lainnya. Halaman statis, walau tidak kompleks tetapi sangat penting. Sebab hampir sebagian besar gambar (*image*) statis, begitu juga dengan sebagian besar halaman HTML. Suatu web server biasa memiliki overhead yang lebih besar ketika menangani halaman statis, dengan kernel Linux hal itu dapat dilakukan di tingkat kernel secara mudah.

Dengan cara ini maka proses pelayanan halaman statis menjadi cepat dan tidak membutuhkan sumber daya komputasi yang besar. Merupakan suatu solusi yang cocok buat situs web dengan beban tinggi dan memiliki halaman statis.

- **Boa**

Boa [<http://www.boa.org>] merupakan server HTTP single task. Berbeda dengan server web lainnya, boa tidak melakukan *fork* untuk setiap koneksi yang datang. Boa juga tidak melakukan *fork* untuk menangani koneksi jamak. Secara internal boa melakukan multipleks seluruh koneksi HTTP dan melakukan *fork* hanya untuk program CGI (yang harus merupakan proses terpisah), melakukan pembuatan direktori otomatis, dan melakukan gunzip berkas secara otomatis. Tujuan utama dari disain Boa adalah kecepatan dan keamanan. Keamanan dalam arti tidak bisa diserang oleh pengguna tak sah, bukan dalam arti pengaturan akses yang ditail ataupun komunikasi yang terenkripsi. Server boa ini tidak ditujukan untuk server lengkap. Dikembangkan pada tahun 1991 oleh Paul Phillips.

Di samping web server untuk menyediakan layanan web maka dapat ditambahkan beberapa program sebagai fungsi penambah kinerja.

## 7 Application Framework

Dalam mengembangkan aplikasi Web, suatu server web tidaklah cukup. Untuk itu dibutuhkan suatu lingkungan yang dapat menyediakan aplikasi web secara lebih mudah. Tersedia beberapa perangkat bantu freeware yang digunakan untuk kebutuhan ini dengan beberapa pendekatan berbeda.

### 7.1 Bahasa pemrograman skrip

Tersedia beragam bahasa pemrograman yang bersifat script yang dapat digunakan untuk mengembangkan aplikasi Web antara lain :

- **Bahasa skrip serba guna**

Dalam mengembangkan aplikasi Web bahasa skrip serba guna seperti Perl [<http://www.perl.org>], Python [<http://www.python.org>], Tcl/Tk [<http://www.scriptics.com/>] dapat digunakan untuk mengembangkan aplikasi pada Web. Bahasa seperti ini telah banyak digunakan dan masih tetap digunakan untuk mengembangkan aplikasi Web. Tcl/Tk sendiri memiliki fungsi seperti applet yang disebut "Tcl-let" hampir mirip dengan Java applet.

- **PHP**

PHP [<http://www.php.net/>] merupakan bahasa skrip serba guna yang kini banyak digunakan untuk mengembangkan aplikasi Web. Program PHP dapat disertakan dalam suatu dokumen HTML Sintaks yang dimilikinya diinspirasi dari C, Java dan Perl dengan beberapa fitur yang sangat khas. Tujuan dari bahasa ini untuk memungkinkan pengembang web menulis halaman secara dinamis dan ini dapat dilakukan dengan cepat dan mudah. PHP sendiri merupakan kepanjangan dari PHP : Hypertext Processor. Berbeda dengan Active Server Page (ASP) yang tidak memiliki bahasa khusus (ASP diprogram dengan Visual Basic Script atau JScript), PHP adalah bahasa tersendiri. Perbedaan lain adalah PHP tersedia di banyak platform baik MS Windows maupun Unix.

Faktor keunggulan lain adalah kecepatan PHP yang telah diakui oleh orang. PHP disertai banyak fitur atau pustaka yang memungkinkan pengembangan aplikasi secara cepat, misal melakukan proses mengakses ke situs lain, koneksi FTP dan lain sebagainya. Dibandingkan Cold Fusion, PHP dapat dikatakan lebih cepat, stabil dan tak membutuhkan sumber daya komputasi yang besar. Salah satu keuntungan PHP dibanding Perl adalah PHP benar-benar didisain untuk kebutuhan pengembangan Web sedangkan Perl adalah

bahasa skrip serba guna. Source code dari PHP juga lebih mudah dipahami dan juga memiliki fungsi yang hampir setara dengan Perl. Tetapi pada beberapa hal sebagai bahasa serba guna, PHP masih tertinggal dibanding Perl, terutama dalam pengolahan string.

- **Ruby**

Ruby [[www.ruby-lang.org](http://www.ruby-lang.org)] merupakan suatu bahasa skrip yang cepat dan mudah serta berorientasi obyek (OOP). Memiliki fitur untuk memproses teks dan melakukan pekerjaan manajemen sistem seperti Perl. Bahasa ini sederhana, langsung, ekstensibel serta portabel. Sintaksnya yang sederhana dipengaruhi oleh Eiffel dan Ada. Ruby memiliki fitur untuk menangani *exception* seperti Java, atau Python. Hal ini memudahkan penanganan ketika terjadinya kesalahan. Operator dapat didefinisikan ulang dengan mudah. Ruby adalah bahasa OOP yang lengkap, jadi semua data di Ruby adalah obyek tetapi bukan dengan pendekatan OOP seperti di Python atau Perl tetapi seperti di Smalltalk, tanpa kecuali semuanya adalah obyek. Saat ini Ruby mulai populer sebagai bahasa untuk mengembangkan aplikasi Web, walau masih kalah populer dengan PHP, Python, atau Perl.

## 7.2 Zope

Zope [<http://www.zope.org/>] adalah suatu server aplikasi (*application server*) Web, jadi bukan sekedar suatu server web yang menyajikan informasi pada situs web. Zope menyediakan fasilitas untuk memproses isi, serta mengelola isi melalui antarmuka web. Terdiri dari :

- **Content Management.**

Menyediakan suatu framework yang komprehensif untuk mengelola isi web melalui web. Dengan interface manajemen yang ekstensif, fasilitas kontrol akses yang lengkap, dan jenis dokumen yang beragam. Serta berbagai fitur yang memungkinkan perubahan isi web melalui web. Kemampuan ini memungkinkan pen delegasian pengelolaan situs tanpa pemrograman.

- **Programming**

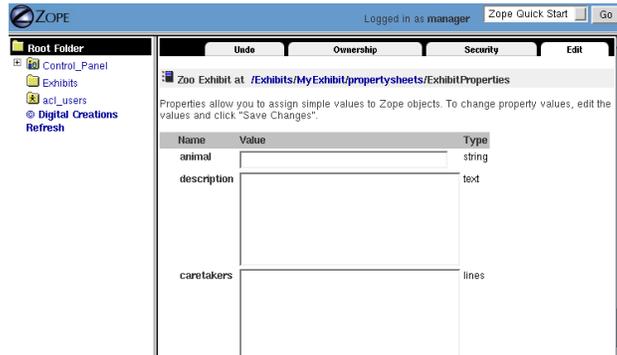
Suatu situs Zope menyediakan fasilitas melalui framework pengelolaan isi dan melalui antar muka aras bawah (low level interface) untuk mengimplementasikan program yang berfungsi sebagai bagian dari infrastruktur web. Dengan cara ini Zope, memungkinkan bagian dari situs web ke dalam program yang melakukan interoperasi dengan bagian lain, dengan isi lain pada situ tersebut, atau dengan sistem lain pada jaringan.

