2011

INSTALL & KONFIGURASI DEBIAN 5.0 (LENNY)

UNIT PRODUKSI

Nama : ……
Kelas : ……
Alamat : ……

3 TKJ
SMKN BANTARKALONG
11/1/2011

CARA MENGINSTALL DEBIAN LENNY (5.0)

Sebelum Kita melakukan instalasi DEBIAN 5.0 (Lenny) kita mesti tau terlebih dahulu sejarah Debian

SEJARAH

1993–1998

Debian ini pertama kali diumumkan pada tanggal 16 Agustus 1993 oleh Ian Murdock. Murdock awalnya disebut sistem "Debian Linux Release". Sebelum rilis Debian, Softlanding Linux System (SLS) telah pertama distribusi GNU / Linux yang dikumpulkan dari berbagai paket perangkat lunak, dan merupakan dasar untuk populer distro lain di 1993-1994. Miskin yang dirasakan pemeliharaan dan prevalensi bug di SLS Murdock termotivasi untuk memulai distribusi baru.

Pada tahun 1993 Murdock juga merilis Debian Manifesto, menguraikan pandangannya untuk sistem operasi baru. Di dalamnya ia menyerukan penciptaan distribusi dipertahankan secara terbuka, dalam semangat Linux dan GNU. Ia membentuk nama "Debian" sebagai kombinasi dari nama pertama nya kemudian pacarnya Debra Lynn dan nama sendiri.
Proyek Debian tumbuh lambat pada awalnya dan merilis versi 0.9x pertama pada tahun 1994 dan 1995. Pelabuhan pertama lain, non-i386 arsitektur dimulai pada tahun 1995, dan yang pertama versi 1.x Debian dirilis pada tahun 1996.

Pada tahun 1996, Bruce Ian Murdock Perens digantikan sebagai pemimpin proyek. Pada tahun yang sama, sesama pengembang Ean Schuessler menyarankan bahwa Debian harus membangun kontrak sosial dengan para penggunanya. Dia suling hasil diskusi di milis Debian ke Kontrak Sosial Debian, dan Debian Free Software Guidelines, mendefinisikan komitmen mendasar untuk pengembangan distribusi. Ia juga memprakarsai pembentukan organisasi payung hukum, Software di Kepentingan Umum.

Perens meninggalkan proyek pada tahun 1998 sebelum rilis glibc pertama berbasis Debian, 2.0.
1999–2004

Proyek terpilih pemimpin baru dan membuat dua lebih 2.x rilis, masing-masing termasuk lebih port dan paket. Advanced Packaging Tool ini digunakan selama waktu dan port pertama non-kernel Linux, Debian GNU / Hurd, dimulai. Distribusi Linux yang pertama berbasis pada Debian, yaitu Libranet, Corel Linux dan Stormix's Storm Linux, yang dimulai pada tahun 1999. 2.2 rilis pada tahun 2000 didedikasikan untuk Joel Klecker, seorang pengembang yang meninggal karena Duchenne distrofi otot.

Pada akhir 2000, proyek membuat perubahan besar untuk mengarsipkan dan melepaskan manajemen, proses arsip perangkat lunak reorganisasi baru "paket renang" dan menciptakan distribusi pengujian sebagai yang berkelanjutan dan relatif stabil wilayah pemanggungan untuk rilis berikutnya. Pada tahun yang sama, para pengembang mulai memegang debconf konferensi tahunan yang disebut dengan pembicaraan dan lokakarya untuk para pengembang dan pengguna teknis.

Pada bulan Juli 2002, Proyek merilis versi 3.0, nama kode kayu, (setelah karakter di film Toy Story, sebuah tren yang berlanjut hingga sekarang), sebuah rilis stabil yang akan melihat relatif sedikit pembaruan sampai rilis berikut.

Panjang siklus rilis yang dipekerjakan oleh Proyek Debian selama waktu ini menarik banyak kritik dari komunitas perangkat lunak bebas, dan ini memicu penciptaan Ubuntu pada tahun 2004, sampai saat ini salah satu yang paling berpengaruh Debian garpu.

2005-sekarang

Sarge 3,1 rilis yang dibuat pada bulan Juni 2005. Ada banyak perubahan besar dalam rilis sarge, kebanyakan karena besar waktu yang dibutuhkan untuk membekukan dan melepaskan distribusi. Tidak hanya melakukan update rilis ini lebih dari 73% dari software dikirim dalam versi sebelumnya, tetapi juga mencakup jauh lebih lunak daripada rilis sebelumnya, hampir dua kali lipat ukuran dengan 9.000 paket baru. Installer baru menggantikan boot-floppy penuaan installer dengan desain modular. Hal ini memungkinkan instalasi lanjutan (dengan RAID, XFS dan dukungan LVM) termasuk deteksi hardware, membuat instalasi lebih mudah bagi pengguna pemula. Sistem instalasi juga menyombongkan penuh dukungan internasionalisasi sebagai perangkat lunak diterjemahkan ke dalam hampir empat puluh bahasa. Instalasi manual dan komprehensif catatan rilis dibebaskan dalam sepuluh dan lima belas bahasa yang berbeda masing-masing. Rilis ini meliputi upaya Debian-Edu/Skolelinux, Debian-Med dan Debian-Accessibility sub-proyek yang meningkatkan jumlah paket pendidikan dan mereka yang memiliki afiliasi medis serta paket yang dirancang khusus untuk para penyandang cacat

Pada tahun 2006, sebagai hasil dari banyak dipublikasikan sengketa, perangkat lunak Mozilla namanya pun kembali berganti dalam Debian, dengan Firefox menjadi Iceweasel, Thunderbird menjadi Icedove, bersama dengan program Mozilla lain. Mozilla Corporation menyatakan bahwa Debian tidak boleh menggunakan merek dagang Firefox jika mendistribusikan Firefox dengan modifikasi yang belum disetujui oleh Mozilla Corporation. Dua alasan yang menonjol Debian Firefox memodifikasi perangkat lunak untuk mengubah karya seni, dan untuk menyediakan patch keamanan. Debian Free Software Guidelines menganggap karya seni Mozilla non-free. Debian menyediakan dukungan jangka panjang untuk versi Firefox di rilis stabil, di mana Mozilla lebih suka yang versi lama tidak didukung. Perangkat lunak sebagian besar program yang dikembangkan oleh Mozilla Corporation itu diganti merek tetapi program 'kode sumber tetap sama hanya dengan perbedaan kecil.

Debian 4.0 (etch) dirilis April 8, 2007 untuk jumlah yang sama seperti pada sarge arsitektur. Ini termasuk pelabuhan AMD64 tapi menjatuhkan dukungan untuk m68k. Pelabuhan yang m68k Namun, masih tersedia dalam distribusi tidak stabil. Ada sekitar 18.200 paket binari dikelola oleh lebih dari 1.030 pengembang Debian.

Debian 5.0 (lenny) diluncurkan 14 Februari 2009 setelah 22 bulan pembangunan. Ini mencakup lebih dari 25.000 paket perangkat lunak. Dukungan ini telah ditambahkan untuk Marvell's Orion platform dan untuk netbook seperti Asus Eee PC. Rilis didedikasikan untuk Thiemo Seufer, pengembang aktif dan anggota masyarakat yang meninggal dalam kecelakaan mobil pada 26 Desember 2008

1. Untuk menginstall linux debian versi 5.0, ikuti langkah langkah dari gambar berikut:

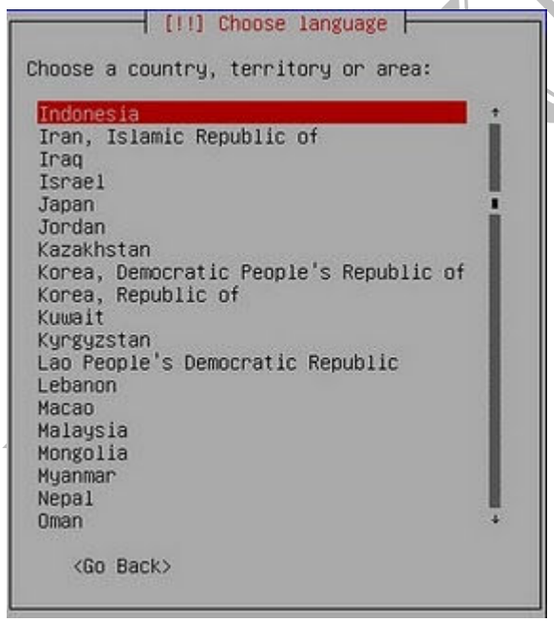
   Pilih Instal

2. Pilih English

3. Pilih Other

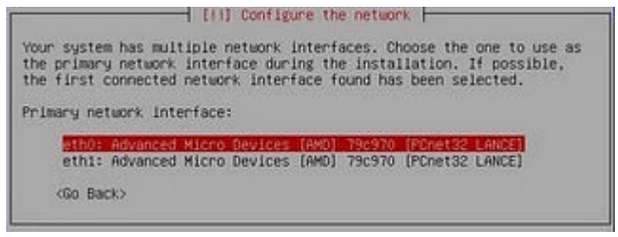
4. Pilih Asia

5. Pilih Indonesia

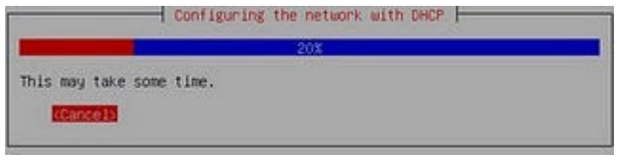


6. Pilih American English

7. Pilih eth0

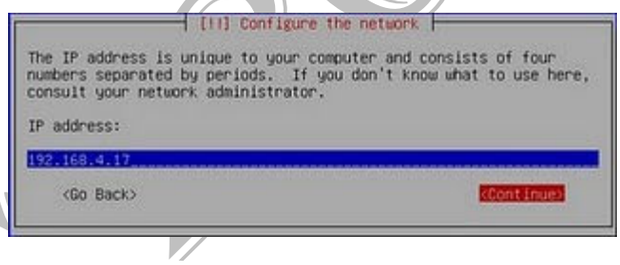


8. Tekan Cancel

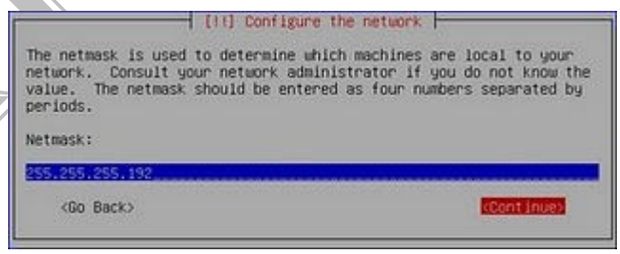


9. Pilih Configure Network Manually

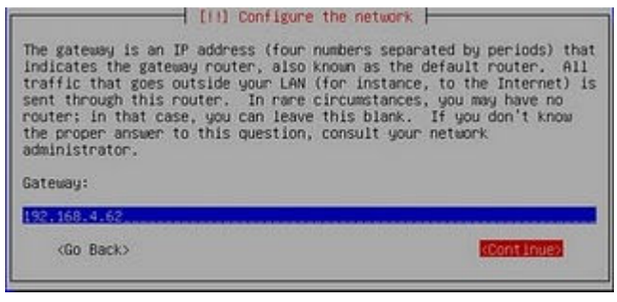
10. Isikan IP address



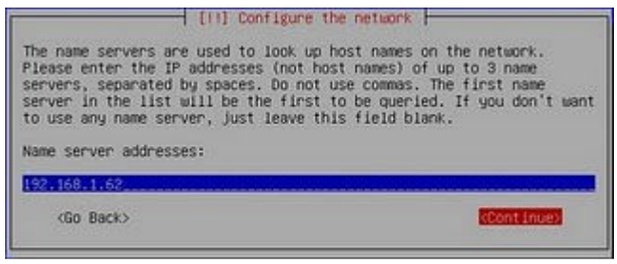
11. Isikan Netmask



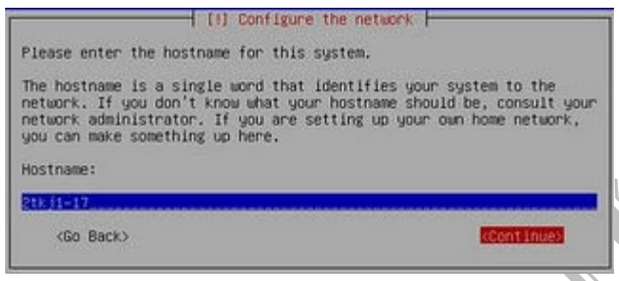
12. Isikan Gateway.



