

**RANCANG BANGUN JARINGAN KOMPUTER
BERBASIS UBUNTU SERVER INTREPID IBEX 8.10
DI SMK NEGERI 6 SURABAYA**

PRAKTEK KERJA LAPANG



Diajukan Oleh:

MUHAMMAD AINUL YAQIN
NPM 0634015073

K e p a d a

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
Universitas Pembangunan Nasional "Veteran" Jawa Timur
S U R A B A Y A
2009**

LEMBAR PENGESAHAN

Judul PKL : RANCANG BANGUN JARINGAN KOMPUTER
BERBASIS UBUNTU SERVER INTREPID IBEX 8.10
DI SMK NEGERI 6 SURABAYA

Nama Mahasiswa : MUHAMMAD AINUL YAQIN

NPM : 0634015073

Program Studi : TEKNIK INFORMATIKA

Jurusan : TEKNIK INFORMATIKA

Menyetujui,
Pembimbing :

ACHMAD JUNAIDI, S.KOM
NPT. 278 110 440 199

Dekan
Fakultas Teknologi Industri

Ketua Jurusan
Teknik Informatika

Ir. BAMBANG WAHYUDI, MS

BASUKI RAHMAT, S.Si. MT

NIP. 030 180 480

NPT. 269 070 640

PRAKTEK KERJA LAPANGAN
RANCANG BANGUN JARINGAN KOMPUTER
BERBASIS UBUNTU SERVER INTREPID IBEX 8.10
DI SMK NEGERI 6 SURABAYA

Disusun Oleh :

MUHAMMAD AINUL YAQIN
NPM. 0634015073

telah dipertahankan dihadapan
dan diterima oleh Tim Penguji PKL
Jurusan Teknik Informatika Fakultas Teknologi Industri
Universitas Pembangunan Nasional “Veteran” Jawa Timur
pada 03 Agustus 2009

Susunan Tim Penguji :

1. Pembimbing :

ACHMAD JUNAIDI, S.KOM
NPT. 278 110 440 199

2. Anggota Tim Penguji

NUR CAHYO WIBOWO, S.KOM, M.KOM
NPT. 279 030 440 197

3. Anggota Tim Penguji

BUDI NUGROHO, S.KOM
NPT. 380 090 540 205

ABSTRAK

Teknik Informatika. Dimana dalam jurusan tersebut terdapat aktivitas jaringan komputer yang begitu kompleks didalamnya, karena selain digunakan oleh para guru dan siswa yang ada disana, tetapi juga banyak terdapat aktivitas dari para siswa, antara lain kegiatan di laboratorium, ruang osis yang terdapat pada SMK Negeri 6 Surabaya.

Permasalahan yang timbul pada praktek kerja lapang ini adalah pada SMK Negeri 6 Surabaya belum terstruktur dan belum ada nya jaringan komputer. Sehingga jaringan di dalamn sekolah tersebut masih terbagi-bagi sesuai kebutuhan. Misalnya saja jaringan yang berada pada laboratorium tidak terhubung dengan jaringan yang ada pada ruang tata usaha.

Dalam kata lain pada SMK Negeri 6 Surabaya memerlukan kebutuhan jaringan komputer yang terstruktur dan terdiri dari router berserta proxy server dan server file yang berbasis linux ubuntu sehingga bisa memudahkan kegiatan dan aktivitas proses belajar mengajar yang ada dalam sekolah tersebut.

Keyword : Linux, Router, Server, Ubuntu

KATA PENGANTAR

Pertama kali ijinkanlah kami mengucapkan puja dan puji syukur ke Hadirat Allah Yang Maha Esa atas selesainya laporan PKL ini. Karena atas berkat rahmat serta hidayah-Nya, kami dapat menyelesaikan laporan PKL ini yang telah menyita banyak waktu, tenaga, namun kami tetap diberi kesehatan dan semangat kerja yang tinggi. Amin.

Pada laporan ini, kami akan membahas tentang Rancang Bangun Jaringan Komputer Berbasis Ubuntu Server Intrepid Ibex 8.10 di SMK Negeri 6 Surabaya.

Penyusunan laporan ini tidak lepas dari bantuan serta kerja sama dari banyak pihak. Untuk itu dengan segala kerendahan hati kami mengucapkan terima kasih yang sebesar – besarnya kepada:

1. Kedua Orang Tua kami tercinta atas semua doa, dukungan serta harapan-harapanya pada saat kami menyelesaikan kerja praktek lapang dan laporan ini. Yang kami minta hanya doa restunya.s
2. Bapak Achmad Junaidi, S. Kom selaku dosen pembimbing PKL. Terima kasih atas semua bimbingannya selama kami menyelesaikan laporan pkl ini. Terima kasih juga atas semua sarannya yang telah diberikan selama bimbingan.
3. Bapak Ir. SUTIYONO, MT selaku Dekan Fakultas Teknologi Industri Universitas Pembangunan Nasional “Veteran” Jawa Timur.

4. Bpk. Nur Cahyo Wibowo, S. Kom, M. Kom dan Bpk. Budi Nugroho, S. Kom selaku dosen penguji PKL yang juga memberikan semangat dan motivasi kepada kami.
5. Dosen – Dosen Jurusan Teknik Informatika UPN “VETERAN” JATIM, yang telah membuat kami membuka pikiran dan merubah pola pikir kami.
6. Kawan – kawan dari KoLU, ILC, MIO, HIMATIFA, IKATIFA, INFORMANIA, BEM-FTI, BEM-UPN “VETERAN” JATIM, BLM-UPN “VETERAN” JATIM, PRING APUS, PRING JAWI, Annasrul Arifin S.Kom, Andi Baskoro S.Kom, Aswin Saputra S.Kom, Dena Chusnaul Ro’in, Sofia Dwi Rizani, Nina Maulidiya, Mersakti Riski Oktariani, Alfian Nugroho, Lahir S.Kom, Deddy Eko Setyawan S.Kom, Rizky Parlika S.Kom, Budi S.T, Sofyan Mubarak, Dhikri Sayeban, Adek Firly, Kresna Shandi Pranajaya, Norman Apriantono, Pak Minarno, Pak Wisnoe, Yusfi Helmi S.Kom, Om Agus Kantin, Psikologi UBAYA, Fasta, Megawati S.Kom, Ari Supriyono, Sena, Erik Sya’dul, Diki Lorenk, Rahadia S.Kom, Dravis S.Kom, Mahmed, Ngok, Ali Njager, Girindani, Ardianto, Frinda S.Kom, Antok, Warkop Depan Giri Loka, Meitha, Zubet, Indah Mangab, Lia TPI, Meheng, Wak Heri, Wak Ewot, Wak Suli, Wak Utom, K.H Heru, K.H Supendi S.H, Heri Ngok, Mas Kamsul, Bang Kodir, Nyambek, Gethok, Memet S.T, Jati, PM, Peri, feby, Pulek, Karang Taruna RT 03 Wiyung, Dhani Moxer, Nina06, dan semua teman-teman kami yang belum disebut, terima kasih atas segala bantuan dan dorongannya.

7. Keluarga Besar SMK Negeri 6 Surabaya, yang selalu memberikan semangat, dorongan serta motivasi yang membuat kami selalu optimis mengerjakan laporan PKL ini.

Kami sebagai manusia biasa pasti mempunyai keterbatasan dan banyak sekali kekurangan, terutama dalam pembuatan laporan ini. Untuk itu kami sangat membutuhkan kritik dan saran yang membangun dalam memperbaiki penulisan laporan ini.

Surabaya, Agustus 2009

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Manfaat	4
1.6 Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Profil SMK Negeri 6 Surabaya	7
2.1.1 Visi dan Misi SMK Negeri 6 Surabaya	7
2.1.2 Struktur Organisasi SMK Negeri 6 Surabaya	8
2.2 Jaringan Komputer	9
2.2.1 Pengertian Jaringan Komputer	9
2.2.2 Topologi Jaringan Komputer	13
2.2.2.1 Topologi Cincin (<i>ring topology</i>)	13
2.2.2.2 Topology Bus (<i>bus topology</i>)	14
2.2.2.3 Topologi Bintang (<i>star topology</i>)	15
2.2.3 Komponen Jaringan Komputer	16
2.2.4 TCP/IP (Transfer Control Protokol/Internet Protokol)	17
2.2.5 IP Address	18
2.2.6 Subnetting	20
2.2.7 Firewall	21
2.3 Linux	23
2.3.1 Mengenal Linux	24
2.3.2 Kelebihan Linux	25
2.3.3 Lisensi Linux	26
2.3.4 Kernel Linux	27
2.3.5 File Sistem Linux	28
2.3.6 Struktur Direktori Linux	28
2.3.7 Device Linux	29
2.3.8 Perintah Dasar Linux	31
2.4 Ubuntu	24

2.5 MySQL	36
2.6 PHP	43
2.7 Extjs	43
2.8 SAMBA	44
2.9 PROXY	45
 BAB III METODOLOGI PRAKTEK KERJA LAPANGAN	 52
3.1 Tempat dan Waktu Penelitian	52
3.2 Hardware dan Software	52
3.2.1 Komputer Router	52
3.2.2 Komputer Server	53
3.3 Pelaksanaan Kerja Praktek	53
 BAB IV HASIL DAN PEMBAHASAN	 54
4. 1 Perancangan Sistem	54
4.1.1 Perencanaan Konsep Jaringan	54
4.1.2 Perancangan Proses	56
4.2 Implementasi	57
4.2.1 Konfigurasi Pada Komputer Router	57
4.2.1.1 Pemasangan IP Address	58
4.2.1.2 Pemasangan DHCP Server dan DNS Forward	59
4.2.1.3 Pemasangan Transparan Proxy	65
4.2.1.4 Penambahan Program Autentifikasi mysql_auth ..	71
4.2.1.5 Management User Pengguna Internet	75
4.2.2 Konfigurasi Pada Komputer Server	79
4.2.2.1 Pemasangan IP Address pada Komputer Server	79
4.2.2.2 Konfigurasi Server SMB	80
 BAB V PENUTUP	 85
5.1 KESIMPULAN	85
5.2 SARAN	86
 DAFTAR PUSTAKA	 87

DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi SMK Negeri 6 Surabaya	8
Gambar 2.2 Topologi jaringan tipe cincin	13
Gambar 2.3 Topologi jaringan tipe bus	14
Gambar 2.4 Topologi star	15
Gambar 2.5 NIC (Network Interface Card)	17
Gambar 2.6 Diagram IP Tables	22
Gambar 2.7 Logo Linux (Tux)	23
Gambar 2.8 Logo Ubuntu	35
Gambar 4.1 Desain Jaringan	55
Gambar 4.2 Proses/Alur Sistem Jaringan	56
Gambar 4.3 IP Router	58
Gambar 4.4 Pengujian IP Address	59
Gambar 4.5 Instalasi DHCP	60
Gambar 4.6 Server DHCP	60
Gambar 4.7 Pengujian IP Address DHCP	63
Gambar 4.8 Tes Koneksi	65
Gambar 4.9 Instalasi Squid	65
Gambar 4.10 Pembuatan mysql_auth	72
Gambar 4.11 Proses Autentifikasi Proxy	73
Gambar 4.12 Web-browser Berhasil Autentifikasi Poxy	74
Gambar 4.13 Web-browser Gagal Autentifkasi	74
Gambar 4.14 Form Pendaftaran User Proxy	75
Gambar 4.15 Login Admin	76
Gambar 4.16 Form User Belum Registrasi	76
Gambar 4.17 Form User Teregristrasi	77
Gambar 4.18 Form User Terblokir	77
Gambar 4.19 Form Ganti Username dan Password Admin	78
Gambar 4.20 Form Ganti Username dan Password User	78
Gambar 4.21 Instalasi Samba	80
Gambar 4.22 Membuat User Pada Sistem Linux	82
Gambar 4.23 Mengaktifkan User Sistem Menjadi User SMB	83
Gambar 4.24 Direktori Sharing	83
Gambar 4.25 Autentifikasi Samba	84
Gambar 4.26 Akses Direktori Samba	84

DAFTAR TABEL

Tabel 2.1 Contoh IP address	18
Tabel 2.2 Kelas IP Address	18
Tabel 2.3 Struktur Direktori	29
Tabel 2.4 Device Harddisk dan Disk Pada Linux	30
Tabel 2.5 Device CDROM Pada Linux	30
Tabel 2.6 Device Khusus Pada Linux	31
Tabel 2.7 Perintah Dasar Pada Linux	32
Tabel 3.1 Pelaksanaan Kegiatan	53

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada saat ini, manfaat dari jaringan komputer sudah sangat banyak dirasakan. Apalagi dalam dunia komunikasi yang serba cepat ini, jaringan komputer sering kali berperan vital dalam kegiatan pendistribusian informasi yang cepat tersebut. Semua dari komponen yang tergabung dalam jaringan komputer tersebut haruslah mampu saling mendukung untuk menghasilkan satu sistem yang kokoh dan handal untuk melayani setiap permintaan informasi yang dibutuhkan oleh pengguna.

Sekian banyak manfaat dalam penerapan jaringan komputer tersebut ternyata belum dioptimalkan pada lingkungan pendidikan Sekolah Menengah Kejuruan Negeri 6 Surabaya. SMK Negeri 6 Surabaya merupakan Sekolah Menengah Kejuruan Negeri yang bertempat di Jl. Margorejo Wonocolo Surabaya. Di SMK Negeri 6 Surabaya terdapat empat jurusan, antara lain Tata Boga, Tata Busana, Tata Kecantikan, Akomodasi Perhotelan dan Hotel & Restoran. Selama ini di SMK Negeri 6 Surabaya, pada Laboratorium Komputer dan Perpustakaan masih dijalankan sebuah system lama, dimana setiap siswanya tanpa melakukan login dengan menggunakan account tunggal yang telah ada di sebuah server untuk akses internet. Setiap siswa juga menyimpan data pekerjaannya di harddisk dimana mereka menggunakan komputer. Sistem lama ini dipandang sangatlah tidak efisien, karena dengan sistem lama ini seorang Guru harus menyalakan semua komputer di laboratoriumnya hanya untuk mengambil data pekerjaan

siswanya, kemudian menilainya. Meskipun di Laboratorium sudah terdapat jaringan komputer, tetapi pemanfaatannya belum optimal. Sehingga manfaat jaringan komputer belum banyak dirasakan. Berangkat dari masalah yang ada tersebut penyusun ingin melakukan optimalisasi terhadap sistem lama itu dengan membangun beberapa server dalam jaringan agar manfaat jaringan komputer benar-benar bisa dirasakan di lingkungan Laboratorium SMK Negeri 6 Surabaya.

Server atau biasa disebut juga dengan hati suatu jaringan komputer merupakan komputer/alat yang digunakan untuk memberikan layanan tertentu yang berupa banyak permintaan secara bersamaan dan dalam kurun waktu yang lama (terus-menerus). Di dalam server juga berisi banyak data-data penting dan biasanya sangat rahasia. Oleh karena itu keamanan harus diperhatikan di semua sisi : keamanan dari pencurian (dari sisi jaringan ataupun fisiknya), kebakaran, keburukan kualitas pasokan listrik dan sebagainya. Untuk memberikan server fasilitas keamanan maka diperlukan jaringan dan sistem operasi yang memungkinkan untuk keamanan yang terjamin.

Salah satu sistem operasi yang cukup menjamin dalam segi keamanan jaringan adalah Linux. Linux merupakan salah satu sistem operasi yang dapat bekerja sebagai server dalam jaringan. Kemampuan linux sebagai server tidak dapat diragukan lagi. Hal ini dapat terbukti dari pilihan para administrator jaringan yang memilih sistem operasi linux sebagai server pada jaringan yang ditanganinya. Tidak hanya perusahaan-perusahaan kecil yang mengandalkan linux sebagai server, tapi banyak perusahaan-perusahaan besar yang telah mengandalkan linux sebagai servernya.

Dengan menggunakan Linux Ubuntu Server Intrepid Ibex 8.10 sebagai sistem operasi, sebuah server bisa dikonfigurasi untuk menjalankan berbagai peranan antara lain, sebagai file server, print server, mail server, terminal server, remote access/vpn server, domain controller, dns server, dhcp server, proxy server, mail sever, web server.

Dalam Kerja Praktek ini akan dilakukan konfigurasi server berbasis Ubuntu Server Intrepid Ibex 8.10 sebagai media untuk Jaringan Komputer di dalam Sekolah SMK Negeri 6 Surabaya.

1.2 Perumusan Masalah

Dengan adanya latar belakang di atas, maka didapatkan beberapa rumusan masalah sebagai berikut:

1. Bagaimana proses untuk dapat mengakses jaringan di SMK Negeri 6 Surabaya.
2. Bagaimana proses untuk menyimpan dan mengambil data di dalam *file server*.
3. Pemberian *user* dan *password* bagi pengguna internet yang ingin mengakses internet melalui proxy SMK Negeri 6 Surabaya.

1.3 Batasan Masalah

Batasan masalah yaitu meliputi:

1. Konfigurasi pada server dan router hanya menggunakan sistem operasi Linux Ubuntu Server Intrepid Ibex 8.10.
2. Untuk klien hanya menggunakan atau diimplementasikan pada sistem operasi Windows XP.

3. Persiapan paket pendukung, pengambilan paket ini mengacu pada *mirror* ubuntu <http://kambing.ui.edu>.
4. Server memanfaatkan samba sebagai *file sharing*
5. Router menyediakan *dhcp* dan *proxy server*.

1.4 Tujuan

Melakukan instalasi dan konfigurasi *dhcp server*, *domain server*, *file server* dan *proxy server* di jaringan komputer SMK Negeri 6 Surabaya.

1.5 Manfaat

Dengan merencanakan dan membangun jaringan komputer di SMK Negeri 6 Surabaya mempunyai manfaat sebagai berikut:

1. Mempermudah bagi pengguna untuk mendapatkan akses internet, karena tidak perlu melakukan tambahan setting;
2. Mempermudah dan mempercepat seorang guru untuk memberikan dan penilaian terhadap hasil pekerjaan siswa/siswinya;
3. Mampu menambah keamanan data milik *user account* yang bersangkutan sehingga data tidak mudah dicuri oleh *user account* lainnya;
4. Mampu mengefektifkan penggunaan *harddisk* pada sisi komputer klien, sehingga *hard disk* pada komputer klien tidak cepat penuh;
5. Mempermudah bagi guru/administrator untuk memonitor history penggunaan akses internet.

1.6 Sistematika Penulisan

Laporan ini akan menjelaskan secara lengkap tentang Rancang Bangun Jaringan Komputer di SMK Negeri 6 Surabaya. Agar lebih memahami materi, laporan ini di bagi menjadi lima bab yang dilengkapi dengan penjelasan pada tiap bab.

BAB I PENDAHULUAN

Bab ini akan menjelaskan tentang Latar Belakang, Perumusan Masalah, Batasan Masalah, Tujuan Praktek Kerja Lapangan, Manfaat, dan Sistematika Penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini akan menjelaskan tentang Landasan Teori yang dipakai sebagai penunjang pembuatan laporan ini. Penjelasan nya meliputi pengetahuan tentang jaringan komputer, sejarah dan pengenalan linux, pengenalan tentang ubuntu.

BAB III METODOLOGI PRAKTEK KERJA LAPANG

Bab ini menjelaskan tentang Tempat dan Waktu Penelitian, Metode Pelaksanaan Praktek Kerja Lapang, dan Pelaksanaan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang gambaran konsep jaringan yang dipakai, pembuatan jaringan komputer dengan menggunakan Linux

Ubuntu Server Intrepid Ibex 8.10, dan implementasi aplikasi-aplikasi untuk kebutuhan server.