- **Kombinasi**

Pada penggunaannya, antarmuka Content Management akan berada di atas fasilitas pemrograman Zope. Fasilitas ini dan lainnya, tersedia sebagai komponen pemrograman. Sehingga banyak cara bagi pemrogram untuk dapat merubah dan menambah kemampuan Zope, banyak pemrograman dapat dilakukan melalui antar muka content management menggunakan form standard dari Zope, dikenal sebagai **Document Template Markup Language** (DTML). Secara umum, antar muka pengelolaan isi (content management) merupakan cara dalam memprogram Zope. Dengan cara ini maka pengelola content dapat memanfaatkan kemampuan pemrograman dengan tahapan yang terkelola dengan baik tergantung sejauh mana pemrograman yang akan dilakukan

## 7.3 Enhydra

Enhydra [<http://www.enhydra.org>] merupakan *application server* awalnya dikembangkan oleh Lutris Technology dan saat ini konsorsium ObjectWeb yang mensponsori perkembangannya secara Open Source. Istilah Application Server telah digunakan untuk mengacu kepada beragam perangkat lunak, sehingga istilah ini sering membingungkan. Cara termudah untuk mendefinisikannya adalah menganggap sebagai suatu aplikasi yang menyediakan layanan (service) kepada



Gambar 6: Zope

aplikasi lainnya. Sebagai contoh setiap aplikasi web perlu menerima permintaan HTTP, mengelola kondisi tiap session, dan melakukan koneksi database. Alih-alih menulis ulang layanan ini untuk tiap aplikasi, maka dapat dilakukan dengan menjalankan aplikasi tersebut di atas *application server* yang telah menyediakan layanan tersebut.

Enhydra dikembangkan berdasarkan pengalaman di lapangan untuk menyediakan suatu lingkungan untuk mendisain dan mengembangkan aplikasi berbasis Java dan XML berbasis servlet. Platform seperti ini adalah platform yang sering dibutuhkan orang untuk tingkat layanan enterprise, untuk situs web dengan beban besar. Sehingga dapat membantu proses pengembangan cepat (*rapid development*) dengan menggunakan aplikasi berbasis Java dan XML, termasuk e-Commerce, B2B, B2C, juga aplikasi Internet juga untuk web wireless.

Dibandingkan menggunakan skrip CGI biasa dalam menulis aplikasi web, maka dengan Enhydra dapat digunakan framework berorientasi obyek. Aplikasi akan dapat dengan mudah terskala. Koneksi database, dan informasi sesi akan dikelola di memori. Arsitektur 3-tier dari Enhydra memisahkan presentasi, bisnis dan logik data. Programmer dan disainer dapat bekerja secara terpisah. Suatu lingkungan pengembangan yang memisahkan antara disain interface, bussines logic dan data logic. Sehingga dapat digunakan untuk membuat aplikasi yang mudah dirawat dan penerapan yang luwes. Juga ditujukan untuk membentuk aplikasi yang mudah dirawat dan fleksibel dalam penerapannya. Penambahan lapisan persentasi baru, atau penerjemahan ke bahasa antar muka baru, dan juga divais tampilan berbeda dari satu aplikasi tidak perlu menyebabkan perubahan busines atau data logic. Enhydra benar-benar berbasis standard. Prinsip penggunaan kode kembali diterapkan oleh Enhydra.

Salah satu fitur Enhydra yang menarik adalah teknologi XMLC, suatu teknologi berbasis standard berorientasi obyek yang menggantikan JSP. Dengan menggunakan XMLC maka pemisahan kode dan content akan lebih baik lagi sehingga memisahkan kerja antara disainer dan pengembang. Enhydra tidak menggunakan lingkungan pengembangan khusus tetapi menyediakan suatu "plug-in" untuk lingkungan IDE seperti Borland JBuilder, Netbean dan JDeveloper. Enhydra juga menyediakan perangkat bantu pemetaan obyek ke relasi (object to relational mapping), tambahan untuk server web luster, pengelolaan session yang baik, dan antar muka web untuk pengelolaan.

Enhydra menawarkan aplikasi yang lebih skalabel dan pengelolaan yang lebih mudah ketimbang pengembangan aplikasi Web menggunakan Perl atau PHP. Jika anda ingin mengembangkan aplikasi untuk HTML dan WML, maka Enhydra menyediakan perangkat bantu yang dapat digunakan secara cepat. Proses pembelajaran Enhydra juga terbantu dengan beberapa aplikasi contoh. Ketersediaan Enhydra supersevlet mengurangi kompleksitas servlet. Wizard untuk membuat aplikasi baru membantu proses pembuatan proyek dengan cepat. Perangkat bantu DODS menyediakan bantuan pembuatan kode Java-SQL/RDBMS. Enhydra dapat dibandingkan dengan IBM WebSphere, BEA WebXPress walaupun ada beberapa hal yang berbeda. Perbedaan utama adalah Enhydra adalah Open Source.

## 7.4 Mozilla

Mozilla [<http://www.mozilla.org>] kini bukanlah sekedar browser, tetapi merupakan suatu platform yang dapat digunakan untuk mengembangkan aplikasi yang dapat diinstal secara lokal atau dijalankan secara remote melalui Internet. Memang agak ganjil penggunaan browser sebagai suatu application development framework. Tetapi bila kita perhatikan, saat ini orang sudah terbiasa mengakses layanan aplikasi pada Yahoo! Mail, Amazon dan lain sebagainya. Sehingga saat ini bagi pengguna, suatu browser memiliki kemungkinan digunakan sebagai suatu antar muka universal untuk mengakses berbagai aplikasi.

Salah satu keuntungan penggunaan Mozilla bagi developer adalah aplikasi menjadi cross-platform, yang berarti program yang sama dapat dijalankan di MS Windows seperti halnya di GNU/Linux. Juga memungkinkan suatu aplikasi dijalankan pada berbagai platform tanpa memandang sistem operasi yang digunakan. Hal itu dimungkinkan karena Mozilla bertindak sebagai lapisan penerjemah (*interpretation layer*) antara sistem operasi dan aplikasi. Selama Mozilla dijalankan pada komputer tersebut, maka sebagian besar aplikasi berbasis Mozilla akan dapat dijalankan, tanpa memperhatikan sistem operasi yang digunakan. Akan tetapi tidak semua aplikasi Mozilla selalu cross-platform karena memungkinkan mengembangkan suatu aplikasi dengan kode yang sangat spesifik untuk platform tertentu yang hanya dapat dijalankan untuk sistem operasi tertentu. Sebagai contoh Chimera [<http://chimera.mozdev.org/>], suatu browser untuk Mac OS X.

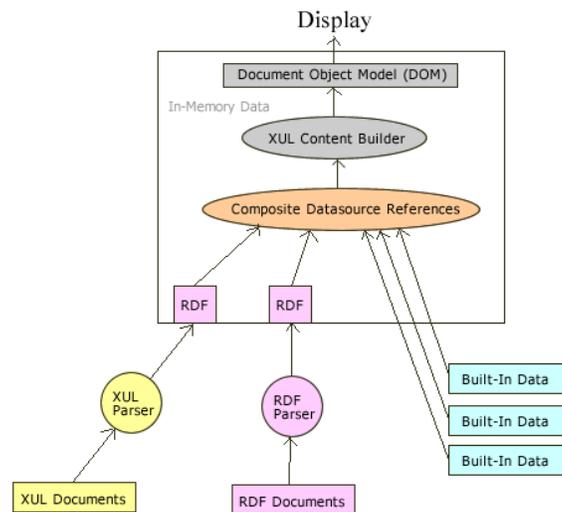
Mozilla dengan framework XPFE menyediakan fasilitas untuk mengembangkan aplikasi cross-platform. XPFE menggunakan beberapa standard web, seperti Cascading Style Sheet, JavaScript dan XML (ada komponen XML berupa bahasa baru yang disebut XUL). Dalam bentuk tersederhananya. XPFE dapat dikatakan sebagai gabungan dari tiap teknologi tersebut. Secara bersamaan teknologi tersebut membentuk framework XPFE. Pada beberapa hal, XPFE menyerupai Dynamic HTML (DHTML), yang merupakan kombinasi HTML, JavaScript dan CSS. DHTML ini memungkinkan pengembang membuat aplikasi di dalam area browser. XPFE merupakan evolusi dari idea ini yang memungkinkan pembuatan aplikasi yang di luar area browser.