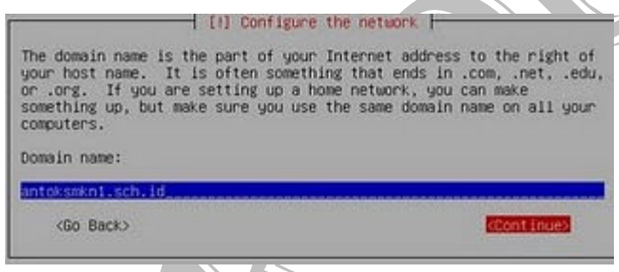
13. Isikan Nameserver. (DNS)



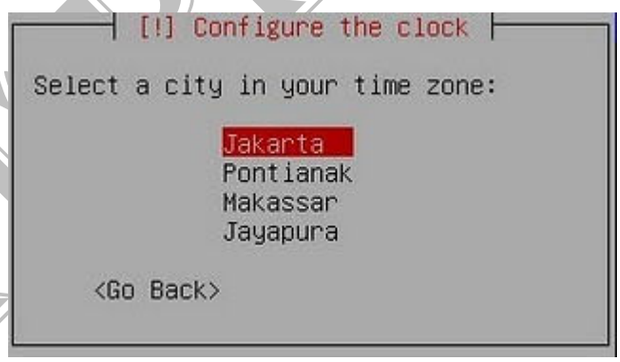
14. Isikan hostname.



15. Isikan Nama Web Lokal



16. Pilih Jakarta

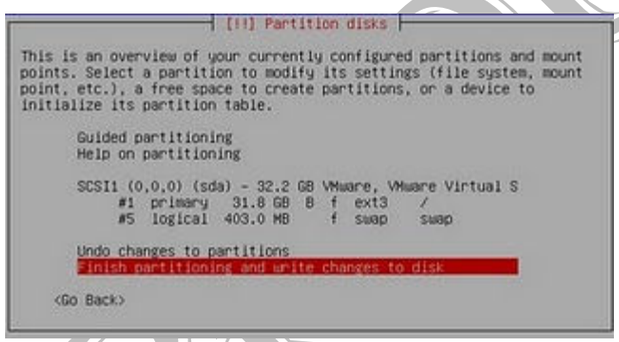


17. Pilih Guided – Use entire disk

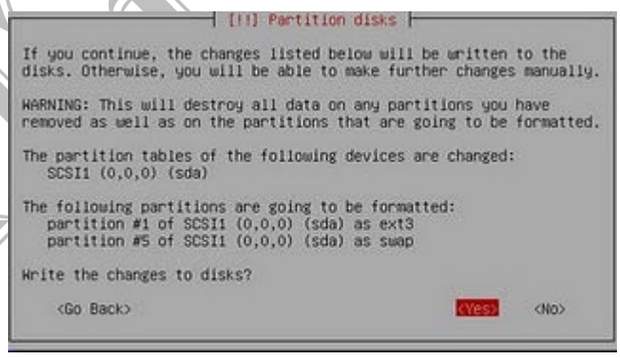
18. Pilih nama harddisk

19. Pilih All files in one partition….

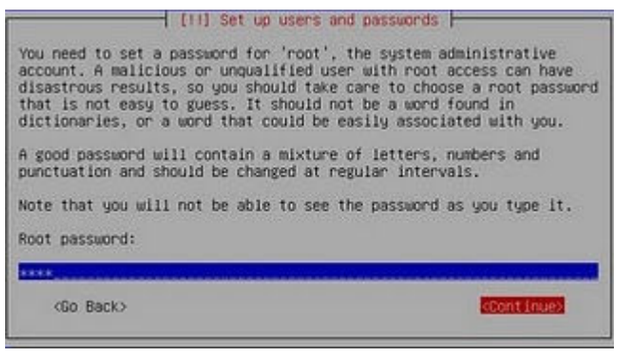
20. Pilih Finish



21. Pilih Yes



22. Masukkan root password

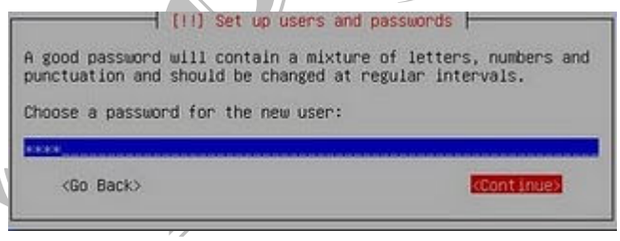


23. Masukkan lagi.

24. Masukkan Full Name User

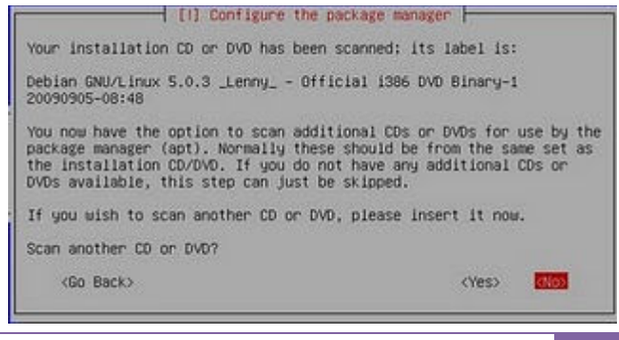
25. Masukkan Username

26. Masukkan Password untuk user.

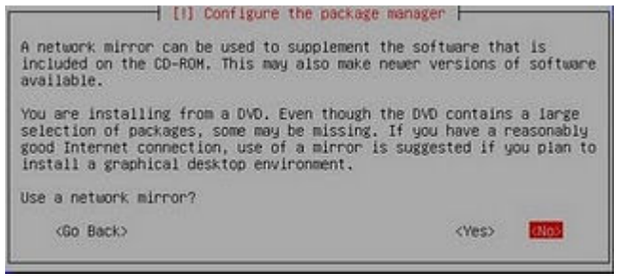


27. Masukkan lagi.

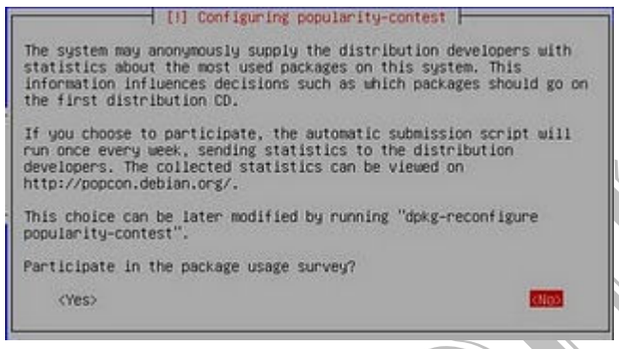
28. Pilih No.



29. Pilih No

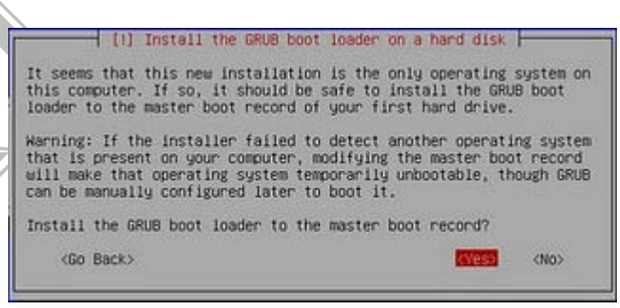


30. Pilih No



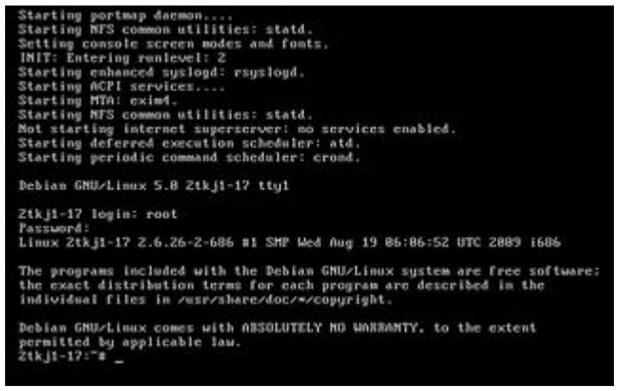
31. Pilih Standard System saja

32. Pilih Yes.



33. Pilih Continue

34. Login Sebagai Root, dan masukkan password.



Dan selesai..!

Konfigurasi Dns sErVEr

1) Teori DNS

DNS (Domain Name System, bahasa Indonesia: Sistem Penamaan Domain) adalah sebuah sistem yang menyimpan informasi tentang nama host maupun nama domain dalam bentuk basis data tersebar (distributed database) di dalam jaringan komputer, misalkan: Internet. DNS menyediakan alamat IP untuk setiap nama host dan mendata setiap server transmisi surat (mail exchange server) yang menerima surat elektronik (email) untuk setiap domain. DNS menyediakan servis yang cukup penting untuk Internet, bilamana perangkat keras komputer dan jaringan bekerja dengan alamat IP untuk mengerjakan tugas seperti pengalamatan dan penjaluran (routing), manusia pada umumnya lebih memilih untuk menggunakan nama host dan nama domain, contohnya adalah penunjukan sumber universal (URL) dan alamat e-mail. DNS menghubungkan kebutuhan ini.

Sejarah singkat DNS

Penggunaan nama sebagai pengabstraksi alamat mesin di sebuah jaringan komputer yang lebih dikenal oleh manusia mengalahkan TCP/IP, dan kembali ke zaman ARPAnet. Dahulu, setiap komputer di jaringan komputer menggunakan file HOSTS.TXT dari SRI (sekarang SIR International), yang memetakan sebuah alamat ke sebuah nama (secara teknis, file ini masih ada - sebagian besa sistem operasi modern menggunakannya baik secara baku maupun melalui konfigurasi, dapat melihat Hosts file untuk menyamakan sebuah nama host menjadi sebuah alamat IP sebelum melakukan pencarian via DNS). Namun, sistem tersebut diatas mewarisi beberapa keterbatasan yang mencolok dari sisi prasyarat, setiap saat sebuah alamat komputer berubah, setiap sistem yang hendak berhubungan dengan komputer tersebut harus melakukan update terhadap file Hosts.

Dengan berkembangnya jaringan komputer, membutuhkan sistem yang bisa dikembangkan: sebuah sistem yang bisa mengganti alamat host hanya di satu tempat, host lain akan mempelajari perubaha tersebut secara dinamis. Inilah DNS. Paul Mockapetris menemukan DNS di tahun 1983; spesifikasi asli muncul di RFC 882 dan 883. Tahun 1987, penerbitan RFC 1034 dan RFC 1035 membuat update terhadap spesifikasi DNS. Hal ini membuat RFC 882 dan RFC 883 tidak berlaku lagi. Beberapa RFC terkini telah memproposikan beberapa tambahan dari protokol inti DNS.