BAB V PENUTUP

Bab ini akan menjelaskan tentang Kesimpulan dari keseluruhan isi dari laporan dan Praktek Kerja Lapang serta Saran yang disampaikan penulis dan perusahaan untuk pengembangan sistem yang ada demi kesempurnaan sistem yang lebih baik.

DAFTAR PUSTAKA

Pada bagian ini akan dipaparkan tentang sumber-sumber literatur yang digunakan dalam pembuatan laporan ini.

LAMPIRAN

Pada bagian ini berisi tentang konsep jaringan komputer dan keseluruhan konfigurasi pada pembuatan konsep jaringan komputer.

BAB II

TINJAUAN PUSTAKA

2.1 Profil SMK Negeri 6 Surabaya

SMK Negeri 6 Surabaya merupakan SMK Kelompok Pariwisata. Program Keahlian yang diselenggarakan terdiri dari Tata Boga, Tata Busana, Tata Kecantikan, Akomodasi Perhotelan dan Hotel & Restoran dengan program pembelajaran menggunakan pendekatan *Competency Based Training* (CBT).

“Terdepan dalam Mutu, Ide dan Pelayanan” merupakan komitmen SMK Negeri 6 Surabaya untuk memberikan kontribusi bagi peningkatan kualitas pendidikan dan terus berkembang seiring dengan kemajuan teknologi yang berlangsung pesat.

SMK Negeri 6 Surabaya telah dipercaya oleh Masyarakat dan instansi Pemerintah dalam penyelenggaraan program Pendidikan dan Pelatihan pariwisata dan Bahasa Asing bagi pegawai, dan berbagai penghargaan telah diterima diantaranya adalah predikat Sekolah Internasional.

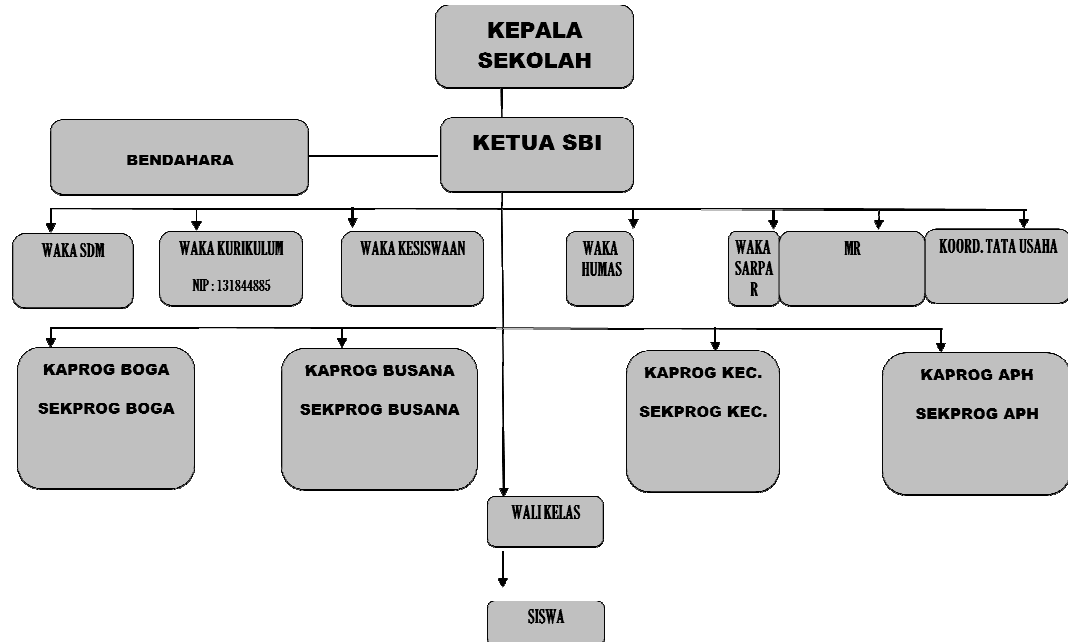
2.1.1 Visi dan Misi SMK Negeri 6 Surabaya

Visi dari SMK Negeri 6 Surabaya adalah mencetak Tenaga Kerja yang kompeten sesuai dengan tuntunan Dunia Usaha atau Dunia Industri di Tingkat Nasional dan Internasional seiring perkembangan jaman.

Misi dari SMK Negeri 6 Surabaya adalah :

1. Memberikan pembelajaran dan pelatihan secara profesional sesuai perkembangan jaman yang dilandasi iman dan taqwa kepada Tuhan Yang Maha Esa.
2. Menyediakan sarana, prasarana dan lingkungan pelatihan yang sesuai kebutuhan industri.
3. Meningkatkan tenaga pendidik dan tenaga kependidikan seiring dengan perkembangan dunia usaha dan dunia industri.
4. Menjalinkan Kerjasama dengan industri Nasional dan Internasional dalam pelatihan dan penyerapan lulusan.
5. Meningkatkan pelayanan Unit Produksi Sekolah.

2.1.2 Struktur Organisasi SMK Negeri 6 Surabaya



Gambar 2.1 Struktur Organisasi SMK Negeri 6 Surabaya

2.2 Jaringan Komputer

Jaringan komputer bukanlah sesuatu yang baru saat ini. Hampir di setiap perusahaan terdapat jaringan komputer untuk memperlancar arus informasi di dalam perusahaan tersebut. Internet yang mulai populer saat ini adalah suatu jaringan komputer raksasa yang merupakan jaringan komputer yang terhubung dan dapat saling berinteraksi.

Hal ini dapat terjadi karena adanya perkembangan teknologi jaringan yang sangat pesat, sehingga dalam beberapa tahun saja jumlah pengguna jaringan komputer yang tergabung dalam Internet berlipat ganda.

2.2.1 Pengertian Jaringan Komputer

Network atau jaringan, dalam bidang komputer dapat diartikan sebagai dua atau lebih komputer yang dihubungkan sehingga dapat berhubungan dan dapat berkomunikasi, sehingga akan menimbulkan suatu efisiensi, sentralisasi dan optimasi kerja. Pada jaringan komputer yang dikomunikasikan adalah data, satu komputer dapat berhubungan dengan komputer lain dan saling berkomunikasi (salah satunya bertukar data) tanpa harus membawa disc ke satu komputer ke komputer lainnya seperti yang biasa kita lakukan.

Ada beberapa jenis jaringan komputer dilihat dari cara pemrosesan data dan pengaksesannya:

1. Host-Terminal

Dimana terdapat sebuah atau lebih *server* yang dihubungkan dalam suatu *dumb terminal*. Karena *Dumb Terminal* hanyalah sebuah *monitor* yang dihubungkan dengan menggunakan kabel RS-232, maka pemrosesan data dilakukan di dalam *server*, oleh karena itu maka suatu *server* haruslah sebuah

sistem komputer yang memiliki kemampuan pemrosesan data yang tinggi dan penyimpanan data yang sangat besar.

2. Client – Server

Dimana sebuah *server* atau lebih yang dihubungkan dengan beberapa *client*. *Server* bertugas menyediakan layanan, bermacam-macam jenis layanan yang dapat diberikan oleh *server*, misalnya adalah pengaksesan berkas, *peripheral*, *database*, dan lain sebagainya. Sedangkan *client* adalah sebuah *terminal* yang menggunakan layanan tersebut. Perbedaannya dengan hubungan *dumb terminal*, sebuah *terminal client* melakukan pemrosesan data di *terminalnya* sendiri dan hal itu menyebabkan spesifikasi dari *server* tidaklah harus memiliki performansi yang tinggi, dan kapasitas penyimpanan data yang besar karena semua pemrosesan data yang merupakan permintaan dari *client* dilakukan di *terminal client*.

3. Peer to Peer

Dimana terdapat beberapa *terminal* komputer yang dihubungkan dengan media kabel. Secara prinsip, hubungan *peer to peer* ini adalah bahwa setiap komputer dapat berfungsi sebagai *server* (penyedia layanan) dan *client*, keduanya dapat difungsikan dalam suatu waktu yang bersamaan.

Sedangkan apabila di lihat dari sisi ruang lingkupnya atau jangkauannya, jaringan dapat di bagi menjadi beberapa jenis, yaitu:

a. LAN (Local Area Network)

Local Area Network merupakan salah satu arsitektur jaringan yang paling sederhana dan dapat dikembangkan menjadi arsitektur jaringan yang lebih luas cakupannya. Luas cakupan LAN itu sendiri tidak melebihi dari satu area yang

terdiri dari beberapa *terminal* yang saling dihubungkan sehingga menambahkan fungsi dari terminal itu sendiri. Layanan-layanan yang dapat diberikan LAN adalah penggunaan *file* bersama (*file sharing*) atau penggunaan printer bersama, (*printer sharing*).

Biasanya LAN menggunakan satu *server* untuk melayani kebutuhan *client*nya, tetapi tidak menutup kemungkinan untuk menggunakan satu *server*, tergantung kebutuhan dari *client* itu sendiri. Biasanya yang menjadi pertimbangan adalah jenis layanan yang dibutuhkan dan performansi jaringan itu sendiri. Apabila jenis layanan yang dibutuhkan banyak (*mail, web, ftp server*), maka sebaiknya *server* yang digunakan lebih dari satu dan hal tersebut akan mempengaruhi kinerja jaringan yang menggunakan layanan-layanan tersebut.

Penamaan terminal dalam suatu jaringan menggunakan apa yang disebut *IP Address (Internet Protocol Address)*. Sedang penamaan penamaan *server* berdasarkan nama *domain*nya disebut *DNS (Domain Name Server)*. Kedua cara penamaan ini merupakan cara penamaan yang biasa digunakan dalam jaringan. Hal-hal lebih lanjut akan di bahas langsung pada pengaplikasian instalasi jaringan pada bahasan selanjutnya.

b. MAN (Metropolitan Area Network)

Sebuah MAN, biasanya meliputi area yang lebih besar dari LAN, misalnya antar wilayah dalam satu propinsi. Dalam hal ini jaringan menghubungkan beberapa buah jaringan-jaringan kecil ke dalam lingkungan area yang lebih besar, sebagai contoh yaitu; jaringan Bank dimana beberapa kantor cabang sebuah Bank di dalam sebuah kota besar dihubungkan antara satu dengan lainnya. Misalnya Bank BNI yang ada di seluruh wilayah Jakarta atau Surabaya.

c. WAN (Wide Area Network)

Merupakan gabungan dari LAN, yang ruang lingkupnya dapat saja satu lokasi, misalnya gedung bertingkat, atau dapat tersebar di beberapa lokasi di seluruh dunia, jaringan jenis ini membutuhkan minimal satu *server* untuk setiap LAN, dan membutuhkan minimal dua *server* yang mempunyai lokasi yang berbeda untuk membentuknya.

d. Internet

Internet adalah sekumpulan jaringan yang berlokasi tersebar di seluruh dunia yang saling terhubung membentuk satu jaringan besar komputer. Dalam jaringan ini dibatasi layanannya sebagai berikut : *FTP, E-Mail, Chat, Telnet, Conference, News Group, Mailing List*. Biasanya jaringan ini menggunakan *protokol TCP/IP*, walaupun ada sebagian kecil yang menggunakan jenis lain (*IPX Novell Netware, NetBios*, dan lain-lainnya)

e. Intranet

Jenis jaringan ini merupakan gabungan dari LAN/WAN dengan Internet. Apabila di lihat dari lingkupannya atau jangkauannya maka jaringan ini adalah jenis LAN/WAN yang memberikan layanan seperti layanan internet kepada *terminal clientnya*. Perbedaan menyolok Intranet dengan Internet adalah Intranet melayani satu organisasi tertentu saja.

2.2.2 Topologi Jaringan Komputer

Apabila dilihat dari jenis hubungannya, maka topologi jaringan dapat dibagi menjadi tiga, yaitu :

1. Topologi Cincin (*ring topology*)
2. Topologi Bus (*bus topology*)
3. Topologi Bintang (*star topology*)

Berikut adalah ilustrasi dari ketiga topologi di atas

2.2.2.1 Topologi Cincin (*ring topology*)

Topologi jenis cincin ini menghubungkan satu komputer di dalam suatu loop tertutup. Pada topologi jenis ini data atau message berjalan mengelilingi jaringan dengan satu arah pengiriman ke komputer selanjutnya terus hingga mencapai komputer yang dituju. Waktu yang di butuhkan untuk mencapai terminal tujuan disebut *walk time* (waktu transmisi).



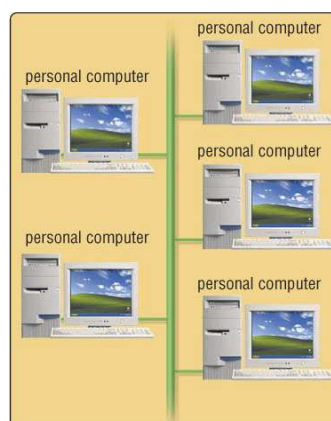
Gambar 2.2 Topologi Jaringan Tipe Cincin

Ada dua hal yang dilakukan oleh suatu terminal ketika menerima data dari komputer sebelumnya, yaitu :

- a. Memeriksa alamat yang dituju dari data tersebut dan menerimanya jika terminal ini merupakan tujuan data tersebut.
- b. Terminal akan meneruskan data ke komputer selanjutnya dengan memberikan tanda negatif ke komputer pengirim. Apabila ada komputer yang tidak berfungsi maka hal tersebut tidak akan mengganggu jalannya jaringan, tapi apabila satu kabel putus akan mengakibatkan jaringan tidak berfungsi.

2.2.2.2 Topology Bus (*bus topology*)

Topologi jaringan jenis ini menggunakan sebuah kabel pusat yang merupakan media utama dari jaringan. Terminal-terminal yang akan membangun jaringan dihubungkan dengan kabel utama yang merupakan inti dari jaringan. Data yang dikirimkan akan langsung menuju terminal yang dituju tanpa harus melewati terminal-terminal dalam jaringan, atau akan di routingkan ke *head end controller*. Tidak bekerjanya sebuah komputer tidak akan menghentikan kerja dari jaringan, jaringan akan tak bekerja apabila kabel utamanya dipotong atau putus.

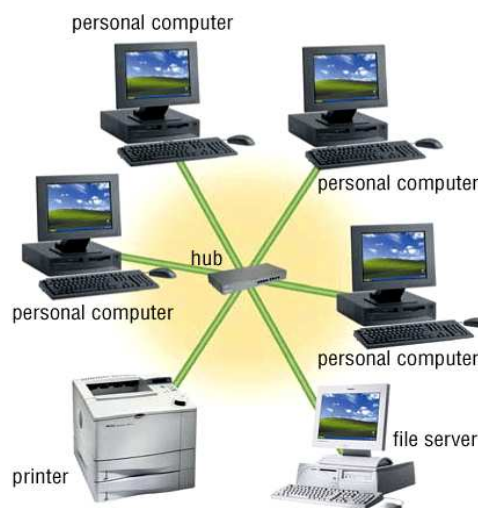


Gambar 2.3 Topologi Jaringan Tipe Bus

Jaringan ini merupakan jaringan yang banyak digunakan karena hanya dalam beberapa meter kabel dapat dihubungkan ke banyak terminal client. Jaringan ini biasanya menggunakan kabel coaxial sebagai media transmisinya. Kabel coaxial dilihat dari bentuk fisiknya mirip dengan kabel antena. Kabel ini mempunyai kapasitas bandwidth yang besar (2MB), sehingga apabila dihubungkan dengan banyak terminal akan terlayani dengan baik.

2.2.2.3 Topologi Bintang (*star topology*)

Topologi jaringan ini menggunakan satu terminal sebagai terminal sentral yang menghubungkan ke semua terminal client. Terminal sentral ini yang mengarahkan setiap data yang dikirimkan ke komputer yang dituju. Jenis jaringan ini apabila ada salah satu terminal client tidak berfungsi atau media transmisi putus atau terganggu maka tidak akan mempengaruhi kerja dari jaringan, karena gangguan tersebut hanya mempengaruhi terminal yang bersangkutan.



Gambar 2.4 Topologi Star

Kelemahan dari jenis topologi jaringan ini adalah ketergantungan terhadap suatu terminal sentral. Hal tersebut merupakan suatu gangguan yang sangat berarti apabila terminal sentral tersebut mendapatkan gangguan, sehingga dicari suatu solusi yang dapat mengatasi masalah tersebut. Salah satu solusi yang banyak dilakukan adalah dengan menggunakan dua buah terminal sebagai server, sehingga apabila satu server dalam keadaan down dapat dialihkan ke server yang kedua dan begitu seterusnya.

2.2.3 Komponen Jaringan Komputer

Jaringan komputer terbentuk karena adanya komponen-komponen berikut ini:

1. Protokol

Protokol adalah suatu aturan-aturan main yang mengatur komunikasi diantara beberapa komputer di dalam sebuah jaringan, aturan itu termasuk di dalamnya petunjuk yang berlaku bagi cara-cara atau metode mengakses sebuah jaringan, topologi fisik, tipe-tipe kabel dan kecepatan transfer data.

2. NOS (Network Operation System)

NOS atau *Network Operation System* merupakan *software* yang memiliki protokol tertentu sehingga dapat berkomunikasi dan berbagi data sebagai bagian dari ketersediaan sumber daya yang terdapat di jaringan

3. NIC (Network Interface Card)

NIC atau sering disebut dengan *LAN Card* adalah suatu *Interface* berkecepatan tinggi yang menghubungkan PC ke dalam suatu jaringan. Jaringan yang dimaksud adalah jaringan dalam skala LAN, MAN, maupun WAN. *Ethernet* dalam sistem jaringan pertama kali dibuat dan dipatenkan oleh

perusahaan Xerox. *Ethernet* merupakan implementasi metoda CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) yang dikembangkan tahun 1960 pada proyek *wireless* ALOHA di Hawaii University diatas kabel *coaxial*.



Gambar 2.5 NIC (network interface card)

2.2.4 TCP/IP (Transfer Control Protokol/Internet Protokol)

Dalam dunia komunikasi data komputer, protokol mengatur bagaimana sebuah komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer kita dapat menggunakan banyak macam protokol agar sebuah komputer dapat berkomunikasi, keduanya harus menggunakan protokol yang sama. Protokol berfungsi mirip dengan bahasa. Agar dapat berkomunikasi seseorang perlu berbicara dan mengerti bahasa yang mereka gunakan.

TCP/IP (Transfer Control Protokol/Internet Protokol) adalah sekelompok protokol yang mengatur komunikasi data komputer di internet. Komputer-komputer yang terhubung ke internet berkomunikasi dengan protokol TCP/IP, karena menggunakan bahasa yang sama perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. Jadi, jika sebuah komputer menggunakan protokol TCP/IP dan terhubung langsung ke internet, maka komputer tersebut dapat berhubungan dengan komputer di belahan dunia mana pun yang juga terhubung ke internet.

2.2.5 IP Address

IP address adalah alamat yang diberikan pada jaringan komputer dan peralatan jaringan yang menggunakan protokol TCP/IP. *IP address* terdiri atas 32 bit angka biner yang dapat dituliskan sebagai empat kelompok angka desimal yang dipisahkan oleh tanda titik seperti 192.168.0.1.

Tabel 2.1 Contoh IP address

Network ID			Host ID
192	168	0	1

IP address terdiri atas dua bagian yaitu *network ID* dan *host ID*, dimana *network ID* menentukan alamat jaringan komputer, sedangkan *host ID* menentukan alamat *host* (komputer, router, switch). Oleh sebab itu *IP address* memberikan alamat lengkap suatu *host* beserta alamat jaringan di mana *host* itu berada.

1. Kelas-kelas IP Address

Untuk mempermudah pemakaian, bergantung pada kebutuhan pemakai, *IP address* dibagi dalam tiga kelas seperti diperlihatkan pada Tabel 2.2.

