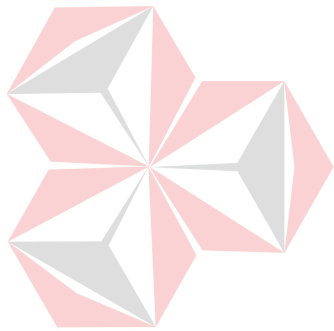


LAPORAN KERJA PRAKTEK
IMPLEMENTASI ADMINISTRASI *SERVER* BERBASIS
LINUX ZORIN



UNIVERSITAS
Dinamika

Oleh :

Nama : AKHMAD RIZKI FAJAR

Nim : 11.41020.0038

Program : S1 (Strata Satu)

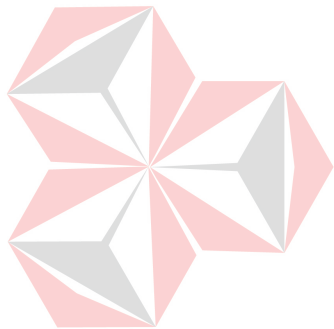
Fakultas : Teknologi dan Informatika

Jurusan : Sistem Komputer

INSTITUT BISNIS DAN INFORMATIKA STIKOM
SURABAYA

2015

LAPORAN KERJA PRAKTEK
IMPLEMENTASI ADMINISTRASI *SERVER* BERBASIS
LINUX ZORIN



UNIVERSITAS
Dinamika

Oleh :

Nama : AKHMAD RIZKI FAJAR

Nim : 11.41020.0038

Program : S1 (Strata Satu)

Fakultas : Teknologi dan Informatika

Jurusan : Sistem Komputer

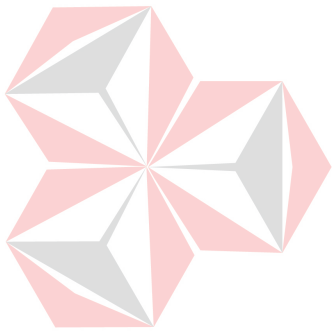
INSTITUT BISNIS DAN INFORMATIKA STIKOM
SURABAYA

2015

LAPORAN KERJA PRAKTEK
IMPLEMENTASI ADMINISTRASI *SERVER* BERBASIS
LINUX ZORIN

Diajukan sebagai salah satu syarat untuk menempuh ujian Tahap Akhir

Program Strata Satu (S1)



Disusun Oleh :

Nama : AKHMAD RIZKI FAJAR

Nim : 11.41020.0038

Program : S1 (Strata Satu)

Fakultas : Teknologi dan Informatika

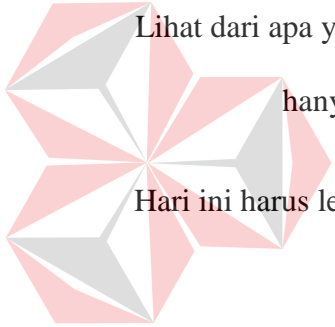
Jurusan : Sistem Komputer

INSTITUT BISNIS DAN INFORMATIKA STIKOM

SURABAYA

2015

Moto



Lihat dari apa yang ia perbuat, jangan hanya lihat dari luarnya saja karena itu hanya tipuan sesaat yang bisa mengalihkan segalanya.

Hari ini harus lebih baik daripada hari kemarin dan hari esok harus lebih baik daripada hari ini.

HALAMAN PERSEMBAHAN

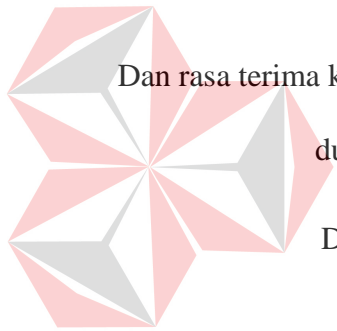
Puji syukur tetap terlimpahkan pada Allah SWT yang telah memberikan rahmat dan hidayahnya. Berhasil terselesaikannya laporan Kerja Praktek ini bukanlah semata-mata karena usaha dan kerja individu penulis sendiri, tetapi mendapatkan banyak bantuan dari berbagai pihak, untuk itu penulis menyampaikan terimakasih kepada :

1. Orang Tua dan Saudari saya yang telah memberikan banyak dukungan, motivasi dan doa.
2. Bapak Anjik Sukmaaji, S.Kom., M.Eng, selaku Kepala Program Studi S1 Sistem Komputer IBIS Surabaya.
3. Bapak Jonathan Suatmojo, S.ST, selaku penyelia dan juga sekaligus sebagai mentor saya selama Kerja Praktek, yang telah berkenan meluangkan waktunya untuk membimbing dan mengarahkan Penulis selama penulisan laporan ini.
4. Bapak Johan Pamungkas ST, selaku Dosen Pembimbing dan sekaligus dosen kami selama Kerja Praktek yang membimbing dan mendukung kami dalam menyelesaikan laporan ini.
5. Teman-teman satu jurusan dan semua pihak yang tidak bisa disebutkan satu per satu. Kalian semua selalu jadi yang istimewa untukku. Terimakasih buat bantuan, dukungan dan do'a kalian selama ini.

Semoga Allah SWT membalas seluruh kebaikan kalian dengan kebaikan yang lebih baik, Amin!.

Saya persembahkan kepada

Aba, Umik dan kakak saya tercinta



Dan rasa terima kasih sebanyak mungkin kepada orang-orang yang memberikan dukungan serta do'a yang dikhususkan bagi saya.

Dan semua orang yang sangat menyayangi saya.

UNIVERSITAS
Dinamika

LEMBAR PERSETUJUAN KERJA PRAKTEK

Laporan Kerja Praktek dengan Judul :

IMPLEMENTASI ADMINISTRASI *SERVER* BERBASIS

LINUX ZORIN


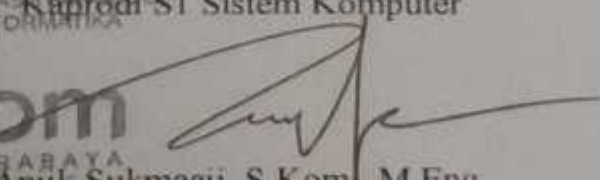
Telah diperiksa, diuji dan disetujui

Surabaya, 4 Februari 2015

Disetujui :

 <p>Pembimbing  Johan Ramungkas, S.T. NIDN 140822</p>	<p>Sekretaris Dinas Komunikasi dan Informatika  Gabyo Utomo, S.T., M.TP Pembina NID 1970 0524 1997 0310 06</p> 
--	--

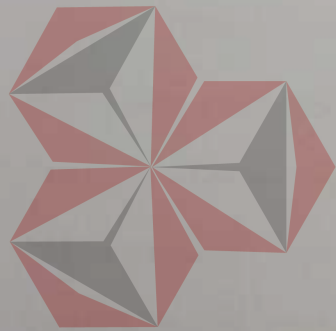
Mengetahui :

 <p>FAKULTAS TEKNOLOGI DAN INFORMATIKA stikom SURABAYA</p>	<p>Kaprodi SI Sistem Komputer  Anjik Sukmaaji, S.Kom, M.Eng. NIDN 0731057301</p>
--	--

PERNYATAAN

Dengan ini menyatakan dengan benar, bahwa laporan Kerja Praktek ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam Kerja Praktek adalah semata hanya rujukan yang dicantumkan dalam daftar pustaka saya.

Apabila kemudian hari ditemukan adanya tindakan plagiat pada karya laporan Kerja Praktek ini maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.



UNIVERSITAS
Dindamika

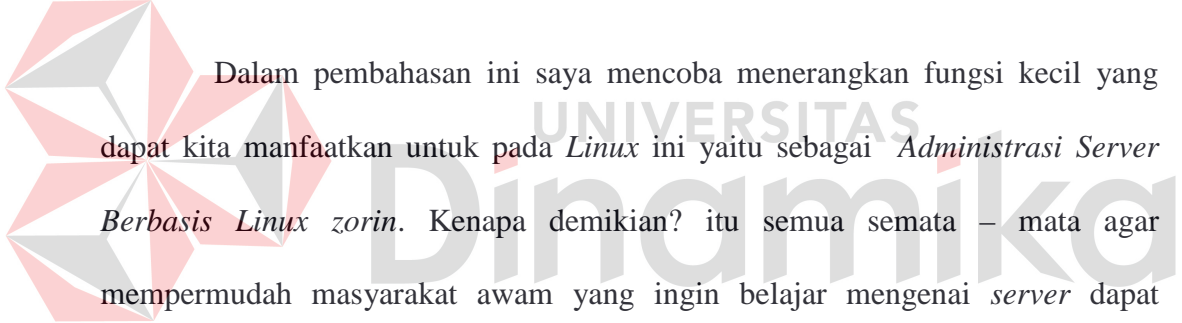
Surabaya, 4 Februari 2015

Akumad Rizki Fajar

NIM 11410200938

ABSTRAKSI

Linux Zorin adalah distribusi *Linux* berbasis *Ubuntu* yang dirancang khusus bagi pendatang baru untuk *Linux*. Memiliki antarmuka pengguna *grafis* seperti *Windows* dan dilengkapi program-program serupa dengan yang ada dalam *Windows*. *Linux Zorin* juga dilengkapi dengan sebuah aplikasi yang memungkinkan pengguna menjalankan banyak program *Windows*. Tujuan utama distribusi ini adalah untuk memberikan alternatif *Linux* ke *Windows* dan membiarkan pengguna *Windows* menikmati semua fitur dari *Linux* tanpa komplikasi.



Dalam pembahasan ini saya mencoba menerangkan fungsi kecil yang dapat kita manfaatkan untuk pada *Linux* ini yaitu sebagai *Administrasi Server Berbasis Linux zorin*. Kenapa demikian? itu semua semata – mata agar mempermudah masyarakat awam yang ingin belajar mengenai *server* dapat dengan mudah menggunakannya dengan cara yang mudah dan biaya yang sangat murah. Dengan demikian akan semakin banyak masyarakat yang semakin tertarik dengan kemajuan teknologi.

Dalam kesempatan ini yang awal kita lakukan ialah menginstall *Linux Zorin* itu sendiri dan di lanjutkan dengan menambahkan beberapa kebutuhan yang di perlukan untuk membangun *Administrasi Server Berbasis Linux zorin* itu sendiri seperti *MySQL*, *PhpMyAdmin* dan lain sebagainya yang saya bahas lebih dalam di halaman selanjutnya.

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan - Nya, sehingga penulis dapat menyelesaikan Laporan Kerja Praktek ini. Penulisan Laporan ini adalah sebagai salah satu syarat Menempuh Tugas Akhir pada Program Studi Sistem Komputer IBIS Kota Surabaya.

Dengan segala kerendahan hati penulis mengaku dan menyadari bahwa penyusunan laporan Kerja Praktek ini masih jauh dari kesempurnaan, walaupun penulis telah berusaha mencurahkan segala kemampuan yang ada. Untuk itu penulis mengharapkan adanya kritik dan saran yang dapat penulis jadikan sebagai bahan acuan untuk penyusunan laporan-laporan yang selanjutnya.

Dalam usaha menyelesaikan penulisan Laporan Kerja Praktek ini penulis banyak mendapat bantuan dari berbagai pihak baik masukan maupun dukungan.

Oleh karena itu penulis mengucapkan terima kasih dan penghargaan setinggi - tingginya kepada :

1. Orang Tua dan Kakak saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Kerja Praktek maupun laporan ini.
2. Bapak Jonathan Suatmojo, S.ST, selaku penyelia. Terima kasih atas bimbingan yang diberikan sehingga penulis dapat melaksanakan Kerja Praktek dengan baik.

3. Bapak Anjik Sukmaaji, S.Kom., M.Eng selaku Kepala Program Studi Sistem Komputer Surabaya atas ijin yang diberikan untuk melaksanakan Kerja Praktek di Dinas Komunikasi dan Informatika Kota Surabaya.
4. Bapak Johan Pamungkas ST, selaku pembimbing saya sehingga dapat menyelesaikan laporan Kerja Praktek.
5. Ibu Maria Novita selaku Koordinator Kerja Praktek di IBIS SURABAYA terima kasih atas bantuan yang telah diberikan.
6. Teman- teman seperjuangan , alumni yang selalu memberikan semangat dan bantuannya.