Teori bekerja DNS

Pengelola dari sistem DNS terdiri dari tiga komponen:

- ***DNS resolver***, sebuah program klien yang berjalan di komputer pengguna, yang membuat permintaan DNS dari program aplikasi.
- ***Recursive DNS server***, yang melakukan pencarian melalui DNS sebagai tanggapan permintaan dari resolver, dan mengembalikan jawaban kepada para resolver tersebut;dan .
- ***Authoritative DNS*** server yang memberikan jawaban terhadap permintaan dari recursor, baik dalam bentuk sebuah jawaban, maupun dalam bentuk delegasi (misalkan: mereferensikan ke authoritative DNS server lainnya)

Pengertian Beberapa Bagian Dari Nama Domain

Sebuah nama domain biasanya terdiri dari dua bagian atau lebih (secara teknis disebut label), dipisahkan dengan titik.

- Label paling kanan menyatakan top-level domain - domain tingkat atas/tinggi (misalkan, alamat www.wikipedia.org memiliki top-level domain org).
- Setiap label di sebelah kirinya menyatakan sebuah sub-divisi atau subdomain dari domain yang lebih tinggi. Catatan: "subdomain" menyatakan ketergantungan relatif, bukan absolut. Contoh: wikipedia.org merupakan subdomain dari domain org, dan id.wikipedia.org dapat membentuk subdomain dari domain wikipedia.org (pada prakteknya, id.wikipedia.org sesungguhnya mewakili sebuah nama host - lihat dibawah). Secara teori, pembagian seperti ini dapat mencapai kedalaman 127 level, dan setiap label dapat terbentuk sampai dengan 63 karakter, selama total nama domain tidak melebihi panjang 255 karakter. Tetapi secara praktek, beberapa pendaftar nama domain (domain name registry) memiliki batas yang lebih sedikit.
- Terakhir, bagian paling kiri dari bagian nama domain (biasanya) menyatakan nama host. Sisa dari nama domain menyatakan cara untuk membangun jalur logis untuk informasi yang dibutuhkan; nama host adalah tujuan sebenarnya dari nama sistem yang dicari alamat IP-nya. Contoh: nama domain www.wikipedia.org memiliki nama host "www".

Sebuah contoh dari teori rekursif DNS

Sebuah contoh mungkin dapat memperjelas proses ini. Andaikan ada aplikasi yang memerlukan pencarian alamat IP dari www.wikipedia.org. Aplikasi tersebut bertanya ke DNS recursor lokal.

- Sebelum dimulai, recursor harus mengetahui dimana dapat menemukan root nameserver; administrator dari recursive DNS server secara manual mengatur (dan melakukan update secara berkala) sebuah file dengan nama root hints zone (panduan akar DNS) yang menyatakan alamat-alamt IP dari para server tersebut.
- Proses dimulai oleh recursor yang bertanya kepada para root server tersebut - misalkan: server dengan alamat IP "198.41.0.4" - pertanyaan "apakah alamat IP dari www.wikipedia.org?"
- Root server menjawab dengan sebuah delegasi, arti kasarnya: "Saya tidak tahu alamat IP dari www.wikipedia.org, tapi saya "tahu" bahwa server DNS di 204.74.112.1 memiliki informasi tentang domain org."
- Recursor DNS lokal kemudian bertanya kepada server DNS (yaitu: 204.74.112.1) pertanyaan yang sama seperti yang diberikan kepada root server. "apa alamat IP dari www.wikipedia.org?". (umumnya) akan didapatkan jawaban yang sejenis, "sayatidak tahu alamat dari www.wikipedia.org, tapi saya "tahu" bahwa server 207.142.131.234 memiliki informasi dari domain wikipedia.org."
- Akhirnya, pertanyaan beralih kepada server DNS ketiga (207.142.131.234), yang menjawab dengan alamat IP yang dibutuhkan. Aplikasi untuk membuat atau membangun sebuah DNS SERVER kita menggunakan bind9,

Aplikasi bind9 ini ada tidak akan terpasang langsung dalam sebuah pc server jika dalam penginstalan tidak di otomatiskan untuk membangun sebuah DNS SERVER, maka untuk membuat atau membangunnya dapat kita install secara manual, yaitu dengan perintah apt-get install, paket DNS server atau bind9, sudah disertakan pada cd atau dvb debian yang kita gunakan, semisal kita menggunakan dvd sebagai medianya, maka paket bind9 tersebut berada pada dvd biner 1. Kita langsung aja pada tahap instalasi dan konfigurasi,, .

2) Instalasi paket bind9 pada deibian 5.0

- Masukan DVD binary 1,
- Ketikan perintah

```
# apt-get install bind9
```

- Lalu enter,, tungu hingga proses instalasi selesai..,

Catatan,
"untuk semua penginstalan atau konfigurasi suatu paket, kita harus berada di superusre atau root (#), atau kita harus bekerja di dalam root, jangan didalam userer biasa ($).. jadi kalau tanda kita berada dalam root, maka tandanya di awal console bertanda pagar (#)..

3) Konfigurasi Bind9

File yang kita konfigurasi dalam pembuatan atau membangun sebuah DNS server kita hanya mengkonfigurasi atau mengeditg 3 file,, yaitu file-file sbb..:

- named.conf
- db.domain
- db.ip

File named.conf yaitu file konfigurasi utama dalam bind9 yang secara default sudah terkonfiugarasi sebagai dns cache (resolver) pada saat instalasi bind9, file named.conf ini yang mengkonfigurasi penyetingan atau konfigurasi-konfigurasi untuk pembangunan sebuah dns server,, yaitu yang menjalankan dan menunujukan file database-data base yang menjadi sebuah file data base yang menjalankan proses jalannya bind9 dalam server dns,

Dan kedua file yang lainnya yaitu db.domain dan db.ip itu adalah file yang dikonfigurasi oleh named.conf, yaitu file database yang menjalankan program domain name pada system server, dan di situ ada file db.domain, yaitu file database yang mengatur suatu domain bekerja dalam suatu system,, dan file db.ip adalah file database yang mengatur atau mengkonfigurasi suatu alamat ip server yang dijadikan domain pada suatu server dns,, yang memungkinkan keterkaitan antara file db.domain dan db.ip dan file named.conf yang menjalankan kedua file tersebut,, jadi file named.conf hanyalah sebuah file yang mengatur dan mengkolnfigurasi kedua file database tersebut dapat berjalan,,

File db.domain maksudnya adalah file data base yang mengatur suatu alamat domain, yaitu untuk mentranslitkan alamat domain yang kita buat ke dalam suatu alamat ip., misalkan kita mempunya domain www.smknbtkl.sch.id, maka db.domain yang kita buat misalkan db.smknbtkl,

File db.ip maksudnya adalah file data base yang mengatur suatu alamat ip yang dijadikan server dns oleh kita, yaitu nantinya untuk pentranslitan dari alamat ip yang kita gunakan sebagai server dns ke dalam suatu nama domain, yaitu missal www.smknbtkl.sch.id, semissal kita mempunyai host yang mempunyai ip address 192.168.1.3 dan mau kita buat domainnya adalah www.smknbtkl.sch.id, makan db.ip yang kita buat misalkan db.192, (ini tidak terlalu diharuskan, kita dpt bebas menggunakan nama apa saja,, asalkan di named.conf kita harus menujukan ke db.ip yang kita buat,,)

File db.domain dan db.ip, seharusnya kita harus buat secara menual, yaitu dengan perintah touch atau yg lainnya, tapi supaya kita tidak susah kita copi pase aja dari file db.local dan db.127, yang masih berada di folder bind, (/etc/bind).

Semua file-file konfiugurasi untuk pengaturan dns server berada di /etc/bind/. Oke kita langung pada proses konfigurasi, semisal tadi kita akan buat buat domain dengan ip address 192.168.1.3 dan domainnya adalah www.smknbtkl.sch.id.

Makan langkah-langkah'nya adalah..

- edit file named.conf,

```
# nano /etc/bind/named.conf
```

- cari kata zone "localhost". . . . . .
  yaitu terdapat pada line/garis 20 (ctrl+w+t, ketik 20), enter,,
  maka akan tampil

```
zone "localhost" {
     type master;
     file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
     type maser;
     file "/etc/bind/db.127";
};
```

- setting dan edit sehingga menjadi seperti pada gambar berikut.

```
zone "smknbtkl.sch.id" {
     type master;
     file "/etc/bind/db.smknbtkl";
};
zone "3.1.168.192.in-addr.arpa" {
     type maser;
     file "/etc/bind/db.192";
};
```

- lalu simpan perubahan dan keluar, yaitu dengan menekan ctrl+x & Y..

Penjelasan2

» zone adalah statement yang menyatakan zone, zone yang pertama menunjukan nama domain yang akan kita buat, dan yang ke2'nya di isi dengan zone untuk menyatakan address dari domain tersebut.

» Type menunjukan tipe dari zone kita, ada 2 tipe yang terdapat, yaitu master dan slave, master yaitu file database langsung dari kita atau dari server yang kita baut, atau kita lah server sebenar'nya (tidak mengainduk pada server yang lain) dan slave adalah server dns yang file database'nya mengambildari server lain.

» File menjunjukan letak file database yang akan mengkonfigurasi pembuatan dns yang akan kita buat, dan file named.conf ini yang menunjukan dimana letak file yang mentranslitkan domain ke ip, dan dimana letak file yang mentranslitkan ip ke domain tersebut.

- Langkah selanjut'nya memBuat file db.domain dan db.ip

  Contoh kita buat dg db.smknbtkl dan db.192

  Sebenar'nya kita harus membuat ke2 file tersebut, tapi untuk memudahkan kita copi aja dari file db.local dan db.127, karena sama isi'nya, db.local dicopi ke db.smknbtkl dan db.127 copi ke db.192 Perintah'nya sebagai berikut.,

```
# cp db.local db.smknbtkl
```

```
# cp db.127 db.192
```

- Edit ke 2 file terebut., pertama edit db.smknbtkl

```
# nano db.smknbtkl
```

- Lihat sebelum dan sesudah konfiguruasi, dan sebagai contoh, ikuti setelah yang dikonfigurasi,,

```
; BIND data file for local loopback interfaces
;
$ TTL       604880
@     IN    SOA   localhost.root.localhost. (
                        2           ; serial
                    604800          ; refresh
                    86400           ; retry
                    2419200         ; expire
                    604800 )        ; Minimum  TTL
;
@     IN    MS    localhost.
@     IN    A     127.0.0.1
@     IN    AAA   ::1
```

Ubah menjadi

```
; BIND data file for zone smknbtkl.sch.id
;
$ TTL       604880
@     IN    SOA   server.smknbtkl.sch.id. root.smknbtkl.sch.id. (
                        2                   ; serial
                    604800                  ; refresh
                    86400                   ; retry
                    2419200                 ; expire
                    604800 )                ; Minimum  TTL
;
@           IN    NS    server.smknbtkl.sch.id.
@           IN    MX 10 mail.smknbtkl.sch.id.
@           IN    A     192.168.1.3
Server      IN    A     192.168.1.3
WWW         IN    A     192.168.1.3
Mail        IN    A     192.168.1.3
ftp         IN    A     192.168.1.3
```

❖ Edit file db.192

```
# nano db.192
```

❖ Lihat sebelum dan sesudah konfiguruasi

sebelum

```
; BIND reverse data file for local loopback interface
;
$ TTL      604880
@     IN    SOA   localhost.root.localhost. (
                        2          ; serial
                   604800          ; refresh
                   86400           ; retry
                   2419200         ; expire
                   604800 )        ; Minimum  TTL
;
@     IN    NS    localhost.
1.0.0 IN    PTR   localhost
```

ubah menjdi

```
; BIND reverse data for file zone smknbtkl.sch.id
;
$ TTL      604880
@     IN    SOA   server.smknbtkl.sch.id. admi@smknbtkl.sch.id. (
                        2                    ; serial
                   604800                    ; refresh
                   86400                     ; retry
                   2419200                   ; expire
                   604800 )                  ; Minimum  TTL
;
@     IN    NS    server.smknbtkl.sch.id.
@     IN    MX 10 mail.smknbtkl.sch.id.
3.1.168.192      IN    PTR   smknbtkl.sch.id.
3.1.168.192      IN    PTR   server.smknbtkl.sch.id.
3.1.168.192      IN    PTR   www.smknbtkl.sch.id.
3.1.168.192      IN    PTR   mail.smknbtkl.sch.id.
3.1.168.192      IN    PTR   ftp.smknbtkl.sch.id.
```

❖ Simpan perubahan,,

Ctrl+x+y

Penjelasan2

- IN adalah standar untuk internet,
- SOA (star of authority) mengidentifikasi authority untuk data zone
- Tipe record
  - . NS (Name Server) : menunjukan host DNS server
  - . A (Address) : yang memetakan domain ke alamat ip
  - . MX (Mail eXchanger) : menunjukan Host yang berfungsi sebagai email, merupkan data dengan tipe special untuk spesi Pikasi layanan email pada sutu domain, dlm mel

Indungi proses lalu lintas data email, data MX mempunyai nilai prioritas dalam bilangan numeric.