Walau HTML telah banyak digunakan dan berkembang awalnya didisain sebagai suatu sistem yang sederhana untuk menghubungkan dokumen-dokumen di Internet. Penambahan kemudian pada HTML menambah fungsi, tetapi tetap belum membuatnya menjadi bahasa yang cocok untuk mengembangkan User Interface. XUL merupakan suatu bahasa yang khusus didisain untuk mengembangkan antar muka (*user interface*), sehingga XPFE lebih cocok digunakan untuk mengembangkan aplikasi ketimbang DHTML. Karena struktur XUL mirip dengan HTML, maka pengetahuan yang dimiliki dalam mengembangkan halaman web dapat dimanfaatkan untuk mengembangkan aplikasi Mozilla. XUL menggunakan tag yang mudah dipelajari.

Suatu aplikasi Mozilla memanfaatkan beberapa hal di bawah ini :

- XML-based User Interface Language (XUL)  
Dapat digunakan untuk membuat struktur dan isi suatu aplikasi. Sebagai contoh program browser Phoenix dan E-One
- Cascading Style Sheets (CSS)  
Digunakan untuk membuat bagaimana penampilan dan gaya tampilan suatu aplikasi
- JavaScript  
Digunakan untuk membuat fungsi suatu aplikasi. Dapat juga digunakan bahasa script lainnya seperti Python, Perl atau Ruby untuk menggantikan JavaScript ini.
- Cross-Platform Install (XPInstall)  
Digunakan untuk mempacketkan aplikasi sehingga dapat diinstal di platform apapun.
- eXtensible Binding Language (XBL)  
Digunakan untuk membuat widget (komponen GUI) yang dapat digunakan ulang dengan kombinasi XUL dan JavaScript.

- XPCOM/XPCConnect  
Digunakan untuk memungkinkan JavaScript atau bahasa skrip lainnya untuk mengakses dan memanfaatkan pustakan C dan C++
- XUL Templates  
Digunakan untuk membuat suatu framework untuk mengimport data ke dalam suatu aplikasi dengan memanfaatkan RDF dan XUL.
- Resource Description Framework (RDF)  
Digunakan untuk menyimpan data dan mengirimkan informasi. Secara umum ini adalah sisi paling kompleks dari XPFE.
- Document Type Definition (DTD)  
Digunakan untuk menangani bahasa dan karakter yang beragam lebih dikenal dengan istilah localization (L10N) dan internationalization (I18N).

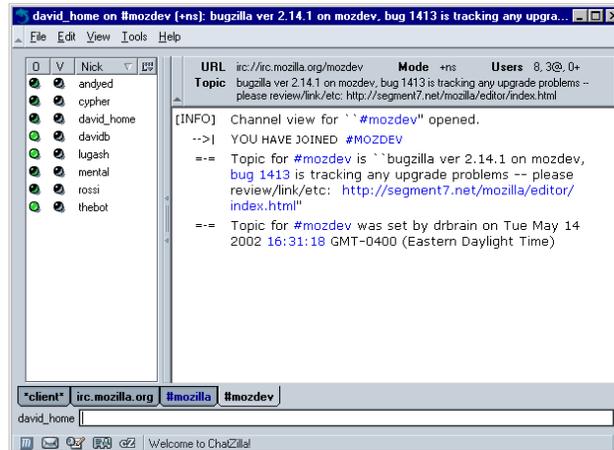


Gambar 7: Komponen platform Mozilla

Beberapa teknologi baru ini kini telah menjadi standard dan dimajukan dalam proses diakui sebagai standard oleh W3C. Teknologi-teknologi di atas tersebut berperan dalam mengembangkan aplikasi menggunakan Mozilla. Tetapi tidak semuanya perlu dipakai. Beberapa teknologi adalah utama untuk membuat aplikasi Mozilla, dan lainnya dapat digunakan untuk menambah kemampuan aplikasi tersebut. Sebagai contoh bisa saja mengembangkan aplikasi tanpa menggunakan DTD (walaupun menjadi sulit untuk digunakan dengan bahasa/karakter lainnya). Tanpa XUL sangat sulit mengembangkan aplikasi, karena tanpa XUL program yang dikembangkan tak memiliki antar muka.

Mozilla menangani aplikasi menggunakan XPFE sama halnya dengan menangani halaman web biasa. Memang tidak mengherankan karena aplikasi Mozilla secara konseptual tidak berbeda dengan halaman web. Keduanya menggunakan bahasa penandaan - *markup language* dengan stylesheet, javascript dan teknologi lainnya. Mekanisme rendering yang digunakan di browser Mozilla juga melakukan layout pada aplikasi Mozilla. Konsep ini tidak terasa, karena aplikasi Mozilla akan berada di window yang terpisah, alih-alih di browser. Gecko (*rendering engine*) dari Mozilla tidak membedakan antara merender aplikasi Mozilla atau halaman web. Sehingga programmer harus menentukan secara khusus pada Gecko bahwa harus menangani hal yang berbeda terhadap suatu aplikasi Mozilla.

Salah satu aplikasi Mozilla yang paling dikenal adalah browser Netscape yang telah menyatukan browser, editor HTML, instant messenger, dan chat. Aplikasi ChatZilla adalah suatu aplikasi IRC yang dibangun di atas platform Mozilla. Browser Phoenix [<http://mozilla.org/projects/phenix/>] contoh proyek yang dikembangkan dengan platform Mozilla [<http://www.mozdev.org/projects.html>]



Gambar 8: Contoh aplikasi Mozilla

## 7.5 SashXB

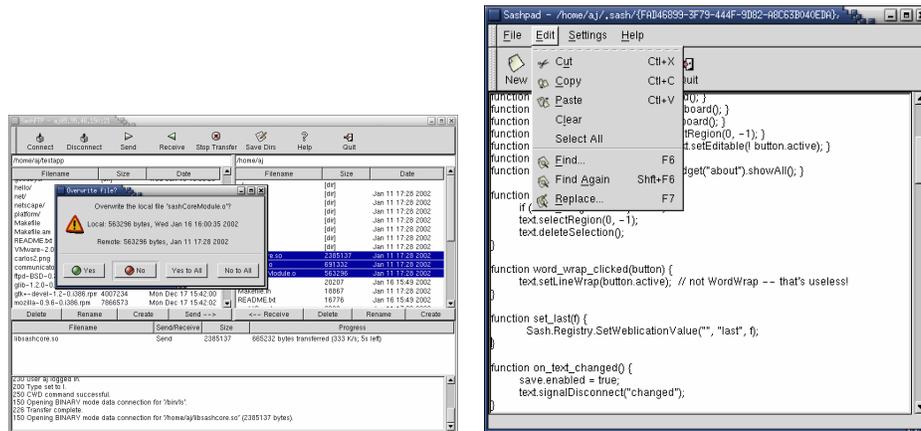
SashXB [<http://www.sashxb.org/>] ini merupakan lingkungan aplikasi Open Source yang memberikan fungsi dengan kemudahan seperti JavaScript. Sangat cocok bagi pengembang yang menguasai HTML dan JavaScript bila ingin membuat aplikasi biasa. Juga dapat digunakan oleh para pengembang yang berpengalaman yang membutuhkan solusi cepat untuk mengembangkan aplikasi (Rapid Application development).

Menggunakan berbagai proyek Open Source lainnya termasuk, Mozilla, GNOME, Glade, Gdome, dan OpenLDAP. Suatu program SashXB disebut **weblication** dan dapat diinstal atau dihapus di lokal atau dari web. SashXB juga menanggapi pemeriksaan keterkaitan (dependency) dengan suatu layanan pencarian lokasi untuk menginstal komponen yang dibutuhkan. Model sekuriti yang diterapkan dapat disesuaikan oleh pengguna. Kemampuan SashXB dapat ditambahkan dan sangat luwes. Komponen dapat ditambahkan dengan mudah dengan instalasi semudah weblication tersebut. Komponen ini akan mendukung kemampuan OS secara native.