Semoga bantuan yang diberikan kepada penulis selama penyusunan dan penulisan laporan Kerja Praktek ini mendapatkan kebaikan yang berlipat ganda dari Allah SWT.



UNIVERSITAS
Dinamika
Surabaya, 25 Desember 2014

Penulis

DAFTAR ISI

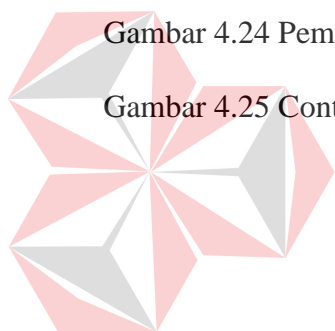
	Halaman
HALAMAN SAMPUL	i
HALAMAN SYARAT	ii
HALAMAN MOTO	iii
HALAMAN PERSEMBAHAN	iv
HALAMAN PENGESAHAN.....	vi
HALAMAN PERNYATAAN	vii
ABSTRAKSI	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Kerja Praktek.....	2
1.5 Waktu dan Lama Kerja Praktek	3
1.6 Ruang Lingkup Kerja Praktek.....	3
1.7 Sistematika Laporan	4

BAB II PROFIL PERUSAHAAN	5
2.1 Dinas Komunikasi dan Informatika Surabaya.....	5
2.2 Visi dan Misi Perusahaan	6
2.3 Struktur Organisasi.....	6
2.4 Bidang Kerja Perusahaan	7
BAB III TEORI PENUNJANG	9
3.1 MySQL.....	9
3.2 Apache.....	14
3.3 PHP	16
3.4 Web Server	18
3.5 PHPMysqladmin	22
3.6 DNS (<i>Domain Name System</i>).....	23
3.7 BIND (<i>Berkeley Internet Name Domain</i>).....	27
3.8 Mail Server	30
3.9 Squirrelmail	31
BAB IV Pembahasan	32
4.1 Database dan Webserver Pada <i>Linux Zorin</i>	32
4.2 DNS Server.....	36
4.3 Mail Server	44
BAB V Penutup	51
5.1 Kesimpulan.....	51
5.2 Saran.....	51
DAFTAR PUSTAKA	52
LAMPIRAN	53

DAFTAR GAMBAR

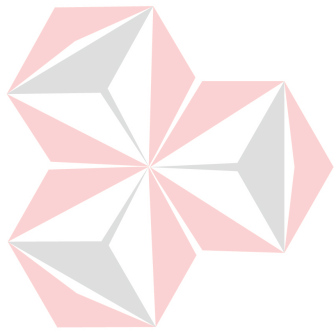
	Halaman
Gambar 3.1 Logo MySQL	9
Gambar 3.2 Logo Apache	14
Gambar 3.3 Logo PHP	16
Gambar 3.4 Logo PHPMyAdmin	22
Gambar 3.5 Struktur DNS	24
Gambar 3.6 Pencarian <i>host pikachu.indolinux.com</i>	28
Gambar 4.1 Hasil Tampilan Perintah Di Atas	34
Gambar 4.2 Hasil Tampilan Perintah Di Atas	34
Gambar 4.3 Hasil Ketika Instalasi Berhasil	35
Gambar 4.4 Hasil Ketika Instalasi Berhasil	36
Gambar 4.5 Blok Diagram <i>Hirarki Domain</i>	37
Gambar 4.6 Hasil Tampilan Perintah Di Atas	39
Gambar 4.7 Contoh <i>IP Address, Netmask, Gateway dan dns-nameserver</i>	39
Gambar 4.8 Hasil Tampilan Perintah Di Atas	40
Gambar 4.9 Hasil Tampilan Perintah Di Atas dan Hal Yang Di Rubah	40
Gambar 4.10 Hasil Tampilan Perintah Untuk Memasukkan <i>zone domain</i> dan <i>zone IP Address dns-nameserver</i>	41
Gambar 4.11 Hasil Tampilan Perintah Di Atas	41
Gambar 4.12 Hasil Konfigurasi file <i>nama_domain.zone</i>	42
Gambar 4.13 Hasil Konfigurasi File <i>nama_domain,rev</i>	42
Gambar 4.14 Contoh Tampilan Saat Melakukan Restart	43

Gambar 4.15 Contoh Pengujian Konfigurasi DNS	43
Gambar 4.16 Hasil Pengujian Konfigurasi DNS Yang Telah Berhasil Di Lakukan	44
Gambar 4.17 Contoh Perintah Yang Harus Di Kerjakan.....	45
Gambar 4.18 Contoh Perintah Yang Harus Di Kerjakan.....	46
Gambar 4.19 Langkah – Langkah Konfigurasi.....	47
Gambar 4.20 Pemilihan Step Selanjutnya.....	47
Gambar 4.21 Pengisian Nama <i>Domain Mail Server</i>	48
Gambar 4.22 Pengisian pada Pop up.....	48
Gambar 4.23 Pengisian Alamat Network Yang Kita Miliki.....	49
Gambar 4.24 Pemilihan Step Selanjutnya.....	49
Gambar 4.25 Contoh Tampilan Saat Melakukan Restart.....	50



DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Surat Balasan dari Instansi / Perusahaan	53
Lampiran 2. Form KP 5 (Acuan Kerja Halaman 1 dan 2)	54
Lampiran 3. Form KP 6 (Log Harian Halaman 1 dan 2)	56
Lampiran 4. Form KP 7	58
Lampiran 5. Kartu Bimbingan Kerja Praktek	59



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Linux Zorin adalah distribusi *Linux* berbasis *Ubuntu* yang dirancang khusus bagi pendatang baru untuk *Linux*. Memiliki antarmuka pengguna grafis Seperti *Windows* dan dilengkapi program - program serupa dengan yang ada dalam *Windows*. *Linux Zorin* juga dilengkapi dengan sebuah aplikasi yang memungkinkan pengguna menjalankan banyak program *Windows*. Tujuan utama distribusi ini adalah untuk memberikan alternatif *Linux* ke *Windows* dan membiarkan pengguna *Windows* menikmati semua fitur dari *Linux* tanpa kesulitan.

Linux Zorin juga merupakan suatu program yang diciptakan unik, hal ini terlihat dari mudahnya mengubah antar muka pengguna dengan satu sentuhan tombol.

Linux Zorin juga memberikan fitur yang lebih mudah bagi penggunanya, antara lain dapat menggunakan *Linux Zorin* bersama sistem operasi yang anda miliki saat ini dan menjalankan program - program *Microsoft Windows* dengan bantuan *Wine* dan *PlayOnLinux*.

Beberapa kelebihan *Linux Zorin* antara lain ialah sistem operasi yang gratis, bebas diperbanyak, sangat cepat bahkan meskipun dijalankan pada *PC* atau laptop tua dengan spesifikasi rendah yang anda sendiri sudah putus asa karena tidak lagi bisa digunakan dengan *Windows*. Sistem operasi yang aman dan cepat, namun dengan tampilan yang familiar seperti *Windows*.

Dengan demikian dapat disimpulkan bahwa *Linux Zorin* ialah suatu sistem operasi yang bisa menjadi ajang belajar dan bereksperimen tentang *Linux* bagi masyarakat awam.

Yang akan dibahas pada kerja praktek dalam laporan ini yaitu mengenai *Pemanfaatan Aplikasi Yang Ada Di Dalam Linux Zorin* untuk pembuatan buku tentang *administrasi server berbasis Linux Zorin*.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas perlu dirumuskan suatu masalah.

Perumusan masalah tersebut dijabarkan di bawah ini :

1. Bagaimana cara menggunakan *Linux Zorin* ?
2. Bagaimana cara instalisasi administrasi *server* pada *Linux Zorin* ?

1.3 Batasan Masalah

Penulis membatasi masalah pada :

1. Penjelasan *Linux Zorin* secara singkat
2. Instalisasi administrasi *server* pada *Linux Zorin*

1.4 Tujuan Kerja Praktek

Memiliki tujuan sebagai berikut :

1. Tujuan Umum

- a. Untuk menambah wawasan serta mempraktekkan ilmu-ilmu yang diperoleh di bangku kuliah
- b. Belajar tepat waktu dan disiplin serta belajar untuk bisa merasakan bagaimana lingkungan kerja yang sesungguhnya
- c. Belajar berinteraksi dengan orang luar selain orang yang biasa kita temui di wilayah kampus atau rumah

2. Tujuan Khusus

- a. Mengetahui dan memahami tentang *Linux Zorin*
- b. Memahami pemanfaatan lebih dalam mengenai *Linux zorin*

1.5 Waktu dan Lama Kerja Praktek

Adapun waktu dan lama Kerja Praktek di Dinas Komunikasi dan Informatika Kota Surabaya dilaksanakan selama 4 minggu yang dimulai pada tanggal 1 Juli 2014

– 31 Juli 2014.

1.6 Ruang Lingkup Kerja Praktek

Sasaran kerja praktek adalah agar mahasiswa mendapatkan pengalaman belajar melalui pembelajaran *Linux Zorin* secara individu yang dapat dimanfaatkan untuk masyarakat umum.

1.7 Sistematika Penulisan

Berikut ini adalah sistematika penulisan laporan hasil Kerja Praktek di Dinas Komunikasi dan Informatika Surabaya

1. BAB I PENDAHULUAN

Berisi Latar Belakang Masalah, Perumusan Masalah, Batasan masalah, Tujuan, Kontribusi serta Sistematika Penulisan di Dinas Komunikasi dan Informatika Kota Surabaya

2. BAB II GAMBARAN UMUM PERUSAHAAN

Pada bab ini membahas tentang gambaran umum Dinas Komunikasi dan Informatika kota Surabaya, sejarah, struktur organisasi, visi, dan misi.

3. BAB III TEORI PENUNJANG

Teori penunjang ini berisi tentang penjabaran yang akan dijadikan sebagai acuan analisa dan pemecahan permasalahan yang dibahas, sehingga memudahkan penulis dalam menyelesaikan masalah.

4. BAB IV PEMBAHASAN

Bagian ini memuat uraian tentang pembahasan laporan kerja praktek mengenai *Implementasi Administrasi Server Berbasis Linux Zorin*.

5. BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan serta saran sehubungan dengan adanya kemungkinan pengembangan sistem pada masa yang akan datang.

BAB II

PROFIL PERUSAHAAN

2.1 Dinas Komunikasi Dan Informatika Surabaya

Dinas Komunikasi dan Informatika adalah Dinas yang mempunyai tugas melaksanakan kewenangan daerah di bidang pengelolaan Teknologi Informasi dan Komunikasi serta melaksanakan tugas pembantuan yang diberikan oleh Pemerintah Pusat dan Pemerintah Provinsi dimana dalam setiap kegiatannya selalu berhubungan dengan Pembangunan dan Pengembangan Sistem Informasi, Pengembangan dan Pemeliharaan Jaringan Komputer Antar Bidang, Pengelolaan Produksi Informasi dan Publikasi, Pengelolaan dan Pengembangan Komunikasi Publik, yang mana pada setiap kegiatan-kegiatan tersebut terbagi menjadi 3 bidang serta 1 Sekretariat, Dimana disetiap bidangnya dibawah oleh kepala bidang.

Sebagai Lembaga pemerintahan yang mempunyai tanggung jawab besar dan bergerak di dalam lingkungan Pemerintah Kota Surabaya, maka DINKOMINFO mempunyai tugas pokok dan fungsi yang besar dalam membangun Teknologi Informasi dan Komunikasi (TIK) di Kota Surabaya.

2.2 Visi dan Misi Perusahaan

Visi :

Terciptanya sistem informasi pemerintah yang terpadu melalui teknologi informasi dan komunikasi.

Misi :

1. Meningkatkan kapasitas pelayanan informasi dan pemberdayaan potensi masyarakat dalam rangka mewujudkan masyarakat berbudaya informasi.
2. Meningkatkan kerjasama kemitraan dan pemberdayaan lembaga komunikasi dan informatika pemerintah dan masyarakat.
3. Meningkatkan daya jangkau infrastruktur komunikasi dan informatika untuk memperluas aksesibilitas masyarakat terhadap informasi dalam rangka mengurangi kesenjangan informasi.
4. Meningkatkan sumber daya manusia di bidang komunikasi dan informatika menuju profesionalisme.