. PTR (Pointer) : yang memetakan dari alamat ip ke domain

- Setelah selesai semuanya, restar aplikasi bind9'nya..

```
# /etc/init.d/bind9 restart
```

- Untuk mengecek, gunakan perintah,.

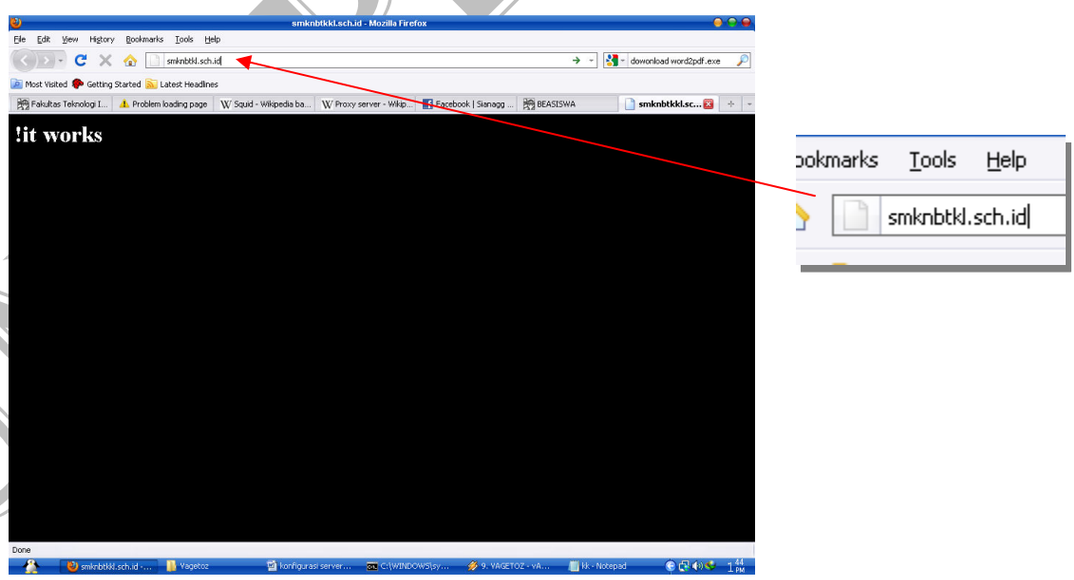
```
# nslookup smknbtkl.sch.id
```

- Cek semua domain yang sudah kita konfigurasi, seperti smknbtkl.sch.id, mail.smknbtkl.sch.id dsb, ditakut'kan ada kesalah dalam konfigurasi, Kalau tidak ada pesan eror,, berarti dns sudah berhasil dibuat,, tinggal cek di clien dengan web browser ataupun dengan cmd, yaitu dengan ping domain, kalau di web browser langsung ketaikan aja di url domain kita, tapi untuk mengecek di dg web browser di sever'nya harus sudah terinsall web server, contoh'nya apche2,, untuk itu kita nstall dulu web server'nya contoh kita pake apache2, langkah-langkah'nya adalah sbb,.:

```
# apt-get install apache2
```

Untuk mengedit tampilan dari halaman web, secara default file html'nya terdapat di /var/www/index.html, dan isi'nya hanya Its works !, kita dapat mengubah'nya sesuai dengan kebutuhan,,

- Kalau sudah, semua'nya restart apache2 dan bind9'nya,, dan coba cek di clien dengan domain kita., seperti pada gambar berikut.



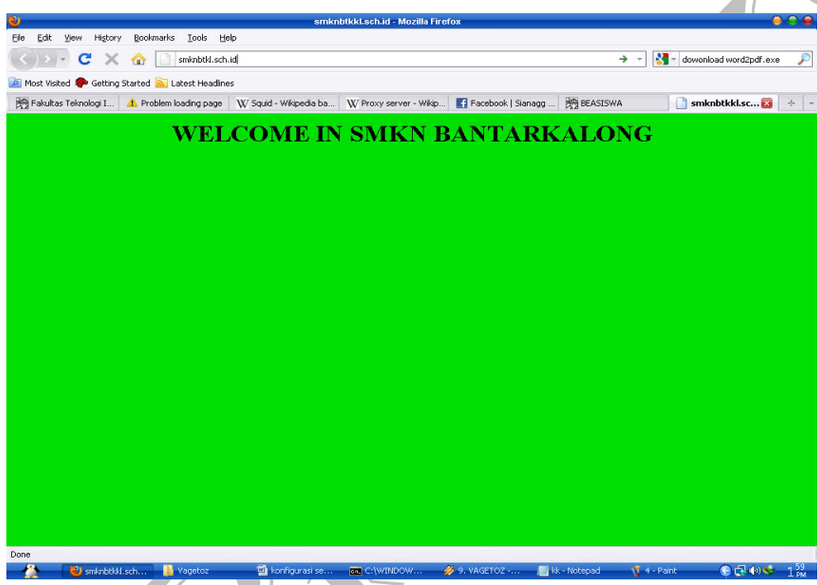
Kalau kita tidak melakukan perubahan untuk tampilan halaman web'nya maka tampilan'nya akan seperti itu,, tapi kalau kita ingin melakukan perubahan pada tampilah halamannya,, maka file yang harus di konfigurasi'nya adalah index.html, yang ada di /var/www/,, begini perintah'nya..

```
# nano /var/www/index.html
```

Missal kita ubah menjadi seperti ini,,

```
<html>
<head>
<title>smknbtkl.sch.id</title>
</head>
<body bgcolor=”gren”>
<h1  align=”center”> WELCOME IN SMKN BANTARKALONG </h1>
</body>
</html>
```

- Pasti tampilannya akan seperti gabar dibawah ini,,



- Ok kalau tampil seperti itu, berarti kita sudah berhasil..
- Kita Bisa Juga Ngecek dengan Perintah. w3m nama Domain Yang kita Buat.

```
# w3m smknbtkl.sch.id
```

- **SELESAI**

Konfigurasi DHCP SERVER

1. Teori

DHCP (Dinamic Host Configuration Protocol) merupakan protocol yang berbasis arsitektur clien/server yang dipakai untuk memudahkan pengalokasian alamat ip dalam satu jaringan. Jadi semua clien yang terhubung ke server tidak usah mengisi alamat ip secara manual, tapi sudah diberi oleh server.,

Pada distro debian program yang digunakan untuk membangun sebuah server DHCP adalah dhcp3-server, berikut langkah2 instalasi beserta konfigurasinya..,

2. Instalasi

Installasi dhcp3-server

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

Tunggu hingga selesai,,,,,,

3. Konfigurasi

File konfigurasi untuk dhcp3-server terdapat di /etc/dhcp3/dhcpd.conf dan /etc/default/dhcp3-server. File ini berisi konfigurasi interface yang digunakan untuk dhcp3-server.

- Edit file dhcpd.conf

```
# nano /etc/dhcp3/dhcpd.conf
```

- Cari bari # A slightly different…… dan tampak'nya seperti berikut..

```
# A slightly different configuration for an internal subnet
# subnet 10.5.5.0 netmask 255.255.255.224 {
# range 10.5.5.26 10.5.5.30;
# option domain-name-servers ns1internal.examples.org;
# option domain-name "interna.example.org";
# option routers 10.5.5.1;
# option broadcast-address 10.5.5.31;
# default-lease-time 600;
# max-lease-time 7200;
#}
```

- Edit hingga serperti pada gambar berikut.

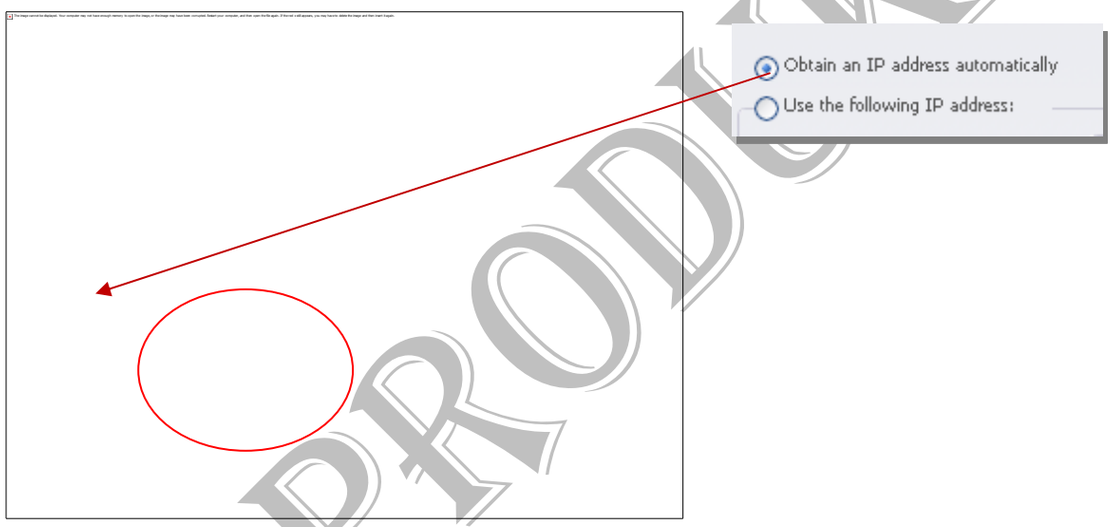
```
# A slightly different configuration for an internal subnet
  subnet 192.168.1.3 netmask 255.255.255.0 {
  range 192.168.1.1 192.168.1.30;
  option domain-name-servers 192.168.1.3;
  option domain-name "smknbtkl.sch.id";
  option routers 192.168.1.4;
  option broadcast-address 192.168.1.255;
  default-lease-time 600;
  max-lease-time 7200;
 }
```

- Lalu save dgn ctr+x+y, dan restart servis'nya (dhcp3-server)

