

TUGAS AKHIR

REMASTERING DISTRO UBUNTU UNTUK GATEWAY, DISTRIBUTED FILTERING DAN SECURITY

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Strata-1**

Teknik Informatika



Oleh :

Nama : Lantip Agil Nugroho

No. Mahasiswa : 06523120

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

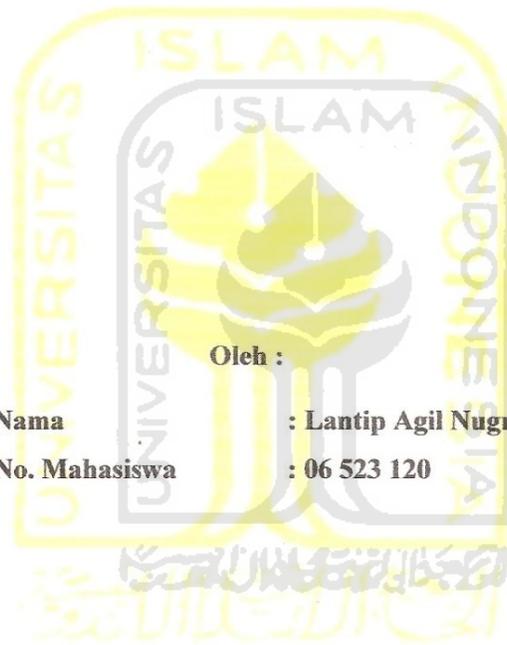
UNIVERSITAS ISLAM INDONESIA

YOGYAKARTA

LEMBAR PENGESAHAN PEMBIMBING

**REMASTERING DISTRO UBUNTU UNTUK GATEWAY,
DISTRIBUTED FILTERING DAN SECURITY**

TUGAS AKHIR



Oleh :

Nama : Lantip Agil Nugroho

No. Mahasiswa : 06 523 120

Yogyakarta, 15 Februari 2011

Pembimbing

A handwritten signature in black ink, appearing to be 'Yudi Prayudi', written over a thin horizontal line.

Yudi Prayudi, S.Si, M.Kom.

LEMBAR PENGESAHAN PENGUJI

**REMASTERING DISTRO UBUNTU UNTUK GATEWAY,
DISTRIBUTED FILTERING DAN SECURITY**

TUGAS AKHIR

Oleh :

Nama : Lantip Agil Nugroho

No. Mahasiswa : 06 523 120

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Informatika Fakultas
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Februari 2011

Tim Penguji

Yudi Prayudi, S.Si, M.Kom.

Ketua

Ari Sujarwo, S.Kom.

Anggota I

Irving Vitra Papatungan, S.T, M.Sc.

Anggota II

**Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia**

Yudi Prayudi, S.Si, M.Kom.

LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR

Saya yang bertandatangan di bawah ini,

Nama : Lantip Agil Nugroho

No. Mahasiswa : 06523120

Menyatakan bahwa seluruh komponen isi dalam Laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti bahwa ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, maka saya siap menanggung resiko dan berkonsekuensi apapun.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagai mana mestinya.

Yogyakarta, 15 Februari 2011

Lantip Agil Nugroho

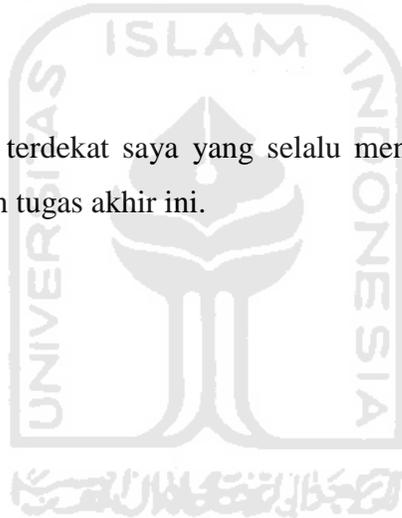
HALAMAN PERSEMBAHAN

Tugas akhir ini kupersembahkan untuk:

Allah S.W.T yang telah memberikan rezeki, rahmat dan hidayah serta anugerah yang selalu membimbing saya hingga sampai sekarang ini.

Orang tua yang selalu mendukung, di setiap langkah, selalu memberikan kasih sayang terhadap anak-anaknya. Semoga kelak saya bisa membalas semua kasih sayangmu

Teman, kerabat dan orang terdekat saya yang selalu mendukung dari awal hingga akhir perjuangan pengerjaan tugas akhir ini.



MOTTO

“kesempurnaan tercipta dari hal-hal yang kecil, namun kesempurnaan bukanlah sesuatu yang kecil”.

“setiap orang membutuhkan tiga hal yang akan membuat kita bahagia di dunia ini, yaitu seseorang untuk dicintai, sesuatu untuk dilakukan, dan sesuatu untuk diharapkan.”



KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Alhamdulillah, dengan mengucap syukur kehadiran Allah SWT yang senantiasa memberikan petunjuk serta melimpahkan berkah dan rahmat-Nya, sehingga Tugas Akhir ini dapat diselesaikan dengan judul “Remastering Distro Ubuntu untuk Gateway, Distributed Filtering dan Security”.

Tugas Akhir ini membahas pembungkusan ulang atau yang sering disebut dengan istilah remastering distro yang sudah ada yaitu ubuntu untuk disesuaikan dengan kebutuhan dalam pembangunan jaringan lokal. Dunia komputer saat ini sudah semakin luas dan menyentuh segala aspek kehidupan. Hampir setiap kebutuhan hidup manusia membutuhkan peran komputer untuk membantu penyelesaian masalah agar lebih efektif dan efisien. Pengguna komputer semakin beragam dengan tingkatan kemampuan masing-masing. Dengan teknik remastering ini diharapkan mampu menciptakan sebuah distro yang lebih mudah digunakan dan sesuai dengan kebutuhan masing-masing pengguna.

Ucapan terimakasih kami ucapkan kepada segenap pihak yang secara langsung maupun tidak langsung telah membantu serta memudahkan proses terselesaikannya Tugas Akhir ini. Semoga Allah SWT melimpahkan rahmat serta memberikan balasan yang sebaik-baiknya untuk semua. Ucapan terima kasih saya tujukan kepada:

1. Allah SWT atas segala berkah dan rahmat-Nya sehingga Tugas Akhir ini dapat diselesaikan.
2. Orang tua, Kakak, dan adik atas kasih sayang, segala limpahan do'a dan dukungan.
3. Yang saya hormati Bapak Ir. Gumbolo HS, M.SC selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.

4. Bapak Yudi Prayudi S.Si,M.Kom selaku Ketua Jurusan Teknik Informatika FTI UII serta sebagai dosen pembimbing dalam penyusunan Tugas Akhir ini.
5. Seluruh Dosen di Jurusan Teknik Informatika Universitas Islam Indonesia yang dengan sabar mewariskan ilmunya kepada kami, serta memberikan motivasi dan semangat. Semoga ilmu yang telah diajarkan dapat menjadi amal.
6. Teman-teman angkatan 2006 Informatika UII.
7. Teman-teman kos yang selalu setia mendampingi saya dalam pengerjaan Tugas Akhir ini.
8. Orang terdekat yang selalu dengan setia mendukung serta memberikan semangat, sehingga Tugas Akhir ini bisa diselesaikan tepat waktu.
9. Semua pihak yang tidak bisa disebutkan satu per Satu

Kiranya tiada gading yang tak retak, maka saya pun tidak terlepas dari kesalahan. Saya berharap laporan ini dapat menjadi sesuatu yang bermanfaat bagi kita semua untuk menambah wawasan dan pengetahuan, Amin.

Wassalamu'alaikum Wr.Wb

Yogyakarta, 15 Februari 2011

Lantip Agil Nugroho

SARI

Sistem Operasi Linux merupakan sistem operasi yang sangat cocok untuk digunakan dan diimplementasikan dalam pengaturan keamanan jaringan. Hal ini dikarenakan linux mampu bekerja dengan sangat tangguh dan mempunyai tingkat kemanan yang baik. Namun masalah yang dihadapi adalah ketika pengguna yang baru pertama kali mengoperasikan Linux. Diantaranya yaitu konfigurasi harus dilakukan secara manual dengan menggunakan *command* pada *terminal*. *Command* yang digunakan sangatlah beragam dan sangat sulit untuk diingat. Oleh karena itu dibutuhkanlah sebuah *tools* yang dapat membantu pengguna dalam hal konfigurasi dengan cara yang lebih mudah. Maka dibangunlah sebuah distro atau sistem operasi berbasis Linux ubuntu yang sudah dilengkapi dengan aplikasi, *tools* dan *interface* yang sangat *user friendly* agar lebih mudah digunakan untuk membangun jaringan lokal yang handal, aman dan efisien menggunakan teknik remastering.

Remastering adalah sebuah teknik atau cara untuk merombak kembali sebuah sistem operasi atau distro kemudian membungkusnya kembali menjadi sebuah distro ubuntu yang baru dan sudah dilengkapi dengan aplikasi-aplikasi yang diinginkan serta konfigurasi yang sudah ditentukan.

Proses remastering dilakukan secara bertahap, mulai dari persiapan sistem operasi, konfigurasi *default*, penyatuan masing-masing aplikasi yang akan disertakan dan yang terakhir adalah proses remastering. Aplikasi yang bisa digunakan untuk membantu proses remastering sangatlah beragam dan masing-masing mempunyai kekurangan dan kelebihan masing-masing. Pemilihan aplikasi yang digunakan disesuaikan dengan kebutuhan. Aplikasi yang digunakan adalah aplikasi yang mampu membuat sebuah dokumentasi *image* dari sistem yang sudah ada dan sudah dikonfigurasi.

Kata kunci : Linux, Distro, Gateway, Distributed Filtering.

TAKARIR

<i>anonymous</i>	pengguna tanpa nama
<i>auto scheduling</i>	penjadwalan otomatis
<i>backup</i>	penyimpanan data / penggandaan data
<i>boot</i>	proses pemuatan
<i>bundle</i>	paket
<i>cache</i>	penyimpanan sementara
<i>client</i>	komputer yang meminta layanan
<i>compile</i>	menyusun
<i>console</i>	aplikasi kerja berbasis text
<i>database</i>	penyimpanan data
<i>default</i>	normal
<i>directory</i>	folder / map
<i>distro</i>	sistem operasi varian linux
<i>explore</i>	pengembangan, penjelajahan
<i>file</i>	berkas
<i>filtered object</i>	objek situs yang disaring
<i>firewall</i>	aplikasi untuk menyaring koneksi
<i>gateway</i>	server sebagai pintu keluar
<i>install</i>	penanaman aplikasi
<i>interface</i>	tampilan
<i>internal</i>	didalam sistem
<i>kernel</i>	inti sistem operasi
<i>keyword</i>	kata kunci
<i>library</i>	paket pelengkap
<i>password</i>	kata kunci rahasia

<i>platform</i>	mesin komputer
<i>pool</i>	jangkauan / wadah
<i>repository</i>	kumpulan aplikasi
<i>request</i>	permintaan
<i>router</i>	komputer yang mencari jalur
<i>script</i>	naskah
<i>security</i>	keamanan sistem
<i>server</i>	komputer yang memberikan layanan
<i>synchronize</i>	penyesuaian
<i>URL</i>	alamat website
<i>user friendly</i>	mudah digunakan
<i>user</i>	pengguna
<i>update</i>	pembaruan



DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	iii
LEMBAR PERNYATAAN KEASLIAN HASIL TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
MOTTO	vi
KATA PENGANTAR	vii
SARI	ix
TAKARIR	x
DAFTAR ISI	xii
DAFTAR GAMBAR	xv
BAB 1	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Metode Penelitian	3
1.6.1 Metode Pengumpulan Literatur	3
1.6.2 Implementasi dan Konfigurasi Sistem	3
1.7 Sistematika Penulisan	4
BAB 2	6
2.1 Konsep Dasar Sistem Operasi	6
2.2 Linux	7
2.2.1 GNU	9

2.3	Remastering.....	10
2.4	Lancop.....	11
2.5	Web Filtering.....	11
2.5.1	Proxy.....	12
2.5.2	Distributed Filtering Proxy.....	14
2.6	FTP.....	17
2.7	DHCP.....	18
BAB 3	20
3.1	Analisis Masalah.....	20
3.2	Desain Alur Data dan Proses Aplikasi Sistem Operasi.....	22
3.3	Desain Arsitektur Jaringan Implementasi Distro.....	23
3.4	Analisis dan Persiapan Kebutuhan Sistem.....	24
3.4.1	Kebutuhan Perangkat Keras.....	24
3.4.2	Kebutuhan Perangkat Lunak.....	25
3.5	Instalasi dan Konfigurasi Sistem.....	27
BAB 4	30
4.1	Implementasi Secara Umum.....	30
4.2	Tahapan Implementasi Perangkat Lunak.....	30
4.3	Implementasi Hasil Perancangan.....	32
4.3.1	Instalasi dan konfigurasi squid 3.0.....	32
4.3.2	Instalasi dan konfigurasi Apache dan php.....	34
4.3.3	Instal dan konfigurasi FTP.....	35
4.3.4	Install dan konfigurasi MySQL dan PHPMyAdmin.....	38
4.3.5	Instalasi dhcp server.....	39
4.3.6	Pembuatan <i>Script</i> bash untuk menghubungkan antar aplikasi.....	40
4.3.7	Membuat web interface.....	41
4.3.8	Customisasi Tampilan Lancop.....	45
4.3.9	Instalasi Remastersys.....	47

4.4	Pengujian Hasil Implementasi.....	50
4.4.1	Tampilan Awal.....	50
4.4.2	Komponen aplikasi dan service sistem Distro	51
4.4.3	Konfigurasi Via Web Interface	52
BAB 5	58
5.1	Kesimpulan.....	58
5.2	Saran.....	58



DAFTAR GAMBAR

Gambar 2.1 Konsep Dasar Sistem Operasi.....	6
Gambar 2.2 <i>Proxy Server</i>	13
Gambar 2.3 Konsep Distributed Filtering.....	16
Gambar 3.1 Desain Arsitektur Distro	22
Gambar 4.1 web interface Lancop	41
Gambar 4.2 custom background welcome screen.....	46
Gambar 4.3 welcome screen	47
Gambar 4.4 <i>homescreen reconstructor</i>	48
Gambar 4.5 Konfigurasi dasar remastersys	48
Gambar 4.6 proses remastering.....	49
Gambar 4.7 tampilan login	50
Gambar 4.8 tampilan desktop	51
Gambar 4.9 menambah blocked URL	53
Gambar 4.10 menghapus URL yang terblokir.....	53
Gambar 4.11 update rule dari FirstServer.....	54
Gambar 4.12 pendaftaran untuk mendapat code update.....	55
Gambar 4.13 konfigurasi dhcp.....	56
Gambar 4.14 restart aplikasi	57

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan dan kemajuan komputer semakin pesat. Hampir setiap segi aspek kehidupan membutuhkan bantuan komputer, seperti aspek pendidikan, sosial, dan ekonomi. Seiring dengan perkembangan dan kemajuan komputer, maka semakin banyak pengguna yang terlibat di dalamnya dengan berbagai kebutuhan yang berbeda-beda. Dengan keanekaragaman kebutuhan ini, maka dibutuhkanlah sebuah alat bantu *software* yang mudah digunakan dan dioperasikan.