2.3 Struktur Organisasi

Dinas Komunikasi dan Informatika adalah Dinas yang mempunyai tugas melaksanakan kewenangan daerah di bidang pengelolaan Teknologi Informasi dan Komunikasi serta melaksanakan tugas pembantuan yang diberikan oleh Pemerintah Pusat dan Pemerintah Provinsi dimana dalam setiap kegiatannya selalu berhubungan dengan Pembangunan dan Pengembangan Sistem Informasi, Pengembangan dan

Pemeliharaan Jaringan Komputer Antar Bidang, Pengelolaan Produksi Informasi dan Publikasi, Pengelolaan dan Pengembangan Komunikasi Publik, yang mana pada setiap kegiatan-kegiatan tersebut terbagi menjadi 3 bidang yang dibawah oleh kepala bidang dari setiap bidangnya.

Sebagai Lembaga pemerintahan yang mempunyai tanggung jawab besar dan bergerak di dalam lingkungan Pemerintah Kota Surabaya maka tidak menutup kemungkinan DINKOMINFO mempunyai tugas pokok dan fungsi yang besar dalam membangun Teknologi Informasi dan Komunikasi (TIK) di Kota Surabaya.

Secara umum DINKOMINFO membawahi 50 personil yang terbagi dalam 4 (empat) bidang yaitu :

1. Sekretariat
2. Bidang Sarana Komunikasi dan Diseminasi Informasi (SKDI)
3. Bidang Aplikasi dan Telematika (APTEL)
4. Bidang Pos dan Telekomunikasi (POSTEL)

2.4 Bidang Kerja Perusahaan

Dinas Komunikasi dan Informatika mempunyai tugas melaksanakan urusan pemerintahan daerah berdasarkan azas otonomi dan tugas pembantuan di bidang komunikasi dan informatika.

Selain itu Dinas Komunikasi dan Informatika kota Surabaya juga mempunyai fungsi sebagai berikut:

- a. Perumusan kebijakan teknis di bidang Komunikasi dan Informatika;
- b. Penyelenggaraan urusan pemerintahan dan pelayanan umum;
- c. Pembinaan dan pelaksanaan tugas
- d. Pengelolaan ketatausahaan Dinas
- e. Melaksanakan tugas lain yang diberikan oleh Kepala Daerah sesuai dengan tugas dan fungsinya.



UNIVERSITAS
Dinamika

BAB III

TEORI PENUNJANG

3.1 MySQL



Gambar 3.1 Logo MySQL

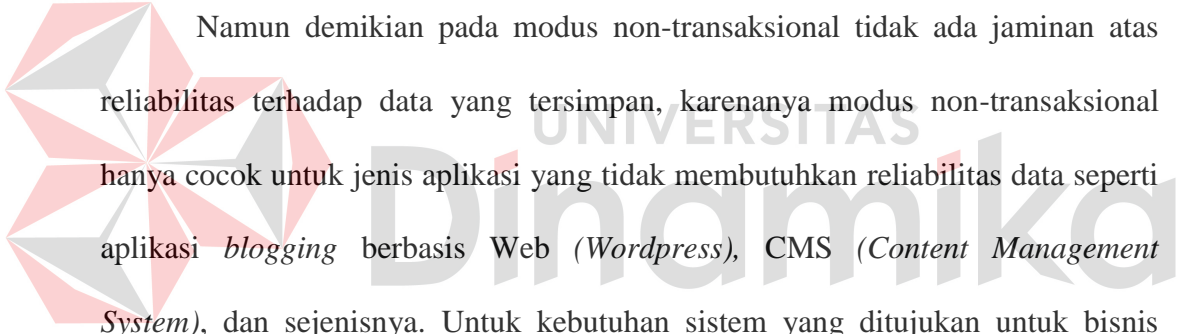
MySQL adalah sebuah *server database open source* yang terkenal yang digunakan berbagai aplikasi terutama untuk *server* atau membuat *WEB*. MySQL berfungsi sebagai SQL (*Structured Query Language*) yang dimiliki sendiri dan sudah diperluas oleh MySQL dan umumnya digunakan bersamaan dengan PHP untuk membuat aplikasi *server*.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana perusahaan tersebut memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL adalah: David Axmark, Allan Larsson, dan Michael “Monty” Widenius.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya, SQL (*Structured Query Language*). SQL

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

Kehandalan suatu sistem basis data (*Database Management System*) dapat diketahui dari cara kerja pengoptimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program - program aplikasi yang memanfaatkannya. Sebagai penyedia basisdata, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan kompetitor lainnya.



Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis Web (*Wordpress*), CMS (*Content Management System*), dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

Dan oleh karena itu MySQL mempunyai kelebihan sebagai berikut :

1. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga*, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL (*General Public License*) sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah.
4. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask, hostname*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket (UNIX)*, atau *Named Pipes (NT)*.

10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Selain itu MySQL juga mempunyai kekurangan sebagai berikut :

1. Untuk koneksi ke bahasa pemrograman visual seperti VB (*Visual Basic*), *delphi*, dan *foxpro*, MySQL kurang mendukung, karena koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program visual tersebut, dan ini yang menyebabkan MySQL jarang dipakai dalam program visual.
2. Data yang ditangani belum begitu besar. Tidak cocok baik menyimpan data maupun untuk memproses data.
3. Program hanya dapat running di *Windows*. Itupun untuk *Windows* lama, untuk *Windows* terbaru seperti Vista, Visual Basic sangat tidak stabil

berjalan di platform tersebut, banyak kantor yang dapat menghubungkan laptop pegawai mereka untuk bergabung dalam sistem jaringan kantor.

4. Tidak Praktis apabila diterapkan pada jaringan komputer. Itu dikarenakan VB program yang berdiri sendiri yang berarti harus diinstalasikan pada tiap komputer pada jaringan tersebut.
5. Aplikasi yang rentan terkena virus, program yang dibuat menggunakan *Visual Basic* adalah berupa execution program (*.exe) yang dapat terinfeksi virus seperti virus *Sality*, program menjadi rusak dan tidak bisa dijalankan. Apabila program terinfeksi tersebut di scan oleh antivirus, program akan dihapus dari sistem, sangat tidak aman untuk sebuah program yang menjadi tulang punggung perusahaan.
6. Mudah *error* dan *crash*, program dari VB seringkali menjadi *not responding* lalu mati tiba-tiba.
7. Memiliki keterbatasan kemampuan kinerja dalam server ketika data yang disimpan telah melebihi batas maksimal kemampuan daya tampung *server* karena tidak menerapkan konsep *Technology Cluster Server*.

3.2 Apache



Gambar 3.2 Logo Apache

Server HTTP Apache atau *Server Web/WWW Apache* adalah *Server Web* yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows* dan *Novell Netware* serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *Web*. Protokol yang digunakan untuk melayani fasilitas *Web/www* ini menggunakan *HTTP*.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, *otentikasi* berbasis *basisdata* dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*Graphical User Interface*) yang memungkinkan penanganan *server* menjadi mudah.

Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation*.

Dan oleh karena itu Apache mempunyai kelebihan sebagai berikut :

1. Apache termasuk dalam kategori *freeware*.
2. Apache mudah sekali proses instalasinya jika dibanding *Web Server* lainnya seperti *NCSA, IIS*, dan lain-lain.
3. Mampu beroperasi pada berbagai *platform* sistem operasi.

4. Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan *peripheral* lainnya ke dalam *platform Web Servernya*.

Selain itu Apache juga mempunyai kekurangan sebagai berikut :

1. *Web Server* Apache tidak memiliki kemampuan mengatur *load* seperti *IIS*, sehingga akan terus membagi proses baru hingga nilai *MaxClients* tercapai atau hingga batas yang diizinkan oleh *OS*.
2. Apache tidak memproses karakter kutip dalam *String Referrer* dan *User Agent* yang dikirimkan oleh *client*. Ini berarti *client* dapat memformulasi inputnya secara hati-hati untuk merusak format baris *log akses*.
3. Terganggunya proses upload data, yang bisa menyebabkan *software* salah dalam menerjemahkan ukuran data yang masuk. Dengan celah tersebut, *hacker* dikabarkan dapat mengeksploitasi kerentanan dengan cara mengirimkan *request* pada *server* Apache bersangkutan.

3.3 PHP



Gambar 3.3 Logo PHP

PHP merupakan singkatan dari *Hypertext Preprocessor*. PHP adalah sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaksnya mirip dengan bahasa pemrograman *C, Java, asp dan Perl*, ditambah beberapa fungsi PHP yang spesifik dan mudah dimengerti.

Sejarah PHP, awalnya PHP merupakan kependekan dari *Personal Home Page* dan PHP itu sendiri pertama kali di buat oleh Rasmus Lerdorf pada tahun 1995, dan pada saat PHP masih bernama FI (*Form Interpreter*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data *form* dari *Web* dan selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP.

Banyak sekali kelebihan yang dimiliki PHP dibandingkan dengan bahasa pemrograman yang lain, Diantaranya :

1. Bisa membuat *Web* menjadi *Dinamis*.
2. PHP bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.

3. Program yang dibuat dengan PHP bisa dijalankan oleh semua sistem operasi karena PHP berjalan secara *Web Base* yang artinya semua sistem operasi bahkan HP yang mempunyai *Web Browser* dapat menggunakan program PHP.
4. Aplikasi PHP lebih cepat dibandingkan dengan *ASP* maupun *Java*.
5. Mendukung banyak paket *Database* seperti MySQL, Oracle, PostgreSQL, dan lain-lain.
6. Bahasa pemrograman PHP tidak memerlukan Kompilasi / *Compile* dalam penggunaannya.
7. Banyak *Web Server* yang mendukung PHP seperti Apache, Lighttpd, IIS dan lain-lain.
8. Pengembangan Aplikasi PHP mudah karena banyak *Dokumentasi, Refrensi & Developer* yang membantu dalam pengembangannya.
9. Banyak bertebaran *Aplikasi & Program* PHP yang gratis dan siap pakai seperti WordPress, PrestaShop, dan lain-lain.

Selain mempunyai kelebihan, PHP juga mempunyai kekurangan. Namun masalah kekurangannya sangat sedikit, diantaranya :

1. PHP tidak mengenal *Package*.
2. Jika tidak di *encoding*, maka kode PHP dapat dibaca semua orang dan untuk meng-*encodingnya* dibutuhkan *tool* dari *Zend* yang mahal sekali biayanya.
3. PHP memiliki kelemahan keamanan. Jadi Programmer harus jeli dan berhati-hati dalam melakukan pemrograman dan konfigurasi PHP.

3.4 *Web Server*

Web Server adalah suatu perangkat lunak yang dapat menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *Web Browser* seperti *mozilla firefox* atau *google chrome* serta menjawab permintaan *client* dengan mengirimkan halaman-halaman *website* yang diminta oleh *client* yang pada umumnya *website* tersebut berbentuk dokumen HTML.

Pengertian dari *website* atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Istilah *Web Page* sering digunakan untuk menunjukan sebuah halaman *website*. *Home page* juga merupakan sebuah halaman *website* (letaknya di bagian depan), maka *home page* juga disebut *Web Page*. Tetapi, tidak semua *Web Page* dapat di sebut *home page*.

Ada beberapa *aplikasi* yang dapat digunakan sebagai *Web Server* :

1. Apache
2. *Microsoft Windows Server 2003 Internet Information Services (IIS)*
3. Appserv
4. Lighthtpd
5. *Sun Java System Web Server*

6. *Xitai Web Server*

7. *Zeus Web Server*

Dari berbagai aplikasi web di atas yang sering digunakan adalah Apache dan *Microsoft Internet Information Services (IIS)*.

Internet Information Service (IIS) adalah komponen yang dapat digunakan untuk mengelola *Web*, *File Transfer Protocol (FTP)*, *Gopher*, dan *NNTP*. Komponen IIS terdapat pada sistem operasi *Windows NT*, *Windows 2000*, *Windows XP*, 7 dan *Windows Server 2003*.