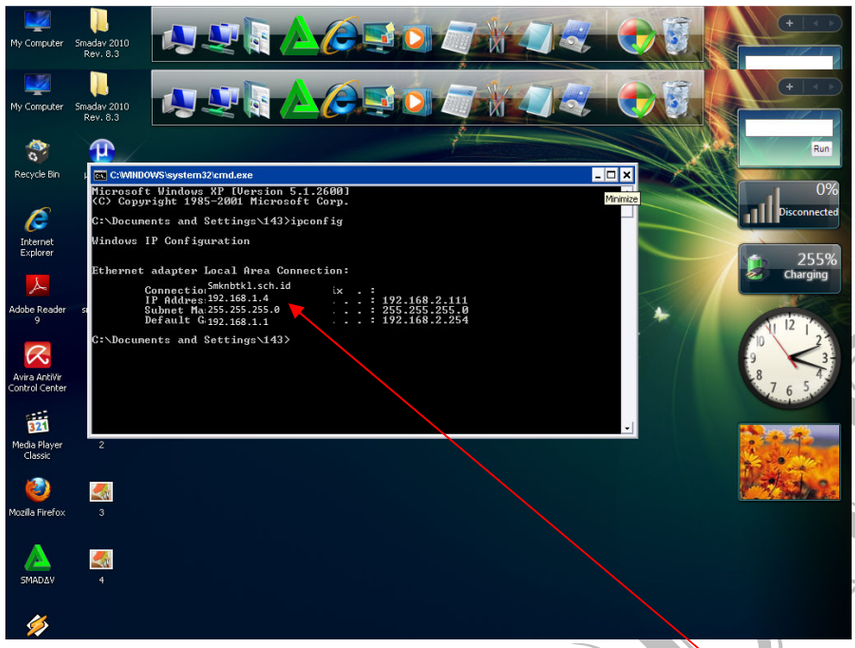
```
# /etc/init.d/dhcp3-server restart
```

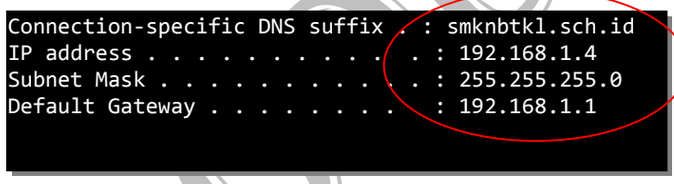
- Jika tidak ada pesan failid, berarti instalasi lancar, biasanya pas pertama di restart, akan muncul pesan failid (tulisan berwarna merah), tapi ke2'nya tidak,, dan itu menandakan dhcp telah bagus, dan tidak ada masalah,, tinggal di-stop, star dan di-restart kembali untuk memastikan., tapi kalau yang pertama ada pesan failed dan ke2 kali'nya lagi ada pesan berwarna merah lagi, berarti ada yang eror, atau ada yang salah pada konfigurasi,, cek lagi, takut ada yang salah, biasanya ada yang salah pada subneting, atau penghitungan IP.

- Langkah terakhir cek di clien, masuk ke cmd, jangan lupa pengaturan tcp'nya di atutomais dhcp kan.., seprti pada gambar di bawah.



- cek di cmd dengan perintah ipconfig, apakah memberi ip address dan domain atau tidak, kalau memberi berarti dhcp tela berjalan baik.





- **SELEASAI**

Konfigurasi ftp SERVER

1. Pengertian

FTP menggunakan protokol transport TCP untuk mengirimkan data/*file*. TCP dipakai sebagai protokol *transport* karena protokol ini memberikan garansi pengiriman dengan FTP yang dapat memungkinkan user mengakses *file* dan direktori secara interaktif, diantaranya :

- Melihat daftar file pada direktori *remote* dan lokal.
- Menganti nama dan menghapus file.
- Transfer file dari komputer *remote* ke lokal (*download*).
- Transfer file dari komputer lokal ke *remote (upload).*

Contoh aplikasi FTP server :

- Proftpd
- Vsftpd
- Wuftpd
- IIS (didalamnya terdapat FTP Server)

Contoh aplikasi FTP client

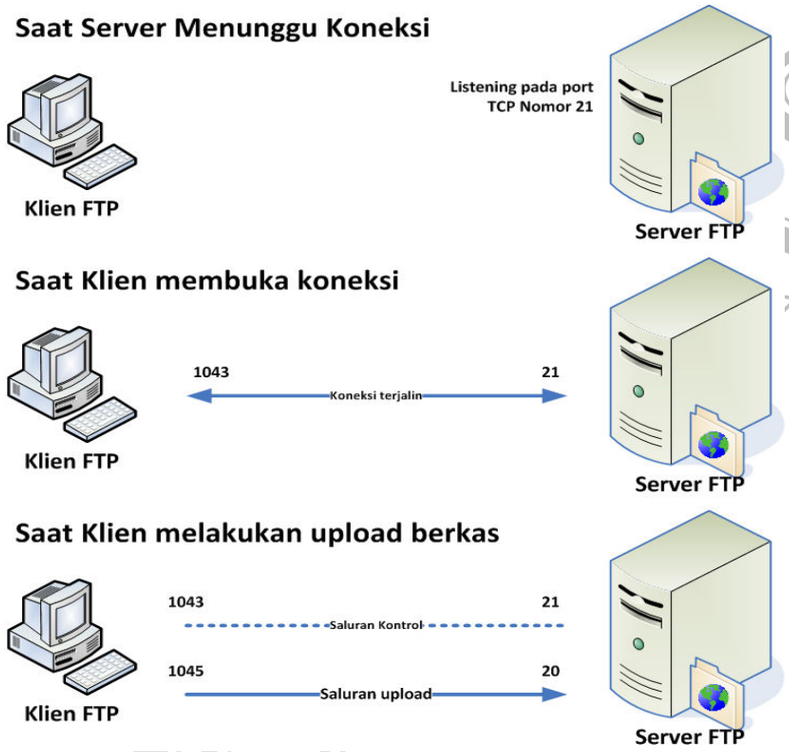
- CuteFTP, Wget
- WsFTP
- GetRight
- AbsoluteFTP
- SmartFTP
- Filezilla(Mendukung SFTP)

FTP (singkatan dari *File Transfer Protocol*) adalah sebuah protokol Internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*file*) komputer antar mesin-mesin dalam sebuah *internetwork*.

FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan (*download*) dan penggugahan (upload) berkas-berkas komputer antara klien FTP dan server FTP. Sebuah Klien FTP merupakan aplikasi yang dapat mengeluarkan perintah-perintah FTP ke sebuah server FTP, sementara server FTP adalah sebuah Windows Service atau

daemon yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah klien FTP. Perintah-perintah FTP dapat digunakan untuk mengubah direktori, mengubah modus transfer antara biner dan ASCII, menggugah berkas komputer ke server FTP, serta mengunduh berkas dari server FTP.

Sebuah server FTP diakses dengan menggunakan *Universal Resource Identifier* (URI) dengan menggunakan format ftp://namaserver. Klien FTP dapat menghubungi server FTP dengan membuka URI tersebut.



FTP menggunakan protokol *Transmission Control Protocol* (TCP) untuk komunikasi data antara klien dan server, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum transfer data dimulai. Sebelum membuat koneksi, port TCP nomor 21 di sisi server akan "mendengarkan" percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai port pengatur (*control port*) untuk (1) membuat sebuah koneksi antara klien dan server, (2) untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada server dan juga (3) mengembalikan respons *server* ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka server akan mulai membuka port TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mentransfer data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan *password*-nya untuk mengakses, men-*download*, dan meng-*upload* berkas-berkas yang ia kehendaki. Umumnya, para

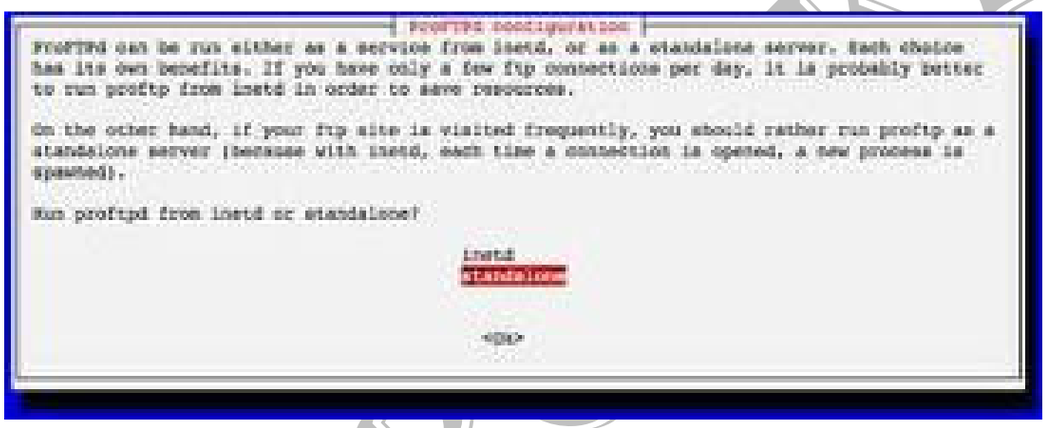
pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode *anonymous login*, yakni dengan menggunakan nama pengguna anonymous dan *password* yang diisi dengan menggunakan alamat *e-mail*..

2. Instalasi

Untuk membangun sebuah server ftp, disini saya menggunakan proftpd, langkah instalasi'nya yaitu sebagagi berikut..

```
# apt-get install proftpd
```

Akan ada tampilan sebagai berikut, dan pilih'lah standalone,,,



Standalone adalah tipe atu status dari proftpd yang akan kita jalankan atau bangun, karenan disini ada dua tipe atau mode istilah'nya, yaitu *inetd* dan *standalond, inetd* adalah ketika suatu server ftp hanya diakses sesekali, pengguna disarankan memilih modi ini, dengan alasan untuk menghemat sumber daya, dan alasan memilih standalone adalah untuk sever ftp yang akan di akses secara masal,

Tunggu proses instalasi selesai,,

3. Konfigurasi

Untuk terlebih dahulu, kita haru tau dulu, apa tujuan kita, dan kita harus tau dulu, apa yang dimaksud dengan anonymous & user,, kita harus tau mau apa tujuan kita apakah mau user atau mau anonymous,,

Pada saat pertama kita ngistal dan belum di konfigurasi apapun, maka status proftpd'nya adalah user, artinya setiap pengguna yang ingin mengakses proftd harus terlebih dahulu mengisikan username dan password,, sedangkan anonymous tidak,, hak aksesnya dibebaskan,, nah pada setelah instalasi selesai dan kita restar proftd'nya maka status'nya adalah user,, tapi user'nya tidak di batasi, maksudnya kalau kita sudah masuk dalam satu user maka kita bias melihat user yang lainnya,, untuk status user yang akan kita batasi, misalkan ada 10 user yang ada di server, misalkan kita mau memberikan

akses ftp kepada 5 user, nah user yang sudah kita daftarkan akan bisa masuk untuk mengakses ftp, tapi yang lainnya tidak,, dan untuk tidak dapat membuka user yang lainnya, bisa dilihat pada konfigurasi nanti,,

❖ **Proftpd dengan Anonymous**

Anonymous berarti bebas, jadi sever ftp memberi hak akses bebas kepada semua pengguna atau user untuk dapat mengakses file-file dari server,, berikut konfiugurasinya,,

❖ Edit file proftpd.conf, yang berada di /etc/proftpd/proftpd.conf/

```
# nano /etc/proftd/proftpd.conf
```

Edit atau hilangkan tanda pagar pada baris-bari sbb..

```
ServerName                  "ServerMu"
serverType                  standalone
DeferWelcome          off
DefaultRoot           ~
<Anonymous ~ftp>
User                  ftp
Group           nogroup
UserAlias       anonymous ftp
dirFakeUser on ftp
DirFakeGroup on ftp
RequireValidShell     off
maxClents                   10
DisplayLogin          welcome.msg
DisplayFirstChdir     .message
</Anonymous>
```

Restart dengan Perintah ***#/etc/init.d/proftpd restart***

Keterangan

~ftp : berarti folder yang dipakai untuk anonymous ftp server, yaitu adalah di /home direcroty dari user ftp yaitu /home/ftp/. Sehingga bisa ditulis dengan /home/ftp. Berikut contonya..

<Anonymous/home/ftp>

- Simpan konfigurasi yang telah kita konfigurasi..
- Cek di cliet,, apakah minta username dan password atau tidak, dan benar tidak poder yang akan kita share'kan polder itu,, missal /home/ftp. Kalau benar semuanya, berarti ftp dengan anonymous sudah berhasil dibuat..