Linux merupakan sebuah sistem operasi yang mulai banyak dikenal akhir-akhir ini. Hal ini dikarenakan Linux merupakan *software* yang bersifat *free* dan *open source*. Seorang pengguna mempunyai hak untuk menggunakan, menyebarkan dan memodifikasi *software* ini tanpa harus membayar lisensi. Selain itu Linux juga merupakan sistem operasi yang sangat cocok untuk digunakan dan diimplementasikan dalam dunia jaringan komputer. Hal ini dikarenakan linux mampu bekerja dengan sangat tangguh dan mempunyai tingkat keamanan yang baik.

Untuk *administrator* pemula, konfigurasi sistem operasi Linux merupakan hal yang rumit, karena konfigurasi harus dilakukan secara manual dengan menggunakan *command line* pada *console*. *Command* yang digunakan sangatlah beragam dan sangat sulit untuk diingat, oleh karena itu, bisa menjadi sebuah kesulitan bagi *administrator* dalam melakukan tugas-tugas konfigurasinya. Oleh karena itu dibutuhkanlah sebuah *tools* yang dapat membantu *administrator* dalam hal konfigurasi dengan cara yang lebih mudah. Maka dibangunlah sebuah distro atau sistem operasi berbasis Linux yang sudah dilengkapi dengan aplikasi, *tools* dan *interface* yang sangat *user friendly* agar lebih mudah digunakan untuk membangun jaringan lokal yang handal, aman dan efisien menggunakan teknik remastering.

Remastering adalah sebuah teknik atau cara untuk merombak kembali sebuah sistem operasi atau distro kemudian membungkusnya kembali menjadi sebuah distro ubuntu yang baru dan sudah dilengkapi dengan aplikasi-aplikasi yang diinginkan serta konfigurasi yang sudah ditentukan. Dengan teknik remastering ini diharapkan bisa menghasilkan distro Ubuntu yang lebih mudah digunakan dan dikonfigurasi agar bisa bekerja sesuai dengan kebutuhan dan kepentingan pengguna.

1.2 Rumusan Masalah

1. Adanya kesulitan penggunaan pada sebagian besar sistem operasi berbasis Linux.
2. Masih kurang lengkapnya fasilitas yang dimiliki oleh sebagian besar sistem operasi linux.

1.3 Batasan Masalah

Adapun batasan masalah yang diangkat dalam penelitian ini sebagai berikut :

1. Kesulitan yang akan diselesaikan diarahkan pada pengaturan konfigurasi jaringan.
2. Fasilitas yang ditambahkan terfokus pada gateway, distributed filtering dan security.

1.4 Tujuan Penelitian

Membangun sebuah distro atau sistem operasi berbasis Linux ubuntu yang sudah dilengkapi dengan aplikasi, *tools* dan *interface* yang sangat *user friendly* agar lebih mudah digunakan untuk membangun jaringan lokal yang handal, aman dan efisien.

1.5 Manfaat Penelitian

Memudahkan para pengguna komputer yang ingin lebih memahami tentang jaringan secara mendasar dan para *network administrator* yang akan membangun sebuah jaringan lokal yang aman dan mudah.

1.6 Metode Penelitian

1.6.1 Metode Pengumpulan Literatur

Pada penelitian ini, metode pengumpulan literatur yang digunakan:

Metode studi pustaka, yaitu metode pengumpulan literatur melalui buku-buku dan Internet yang berkaitan dengan penelitian yang dilakukan meliputi pemilihan perangkat lunak yang digunakan serta pencarian referensi mengenai tahapan dalam implementasi uji performa sistem operasi hasil dari *remastering*.

1.6.2 Implementasi dan Konfigurasi Sistem

Metode implementasi disusun berdasarkan hasil perolehan dari metodologi pengumpulan literatur yang meliputi:

a. Analisis Masalah

Tahap ini merupakan tahap pengamatan dan perincian masalah yang ada pada proses pembuatan dan implementasi distro hasil *remastering*.

b. Desain Arsitektur Jaringan

Tahap ini merupakan tahap perancangan arsitektur jaringan komputer yang akan diterapkan dalam penggunaan dan pengimplementasian distro hasil *remastering*.

c. Desain Alur data proses aplikasi

Tahap ini merupakan perancangan visualisasi data yang mengalir pada setiap proses yang terjadi pada masing-masing aplikasi dari awal hingga akhir.

d. Analisis dan Pengadaan Perangkat Keras

Tahapan ini merupakan tahap pengadaan perangkat keras yang meliputi komputer untuk Firstserver, dan Secondserver.

e. Instalasi dan Konfigurasi Sistem

Tahapan ini merupakan tahap instalasi dan konfigurasi dari masing-masing aplikasi dan sistem yang didalamnya terdapat *proxy*, *FTP*, *Mysql*, *webserver*, dan pembuatan aplikasi sederhana dengan metode *scripting shell Linux*.

f. Remastering

Tahapan ini merupakan tahap pembungkusan sistem operasi yang sudah terinstal semua aplikasi dan konfigurasi yang dibutuhkan sehingga menghasilkan sebuah distro Linux baru yang siap pakai.

g. Pengujian

Setelah *remastering* selesai dilakukan, tahap selanjutnya adalah mencoba melakukan uji coba instalasi pada perangkat yang sudah disediakan. Setelah itu dilakukan ujicoba terhadap semua aplikasi, tool dan perubahan yang dilakukan.

1.7 Sistematika Penulisan

BAB I Pendahuluan

Membahas tentang latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian serta manfaat dari penelitian dan metodologi penelitian yang diangkat menjadi materi penulisan laporan tugas akhir *remastering* distro ubuntu untuk gateway, distributed filtering dan security

BAB II Landasan Teori

Membahas dasar-dasar teori yang digunakan dalam perancangan dan pembangunan sebuah distro baru dengan teknik *remastering*. Dasar Teori tersebut

BAB III Metodologi

Memuat urian tentang analisis masalah, gambaran umum sistem, analisis kebutuhan sistem yang mencakup kebutuhan perangkat keras dan perangkat lunak yang digunakan untuk membantu penyelesaian tugas akhir.

BAB IV Hasil dan Pembahasan

Memuat dokumentasi mulai dari tahap instalasi, konfigurasi dan pengujian terhadap aplikasi, yaitu dengan uji coba *update filtered object* secara otomatis dari client kepada server serta fitur-fitur lain yang disertakan pada distro hasil *remastering*.

BAB V Penutup

Memuat kesimpulan-kesimpulan dari seluruh rangkaian proses implementasi perangkat lunak, baik pada tahap analisis, perancangan, implementasi, terutama pada analisis kinerja aplikasi. Bab ini juga membahas saran yang dapat digunakan oleh pihak yang berkepentingan maupun untuk peneliti terhadap kekurangan serta keterbatasan dalam penelitian ini.

BAB 2

LANDASAN TEORI

2.1 Konsep Dasar Sistem Operasi

Secara sederhana sistem komputer terdiri atas *hardware*, sistem operasi, program aplikasi dan *user* yang menggunakan sistem komputer tersebut. Secara sederhana sistem operasi dapat didefinisikan sebagai antar muka antara *user* dengan *hardware*. Atau dengan kata lain, sistem operasi merupakan *software* yang digunakan untuk mengatur kerja *hardware* serta menyediakan lingkungan dimana seorang *user* dapat menjalankan program aplikasi. Biasanya, istilah Sistem Operasi sering ditujukan kepada semua *software* yang masuk dalam satu paket dengan sistem komputer sebelum aplikasi-aplikasi *software* terinstall. Sistem operasi bertugas untuk melakukan kontrol dan manajemen perangkat keras serta operasi-operasi dasar sistem, termasuk menjalankan *software* aplikasi seperti program-program pengolahan kata dan *web browser*, menjamin aplikasi *software* lainnya dapat menggunakan memori, melakukan *input* dan *output* terhadap peralatan lain, dan memiliki akses kepada sistem *file*. Gambar 2.1 menjelaskan tentang susunan komponen dari paling dasar sampai dengan pengguna.



Gambar 2.1 Konsep Dasar Sistem Operasi

Secara umum, sistem operasi adalah *software* pada lapisan pertama yang diletakkan pada memori komputer pada saat komputer dihidupkan. Sedangkan *software-software* lainnya dijalankan setelah sistem operasi berjalan.

Semua sistem operasi terdiri atas tiga bagian, yaitu:

- 1 Kernel

Kernel merupakan program yang dimuat ke dalam memori ketika komputer dinyalakan. *Kernel* bertanggung jawab untuk mengendalikan *hardware*, menyediakan lingkungan agar program lainnya dapat berjalan serta mengatur alokasi waktu dan memori untuk program-program yang sedang dijalankan oleh komputer. Selain itu *kernel* juga bertugas untuk mengendalikan bagaimana *file* dan *directory* disimpan dalam *harddisk*.

- 2 Utilitas Standar

Utilitas standar adalah program kecil yang digunakan untuk melakukan fungsi-fungsi sederhana seperti menampilkan isi *file*, menampilkan daftar *file* dalam suatu *directory*, menyalin *file* serta fungsi-fungsi sederhana lainnya.

- 3 File Database Sistem

File database sistem merupakan *berkas-berkas* yang dibutuhkan agar sistem operasi dapat berjalan. Contoh *file database sistem* adalah *file /etc/passwd*, */etc/shadow* pada sistem operasi UNIX.

2.2 Linux

Linux merupakan salah satu jenis sistem operasi. Linux adalah nama yang diberikan kepada semua sistem operasi komputer yang bertipe Unix. Sistem operasi ini merupakan salah satu contoh hasil pengembangan *Free and Open Source Software*. *Free and Open Source Software* merupakan sebuah *software* yang bebas untuk digunakan, disebar dan dimodifikasi oleh siapa saja dan untuk tujuan apa saja tanpa harus membayar lisensi pada pihak manapun. Hal inilah yang memacu

Linux untuk berkembang sangat pesat, karena makin banyaknya pengembang yang turut andil dalam memajukan dan terus memperbaiki kinerja sistem operasi Linux.

Linux dikembangkan oleh mahasiswa Finlandia yang bernama Linus Benedict Torvalds. Dulunya, Linux merupakan proyek hobi yang diinspirasi dari Minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum. Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991. Tanggal 5 Oktober 1991, Linux mengumumkan versi resmi Linux, yaitu versi 0.02 yang pada saat itu hanya dapat menjalankan bash (GNU Bourne Again Shell) dan GCC (GNU C Compiler). Istilah Linux sebenarnya hanya mengacu pada nama sebuah *kernel*.

Agar menjadi sebuah sistem operasi yang lengkap, *kernel* Linux kemudian digabungkan dengan berbagai program aplikasi. Maka kemudian muncullah istilah distro. Distro adalah *bundle* dari *kernel* Linux beserta sistem dasar Linux, program instalasi, *tools-tools* standar, dan program-program lain yang kebanyakan berlisensi GPL (General Public License). Terdapat banyak sekali distro Linux, diantaranya.

- RedHat
- Fedora
- Ubuntu
- Mandriva
- SuSE
- dan lain sebagainya

Linux memiliki berbagai kelebihan jika dibandingkan dengan sistem operasi lainnya. Diataranya adalah:

- 1 Portabilitas, yaitu Linux dapat dijalankan atau bahkan dipindahkan dari satu *platform* komputer ke *platform* komputer lainnya.
- 2 Linux menyertakan aplikasi dan *library* untuk pengembangan sistem.
- 3 Murah dan mudah untuk didapatkan serta mendidik untuk pintar, karena setiap proses dan *source code* linux bersifat terbuka, sehingga bisa dipelajari secara langsung baik sistem maupun cara kerjanya.

2.2.1 GNU

GNU (GNU's Not Unix) adalah sebuah yayasan yang dibuat untuk pengembangan *software* bebas. Tujuan utama dari proyek tersebut adalah untuk membuat sistem operasi mirip UNIX namun bebas. Keberhasilan pertama proyek ini adalah membuat GNU Emacs, sebuah *text editor* multifungsi. Sementara keberhasilan berikutnya adalah GCC (GNU C Compiler), yaitu sebuah compiler untuk bahasa C. Program-program tersebut dilisensikan dibawah bendera GNU GPL (General Public License).

Sebagian besar *software* yang terdapat pada GNU/Linux, termasuk *kernel* Linux itu sendiri, juga menganut lisensi GNU General Public License (GPL). Lisensi ini disebut juga *Free Software*, yang sering disalahartikan sebagai *software* gratis (*freeware*), padahal maksud sebenarnya adalah *software* bebas.

Secara garis besar, GPL menjamin tiap orang yang menggunakan program dengan lisensi GPL untuk memiliki empat macam kebebasan sebagai berikut:

- 1 Bebas untuk menjalankan program komputer, untuk tujuan apa saja.
- 2 Bebas untuk mempelajari bagaimana cara kerja program, dan memodifikasinya sesuai dengan kebutuhan.
- 3 Bebas untuk mendistribusikan salinan, baik gratis maupun dengan menarik biaya.
- 4 Kebebasan untuk memperbaiki dan mengeluarkan perbaikan tersebut kepada umum, sehingga mendatangkan keuntungan bagi masyarakat secara keseluruhan.

Dalam rangka pendistribusian, diperkenankan untuk memungut biaya atas penyalinan dan garansi atas program yang memiliki lisensi tersebut. Dan satu lagi point penting adalah program yang menggunakan lisensi GPL mengharuskan setiap perubahan, penambahan atau perbaikan dari program komputer tersebut dilisensikan dengan menggunakan lisensi GPL.

2.3 Remastering

Dalam dunia sistem operasi, khususnya dalam sistem operasi Linux, terdapat sebuah istilah yang serint terdengar yaitu *remastering*. *Remastering* Linux adalah sebuah proses untuk membuat sebuah Linux baru dari Linux yang sudah ada. Hasil *remastering* Linux adalah Linux yang mirip dengan Linux induk namun telah mengalami beberapa modifikasi yang membuatnya berbeda dibandingkan dengan Linux induk, misal tema tampilan, perangkat lunak yang terbundel dengannya dan sebagainya.

Tujuan dari *remastering* Linux adalah Linux induk yang ada belum bisa memenuhi kebutuhan penggunanya. Jika bisa memenuhi, diperlukan suatu proses atau usaha misalkan meng-*install* aplikasi-aplikasi yang diperlukan. Proses *remastering* ini membuat usaha tersebut tidak perlu dilakukan karena proses instalasi telah dilakukan oleh pengembang. Sebagai contoh adalah OSGX. OSGX dirancang dan dibuat berdasar analisis tentang perangkat lunak yang diperlukan selama belajar di ITB. OSGX diharapkan kebutuhan aplikasi mahasiswa ITB bisa terpenuhi dengan sekali proses instalasi.