Pada *Windows 98*, supaya dapat mengelola *Web* diperlukan komponen *Personal Web Server (PWS)* walaupun tidak terdapat fasilitas untuk *FTP*. *PWS* juga merupakan bagian dari *IIS*. Komponen *PWS* terdapat pada CD master *Windows 98* dan terletak pada direktori *add-onspws*. Instalasi *PWS* dapat dilakukan dari : **ADD-ONS|PWS|SETUP**.

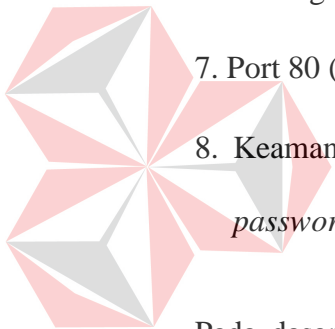
Meskipun sering digunakan, *IIS* tetap mempunyai beberapa komponen pendukung antara lain :

1. Protocol jaringan *TCP/IP*.
2. *Domain Name System (DNS)*.
3. Direkomendasikan untuk menggunakan format *NTFS* demi keamanan.
4. *Software* untuk membuat situs *Web*, salah satunya *Microsoft FrontPage*.

Pada pembahasan *Internet Information Service* ini, yang banyak dibahas adalah *Internet Information Service* yang terdapat pada *Windows 2000 Server* dan hanya terbatas pada *Files Transfer Protocol (FTP)* dan *Web* saja.

Internet Information Service (IIS) sendiri mempunyai beberapa kelebihan dan kekurangan sebagai berikut :

1. Lebih *kompatibel* dengan *Windows* karena memang IIS adalah keluaran *Microsoft*.
2. Unjuk kerja untuk PHP lebih stabil, handal, dan cepat.
3. Adanya kemampuan diagnostik untuk pelacakan kesalahan (bisa dikustomisasi).
4. Untuk *platform .NET*, user hanya bisa menggunakan IIS.
5. Memiliki nilai tertinggi dari *Web Server* lainnya untuk masalah *bug*.
6. Tidak gratis (*Web Server* berbayar).
7. Port 80 (Port untuk layanan *web*) sangat mudah diserang oleh *cracker*.
8. Keamanan file *log* juga sangat mudah ditembus sehingga system *password* pun akan mudah didapatkan.



UNIVERSITAS
Dinamika

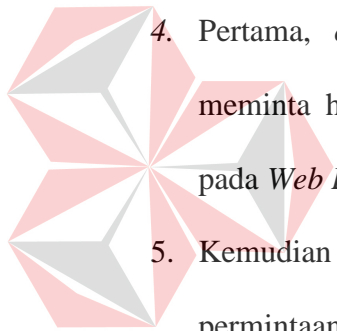
Pada dasarnya tugas *Web Server* hanya ada 2 (dua) yang secara umum digunakan, yaitu:

1. Menerima permintaan (*request*) dari *client*, dan
2. Mengirimkan apa yang diminta oleh *client* (*response*).

Adapun cara kerja atau urutan kerja *Web Server* itu sendiri yang secara umum digunakan sebagai berikut :

1. Client terhubung dengan *Web Browser* dan terhubung dengan *Web Server*.
Client dapat berupa desktop yang memiliki media *browser* yang terhubung dengan internet atau *Web Browser*.

2. *Client* terhubung dengan *Web Browser*. *Web Browser* ini yang meminta data permintaan *client* ke *Web Server*. *Web browser* dan *Web Server* merupakan gabungan atau jaringan dari komputer yang ada di seluruh dunia. *Web browser* dan *Web Server* terhubung dengan *protocol* TCP/IP (*networking protocol*) yang memungkinkan semua komputer dapat berkomunikasi satu sama lain.
3. Pada sisi *server* merupakan sebuah komputer dengan spesifikasi yang memadai sebagai *Web Server* yang di dalamnya terdapat perangkat lunak *Web Server* menggunakan salah satu aplikasi *Web Server*. Contoh dari *Web Server* ini adalah www.google.com, www.polines.ac.id
4. Pertama, *client* akan meminta suatu halaman *web*, misalnya *client* meminta halaman www.google.com dengan mengetikkan URL tersebut pada *Web Browser*.
5. Kemudian *Web Browser* akan meminta data ke *Web Server* dimana permintaan tersebut disampaikan ke *Web Server* dengan menggunakan HTTP (*HyperText Transfer Protocol*). HTTP ini merupakan *protocol* yang digunakan dalam *World Wide Web* (WWW) antar komputer yang terhubung dalam jaringan di dunia ini dalam bahasa sehari-hari. HTTP sama halnya dengan seorang *intrepetter* (penerjemah bahasa) antara dua orang yang sedang berkomunikasi. Permintaan dari *client* melalui *Web Browser* ini diterima oleh *Web Server*. Permintaan itu sendiri disebut dengan HTTP *request*.
6. *Web Server* akan menerima permintaan tersebut, kemudian *Web Server* akan mencari data HTML yang diminta yaitu www.google.com



UNIVERSITAS
Dinamika

7. Jika *Web Page* yang dimaksud ditemukan, maka *Web Server* akan mengirimkan data *Web Page* tersebut ke *Web Browser* untuk disampaikan ke *client*. Proses ini dinamakan dengan *HTTP response*. Tetapi jika data yang dimaksud tidak ditemukan, maka akan menimbulkan *error* dan yang diterima oleh *client* adalah *Web Page* yaitu *Error : 404 Page Not Found*.

3.5 PhpMyadmin



Gambar 3.4 logo PhpMyAdmin

PhpMyAdmin merupakan *software* yang ditulis menggunakan bahasa PHP.

Software ini bisa didapatkan secara gratis karena merupakan program *open source*. Dengan PhpMyAdmin, pengelolaan *database* MySQL menjadi lebih mudah. PhpMyAdmin memiliki banyak fitur, beberapa yang paling penting antara lain mengelola *database*, tabel, relasi, *indeks*, hak akses, dll. PhpMyAdmin hadir dengan dokumentasi yang luas dan memiliki pengguna yang mau berpartisipasi aktif di dalam pengembangan *software* ini.

Jika masih ragu apakah sebuah *provider hosting* menyediakan fitur ini atau tidak, bisa dengan cara melihatnya pada halaman fitur atau menanyakannya

langsung pada *customer support hosting*. Jika ingin mendownloadnya secara langsung, silakan kunjungi websitenya *phpmyadmin.net*.

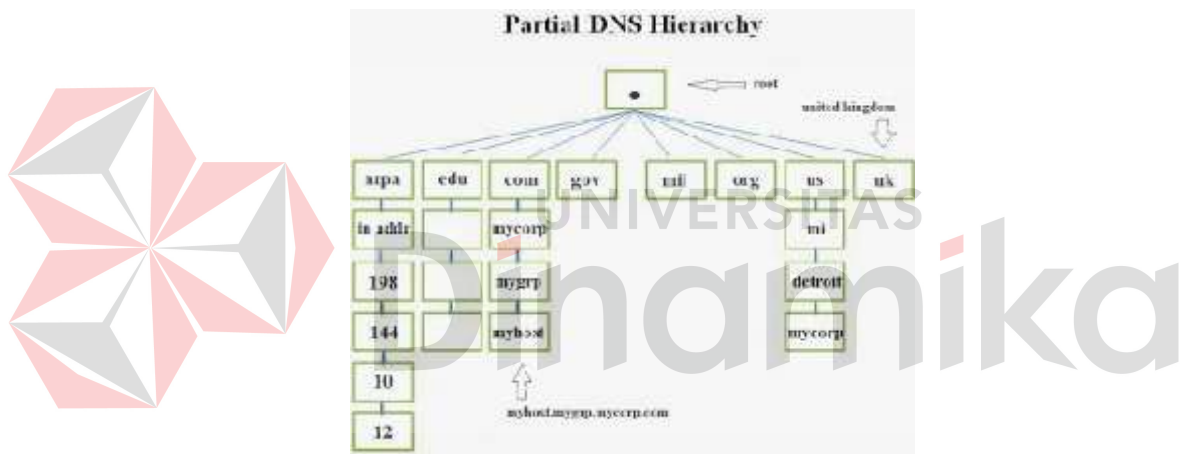
PhpMyAdmin memiliki beberapa fitur yang mendukung kita untuk menggunakannya, diantaranya :

1. Mendukung hampir semua fitur MySQL
2. Mengimpor data dari CSV dan SQL.
3. Mengekspor data dalam berbagai format, seperti : CSV, SQL, XML, PDF, ISO/IEC 26300 – *OpenDocument Text and Spreadsheet*, Word, $L^A T_E X$.
4. Mengatur banyak *server* sekaligus.
5. Membuat grafik PDF dari susunan *database*.
6. Fitur pencarian *database* yang detail.
7. Mengubah data yang tersimpan ke dalam format apapun menggunakan fungsi yang telah dikenal, seperti menampilkan data BLOB (*Binary Large Object*) sebagai gambar atau *link download*, dll.

3.6 DNS (*Domain Name System*)

Domain Name System (DNS) adalah *Distribute Database System* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang menggunakan TCP/IP (*Transmission Control Protocol / Internet Protocol*). DNS merupakan *aplikasi service* yang biasa digunakan di internet seperti *Web Browser* atau *e-mail* yang menerjemahkan sebuah *domain name* ke *IP Address*. Selain digunakan di internet, DNS juga dapat diimplementasikan ke intranet atau *private network* (VPN).

Sebelum DNS digunakan, jaringan komputer menggunakan file HOST.TXT yang berisi informasi dari nama komputer dan *IP Address*. Di internet, file ini dikelola secara terpusat dan disetiap lokasi harus menambahkan versi terbaru dari file HOSTS tersebut. Bisa dibayangkan betapa repotnya jika ada penambahan 1 komputer di jaringan karena harus menambahkan versi terbaru file ini ke setiap lokasi *host* komputer. Semakin meluasnya jaringan internet, hal itu tentu semakin merepotkan.



Gambar 3.5 Struktur DNS

Struktur domain ditentukan berdasarkan tingkatan kemampuan yang ada di struktur hirarki yang disebut dengan level, dan berikut adalah level hirarki dari domain tersebut :

- **Root Domain**

Root Domain adalah *domain* teratas, dimana domain ini ditulis dalam bentuk notasi titik (.).

- **Top Level Domains**

Top Level Domains yaitu semua *node* yang tepat berada dibawah *root domain*.

Fungsi dari DNS adalah menerjemahkan nama komputer ke *IP Address* (memetakan nama komputer menjadi *IP Address*). *Client* DNS disebut *resolvers* dan DNS *Server* disebut dengan *name servers*. *Resolver* atau *client* mengirimkan permintaan ke *name servers* berupa *queries*.

Name servers akan memproses dengan cara mengecek ke *local database* DNS, dan bila mana yang dicari tidak terdapat di *local database*, maka *name servers* tersebut akan menghubungi *name servers* lain atau akan mengirimkan *message failure*. Jika ternyata permintaan dari *client* tidak ditemukan, proses tersebut disebut dengan *Forward Lookup Query*, yaitu permintaan *client* diproses dengan cara memetakan nama komputer (*host*) ke *IP Address*. Berikut adalah prinsip kerja dari DNS :

1. *Resolvers* mengirimkan *queries* ke *name server*.
2. *Name server* mengecek ke *local database*, atau menghubungi *name server* lainnya. Jika ditemukan maka akan diberitahukan ke *resolvers* dan jika tidak maka akan mengirimkan *failure message*.
3. *Resolvers* menghubungi *host* yang dituju dengan menggunakan *IP Address* yang diberikan *name server*.

DNS sendiri mempunyai beberapa hal yang mendukung kinerjanya, antara lain :

1. *DNS Resolver*, merupakan sebuah program *DNS client* yang dijalankan pada komputer *user* dan menghasilkan *DNS request* untuk keperluan program aplikasi. *Resolver* adalah bagian dari program aplikasi yang berfungsi untuk menjawab pertanyaan program aplikasi tentang *domain*.
2. *Recursive DNS server*, yang akan meneruskan pencarian DNS melalui *respons* (balasan) *query* dari *resolver*, dan mengembalikan jawaban ke *resolver*.
3. *Authoritative DNS server*, adalah bagian yang menangani jawaban - jawaban keluar ke *query* dari *recursor*, pada tiap - tiap bagian jawaban, atau bagian dari penunjukan/penyerahan.