- **Proftpd dengan user**

  Disini berarti setiap pengguna yang ingin mengakses ftp diharuskan untuk mengisikan username dan password ntuk dapa masuk.

  Untuk ftp dengan akses user, kita tingaal tambahkan di baris paling bawah, atau baris terakhir secrip berikut,

```
<Limit LOGIN>
AllowUser          nama user1
AllowUser          nama user2
Deny All
</limit>
```

```
<Limit LOGIN>
AllowUser          admin1
AllowUser          admin2
Deny All
</limit>
```

  Berarti user yang dapat memasuki atau mengakses file dari server ftp hanya'lah enas dan cuy, dan kalau mau menambahkan lagi user yang akan mengakses file ftp, maka tinggal tambahkan lagi bari

  AllowUser nam usre3, jadi mesikipun banyak user di server 20, tapi yang kita masukan hanyalah 5, maka tetap yang akan masuk atau yang dapat mengakses file ftp hanyalah 5 user itu.,

  Setelah selesai, simpan dan cek di clien…

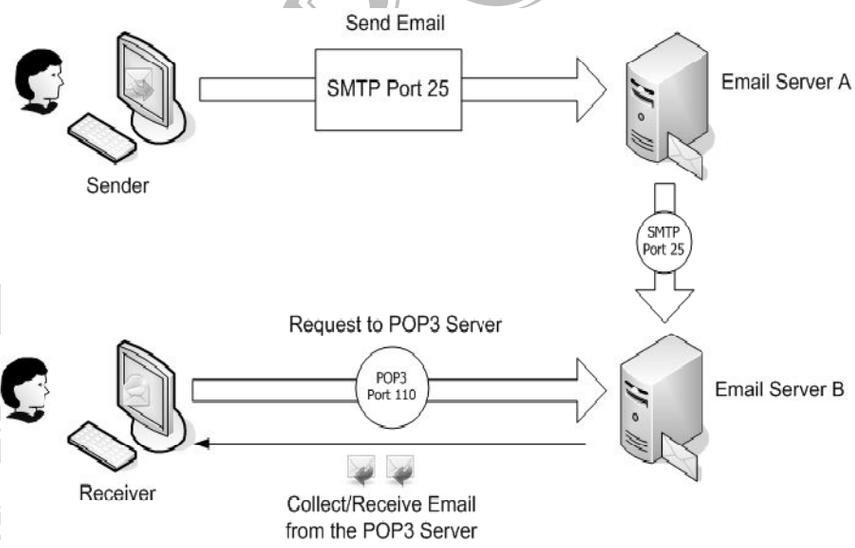
Konfigurasi Mail SERVER

1. Teori

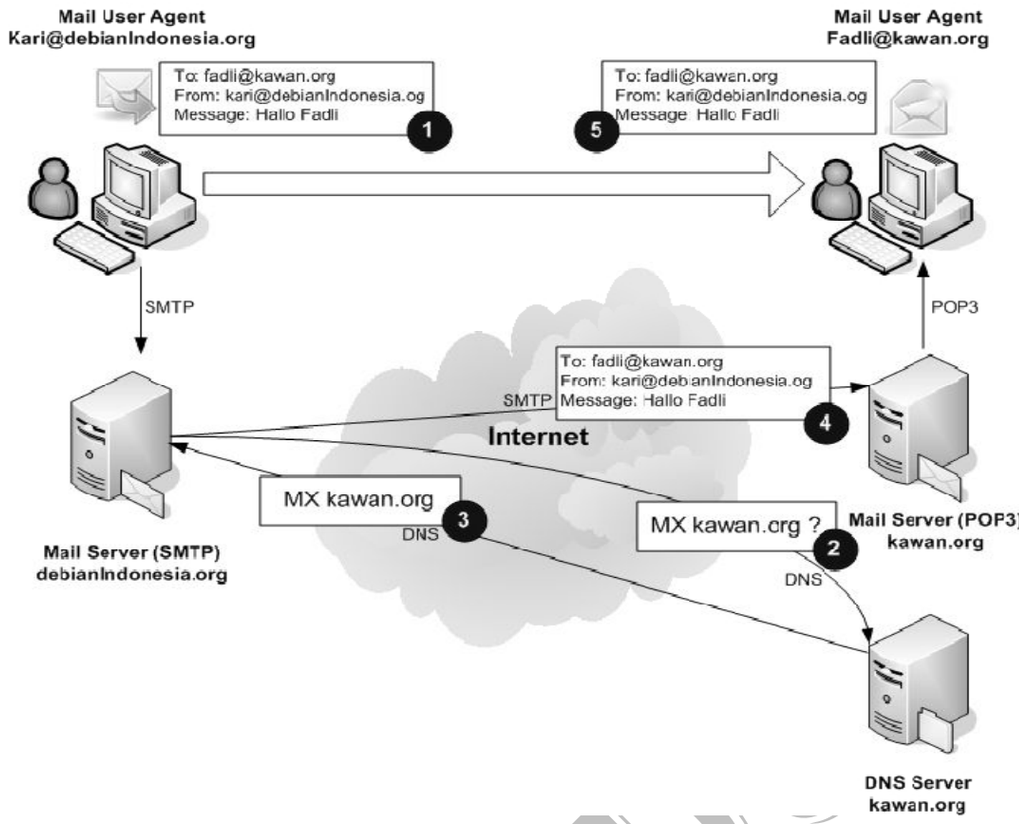
E-mail merupakan aplikasi internet yang banyak digunakan saat ini untuk komunikasi dua arah. Selain karena kemudahan dalam penggunaannya juga karena kemurahan teknologi yang ditawarankan oleh email ini. E-mail singkatan dari electronic mail juga dapat dianalogikan dengan pengiriman surat yang lazim digunakan saat ini melalui kantor pos, atau melalui jasa pengirim surat atau barang. Pengiriman e-mail dilakukan melalui perangkat elektronik seperti komputer atau HP/PDA.

Proses pengiriman/penerimaan e-mail melibatkan protokol Simple Mail Transfer Protocol (SMTP) dan Post Office Protocol version 3 (POP3). Protokol SMTP bertugas untuk proses pengiriman mail (outgoing mail) dan POP3 bertugas untuk proses penerimaan email (Incoming mail).

Jika User atau pemilik e-mail tidak sedang aktif/offline untuk pengaksesan e-mail, maka e-mail yang tertuju kepadanya akan ditampung sementara oleh server e-mail sampai pemilik email tersebut mengaksesnya. Hal ini bisa terjadi karena adanya protokol POP3.



Proses pengiriman e-mail secara detail dapat dilihat pada gambar berikut yang melibatkan beberapa komponen server seperti DNS server, mail server meliputi SMTP server, Mail Transfer Agent (MTA), dan POP3 server.



Mari kita asumsikan bahwa penulis ingin mengirim sebuah e-mail ke rekan yang berada di perusahaan lain (fadli@kawan.org). Bagaimana e-mail yang penulis kirim dapat sampai ke rekan yang berada di perusahaan lain tersebut?. Berikut deskripsi proses transfer/receive e-mail yang melibatkan protokol-protokol di atas.

- Suatu client yang akan melakukan koneksi ke SMTP server di mail.debianIndonesia.org menggunakan port 25.

  Suatu Clien melakukan percakapan dengan SMTP server tentang alamat email dari pengirim/sender, alamat e-mail tujuan serta isi dari e-mail tersebut

  SMTP server akan mengambil alamat e-mail 'To' tujuan dan memecah menjadi:

  - Nama pemilik - fadli
  - Nama domain – kawan.org

  Jika user tujuan merupakan user lain yang masih berada di domain yang sama (debianIndonesia.org), maka SMTP server akan memberikan email tersebut ke POP3 server di debianIndonesia.org. Untuk kasus di atas, e-mail tujuan tidak berada di domain yang sama, maka SMTP server akan berkomunikasi terlebih dahulu dengan domain tujuan.

- SMTP server akan berkomunikasi dengan Domain tujuan dan meminta IP address dari domain tersebut yakni kawan.org
- Domain tujuan akan mereply dengan sebuah alamat SMTP server tujuan

- SMTP server debianIndonesia.org melakukan koneksi ke SMTP server kawan.org menggunakan port 25.
- Selanjutnya e-mail tersebut akan diserahkan ke POP3 server menggunakan port 110 yang terdapat pada domain tersebut. Selama user Fadli berstatus offline, maka e-mail yang tertuju kepadanya akan tetap tersimpan di POP3 server sampai user Fadli mengakses MUA.

Untuk membangun mail server dan webmail dapat menggunakan beberapa koponen sebagai berikut.

- ❖ MTA : postfix, Qmail dan Sendmail.
- ❖ POP3/IMAP server: dovecot, couirier, UW-IMAP
- ❖ Webmail : squirrelmail

2. Installasi

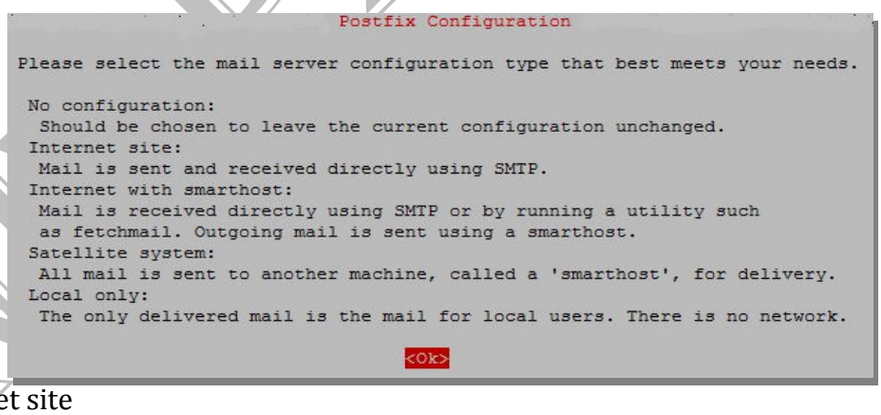
Untuk membangun sebuah mail server dan web mail server, aplikasi-aplikasi yang di install diantara'nya.., (masih dalam DVD binary 1)

- Postfix (MTA)
- Dovecot-imapd courier-imap & courier-pop (imap/pop3)
- Squirrelmail (webmail)

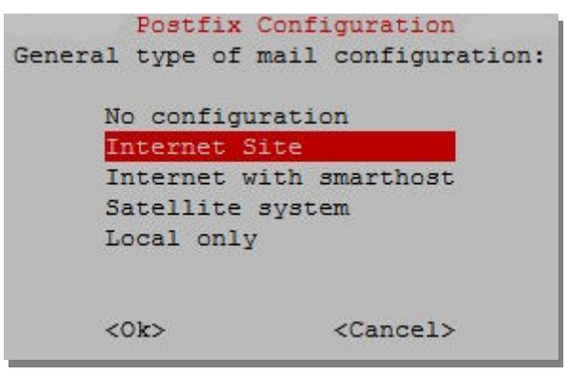
❖ Intalasi postfix

```
# apt-get install postfix
```