Pada mulanya *remastering* merupakan sebuah proses *backup* sebuah sistem yang sudah ada. Pada perkembangan pertama, *backup* hanya bisa dilakukan hanya sebatas pada *file* saja. Seiring dengan perkembangan teknologi dan kebutuhan, maka muncullah fitur yang lebih kompleks, yaitu *backup* yang sampai pada *level* sistem operasi, konfigurasi serta database.

Proses *remastering* membutuhkan bantuan dari aplikasi. Aplikasi yang bisa digunakan sangatlah beragam, dan masing-masing mempunyai fitur-fitur yang berbeda. Berdasarkan fitur-fitur dari aplikasi *remastering*, maka aplikasi tersebut bisa digolongkan menjadi dua tipe. Tipe pertama adalah *remastering live CD* dan yang kedua adalah *remastering backup*. *Remastering live CD* yaitu *remastering* yang hanya bisa dilakukan terhadap file image pada *live cd* dan kemudian hanya sebatas level itulah perubahan bisa dilakukan. Perubahan yang dilakukan umumnya sangat

seederhana, hanya terbatas pada tampilan dan penambahan aplikasi. Sedangkan remastering backup adalah remastering yang lebih lengkap dan menyeluruh terhadap sebuah sistem yang sudah ada. Pada remastering tipe ini, tahap remastering dilakukan pada tahap akhir atau bisa dikatakan remastering dilakukan setelah konfigurasi dan persiapan sistem benar-benar sudah matang dan sudah siap. Berbeda dengan remastering *live CD* yang justru proses remastering dilakukan pada awal proses, sehingga memungkinkan persiapan dan rancangan sistem kurang matang.

2.4 Lancop

Pada penelitian ini, remastering dilakukan untuk dua distro baru yang berbeda. Distro pertama adalah FirstServer dan yang kedua adalah SecondServer (Lancop). FirstServer merupakan distro yang sudah dibekali dengan aplikasi-aplikasi tertentu yang dikhususkan untuk tujuan penyimpanan dan gudang *keyword* yang akan digunakan oleh proxy. Distro kedua adalah SecondServer. Distro ini memiliki *nickname* Lancop. Aplikasi dan konfigurasi dari distro ini dikhususkan agar distro Lancop mampu melakukan *update keyword* yang berada pada FirstServer. FirstServer dikhususkan sebagai server pusat, sedangkan Lancop digunakan sebagai gateway yang mampu melakukan *update* pada FirstServer, sehingga aturan pada distro Lancop yang terinstall akan selalu terbaharui.

2.5 Web Filtering

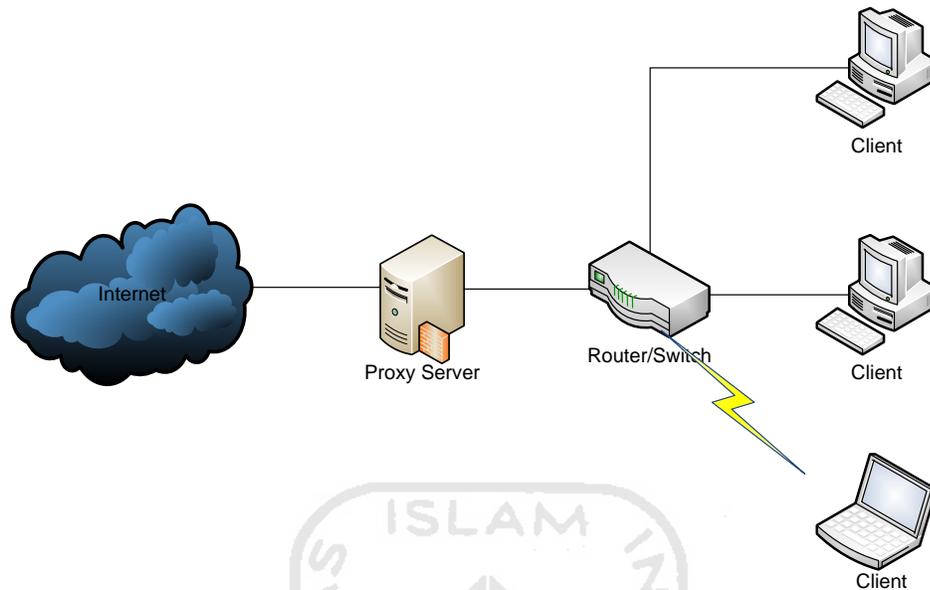
Web filtering adalah program yang dapat mengatur halaman *web* yang masuk dan menentukan apakah halaman tersebut berhak ditampilkan atau tidak kepada pengguna. *Filter* memeriksa asal atau isi dari halaman *web* terhadap seperangkat aturan yang disediakan oleh administrator. *Web filtering* memungkinkan untuk memblokir halaman halaman dari situs *web* yang mungkin mengandung konten pornografi, *spyware*, virus, dan konten yang lainnya. Beberapa produk *web filtering*

juga menyertakan *report*, sehingga administrator dapat melihat jenis lalu lintas yang sedang terjadi, sehingga lalu lintas data lebih tercontrol.

2.5.1 Proxy

Proxy server adalah sebuah komputer server atau program komputer yang dapat bertindak sebagai makelar sebuah koneksi dari client kepada sebuah server. *Proxy* server bertindak sebagai *gateway* terhadap sebuah jaringan yang terdapat di atasnya. *Proxy* server bekerja dan beroperasi tanpa terlihat oleh komputer client, seroang pengguna yang berinteraksi dengan Internet melalui sebuah *proxy* server tidak akan mengetahui bahwa sebuah *proxy* server sedang menangani request yang dilakukannya. Server yang menerima request dari *proxy* server akan menerjemahkan request-request tersebut seolah-olah request itu datang secara langsung dari komputer.

Proxy server juga dapat digunakan untuk mengamankan jaringan pribadi yang dihubungkan ke sebuah jaringan publik. *Proxy* server memiliki fungsi lebih banyak dibandingkan dengan *router* yang hanya memiliki fitur packet filtering, karena memang *proxy* server beroperasi pada level yang lebih tinggi dan memiliki control yang lebih menyeluruh terhadap akses jaringan. Posisi *Proxy* server dijelaskan pada gambar 2.2.



Gambar 2.2 Proxy Server

Dilihat dari gambar 2.2, dapat dijelaskan bahwa sesuai posisi *proxy* server yang berada diantara client dan server, maka *proxy* server bisa berfungsi sebagai:

1 Firewall

Proxy server yang dikonfigurasi secara benar, akan meningkatkan performa dan keamanannya. Karena *proxy* bekerja pada layer aplikasi (dalam OSI Layer), maka *filtering* yang dilakukan oleh *proxy* lebih cerdas daripada *firewall* biasa. *Proxy web* server dapat mengecek URL dari *outgoing request* (permintaan akses keluar) untuk halaman *web* dengan memeriksa pesan HTTP, GET dan POST. Dengan kemampuan ini, administrator dapat melarang atau mengijinkan akses ke domain tertentu. *Firewall* biasa, tidak dapat melihat nama domain di dalam pesan tersebut, karena *firewall* hanya memeriksa *header* dari paket data.

2 Gateway

Untuk dapat mengakses Internet, sebuah komputer harus memiliki sebuah ip publik. Hal ini menjadi mustahil karena IP yang tersedia di dunia sangat terbatas, sehingga untuk dapat mengakses Internet secara bersama-sama dengan menggunakan satu IP publik, dibutuhkan sebuah komputer yang memiliki IP public, yang digunakan sebagai gateway untuk komputer lain. Dalam hal ini *proxy* server juga berfungsi sebagai gateway. Server ini mempunyai dua *interface*, satu untuk antarmuka dengan Internet dan satu untuk antarmuka dengan jaringan local.

3 Cache

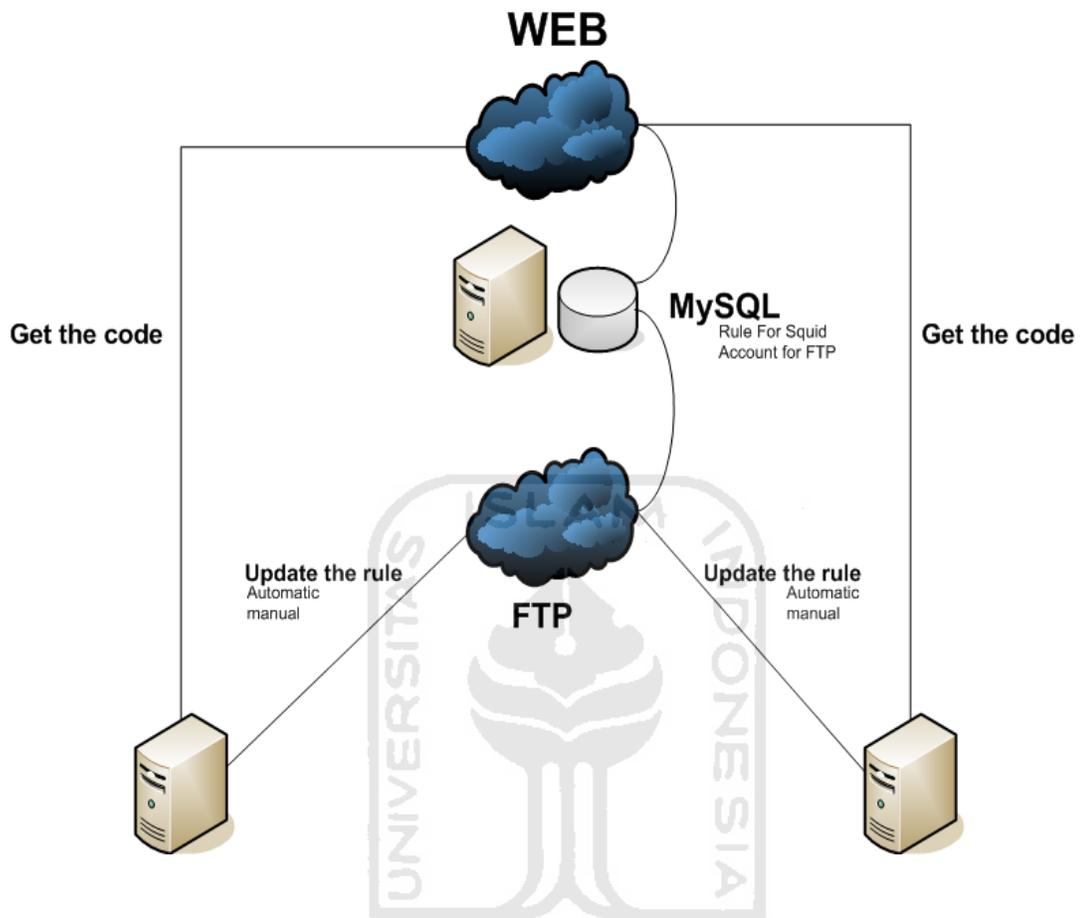
Fungsi *proxy* server yang lain adalah untuk *web caching*. *Caching* di sini diartikan sebagai penyimpanan Internet object (gambar/halaman *web*) dari suatu *website* yang sudah pernah diakses, sehingga bila akan mengakses objek yang sama di Internet, tidak perlu mengambil dari Internet, tetapi cukup dari *proxy* karena sudah disimpan. Dengan adanya *caching* ini, *bandwidth* yang dipakai akan lebih hemat, dan dapat mempercepat akses ke *web* site. Begitu juga bila *web* server yang dituju ternyata mati atau mengalami gangguan, client tetap bisa mengaksesnya.

2.5.2 Distributed Filtering Proxy

Salah satu kemampuan dari *proxy* server adalah *firewall*, yaitu *proxy* server mampu untuk menyaring koneksi yang diminta oleh client. *Proxy* server mampu memblokir sebuah request dari client kepada sebuah halaman *web* berdasarkan URL yang dikirimkan. Sebuah *proxy* server akan menentukan kebijakan apa yang akan diberikan kepada sebuah request berdasarkan dari *keyword* dari URL yang dituju. Sebagai contoh ketika seorang administrator membuat sebuah rule bahwa semua URL yang mengandung kata “sex” akan ditolak, maka setiap pengguna yang mengarahkan ke semua URL yang mengandung kata sex akan ditolak.

Terdapat 2 mekanisme pendefinisian *keyword* yang akan diberikan kebijakan, yang pertama adalah dengan pendefinisian secara langsung dalam *file* konfigurasi *proxy*, dan yang kedua adalah dengan pendefinisian dalam sebuah *file* yang terintegrasi dengan *file* konfigurasi. Terdapat keuntungan dan kerugian dari masing-masing mekanisme tersebut. Namun akan lebih efisien jika seorang administrator menggunakan mekanisme yang kedua, karena kita tidak akan banyak menyentuh *file* konfigurasi utama yang jika terjadi kesalahan bisa berdampak secara langsung terhadap kinerja *proxy* yang berjalan. Selain itu juga akan terlihat lebih rapih dan teratur ketika *keyword-keyword* yang diakan diberikan kebijakan terisolasi di dalam sebuah *file*.

Distributed filtering merupakan sebuah konsep pendistribusian aturan-aturan dari *proxy* server yang bisa didapat oleh *FirstClient* dari *Secondserver* melalui servis FTP. *Keyword-keyword* yang dituliskan kedalam sebuah *file* akan didistribusikan kepada *Secondserver*, sehingga pada *Secondserver* memiliki kemampuan untuk *synchronizing* aturan yang ada pada *Firstserver*. Keuntungan dari konsep ini adalah, ketika sebuah jaringan memiliki beberapa *proxy* yang berbeda untuk masing-masing divisi, maka akan sangat mudah dengan mengkonsentrasikan semua aturan dari *proxy* ke dalam sebuah server. Sehingga pendefinisian rule baru akan lebih mudah dan bisa digunakan dan diterapkan pada banyak *proxy* server dalam sekali waktu. Gambar 2.3 menjelaskan konsep *distributed filtering*.



Gambar 2.3 Konsep Distributed Filtering

Keterangan:

- 1 Ada dua buah jenis server Firstserver dan Secondserver. Firstserver menjadi *database* untuk menyimpan data *keyword-keyword* dan aturan yang akan didistribusikan kepada Secondserver.
- 2 Tidak semua Secondserver bisa mendapatkan *update* dan aturan dari Firstserver, Secondserver hanya bisa mendapatkan *update* jika sudah mendaftarkan diri dan mendapatkan kode autentikasi yang bisa didaftarkan melalui *website* dari Firstserver.

- 3 Jika kode autentikasi sudah didapatkan, maka *update* bisa dilakukan secara otomatis maupun manual.

2.6 FTP

FTP (*File Transfer Protocol*) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*File*) antar komputer dalam sebuah jaringan. FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan dan pengunggahan berkas komputer antara client FTP dan server FTP. Sebuah Client FTP merupakan aplikasi yang dapat menjalankan perintah-perintah FTP ke sebuah server FTP, sementara server FTP adalah sebuah *Windows Service* atau *daemon* yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah client FTP. Perintah-perintah FTP dapat digunakan untuk mengubah *directory*, mengubah modus transfer antara biner dan ASCII, *upload* berkas komputer ke server FTP, serta mengunduh berkas dari server FTP.