Dari penjelasan di atas DNS dapat disimpulkan mempunyai beberapa keunggulan, diantaranya :

1. Mudah, karena *user* tidak lagi direpotkan untuk mengingat *IP Address* sebuah komputer cukup *host name* saja.
2. Konsisten, *IP Address* sebuah komputer bisa saja berubah, tapi *host name* tidak harus berubah.
3. Simple, *user* hanya menggunakan satu nama untuk mencari nama *domain* baik di internet maupun intranet, meskipun ada banyak *mirror server* yang digunakan.

Dan beberapa kekurangan DNS itu sendiri, diantaranya :

1. DNS tidak mudah untuk diimplementasikan
2. User tidak dapat menggunakan banyak nama untuk mencari nama domain baik di internet maupun intranet.
3. Tidak bisa membuat banyak nama domain.

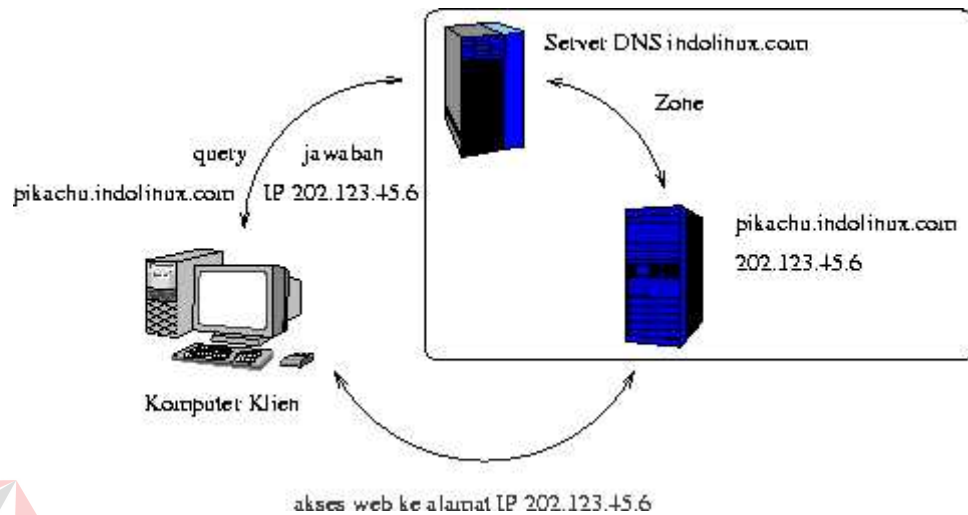
3.7 BIND (*Berkeley Internet Name Domain*)

BIND merupakan salah satu implementasi dari DNS yang paling banyak digunakan pada *server* di Internet. Implementasi DNS pertama adalah JEEVES buatan Paul Mockapetris. BIND dibuat untuk *sistem* operasi BSD UNIX 4.3 oleh Kevin Dunlap, tapi kemudian banyak di *porting* ke banyak turunan UNIX termasuk *Linux*. BIND sampai sekarang masih dikoordinasi oleh *Internet Software Consortium*.

Program utama dari BIND bernama *named* yaitu sebuah *daemon* yang bila dijalankan akan menunggu koneksi pada port 53 (*default*). Koneksi pada port 53 ini adalah koneksi permintaan informasi pemetaan dari nama domain/mesin ke *IP Address* dan sebaliknya. Jika sebuah *server* DNS mempunyai otorisasi terhadap suatu domain, maka *server* DNS tersebut akan memberikan informasi mengenai nama-nama mesin/domain yang berada di bawah domain yang dipegangnya.

Misal, *server* DNS penulis mempunyai otorisasi terhadap domain *indolinux.com*, maka ketika ada sebuah komputer dari Internet ingin mengakses *pikachu.indolinux.com*, maka sang komputer itu akan menghubungi *server* DNS penulis untuk mengetahui informasi *IP Address* dari *pikachu.indolinux.com*. Sang *server* DNS akan menjawab permintaan/*query* dari komputer peminta tersebut

dengan *IP Address* yang sesuai kemudian komputer tersebut dapat mengakses *pikachu.indolinux.com* melalui alamat IP yang diberikan oleh *name server* tersebut.



Gambar 3.5 Pencarian *host pikachu.indolinux.com*

Versi terakhir pada saat artikel ini dibuat adalah BIND versi 9.2.1.

Sebagian besar distribusi *Linux* menyertakan paket BIND. Jadi anda tinggal konfigurasi dan aktifkan saja jika sudah terinstal.

Ada beberapa berkas pendukung BIND, diantaranya :

1. */etc/resolv.conf*

Berkas konfigurasi berisi *domain* atau *IP Address name server* yang pertama dicari oleh *resolver* ketika sebuah domain/nama mesin diminta untuk dipetakan ke *IP Address*.

2. */etc/nsswitch.conf*

Berkas konfigurasi sistem untuk melakukan mekanisme *switch* sistem

database dan *name-service*. *Switch* dapat melalui berkas, *name-server*, atau NIS *server*.

3. */etc/named.conf*

Berkas konfigurasi dari BIND yang utama, berisi informasi mengenai bagaimana *klien* DNS mengakses port 53, letak dan jenis berkas *database* yang diperlukan. Umumnya berisi letak berkas konfigurasi *name server root*, *domain*, *localhost/loopback*, dan *reverse-mapping*.

4. */var/named/named.ca*

Berkas *database name-server root* yang bertanggung jawab terhadap *Top Level Domain* di Internet. Digunakan untuk mencari *domain* di luar

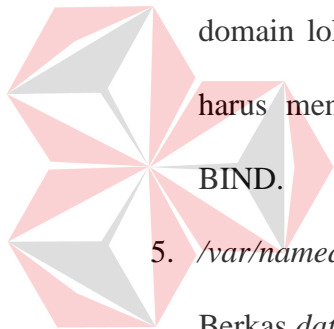
domain lokal. Nama berkas ini bisa anda definisikan sendiri tetapi tetap harus mengacu ke *named.conf* sebagai berkas konfigurasi utama dari BIND.

5. */var/named/named.local*

Berkas *database name-server* untuk alamat *loopback/host* lokal/alamat diri sendiri. Nama berkas ini bisa anda definisikan sendiri tetapi tetap harus mengacu ke *named.conf* sebagai berkas konfigurasi utama dari BIND.

6. */var/named/db.domain-kita.com*

Berkas *database name server* untuk domain *kita.com* yang berisi *resource record*, informasi nama *host* dan *IP Address* yang berada di bawah domain *domain-kita.com*. Berkas ini bisa lebih dari satu tergantung jumlah *domain* yang kita kelola. Nama berkas ini bisa anda definisikan sendiri tetapi tetap harus mengacu ke *named.conf* sebagai berkas konfigurasi utama dari BIND.



7. */etc/rndc.conf*

Berkas konfigurasi program rndc yaitu suatu program untuk administrasi dan kontrol operasi *name-server* BIND.

3.8 *Mail Server*

Mail server adalah sebuah komputer yang berfungsi sebagai kantor pos elektronik untuk *email*. *Mail server* dipertukarkan melalui jaringan dilewatkan antara *server* yang dirancang khusus menggunakan perangkat lunak yang dibuat berdasarkan *protokol* standar untuk menangani *mail*, dan file lampiran.

Internet Service Provider (ISP) memiliki *mail server* untuk menangani *mail klien* mereka, kadang-kadang disebut sebagai *server mail* pribadi. Beberapa situs juga menawarkan layanan *email publik*, memanfaatkan *server* sendiri seperti *gmail* atau *yahoo mail*.

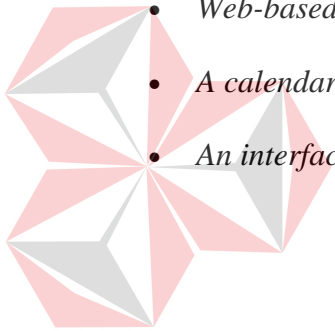
Mail server dapat dibagi menjadi dua kategori utama yaitu *server surat* keluar dan *server surat* masuk. *Server surat* keluar dikenal sebagai SMTP, atau *Simple Mail Transfer Protocol*. *Server surat* masuk datang dalam dua *varietas* utama. POP3, atau *Post Office Protocol* versi 3, *server* yang terbaik dikenal untuk menyimpan pesan yang dikirim dan diterima pada *PC hard drive lokal*. IMAP, atau *Internet Message Access Protocol*, *server* selalu menyimpan salinan pesan pada *server*. Kebanyakan *server* POP3 dapat menyimpan pesan pada *server* juga yang jauh lebih nyaman.

3.9 Squirrelmail

Squirrelmail merupakan aplikasi *webmail* yang ditulis menggunakan PHP dan di dukung dengan PHP murni untuk *protocol* IMAP dan SMTP. Diantara banyak *script webmail Squirrelmail* adalah salah satu *webmail* yang sudah sangat populer.

Beberapa fitur pendukung Squirrelmail antara lain :

- *Spell checking (squirrelspell)*
- *Mail filters (filters)*
- *Web-based administration of SquirrelMail (administrator)*
- *A calendar (calendar)*
- *An interface to submit bug reports semi-automatically (bug_report)*



BAB IV

PEMBAHASAN

4.1 Database dan Webservice Pada *Linux Zorin*

Dalam pembuatan *database* dan *Webservice* pada *Linux Zorin* ini, kita akan menggunakan *MySQL*, *Apache* dan *PhpMyadmin*. *Webservice* sendiri adalah *software* yang menjadi tulang belakang dari *World Wide Web* (*www*). *Webservice* menunggu permintaan *HTTP* atau *HTTPS* dari *client* yang menggunakan *browser* seperti *Netscape Navigator*, *Internet Explorer*, *Mozilla*, dan program *browser* lainnya.

Jika ada permintaan dari *browser*, maka *Webservice* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser*. Data ini mempunyai format yang *standart*, disebut dengan format *SGML* (*Standar General Markup Language*). Data yang berupa format ini kemudian akan ditampilkan oleh *browser* sesuai dengan kemampuan *browser* tersebut.

MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis dibawah lisensi *GNU General Public License* (*GPL*), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*.

Server HTTP Apache atau *server Web/WWW Apache* adalah *Webservice* yang dapat dijalankan di banyak sistem operasi (*Unix*, *BSD*, *Linux*, *Microsoft*

Windows dan Novell Netware serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini menggunakan HTTP.