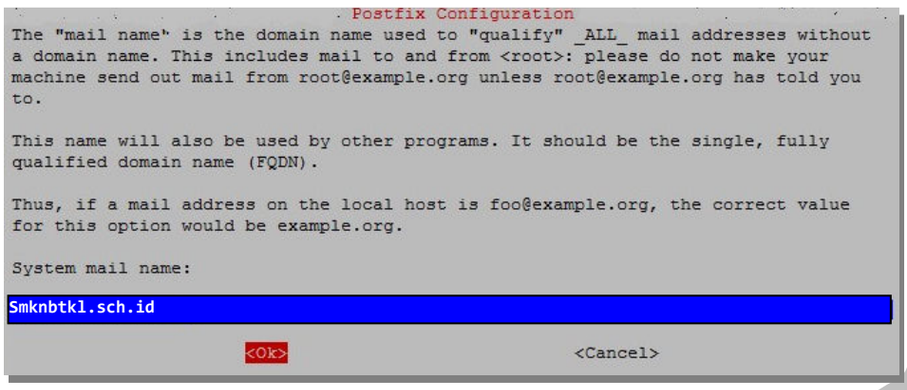
Pilih ok



Pilih internet site



Isi dengan domain



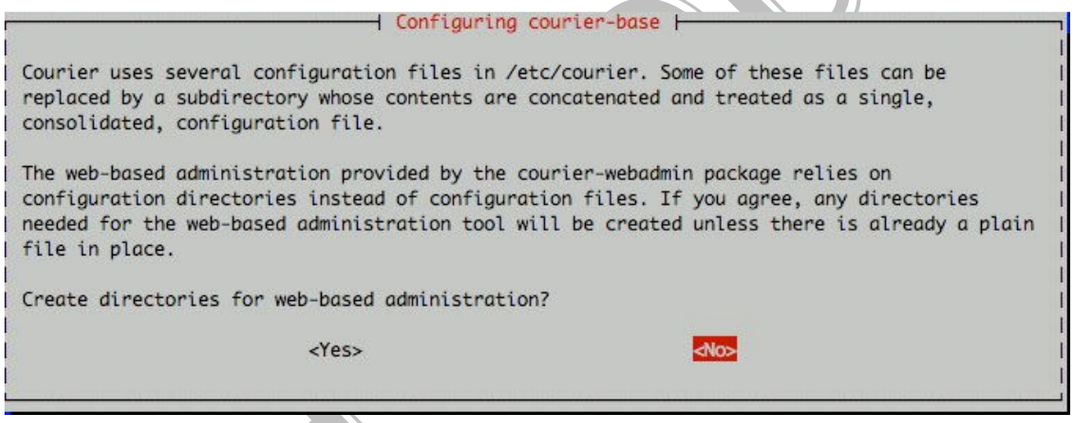
- Installasi dovecot-imapd

```
# apt-get install dovecot-imapd
```

- Installasi courier-imap & courier-pop

```
# apt-get install courier-imap courier-pop
```

web-based administration? Pilih NO



- Install squirrelmail

```
# apt-get install squirrelmail
```

Instalasi cukup hanya dengan sampai sini, restart semua yang sudah di install tadi, beserta web server'nya (apache2), dan coba cek di url clien, dengan domain, http://smknbtkl.sch.id/squirrelmail, sehingga tampil halaman seperti gambar dibawah...

3. Konfigurasi

❖ Konfigurasi webmail (squirrelmail)

Di sini kita akan mengkonfigurasi squirrelmail, artinya kita akan melakukan perubahan atau pengkonfigurasian pada squirrelmail, di sini kita mengkonfigurasi squirrelmail supaya kita pas di serch di url www.mail.smknbtkl.sch.id yang muncul'nya adalah sebuah web mail, yaitu kita disini kita menggunakan squirrelmail, jadi kalau kita mau masuk ke login mail, seperti pada gambar diatas kia tidak perlu mengtik, *smknbtkl.sch.id/squirrelmail* kita hanya mengetikan mail.smknbtkl.sch.id. karena meskipun di DNS kita mengkonfigurasikan Domain untuk mail, yaitu *mail.smknbtkl.sch.id*, maka tetap saja yang muncul akan isi dari domain induk, yaitu smknbtkl.sch.id. bukan tampilan dari webmail.. maka untuk itu kalau kita ingin tampilannya langsung pada login mail atau tampilan dari web mail, kita perlu mengkonfigurasinya dari webserver'nya (apache2),, dan berikut langkah-langkah'nya.,,

- Edit file apache.conf yang ada di apache2 (/etc/apache2/apache2.conf/),

```
# nano /etc/apache2/apache2.conf
```

Tambahakan pada baris yang paling bawah, scrip berikut,. tujuannya adalah untuk menambahkan file pengkonfigurasian untuk squirrelmail pada webserver (apache.conf), atau memvirtualhost kan untuk squirrelmail,.

```
# Include generic snippets of statements
Include /etc/apache2/conf.d/

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/
Include /etc/squirrelmail/apache.conf
```

- Konfigurai atau edit apache.conf yang ada di squirrelmail (/etc/squirrelmail/apache.conf).

```
# nano /etc/squirrelmail/apache.conf
```

- Tambahkan scrip beriktu untuk membuat virtualhost, bisa di edit atau bias juga di tambahkan dibawah'nya... cari line usr will. . . . atau perhatikan gambar berikut.,

```
# users will prefer a simple URL like http://webmail.example.com
#<VirtualHost 1.2.3.4>
#  DocumentRoot /usr/share/squirrelmail
#  ServerName webmail.example.com
#</VirtualHost>
```

Tambahkan/ buat scrip yang ada di lingkaran,,, tepat dibawah'nya </*VirtualHost*>

```
# users will prefer a simple URL like http://webmail.example.com
#<VirtualHost 1.2.3.4>
#  DocumentRoot /usr/share/squirrelmail
#  ServerName webmail.example.com
#</VirtualHost>

<VirtualHost *:80>
  DocumentRoot /usr/share/squirrelmail
  ServerName mail.smknbtkl.sch.id
</VirtualHost>
```

Simpan perubahan,,, dan restart apache2'nya,,,

```
# /etc/init.d/apache2 restart
```

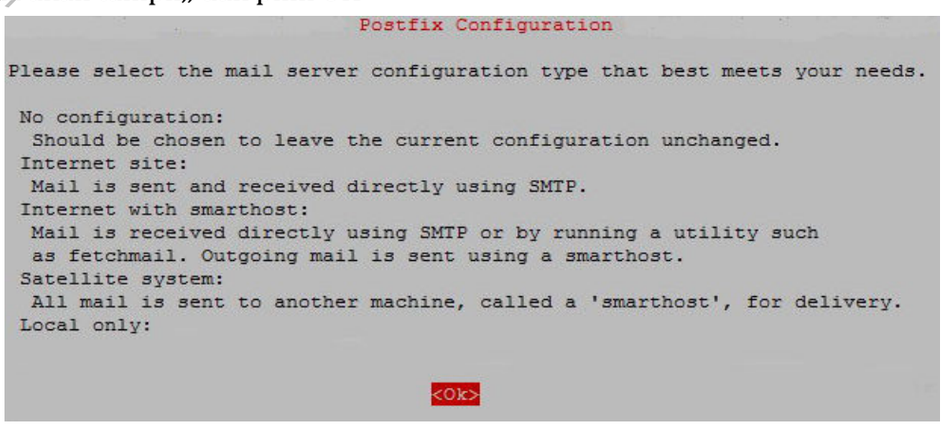
Kemudian pada computer Client Coba buka browser dan ketikkan "mail.smknbtkl.sch.id" Lalu anda akan otomatis di-redirect ke alamat "mail.smknbtkl.sch.id/scr/login.php" dan tidak akan masuk ke tampilan web domain smknbtkl.sch.id.. dan sama juga hasil redirect'nya kalo kita mengetikan "smknbtkl.sch.id/squirrelmail., dan maka hasi'nya akan seperti ini

❖ Konfigurasi mail server (postfix)

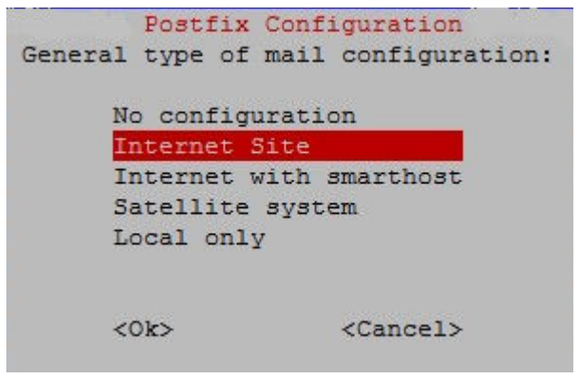
Selanjut'nya kita mengkonfigurasi ulang postfix yang sudah kita insall tadi,, untuk postfix kita bisa konfigurasi di /etc/postfix/main.cf, dan bisa juga kita edit langsung dengan perintah,, yaitu sebagi berikut,,.:

```
# dpkg-reconfigure postfix
```