Tabel 2.2 Kelas IP Address

Kelas	Network ID	Host ID	Default Subnet Mask
A	xxx.0.0.1	xxx.255.255.254	255.0.0.0
B	xxx.xxx.0.1	xxx.xxx.255.254	255.255.0.0
C	xxx.xxx.xxx.1	xxx.xxx.xxx.254	255.255.255.0

IP address kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar. *Range IP* 1.xxx.xxx.xxx. – 126.xxx.xxx.xxx, terdapat 16.777.214 (16 juta) *IP address* pada tiap kelas A. *IP address* kelas A diberikan untuk jaringan dengan jumlah *host* yang sangat besar. Pada *IP address* kelas A, *network*

ID ialah 8 bit pertama, sedangkan *host ID* ialah 24 bit berikutnya. Dengan demikian, cara membaca *IP address* kelas A, misalnya 113.46.5.6 ialah:

Network ID = 113

Host ID = 46.5.6

Sehingga *IP address* diatas berarti *host* nomor 46.5.6 pada *network* nomor 113.

IP address kelas B biasanya dialokasikan untuk jaringan berukuran sedang dan besar. Pada *IP address* kelas B, *network ID* ialah 16 bit pertama, sedangkan *host ID* ialah 16 bit berikutnya. Dengan demikian, cara membaca *IP address* kelas B, misalnya:

132.92.121.1

Network ID = 132.92

Host ID = 121.1

Sehingga *IP address* di atas berarti *host* nomor 121.1 pada *network* nomor 132.92. dengan panjang *host ID* 16 bit, *network* dengan *IP address* kelas B dapat menampung sekitar 65000 *host*. Range *IP* 128.0.xxx.xxx – 191.155.xxx.xxx.

IP address kelas C awalnya digunakan untuk jaringan berukuran kecil (LAN). *Host ID* ialah 8 bit terakhir. Dengan konfigurasi ini, bisa dibentuk sekitar 2 juta *network* dengan masing-masing *network* memiliki 256 *IP address*. Range *IP* 192.0.0.xxx – 223.255.255.x. Pengalokasian *IP address* pada dasarnya ialah proses memilih *network Id* dan *host ID* yang tepat untuk suatu jaringan. Tepat atau tidaknya konfigurasi ini tergantung dari tujuan yang hendak dicapai, yaitu mengalokasikan *IP address* seefisien mungkin.

2.2.6 Subnetting

Subnetting berfungsi untuk menyembunyikan detail dari *internal network* suatu organisasi ke router *eksternal*. Selain itu, *subnetting* juga mempermudah manajemen jaringan dan menambah efisiensi dari jaringan tersebut. Dengan *subnetting* kita dapat membatasi jumlah maksimal host yang dapat dialokasikan pada suatu subnet. Dengan *subnetting* kita dapat memeriksa kesalahan jaringan dengan cepat karena kesalahan tersebut sudah terlokalisasi. Bayangkan jika organisasi yang memiliki 1000 komputer tidak melakukan *subnetting*, jika terjadi satu kesalahan, maka semua *network* akan *down*. Demikian pula administrator yang harus memperbaiki kesalahan tersebut harus mencari kesalahan satu per satu dalam 1000 komputer tersebut. *Network* tanpa *subnetting* juga akan memperberat tugas *router* karena *routing table*-nya yang sangat banyak dan harus *membroadcast host* sekian banyak tersebut.

IP Address v4 memiliki struktur alamat yang tersusun atas bilangan 32 bit. *Subnet mask* adalah suatu bilangan biner 32 bit yang akan di-AND-kan dengan *IP Address* untuk mendapatkan *subnet host*. Misalnya alamat *IP Address* 10.126.11.16 dan 10.126.11.17 dengan *netmask* 255.255.255.0. Untuk mendapatkan lokasi *subnet host* ini, *IP Address* 10.126.11.16 akan di-AND dengan *netmask*-nya. Jika *IP Address* 10.126.11.17 di-AND dengan *netmask*-nya menghasilkan hasil yang sama dengan hasil AND dari 10.126.11.16 dengan *netmask*-nya, maka 2 *IP Address* tersebut berada dalam satu subnet. *IP Address* dengan alamat 10.126.11.x dengan *netmask* 255.255.255.0 (24 bit) akan memiliki *Net Address* 10.126.11.0 dan *Broadcast Address* 10.126.11.255. Alamat IP diantara 10.126.11.0 dan 10.126.11.255 adalah alamat IP yang dapat dialokasikan

dalam *subnet* tersebut. Dengan demikian, semakin sedikit jumlah bit *netmask* yang digunakan, maka semakin banyak jumlah IP Address yang dapat dialokasikan begitu pula sebaliknya. (Awaludin, 2009)

2.2.7 Firewall

Firewall adalah alat yang digunakan untuk melindungi jaringan privat dari jaringan publik (internet). Firewall melindungi jaringan privat dengan cara mengendalikan paket yang melewatinya. Firewall dirancang untuk mengendalikan aliran paket berdasarkan pada asal, tujuan, port dan informasi tipe paket yang terdapat pada masing-masing paket. Firewall berisi sederet daftar aturan yang digunakan untuk menentukan nasib paket yang akan datang atau pergi dari Firewall menurut kriteria dan parameter tertentu.

Semua paket yang diperiksa Firewall akan melalui sederetan table dalam prosesnya. Masing-masing tabel dikenakan untuk tipe aktivitas paket tertentu dan dikendalikan oleh rantai aturan filter paket yang sesuai. Rantai (*chains*) adalah daftar aturan yang dibuat untuk mengendalikan paket.

Pada Firewall terjadi beberapa proses yang memungkinkan Firewall dapat melindungi jaringan. Proses yang terjadi pada Firewall ada tiga macam yaitu :

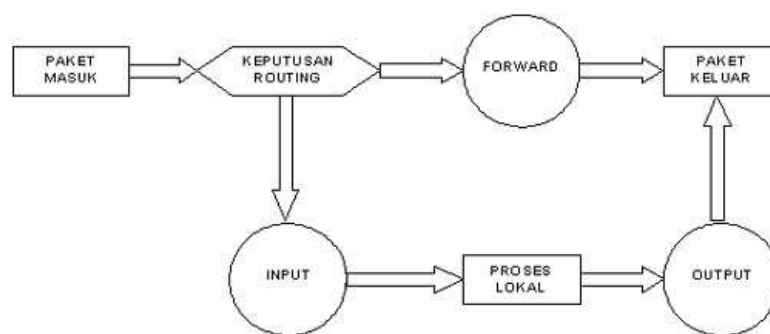
- a. Modifikasi header paket;
- b. Translasi alamat jaringan; dan
- c. Filter paket.

Modifikasi header paket digunakan untuk memodifikasi kualitas layanan bit paket TCP sebelum terjadi routing. Translasi alamat jaringan yang terjadi pada Firewall adalah antara jaringan privat dan jaringan public. Translasi yang terjadi

dapat berupa translasi satu ke satu (*one to one*) yang mana satu alamat IP privat dipetakan ke satu alamat IP publik serta translasi banyak ke satu (*many to one*) yang mana beberapa IP privat dipetakan ke satu alamat IP publik. Filter paket digunakan untuk menentukan nasib paket apakah dapat diteruskan atau tidak.

Salah satu perangkat lunak yang banyak digunakan untuk keperluan proses pada Firewall adalah iptables. Program iptables adalah program administratif untuk filter dan NAT (*Network Address Translation*). Untuk menjalankan fungsinya, iptables dilengkapi dengan tabel mangle, nat, dan filter.

Proses yang terjadi pada paket yang melewati suatu Firewall dapat digambarkan sebagai berikut :



Gambar 2.6. Diagram IP Tables

Ketika paket datang dari suatu jaringan masuk pada Firewall melalui kartu jaringan, pertama kali paket akan diperiksa oleh aturan rantai PREROUTING pada tabel mangle. Selanjutnya paket diperiksa oleh aturan rantai PREROUTING pada tabel nat, apakah paket memerlukan DNAT. Setelah itu paket kemudian mengalami routing.

Jika tujuan paket adalah jaringan lain, maka paket akan difilterkan oleh aturan rantai FORWARD pada tabel filter. Jika perlu, paket akan diperiksa oleh aturan rantai POSTROUTING pada tabel nat, apakah paket memerlukan SNAT.

Jika tujuan paket adalah Firewall, maka paket akan difilterkan oleh aturan rantai INPUT pada tabel filter. Selanjutnya paket akan mengalami proses lokal.

Proses local yang terjadi pada Firewall dapat berupa pengiriman paket kembali. Paket ini akan diperiksa oleh aturan rantai OUTPUT pada tabel mangle. Selanjutnya paket diperiksa oleh aturan rantai OUTPUT pada tabel nat, apakah memerlukan DNAT. Sebelum routing, paket akan difilterkan oleh aturan rantai OUTPUT pada tabel filter. (Wagito. ST. MT, 2005)

2.3 Linux

Linux [LINus UniX] adalah sebuah sistem operasi yang dibuat oleh Linus Torvalds, seorang remaja 21 Tahun yang kuliah di universitas Helsinki Finlandia. *Linux* merupakan sebuah sistem operasi turunan dari *UNIX* yang lengkap, karena bisa digunakan untuk jaringan, pengembangan perangkat lunak, multimedia, pekerjaan kantor dan lain sebagainya. *Linux* biasanya dipaket dengan program-program dari proyek *GNU* ataupun program-program lain yang dibundel dengan sebutan “*distro Linux*”.



Gambar 2.7. Logo Linux (Tux)

2.3.1 Mengenal Linux

Sistem operasi (*operating system*) atau biasa disebut dengan singkatan “OS” merupakan program yang mengendalikan dan mengkoordinasi semua kegiatan operasi dari sistem komputer.

Linux (Linus UNIX) terinspirasi oleh Minix (Mini UNIX], Minix merupakan sebuah sistem operasi kecil yang dikembangkan oleh Andrew S Tanenbaum. Perkembangan Linux awalnya dimulai sekitar tahun 1991, ketika seorang mahasiswa Finlandia yang bernama Linus Benedict Torvalds menulis Linux.

Pada awalnya Linux merupakan sebuah proyek hobi yang dikerjakan oleh Linus Torvalds, remaja 21 Tahun yang kuliah di universitas Helsinki Finlandia. Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991 yang kemudian di-*posting* ke *mailing list* comp.os.minix, dengan maksud menawarkan sumber kode (*source code*) dari apa yang telah Linus Torvalds buat dan mengundang para *programmer* lainnya untuk bergabung serta turut berpartisipasi di dalam pengembangan proyek tersebut. Linux merupakan sebuah sistem operasi turunan dari UNIX, yang merupakan implementasi independen dari standar IEEE (*Institute of Electrical and Electronics Engineers*) untuk sistem operasi yang bernama *Portable Operating System Interface for Computing Environments* atau biasa disebut POSIX. Linux seperti layaknya sistem operasi UNIX lainnya, mendukung banyak *software* mulai dari TEX, X-window, GNU C/C++ hingga TCP/IP.

Pada awal perkembangannya, *source code* Linux disediakan secara bebas dengan lisensi GPL (*General Public License*) dan melalui internet. Hasilnya, sejarah Linux merupakan kolaborasi banyak pengguna (*user*), dan semuanya

dilakukan melalui internet. Dari *kernel* awal yang hanya mengimplementasikan subset kecil dari sistem UNIX, sistem Linux telah bertumbuh dimana telah mampu memasukkan banyak fungsi UNIX.

Kernel Linux perlu dibedakan dari sebuah sistem Linux, *kernel* Linux merupakan sebuah perangkat lunak asli (*original*) yang dibuat oleh komunitas Linux. Sedangkan sistem Linux yang diketahui saat ini, mengandung banyak komponen yang dibuat sendiri atau dipinjam dari proyek lain.

2.3.2 Kelebihan Linux

Kata "Linux" untuk saat ini sudah tidak asing lagi bagi para pengguna internet dan komunitas mahasiswa. Secara teknis dan singkat dapat dikatakan, Linux adalah suatu sistem operasi yang bersifat *multiuser* dan *multitasking*, yang dapat berjalan di berbagai *platform* termasuk pada prosesor Intel 386, dan AMD64 atau mungkin ada yang lebih tinggi.

Sistem operasi Linux mempunyai banyak kelebihan dibandingkan dengan beberapa sistem operasi yang lainnya. Berikut ini adalah beberapa fakta dari hal-hal yang menguntungkan dengan menggunakan program dan *file* Linux:

1. Bebas (*free*). Lebih dari sekedar gratis, karena lisensi Linux adalah GNU/GPL. Sehingga pengguna (*user*) lainnya bebas dalam mendistribusikannya kembali, menulis atau membuat ulang selama masih mengikuti dan mentaati aturan-aturan yang berlaku dari lisensi GNU/GPL;
2. Mengurangi ketergantungan pada *vendor* perangkat lunak (*software*) tertentu;
3. Lebih murah, karena merupakan *free-software*;

4. Pengembangannya sangat cepat, karena *source code* dapat dikembangkan oleh siapa saja yang ingin berpartisipasi di dalam proyek Linux;
5. Pada dasarnya semua data tersimpan di dalam harddisk, meski ada beberapa kondisi dimana data tersimpan pada simpanan sementara lainnya, seperti disket, flashdisk, dan CD (*Compact Disc*). Linux memberikan beberapa proses spesial dimana terminal, printer dan *device hardware* lainnya dapat diakses seperti mengakses *file* yang tersimpan dalam harddisk atau simpanan sementara lainnya;
6. Menyediakan layanan (*service*) untuk membuat, memodifikasi program, proses dan *file*;
7. Mendukung struktur *file* yang bersifat hirarki (*hierarchy*) atau terstruktur;
8. Merupakan salah satu sistem operasi yang termasuk ke dalam kelas sistem operasi yang dapat melakukan *multitasking*. *Multitasking* sendiri adalah keadaan dimana suatu sistem operasi dapat melakukan banyak pekerjaan pada saat yang bersamaan;
9. Selain *multitasking*, Linux juga dapat mendukung *multiuser*. Yaitu sistem operasi yang pada saat bersamaan dapat digunakan oleh lebih dari satu pengguna (*user*) yang masuk ke dalam sistem.

2.3.3 Lisensi Linux

Satu hal yang membedakan Linux terhadap sistem operasi lainnya, Linux merupakan sistem operasi yang memiliki sekumpulan lengkap aplikasi dan program-program lainnya, yang kebanyakan dibawah naungan proyek GNU (*GNU Is Not UNIX*) dari *Free Software Foundation*.

Menurut *Free Software Foundation*, definisi dari *free software* adalah sebuah bentuk kebebasan yang mengacu kepada pengguna (*user*) untuk menggunakan, meng-*copy*, mendistribusikan, mempelajari, mengubah, maupun meningkatkan perangkat lunak (*software*) tersebut.

Secara spesifik terdapat empat arti dari kebebasan, yaitu:

1. Bebas menjalankan program;
2. Bebas mempelajari program dan mengadaptasi sesuai kebutuhan;
3. Bebas mendistribusikan ulang;
4. Bebas menambahkan atau meningkatkan program agar lebih baik dan dapat mempublikasikannya.

Dalam proyek GNU digunakan dua lisensi untuk melindungi kebebasan secara legal, yaitu:

1. *Copyleft*: siapapun yang mendistribusikan ulang program dengan atau tanpa perubahan harus memberikan kebebasan juga untuk didistribusikan lagi atau diubah. *Copyleft* menjamin setiap pengguna mempunyai kebebasan;
2. *Non-copyleft*: siapapun yang mendistribusikan atau mengubah harus mendapat izin dari pembuat programnya.

2.3.4 Kernel Linux

Kernel Linux terdiri dari beberapa bagian penting, seperti: manajemen proses, manajemen memori, manajemen jaringan, *hardware device drivers*, *filesystem drivers*. Namun bagian yang terpenting ialah manajemen proses dan manajemen memori. Manajemen memori menangani daerah pemakaian memori, daerah *swap*, dan bagian-bagian *kernel*. Manajemen proses menangani pembuatan

proses dan penjadwalan proses (*process scheduling*). Pada bagian dasar *kernel* berisi *hardware device drivers* untuk setiap jenis *hardware* yang didukung.

Dengan demikian *Kernel* merupakan program inti yang mengatur komponen penting komputer sebagai contoh prosesor dan memori, serta beberapa *device driver* yang mengatur sistem perangkat keras (*hardware*) yang lain seperti kartu jaringan (*network card*), kartu suara (*sound card*), VGA (*Video Graphics Accelerator*). *Kernel* mengatur segala sesuatu yang berhubungan dengan program-program untuk dapat berkomunikasi dengan perangkat kerasnya (*hardware*).

Kernel ini menyediakan semua fungsi yang diperlukan untuk menjalankan proses, dan disediakan *system service* untuk memberikan pengaturan dan proteksi akses ke sumber daya (*resource*) perangkat keras (*hardware*).

2.3.5 File Sistem Linux

File sistem (filesystem) adalah sekumpulan dari inode-inode dengan satu inode pembeda yaitu "*root*". Inode lainnya diakses mulai dari *root* inode dan pencarian nama *file* untuk menuju ke inode lainnya. *File sistem* mempunyai beberapa karakteristik yang mencakup seluruh inode dalam *file sistem*. Salah satu yang terpenting adalah *blocksize*. *Blocksize* merupakan ukuran blok untuk *file* dalam satuan byte seperti 1024 byte.

2.3.6 Struktur Direktori Linux

Direktori (*directory*) *root* pada Linux mempunyai beberapa direktori yang merupakan standar direktori pada banyak distronya.

Tabel 2.3. Struktur Direktori.

Direktori	Isi
/bin	Berisikan <i>file binary</i> standar yang dapat dipergunakan oleh seluruh pengguna (<i>user</i>), baik <i>user</i> biasa maupun <i>super user</i> .
/boot	Berisi <i>file</i> yang digunakan untuk <i>booting</i> Linux, termasuk <i>kernel image</i> .
/dev	Berisikan <i>filesystem</i> khusus yang merupakan refleksi dari <i>device hardware</i> yang dikenali dan digunakan oleh sistem.
/etc	Berisikan <i>file</i> konfigurasi sistem, biasanya konfigurasi hanya boleh dilakukan oleh <i>super user</i> .
/home	Berisikan direktori yang merupakan direktori <i>home</i> untuk <i>user</i> biasa dan aplikasi tertentu.
/lib	Berisikan <i>file library</i> yang dipergunakan untuk mendukung kerja <i>kernel</i> Linux.
/mnt	Merupakan direktori khusus yang disediakan untuk mengaitkan (<i>mounting</i>) <i>device disk storage</i> ke sistem dalam bentuk direktori.
/proc	Berisikan <i>filesystem</i> khusus yang menunjukkan data <i>kernel</i> setiap saat.
/root	Direktori <i>home</i> untuk <i>user root</i> (pengguna khusus dengan <i>privileges</i> hampir tak terbatas).
/sbin	Sama seperti direktori <i>bin</i> , tetapi hanya <i>super user</i> yang sebaiknya menggunakan <i>binary</i> tersebut, mengingat fungsi <i>binary</i> yang terdapat pada direktori ini untuk pemeliharaan (<i>maintenance</i>) sistem.
/tmp	Berisikan <i>file</i> sementara yang dibutuhkan sebuah aplikasi ketika sedang berjalan.
/usr	Berisikan <i>library</i> , <i>binary</i> , dokumentasi, dan <i>file</i> lainnya hasil dari instalasi <i>user</i> .
/var	Berisikan <i>file log</i> , <i>mailbox</i> dan data aplikasi.

2.3.7 Device Linux

Semua alat (*device*) pada linux dinyatakan dalam bentuk *file*. Apabila nantinya *device* ini perlu untuk diakses, maka hanya melihat isi dari direktori /dev. Berikut adalah tabel dari penamaan *device* pada Linux:

1. Harddisk dan disk.

Tabel 2.4. Device Harddisk dan Disk Pada Linux.