Berikut ini adalah langkah – langkah dalam menginstall Database dan Webservice pada Linux Zorin:

1. Buka Terminal kemudian install Apache2

```
fajar@fajar-virtual-machine:~$ sudo apt-get install apache2
```

2. Sudo apt-get install mysql-server

```
fajar@fajar-virtual-machine:~$ sudo apt-get install mysql-server
```

3. Sudo apt-get install php5-mysql

```
fajar@fajar-virtual-machine:~$ sudo apt-get install php5-mysql
```

4. Sudo mysql_install_db

```
fajar@fajar-virtual-machine:~$ sudo mysql_install_db
```

5. Sudo /usr/bin/mysql_secure_installation

```
fajar@fajar-virtual-machine:~$ sudo /usr/bin/mysql_secure_installation
```

6. Sudo apt-get install php5

```
fajar@fajar-virtual-machine:~$ sudo apt-get install php5
```

7. Sudo apt-get install libapache2-mod-php5

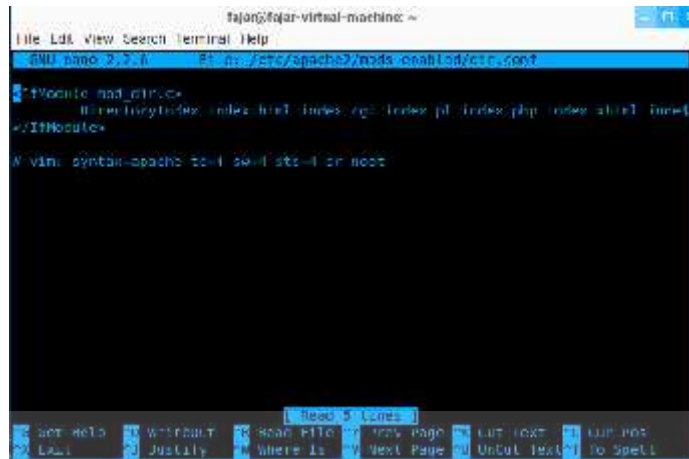
```
fajar@fajar-virtual-machine:~$ sudo apt-get install libapache2-mod-php5
```

8. Sudo apt-get install php5-mcrypt

```
fajar@fajar-virtual-machine:~$ sudo apt-get install php5-mcrypt
```

9. Sudo pico /etc/apache2/mods-enabled/dir.conf → cek apakah ada tulisan "index.php" (tanpa tanda petik)? Jika tidak ada, tambahkan tulisan tersebut.

```
fajar@fajar-virtual-machine:~$ sudo pico /etc/apache2/mods-enabled/dir.conf
```

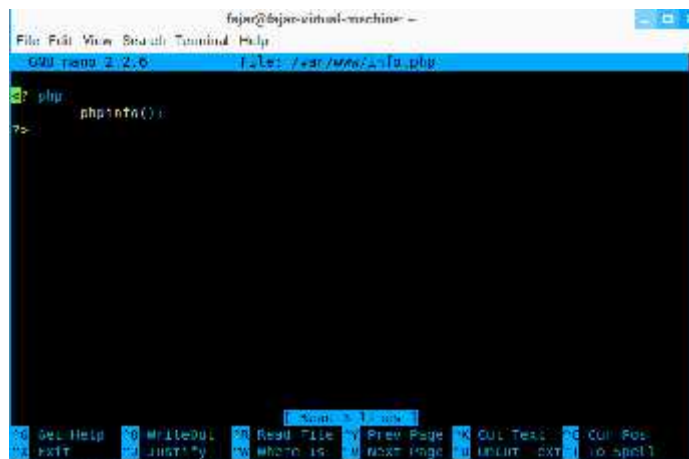


Gambar 4.1 Hasil Tampilan Perintah Di Atas

10. Sudo pico /var/www/info.php

```
fajar@fajar-virtual-machine:~$ sudo pico /var/www/info.php
```

Tambahkan tulisan seperti gambar dibawah ini :



Gambar 4.2 Hasil Tampilan Perintah Di Atas

11. Setelah semua selesai dilakukan, maka masukkan perintah ini di terminal :

Sudo service apache2 restart

12. Kemudian lakukan pengecekan apakah *Database* dan *Webserver* telah berhasil di install. Ketikkan di *url browser localhost linux*.



Gambar 4.3 Hasil Ketika Instalasi Berhasil

Jika tampilannya seperti diatas, maka instalasi telah berhasil.

Setelah proses diatas selesai, selanjutnya adalah melakukan instalasi PhpMyAdmin. Berikut ini adalah langkahnya :

1. install PhpMyAdmin

```
fajar@fajar-virtual-machine:~$ sudo apt-get install phpmyadmin
```

2. setelah install apache 2 selesai, pindah direktori ke /etc/phpmyadmin

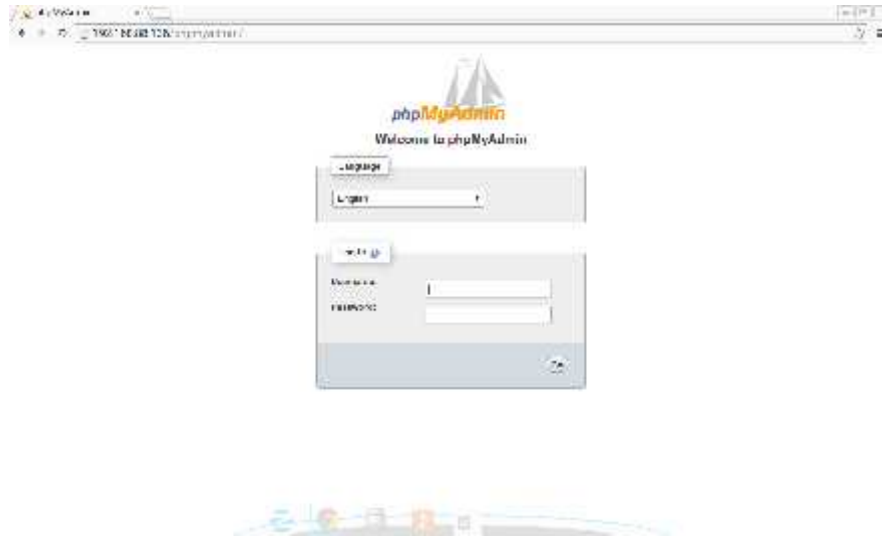
```
fajar@fajar-virtual-machine:~$ cd /etc/phpmyadmin/
```

3. cat apache.conf >> /etc/apache2/apache2.conf

```
fajar@fajar-virtual-machine:/etc/phpmyadmin$ sudo cat apache.conf >> /etc/apache2/apache2.conf
```

4. Setelah semua selesai dilakukan, maka masukkan perintah ini di terminal :
sudo service apache2 restart

untuk melihat apakah proses instalasi telah berhasil, buka browser kemudian ketikkan di *url browser localhost linux/phpmyadmin*. Dan hasilnya akan seperti ini :



Gambar 4.4 Hasil Ketika Instalasi Berhasil

4.2 DNS Server

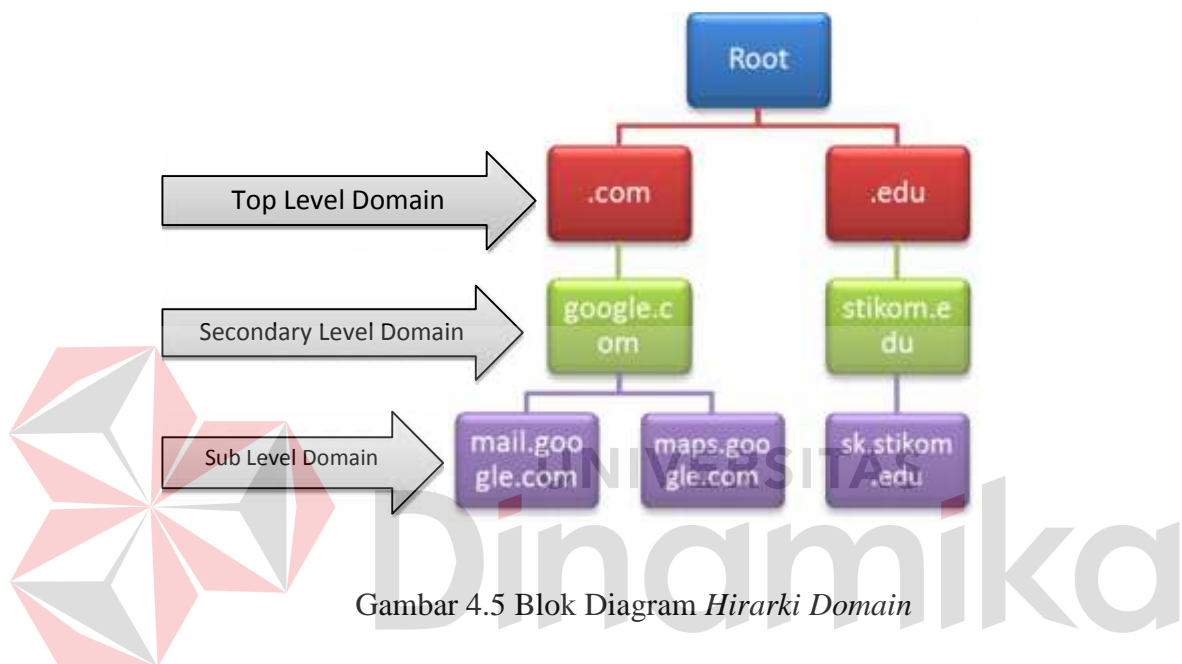
DNS (*Domain Name Server*) adalah sebuah aplikasi yang berfungsi untuk memetakan *IP Address* ke *domain*, atau juga sebaliknya. DNS sangat penting didalam dunia internet, karena jika tidak ada DNS maka akan sulit untuk mengunjungi suatu *website* karena kita harus mengingat *IP Address website* tersebut ataupun *website* lain yang ingin kita kunjungi. Karena itulah DNS dibuat untuk memecahkan permasalahan pemetaan *alamat IP website* tersebut. Pada dasarnya, DNS dibagi menjadi 2 jenis, yaitu :

1. *Primary Name Server* : Bertanggung jawab atas seluruh *domain* (PNS) dan *subdomain* yang ada diseluruh

jaringan internet.

2. *Secondary Name Server* : Sama seperti PNS, namun data yang ada (SNS) pada SNS diambil dari PNS.

DNS juga memiliki *Hirarki* (tingkatan) *domain* seperti yang ada pada gambar dibawah ini.



Gambar 4.5 Blok Diagram *Hirarki Domain*

- Root** : Merupakan *hirarki domain* tertinggi
- Top Level Domain*** : Kata – kata yang ada dibelakang *domain*.
Contohnya : .com , .edu , dll.
- Secondary Level Domain*** : Nama *domain* yang didaftarkan. Contohnya :
stikom.edu, kata stikom pada stikom.edu
merupakan *Secondary Level Domain*.

Sub Level Domain : Merupakan bagian dari *Secondary Level Domain*.
 Contohnya : mail.google.com, kata mail pada mail.google.com merupakan *sub level domain* yang ada pada website google.com.

Berikut ini adalah cara kerja DNS *Server* :

1. Komputer *Client* menggunakan aplikasi *browser* (http) untuk mencari suatu *website*. Aplikasi yang digunakan *client* disebut dengan *Resolver*. Kemudian *resolver* menghubungi DNS *Server*.
2. DNS *Server* akan mengecek apakah *website* yang dicari *client* ada pada *databasenya*. Jika ditemukan *website* yang dicari maka DNS *Server* akan mengembalikan *IP Address website* ke *Resolver*.
3. Kemudian baru *client* bisa secara langsung berhubungan dengan *website* yang diminta.

Berikut adalah langkah – langkah instalasi DNS *server* pada *Linux Zorin* :

1. Buka terminal, kemudian ketikkan *command* “*sudo apt-get install bind9*”.

Setelah itu tunggu sampai proses *instalasi* selesai.

```
fajar@fajar-virtual-machine:~$ sudo apt-get install bind9
```

2. Lakukan konfigurasi di */etc/network/interface*.

```
fajar@fajar-virtual-machine:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/network/interfaces
interfaces(5): file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
Read 3 Lines
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Gambar 4.6 Hasil Tampilan Perintah Di Atas

Kemudian masukan *IP Address*, *Netmask*, *Gateway* dan *dns-nameserver*.

Contohnya seperti ini :

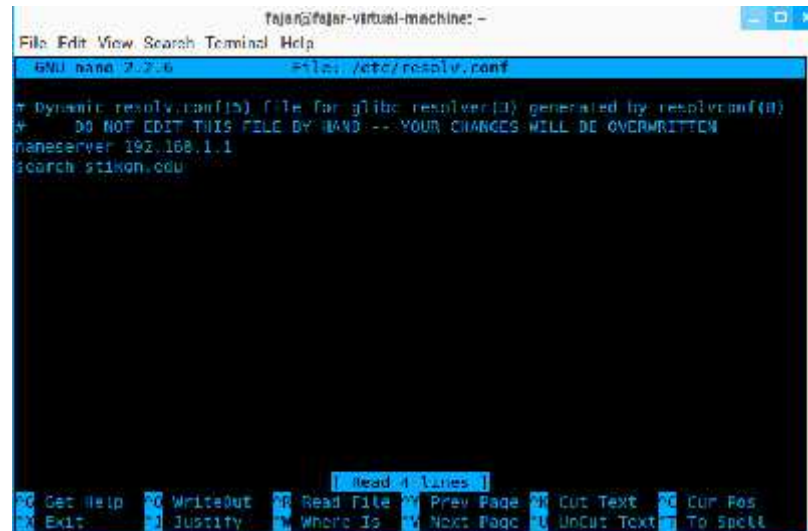
```
fajar@fajar-virtual-machine:~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/network/interfaces
interfaces(5): file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.1
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameserver 192.168.1.1
Read 11 Lines
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Gambar 4.7 Contoh *IP Address*, *Netmask*, *Gateway* dan *dns-nameserver*.