FTP menggunakan protokol *Transmission Control Protocol* (TCP) untuk komunikasi data antara client dan server, sehingga diantara kedua komponen tersebut akan dibuat sebuah sesi komunikasi sebelum transfer data dimulai. FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan *password* untuk mengakses, mengunduh dan mengunggah berkasnya.

Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa *directory*, sehingga mereka dapat membuat berkas, membuat *directory*, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode *anonymous login*, yaitu dengan menggunakan nama pengguna *anonymous* dan *password* kosong.

2.7 DHCP

Dynamic Host Configuration Protocol (*DCHP*) adalah protokol yang berbasis arsitektur client/server yang dipakai untuk memudahkan pengalokasian alamat IP dalam satu jaringan. Sebuah jaringan lokal yang tidak menggunakan DHCP harus memberikan alamat IP kepada semua komputer secara manual. Jika DHCP dipasang di jaringan lokal, maka semua komputer yang tersambung di jaringan akan mendapatkan alamat IP secara otomatis dari server DHCP. Selain alamat IP, banyak parameter jaringan yang dapat diberikan oleh DHCP, seperti *default gateway* dan DNS server.

DHCP merupakan sebuah protokol yang menggunakan arsitektur client/server, maka dalam DHCP terdapat dua pihak yang terlibat, yakni DHCP Server dan DHCP Client.

- DHCP Server merupakan sebuah mesin yang menjalankan layanan yang dapat menyewakan alamat IP dan informasi TCP/IP lainnya kepada semua client yang memintanya. Beberapa sistem operasi jaringan seperti Windows, Windows Server, atau GNU/Linux memiliki layanan ini.
- DHCP client merupakan mesin client yang menjalankan perangkat lunak client DHCP yang memungkinkan mereka untuk dapat berkomunikasi dengan DHCP. Hampir setiap sistem operasi memiliki perangkat lunak ini.

DHCP server umumnya memiliki sekumpulan alamat yang diizinkan untuk didistribusikan kepada client, yang disebut sebagai **DHCP Pool**. Setiap client kemudian akan menyewa alamat IP dari DHCP *Pool* ini untuk waktu yang ditentukan oleh DHCP. Ketika waktu penyewaan alamat IP tersebut habis, maka client akan meminta ulang kepada server untuk memberikan alamat IP yang baru atau memperpanjangnya.

Berikut ini adalah langkah-langkah yang dilakukan oleh DHCP Client ketika hendak melakukan peminjaman IP kepada DHCP Server.

1. DHCPDISCOVER: DHCP client akan menyebarkan request secara broadcast untuk mencari DHCP Server yang aktif.
2. DHCPOFFER: Setelah DHCP Server mendengar broadcast dari DHCP Client, DHCP server kemudian menawarkan sebuah alamat kepada DHCP client.
3. DHCPREQUEST: Client meminta DHCP server untuk menyewakan alamat IP dari salah satu alamat yang tersedia dalam DHCP Pool pada DHCP Server yang bersangkutan.
4. DHCPACK: DHCP server akan merespons permintaan dari client dengan mengirimkan paket acknowledgment. Kemudian, DHCP Server akan menetapkan sebuah alamat (dan konfigurasi TCP/IP lainnya) kepada client, dan memperbarui basis data miliknya. Client selanjutnya akan memulai proses *binding* dengan tumpukan protokol TCP/IP dan karena telah memiliki alamat IP, client pun dapat memulai komunikasi jaringan.

Empat tahap di atas hanya berlaku bagi client yang belum memiliki alamat. Untuk client yang sebelumnya pernah meminta alamat kepada DHCP server yang sama, hanya tahap 3 dan tahap 4 yang dilakukan, yakni tahap pembaruan alamat (*address renewal*).

BAB 3 METODOLOGI

3.1 Analisis Masalah

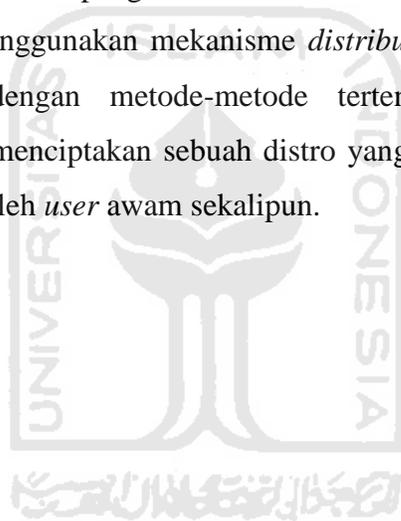
Inti dan tujuan utama dari perombakan ulang atau yang sering disebut dengan istilah *remastering* distro ini adalah untuk menciptakan sebuah distro baru yang sesuai dengan kebutuhan dan mudah untuk digunakan. Ketika kita hendak membutuhkan system operasi ini untuk kebutuhan lain, kita tidak perlu repot-repot untuk mengkonfigurasi ulang, karena semua sudah tersedia di distro baru tersebut.

Remastering merupakan solusi mudah untuk menciptakan atau merombak ulang distro agar bisa menjadi sebuah distro yang sesuai dengan tujuan *remastering*. *Remastering* menjadi sebuah langkah akhir untuk perombakan ulang distro. Sebelumnya harus disiapkan terlebih dahulu semua *tools* dan konfigurasi yang dibutuhkan.

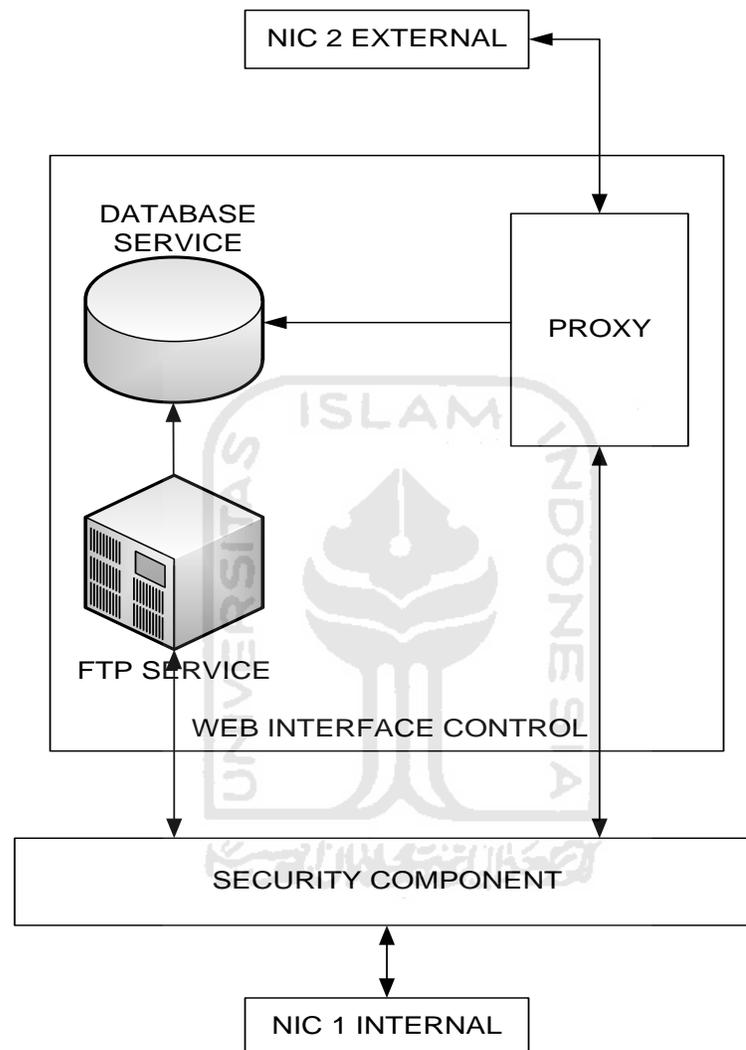
Remastering distro untuk kebutuhan kemudahan dalam membangun sebuah *gateway* yang aman memerlukan banyak sekali konfigurasi dan kebutuhan *tools* yang banyak. Setiap konfigurasi baik aplikasi maupun sistem, secara *default* dilakukan dengan *console*. Dengan aturan dan pengetahuan tertentu yang lebih jauh. Langkah ini tentu saja akan sangat tidak efisien untuk dilakukan oleh pengguna baru yang belum terlalu banyak mengetahui konsep Linux secara lebih mendalam. Untuk itu dibutuhkan sebuah *interface* yang digunakan untuk memudahkan semua konfigurasi. Interface tersebut akan lebih mudah digunakan dengan *web* base. Karena *web* sudah sangat familiar dan sangat mudah digunakan oleh orang awam. *Web interface* ini yang nantinya akan menjadi penengah atau jembatan bagi *user*. Melalui *web interface* ini *user* bisa mengkonfigurasi aplikasi dan system agar bisa berjalan sesuai dengan kebutuhan. Oleh karena itu dibutuhkan sebuah *web interface* yang mampu menghubungkan antara *user* dengan system Linux. Dalam *web interface* ini

akan banyak disertakan *script-script* php yang memanggil fungsi agar bisa menjalankan perintah dasar dari *console* Linux. Maka diperlukan juga versi php yang mampu mengakomodasi atau mendukung fungsi tersebut. Tentu saja dalam pembangunan *web interface* ini juga harus diperhatikan bagian keamanannya, karena salah satu tujuan dari distro ini adalah untuk kemananan.

Keamanan dalam sistem komputer merupakan sebuah hal yang sangat luas. Dalam sistem komputer kemanan juga identik dengan sebuah pembatasan terhadap akses tertentu baik dari dalam maupun dari luar. Dalam *remastering* distro baru ini kemanan akan difokuskan untuk pengaksesan dari dalam atau *internal*. Pembatasan akses dari *user* dengan menggunakan mekanisme *distributed filtering* dan imunitas anti serangan-serangan dengan metode-metode tertentu. Diharapkan dengan mekanisme ini, akan bisa menciptakan sebuah distro yang lebih handal namun bisa dengan mudah digunakan oleh *user* awam sekalipun.



3.2 Desain Alur Data dan Proses Aplikasi Sistem Operasi



Gambar 3.1 Desain Arsitektur Distro

Gambar 3.1 merupakan gambar yang menjelaskan gambaran secara umum kebutuhan, desain arsitektur serta konfigurasi yang akan disertakan kepada distro sebelum langkah *remastering*. Di dalam sistem tersebut akan disediakan sebuah *web interface* yang nantinya bisa digunakan untuk memudahkan dalam pengkonfigurasian

aplikasi-aplikasi yang tersedia di dalam distro. Aplikasi-aplikasi yang disertakan secara *default* dalam distro adalah *proxy*, *FTP*, *database* dan *web server*. Semua aplikasi-aplikasi tersebut dikonfigurasi supaya bisa saling bekerja sama dan berkomunikasi antar komponen dan menciptakan sebuah metode baru yaitu *distributed filtering*. *Proxy* yang akan digunakan adalah *transparent proxy*, agar semua client yang terdapat dalam jaringan *internal* distro Lancop akan otomatis menggunakan *proxy* tanpa pengaturan tertentu di sisi client.

Distro Lancop secara *default* menggunakan dua buah *interface card*. Masing-masing mempunyai fungsi tertentu, yaitu:

1. Eth0 : yaitu *network interface card* pertama yang akan digunakan sebagai jalur untuk koneksi dari *internal*.
2. Eth1 : yaitu *network interface card* pertama yang akan digunakan sebagai jalur untuk koneksi menuju jaringan yang ada diatas (*external*) dari jaringan *internal*.

Untuk menghubungkan agar komputer client bisa dikenal oleh jaringan *external*, maka dibutuhkan *firewall* yang berfungsi untuk masquerading paket yang berasal dari local (*internal*).

3.3 Desain Arsitektur Jaringan Implementasi Distro

Distro Lancop dirancang sebagai sebuah komputer gateway, yang digunakan oleh client yang berada dibawahnya sebagai pelayan untuk menghubungkan dengan network yang berada di atas jaringan tersebut. Lancop dilengkapi dengan aplikasi-aplikasi yang memungkinkan untuk menyaring akses koneksi yang melewati port 80. Authorisasi yang diberikan oleh *proxy* kepada setiap request yang melewati port 80 akan didasarkan pada *keyword* dari URL yang diberikan. *Keyword-keyword* tersebut disimpan di dalam aplikasi *database* yang menggunakan *mysql*. Dari *database*

tersebut kemudian di ubah menjadi *file* txt yang kemudian bisa ditransaksikan kepada secondserver melalui aplikasi FTP.

3.4 Analisis dan Persiapan Kebutuhan Sistem

3.4.1 Kebutuhan Perangkat Keras

Berdasarkan gambar 3.2 dan gambaran umum sistem, sudah jelas tergambar bahwa perangkat-perangkat keras yang dibutuhkan untuk menunjang implementasi dari semua fitur-fitur yang ada pada distro Lancop adalah sebagai berikut:

1. Komputer sebagai Firstserver

Komputer ini merupakan komputer yang dikhususkan sebagai gudang *keyword-keyword* yang digunakan oleh *proxy* untuk mengauthorize setiap koneksi dari client yang menggunakan port 80.

Spesifikasi minimal yang dibutuhkan adalah:

- a. Prosesor 1.6 GHz
- b. Memori 256
- c. Harddisk dengan kapasitas 5 GB
- d. 1 buah Network Interface Card

2. Komputer sebagai Secondserver

Komputer ini merupakan komputer yang mampu untuk meng-*update* baik manual maupun otomatis *keyword-keyword* dari first server. Komputer ini juga merupakan gateway untuk client yang ada dibawahnya.

- a. Prosesor 1.6 GHz
- b. Memory 512
- c. Hardisk dengan kapasitas 6 GB
- d. 2 buah Network Interface Card

3. Computer sebagai client

Komputer yang digunakan untuk pengujian.

- a. Prosesor 1.6 GHz

- b. Memory 512
- c. Hardisk dengan kapasitas 6 GB
- d. 1 buah Network Interface Card

3.4.2 Kebutuhan Perangkat Lunak

Pada implementasi dan pembuatan distro Lancop, perangkat lunak yang dibutuhkan adalah:

1. Remastersys

Remastersys adalah sebuah *software* untuk sistem operasi ubuntu/kubuntu dan variannya. Remastersys merupakan *tools* yang terinspirasi dari fungsi *mklive Script* pada distro Mandriva yang dapat digunakan untuk membackup system. Pada awalnya remastersys diciptakan untuk kepentingan backup data dari sebuah sistem yang *ter-install*. Setelah mengalami berbagai kemajuan, akhirnya tools ini dilengkapi dengan backup ke dalam bentuk file format *.iso* yang kemudian bisa di *boot* sehingga secara otomatis dapat menciptakan sebuah file yang *bootable*. Ada dua hal menarik yang dapat dilakukan remastersys pada distro linux debian based.

- Membuat *full sistem backup* termasuk personal data ke live cd/dvd yang dapat digunakan di manapun dan dapat di-*install* (mirip ghost pada windows)
- Dapat digunakan untuk membuat salinan dari sistem anda untuk dibagikan kepada rekan-rekan. Tapi tanpa ada data dari *user* sebelumnya.