Sebagai contoh weblication SashXB dapat menjalankan suatu window yang sederhana pada suatu User interface yang didisain dengan Glade pada suatu panel GNOME, atau bahkan dapat beroperasi sebagai lingkungan skrip pada aplikasi konsol. Pada perkembangan mendatang akan disertakan lkomponen seperti Screensarver, Nautilus, komponen BONOBO. Juga telah diberikan komponen untuk mengakses sistem berkas (*filesystem*), menjalankan berkas Vorbis (lagu), mem-parsing dan membentuk dokumen XML, berkomunikasi dengan mesin lain menggunakan protokol Jabber (*peer to peer*), menggunakan FTP, berinteraksi dengan User Interface menggunakan GTK dan Glade dan banyak hal lainnya. Beberapa contoh penggunaan seperti Newsbar (untuk ticker headline), text editor, FTP client, browser sederhana, game dan lains sebagainya. Semua aplikasi ini ditulis dengan kode tidak lebih dari 1 Kbyte dan sekitar 100 baris. Jadi sangat cepat dan efektif.

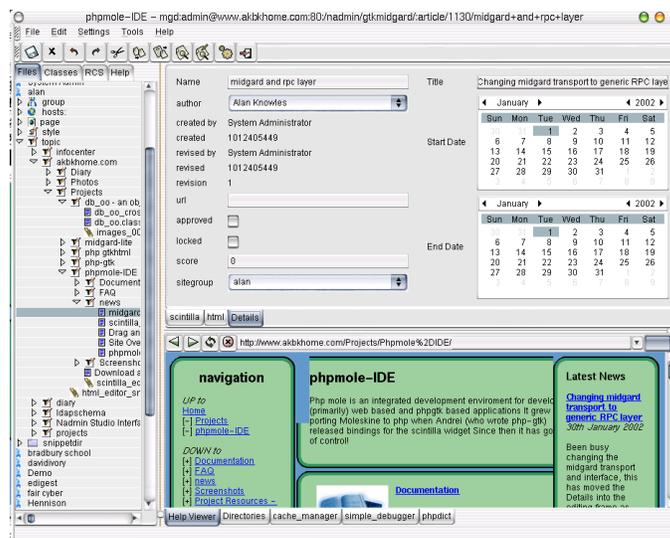
## 7.6 phpGTK

PHP-GTK [<http://gtk.php.net/>] ini merupakan ekstensi dari bahasa pemrograman PHP yang mengimplementasikan “binding” ke perangkat bantu GUI, GTK+. Sehingga menyediakan inter-



Gambar 9: Contoh aplikasi SashXB

face object oriented ke kelas dan fungsi GTK+. Sehingga memudahkan untuk menulis aplikasi client GUI yang cross platform.



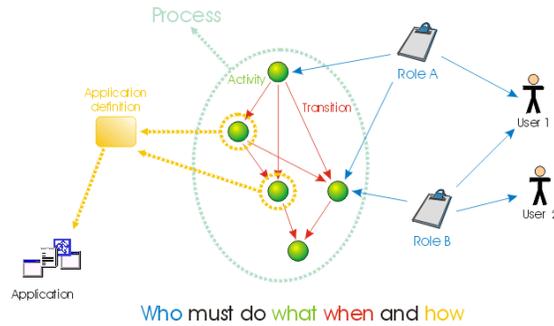
Gambar 10: Contoh aplikasi phpGTK

## 7.7 Document Flow

Workflow merupakan perangkat lunak yang berupa koleksi perangkat bantu yang menangani data sebagai suatu aliran kerja (workflow). Antar muka ke pengguna mirip dengan email. Ketika pengguna mengklik pada link atau tombol "Submit" perangkat bantu lainnya akan memindahkan dan memanipulasi data sebagai suatu aliran berdasarkan pada apa yang ingin dilakukan pengguna pada data tersebut. Misal pengguna dapat memberikan isian pada formulir HTML dan mengirimkannya. Lalu administrator/supervisor dapat menyetujui, mengedit atau menghapus informasi. Pengguna juga dapat mengupload dokumen jenis lainnya (misal Postscript) dan akan digabung dengan dokumen lainnya pada tahap akhir.

Ada beberapa solusi workflow yang tersedia bebas antara lain :

- OpenFlow



Gambar 11: Workflow (dari situs OpenFlow)

Open Flow [[http://www.openflow.it/EN/index\\_html](http://www.openflow.it/EN/index_html)] ini dikembangkan oleh pemerintah Italia, untuk memenuhi kebutuhan sistem document flownya.

- **WorkFlow Tool**

Workflow Tool [<http://www.itlab.musc.edu/workflow/>] ini berbasis web dan berbasis form HTML. Menggunakan SQL dump. Workflow dihasilkan berbasis XML map. Otentikasi sederhana yang digunakan untuk mengakses dan mengendalikan aliran data. Data workflow dapat diekspor ke lembar kerja Excel. Data dan dokumen dapat diberikan ke pengelola workflow menggunakan XML. Dengan menggunakan XML ini maka sistem dapat dikembangkan lebih mudah.

## 8 Dukungan QoS

Pada dasarnya QoS ini digunakan sebagai ukuran formal untuk menentukan seberapa baiknya suatu service. Yaitu :

1. **Ketersediaan service (*Service availability*).**

Yaitu keandalan koneksi yang disediakan untuk user. Diharapkan ketersediaan ini selalu mendekati 100%. Ketersediaan ini berkaitan erat dengan *downtime* dari penyedia jasa tersebut. Untuk service yang kritis biasanya diinginkan nilai ini di atas 99,9999 % (biasa disebut dengan *six nine* ).

2. ***Throughput*.**

Kecepatan efektif transfer data di jaringan, yang biasanya diukur dengan bit per detik. Ini tidak sama dengan kapasitas maksimum dari jaringan, atau kecepatan link. Penggunaan bersama jaringan akan merendahkan nilai throughput efektif ini. Biasanya nilai minimal troughput yang dijamin oleh penyedia jasa.

3. ***Packet loss*.**

Peralatan seperti router, switch kadang-kadang memiliki kegagalan penanganan paket. Misal disebabkan karena link macet, dan buffer terpaksa menampung paket yang terlalu banyak. Seharusnya paket yang hilang harus ditransmisi ulang. Pada sistem yang baik biasanya dikehendaki nilai ini kurang dari 1 persen rata-rata per bulan.

4. ***Latency* atau tundaan (*delay*).**

Waktu yang dibutuhkan paket untuk mencapai tujuan dari asal. Latensi ini sekitar 25 ms untuk koneksi sekitar 5000 km (suara), kecuali bila ada koneksi satelit. Biasanya untuk koneksi Internet publik, suara mengalami latency sekitar 150 ms. Hal ini disebabkan waktu yang dibutuhkan untuk konversi dan queueing.

## 5. *Jitter* .

Variasi dari waktu tundaan yang dialami suatu paket yang sama ketika melewati route yang sama pada jaringan. Dengan kata lain variasi dari latency, hal ini disebabkan banyak hal, misal antrian queue, variasi dari waktu proses dari tiap paket, perjalanan melalui path yang berbeda.