Setelah melakukan perubahan, simpan konfigurasi tersebut.

3. Konfigurasi */etc/resolv.conf*. kemudian isikan nama *domain* dan alamat *dns-nameserver*.

Setelah selesai konfigurasi, *save* dan *restart* dengan cara "*sudo /etc/init.d/networking restart*".




```

fajar@fajar-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /etc/resolv.conf
# Dynamic resolv.conf(15) file for glibc resolver(11) generated by resolvconf(10)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.1
search stikom.edu
  
```

Gambar 4.8 Hasil Tampilan Perintah Di Atas

4. Konfigurasi `/etc/bind/named.conf.options`. kemudian ubah *forwarders* menjadi aktif dan rubah `0.0.0.0` menjadi alamat *IP dns-nameserver*.



```

fajar@fajar-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.6 File: /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
    // parts to talk.  See http://www.kb.cert.org/wiley/id060115

    // 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's placeholders.

    forwarders {
        192.168.1.1;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys.  See https://www.isc.org/bind-keys
  
```

Gambar 4.9 Hasil Tampilan Perintah Di Atas dan Hal Yang Di Rubah

Save setelah selesai mengkonfigurasi.

5. Konfigurasi `/etc/bind/named.conf.local`. kemudian masukan *zone domain* dan *zone IP Address dns-nameserver*.


```
fajar@fajar-virtual-machine: -
File Edit View Search Terminal Help
GNU nano 2.2.6 /etc/bind/named.conf.local Modified
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include */etc/bind/zones.rfc1918*;

zone "stikom.edu" {
    type master;
    file "/etc/bind/stikom.zone";
};

zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/stikom.rev";
};

Go Get Help Write Out Read File Prev Page Cut Text Cut Pos
Exit Justify Where Is Next Page InOut Text To Spell
```

Gambar 4.10 Hasil Tampilan Perintah untuk memasukkan *zone domain* dan *zone IP Address dns-nameserver*.

Save konfigurasi tersebut jika sudah selesai.

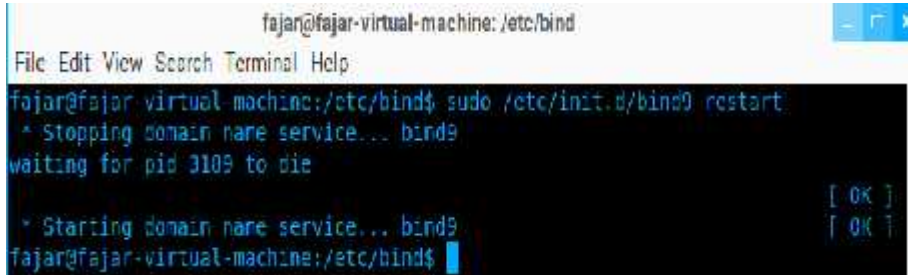
- Pindah ke *direktori bind* dengan mengetikkan “`cd /etc/bind`” kemudian *copy* file `db.local` menjadi file `.zone` dan `.rev` dengan menggunakan perintah “`sudo cp db.local nama_domain.zone`” dan “`sudo cp db.local nama_domain.rev`”. Rubah “*nama_domain*” sesuai dengan nama domain yang dibuat.

```
fajar@fajar-virtual-machine: /etc/bind
File Edit View Search Terminal Help
fajar@fajar-virtual-machine:~$ cd /etc/bind/
fajar@fajar-virtual-machine:/etc/bind$ sudo cp db.local stikom.zone
fajar@fajar-virtual-machine:/etc/bind$ sudo cp db.local stikom.rev
fajar@fajar-virtual-machine:/etc/bind$ ls
bind.keys  db.empty  named.conf.default-zones  stikom.rev
db.0       db.local  named.conf.local          stikom.zone
db.127    db.root   named.conf.options       zones.rfc1918
db.255    named.conf  rndc.key

fajar@fajar-virtual-machine:/etc/bind$
```

Gambar 4.11 Hasil Tampilan Perintah Di Atas

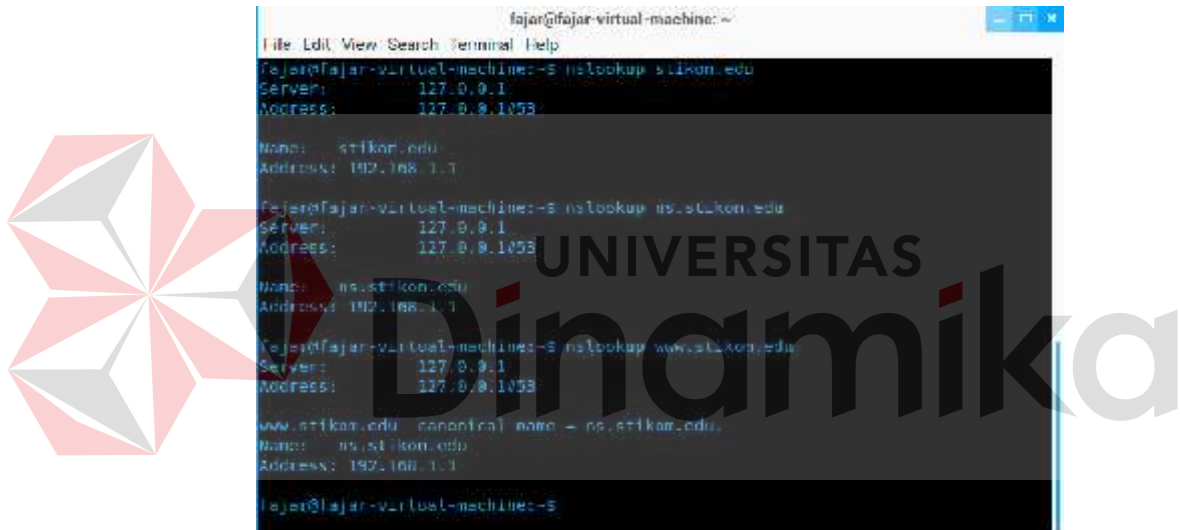
- Konfigurasi file `nama_domain.zone`. kemudian rubah semua kata “`localhost`” menjadi nama *domain* yang dibuat dan rubah tatanan IN menjadi seperti berikut :



```
fajar@fajar-virtual-machine: /etc/bind
File Edit View Search Terminal Help
fajar@fajar-virtual-machine:/etc/bind$ sudo /etc/init.d/bind9 restart
* Stopping domain name service... bind9
waiting for pid 3109 to die
[ OK ]
* Starting domain name service... bind9
[ OK ]
fajar@fajar-virtual-machine:/etc/bind$
```

Gambar 4.14 Contoh Tampilan Saat Melakukan *Restart*

9. Kemudian lakukan pengujian apakah konfigurasi DNS yang dibuat telah berhasil atau belum. Dengan perintah “nslookup” melalui terminal ke alamat *domain* yang dibuat .



```
fajar@fajar-virtual-machine: ~
File Edit View Search Terminal Help
fajar@fajar-virtual-machine:~$ nslookup stikan.edu
Server: 127.0.0.1
Address: 127.0.0.1058
Name: stikan.edu
Address: 192.168.1.1
fajar@fajar-virtual-machine:~$ nslookup ns.stikan.edu
Server: 127.0.0.1
Address: 127.0.0.1058
Name: ns.stikan.edu
Address: 192.168.1.1
fajar@fajar-virtual-machine:~$ nslookup www.stikan.edu
Server: 127.0.0.1
Address: 127.0.0.1058
www.stikan.edu canonical name = ns.stikan.edu
Name: ns.stikan.edu
Address: 192.168.1.1
fajar@fajar-virtual-machine:~$
```

Gambar 4.15 Contoh Pengujian Konfigurasi DNS

10. Lakukan juga pengujian pada komputer *client* dengan cara *setting* terlebih dahulu *IP Address* yang ada di komputer *client*. Kemudian isikan juga *gateway* dan DNS dengan *IP Address* dari *server Ubuntu* yang telah dibuat sebelumnya. Setelah selesai melakukan *setting IP Address*, buka *browser* dan ketikkan nama *domain* yang telah dibuat. Jika muncul tampilan dibawah ini pada *browser*, *dns server* telah berhasil dibuat.

It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Gambar 4.16 Hasil Pengujian Konfigurasi DNS Yang Telah Berhasil

Dilakukan

4.3 Mail Server

Mail server digunakan untuk mengirimkan surat elektronik (*e-mail*) melalui media internet. *Email* membuat pekerjaan pengiriman surat menjadi lebih cepat dan efisien. Untuk membuat *mail server* harus terdapat SMTP dan POP yang digunakan untuk mengirim dan menerima surat elektronik.

Proses pengiriman *email* bisa terjadi karena adanya SMTP (*Simple Mail Transfer Protocol*). Setelah dikirim *email* tersebut akan ditampung sementara di POP (*Post Office Protocol*). Ketika *user email* akun sedang *online*, secara otomatis *mail client* melakukan *sinkronisasi* dari POP.

Berikut adalah langkah – langkah instalisasi mail server pada *Linux Zorin* sebagai berikut:

1. Buka terminal, kemudian ketikkan *command* “*sudo apt-get install squirrelmail postfix courier-imap courier-pop*”. Pada saat *instalasi* akan muncul pop up, untuk mengisinya ikuti contoh berikut :

- *General Type of Mail Configuration* : *internet site*
- *System Mail Name* : *stikom.com*

Setelah itu tunggu sampai proses *instalasi* selesai.

2. Setelah proses *instalasi* selesai, hal pertama yang dilakukan adalah mengkonfigurasi file *apache2.conf*. Buka file tersebut dengan menggunakan perintah “*sudo pico /etc/apache2/apache2.conf*” (bisa juga menggunakan perintah editor yang lain). kemudian isikan *script* berikut dibagian paling bawah dari isi file tersebut :

Include /etc/squirrelmail/apache.conf



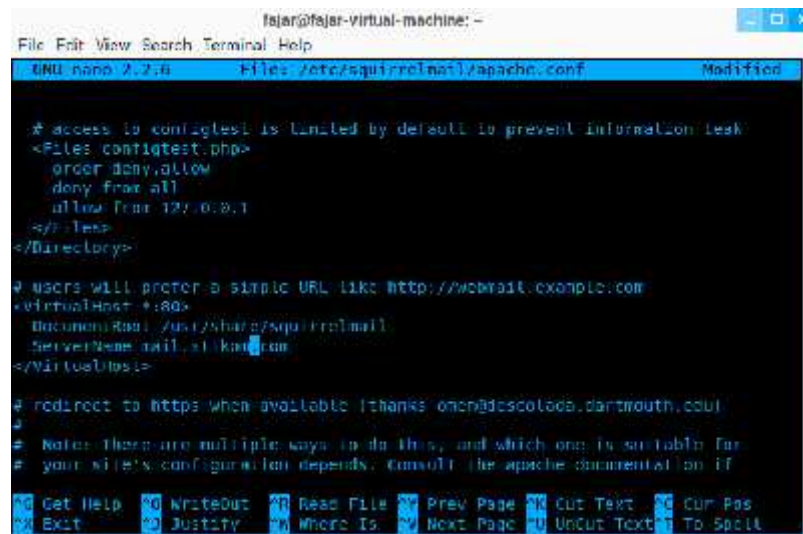
```
fajar@fajar-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.8 File: /etc/apache2/apache2.conf Modified
  Deny from All
</Directory>
<Directory /usr/share/phpmyadmin/setup/lib>
  Order Deny,Allow
  Deny from All
</Directory>
include /etc/squirrelmail/apache.conf
```

Gambar 4.17 Contoh Perintah Yang Harus Di Kerjakan

Kemudian *save* file setelah menambahkan *script* tersebut.