Aplikasi ini merupakan proses *remastering* yang dalam penggunaannya masih kurang *user friendly*. Hal ini dikarenakan setiap proses perubahan tidak semuanya dilayani dengan service GUI, namun terdapat beberapa proses yang harus dilakukan melalui *console*. Setelah semua *customization* dilakukan, maka semua perubahan itu akan disimpan dan kemudian di-*export* ke dalam file berformat *.iso bootable* yang kemudian bisa langsung dituliskan ke dalam

cd ataupun flashdisk. Sehingga memungkinkan untuk membuat sebuah distro baru yang *distributable*.

2. Apache2

Apache adalah sebuah aplikasi yang digunakan untuk membangun sebuah *web* server. Protocol yang digunakan untuk melayani fasilitas web ini menggunakan HTTP. *Web* server berfungsi menerima permintaan HTTP atau HTTPS dari client yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML. Apache merupakan perangkat lunak *open source* yang dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang di bawah naungan *Apache Software Foundation*.

3. ProFTPd

ProFTPd adalah aplikasi berbasis Linux yang digunakan untuk membangun sebuah FTP server. FTP server merupakan sebuah tipe server yang memberikan layanan berupa *file* transfer yang bisa dilakukan dari client ke server. FTP server bekerja dengan menggunakan protokol FTP (*File Transfer Protokol*).

4. FTPsync

FTPsync merupakan sebuah aplikasi sederhana yang dibangun dengan menggunakan bahasa perl yang berfungsi untuk men-*synchronize* data melalui protokol FTP.

5. Mysql

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL. MySQL berfungsi sebagai media penyimpanan data yang bersifat dinamis dan disimpan dalam tabel-tabel yang tersedia di dalamnya. MySQL merupakan implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). SQL (Structured Query Language) adalah sebuah konsep pengoperasian

basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

6. Squid

Squid adalah sebuah aplikasi yang digunakan sebagai *proxy* server dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat server *web* dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, *caching* situs *web*, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Squid umumnya dirancang untuk berjalan diatas sistem operasi yang berbasis UNIX.

7. Dhcpd

Dhcpd merupakan aplikasi berbasis Linux yang digunakan untuk membangun sebuah dhcp server. Dhc server merupakan sebuah server yang memberikan layanan berupa peminjaman IP secara dinamis kepada client-client yang membutuhkan IP untuk berkomunikasi dalam sebuah jaringan.

3.5 Instalasi dan Konfigurasi Sistem

Semua kebutuhan perangkat lunak di-*install* pada mesin dengan metode yang berbeda-beda. Secara umum paket *installer* adalah berbentuk *binary* dan *source*. Pemilihan dua jenis paket tersebut akan didasarkan pada kebutuhan sistem.

Berikut adalah detail dari paket, instalasi dan konfigurasi secara umum:

Perangkat-perangkat lunak akan disertakan pada dua buah mesin yang berbeda, yaitu Lancop Firstserver dan Lancop Secondserver. Berikut adalah detail pembagian perangkat-perangkat lunak pada masing-masing mesin:

- Apache
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan perintah *apt*.
 - Konfigurasi yang dilakukan adalah terkait dengan penentuan lokasi file web yang akan ditampilkan serta perubahan pada jenis *user* yang diberikan oleh layanan Apache.
- ProFTPD
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan perintah *apt*.
 - Konfigurasi yang dilakukan adalah terkait penentuan lokasi *directory* tempat penyimpanan file yang akan diunduh serta konfigurasi pengintegrasian dengan aplikasi basis data MySQL.
- MySQL
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan *apt*.
 - Konfigurasi yang dilakukan adalah terkait dengan penentuan *user* dan password yang dilayani oleh aplikasi MySQL.
- Squid
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan paket source yang bisa diunduh di Internet. Dalam instalasi paket source akan dibutuhkan beberapa cara khusus, yaitu dengan menggunakan perintah
 - *./configure* (dengan tambahan parameter)
 - *Make*
 - *Make install*
 - Konfigurasi yang dilakukan adalah terkait dengan
 - *Port* yang digunakan untuk berhubungan dengan client
 - *user* dan group pengelola aplikasi

- penentuan ukuran dan lokasi *cache*
 - metode penyimpanan *cache*
 - hak akses terhadap *request* dari client
 - ukuran maksimal dan minimal objek yang akan disimpan pada *cache*
 - Penentuan ukuran *memory* yang digunakan oleh aplikasi
 - Penentuan nama *hostname* aplikasi
- Dhcpcd
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan apt.
 - Konfigurasi yang dilakukan adalah terkait dengan pembuatan file sh sebagai file yang digunakan untuk pengaturan konfigurasi server melalui *web interface*.
 - Remastersys
 - Jenis paket yang digunakan adalah tipe binary yang diinstal dengan menggunakan apt, namun dengan lokasi tujuan repository yang khusus disediakan oleh remastersys, yaitu pada deb <http://www.geekconnection.org/remastersys/repository> karmic/.
 - Konfigurasi yang dilakukan adalah terkait dengan penentuan nama dan *directory* kerja dari hasil remastering.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Implementasi Secara Umum

Implementasi sistem merupakan tahap dimana sistem mampu diaplikasikan dalam keadaan yang sesungguhnya. Dari implementasi ini akan diketahui apakah sistem yang dibuat dapat berjalan dengan baik atau tidak. Serta apakah sistem menghasilkan *output* yang sesuai dengan perancangan yang telah dibuat.

4.2 Tahapan Implementasi Perangkat Lunak

Dalam pengaplikasian perangkat lunak ini, terdapat beberapa tahap yang terdiri dari :

1. Tahap instalasi dan konfigurasi FirstServer
 - a. Instalasi dan konfigurasi proxy

Pada tahap ini dilakukan instalasi proxy server dengan menggunakan aplikasi Squid 3.0.
 - b. Instalasi dan konfigurasi Database SQL

Pada tahap ini dilakukan instalasi Database SQL yang digunakan untuk menyimpan rule-rule dari proxy server. Aplikasi yang digunakan adalah MySql.
 - c. Instalasi dan konfigurasi FTP

Pada tahap ini dilakukan instalasi FTP server yang digunakan untuk FirstServer dan ClientServer untuk saling bertransaksi rule-rule proxy server.
 - d. Instalasi dan konfigurasi web server

Pada tahap ini dilakukan instalasi web server yang digunakan untuk menampilkan halaman interface web agar lebih mudah digunakan.

- e. Pembuatan *Script* penghubung antar aplikasi

Pada tahap ini dilakukan pembuatan *Script* dengan menggunakan bahasa shell dan perl yang digunakan untuk saling menghubungkan antar aplikasi agar bisa saling bekerja sama.

- f. Pembuatan interface web untuk konfigurasi server

Pada tahap ini dilakukan pembuatan web yang berfungsi untuk mempermudah pengaturan pada server.

- g. Instalasi Reconstructor dan pembuatan distro baru

Pada tahap ini dilakukan instalasi aplikasi Reconstructor yang berfungsi untuk pembuatan file image baru dari linux yang sudah disiapkan.

2. Tahap instalasi SecondServer

- a. Instalasi dan konfigurasi proxy

Pada tahap ini dilakukan instalasi proxy server dengan menggunakan aplikasi Squid 3.0.

- b. Instalasi dan konfigurasi Database SQL

Pada tahap ini dilakukan instalasi Database SQL yang digunakan untuk menyimpan rule-rule dari proxy server. Aplikasi yang digunakan adalah MySql.

- c. Pembuatan aplikasi sinkronisasi

Pada tahap ini dilakukan pembuatan aplikasi sederhana dengan menggunakan bahasa perl yang digunakan untuk *me-request update* berupa keyword dari FirstServer.

- d. Instalasi dan konfigurasi web server

Pada tahap ini dilakukan instalasi web server yang digunakan untuk menampilkan halaman interface web agar lebih mudah digunakan.

e. Pembuatan *Script* penghubung antar aplikasi

Pada tahap ini dilakukan pembuatan *Script* dengan menggunakan bahasa shell dan perl yang digunakan untuk saling menghubungkan antar aplikasi agar bisa saling bekerja sama.

f. Pembuatan interface web untuk konfigurasi server

Pada tahap ini dilakukan pembuatan web yang berfungsi untuk mempermudah pengaturan pada server.

g. Instalasi Reconstructor dan pembuatan distro baru

Pada tahap ini dilakukan instalasi aplikasi reconstructor yang berfungsi untuk pembuatan file image baru dari linux yang sudah disiapkan.

4.3 Implementasi Hasil Perancangan

4.3.1 Instalasi dan konfigurasi squid 3.0

1. Dalam tahap ini dibutuhkan installer dari squid 3.0 yang bisa diunduh dari situs www.squid-cache.com secara gratis.
2. Lakukan instalasi paket yang sudah didapat dengan cara:

```
Ekstrak file installer
#tar -xvzf squid-3.0.STABLE25.tar.gz
#cd squid-3.0.STABLE25

Compile file hasil ekstraksi
#./configure -prefix=/usr/local/squid -enable-dl-malloc -
enable-snmp -enable-kil-parent-hack

Proses installasi
#make
#make install
```

3. Kemudian lakukan konfigurasi proxy dengan mengubah file yang ada pada `/usr/local/squid/etc/squid.conf`

```
#Konfigurasi untuk autentikasi
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours

#pendefinisian alamat yang bisa menggunakan proxy
```

```

acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16a
acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # FTP
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl CONNECT method CONNECT

#pendefinisian hak akses dan access list
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access allow localnet
#====aturan====#
acl blok URL_regex -i "/usr/local/squid/getacl/getaclsql.txt"
http_access deny blok
http_access allow all
icp_access allow localnet
icp_access deny all
htcp_access allow localnet
htcp_access deny all

#pendefinisian port yang digunakan oleh squid
http_port 3120

#pendefinisian memori dan algoritma replacement
cache_mem 64 MB
memory_replacement_policy lru
cache_replacement_policy lru
cache_dir ufs /usr/local/squid/var/cache 100 16 256

#pendefinisian ukuran object maksimum dan minimum
minimum_object_size 0 KB
maximum_object_size 4096 KB

#pendefinisian penyimpanan file log
access_log /usr/local/squid/var/logs/access.log squid
cache_log /usr/local/squid/var/logs/cache.log
cache_store_log /usr/local/squid/var/logs/store.log

```

```
#pendefinisian user dan group squid
cache_effective_user squid
cache_effective_group squid
visible_hostname Lancop.proxy
109 log_icp_queries on

#pendefinisian file error handling
error_directory /usr/local/squid/share/errors/templates
```

4. Buat file pada directory `/usr/local/squid/policies/update`. File ini adalah file yang digunakan sebagai rujukan untuk menyimpan keyword-keyword URL yang akan diberikan policies oleh squid.

```
# nano /usr/local/squid/policies/update
Isikan dengan kata "sex".
```

5. Sesuaikan hak akses pada masing2 file konfigurasi squid agar bisa dijalankan oleh *user* squid.

```
##pengubahan kepemilikan dan hak akses terhadap direktori.
#chown root:root /usr/local/squid/
#chmod 755 /usr/local/squid/
#cd /usr/local/squid/bin
#chown root:root . chown root:root *
#chmod 755 . *
#cd /usr/local/squid/sbin
#chown root:root . chown root:root *
#chmod 755 . *
#cd /usr/local/squid/etc
#chmod 2775 .
#chown root:squidadm . *
#chown squid:squidadm /usr/local/squid/var/logs
#chmod 770 /usr/local/squid/var/logs

###pembuatan dan pengubahan kepemilikan serta hak akses.
#mkdir /usr/local/squid/var/cache/
#chown squid:squidadm /usr/local/squid/var/cache/
#chmod 770 /usr/local/squid/var/cache/
```

4.3.2 Instalasi dan konfigurasi Apache dan php

1. Instalasi Apache dan php

```
# apt-get install Apache2 php5
```

2. Mengubah `DirectoryRoot`, arahkan ke alamat `/var/www/web/show`. `DocumentRoot` merupakan keyword yang digunakan untuk menjabarkan direktori tempat file web akan disimpan.

```
# nano /etc/Apache2/sites-available/default
Ubah
DocumentRoot /var/www
Menjadi
DocumentRoot /var/www/web/show
```

3. Ubah *user* yang digunakan oleh Apache. Secara *default*, ketika *user* menghubungi sebuah server melalui port 80 milik Apache, maka *user* tersebut akan diberikan hak akses dengan nama “`www-data`”. *User* ini kita ganti agar pengaturan hak aksesnya bisa disejajarkan dengan hak akses dari squid. Oleh karena itu kita ganti *user*nya dengan squid.

```
# nano /etc/Apache2/envvars
Ubah baris
export APACHE_RUN_USER=www-data
menjadi
export APACHE_RUN_USER=squid
```

4.3.3 Instal dan konfigurasi FTP

1. Instalasi FTP

```
# apt-get install proFTPd
Pilih standalone
```

2. Kondisikan agar *user* dan password yang digunakan oleh proFTPd sesuai dengan database mysql, dengan mengubah konfigurasi pada file `/etc/proFTPd/proFTPd.conf`, `/etc/proFTPd/modules.conf`, dan `/etc/proFTPd/sql.conf`.