Parameter nomor 1 dan 2 akan selalu dicoba untuk dimaksimalkan sedangkan nomor 3,4, 5 akan diminimalkan. Seperti diterangkan pada tabel di bawah ini, setiap jenis layanan pada Internet memiliki kebutuhan dengan parameter kualitas layanan yang berbeda. Berikut ini dipaparkan jenis layanan tersebut serta kebutuhannya terhadap parameter QoS tersebut :

| Jenis trafik     | Sensitivitas  |          |          |        |
|------------------|---------------|----------|----------|--------|
|                  | Loss          | Delay    | Jitter   |        |
| Bandwidth        |               |          |          |        |
| Suara            | Sangat rendah | Menengah | Tinggi   | Tinggi |
| E-Commerce       | Rendah        | Tinggi   | Tinggi   | Rendah |
| Transaksi        | Rendah        | Tinggi   | Tinggi   | Rendah |
| E-mail           | Rendah        | Tinggi   | Rendah   | Rendah |
| Telnet           | Rendah        | Tinggi   | Menengah | Rendah |
| Browsing biasa   | Rendah        | Menengah | Menengah | Rendah |
| Browsing serius  | Menengah      | Tinggi   | Tinggi   | Rendah |
| Transfer file    | Tinggi        | Menengah | Rendah   | Rendah |
| Konferensi video | Tinggi        | Menengah | Tinggi   | Tinggi |
| Multicasting     | Tinggi        | Tinggi   | Tinggi   | Tinggi |

Tabel 1: Prasyarat tiap layanan

Dari perbedaan kebutuhan akan tampak bahwa kebutuhan pemenuhan QoS akan membutuhkan suatu mekanisme untuk memprioritaskan paket merupakan salah satu kebutuhan utama dari penyediaan jasa yang beragam pada Internet. Kebutuhan penentuan prioritas ini pada dasarnya meliputi dua persyaratan utama yaitu :

- Metoda untuk menandai aliran paket bergantung prioritasnya
- Mekanisme jaringan yang mengenali penandaan tersebut dan melakukan penanganan berdasarkan prioritas tersebut.

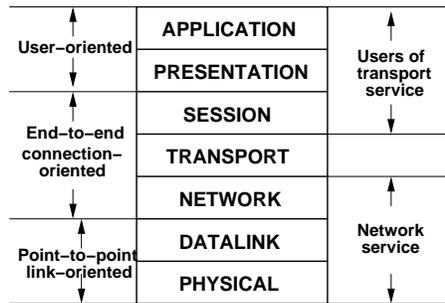
Masalah QoS timbul karena :

- Pengirim paket (end user) menginginkan mengirim paket sebarang waktu, dengan beban tinggi, dan *burstiness* (lonjakan data) tinggi
- Penerima (end user) mengharapkan waktu tundaan rendah, dan throughput yang tinggi
- Penyedia jasa (provider) menginginkan meminimalkan infrastruktur, sehingga kapasitas diinginkan serendah mungkin

Jika salah satu pihak bersedia menyerah, maka tak ada masalah. Sayangnya di dalam dunia nyata hal tersebut tidak bisa terjadi.

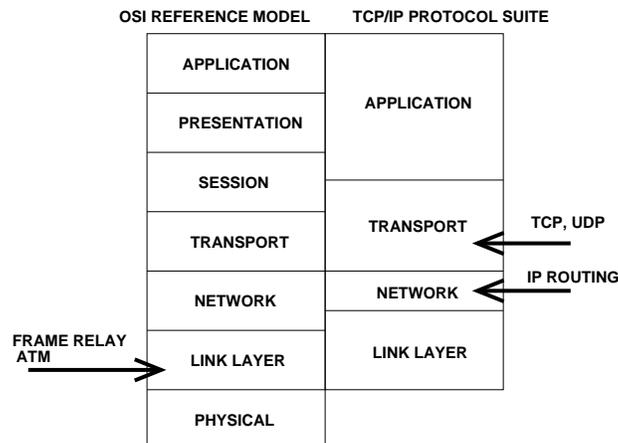
## 8.1 Permasalahan teknis TCP/IP

Dalam menjabarkan permasalahan jaringan, sering digunakan istilah *layer*. Layer ini adalah suatu pembagian logis terhadap fungsi-fungsi dari layanan yang ada. Pengelompokan fungsi ini untuk memudahkan standardisasi dan pembahasan. Pada Gambar 12 dijabarkan suatu lapisan OSI yang menjadi suatu standard internasional. Pembahasan jaringan biasanya dilakukan dengan mengacu pada lapisan ini.



Gambar 12: Lapisan OSI dan TCP/IP

Standard TCP/IP memang disusun lebih sederhana dibandingkan standard OSI. Hal ini ditunjukkan pada Gambar 13. Sering pula TCP/IP ini digunakan di "atas" sistem lainnya seperti di atas Frame Relay ataupun ATM. Hal ini disebabkan popularitas dan keluwesan yang dimiliki oleh TCP/IP dibanding protokol lainnya.

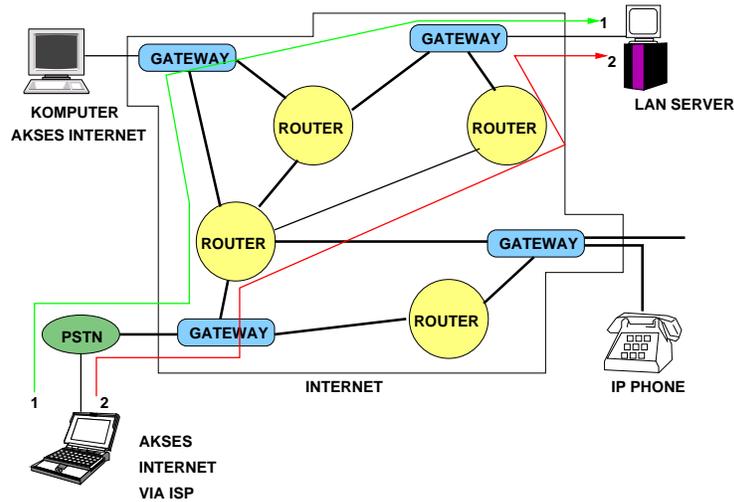


Gambar 13: Layer TCP/IP terhadap lainnya

Berbeda dengan protocol Asynchronous Transfer Mode (ATM) yang telah mempehitungkan hal ini, TCP/IP yang menjadi tulang punggung Internet belum memiliki mekanisme tersebut. IP merupakan protokol yang berdasarkan *best effort*, artinya protokol ini tak menjamin sampainya data. Jaminan ini diberikan oleh protocol TCP yang berada di atas IP. Jika sebuah paket TCP tak diterima, maka akan selalu dicoba untuk dikirim ulang. Hal ini memang efektif dan handal (sesuai dengan tujuan TCP/IP) tetapi menyebabkan TCP menjadi lambat. Sehingga pada umumnya TCP digunakan pada aplikasi yang tak sensitif dengan waktu.

Aplikasi real time tak cocok digunakan dengan TCP, waktu yang dibutuhkan untuk mengetahui paket yang hilang, dan mengirimkan ulang terlalu lama. Sehingga digunakan User Datagram Protocol (UDP). UDP ini berjalan lebih cepat dari TCP tetapi memiliki keterbatasan dibanding TCP. TCP/IP memang memberikan kemampuan pengiriman paket yang efisien, sayangnya untuk mengirim paket multimedia atau seperti voice ada beberapa kendala yang disebabkan beberapa karakteristik dari jaringan TCP/IP yaitu :

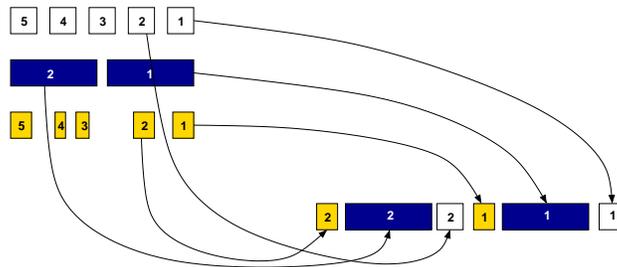
- Paket TCP/IP dapat melalui rute yang berbeda untuk sumber-tujuan sama . Seperti pada Gambar 14 disajikan contoh permasalahan ini, yaitu pada contoh paket 1 dan 2. Walaupun tujuannya sama tetapi melalui rute yang berbeda.
- TCP/IP tidak memiliki mekanisme penjamin saluran seperti halnya ATM ataupun PSTN. Sehingga berapa besar bandwidth suatu saluran tidak dapat diketahui dimuka. Bisa saja



Gambar 14: Permasalahan routing TCP/IP

tiba-tiba terjadi penyempitan saluran (*throughput*)

- Delay paket TCP/IP tak dapat dijamin.
- TCP/IP dirancang untuk menjamin reliabilitas sampainya paket ketimbang kesamaan rute.