3. Lakukan konfigurasi file *apache.conf* yang berada pada direktori *squirrelmail* dengan menggunakan perintah “*sudo pico /etc/squirrelmail/apache.conf*”. Carilah *script* pada file tersebut yang

mempunyai tag “<virtualhost></virtualhost>”, kemudian rubahlah sehingga menjadi seperti gambar dibawah ini :



```

fajar@fajar-virtual-machine: ~
File Edit View Search Terminal Help
GNU nano 2.2.6      File: /etc/squirrelmail/apache.conf      Modified

# access to configtest is limited by default to prevent information leak
<File: configtest.php>
  order deny,allow
  deny from all
  allow from 127.0.0.1
</File>
</Directory>

# users will prefer a simple URL like http://www.vobvaid.com
<VirtualHost *:80>
  DocumentRoot /usr/share/squirrelmail
  ServerName mail.vobvaid.com
</VirtualHost>

# redirect to http://www.vobvaid.com when available (thanks onch@ascotada.dartmouth.edu)
#
# Note: There are multiple ways to do this, and which one is suitable for
# your site's configuration depends. Consult the apache documentation if
# you need more information.

Get Help  Writeout  Read File  Prev Page  Cut Text  Cur Pos
Exit     Justify   Where Is  Next Page  UnCut Text To Spell
  
```

Gambar 4.18 Contoh Perintah Yang Harus Di Kerjakan

Jika sudah, *save* file tersebut.

- Langkah selanjutnya adalah membuat file untuk *mailbox* (pesan masuk) untuk *user*. Hal ini dilakukan untuk menampung *email* yang masuk, dan akan tertampung di file tersebut. Gunakan perintah berikut untuk membuatnya :

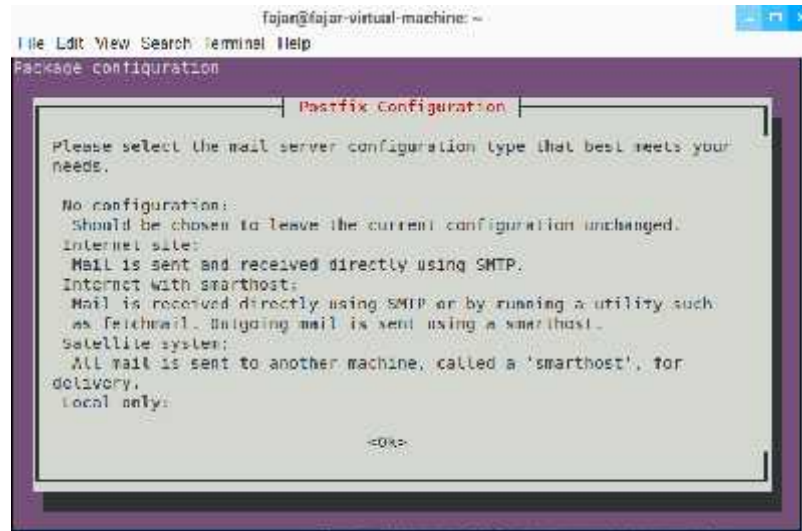
Maildirmake /etc/skel/Maildir

- Buatlah *user* baru untuk akun *email* yang nantinya digunakan dalam *web mail* dengan menggunakan perintah berikut :

Adduser nama_user

Ganti “nama_user” sesuai dengan keinginan.

- Konfigurasi paket *postfix* yang ada dengan menggunakan perintah “*dpkg-reconfigure postfix*”. Setelah beberapa saat kemudian akan muncul pop up untuk menanyakan beberapa setingan. Ikuti contoh gambar berikut ini :



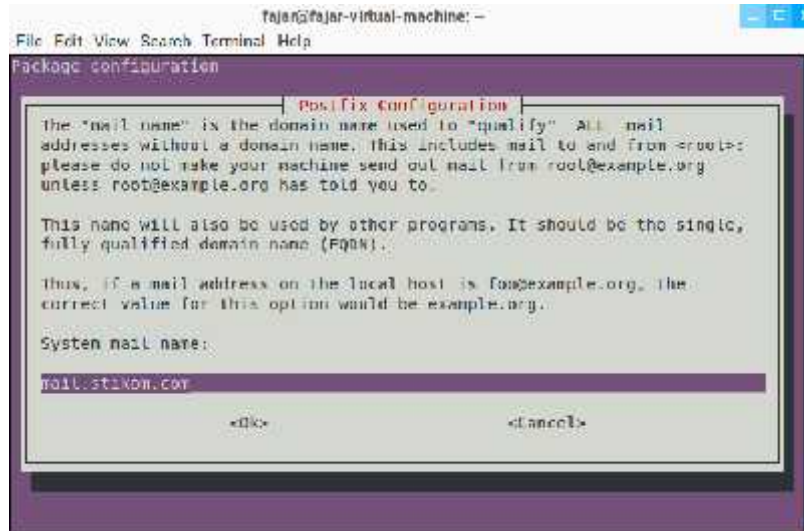
Gambar 4.19 Langkah – Langkah Konfigurasi

Pilih OK, kemudian pilih *internet site* pada pilihan selanjutnya



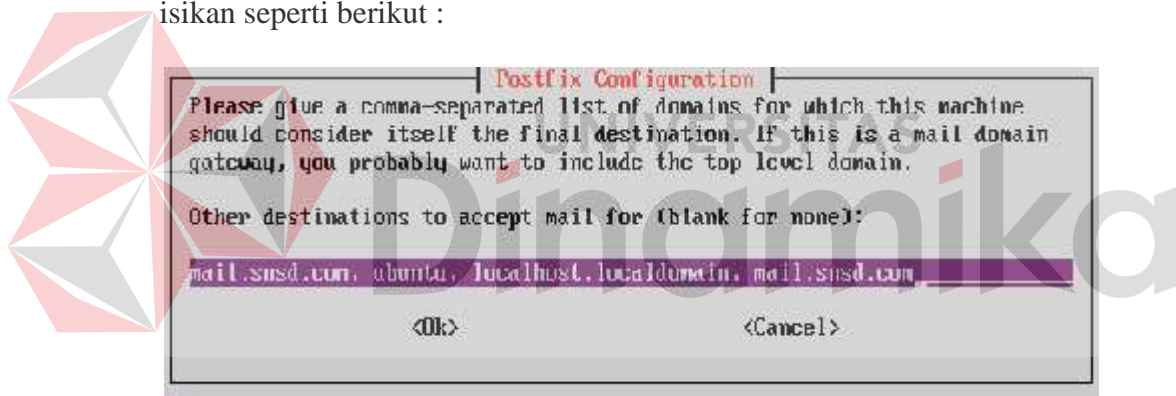
Gambar 4.20 Pemilihan Step Selanjutnya

Lalu isikan nama *domain mail server* seperti berikut :



Gambar 4.21 Pengisian Nama *Domain Mail Server*

root dan *postmaster* dikosongkan, kemudian Ok. Pada pop up selanjutnya isikan seperti berikut :



Gambar 4.22 Pengisian pada Pop Up

untuk *mail queue* pilih saja no, dan pada pop up selanjutnya isikan bagian depan dengan alamat network yang kita punya seperti berikut :



Gambar 4.23 Pengisian Alamat Network Yang Kita Miliki

Untuk *mailbox size* dan *local address* dikosongkan saja. Dan untuk *internet protocol* pilih yang *ipv4*, seperti berikut :



Gambar 4.24 Pemilihan Step Selanjutnya

7. Setelah melakukan konfigurasi *postfix*, selanjutnya adalah mengkonfigurasi file *main.cf* yang ada pada *direktori postfix* dengan menggunakan perintah "*sudo pico /etc/postfix/main.cf*". Pada bagian bawah dari isi file tersebut, tambahkan script dibawah ini :

home_mailbox = Maildir/

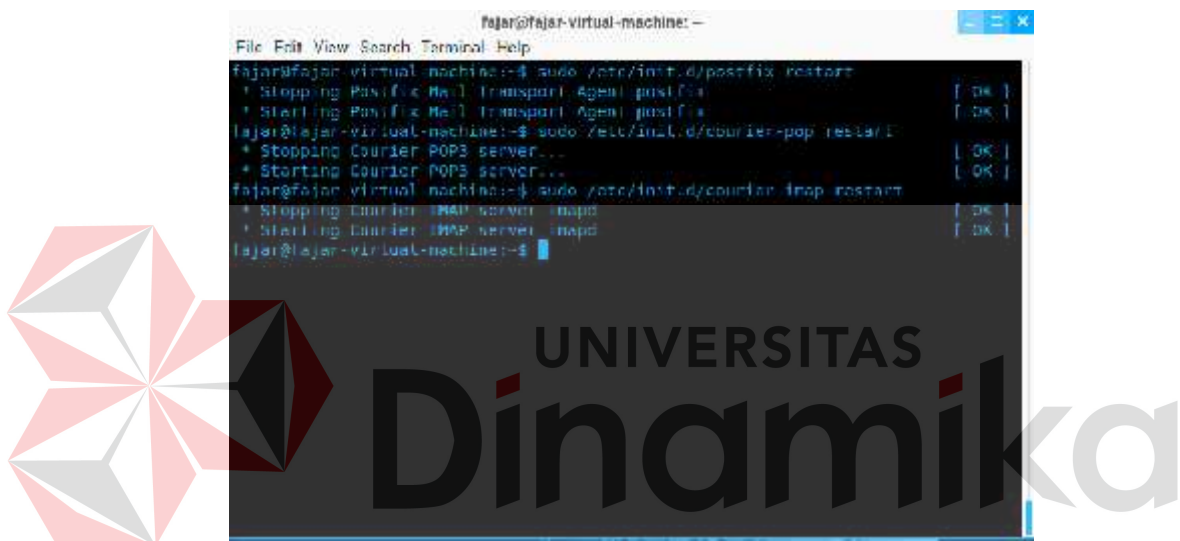
script tersebut digunakan sebagai tempat penampungan jika ada *mailbox* yang masuk. Jika sudah menambahkan *script* diatas, *save* file tersebut.

- Restart paket postfix, courier-pop dan courier-imap dengan menggunakan 3 perintah dibawah ini :

```
sudo /etc/init.d/postfix restart
```

```
sudo /etc/init.d/courier-pop restart
```

```
sudo /etc/init.d/courier-imap restart
```



Gambar 4.25 Contoh Tampilan Saat Melakukan *Restart*

- Untuk pengujian, bukalah browser dan ketikkan nama *domain* dari *mail server* yang dibuat. Kemudian cobalah login menggunakan *username* dan *password* yang telah dibuat sebelumnya.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil kerja praktek dapat diperoleh kesimpulan sebagai berikut :

1. *Linux Zorin* mempunyai banyak kelebihan dibandingkan *Ubuntu*, pengguna baru yang ingin menggunakan *Linux* direkomendasi untuk menggunakannya karena tampilannya seperti pada *Windows* sehingga sangat mudah di gunakan bagi pengguna awam.
2. Untuk pemanfaatan *Linux Zorin* sebagai server juga sangat mudah dibandingkan kita menggunakan *Windows*. *Command –command* yang digunakan sama seperti *Ubuntu* pada umumnya jadi sangat memudahkan pengguna awam untuk menggunakannya.

5.2 Saran

Lebih mengeksplor lagi apa saja yang dapat kita manfaatkan untuk *Linux Zorin* kedepannya. Contoh bisa membahas tentang tampilan yang hampir sama seperti *Windows* atau lebih dalam bisa membahas tentang masalah keamanan jaringan berbasis *Linux Zorin* itu sendiri.

DAFTAR PUSTAKA

Maniacms. 2012. Pengertian apache web server. (online)

<http://www.maniacms.web.id/2012/01/pengertian-apache-web-server.html>

. Diakses pada tanggal 31 Oktober 2014.

Kompasiana. 2013. Mengenal Web Server. (online)

[http://teknologi.kompasiana.com/internet/2013/01/18/mengenal-web-](http://teknologi.kompasiana.com/internet/2013/01/18/mengenal-web-server-525773.html)

[server-525773.html](http://teknologi.kompasiana.com/internet/2013/01/18/mengenal-web-server-525773.html). Diakses pada tanggal 31 Oktober 2014.

Hastomo. 2013. Pengertian dan Kelebihan Apache Server. (online)

<http://hastomo.net/php/pengertian-dan-kelebihan-apache-server/> Diakses pada tanggal 31 Oktober 2014.

Hastomo. 2013. Pengertian dan Kelebihan Database MySQL. (online)

<http://hastomo.net/php/pengertian-dan-kelebihan-database-mysql/> Diakses pada tanggal 31 Oktober 2014.

Hastomo. 2013. Kelebihan PHP. (online)

<http://hastomo.net/php/kelebihan-php/> Diakses pada tanggal 31 Oktober 2014.