/etc/proFTPd/proFTPd.conf

```
Include /etc/proFTPd/modules.conf
UseIPv6                                on
IdentLookups                            off
ServerName                              "Debian"
ServerType                              standalone
DeferWelcome                            off
MultilineRFC2228                        on
DefaultServer                           on
ShowSymlinks                            on

TimeoutNoTransfer                        600
TimeoutStalled                          600
```

```

TimeoutIdle          1200

DisplayLogin          welcome.msg
DisplayChdir          .message true
ListOptions           "-l"

DenyFilter            \*.*

#konfigurasi untuk menentukan lokasi direktori untuk user
pengguna FTP
DefaultRoot           /up2date
RequireValidShell    off
##konfigurasi port yang digunakan oleh FTP
Port                  21
MaxInstances          30

##konfigurasi untuk menentukan user dan group administrasi
User                  FTPuser
Group                 FTPgroup

Umask                 022 022
AllowOverwrite        on
TransferLog           /var/log/proFTPD/xferlog
SystemLog             /var/log/proFTPD/proFTPD.log
<IfModule mod_quotatab.c>
QuotaEngine off
</IfModule>

<IfModule mod_ratio.c>
Ratios off
</IfModule>

<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
ControlsEngine        off
ControlsMaxClients    2
ControlsLog           /var/log/proFTPD/controls.log
ControlsInterval      5
ControlsSocket        /var/run/proFTPD/proFTPD.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine  off
</IfModule>

##konfigurasi untuk mengintegrasikan FTP dengan aplikasi
MySQL
Include /etc/proFTPD/sql.conf

```

/etc/proFTPd/modules.conf

```

ModulePath /usr/lib/proFTPd
ModuleControlsACLs insmod,rmmod allow user root
ModuleControlsACLs lsmod allow user *

LoadModule mod_ctrls_admin.c
LoadModule mod_tls.c

##modul yang harus aktif untuk mengintegrasikan ProFTP
##dengan aplikasi MySQL
LoadModule mod_sql.c
LoadModule mod_sql_mysql.c

LoadModule mod_radius.c
LoadModule mod_quotatab.c
LoadModule mod_quotatab_file.c

LoadModule mod_quotatab_radius.c
LoadModule mod_wrap.c
LoadModule mod_rewrite.c
LoadModule mod_load.c
LoadModule mod_ban.c
LoadModule mod_wrap2.c
LoadModule mod_wrap2_file.c
LoadModule mod_dynmasq.c
LoadModule mod_vroot.c

LoadModule mod_ifsession.c

```

/etc/proFTPd/sql.conf

```

###menentukan tipe database yang digunakan
SQLBackend                mysql

##menentukan nama user, password dan nama database yang
##digunakan untuk menyimpan data user FTP
SQLConnectInfo  FTP@localhost Lancop abcdef

##menentukan nama field dari tabel pertama yang terdapat
##pada database yang digunakan
SQLUserInfo     FTPuser userid passwd uid gid homedir shell

##menentukan nama field dari tabel kedua yang terdapat pada
##database yang digunakan
SQLGroupInfo    FTPgroup groupname gid members

##konfigurasi untuk menghitung jumlah login user FTP
Update count every time user logs in

```

```

SQLLog PASS updatecount
SQLNamedQuery updatecount UPDATE "count=count+1, 2
accessed=now() WHERE userid='%u'" FTPuser

# Update modified everytime user uploads or deletes a file
SQLLog STOR,DELE modified
SQLNamedQuery modified UPDATE "modified=now() WHERE
userid='%u'" FTPuser

```

4.3.4 Install dan konfigurasi MySQL dan PHPMyAdmin

1. Instalasi mysql dan php

```
# apt-get install mysql-server phpmyadmin
```

2. Buat database untuk penyimpanan data keyword-keyword dari squid

```

# CREATE TABLE `data`.`URL_keyword` (
`id_URL` INT NOT NULL AUTO_INCREMENT PRIMARY KEY ,
`keyword` INT NOT NULL ,
`keterangan` INT NOT NULL
) ENGINE = MYISAM ;

```

Keterangan:

Script ini digunakan untuk membuat sebuah database dengan nama data, yang di dalamnya terdapat table URL_keyword. Di dalam tabel tersebut terdapat tiga field, yaitu id_URL, keyword dan keterangan.

3. Buat database untuk penyimpanan data user dan password FTP

```

#mysql -u root
create database FTP;
GRANT SELECT, INSERT, UPDATE, DELETE ON FTP.* TO
'proFTPd'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;

USE FTP;

CREATE TABLE FTPgroup (
groupname varchar(16) NOT NULL default '',
gid smallint(6) NOT NULL default '5500',
members varchar(16) NOT NULL default '',
KEY groupname (groupname)
) TYPE=MYISAM COMMENT='ProFTP group table';

CREATE TABLE FTPuser (
id int(10) unsigned NOT NULL auto_increment,
userid varchar(32) NOT NULL default '',
passwd varchar(32) NOT NULL default '',
uid smallint(6) NOT NULL default '5500',

```

```

gid      smallint(6)      NOT      NULL      default  '5500',
homedir  varchar(255)          NOT      NULL      default  '',
shell    varchar(16)     NOT      NULL      default  '/sbin/nologin',
count    int(11)             NOT      NULL      default  '0',
accessed datetime       NOT      NULL      default  '0000-00-00 00:00:00',
modified datetime       NOT      NULL      default  '0000-00-00 00:00:00',
PRIMARY KEY (id),
UNIQUE KEY (userid)
) TYPE=MyISAM COMMENT='ProFTP user table';

```

Keterangan:

Script ini digunakan untuk membuat sebuah database dengan nama FTP yang di dalamnya terdapat dua buah tabel yaitu *FTPuser* dan *FTPgroup*. Berikut rincian field dari masing-masing tabel.

- *FTPuser* : id, *userid*, *passwd*, *uid*, *gid*, *homedir*, *shell*, *count*.
- *FTPgroup* : *groupname*, *gid*, *members*

4.3.5 Instalasi dhcp server

1. Install dhcp server

```
#apt-get install dhcp3-server
```

2. Konfigurasi dhcp server melalui file /etc/dhcp3/dhcpd.conf

```

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.2 192.168.1.30;
#option domain-name-servers ns1.internal.example.org;
#option domain-name "internal.example.org";
option routers 192.168.1.1;
option broadcast-address 192.168.1.255;
default-lease-time 600;
max-lease-time 7200;
}

```

Keterangan:

- Baris 1 : menentukan alamat network dan netmask yang digunakan oleh server dan client.
- Baris 2 : menentukan jangkauan IP yang akan dipinjamkan atau disewakan kepada client.
- Baris 5 : menentukan alamat gateway yang akan disewakan kepada client.
- Baris 7 : menentukan waktu minimal peminjaman IP.
- Baris 8 : menentukan waktu maksimal peminjaman IP.

4.3.6 Pembuatan *Script* bash untuk menghubungkan antar aplikasi

1. Membuat file *Script* inc.conf

```
# nano /usr/local/squid/getacl/inc/inc.conf
Lancop      data  abcdef
```

Keterangan:

Script ini digunakan untuk mendefinisikan nama database, *user* dan password yang akan digunakan oleh file *getacl.sh* untuk mengambil data dari database MySQL.

Membuat file *Script* getacl.sh

```
# nano /usr/local/squid/getacl/getacl.sh

#!/bin/sh
user=`cut -f 1 /usr/local/squid/getacl/inc/inc.conf`
dbase=`cut -f 2 /usr/local/squid/getacl/inc/inc.conf`
pass=`cut -f 3 /usr/local/squid/getacl/inc/inc.conf`

mysql -u $user -p $dbase --password=$pass --execute="SELECT
*      FROM      URL_keyword      ORDER      BY      id_URL" >
/usr/local/squid/getacl/getaclsql1.txt
sed -n 2,1000p /usr/local/squid/getacl/getaclsql1.txt | cut
-f2 > /usr/local/squid/getacl/getaclsql.txt
cp      -f      /usr/local/squid/getacl/getaclsql.txt
/up2date/update/update
```

Keterangan:

Script ini digunakan untuk mengambil data keyword dari database MySQL

2. Membuat *Script* koneksi agar web interface bisa mengubah file /etc/dhcp3/dhcpd.conf

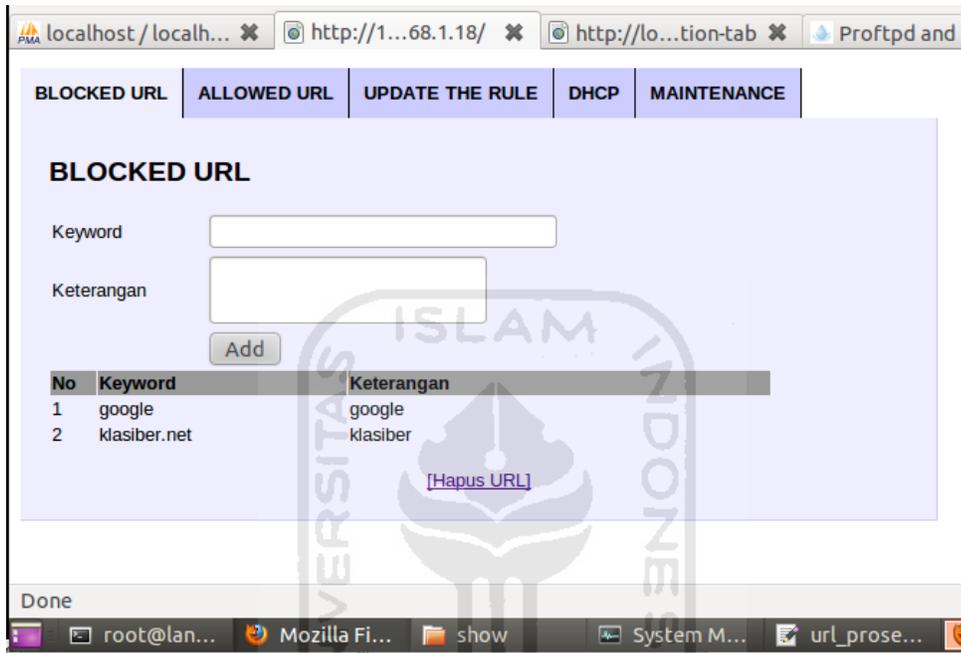
```
#nano /etc/dhcp3/dhcp.sh

Echo `#!/bin/bash
subnet $1 netmask $2 {
range $3 $4;
#option domain-name-servers ns1.internal.example.org;
#option domain-name "internal.example.org";
option routers 192.168.1.1;
#option broadcast-address 192.168.1.255;
default-lease-time 600;
max-lease-time 7200;
}` > /etc/dhcp3/dhcpd.conf
```

Keterangan:

Script ini digunakan untuk mengatur agar dhcp server bisa dikonfigurasi melalui *web interface*.

4.3.7 Membuat web interface



Gambar 4.1 web interface Lancop

1. Untuk mempermudah dalam pengaturan service server maka, disediakan web interface yang sangat *user friendly*

File koneksi `inc_connect.php`

```
<?php
$my['host']      = "localhost";
$my['user']     = "secondserver";
$my['pass']     = "abcdef";
$my['dbase']    = "data_second";

$koneksi       = mysql_connect($my['host'], $my['user'],
$my['pass']);
if (! $koneksi) {
    echo "koneksi gagal";
    mysql_error();
}

mysql_select_db($my['dbase'])
    or die ("database tidak ada".mysql_error());
?>
```

File URL_add.php

```

        <form action="URL_proses.php?mode=input" method="post">
        <table width=505 border=0 cellspacing=1 >
<?php
    echo "
    <tr>
        <td>Keyword</td>
        <td><input type='text' name='keyword' size=25
maxlength=25 /></td>
    </tr>
    <tr>
        <td>Keterangan</td>
        <td><textarea name='keterangan' cols='25'
rows='2'></textarea></td>
    </tr>
    <tr>
        <td></td>
        <td><input type=submit value=Add></td>
    </tr>
    </table>
    </form>";
?>
<table width=505 border=0 cellspacing=1>
<tr bgcolor="#999999">
    <td width=30><b>No</b></td>
    <td width=175><b>Keyword</b></td>
    <td width=300><b>Keterangan</b></td>
</tr>

<?php
include "../inc/inc_connect.php";

$sqlTampil = "SELECT * FROM URL_keyword order by id_URL";
$qryTampil = mysql_query($sqlTampil, $koneksi) or die
("query gagal".mysql_error());
$no = 1;
while($dataTampil=mysql_fetch_array($qryTampil))
{
?>
<tr>
    <td width=30><?php echo $no; ?></td>
    <td width=175><?php echo $dataTampil['keyword']; ?></td>
    <td width=200><?php echo $dataTampil['keterangan'];
?></td>
</tr>
<?php
$no++;
}
?>
</table>

```

```

<br/></br/>
<center><a href="http://localhost/index.php#usage-tab">
[Hapus URL] </a></center>

```

File URL_delete.php

```

<?
include "../inc/inc_connect.php";

$stampil=mysql_query("select * from URL_keyword");
$jumlah=mysql_num_rows($stampil);

if(!empty($jumlah))
{
    ?>
    <form action="URL_proses.php?mode=delete"
method="post">
    <table width="505" border="0" cellspacing="1"
align="center">
    <tr align=left bgcolor="#999999"><th></th><th><font
color="#FF0000">Delete</font>
URL</th><th>Keterangan</th></tr>
    <?
    $no=0;
    while($data=mysql_fetch_array($stampil)){
    echo "
    <tr>
    <td><input type=checkbox name=data[]
value=$data[id_URL] id=$no></td>";
    $no++;
    echo "<td>$data[keyword]</td>
    <td>$data[keterangan]</td>
    </tr>";
    }
    system('sh /usr/local/squid/getacl/getacl.sh');
    shell_exec("/usr/local/squid/sbin/squid -k
reconfigure");

    echo "
    <tr>
    <td colspan=2 align=center>
    <input type=radio name=pilih onClick='for
(i=0;i<$no;i++){document.getElementById(i).checked=true;}'>Ch
eck All
    <input type=radio name=pilih onClick='for
(i=0;i<$no;i++){document.getElementById(i).checked=false;}'>U
ncheck All
    </td>
    </tr>
    <tr>
    <td></td>

```

```

        <td><input type=submit value=Delete></td>
        <td></td>
    </tr>
</table>
</form>
<br/></br/>
<center><a
href='http://localhost/index.php#deDescription-tab'> [Tambah
URL] </a></center>";

}
else
{
    echo "Data URL terhapus SEMUA! Silahkan masukan lagi
datanya pada bagian <a
href=http://localhost/index.php#deDescription-tab>[Input
URL]</a>";
    system('sh /usr/local/squid/getacl/getacl.sh');
    shell_exec("/usr/local/squid/sbin/squid -k
reconfigure");
}
?>

```

File URL_proses.php

```

<?
include "../inc/inc_connect.php";

///tangkap nilai parameter
$data=$_POST['data'];

//mode delete atau input
$mode=$_GET['mode'];

if($mode=='delete'){
    //hitung jumlah yang di cek
    $jumlah=count($data);

    for($i=0;$i<$jumlah;$i++){
        $hapus=mysql_query("delete from URL_keyword where
id_URL='$data[$i]'");
        //untuk mengetahui nilai array :
        //echo $cek[$i];
    }
    echo " <meta http-equiv=\"refresh\"
content=\"0;URL=http://localhost/index.php#usage-tab\">";
}

else if($mode=='input')
{

```

```

        $data=mysql_query("insert into
URL_keyword(keyword, keterangan)
values('$ _POST[keyword]','$ _POST[keterangan]');
        system('sh /usr/local/squid/getacl/getacl.sh');
        shell_exec("/usr/local/squid/sbin/squid -k
reconfigure");
        echo "<meta http-equiv=\"refresh\"
content=\"0;URL=http://localhost/#deScription-tab\">";
    }
?>

```

4.3.8 Pengaturan Tampilan Lancop

Untuk kepentingan identitas dan kenyamanan dalam penggunaan maka sebaiknya jika tampilan desktop dan properti lainnya diubah sesuai dengan selera masing-masing pengguna. Berikut langkah-langkah untuk mengganti *desktop property*.

1. Desktop Customize

Agar setiap *user* yang terdapat di dalam sistem mempunyai tampilan yang sama secara *default*, maka file *profile* dari masing-masing *user* harus diganti. Berikut langkah-langkahnya:

```

Tekan alt+f2
#gksu nautilus

Tekan Ctrl+H
Pindahkan 4 file berikut ke dalam folder /etc/skel
.gconf
.gconfd
.config
.gnome2

```

2. Tampilan Welcome Screen

Merupakan tampilan yang akan dimunculkan pada saat seorang *user* hendak *login* ke dalam sistem.