Nama <i>File</i>	Nama <i>Device</i>
/dev/fd0	Drive <i>floppy</i> pertama
/dev/fd1	Drive <i>floppy</i> kedua
/dev/had	Harddisk bus AT pertama
/dev/hda1 - /dev/hda15	Partisi dari harddisk bus AT pertama
/dev/hdb	Harddisk bus AT kedua
/dev/sda	Harddisk SCSI pertama
/dev/sda1 - /dev/sda15	Partisi dari harddisk SCSI pertama
/dev/sdb	Harddisk SCSI kedua
/dev/sdc	Harddisk SCSI ketiga

2. Drive CD-ROM.

Tabel 2.5. Device CDROM Pada Linux.

Nama <i>File</i>	Nama <i>Device</i>
/dev/cdrom	<i>Link</i> ke CD-ROM yang digunakan
/dev/hda - /dev/hdd	ATAPI IDE CD-ROM.
/dev/scd0 - /dev/scd1	Drive CD-ROM SCSI.

3. *Device* khusus.

Tabel 2.6. Device Khusus Pada Linux.

Nama <i>File</i>	Nama <i>Device</i>
/dev/null	Data apapun (data bin) akan di masukkan.
/dev/tty1 - /dev/tty8	Konsol <i>virtual</i> .
/dev/zero	Menghasilkan keluaran bilangan dengan byte null.

2.3.8 Perintah Dasar Linux

Seperti halnya bila mengetikkan perintah pada DOS (*Distributed Operating System*), *command line* atau baris perintah pada Linux juga diketikkan di prompt atau shell dan diakhiri dengan menekan *enter* untuk mengeksekusi perintah tersebut. Baris perintah merupakan cara yang lebih praktis dan mudah, karena dengan hanya mengetik saja untuk melakukan atau menyelesaikan suatu pekerjaan, oleh karena itu pengguna Linux tetap mengandalkan cara seperti ini.

Sebaiknya pengguna (*user*) juga harus mengetahui dan sedikitnya pernah menggunakan perintah-perintah di Linux, karena suatu saat pengetahuan akan perintah-perintah sangat diperlukan dalam pengoperasian Linux. Beberapa perintah dasar Linux yang mungkin akan sering digunakan pada beberapa distro Linux adalah sebagai berikut:

Tabel 2.7. Perintah Dasar Pada Linux.

Perintah	Keterangan
man [perintah]	Untuk melihat dan mempelajari perintah pada Linux.
Ls	Melihat isi <i>file</i> dari direktori aktif.

ls -al	Melihat seluruh isi <i>file</i> pada direktori aktif beserta <i>filehidden</i>
cd [direktori]	<i>Change directory</i> . Menggunakan "cd" tanpa nama direktori, akan menghantarkan ke direktori <i>root</i> , dan "cd .." akan menghantarkan ke direktori sebelumnya.
cp [sumber] [tujuan]	Meng-copy suatu <i>file</i> (<i>copy</i>).
mv [sumber] [tujuan]	Memindahkan suatu <i>file</i> (<i>move</i>).
mv [namalama] [namabaru]	Merubah nama <i>file</i> (<i>rename</i>).
rm [sumber]	Menghapus suatu <i>file</i> (<i>remove</i>). Gunakan opsi -r untuk <i>recursive remove</i> , akan menghapus <i>file</i> , direktori, dan subdirektorinya. Contoh: rm -r [sumber]
mkdir [direktori]	Membuat direktori baru (<i>make directory</i>).
rmdir [direktori]	Menghapus direktori (<i>remove directory</i>).
cat	Menampilkan isi dari sebuah <i>file</i> .
nano [namafile]	Membuat dan <i>edit file</i> . Dapat menggunakan teks editor yang lainnya seperti "pico, vim, vi".
tar -xvf namafile.tar	Meng-untar <i>file</i> tar yang tidak terkompres (*.tar).
tar -xjf namafile.tar.bz2	Meng-untar sebuah <i>file</i> tar sekaligus meng-uncompress <i>file</i> tersebut (*.tar.bz2).
bunzip2 namafile.bz2	Meng-uncompress <i>file</i> dengan format (*.bz2), dengan menggunakan "bzip2".
gunzip namafile.gz	Meng-uncompress <i>file</i> dengan format (*.gz). dengan menggunakan "gzip"
unzip namafile.zip	Meng-uncompress <i>file</i> dengan format (*.zip), dengan menggunakan "unzip".
Make	Kegunaan dari <i>make</i> adalah untuk menentukan secara otomatis potongan dari suatu program yang perlu di-compile ulang. Dengan menggunakan "make".

Fakeroot	Fakeroot merupakan perintah untuk memanipulasi <i>file</i> , fakeroot ini dipergunakan oleh pengguna (<i>user</i>) untuk menciptakan <i>file</i> (.deb). Dengan menggunakan "fakeroot".
Mount	Me-mount <i>filesystem</i> ke suatu direktori yang telah ditentukan. Hanya <i>super user</i> yang dapat melakukan ini.
Umount	Kebalikan dari <i>mount</i> , yaitu untuk meng-unmount <i>filesystem</i> . Setelah "umount" ini dijalankan, direktori yang ditentukan tidak bisa digunakan lagi.
Fd	Untuk membuat <i>image</i> , menggunakan "debootstrap".
Debootstrap	Untuk mengisi <i>filesystem</i> dengan <i>base system</i> . Dengan menggunakan "debootstrap"
./namafile	Menjalankan atau mengeksekusi suatu <i>file</i> , dapat menggunakan "sh [namafile]".
Startx	Menjalankan X-window.
shutdown -h now	Turn Off sistem, atau gunakan "shutdown -r now" untuk restart sistem. Dapat pula menggunakan "init 0" untuk turn off dan "init 6" untuk restart.
Halt	Reboot halt atau reboot mesin, lebih praktis dari perintah diatas.
Hostname	Menampilkan nama <i>local host</i> .
adduser [namapengguna]	Menambah pengguna (<i>user</i>).
ps -aux	Melihat berupa <i>list</i> seluruh proses yang dijalankan. Dapat menggunakan "ps axu".
Free	Informasi memori dalam kilobyte.
uname -a	Informasi <i>kernel</i> .
df -h	Melihat informasi pemakaian harddisk.
Clear	Membersihkan layar.

Perintah tersebut merupakan perintah yang mungkin akan sering digunakan, akan tetapi tidak menutup kemungkinan perintahnya dikombinasikan atau ditambahkan dengan perintah-perintah Linux lainnya, untuk mengetahui lebih jelas mengenai perintah-perintah dan kombinasinya tersebut, maka dapat menggunakan perintah “man” atau dapat menggunakan “help”. (Wagito. ST. MT,2005)

2.4 Ubuntu

Kata "*Ubuntu*" berasal dari bahasa kuno Afrika, yang berarti "rasa perikemanusiaan terhadap sesama manusia". *Ubuntu* juga bisa berarti "aku adalah aku karena keberadaan kita semua". Tujuan dari distribusi *Linux Ubuntu* adalah membawa semangat yang terkandung di dalam *Ubuntu* ke dalam dunia perangkat lunak.

Ubuntu adalah sistem operasi lengkap berbasis *Linux*, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional. *Ubuntu* sendiri dikembangkan oleh komunitas sukarelawan *Ubuntu* dan kami mengundang pengguna *Ubuntu* untuk turut serta berpartisipasi mengembangkan *Ubuntu*.

Komunitas *Ubuntu* dibentuk berdasarkan gagasan yang terdapat di dalam filosofi *Ubuntu*, bahwa perangkat lunak harus tersedia dengan bebas biaya, bahwa aplikasi perangkat lunak tersebut harus dapat digunakan dalam bahasa lokal masing-masing dan untuk orang-orang yang mempunyai keterbatasan fisik, dan

bahwa seseorang harus mempunyai kebebasan untuk mengubah perangkat lunak sesuai dengan apa yang mereka butuhkan.



Gambar 2.8 Logo Ubuntu

Perihal kebebasan inilah yang membuat *Ubuntu* berbeda dari perangkat lunak berwujud atau *proprietary*, bukan hanya peralatan yang Anda butuhkan tersedia secara bebas biaya, tetapi Anda juga mempunyai hak untuk memodifikasi perangkat lunak Anda sampai perangkat lunak tersebut bekerja sesuai dengan yang Anda inginkan.

Berikut ini adalah komitmen publik tim *Ubuntu* untuk para penggunanya sejak *Ubuntu* lahir hingga sekarang:

1. *Ubuntu* akan selalu bebas dari biaya, maka dari itu tidak akan ada biaya tambahan untuk "edisi *enterprise*", kami akan membuat semua pekerjaan terbaik *Ubuntu* tersedia untuk semua orang dengan istilah Bebas yang sama.
2. *Ubuntu* juga menyediakan dukungan komersial dari ratusan perusahaan di seluruh dunia. *Ubuntu* dirilis secara tetap dan dapat Anda prediksi rilis *Ubuntu* terbaru tersedia setiap enam bulan. Setiap rilis akan didukung oleh *Ubuntu* dengan perbaikan pada keamanan dan perbaikan lainnya secara bebas selama sekurangnya 18 bulan.
3. *Ubuntu* akan menyertakan terjemahan dan prasarana aksesibilitas terbaik yang dimiliki oleh komunitas Perangkat Lunak Bebas, hal ini berguna untuk

membuat *Ubuntu* dapat dipergunakan oleh banyak orang. Kami juga bekerja sama dengan seluruh komunitas Perangkat Lunak Bebas dalam hal perbaikan *bug* dan saling membagi kode.

4. *Ubuntu* berkomitmen secara penuh terhadap prinsip-prinsip dari pengembangan perangkat lunak bebas; untuk ini kami mendorong masyarakat untuk menggunakan perangkat lunak bebas dan *open source*, lalu memperbaikinya dan kemudian menyebarkannya kembali.

(Komunitas Ubuntu Indonesia, 2009)

2.5 MySQL

Banyak aplikasi yang web developer gunakan untuk mempermudah penggunaan dan standarisasi database seperti menyimpan, mengatur dan mengakses informasi. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai *open source* dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL dapat digabungkan dengan Perl dengan menggunakan Module DBI (*DataBase Independent Interface*). DBI adalah aplikasi program interface (API) yang memungkinkan perlu untuk dikoneksikan ke beberapa jenis SQL database (MySQL, mSQL, PostgreSQL, Oracle, Sybase dan Informix). Banyak aplikasi-aplikasi API dalam bentuk bahasa pemrograman yang memungkinkan

untuk mengakses database MySQL, misalnya: bahasa pemrograman C, C++, C#, Java, Python, PHP dan lain sebagainya.

Dalam penggunaannya MySQL sangat populer dalam aplikasi web seperti PHPTriad, PHP-Nuke, WAMP, LAMP yang berfungsi sebagai komponen database. Popularitasnya sebagai aplikasi web dikarenakan keterkaitannya dengan PHP, sehingga MySQL dan PHP seringkali disebut sebagai *Dinamic Duo*.

Untuk melakukan administrasi dalam database MySQL, dapat menggunakan modul yang sudah termasuk yaitu *command-line* (perintah: MySQL dan MySQLadmin). Juga dapat diambil dari situs MySQL yaitu sebuah modul berbasis grafik (GUI): MySQL Administrator di <http://mysql.com/product/tools/administrator> dan MySQL Query Browser di <http://mysql.com/product/tools/query/browser>. Selain itu terdapat juga sebuah perangkat lunak gratis untuk administrasi basis data MySQL berbasis web yang sangat populer yaitu phpMyAdmin. Untuk perangkat lunak dalam administrasi basis data MySQL yang dijual; secara komersial antara lain: MySQL front, Navicat dan EMS SQL Manager for MySQL.

Beberapa perintah dasar yang sering digunakan dalam MySQL di sistem operasi berbasis Linux antara lain:

```
$ mysql -u root -p
```

Yaitu *command* untuk mencoba koneksi dengan MySQL server sebagai root dengan memasukkan *password root* MySQL pada saat instalasi. Jika output jawaban adalah seperti dibawah ini, berarti server MySQL anda sudah dapat digunakan atau dalam keadaan *running*.

```
root@proxy:~$ mysql -u root -p
```

```

Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.0.67-0ubuntu6 (Ubuntu)
Type 'help' or '\h' for help. Type '\c' to clear the
buffer.
mysql>

```

Untuk mengetahui *command* apa saja yang terdapat pada MySQL, bisa menggunakan perintah help; disertai dengan menekan enter.

```
mysql> help;
```

```

For information about MySQL products and services,
visit:
  http://www.mysql.com/
For developer information, including the MySQL
Reference Manual, visit:
  http://dev.mysql.com/
To buy MySQL Network Support, training, or other
products, visit:
  https://shop.mysql.com/

```

```

List of all MySQL commands:
Note that all text commands must be first on line and
end with ';'
?          (\?) Synonym for `help'.
clear      (\c) Clear command.
connect    (\r) Reconnect to the server. Optional
arguments are db and host.
delimiter (\d) Set statement delimiter. NOTE: Takes the
rest of the line as new delimiter.
edit       (\e) Edit command with $EDITOR.
ego        (\G) Send command to mysql server, display
result vertically.
exit       (\q) Exit mysql. Same as quit.
go         (\g) Send command to mysql server.
help       (\h) Display this help.
nopager    (\n) Disable pager, print to stdout.
notee      (\t) Don't write into outfile.
pager      (\P) Set PAGER [to_pager]. Print the query
results via PAGER.
print      (\p) Print current command.
prompt     (\R) Change your mysql prompt.
quit       (\q) Quit mysql.

```

```

rehash      (\#) Rebuild completion hash.
source      (\.) Execute an SQL script file. Takes a file
             name as an argument.
status      (\s) Get status information from the server.
system      (\!) Execute a system shell command.
tee          (\T) Set outfile [to_outfile]. Append
             everything into given outfile.
use          (\u) Use another database. Takes database
             name as argument.
charset      (\C) Switch to another charset. Might be
             needed for processing binlog with multi-byte
             charsets.
warnings     (\W) Show warnings after every statement.
nowarning    (\w) Don't show warnings after every
             statement.

```

For server side help, type 'help contents'
mysql>

Untuk lebih memahami tentang perintah yang ada diatas, maka akan dijelaskan seperti berikut ini:

1. Perintah show databases dan create database.

Sebelum membuat database baru, lakukan pengecekan database yang sudah ada untuk memastikan bahwa database yang akan dibuat belum ada. Jika sudah dicek dan tidak ada nama database yang akan dibuat maka bisa membuat database baru tersebut:

```

mysql> show databases;

+-----+
| Database |
+-----+
| information_schema |
| dbproxy |
| mysql |
+-----+
3 rows in set (0.08 sec)
mysql>

```

```
mysql> create database mail;
Query OK, 1 row affected (0.07 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mail |
| dbproxy |
| mysql |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Satu hal yang terpikirkan bahwa dalam database terdapat beberapa tabel yang terelasikan. Tiap tabel terdiri dari kolom yang tiap kolomnya menyimpan data untuk satu *record*, setiap *record* berisi beberapa bagian informasi yang disebut *fields*.

2. Perintah Use

Sebelum melakukan sesuatu terhadap database yang baru dibuat, maka tindakan yang harus dilakukan yaitu mengkoneksikan database tersebut terlebih dahulu dengan menggunakan perintah use:

```
mysql> use mail;
Database changed
mysql>
```

3. Perintah show table dan create table

Setiap tabel dalam database harus didefinisikan dan dibuat. Hal ini dapat dilakukan dengan menggunakan perintah create table.

Disini akan dicontohkan membuat tabel dengan nama users yang berisi email, password, dan quota. Pada perintah SQL (MySQL) membutuhkan

informasi jenis data yang akan disimpan pada tiap-tiap field. Pada contoh ini email dan password adalah variabel karakter strings yang memiliki lebar data maksimal 80 variabel karakter dan 20 variabel karakter yang harus berisikan data, dan quota berisi data integer maksimal 10 yang didefinisikan secara *default* sebesar 10485760 atau sebesar 10 Mb dan email digunakan sebagai *primary key*:

```
mysql> use mail;
Database changed
mysql> create table users (
    -> email varchar(80) NOT NULL,
    -> password varchar(20) NOT NULL,
    -> quota INT(10) DEFAULT '10485760',
    -> PRIMARY KEY (email)
    -> )TYPE=MyISAM;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Dengan munculnya keterangan Query OK maka disini tabel telah sukses dibuat, untuk meyakinkan bisa menggunakan perintah `show tables;`, jika terjadi *error* bisa dilakukan dengan menghapus tabel tersebut dengan perintah `drop users;`

4. Perintah Describe

Perintah `describe` memberikan informasi tentang *fields* pada tabel.

```
mysql> describe users;
```

Field	Type	Null	Key	Default	Extra
email	varchar(80)	NO	PRI	N	
password	varchar(20)	NO		N	
quota	int(10)	NO		10485760	

```

+-----+-----+-----+-----+-----+-----+
-+
3 rows in set (0.03 sec)

```

Perintah `show columns from users;` juga akan menghasilkan informasi yang dengan perintah `describe users;` untuk melakukan perintah `describe users;` bisa juga menggunakan singkatan `desc` sehingga perintahnya menjadi `desc users;`

5. Perintah Insert

Supaya tabel yang telah dibuat dapat berguna, perlu menambahkan / mengisi informasi ke dalamnya. Hal ini bisa dilakukan dengan perintah `insert`:

```

mysql> insert into users
-> (email, password
-> varchar(20) NOT NULL,
-> quota INT(10) DEFAULT '10485760',
-> PRIMARY KEY (email)
-> )TYPE=MyISAM;
Query OK, 0 rows affected (0.00 sec)

```

Syntax perintah `insert into`, diikuti nama tabel yang akan diisi, nama field yang akan diisi dan values/isi data yang akan dimasukkan.

6. Perintah Select

Memilih record dalam database MySQL menggunakan perintah `select`. Ketika perintah ini dieksekusi dari shell, MySQL mencetak semua record yang cocok dengan query yang dimasukkan. Perintah yang paling mudah adalah:

```
mysql> select * from users
```

Tanda `*` berarti "menampilkan isi dari semua field pada tabel"; *from* tabel yang akan dibuka. Ada banyak cara untuk menggunakan perintah `select` sesuai dengan pengurutan ataupun kriteria yang akan dituliskan dalam query.

7. Perintah Update

Perintah .untuk merubah informasi yang berada di dalam database atau bisa disebut juga dengan memodifikasi data pada database.

8. Perintah Delete

Dalam sebuah tabel kadang-kadang terdapat *record* yang perlu dihapus.

Untuk menghapus *record* dalam tabel bisa menggunakan perintah delete.

```
mysql> delete from users where email = 'andi';
Query OK, 1 rows affected (0.00 sec)
```

(<http://id.wikipedia.org/wiki/MYSQL>)

2.6 PHP

PHP adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan di sisi *server*. Sebagian besar perintahnya berasal dari C, Java dan Perl dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi Web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP juga dimaksudkan untuk mengganti teknologi lama seperti CGI. (<http://id.wikipedia.org/wiki/PHP>)

2.7 Extjs

ExtJS adalah sebuah library (framework) Javascript yang powerfull yang dapat menyederhanakan pembuatan aplikasi web berbasis AJAX. Selain mempermudah proses request dan response secara asynchronous, ExtJS juga menyediakan komponen-komponen yang bisa kita gunakan untuk membangun antarmuka aplikasi web. Komponen-komponen yang disediakan juga sangat banyak seperti tombol, grid, tab, tree, menu dan lainnya. ExtJS dapat dijalankan

pada semua web browser yang populer saat ini dengan tampilan yang sama antar browser (cross browser). Beberapa web browser yang mendukung ExtJS diantaranya adalah:

- Internet Explorer versi 6 keatas
- Mozilla Firefox versi 1.5 keatas
- Apple Safari versi 2 keatas
- Opera versi 9 keatas

(<http://yudiantosujana.wordpress.com/tag/extjs/>)

2.8 SAMBA

Samba adalah himpunan aplikasi yang bertujuan agar komputer dengan system operasi Linux, BSD (Berkeley Software Distribution) atau Unix dapat bertindak sebagai file dan print server yang berbasis protocol SMB (Session Message Block). Jaringan yang semacam ini biasa dijumpai pada Windows Workgroup atau Windows NT Domain. Samba juga dilengkapi dengan beberapa program bantu sehingga Sistem Operasi Linux (dan Unix lainnya) bisa mengakses sumber daya yang ada pada jaringan Windows yang telah ada. Bisa dikatakan, Samba adalah jembatan penghubung antara Windows dan Unix.