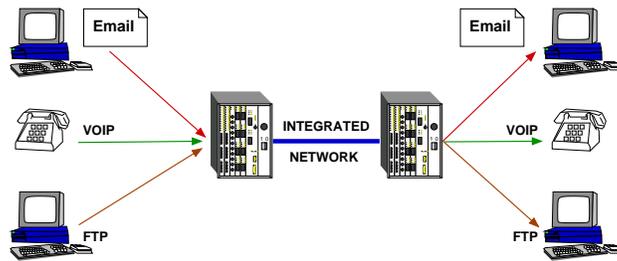
Gambar 15: Problem urutan tiba TCP/IP

Jaminan waktu dan tak ada paket yang hilang ini menjadi salah satu permasalahan dalam penerapan multimedia termasuk VoIP di Internet. Paket data voice atau multimedia adalah paket yang sensitif terhadap urutan paket-paketnya. Dengan kata lain paket tersebut memiliki *flow* yang perlu dipertimbangkan. Berbeda dengan paket seperti email ataupun halaman Web yang tak perlu memperhatikan flow ini. Flow dalam hal ini adalah deretan packet, yg urutan dan rentang waktunya harus dijaga (ada batasan waktu). Pada TCP/IP ketika ada "flow" dari application, maka oleh TCP/IP dianggap hanya sebagai "paket-paket" terpisah yg satu sama lainnya tidak berhubungan (dari *time constraint*). Hal ini jelas menimbulkan permasalahan pada implementasi VoIP pada jaringan TCP/IP yaitu :

- TCP/IP adalah *First Come First Server (Best Effort Server)*. Pada pendekatan ini aplikasi mengirim data kanan saja dia suka. Element jaringan akan berusaha menyampaikan data ini sebaik mungkin. Tetapi akan segera berhenti jika suatu limit tak terpenuhi dan tanpa memberi tahu pada pengirim dan penerima. Paket yang datang lebih dulu akan dikirimkan lebih dulu.
- Paket-paket TCP/IP tidak terjamin urutan tibanya. Sehingga bisa saja suatu segment suara yang belakangan akan tiba lebih dahulu. Pada Gambar 15 disajikan bagaimana seandainya

suatu paket-paket TCP/IP melewati suatu gateway. Maka urutan paket antar aplikasi tidak dijamin tibanya.

- Tiap aplikasi memiliki kebutuhan yang berbeda, ada yang mementingkan urutan (misal voice, video), ada yang tidak
- Paket multimedia (voice dan video) membutuhkan bandwidth yang besar. Tanpa adanya penanganan tingkat layanan maka sulit sekali tidak terjadi kerugian di pihak konsumen lainnya.



Gambar 16: Masalah pada integrasi layanan

Permasalahan lainnya adalah dengan diintegrasikan layanan. Saat ini semua layanan baik multimedia, voice, email, web disatukan pada satu jalur TCP/IP seperti yang tampak pada Gambar 16. Kondisi ini sebetulnya mensyaratkan bahwa tiap jenis layanan harus ditangani secara berbeda, karena memiliki kebutuhan yang berbeda.

## 8.2 Dukungan TCP/IP untuk QoS

Awalnya TCP/IP memang tidak didisain dengan konsep layanan yang berbeda tingkatatnya. Hingga akhirnya perlu dilakukan penyesuaian terhadap layer TCP/IP yang digunakan. Dan ini dikenal dengan dukungan arsitektur QoS untuk TCP/IP. Karena layer yg dimaksud ini adalah bersifat "logis" maka yg dilakukan adalah perubahan "atau penambahan" stack TCP/IP. Dengan kata lain perubahan perangkat lunak. Bukan perubahan koneksi fisik.

|                    |  |
|--------------------|--|
| <b>Application</b> |  |
| <b>Transport</b>   | <b>Integrated Service/RSVP, Differentiated Service</b> |
| <b>Network</b>     | <b>Constraint Based Routing</b>                        |
| <b>Link</b>        | <b>MPLS</b>  |

Gambar 17: Perubahan TCP/IP untuk QoS

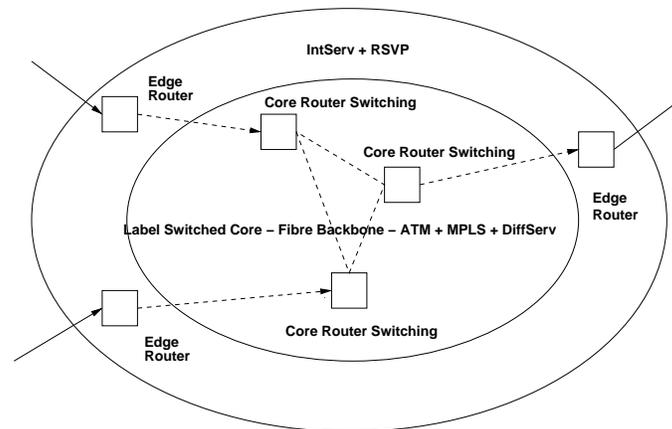
Pada Gambar 17 diberikan perubahan atau penambahan yang harus dilakukan pada suatu sistem TCP/IP agar mendukung QoS dengan baik. Pada Gambar 18 disajikan bagaimana implementasi dari arsitektur pendukung QoS ini pada jaringan TCP/IP.

- **Differentiated Service** [<http://diffserv.sourceforge.net/>]  
 Pada metoda ini suatu layanan telah dibagi-bagi kelasnya. Dan setiap layanan akan di"lewatkan" atau diantrikan secara berbeda. Dukungan GNU/Linux untuk DiffServ ini merupakan bagian dari arsitektur Traffic Control pada kernel GNU/Linux 2.2, dan 2.4. Pada model DiffServ ini layanan dibagi atas :

- Premium Service : low delay and low jitter

- Assured Services: better reliability
  - Olympic Services : dibagi menjadi Golden, Silver, Bronze
- **Integrated Service .**  
Metoda ini menggunakan suatu standard IETF yang menerapkan pembagian klas untuk trafik dengan tujuan menjamin QoS oleh jaringan. Dengan cara ini tiap aplikasi harus mengetahui sebelumnya dan mengalokasikan sumber daya sesuai dengan kebutuhan trafik tersebut. Sebagian dari standard ini yang dikenal dengan istilah *Admission Control*
    - Guaranteed Load Services : fixed delay bound
    - Controlled Load Services : reliable and enhanced best services
  - **MPLS (Multi Protocol Label Switching)** [<http://sourceforge.net/projects/mpls-linux>]  
Agar proses routing dapat menjaga flow, dan proses alokasi bandwidth. GNU/Linux juga telah mengimplementasikannya.
  - **Bandwidth control**  
Agar pengguna yang satu tidak memakan habis fasilitas bagi pengguna lainnya. Untuk link yang menyatukan berbagai layanan hal ini harus diterapkan, agar tidak ada pengguna yang dirugikan. Yang paling sering digunakan di lingkungan GNU/Linux adalah **CBQ** [<http://www.icir.org/floyd/cbq.html>]

Pada Gambar ?? dijabarkan berbagai jenis QoS. Untuk itu perlu adanya kesepakatan pada SLA mengenai parameter yang menjadi ukuran layanan.