- Perubahan *icon welcome screen*

```

Simpan file berformat .png dengan pixel kurang lebih
64x64 pada directory /usr/share/Icons/LoginIcon/apps/64

```

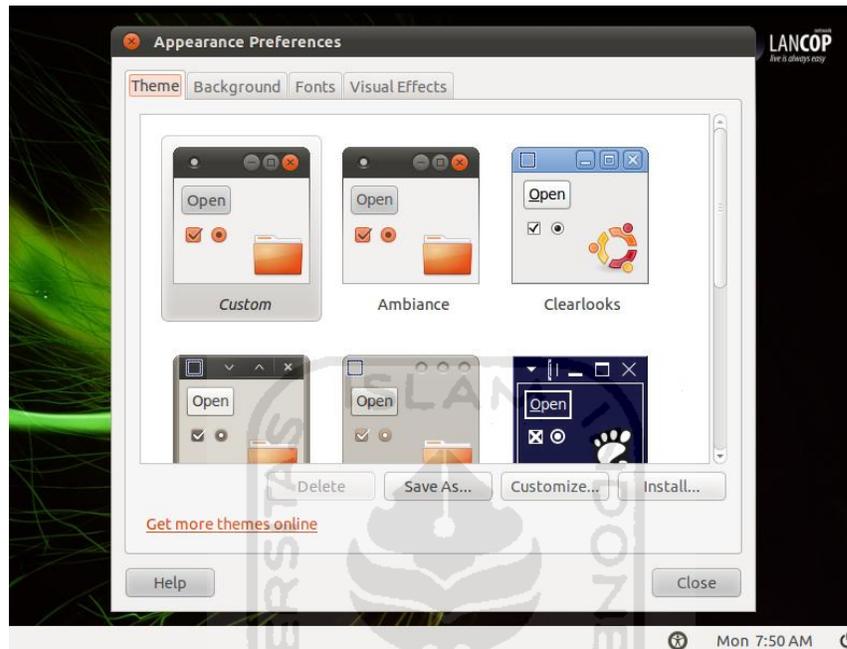
- Perubahan *background welcome screen*

```

#sudo cp /usr/share/applications/gnome-appearance-
properties.desktop
/usr/share/gdm/autostart/LoginWindow/

```

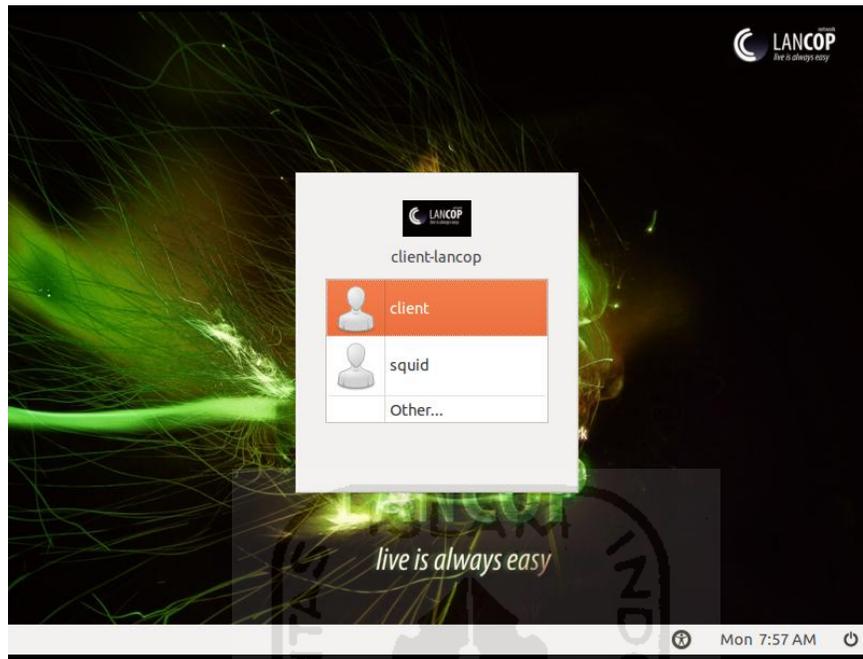
- *Logout*, muncul jendela *background property*, kemudian pilih *background*.



Gambar 4.2 custom background welcome screen

- *Login* ulang, kemudian jalankan perintah berikut

```
#sudo unlink
/usr/share/gdm/autostart/LoginWindow/gnome-
appearance-properties.desktop
```



Gambar 4.3 welcome screen

4.3.9 Instalasi Remastersys

1. Untuk penginstalan aplikasi remastersys dibutuhkan sebuah *link repository* khusus dari remastersys. Berikut langkah-langkah untuk menambahkan *repository* dan cara instalasi remastersys.

```
#sudo nano /etc/apt/sources.list

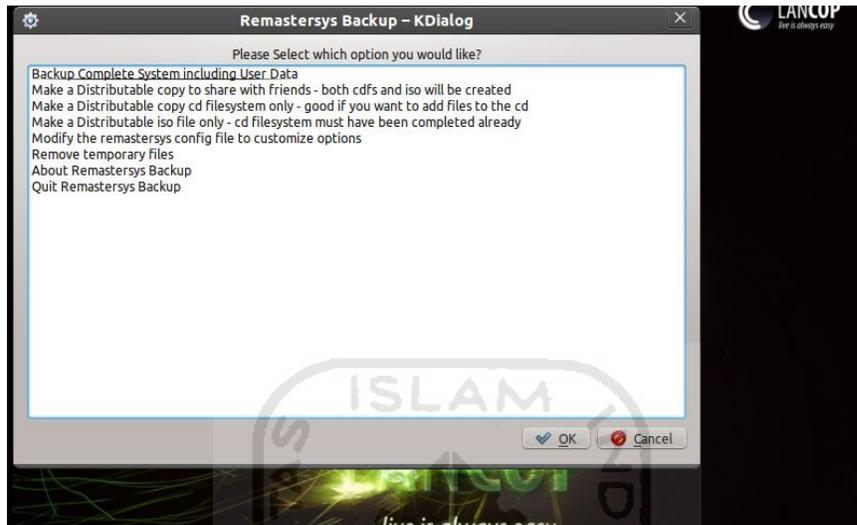
Tambahkan Script berikut pada baris paling bawah
# Remastersys
deb      http://www.geekconnection.org/remastersys/repository
karmic/

#kemudian lakukan update serta proses instalasi paket
#remastersys
sudo apt-get update
sudo apt-get install remastersys
```

2. Setelah proses instalasi selesai maka langkah selanjutnya adalah instalasi kubuntu-desktop. Hal ini dibutuhkan, karena remastersys secara *default* bisa menampilkan GUI pada *environment* KDE. Berikut langkah untuk instal KDE.

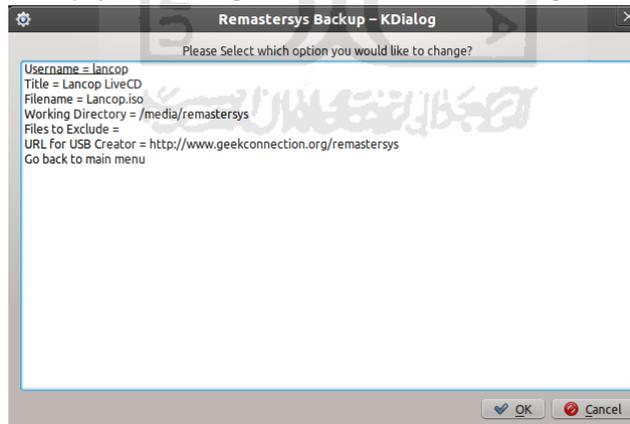
```
#sudo apt-get install kubuntu-desktop
```

3. Setelah instalasi selesai, jalankan aplikasi Remastersys dengan memilih menu *System > Administration > Remastersys*



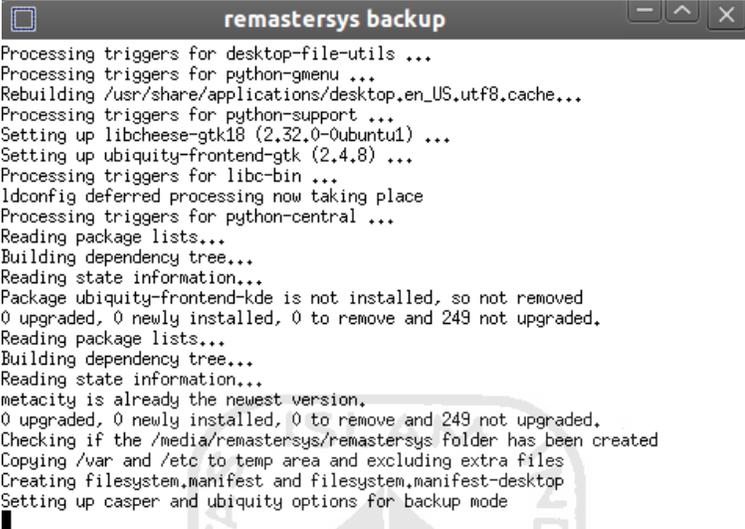
Gambar 4.4 homescreen reconstructor

4. Lakukan konfigurasi remastersys dengan memilih menu *modify the remastersys config*. Berikut gambar dan contoh konfigurasi



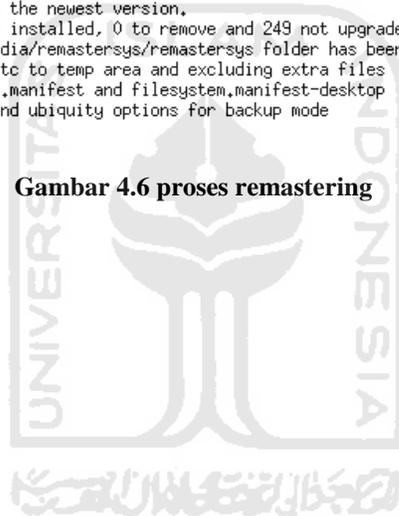
Gambar 4.5 Konfigurasi dasar remastersys

5. Kembali ke menu utama, kemudian pilih *backup complete system*, Kemudian proses remastering berjalan.



```
remastersys backup
Processing triggers for desktop-file-utils ...
Processing triggers for python-gmenu ...
Rebuilding /usr/share/applications/desktop.en_US.utf8.cache...
Processing triggers for python-support ...
Setting up libcheese-gtk18 (2.32.0-0ubuntu1) ...
Setting up ubiquity-frontend-gtk (2.4.8) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Processing triggers for python-central ...
Reading package lists...
Building dependency tree...
Reading state information...
Package ubiquity-frontend-kde is not installed, so not removed.
0 upgraded, 0 newly installed, 0 to remove and 249 not upgraded.
Reading package lists...
Building dependency tree...
Reading state information...
metacity is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 249 not upgraded.
Checking if the /media/remastersys/remastersys folder has been created
Copying /var and /etc to temp area and excluding extra files
Creating filesystem.manifest and filesystem.manifest-desktop
Setting up casper and ubiquity options for backup mode
```

Gambar 4.6 proses remastering



4.4 Pengujian Hasil Implementasi

Teknik remastering dengan menggunakan remastersys akan menghasilkan sebuah file *image* dengan format *.iso. File inilah yang nantinya harus dituliskan ke dalam *cd-rom* atau USB dan kemudian langsung bisa digunakan untuk *booting*. Setelah *file* tersebut ditulis kedalam media CD atau USB, maka media tersebut bisa digunakan untuk *booting*. Ketika *booting* melewati media ini, maka akan menghasilkan dua pilihan mode penggunaan distro ini

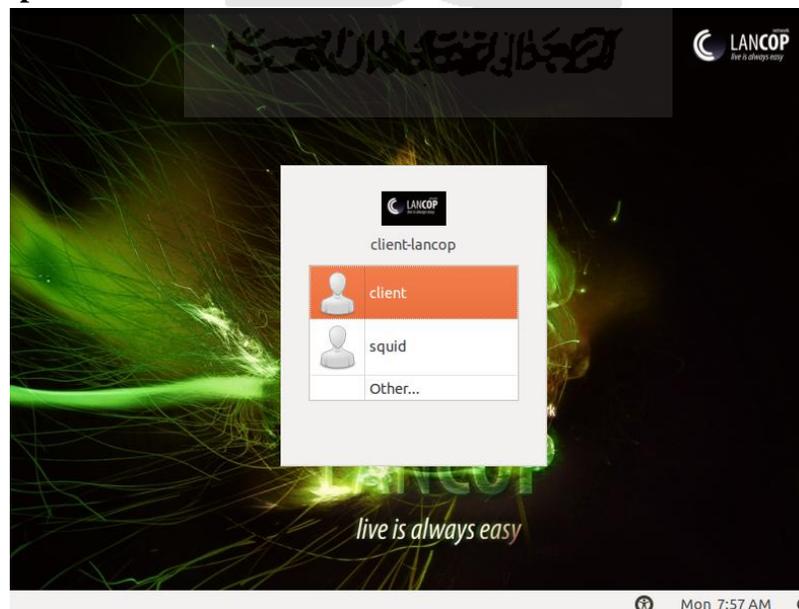
- Live

Dengan mode ini, maka kita hanya menggunakan sistem operasi sebatas *read only*. Mode ini hanya digunakan untuk pengujian dan percobaan sistem operasi tanpa instal. Perubahan konfigurasi dan file pada mode ini hanya akan bersifat sementara dan akan hilang ketika proses *boot* berikutnya.

- Install

Mode ini memungkinkan untuk menanam sistem informasi pada mesin. Sehingga perubahan akan bersifat permanen.