Samba terdiri atas dua program yang berjalan di background: SMBD (Server Message Block Daemon) dan NMBD (NetBIOS Name Block Daemon). Secara singkat dapat disebutkan bahwa SMBD (Server Message Block Daemon) adalah program yang akan menghasilkan proses baru untuk setiap klien yang aktif, sementara NMBD (NetBIOS Name Block Daemon) bertugas

mengkonversi nama komputer (NetBIOS) menjadi alamat IP (Internet Protocol) sekaligus juga memantau share yang ada di jaringan. Kerja SMBD (Server Message Block Daemon) sendiri diatur melalui sebuah file konfigurasi. Dengan membuat file konfigurasi yang tepat, Samba dapat dijadikan file server, print server, Domain Controller, dan banyak fungsi lainnya. (pinguingilo, 2009).

2.9 PROXY

Proxy server adalah sebuah perantara antar berbagai komputer host dalam jaringan (network) yang berbeda, misalnya antara *Local Area Network* (LAN) dan jaringan internet. Proxy server digunakan untuk melakukan *caching* (penampungan sementara) data, dan untuk pengendalian administrasi dan keamanan. Selain itu Proxy server juga dapat diartikan sebuah komputer server atau program komputer yang dapat bertindak sebagai komputer lain untuk melakukan request terhadap content dari internet dan intranet

Proxy server dapat memiliki berbagai fungsi, seperti *proxy caching* dan *Network Address Translation* (NAT). Sesuai dengan namanya, proxy server bertindak sebagai server kepada sekelompok *client*, yaitu memberikan pelayanan sebagai proxy kepada berbagai komputer host yang membutuhkan pelayanan tersebut. Pada model referensi OSI (Open System Interconnection) yang memiliki 7 layer, proxy server merupakan layanan yang jalan pada layer Application (layer 7). Karena proxy bekerja pada layer aplikasi, proxy server dapat berjalan pada banyak aplikasi antara lain HTTP Proxy atau Web Proxy untuk protokol HTTP atau Web, FTP Proxy, SMTP/POP Proxy untuk email, NNTP proxy untuk Newsgroup, RealAudio/RealVideo Proxy untuk multimedia streaming, IRC proxy

untuk Internet Relay Chat (IRC), dan lain-lain. Masing-masing hanya akan menerima, meneruskan atau melakukan filter atas paket yang dihasilkan oleh layanan yang bersesuaian. Proxy aplikasi spesifik memiliki pilihan konfigurasi yang sangat banyak. Sebagai contoh, Web Proxy dapat dikonfigurasi untuk menolak akses ke situs web tertentu pada waktu-waktu tertentu. Demikian juga proxy yang lain, misalnya dapat dikonfigurasi untuk hanya memperbolehkan download FTP dan tidak memperbolehkan upload FTP, hanya memperbolehkan pengguna tertentu yang bisa memainkan file-file RealAudio, mencegah akses ke email server sebelum tanggal tertentu, dan masih banyak lagi.

Proxy server juga sangat baik dalam hal kemampuan menyimpan catatan (logging) dari trafik jaringan, dan dapat digunakan untuk memastikan bahwa koneksi untuk jenis trafik tertentu harus selalu tersedia. Sebagaimana biasa, kelemahan dari konfigurasi yang sangat fleksibel dan banyak pilihan adalah timbulnya kompleksitas. Aplikasi pada sisi pengguna seperti Web Browser atau RealAudio Player harus ikut dikonfigurasi untuk bisa mengetahui adanya proxy server dan bisa menggunakan layanannya. Bila suatu layanan baru dibuat di internet yang berjalan pada layer aplikasi, dengan menggunakan protokol baru dan port yang baru, maka harus dibuat juga proxy yang spesifik dan bersesuaian dengan layanan tersebut. Proses penambahan pengguna dan pendefinisian aturan akses pada suatu proxy juga bisa sangat rumit.

Proxy server akan bertindak sebagai gateway terhadap dunia internet untuk setiap komputer *client*. Dalam menjalankan tugasnya proxy server tidak terlihat oleh komputer *client* sebagai contoh saat seorang pengguna yang berinteraksi dengan Internet melalui sebuah proxy server tidak akan mengetahui bahwa sebuah

proxy server sedang menangani request yang dilakukannya. Web server yang menerima request dari Proxy server akan menginterpretasikan request-request tersebut seolah-olah datang secara langsung dari komputer *client*, bukan dari proxy server.

Sebagai perantara antara pengguna dan server-server di internet, proxy server bekerja dengan cara menerima permintaan layanan dari *user*, dan kemudian sebagai gantinya proxy server akan mewakili permintaan pengguna, ke server-server di internet yang dimaksudkan. Dengan demikian, sebenarnya proxy server hanya meneruskan permintaan pengguna ke server yang dimaksud, akan tetapi disini identitas peminta sudah berganti, bukan lagi pengguna asal, tetapi proxy server tersebut. Server-server di internet hanya akan mengetahui identitas proxy server tersebut, sebagai yang meminta, tetapi tidak akan tahu peminta sebenarnya (yaitu pengguna asalnya) karena permintaan yang sampai kepada server-server di internet bukan lagi dari pengguna asal, tetapi dari proxy server.

Bagi pengguna sendiri, proses yang terjadi pada proxy server diatas juga tidak kelihatan (transparan). Pengguna melakukan permintaan atas layanan-layanan di internet langsung kepada server-server layanan di internet. Pengguna hanya mengetahui keberadaan atau alamat dari proxy server, yang diperlukan untuk melakukan konfigurasi pada sisi pengguna untuk dapat menggunakan layanan dari proxy server tersebut.

Proxy server dalam suatu jaringan memiliki tiga fungsi utama yaitu sebagai Connection sharing, filtering, dan caching.

- Connection Sharing

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu gateway, yang bertindak sebagai batas antara jaringan lokal (privat) dan jaringan luar (publik). Gateway ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas antara jaringan lokal dan internet. Gateway juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh gateway secara bersama-sama (connection sharing). Dalam hal ini, gateway adalah juga sebagai proxy server, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet. singkatnya : 1 IP public dapat digunakan oleh banyak *user*, selain itu juga untuk melindungi jaringan dalam dari serangan luar.

- Filtering

Merupakan sebuah usaha pengamanan atau pembatasan sehingga dengan adanya filtering sebuah proxy server dapat mengamankan dan membatasi hak akses *client* pada jaringan privat. Jadi meskipun mula-mula dibuat sebagai cache nonsekuriti, tujuan utama proxy server sekarang menjadi firewalling. Proxy server memperbarui request layanan pada jaringan eksternal atas nama *client* mereka pada jaringan private. Ini secara otomatis menyembunyikan identitas dan jumlah *client* pada jaringan internal dari jaringan eksternal. Karena posisi mereka di

antara *client* internal dan server publik, proxy juga dapat menyimpan content yang sering diakses dari jaringan publik untuk mengurangi akses ke jaringan publik tersebut. Kebanyakan implementasi nyata proxy sekuriti meliputi pemfilteran paket dan Network Address Translation untuk membangun firewall yang utuh. Teknologi tersebut dapat digabungkan dengan proxy untuk menghilangkan serangan yang terhadapnya proxy rentan.

Manfaat dari proxy server adalah bahwa proxy administrator dapat membuat berbagai aturan (rules) untuk melakukan filtering (pemfilteran) terhadap data yang diakses lewat proxy server. Sebagai contoh, kita dapat melakukan pemfilteran terhadap berbagai permintaan halaman web yang mengandung kata – kata tertentu pada alamat URL web servernya. Misalnya saja, jika tidak ingin *user* tidak bisa mengakses situs – situs yang URLnya terdapat kata “go”, maka kita tinggal membuat berbagai aturan di proxy server untuk melakukan pemfilteran (memblock) semua situ situ, misalnya saja <http://www.google.com>

- Caching (Internet Object caching)

Caching adalah suatu cara untuk menyimpan hasil permintaan internet-object. (seperti: data yang ada dari HTTP, FTP, dan gopher protokol) untuk membuat sistem dekat dengan permintaan daripada ke sumber aslinya. Web browser dapat menggunakan lokal squid cache sebagai proxy HTTP server, ini akan mengurangi waktu akses seperti halnya penghematan bandwidth. Dengan kata lain sebuah *client* tidak harus melakukan kontak dengan server untuk meminta layanan akan tetapi *client* dapat mendapatkan layanan (*data*) yang sudah tersimpan pada proxy server, dengan hal ini maka akses akan semakin cepat.

Salah satu tugas utama dari sebuah proxy server adalah untuk melakukan caching (penampungan data sementara) terhadap halaman – halaman web dan berbagai file FTP untuk proxy *client*. Jenis proxy server ini disebut sebagai *proxy cache server*. Caching meningkatkan kinerja jaringan dengan cara mengurangi jumlah data yang ditransfer dari luar (misalnya dari internet) ke dalam jaringan lokal.

Untuk mengimplementasikan proxy caching, tiap *client* / *workstation* dalam jaringan local perlu dikonfigurasi sebagai proxy *client* untuk pelayanan (service) tertentu, sehingga dapat mengakses dan menggunakan proxy cache server. Sebagai contoh, agar sebuah *workstation* dapat menjadi sebuah web proxy *client*, maka web browser untuk *workstation* tersebut perlu dikonfigurasi setting proxy-nya sehingga mengakses internet lewat proxy cache server dan tidak mengaksesnya secara langsung. Pada saat *client* / *workstation* membuat suatu permintaan lewat web browser (HTTP request) untuk men-download suatu halaman web, web browser tersebut akan mengirim bukan langsung ke web server yang dituju, tetapi ke proxy cache server. Proxy cache server menampung berbagai halaman web yang pernah diakses sebelumnya di dalam cache-nya. Cache ini menampung berbagai halaman web yang telah diakses oleh semua *workstation* yang ada di jaringan lokal.

Proxy cache server memeriksa cache-nya untuk memastikan apakah halaman web yang ingin diakses sudah ada atau belum di dalam cache-nya. Jika halaman web itu sudah ada di dalam cache, maka halaman web tersebut langsung diambil dari cache dan dikirim ke proxy *client* yang membuat permintaan untuk halaman web tersebut. Jika halaman web yang diperlukan belum ada dalam cache,

maka proxycache server langsung men-download halaman web itu dari web server, memasukkannya ke dalam cache, dan kemudian mengirimnya ke *client / workstation* yang membuat permintaan untuk halaman web tersebut.

Untuk menjamin bahwa berbagai halaman yang ada di proxy selalu up-to-date, data yang ada di cache akan “kadaluwarsa” setelah masa waktu tertentu. Misalnya dalam program squid (sebuah program / aplikasi proxy cache server), setting masa waktu ini disebut sebagai *object refresh time*. Refresh time ini digunakan untuk menjamin bahwa data yang sudah ada (lama) tidak akan di kirim ke proxy *client*.

Proses yang telah disebutkan di atas meningkatkan kinerja jaringan karena halaman web secara tidak langsung dan instant di kirim ke *client* dari proxy cache server tanpa harus mengambilnya dari Internet tiap kali. Hal ini mempercepat akses Internet dari jaringan local dan menghemat *bandwidth*. Hanya dengan menyediakan tempat penampungan sementara untuk data yang sering diakses dari Internet, kecepatan browsing dapat ditingkatkan secara drastis dan penggunaan *bandwidth* jaringan dapat menjadi lebih efisien. (Wagito, edisi pertama 2005)

BAB III

METODOLOGI PRAKTEK KERJA LAPANGAN

Pada bab ini, dibahas mengenai metode praktek kerja lapangan yang dilakukan penulis di SMK Negeri 6 Surabaya, baik dari waktu dan tempat penelitian, pelaksanaan, software, dan hardware yang digunakan.

3.1 Tempat dan waktu penelitian

Tempat	: Ruang Server SMK Negeri 6 Surabaya.
Waktu penelitian	: 1 Maret 2009 – 31 Maret 2009

3.2 Hardware dan Software

3.2.1 Komputer Router

Software	: Operating System : <i>Ubuntu Server Intrepid Ibex 8.10</i> Software : VMware Workstation 6.0
Spesifikasi Komputer	: Memory 512 MB <i>Harddisk 3 Gb</i> CDROM Ethernet 0 Ethernet 1 Display Processor

3.2.2 Komputer Server

Software : Operating System :
Ubuntu Server Intrepid Ibex 8.10
 Software : VMware Workstation 6.0

Spesifikasi Komputer : *Harddisk 3 Gb*
CDROM
Ethernet 0
Display
Processor

3.3 Pelaksanaan Kerja Praktek

Praktek Kerja Lapangan ini dilaksanakan di SMK Negeri 6 Surabaya dengan jadwal pelaksanaan yang disusun oleh penulis dan dosen pembimbing lapangan sesuai dengan Tabel 3.1

Tabel 3.1 Pelaksanaan Kegiatan

<i>Nomor</i>	<i>Hari/Tanggal</i>	<i>Kegiatan</i>
1	Senin, 4 Maret 2009	Pengkabelan
2	Rabu, 11 Maret 2009	<i>Install</i> Ubuntu Server Intrepid Ibex 8.10
3	Jumat, 12 Maret 2009	<i>Install</i> DNS, DHCP, Samba, Proxy
4	Selasa, 13 Maret 2009	Mengkonfigurasi ip address, dhcp dan dns server
5	Rabu, 18 Maret 2009	Mengkonfigurasi dan membuat user samba
6	Senin, 25 April 2009	Mengkonfigurasi dan membuat user proxy
7	Kamis, 31 April 2009	Testing

BAB IV

HASIL DAN PEMBAHASAN

Bab ini akan menjelaskan hasil dan pembahasan yang didapat dari pembangunan jaringan komputer berbasis Ubuntu Server Intrepid Ibex 8.10 di SMK Negeri 6 Surabaya.

4.1 Perancangan Sistem

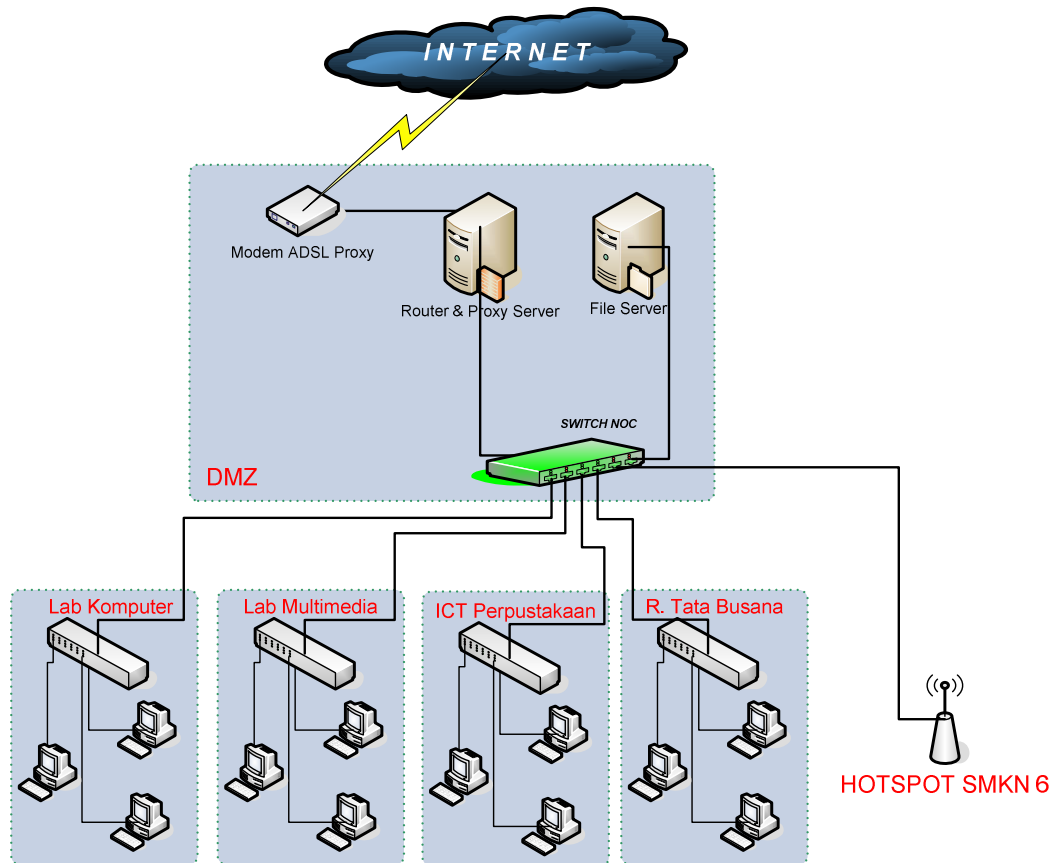
Sub-bab ini akan menjelaskan mengenai proses desain dari sistem jaringan yang akan dibuat. Proses desain sistem dalam sub-bab ini akan dibagi menjadi beberapa tahapan yang akan dijelaskan satu persatu berikut ini.

4.1.1 Perencanaan Konsep Jaringan

SMK Negeri 6 Surabaya sebagai sekolah menengah atas yang dibilang ikut mengikuti perkembangan teknologi informasi belum memiliki jaringan komputer yang memadai walaupun sudah ada fasilitas komputer di dalamnya. Sehingga tempat untuk penyimpanan data/file yang aman, bisa digunakan oleh *user*, dapat dan mempunyai stabilitas yang tinggi belum terwujud. Karena hampir semua hal yang dilakukan seperti penyimpanan data/file, pindah data dan lain sebagainya di SMK Negeri 6 Surabaya dilakukan secara manual, maka perlu dibangun sistem jaringan komputer yang mampu menganani permasalahan tersebut.

Secara umum, desain sistem jaringan komputer yang akan dibangun di SMK Negeri 6 Surabaya yang dapat memenuhi kebutuhan guru dan siswa untuk

bisa mengakses internet dan dapat melakukan penyimpanan(*transfer*) data/file di server secara aman dan lancar dijelaskan melalui gambar berikut:



Gambar 4.1 Desain Jaringan

Pada desain jaringan diatas, terdiri atas empat buah komputer yang terdiri dari komputer server, komputer router, komputer client (koneksi kabel), dan komputer client (*wireless*). Yang penjelasannya sebagai berikut:

1. Komputer Router

Komputer router disini merupakan komputer yang mencakup dua server sekaligus, yaitu sebagai DHCP server dan Proxy server. Komputer router ini sebagai penyedia layanan IP secara otomatis (DHCP) dan juga sebagai tempat permintaan data internet milik *user(client)* serta sebagai filtering dari suatu situs.

2. Komputer Server

Komputer server disini merupakan komputer yang memiliki tugas sebagai tempat penyimpanan data user (*file server*) dalam hal ini siswa, guru, dan admin.