Gambar 18: Arsitektur QoS untuk TCP/IP

## 9 Layanan keamanan

Pada dasarnya GNU/Linux merupakan sistem "multi user", sehingga tiap user memiliki perizinan yang berbeda (previdlege). User memiliki "file atau device" sendiri-sendiri. Sehingga setiap program (atau proses lebih tepatnya) dijalankan pada ruang memori dengan kepemilikan yang tertentu. Berkas (file) juga ditentukan kepemilikannya. Dengan cara ini suatu proses yang tak memiliki "hak" mencukupi maka tak bisa mengubah suatu file, atau mengakses suatu divais.

Dengan model ini, maka misal komputer di rumah digunakan oleh orang tua dan anak, maka orang tua tak perlu khawatir si anak akan menghapus berkas-berkasnya. Atau si anak menghapus berkas sistem yang memaksa harus menginstall ulang sistem. Sehingga masalah virus menjadi sulit timbul di GNU/Linux (walau bukan berarti tak ada), karena suatu program milik pengguna

berjalan dengan "previdge" (perizinan) yang terbatas dan sesuai dengan yang telah didefinisikan. Jadi tidak bisa suatu program yang dijalankan user A, menghapus suatu file, atau menginfeksi file sistem.

Di samping itu keragaman GNU/Linux (karena tiap orang bisa mengkonfigurasi GNU/Linux dan memberikan patch ke bagian manapun dari sistem GNU/Linux maka GNU/Linux menjadi sistem yg tidak monokultur, sehingga bersifat "multikultur". Seperti halnya pertanian monokultur yg lebih tahan terhadap virus (hama) maka hal ini terjadi juga di Linux, karena WORM dan VIRUS menjadi sulit tersebar secara massal di lingkungan Linux.

Pada "kernel Linux" telah diberikan fitur-fitur sekuriti dasar misal :

- multi user dan cabability (untuk membedakan antar user), compartment yang "memenjarakan" suatu aplikasi sehingga berjalan di lingkungan yang aman.
- tcpflooding protection, dan proteksi level kernel untuk TCP/IP
- file system cryptography
- divais `/dev/random` dan `/dev/urandom`. Divais ini ditangani kernel Linux dan menyediakan data random yang sangat dibutuhkan untuk aplikasi seperti kriptografi, "bibil" untuk TCP Sequence number (bila TCP sequence number tidak benar-benar random maka koneksi TCP/IP mudah dihijack (dibajak, sehingga kita diblokkan ke server palsu).
- infrastruktur "netfilter" yang merupakan sistem pemfilteran paket yang mengimplementasikan ipchain dan ipwfadm, sehingga infrastruktur ini menyediakan kemampuan mengubah paket ketika melewati bagian-bagian kernel. Dengan infrastruktur ini maka dapat diberikan fungsi "masquerading" (ip lokal tidak tampak, dan yang berhubungan dengan Internet hanya 1 IP), juga kemampuan "statefull inspection" dapat ditambahkan

Model sekuriti atupun fungsi tambahan yang ada di suatu sistem "Linux" berbeda dengan sistem MS Windows. Di sistem Linux pengguna memiliki kebebasan untuk menentukan sejauh mana keamanan yang ingin diimplementasikannya. Pada Linux pengguna/administrator dapat memilih pada tingkat manakah fitur sekuriti ditambahkan misal :

- Pada tingkat kernel
- Pada tingkat antara kernel - aplikasi
- Pada tingkat aplikasi

Misal pada tingkat kernel dapat diterapkan beberapa patch yang memberikan penambahan fungsi security :

- **Openwall** patch [<http://www.openwall.com>] yang melindungi dari serangan stack, buffer overflow, /tmp dan beberapa serangan.
- **LIDS** (Linux Intrusion Detection) [<http://www.lids.org>] dengan cara ini pemanfaatan capability bit lebih tinggi lagi, sehingga bisa diatur model kepemilikan file, lebih ditail (misal walau root ketika sistem beroperasi tetap tak bisa menghapus log file, atau meload module, dlsb)
- **ACL** patch [<http://acl.bestbits.at/>] digunakan untuk memenuhi prasyarat C2 yg menggunakan MAC. Dengan patch ini maka Linux dapat memberikan bit tambahan untuk kontrol akses. Dikenal dengan penambahan fungsi MAC (Mandatory Access Control)
- **RSBAC** (Rule Set Base Access Control) [<http://www.rsbac.org/>] untuk menambah kontrol akses pada sistem linux. Dengan cara ini seorang pengguna dapat diatur akses-nya sesuai dengan "role" pada organisasi (pada model biasa hanya group saja)

- **Medusa** [<http://medusa.fornax.sk/>] yang menerapkan model virtual space untuk mengakses object dengan menggunakan matrix access. Medusa ini memungkinkan sistem memiliki kebijakan akses yang lebih luwes (bisa menjadi RSBAC, MAC, atau lainnya)
- **SELINUX** (Flask model) [<http://www.nsa.gov/selinux/>] merupakan suatu sistem Linux yg dikembangkan oleh NSA sehingga model akses, model akses sistem call menjadi lebih aman dan tercatat.
- **Grsecurity** [<http://www.grsecurity.net/>], merupakan sistem ACL yang dapat membatasi akses ke berkas, kapabilitas, sumber daya komputasi dan atau socket ke semua pengguna termasuk **root**. Fitur lainnya adalah melindungi serangan yang mencoba mendapatkan root, sehingga mendapatkan root bukan berarti mendapatkan akses penuh ke sistem. Akses dapat diberikan pada suatu proses yang membutuhkan. Hal ini mempersulit diserangnya suatu sistem

Masih banyak lagi patch-patch yg tersedia yang dapat membuat sistem lebih aman (sesuai dengan yang diinginkan Beberapa persh seperti HP telah mengembangkan Trusted Linux suatu sistem Linux dengan patch security, dan akses kontrol yang lebih ketat.

Di level atas Kernel, dapat juga dilakukan security tambahan (misal di level library, atau sebelum aplikasi).

- **PAM** (Pluggable Authentication Module) sebetulnya tidak tepat disebut level library, tetapi bisa dikatakan levelnya antar kernel dan program aplikasi. Beberapa library juga memanfaatkan PAM untuk melakukan proses otentifikasi.
- **compartment** (chroot dan lain sebagainya), ini levelnya antara aplikasi yang dijalankan dan kernel. Sehingga suatu proses dijalankan dalam "*penjara*" nya.

Sedangkan di level aplikasi (distro Linux) biasanya suatu distro telah menyertakan beberapa aplikasi :

- **iptables** (untuk memfungsikan sebagai firewall, masquerading)
- **scanlog** (untuk mencatat ketika ada yang melakukan scan ke port)
- **tcpwrapper** (service TCP/IP tidak langsung berbicara langsung dg client)
- **snort** (intrusion detection)
- **aide** (advanced intruder detection)
- **tripwire** (untuk mengecek apakah binary ada yg diganti, integrity testing)
- **argus** (perangkat monitor jaringan)
- **GPG** (Gnu Privacy Guard)
- **ippl** (IP protocol logger) untuk melakukan log tiap protocol
- **nessus** (penetration test)
- **nmap** (untuk menguji sistem kita, apakah ada port terbuka)
- **amavis**, anti virus
- **honeyd**, yang dapat memberikan ilusi bahwa suatu server tampak sebagai sistem operasi lain.

Dan masih banyak lagi aplikasi security yang dapat diinstal (misal **chkrootkit**, untuk menguji apakah sistem kita dimasuki oleh intruder, yang telah menginstal rootkit), atau **logsurfer** yg memudahkan kita menganalisis log file, atau tool forensik yg tersedia di Linux (misal dari Wietse)

Sekuriti sering dipandang hanyalah merupakan masalah teknis yang melibatkan bisa atau tidak tertembusnya suatu sistem. Pada pandangan makro sekuriti sendiri memiliki konsep yang lebih luas, juga berkaitan dengan ketergantungan suatu institusi terhadap institusi lainnya, atau suatu negara terhadap negara lainnya. Beberapa aspek sekuriti yang harus dipertimbangkan diantaranya adalah secrecy, integrity, authentication, non repudiation, dan accountability. Untuk mengaplikasikan sekuriti dalam sebuah sistem informasi diperlukan juga pertimbangan lainnya misal suatu kebijakan sekuriti yang telah tertata dengan baik, teknologi yang memungkinkan diterapkannya kebijakan tersebut, serta kesepakatan sosial. Pilihan model sekuriti dan model organisasi serta kebijakan terhadap pengolahan data memiliki hubungan mutual.