4.4.1 Tampilan Awal



Gambar 4.7 tampilan login



Gambar 4.8 tampilan desktop

4.4.2 Komponen aplikasi dan service sistem Distro

Distro hasil remaster akan secara langsung terinstal aplikasi-aplikasi yang dibutuhkan. Pengguna tidak perlu repot untuk melakukan instalasi dan konfigurasi *service* yang dibutuhkan dan langsung siap digunakan. Dengan menggunakan “ps aux” maka *user* bisa melihat servis dan aplikasi apa saja yang sudah terinstal dan siap untuk digunakan.

```
# ps aux
squid      1466  0.0  2.0  37488 10616 ?           S    Jan11
0:02 /usr/sbin/Apache2 -k start
squid      1476  0.0  1.4  34956  7600 ?           S    Jan11
0:08 /usr/sbin/Apache2 -k start
root       1635  0.0  0.4   13940  2504 ?           S    Jan09
0:01 /usr/bin/python /usr/lib/system-service/system-service-d
squid      1683  0.0  1.7   36312  8980 ?           S    Jan11
0:02 /usr/sbin/Apache2 -k start
FTPuser    1748  0.0  0.3   10104  1852 ?           Ss   Jan11
0:01 proFTPD: (accepting connections)
squid      1783  0.0  1.7   36232  9000 ?           S    Jan11
0:01 /usr/sbin/Apache2 -k start
root       2704  0.0  0.0    2300    252 ?           Ss   02:53
```

0:00	dhclient								
	squid	2792	0.0	1.7	36584	8988	?	S	04:15
0:01	/usr/sbin/Apache2 -k start								
	root	21000	0.0	0.0	6568	0	?	Ss	Jan09
0:00	/usr/local/squid/sbin/squid start								
	squid	21002	1.1	1.7	17020	8900	?	S	Jan09
41:56	(squid) start								
	squid	21004	0.0	0.0	3072	0	?	S	Jan09
0:00	(unlinkd)								
	root	29254	0.0	1.1	34308	5872	?	Ss	Jan11
0:16	/usr/sbin/Apache2 -k start								
	squid	29267	0.0	2.2	38576	11464	?	S	Jan11
0:09	/usr/sbin/Apache2 -k start								
	squid	29270	0.0	1.9	37440	10144	?	S	Jan11
0:08	/usr/sbin/Apache2 -k start								
	squid	29273	0.0	1.9	37132	10032	?	S	Jan11
0:08	/usr/sbin/Apache2 -k start								
	squid	29305	0.0	1.8	36176	9360	?	S	Jan11
0:43	/usr/sbin/Apache2 -k start								
	root	29508	0.0	0.1	2748	564	?	S<	Jan11
0:00	udev --daemon								
	squid	29558	0.0	2.2	38348	11404	?	S	Jan11
0:09	/usr/sbin/Apache2 -k start								
	mysql	29685	0.1	2.9	156196	14872	?	Ss1	Jan11
1:38	/usr/sbin/mysqld								
	root	29886	0.0	0.0	2748	396	?	S<	Jan11
0:00	udev --daemon								

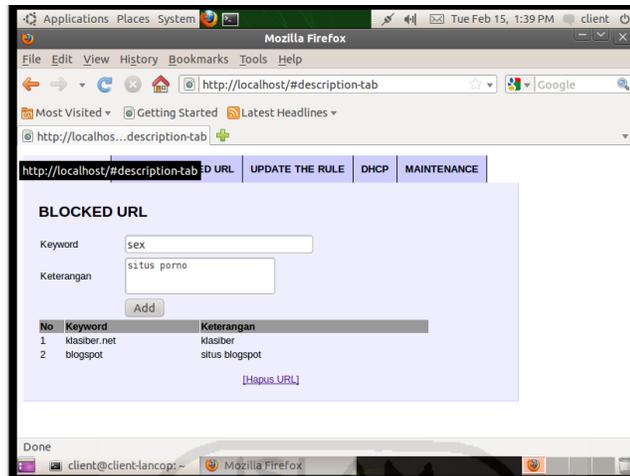
Keterangan:

Script tersebut memperlihatkan aplikasi-aplikasi yang sudah ada dan langsung berjalan tanpa instalasi sebelumnya.

4.4.3 Konfigurasi Via Web Interface

Secara *default* instalasi, konfigurasi dan penggunaan aplikasi server pada linux dilakukan dengan menggunakan *console*. Cara ini sangat tidak familiar untuk digunakan oleh pengguna awam. Dengan menggunakan distro Lancop, maka kebutuhan konfigurasi dan penggunaan aplikasi server bisa dilakukan dengan mudah melalui *web interface* yang sudah disiapkan. Cukup arahkan *browser* ke alamat <http://localhost>

- Menambahkan URL yang akan diblokir.

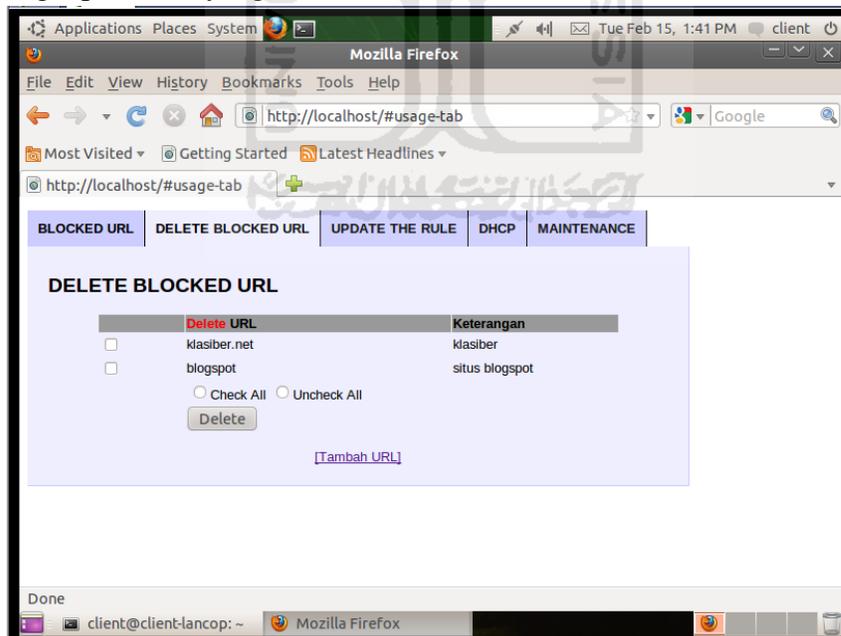


Gambar 4.9 menambah blocked URL

Keterangan:

Halaman ini digunakan untuk menambahkan situs-situs yang akan di blok berdasarkan keyword URL-nya

- Menghapus URL yang terblokir.

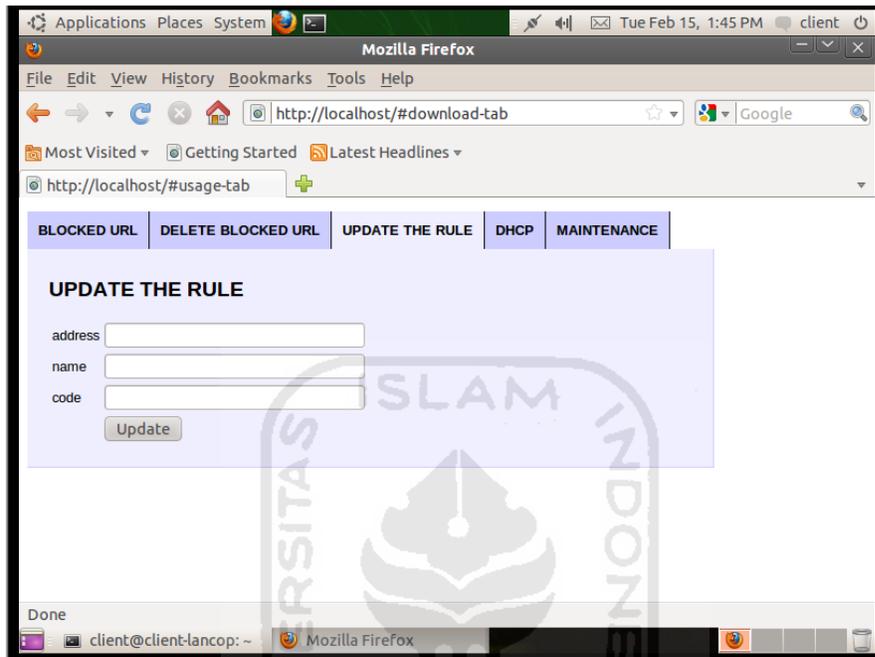


Gambar 4.10 menghapus URL yang terblokir

Keterangan:

Halaman ini digunakan untuk menghapus data situs-situs yang terblokir.

- Mengupdate data situs-situs baik yang diblokir maupun diperbolehkan dari FirstServer.

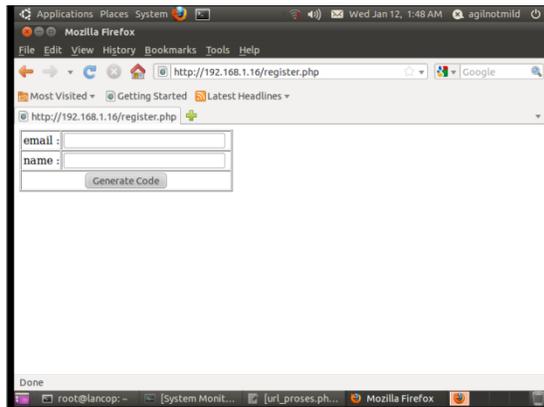


Gambar 4.11 update rule dari FirstServer

Keterangan:

Untuk mendapatkan *update* dari firstserver, maka kita harus terlebih dahulu mengetahui IP dari firstserver serta *name* dan *code* yang bisa didapatkan dengan cara mendaftarkan diri ke firstserver.

Untuk mendaftar ke firstserver, cukup dengan arahkan *browser* ke alamat http://alamat_IP_firstserver/register.php

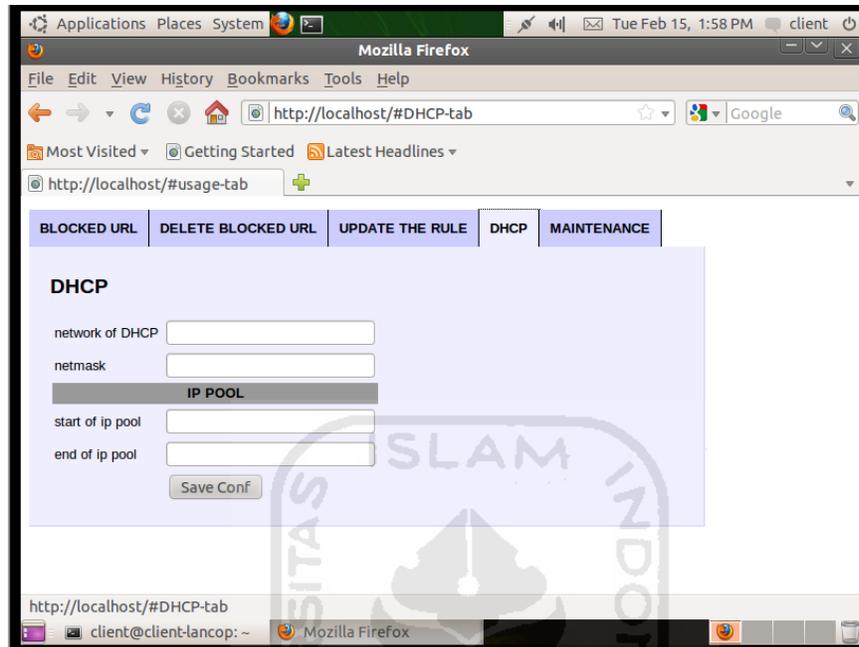


Gambar 4.12 pendaftaran untuk mendapat code update

Setelah mendapatkan *code*, maka masukkan *code* tersebut ke halaman *update*, kemudian tekan tombol *update*.



Konfigurasi DHCP server

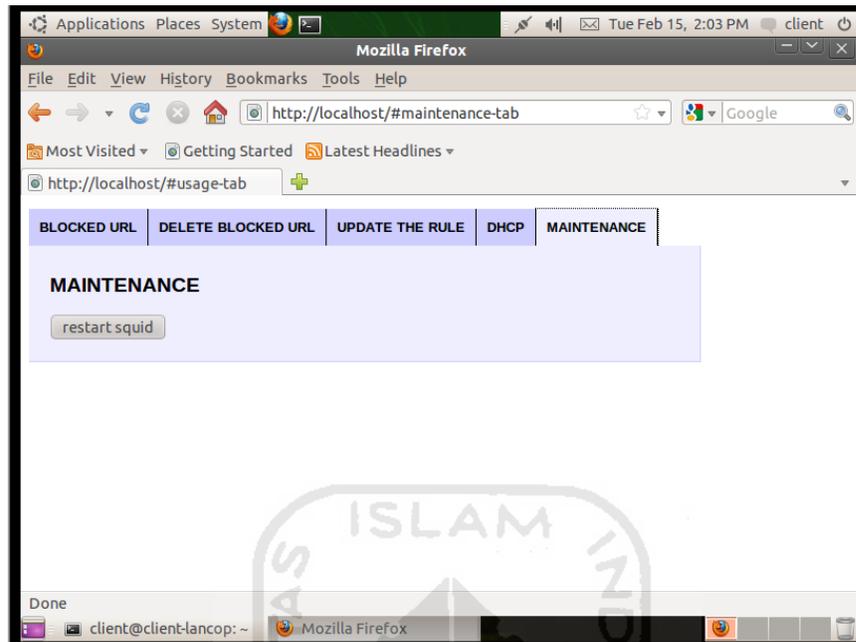


Gambar 4.13 konfigurasi dhcp

Keterangan:

Untuk mengatur konfigurasi dhcp server, cukup lakukan dengan memasukkan alamat *network*, *netmask*, *start and end* IP yang akan disewakan. Dengan halaman ini, pengguna tidak perlu lagi menggunakan *console* untuk pengaturan dhcp.

- Setiap perubahan pada servis-servis aplikasi, maka aplikasi perlu adanya aksi restart agar perubahan bisa terlihat. Maka dengan halaman ini, pengguna tidak perlu melakukan cara yang rumit untuk me-*restart* aplikasi.



Gambar 4.14 restart aplikasi

BAB 5

PENUTUP

5.1 Kesimpulan

Remastering distro Ubuntu untuk *gateway*, *distributed filter* dan *security* telah selesai dibangun dan diupayakan sesuai dengan perencanaan dan perancangan pada tahap analisis kebutuhan perangkat keras, perangkat lunak, desain dan alur data aplikasi serta desain jaringan implementasi. Setelah melakukan tahap pengujian. Maka dapat diambil kesimpulan bahwa:

1. Distro Lancop hasil remastering mampu mengakomodasi kebutuhan pembangunan jaringan *local* dengan cara yang sangat mudah dan aman.
2. *Web interface* sangat efektif digunakan untuk mempermudah penggunaan dan pengaturan aplikasi-aplikasi server tanpa membutuhkan terminal atau yang sering disebut dengan *console*. Sehingga pengguna tingkat awal pun bisa dengan mudah menggunakan distro Lancop.
3. Teknik remastering sangat efektif untuk membangun sebuah distro yang sesuai dengan kebutuhan.

5.2 Saran

Terdapat beberapa hal yang perlu diperhatikan ketika kita akan mengimplementasikan teknik remastering pada distro. Supaya langkah-langkah remastering bisa berjalan dengan baik, ada beberapa saran yang perlu diperhatikan, yaitu:

1. Pada penelitian selanjutnya, akan lebih baik untuk lebih meng-*explore* mengenai masalah *security* dan opsi dari aplikasi yang lebih kompleks.
2. Akan sangat lebih baik jika customisasi menyentuh lebih dalam sampai konsep *re-compile* kernel, sehingga menciptakan distro yang lebih handal.

3. Perancangan dan perencanaan harus lebih matang dengan tema yang sesuai dengan kebutuhan.
4. Remastering membutuhkan *resource hardware* yang tinggi, agar proses bisa berjalan lebih cepat.



DAFTAR PUSTAKA

Hil dan Katie. 2009. Geologi Rock. (On-line) available at <http://www.geologyrocks.co.uk/node/1032>

Pearson, Oskar. 2003. Squid, A User's Guide. (On-line) available at http://oss.org.cn/ossdocs/gnu_linux/squid/html/c777.html

Dodeimedo. 2006. Remastersys – Create custom Ubuntu (Live CD). (On-line) available at <http://www.dedoimedo.com/computers/remastersys.html>