3. Komputer Client (koneksi kabel)

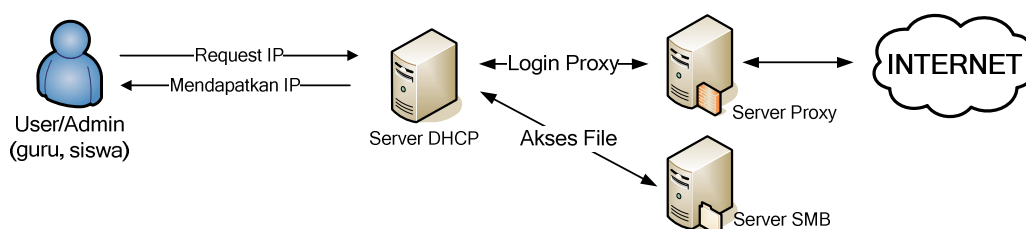
Komputer Client disini dibuat seperti komputer-komputer client yang lain seperti yang ada pada lingkungan jaringan yang terdapat pada jaringan SMK Negeri 6 Surabaya. Hanya saja pada komputer client ini hanya digunakan pada jaringan *Local Area Network* SMK Negeri 6 Surabaya.

4. Komputer Client (*wireless*)

Komputer Client disini dibuat seperti komputer-komputer client yang lain seperti yang ada pada lingkungan jaringan *wireless* yang terdapat pada jaringan SMK Negeri 6 Surabaya.

4.1.2 Perancangan Proses

Perancangan Proses digunakan untuk menggambarkan sejumlah proses terstruktur yang terjadi dalam sistem. Proses atau alur yang dilakukan di dalam sistem jaringan komputer SMK Negeri 6 Surabaya ini akan dijelaskan pada gambar berikut.



Gambar 4.2 Proses/Alur Sistem Jaringan

Proses yang terjadi dalam gambar diatas dapat dijelaskan sebagai berikut :

1. *User* (Siswa) mengakses menggunakan komputer merequest IP dan selanjutnya mendapatkan IP dari DHCP server. Selanjutnya user tersebut dapat mengakses file yang ada di Samba Server dan juga koneksi internet melalui login pada Proxy Server.
2. *User* (Guru) mengakses menggunakan komputer merequest IP dan selanjutnya mendapatkan IP dari DHCP server. Selanjutnya user tersebut dapat mengakses file yang ada di Samba Server dan juga koneksi internet melalui login pada Proxy Server.
3. *Administrator* mengakses menggunakan komputer merequest IP dan selanjutnya mendapatkan IP dari DHCP server. Selanjutnya admin tersebut dapat mengakses file dan mempunyai hak akses penuh yang ada di Samba Server dan juga koneksi internet melalui login pada Proxy Server.

4.2 Implementasi

Pada sub-bab implementasi ini akan dijelaskan tentang langkah-langkah konfigurasi atau cara pengaturan yang dilakukan pada masing-masing komputer sesuai dengan konsep desain jaringan yang sudah dibuat, agar pengujian yang dilakukan pada Praktek Kerja Lapang ini bisa berjalan sesuai dengan yang diharapkan.

4.2.1 Konfigurasi Pada Komputer Router

Komputer *Router* terdiri dari dua server yaitu DHCP server dan Proxy server. Komputer *Router* adalah suatu komputer yang digunakan untuk menerima

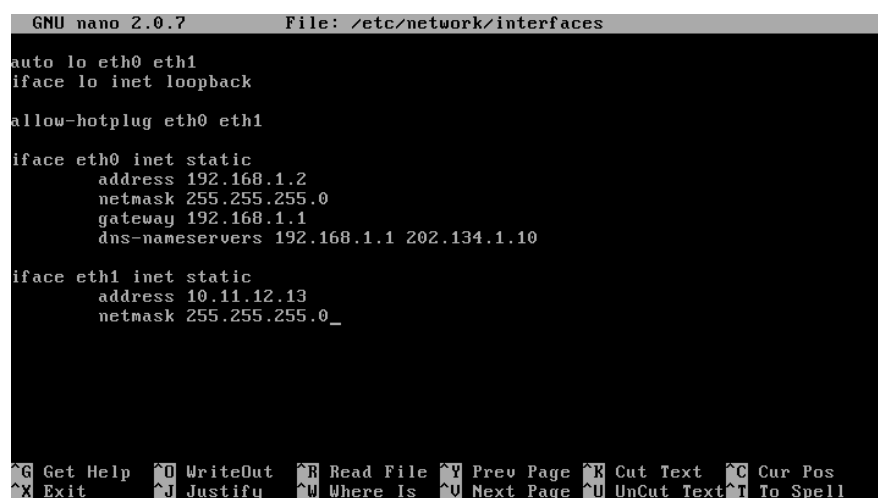
setiap request atau permintaan IP yang diberikan oleh komputer user agar komputer *Router* dapat memberikan servis atau layanan-layanan dengan tujuan supaya komputer client bisa mendapatkan permintaan seperti yang komputer client inginkan. Adapun tahapan-tahapan yang perlu dilakukan supaya komputer *Router* tersebut bisa berjalan seperti konsep *Router* yang sebenarnya.

4.2.1.1 Pemasangan IP Address

Pada sistem operasi Linux, pemasangan IP Address dapat dilakukan dengan beberapa cara sesuai dengan kebutuhan yaitu dengan menggunakan perintah *ifconfig*, *ip addr* dan mengedit file *interfaces* pada direktori “*/etc/network*”. Namun karena penulis menginginkan pemasangan IP Address yang bersifat permanen, maka penulis melakukan cara ketiga yaitu mengedit file *interfaces* pada direktori “*/etc/network*”. Adapun perintah yang dilakukan adalah:

```
root:~# nano /etc/network/interfaces
```

Router mempunyai dua kartu jaringan yaitu *eth0* dan *eth1*. Masing-masing dengan alamat IP seperti pada gambar dibawah ini.



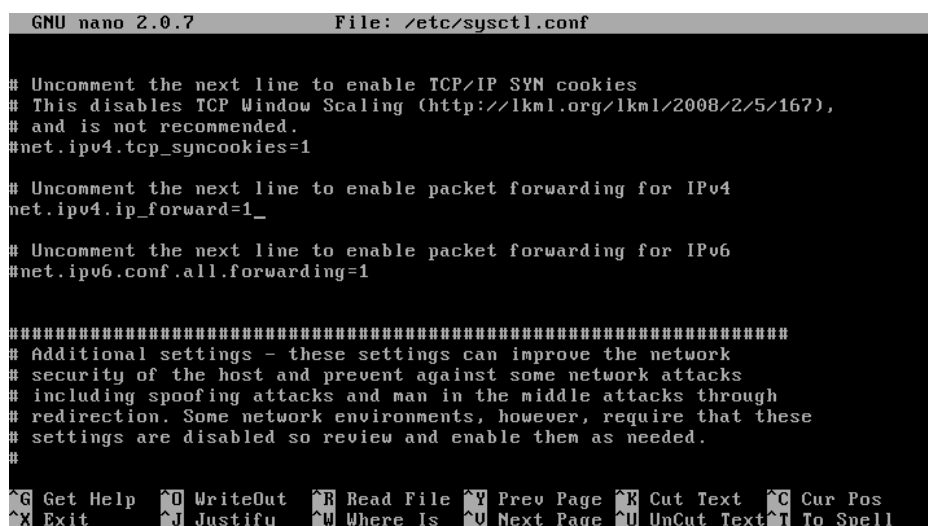
```
GNU nano 2.0.7 File: /etc/network/interfaces
1 auto lo eth0 eth1
2 iface lo inet loopback
3
4 allow-hotplug eth0 eth1
5
6 iface eth0 inet static
7     address 192.168.1.2
8     netmask 255.255.255.0
9     gateway 192.168.1.1
10    dns-nameservers 192.168.1.1 202.134.1.10
11
12 iface eth1 inet static
13     address 10.11.12.13
14     netmask 255.255.255.0_
```

Gambar 4.3 IP Router

Selanjutnya simpan pemasangan IP Address dengan menggunakan tombol “ctrl+x” pada keyboard, tekan tombol “y”. Untuk pengujian lakukan perintah “ping [IP Address]”, maka pemasangan IP komputer *Router* berhasil.

Lanjutnya pada router harus diberi kemampuan forward alamat ip dengan cara mengedit file `sysctl.conf` pada direktory “`/etc/`”, lalu hilangkan tanda pagar ‘#’ pada baris “`net.ipv4.ip_forward=1`” :

```
root:~# nano /etc/sysctl.conf
```



```
GNU nano 2.0.7      File: /etc/sysctl.conf

# Uncomment the next line to enable TCP/IP SYN cookies
# This disables TCP Window Scaling (http://lkm1.org/lkm1/2008/2/5/167),
# and is not recommended.
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1_

# Uncomment the next line to enable packet forwarding for IPv6
#net.ipv6.conf.all.forwarding=1

#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^U Next Page  ^U UnCut Text ^T To Spell
```

Gambar 4.4 IP Forward

4.2.1.2 Pemasangan DHCP Server dan DNS Forward

DHCP server merupakan suatu paket software yang berfungsi untuk mengatur request atau permintaan IP yang dikirimkan oleh komputer user. *DHCPserver* memiliki beberapa fungsi ataupun kemampuan memberikan permintaan IP Address, netmask, default gateway, dan dns Persiapan yang dilakukan sebagai berikut.

1. Instalasi *DHCP Server*

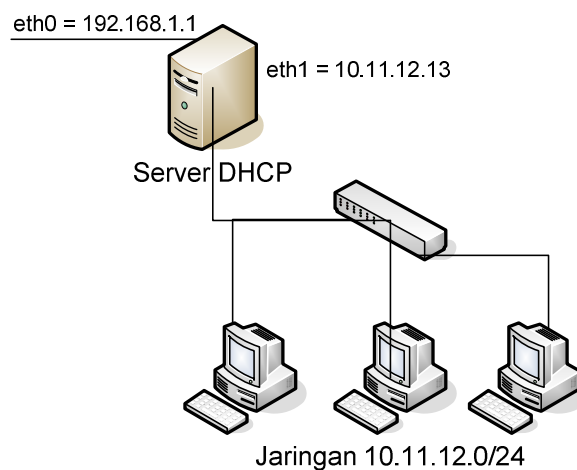
DHCP merupakan salah satu dari beberapa paket software *dhcp3-server* dan *dns* yang sangat populer di dunia karena sifatnya yang *open source*. Adapun cara instalasinya pada *Linux Ubuntu* sebagai berikut:

```
root:~# apt-get install dhcp3-server
```

```
root@router-smkn6:~# apt-get install dhcp3-server bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bind9utils
Suggested packages:
  bind9-doc resolvconf
The following NEW packages will be installed:
  bind9 bind9utils dhcp3-server
0 upgraded, 3 newly installed, 0 to remove and 23 not upgraded.
Need to get 703kB of archives.
After this operation, 1913kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://10.11.12.1 intrepid/main bind9utils 1:9.5.0.dfsg.P2-1ubuntu2 [90.0k
B]
Get:2 http://10.11.12.1 intrepid/main bind9 1:9.5.0.dfsg.P2-1ubuntu2 [243kB]
31% [2 bind9 128612/243kB 52%]_
```

Gambar 4.5 Instalasi Dhcp3-server dan Bind9

Berikut ini diberikan suatu struktur jaringan dengan DHCP server dalam jaringan.



Gambar 4.6 Server DHCP

Konfigurasi server DHCP diatur dalam file `dhcpd.conf` yang terletak dalam direktori `/etc/dhcp3`. Isi dan susunan file `dhcpd.conf` sangat tergantung pada struktur jaringan yang akan dikonfigurasi melalui protokol DHCP.

```
root:~# nano /etc/dhcp3/dhcpd.conf
ddns-update-style none;
subnet 10.11.12.0 netmask 255.255.255.0 {
    range 10.11.12.20 10.11.12.30;
    option domain-name-servers 10.11.12.13;
    option domain-name "smkn6.sch.id";
    option routers 10.11.12.13;
    option broadcast-address 10.11.12.255;
    default-lease-time 6000;
    max-lease-time 72000;
}
```

Konfigurasi diatas mengatur bahwa dalam jaringan terdapat satu subnet dengan alamat jaringan (*network address*) 10.11.12.0 dengan netmask 255.255.255.0 yaitu jaringan kelas A. Jaringan yang diatur dengan layanan DHCP ini mempunyai range alamat IP 10.11.12.20 sampai 10.11.12.30. Keterangan lain tentang konfigurasi `dhcpd.conf` adalah sebagai berikut :

- a. *subnet* : menentukan subnet jaringan
- b. *netmask* : menentukan netmask subnet.
- c. *range* : menentukan range alamat IP.
- d. *option domain-name-server* : menentukan server DNS untuk subnet.
- e. *option domain-name*: menentukan nama domain untuk subnet
- f. *option routers* : menentukan default gateway komputer klien.
- g. *option broadcast-address* : menentukan broadcast untuk subnet.
- h. *default-lease-time* : menentukan waktu lease dalam detik.
- i. *max-lease-time* : menentukan waktu lease maximum dalam detik.

Semua baris konfigurasi diatas diperlukan. Jika beberapa baris konfigurasi tidak diperlukan, maka boleh dihilangkan. Untuk menjalankan layanan server DHCP pada kartu jaringan, di gunakan perintah berikut ini :

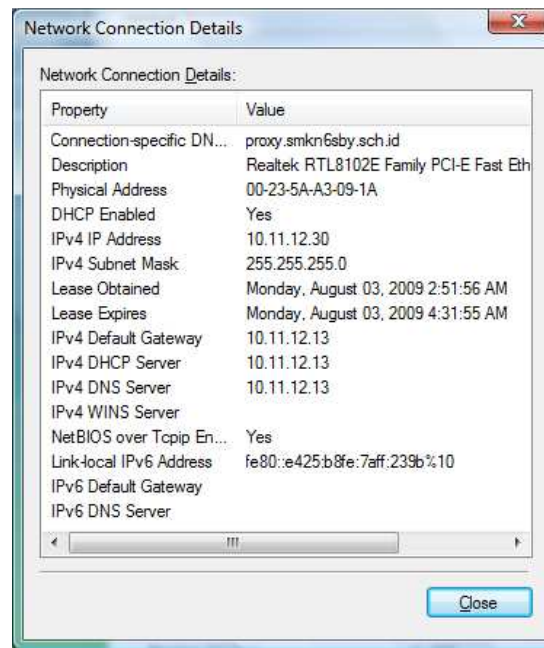
```
root:~# /etc/init.d/dhcp3-server restart
```

Perintah diatas akan mengkonfigurasi alamat IP kartu jaringan eth1 secara otomatis berdasarkan server DHCP 10.11.12.13.

Selanjutnya, untuk mengetahui apakah *DHCP* sudah berhasil terinstall dengan baik pada komputer user menggunakan windows xp adalah dengan cara :

- a. Klik start → control panel → Network Conections
- b. Lalu klik kanan pada “Local area conections”
- c. Muncul kotak dialog “Local area conections properties”
- d. Klik TCP/IP lalu klik properties
- e. Muncul kotak dialog “TCP/IP properties’
- f. Pilih / klik “ obtain an IP address automatically “
- g. Klik ok

Sehingga hasilnya seperti pada gambar di bawah ini.



Gambar 4.7 Pengujian IP Address DHCP

2. Instalasi DNS

Adapun cara instalasinya pada *Linux Ubuntu* pada Gambar 4.5 :

Server DNS yang disusun kali ini hanya menyediakan informasi tentang forward domain tertentu saja, dalam hal ini domain 10.11.12.13 . Jika ada permintaan informasi tentang suatu domain yang tidak dikelola oleh server DNS tersebut, maka server DNS harus mencari informasinya ke server DNS lainnya yang lebih luas. Dalam hal ini antar server DNS harus berhubungan secara hirarki.

Mekanisme yang digunakan server DNS untuk mencari informasi suatu domain yang tidak dikelolanya adalah melalui mekanisme *forward*. Jika informasi tidak dapat ditemukan pada server DNS, maka akan dilakukan *forward* ke server DNS lain. Server DNS ini sebagai acuan dari server DNS yang dibuat. Jika server DNS yang dijadikan acuan juga tidak menemukan informasinya, server DNS

acuan juga melakukan *forward* ke server DNS yang lebih tinggi lagi yaitu server DNS dari ISP.

Mekanisme *forward* ini dapat dilakukan dengan mengaktifkan opsi *forwarders* dalam file konfigurasi sebagai berikut :

```
root:~# nano /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses
    replacing
    // the all-0's placeholder.

    forwarders {
        192.168.1.254;
        202.134.1.10;
    };

    auth-nxdomain no;      # conform to RFC1035
    listen-on-v6 { any; };

};
```

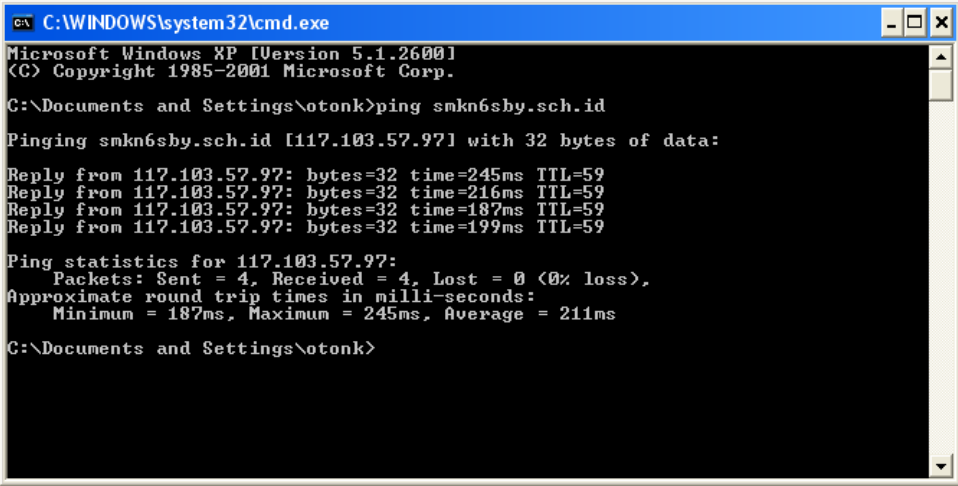
Pada konfigurasi diatas, alamat IP server DNS yang dijadikan acuan ada dua 192.168.1.254 dan 202.134.1.10. Alamat IP server DNS yang dijadikan acuan sebaiknya lebih dari satu, supaya jika salah satu server DNS sedang tidak berfungsi, maka dapat menggunakan server DNS lainnya.

Untuk melakukan restart pada server DNS digunakan perintah berikut ini :

```
root:~# /etc/init.d/bind9 restart
```

3. Pengujian

Untuk *DNS* sudah berhasil terinstall dengan baik pada komputer user menggunakan windows xp adalah dengan cara :



```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\otonk>ping smkn6sby.sch.id

Pinging smkn6sby.sch.id [117.103.57.97] with 32 bytes of data:

Reply from 117.103.57.97: bytes=32 time=245ms TTL=59
Reply from 117.103.57.97: bytes=32 time=216ms TTL=59
Reply from 117.103.57.97: bytes=32 time=187ms TTL=59
Reply from 117.103.57.97: bytes=32 time=199ms TTL=59

Ping statistics for 117.103.57.97:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 187ms, Maximum = 245ms, Average = 211ms

C:\Documents and Settings\otonk>

```

Gambar 4.8 Tes Koneksi

4.2.1.3 Pemasangan Transparan Proxy

Pada penginstalan proxy memerlukan paket-paket pendukung yang akan membantu jalannya proxy “squid”.

```
root:~# apt-get install squid
```

```

root@router-smkn6:~# apt-get install squid
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  openssl-blacklist squid-common ssl-cert
Suggested packages:
  squidclient squid-cgi logcheck-database resolvconf smbclient winbind
The following NEW packages will be installed:
  openssl-blacklist squid squid-common ssl-cert
0 upgraded, 4 newly installed, 0 to remove and 23 not upgraded.
Need to get 7542kB of archives.
After this operation, 19.5MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://10.11.12.1 intrepid/main openssl-blacklist 0.4.2 [6337kB]
0% [1 openssl-blacklist 33543/6337kB 0%] 5529B/s 22min37s_

```

Gambar 4.9 Instalasi Squid

Konfigurasi squid ini dilakukan hanya sebagai percobaan akan berjalannya squid yang akan diberi autentifikasi dengan *mysql_auth*. Untuk itu dibuat konfigurasi sederhana terlebih dahulu yang sesuai dengan jaringan yang ada pada SMK Negeri 6 Surabaya. Disini menggunakan jaringan yang satu subnet

(10.11.12.0/24), dimana proxy terpasang dengan IP 10.11.12.13 dan bersifat *transparent*.