Sulit menyebut sistem keamanan terbaik, karena :

- Keamanan itu adalah suatu proses bukan produk. Jadi bisa saja siste sekarang aman, tapi tidak dirawat, maka bisa saja besok menjadi tidak aman.
- Sekuriti itu seperti rantai. Keamanan maksimum dari suatu sistem adalah tingkat keamanan baigan terlemah dari sistem itu. Bagian terlemah ini termasuk desktop, GUI, dan pengguna.
- Keamanan itu selalu berkait dengan resiko, biaya kehilangan, dan biaya yang kita ingin keluarkan untuk mencegah kehilangan.

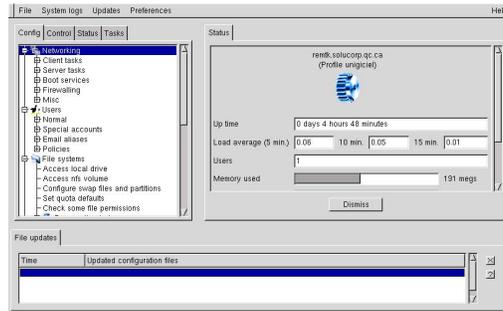
## 10 Administrasi sistem

Sistem GNU/Linux dapat dikelola dengan menggunakan perintah pada CLI (Command Line Interface) ataupun menggunakan beberapa perangkat lunak komersial seperti XPloy [<http://www.trustix.com>]. Memang sistem administrator yang berpengalaman biasanya telah membuat perangkat bantu sendiri dengan bantuan bahasa pemrograman skrip. Tetapi bagi para sistem administrator yang ingin menggunakan perangkat bantu dengan antar muka yang lebih muda telah tersedia pula beberapa aplikasi yang bebas, antara lain :

- **Linuxconf**  
Linuxconf [<http://www.solucorp.qc.ca/linuxconf/>] merupakan program sistem administrasi Linux. Tidak hanya itu saja, program ini merupakan suatu framework baru yang memudahkan penulisan modul baru untuk pekerjaan sistem administrator. Linuxconf memiliki antar muka yang beragam yaitu, berbasis teks, GUI dan web.
- **Webmin**  
Webmin <http://www.webmin.com/> merupakan sistem administrasi berbasis Web. Sehingga memudahkan dalam melakukan administrasi jarak jauh. Yang dibutuhkan hanyalah browser yang mendukung table dan form, dan Java untuk memanfaatkan modul File Manager. Dengan cara ini maka dapat dilakukan administrasi jarak jauh dari mesin apapun misal untuk menambah pengguna, mengatur Apache, DNS, dan lain sebagainya. Webmin ini terdiri dari suatu server web sederhana, dan beberapa program CGI yang akan mengubah berkas konfigurasi seperti `/etc/inetd.conf` ataupun `/etc/passwd`. Server web dan semua program CGI ditulis menggunakan Perl versi 5 dan hanya menggunakan modul Perl yang standard.

Beberapa distro GNU/Linux telah menyertakan aplikasi sistem administrasinya sendiri, misal YaST pada distribusi SuSE. Mandrake dengan MandrakeConf. Bahkan beberapa aplikasi KDE ataupun GNOME juga dapat digunakan untuk melakukan pekerjaan sistem administrasi. Memang di dunia freeware tersedia pilihan yang banyak.

Untuk melakukan pengelolaan sistem jarak jauh, dapat digunakan aplikasi :



Gambar 19: Linuxconf

- **telnet, rlogin dan ssh**

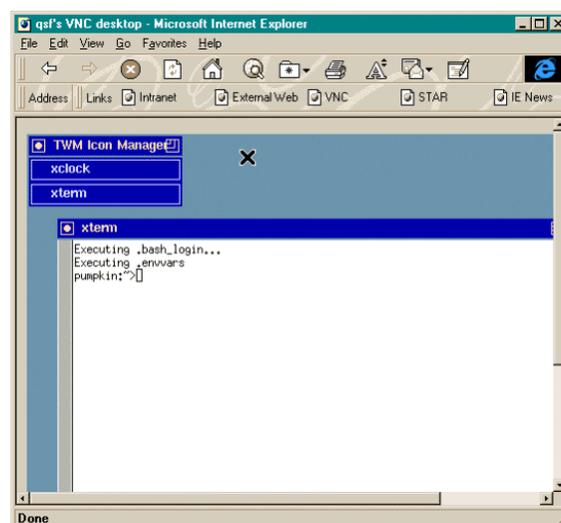
Telnet dan rlogin adalah aplikasi yang banyak digunakan untuk login ke mesin yang jauh dan dapat melakukan pekerjaan administrasi. Perangkat bantu lainnya yang juga sering digunakan adalah **screen**, **expect**. Perlu diketahui saat ini sebaiknya menggunakan **ssh** (ssh 2.x) karena untuk mencegah disadapnya kata sandi (*password*) di jaringan. Telnet dan rlogin mengirimkan password secara tidak terenkripsi.

- **vnc**

VNC [<http://www.uk.research.att.com/vnc/>] merupakan salah satu program aplikasi yang memungkinkan proses administrasi sistem dari jarak jauh. Bukan saja untuk sistem GNU/Linux, atau FreeBSD tetapi juga untuk MS Windows.

- **XWindow**

XWindow memiliki kemampuan untuk mengirim display ke mesin jarak jauh (*export display*). Dengan cara ini maka dimungkinkan proses menjalankan aplikasi di mesin lainnya. Sehingga proses administrasi sistem yang menggunakan program GUI di server dapat juga dijalankan dari jarak jauh.



Gambar 20: VNC

## 11 Penutup

Freeware saat ini sudah tak bisa dianggap sebagai perangkat lunak kelas hobbies lagi. Solusi yang diberikan oleh perangkat lunak freeware dapat digunakan untuk menyusun suatu server yang handal dan memberikan solusi nyata.

Sebelum memanfaatkan freeware sebaiknya dipahami terlebih dahulu mengenai karakter tiap perangkat lunak tersebut, dan melakukan pemilihan sesuai dengan kebutuhannya. Keberadaan source code menjadikan freeware tersebut dapat dimanfaatkan secara maksimal dan disesuaikan dengan kebutuhan yang ada. Beberapa solusi freeware seperti Linux Virtual Server mampu memberikan solusi bagi layanan web ataupun email yang memiliki beban besar.

Perangkat bantu seperti Mozilla, PhpGTK, SashXB memungkinkan mengembangkan aplikasi tidak saja di sisi server, tetapi juga di sisi client dan bersifat portabel.

## Pustaka

- [1] MTA comparison. [<http://shearer.org/en/writing/mtacomparison.html>]
- [2] Jenis-jenis perangkat lunak MTA [<http://shearer.org/en/writing/mtacomparison.html>]
- [3] Creating applications with Mozilla. [<http://books.mozdev.org/chapters/index.html>]
- [4] Message Access Paradigms and Protocols. <ftp://ftp.cac.washington.edu/mail/imap.vs.pop>
- [5] TUX Webserver. An analysis. <http://www.citi.umich.edu/techreports/reports/citi-tr-00-8.pdf> -
- [6] Kernel Korner: kHTTPd, a Kernel-Based Web Server. [<http://www.linuxjournal.com/article.php?sid=4132>]