Di dalam file squid.conf terdapat banyak *direktif* atau *option* yang tidak dapat kita gunakan semuanya dan biasanya opsi ini diawali dengan tanda # (*hash*) yang berarti keterangan/penjelasan ataupun komentar yang tidak akan dijalankan oleh sistem, akan tetapi jika kita hilangkan/hapus tanda # (*hash*) maka hal ini menjadi suatu perintah yang akan dijalankan.

Untuk mendapatkan proxy squid yang sesuai dengan kebutuhan maka perlu merubah isi dari konfigurasi squid.conf yang terletak pada “/etc/squid/squid.conf”. Berikut bagian-bagian yang harus di konfigurasi dan penjelasannya agar squid yang sesuai dengan permasalahan praktek kerja lapangan :

```
http_port 8080 transparent
icp_port 3130
```

Pada http_port ditetapkan pada port 8080 merupakan *port* yang melayani paket http, sedangkan untuk *port* “icp_port” ditetapkan secara *default* nya yaitu 3130.

Cache_mem akan menentukan batas atas jumlah memori yang digunakan untuk menyimpan *intransit objek* yaitu obyek yang dalam masa transisi antara waktu *cache mendownload* sampai obyek disampaikan ke *client* dan *hot object* yaitu obyek yang sering diakses.

```
cache_mem 32 MB
```

Bagian ini berisikan besarnya memory yang akan digunakan oleh squid untuk menyimpan objek yang keluar dan masuk melalui squid. Besar angka yang aman untuk dipakai menyimpan obyek adalah $\frac{1}{4}$ dari memori yang ada.

```
access cache_dir ufs /var/spool/squid 5000 10 256
access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log
```

Perintah pada bagian ini akan menentukan dimana file “log-log” atau data *request* akan disimpan. Pada “access_log” akan diletakkan pada folder “/var/log/squid/” dan nama file “access.log”. Sedangkan “cache_log” juga akan diletakkan di folder yang sama dengan access.log, namun dengan nama file yang berbeda yaitu “cache.log”. Access dan cache log berupa data yang berisikan log yang di request oleh tiap client. Dan untuk “cache_dir ufs” merupakan folder yang akan digunakan sebagai tempat menyimpan cache atau objek website. Cache_dir ini dijadikan dengan tipe “ufs” lalu squid adalah nama folder-nya yang terletak pada “/var/spool/”. Lalu ukuran cache ditetapkan sebesar 1000 MB, untuk 16 dan 256 merupakan jumlah folder yang terdapat di dalam cache pada level 1 dan 2.

```
maximum_object_size 4096 KB
minimum_object_size 0 KB
```

Dengan option ini, ukuran file maksimum yang disimpan oleh squid cache bisa dibatasi. Dengan kata lain objek yang lebih besar dari bilangan ini tidak akan disimpan ke dalam disk yang sudah disisihkan untuk cache. Bagian ini menjelaskan bahwa objek yang dapat ditampung secara maksimal adalah 4096 KB, sedangkan untuk *minimum_objek* merupakan besaran objek yang paling kecil yang akan disimpan dalam cache yang ditetapkan sebesar 0 KB. Dengan option ini maka objek yang lebih kecil dari bilangan yang disebutkan tidak akan disimpan ke dalam disk yang sudah disisihkan untuk cache. Ukuran file minimum yang dicache bertujuan mengoptimalkan *filesystem* di mana biasanya ext2 diformat dengan 4 KB inode.

“Access control list” merupakan penempatan-penempatan kebijakan dalam pemberian hak akses kepada “client”nya. Penulisan untuk “acl” sebagai berikut; “acl [nama control list] [tipe control]”. Berikut merupakan pengaturan default dari squid.conf:

```
acl all src 0.0.0.0/0.0.0.0
acl smkn6 src 10.11.12.0/24
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
```

Pada bagian “acl” access control list merupakan penerapan kebijakan yang pertama, menetapkan “acl all src 0.0.0.0/0.0.0.0” yang berarti nilai IP yang dapat akses merupakan IP dengan source IP 0.0.0.0. Dengan “acl manager proto cache_object” merupakan default dari squid.conf tersebut. Untuk akses localhost ditetapkan dengan “acl localhost scr 127.0.0.1/255.255.255.255” , sama dengan dengan “acl all” namun IP diberikan untuk localhost itu sendiri yaitu 127.0.0.1.

Pada pemberian kebijakan acl yang kedua mengenai port-port yang akan digunakan untuk akses client dari proxy. Berikut acl yang dibutuhkan sesuai dengan keperluan yang ada:

```
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
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 Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
```

Dari keterangan dari tiap acl yang sudah dapat diartikan bahwa akan menggunakan port 443 untuk data paket https, 563 untuk data paket snews, 873

untuk data paket rsync, 80 untuk data paket http, 21 untuk data paket ftp, 70 untuk data paket gopher, 210 untuk data paket wais, 1025-65535 untuk data paket yang “unregistered port” , 280 untuk data paket http-mgmt, 488 untuk data paket gss-http, 591 untuk data paket filemaker, 777 untuk data paket multiling, 631 untuk data paket cups, dan 901 untuk data paket SWAT.

Setelah semua port yang telah ditetapkan, maka perlu ditetapkan juga control list untuk metodenya menjadi

```
acl purge method PURGE
acl CONNECT method CONNECT
```

Untuk menetapkan jaringan yang dapat mengakses atau tidak dapat dipisahkan pada bagian ini. Dengan memisahkan “range” IP yang akan diberikan atau tidak. Berikut konfigurasi yang sesuai dengan kondisi yang sudah ditetapkan yaitu pada subnet 10.11.12.0/24

```
acl smkn6 src 10.11.12.0/24
```

Pada konfigurasi di atas telah ditetapkan bahwa akan memakai satu buah jaringan yaitu pada jaringan 10.11.12.0 dengan netmask 255.255.255.0.

Untuk memberikan hak akses dari tiap acl yang telah dibuat, maka perlu juga ditetapkan perintah untuk “allow” atau “deny” dari tiap-tiap acl yang telah dibuat sebelumnya. Berikut penulisan dari pemberian hak akses;

```
http_access/icp_access [allow/deny] [nama acl]
```

Untuk penetapan hak akses yang akan diterapkan pada permasalahan ini adalah sebagai berikut:

```
http_access allow manager localhost
http_access deny manager
http_access allow smkn6
http_access allow purge localhost
http_access deny purge
http_access deny CONNECT !SSL_ports
http_access allow localhost
http_access deny all
```

```
http_reply_access allow all
icp_access allow all
```

Setelah `squid.conf` dikonfigurasi maka squid harus di restart untuk mengecek adanya error pada konfigurasi diatas. Untuk merestart squid melalui :

```
root:~# /etc/init.d/squid restart
```

Pada sisi client, untuk mengakses internet melalui proxy server di atas tanpa harus mengatur atau mengubah arah proxy yang digunakan karena server Proxy terletak pada mesin yang sama dengan router, router perlu ditambahkan perintah untuk mengubah permintaan yang menuju port 80 (HTTP) menjadi menuju port 8080 (squid server Proxy). Dengan demikian seluruh permintaan port 80 yang masuk pada router akan diubah tujuannya menjadi po 8080. Tentu saja tidak semua permintaan pada port 80 yang diubah menjadi port 8080, tetapi hanya permintaan yang menuju ke jaringan local. Karena permintaan ke jaringan local tidak perlu dilewatkan pada server Proxy. Aturan ini diterapkan pada rantai PREROUTING tabel nat denga proses REDIRECT. Aturan yang diterapkan pada file `rc.local` yang ada pada direktori “/etc/” agar perintah ini dibaca setiap kali server Proxy *booting* adalah sebagai berikut :

```
#Redirect ke Transparent Proxy 10.11.12.0/24
iptables -t nat -A PREROUTING -p tcp -m tcp -i eth1 -d !
10.11.12.0/24 --dport 80 -j REDIRECT --to-port 8080

#NAT subnet smkn6 10.11.12.0/24
iptables -t nat -A POSTROUTING -s 10.11.12.0/24 -d 0/0 -j
MASQUERADE
```

Perintah diatas tersebut menyatakan bahwa sebelum mengalami routing, paket TCP yang masuk melalui antarmuka `eth1`, yang punya tujuan port 80 tetapi tidak menuju ke jaringan 10.11.12.0/24, akan diubah tujuan port menjadi 8080 dan arahnya diubah masuk ke dalam server Proxy.

4.2.1.4 Penambahan Program Autentifikasi mysql_auth

Autentifikasi yang di inginkan merupakan autentifikasi yang terintegrasi dengan menggunakan database. Database yang dipilih merupakan mysql, karena mysql merupakan salah satu database yang cukup handal untuk mengelola data untuk autentifikasi.

Langkah pertaman yaitu menyiapkan software mysql_auth dan paket pendukungnya.

```
root@proxy~# wget http://people.arxnet.hu/airween/mysql_auth/mysql_auth0.8.tar.gz
```

```
root@proxy~# wget http://www.zero-sys.net/portal/download/additionalselect.patch
```

Setelah proses download selesai, kemudian melakukan untar sekaligus meng-*uncompress* pada paket mysql_auth0.8.tar.gz, sehingga akan terdapat direktori baru yaitu "mysql_auth0.8".

```
root@proxy~# tar -xzxvf mysql_auth0.8.tar.gz
```

```
root@proxy~# cd mysql_auth0.8
```

Selanjutnya lakukan patching mysql_auth proses ini memakan waktu yang cukup singkat, yaitu sekitar 6 detik.

```
root@proxy:~/mysql_auth0.8# patch -p1 < /root/additionalselect.patch
```

Setelah selesai proses pachting dilanjutkan dengan proses compile dan install mysql_auth

```
root@proxy:~/mysql_auth0.8# make && make install
```

```

root@proxy:~/mysql_auth-0.8# make && make install
gcc -o mysql_auth src/mysql_auth.c src/confparser.c -lmysqlclient -I/usr/include/
/mysql/ -L/usr/lib/
gcc -o mypasswd src/mypasswd.c src/confparser.c -lmysqlclient -I/usr/include/mys
ql/ -L/usr/lib/
/usr/bin/install -o nobody -g nogroup -m 755 mysql_auth /usr/local/squid/libexec
/mysql_auth
/usr/bin/install -o root -g root -m 700 mypasswd /usr/local/bin/mypasswd
/usr/bin/install -o nobody -g nogroup -m 644 src/mysql_auth.conf /usr/local/squi
d/etc/mysql_auth.conf
/usr/bin/install -o nobody -g nogroup -m 600 src/mysql_auth.conf /usr/local/squi
d/etc/mysql_auth.conf.default
root@proxy:~/mysql_auth-0.8# _

```

Gambar 4.10 Pembuatan mysql_auth

Jika proses compile di atas berhasil dilakukan, maka dapat dicoba program mysql_auth secara manual melalui terminal. Output ERR akan muncul ketika autentikasi gagal, sedangkan output OK akan muncul ketika autentikasi sukses dilakukan

```

#mysql_auth
0634015073 yaqi
ERR
0634015073 yaqin
OK

```

Selanjutnya hanya perlu menambahkan beberapa baris perintah konfigurasi. Baris perintah ini yang akan menghubungkan program ncsa_auth yang sudah dibuat dengan squid. Agar web browser dapat menerima permintaan input berupa nama user dan password dari setiap pengguna yang terdaftar di server Proxy, diperlukan perubahan konfigurasi pada */etc/squid/squid.conf*.

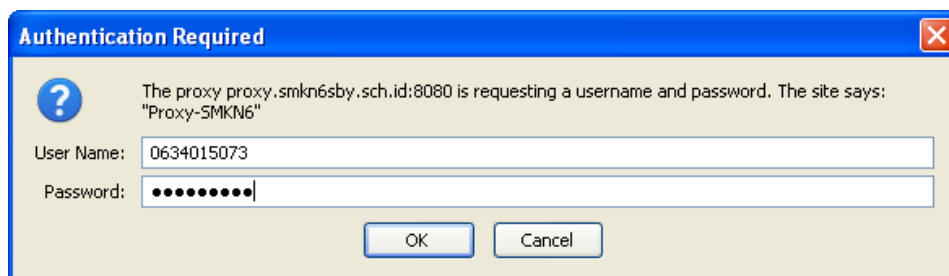
Perubahan tersebut antara lain meliputi beberapa hal pada potongan listing squid.conf berikut ini:

```
...
#Tag : auth_program
auth_param basic realm Proxy-SMKN6
auth_param basic program /usr/local/squid/libexec/mysql_auth
auth_param basic credentialsttl 2 hour
auth_param basic children 5
auth_param basic casesensitive on

...
#ACL list
acl smkn6_password proxy_auth REQUIRED

...
#http_access
http_access allow smkn6_password
http_access deny all
```

Setelah program autentifikasi selesai dibangun dan squid.conf telah dirubah menggunakan mysql_auth. Pada web browser coba kembali mengakses internet maka akan muncul “jendela” baru yang meminta autentifikasi user dan password seperti dibawah ini:



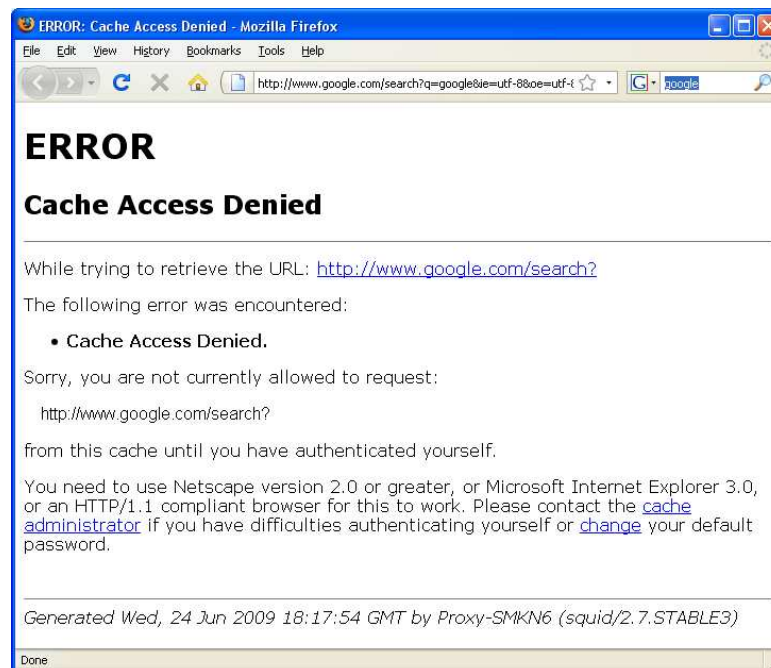
Gambar 4.11 Proses Autentifikasi Proxy

Jika autentifikasi benar, maka hak akses untuk user “siswa” akan diberikan untuk user tersebut. Maka fasilitas internet sudah dapat di akses oleh user tersebut seperti gambar berikut ini:



Gambar 4.12 Web-browser Berhasil Authentifikasi Proxy

Namun jika username dan password tidak cocok dengan data pada mysql atau gagal melakukan autentifikasi akan proxy akan meminta user memasukkan username dan password dengan benar sampai tiga kali. Namun jika masih tetap dapat autentifikasi dengan benar maka fasilitas internet tidak akan dapat diperoleh user tersebut.



Gambar 4.13 Web-browser Gagal Authentifikasi

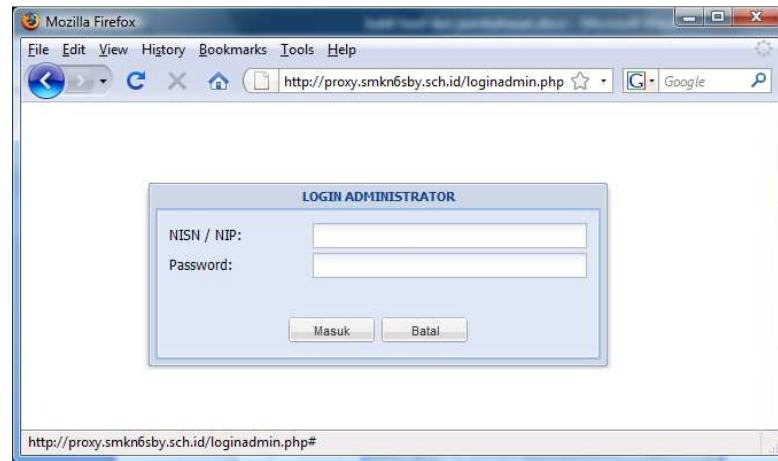
4.2.1.5 Management User Pengguna Internet

Form pendaftaran dapat diakses dengan alamat <http://proxy.smkn6sby.sch.id/proxy> digunakan ketika user membuka web. Form pendaftaran dilakukan ketika user akan daftar untuk melakukan login ke proxy. Pada form pendaftaran terdapat nomor induk, password, password (ulang), nama lengkap, kelas, dan kelamin yang masing-masing textfield harus diisi. Setelah terisi semua textfield, form akan menyimpan data tersebut ke dalam database. Setelah semua data terisi lengkap, tekan tombol simpan yang akan tersimpan dalam database. Klik tombol kosongkan ketika admin ingin menghapus semua data yang akan dimasukkan yang terdapat form pendaftaran user. Sedangkan tombol batal digunakan untuk membatalkan pengisian form daftar. Ketika admin memasukkan data tidak lengkap akan muncul message box “Isian Anda tidak valid!”. Untuk lebih jelasnya lihat Gambar 4.14.

The screenshot shows a web browser window titled "Daftar :: Mozilla Firefox" with the address bar displaying "http://proxy.smkn6sby.sch.id/". The main content area is titled "Form Registrasi Pengguna Internet SMKN 6 Surabaya". On the left, there is a "Silahkan Login Disini!" section with fields for "NISN / NIP" and "Password", and a "Login" button. The main registration form, titled "Daftar", includes the following fields: "NISN / NIP:" with a placeholder "Masukkan Nomor Induk", "Password:", "Password (Ulangi):", "Nama Lengkap:" with a placeholder "Masukkan Nama Lengkap", "Kelas:" with a dropdown menu showing "Kelas", and "Kelamin:" with a dropdown menu showing "Jenis Kelamin". At the bottom of the form are three buttons: "Simpan", "Kosongkan", and "Batal".

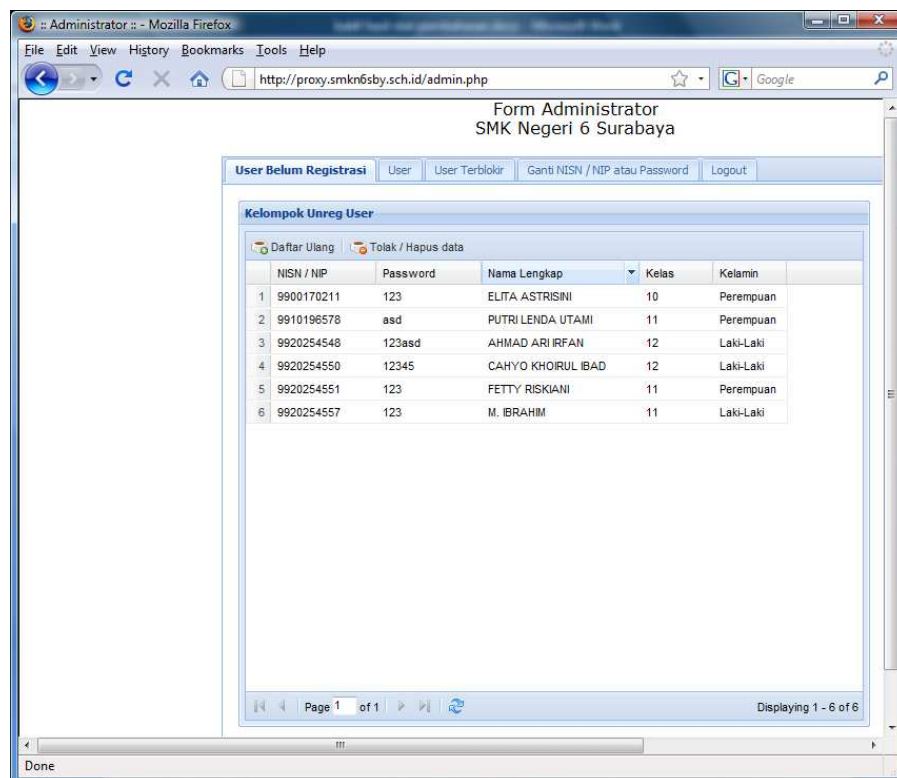
Gambar 4.14 Form Pendaftaran User Proxy

Login administrator digunakan untuk administrator melakukan pendaftaran ulang user yang telah registrasi.



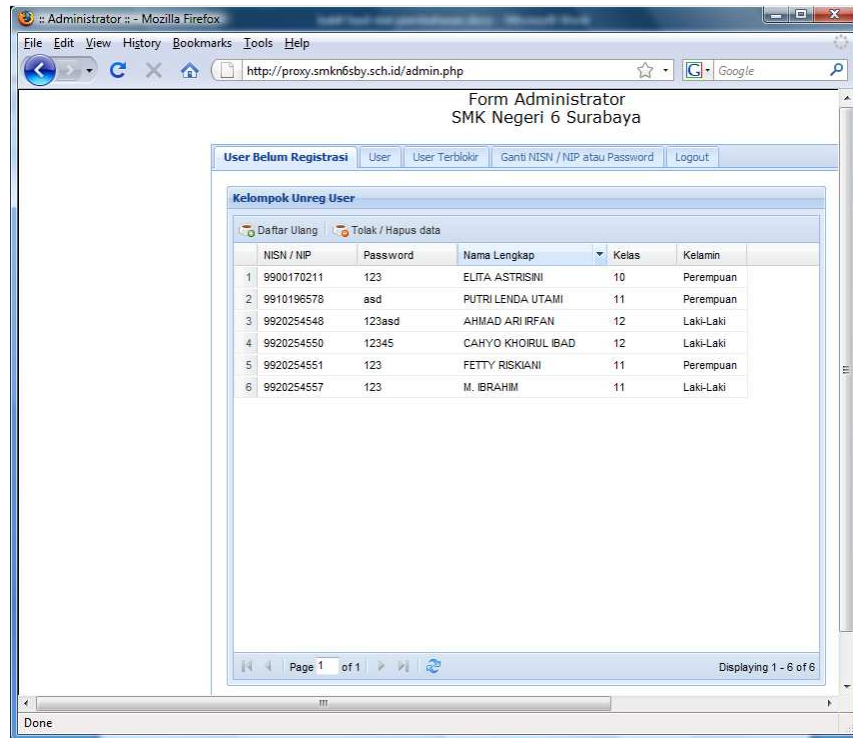
Gambar 4.15 Login Admin

Form ini adalah form yang digunakan oleh admi untuk melakukan pendaftrn ulang user yang telah terdaftar sehingga user dapat melakukan login ke proxy.



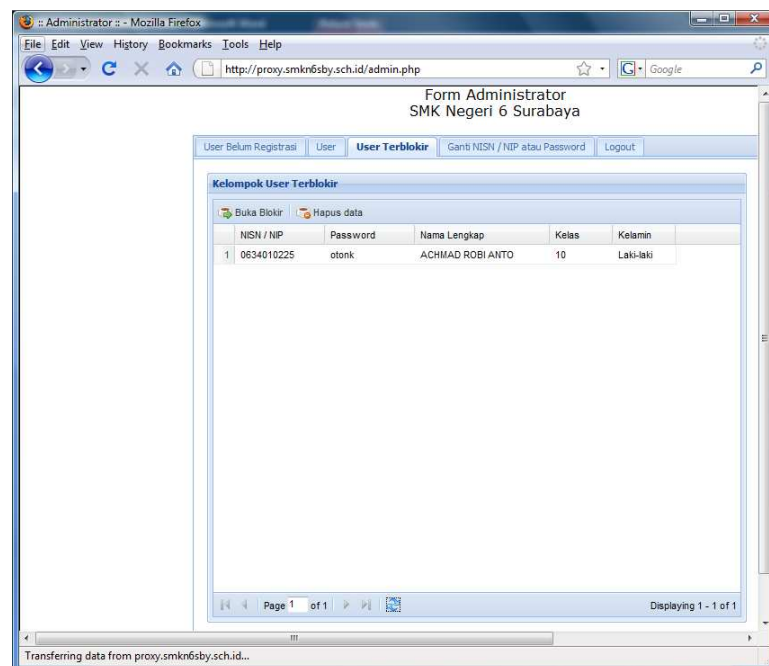
Gambar 4.16 Form User Belum Registrasi

Daftar user yang telah teregistrasi, yang sudah dapat login ke proxy.



Gambar 4.17 Form User Teregistrasi

Daftar user yang terblokir oleh admin karena melakukan beberapa pelanggaran, seperti membuka web site porno.



Gambar 4.18 Form User Terblokir

Form ini digunakan untuk mengupdate khusus user admin.

Gambar 4.19 Form Ganti Username dan Password Admin

Form ini digunakan untuk mengupdate user, yang dapat dilakukan oleh setiap user terdaftar.

Gambar 4.20 Form Ganti Username dan Password User

4.2.2 Konfigurasi Pada Komputer Server

Komputer server disini merupakan komputer yang digunakan sebagai tempat penyimpanan data/ file yang biasanya disebut dengan file server yang menggunakan aplikasi Samba. Samba merupakan kumpulan perangkat lunak yang mengimplementasikan server SMB. Paket samba server menyediakan daemon server `smbd` yang digunakan sebagai penyedia layanan SMB bagi klien. Samba menggunakan protocol NetBIOS over TCP/IP (NetBT).

4.2.2.1 Pemasangan IP Address pada Komputer Server

Seperti halnya pada pemasangan IP Address yang dilakukan di komputer router, yaitu dengan mengedit file *interfaces* pada direktori “*/etc/network/*”. Namun karena penulis menginginkan pemasangan IP Address yang bersifat permanen, maka penulis melakukan cara ketiga yaitu mengedit file *interfaces* pada direktory “*/etc/network/*”. Adapun perintah yang dilakukan adalah:

```
root:~# nano /etc/network/interface
```

Perintah di atas adalah mengedit file *interfaces* dengan menggunakan text editor yang bernama *nano* Pemasangan IP Address bisa dilakukan dibawah ini.

```
auto lo eth0
iface lo inet loopback

iface eth0 inet static
    address 10.11.12.1
    netmask 255.255.255.0
    gateway 10.11.12.13
    dns-nameservers 10.11.12.13
```

Untuk menginstal paket samba server digunakan perintah sebagai berikut :

```
root:~# apt-get install samba
```

```

root@server-smkn6:~# apt-get install samba
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libcups2 libtalloc1 libwbclient0 samba-common update-inetd
Suggested packages:
  cups-common openbsd-inetd inet-superserver smbldap-tools
The following NEW packages will be installed:
  libcups2 libtalloc1 libwbclient0 samba samba-common update-inetd
0 upgraded, 6 newly installed, 0 to remove and 23 not upgraded.
Need to get 8113kB of archives.
After this operation, 22.5MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://10.11.12.1 intrepid-updates/main libcups2 1.3.9-2ubuntu1 [168kB]
Get:2 http://10.11.12.1 intrepid/main libtalloc1 1.2.0~git20080616-1 [14.5kB]
Get:3 http://10.11.12.1 intrepid/main libwbclient0 2:3.2.3-1ubuntu3 [87.2kB]
2% [3 libwbclient0 37926/87.2kB 43%]_

```

Gambar 4.21 Instalasi Samba

4.2.2.2 Konfigurasi Server SMB

Konfigurasi server SMB diatur dalam file `smb.conf` yang terdapat pada direktori `/etc/samba`.

Berikut ini merupakan isi file `smb.conf` yang digunakan untuk menjalankan *server* SMB.

```

[global]
    workgroup = SMKN6
    server string = %h server
    map to guest = Bad User
    obey pam restrictions = Yes
    passdb backend = tdbsam
    pam password change = Yes
    passwd program = /usr/bin/passwd %u
    passwd chat = *Enter\snew\s*\spassword:* %n\n
    *Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
    unix password sync = Yes
    syslog = 0
    log file = /var/log/samba/log.%m
    max log size = 1000
    logon path =
    dns proxy = No
    usershare allow guests = Yes
    panic action = /usr/share/samba/panic-action %d
    invalid users = root

[siswa]
    comment = Tempat File - file siswa
    path = /media/siswa
    valid users = smkn6, siswa, guru
    admin users = smkn6, siswa
    write list = smkn6, siswa
    read only = No
    create mask = 0777
    directory mask = 0777
    guest ok = No

```

```
[guru]
    comment = Tempat File - file guru
    path = /media/guru
    valid users = smkn6, guru
    admin users = smkn6, guru
    write list = smkn6, guru
    read only = No
    create mask = 0777
    directory mask = 0777
    guest ok = No
[sekolah]
    comment = data-data sekolah
    path = /media/sekolah
    valid users = smkn6, guru, siswa
    admin users = smkn6, guru
    write list = smkn6, guru
    read only = No
    create mask = 0777
    directory mask = 0777
    guest ok = No
```

Penjelasan tentang pengaturan di atas dapat diungkapkan pada bagian berikut ini.

1. Bagian [global]

Menjelaskan konfigurasi global pada samba, seperti menentukan workgroup

2. Bagian [siswa]

Menentukan direktori sharing untuk siswa, pada bagian ini user hanya dapat mengakses direktori ini hanya user : smkn6, guru, dan siswa. User siswa dan smkn6 mempunyai akses penuh terhadap direktori ini.

3. Bagian [guru]

Menentukan direktori sharing untuk guru, pada bagian ini user hanya dapat mengakses direktori ini hanya user : smkn6, guru, dan siswa. Hanya user smkn6 dan guru yang mempunyai hak penuh terhadap direktori ini.

4. Bagian [sekolah]

Menentukan direktori sharing untuk siswa, pada bagian ini user hanya dapat mengakses direktori ini hanya user : smkn6, guru, dan siswa. Hanya smkn6 dan guru yang mempunyai hak akses penuh terhadap direktori ini.

Setelah server samba dikonfigurasi ulang, selanjutnya perlu dibuat agar user sistem Linux dapat diaktifkan menjadi user SMB. Untuk mengaktifkan user sistem Linux menjadi user SMB digunakan utilitas *smbpasswd*. Sebelumnya user didaftarkan menjadi user SMB harus dibuat terlebih dahulu dengan perintah *useradd*. Berikut ini akan dibuat tiga user yaitu : smkn6, guru, dan siswa.

```
root@router-smkn6:~# useradd smkn6
root@router-smkn6:~# passwd smkn6
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@router-smkn6:~# useradd guru
root@router-smkn6:~# passwd guru
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@router-smkn6:~# useradd siswa
root@router-smkn6:~# passwd siswa
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@router-smkn6:~# _
```

Gambar 4.22 Membuat User Pada Sistem Linux

Setelah user dibuat pada sistem Linux selanjutnya user tersebut akan diaktifkan menjadi user SMB dengan perintah sebagai berikut :

```

root@router-smkn6:~# smbpasswd -a smkn6
New SMB password:
Retype new SMB password:
root@router-smkn6:~# smbpasswd -a guru
New SMB password:
Retype new SMB password:
Added user guru.
root@router-smkn6:~# smbpasswd -a siswa
New SMB password:
Retype new SMB password:
root@router-smkn6:~# _

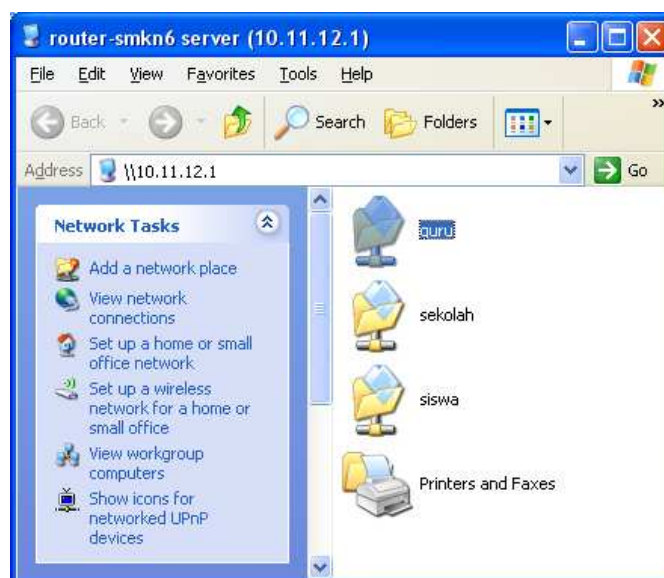
```

Gambar 4.23 Mengaktifkan User Sistem Menjadi User SMB

Sama halnya dengan service-service lainnya harus di restart setelah melakukan konfigurasi ulang :

```
root:~# /etc/init.d/samba restart
```

Setelah server SMB selesai dikonfigurasi dan berhasil dijalankan selanjutnya dapat dicoba untuk diakses dari klien SMB. Pengakses server SMB melalui Windows xp dengan cara mengetikkan IP address server SMB pada Windows explorer \\10.11.12.1.



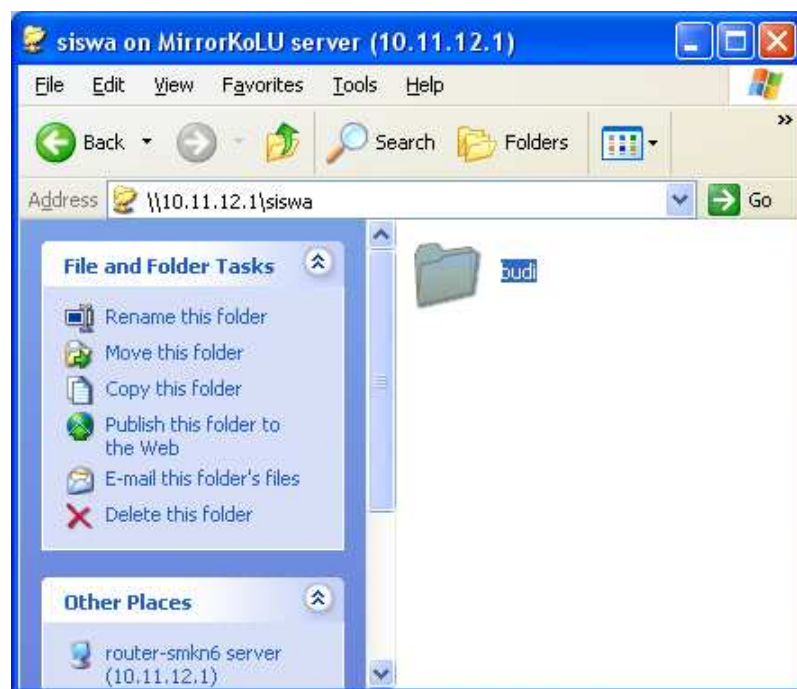
Gambar 4. 24 Direktori Sharing

Setelah direktori sharing muncul maka dapat dipilih salah satu direktori tersebut untuk di akses.



Gambar 4.25 Autentifikasi Samba

Jika autentifikasi benar, maka hak akses untuk user “siswa” akan diberikan untuk user tersebut. Maka layanan Server SMN sudah dapat di akses oleh user tersebut seperti gambar berikut ini.



Gambar 4.26 Akses Direktori Samba

BAB V

PENUTUP

5.1. KESIMPULAN

Terselesaikannya laporan praktek kerja lapangan dengan judul Rancang Bangun Jaringan Komputer Berbasis Ubuntu Server Intrepid Ibex 8.10 di SMK Negeri 6 Surabaya ini diharapkan dapat bermanfaat sehingga apa yang telah dikerjakan tidak menjadi hal yang sia-sia, mengenai materi yang dijelaskan oleh penulis dalam laporan ini, penulis telah memberikan cara yang cukup detail yang diharapkan dapat langsung di mengerti baik oleh para pembaca.

Dengan di lakukannya konfigurasi pada router dimana router menyediakan layanan *dhcp server*, *domain server* dan *proxy server*. Pengguna jaringan di SMK Negeri 6 akan merasa mudah untuk dapat mengakses resource yang ada karena tidak perlu ada konfigurasi tambahan pada komputer pengguna. Hanya saja untuk bisa mendapatkan akses internet pengguna harus mendaftar terlebih dahulu pada *proxy server* dengan alamat <http://proxy.smkn6sby.sch.id> setelah berhasil mendaftar pengguna harus menghubungi admin untuk konfirmasi agar user yang baru saja di daftarkan bisa digunakan untuk akses internet.

Dalam jaringan di SMK Negeri 6 Surabaya tersedia juga sebuah Komputer *file server* yang dapat bertindak sebagai komputer penyedia layanan file sharing yang dapat di akses oleh pengguna. Di mana file server menyediakan tiga direktori sharing yaitu direktori sekolah, direktori guru, dan direktori siswa. Setiap guru dan siswa dapat menyimpan dan mengambil data/file ke dalam *file server* tersebut sesuai kebutuhan dan sesuai dengan direktori masing-masing.

5.2. SARAN

Dari penelitian dan praktek kerja lapangan yang telah dijalankan penulis, maka penulis memiliki beberapa saran ataupun ide yang dirasa dapat dikembangkan diantaranya adalah:

- Pemberian kuota dan batasan pemakaian internet dan file server pada setiap user yang ada.
- Penyamaan autentifikasi yang digunakan pada fasilitas yang ada pada SMK Negeri 6 Surabaya. Seperti penggabungan autentifikasi untuk e-mail, samba, proxy.
- Perlunya perbatasan jam untuk pemakain internet.

DAFTAR PUSTAKA

1. Wagito. ST. MT. *Jaringan Komputer Teori dan Implementasi Berbasis Linux*, Gava Media. Yogyakarta. 2005.
2. Supriyanto, Dodit. *Buku Pintar Pemrograman PHP*. Oase Media, Bandung. 2008.
3. ILC (Indonesia Linux Conference) 2007, source *Linux Webserver Proxy Router*, UNIVERSITAS ATMA JAYA Yogyakarta
4. Ghazali, M. 2008. *Topologi Jaringan*.
5. <http://ghanoz2480.files.wordpress.com/2008/06/ghanoz-2480-topologi-jaringan.pdf> diakses pada 12 April 2009.
6. Komunitas Ubuntu Indonesia. t.th. *Ihwal Ubuntu*. <http://www.ubuntu-id.org> diakses pada 10 April 2009.
7. Ardiyansah, D. *Teknologi Jaringan Komputer*, www.ilmukomputer.com, diakses pada 20 April 2009.
8. Fahrial, J. *Teknik Konfigurasi LAN*, www.ilmukomputer.com , diakses pada 25 April 2009.
9. Yuhefizal. *Tutorial Komputer dan Jaringan*, www.ilmukomputer.com, diakses pada 30 April 2009