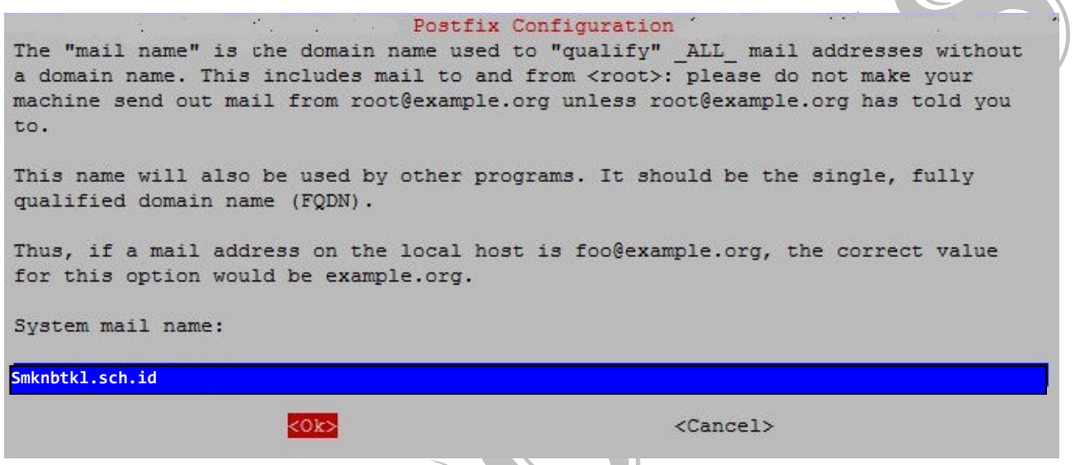
Maka akan tampil,, dan pilih OK



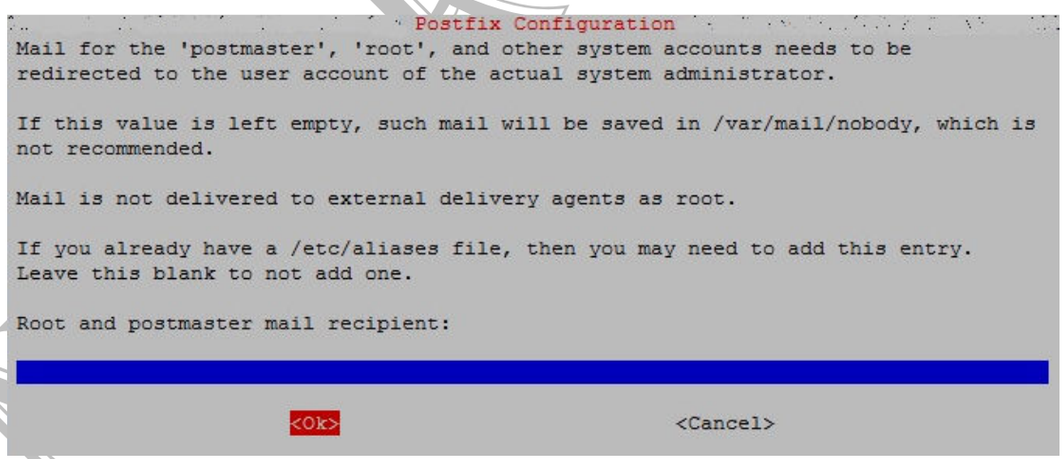
pilih internet site



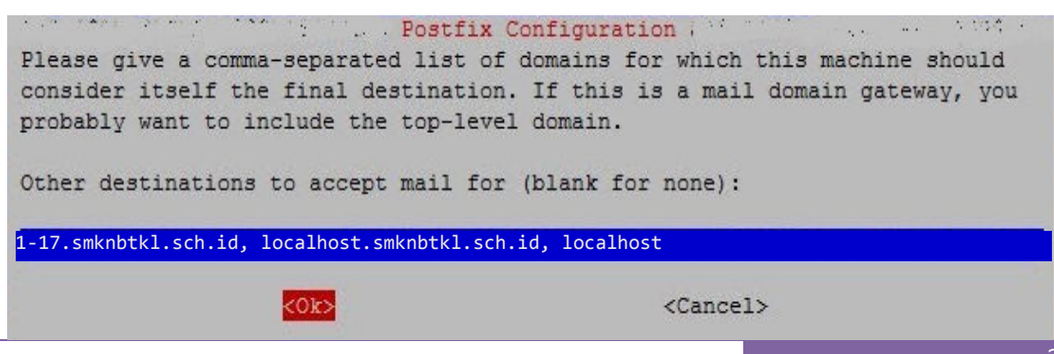
Isi dengan nama domain



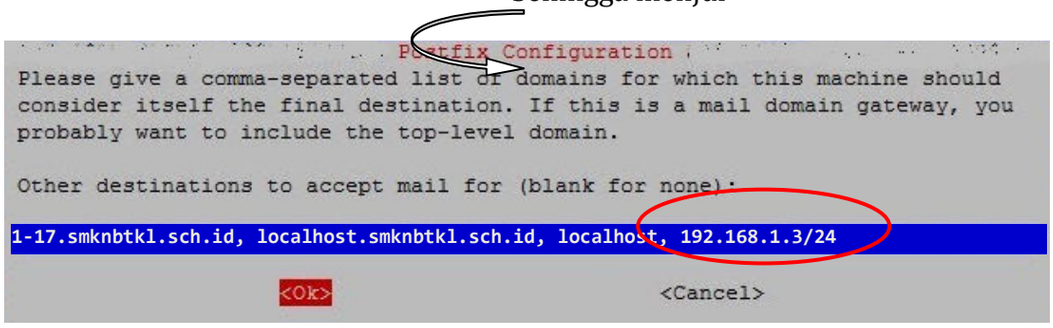
Pilih OK



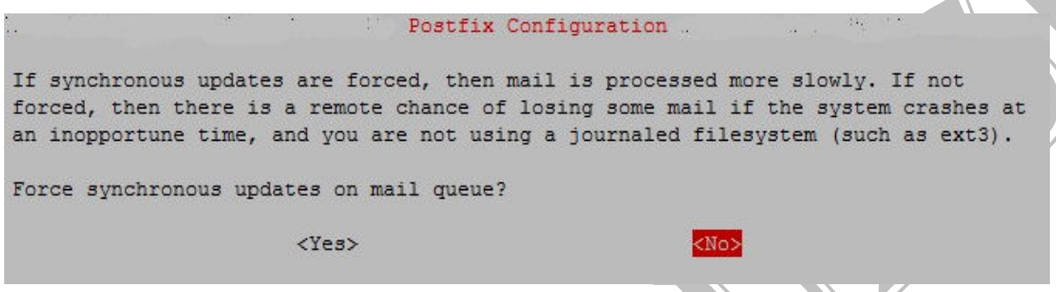
Pada bagian ini, isi'kan alamat ip server yang akan dijadikan dns mail server beserta prepigh'nya... contoh 192.168.1.3/24,, caranya tinggal tambahkan setelah kata localhost, dan setelah localhost beri tanda koma, perhatikan gambar berikut,,



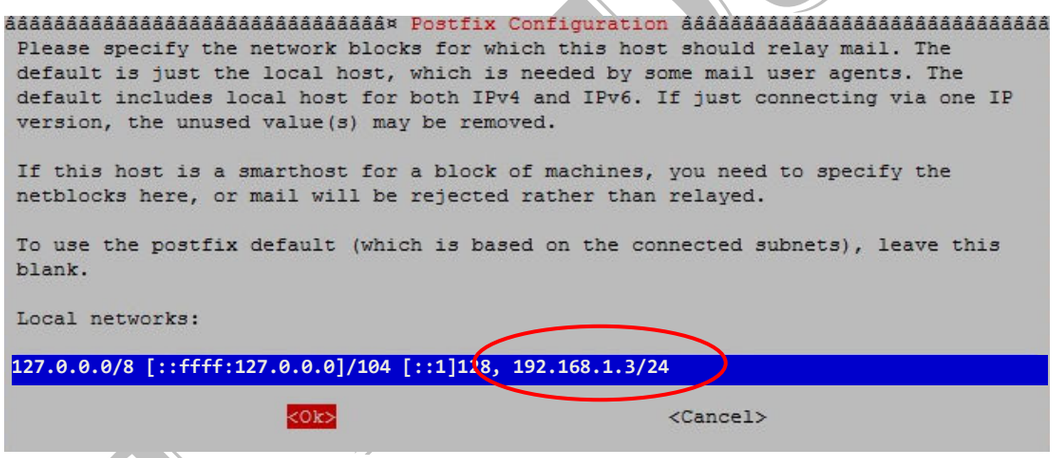
Sehingga menjdi



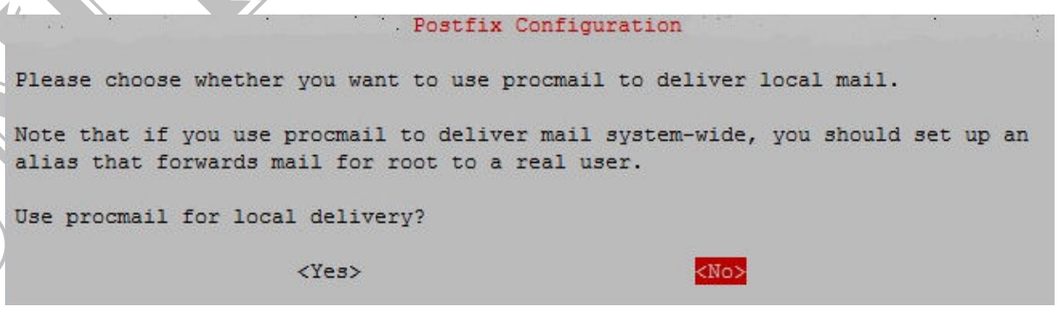
Pilih No



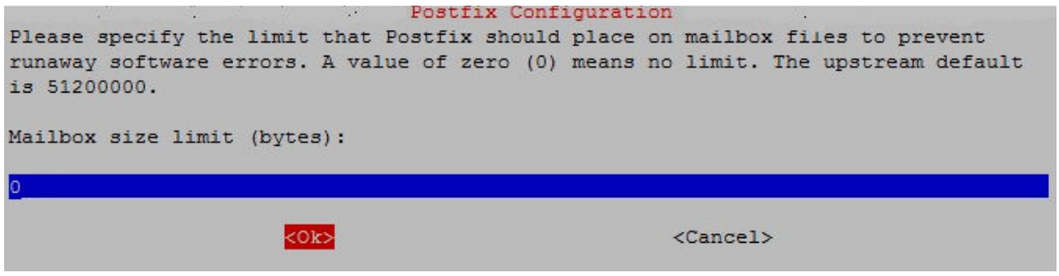
Tambahkan 192.168.1.3/24 pada baris paling belakang, sesudah 128 pake koma,



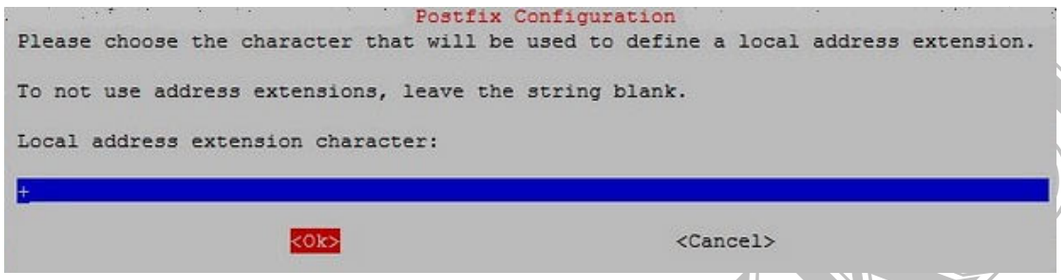
Pilih NO



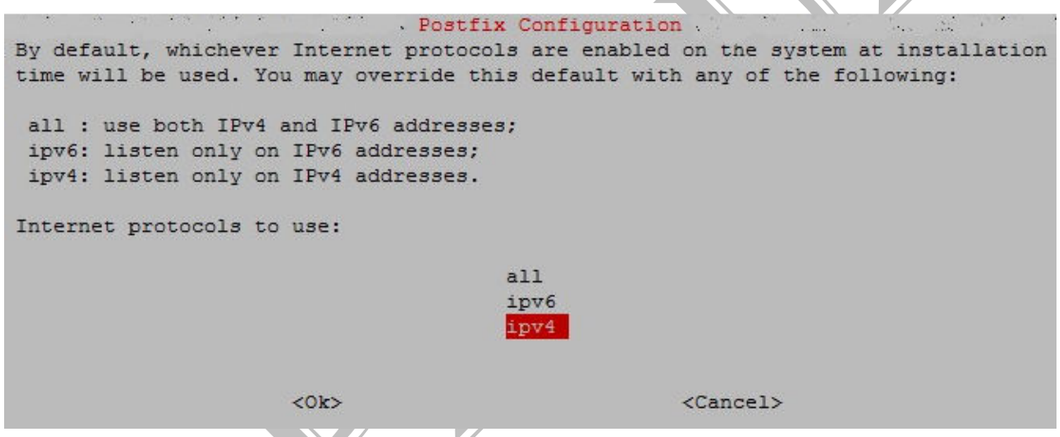
Pilih Ok



Pilih OK



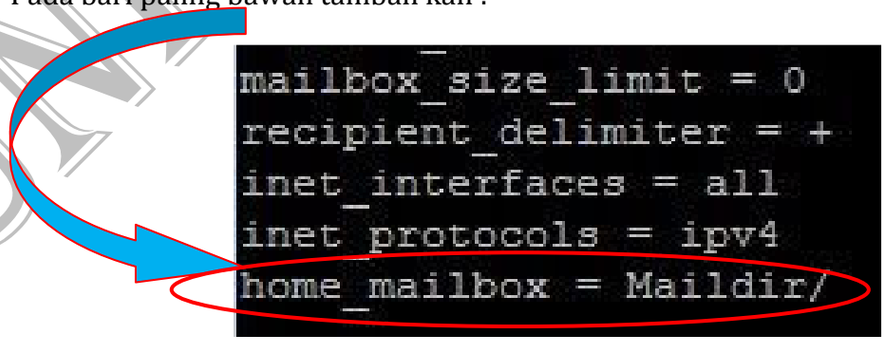
Pilih ipv4, lalu Ok



Proses konfigurasi ulang postfix telah selesai,, selanjut'nya kita harus menambahkan pd main.cf *home_mailbox = Malidir,* beriktu perintah'nya…

```
# nano /etc/postfix/main.cf
```

Pada bari paling bawah tambah'kan :



Setelah selesai simpan perubahan,, dan restar semua paket yang telah di install..

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

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

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

Sekarang kita buat folder mail otomatis, tujuan'nya adalah untuk memberikan folder mail pada setiap user,, agar setiap user mempunyai folder mail,, yaitu caranya adalah sbb:

```
# maildirmake /etc/skel/Maildir
```

Buat beberapa user, untuk mengecek hasi dari pembuatan mail kita,, yaitu untuk mengecek pengiriman email atau mengecek login username,, misalkan saya menbambahkan user enas dan cinta,, berikut perintah'nya..

```
# adduser admin
```

Masukan sebuah password,, dan ikuti perintah permintaan'nya… hingga selesai..

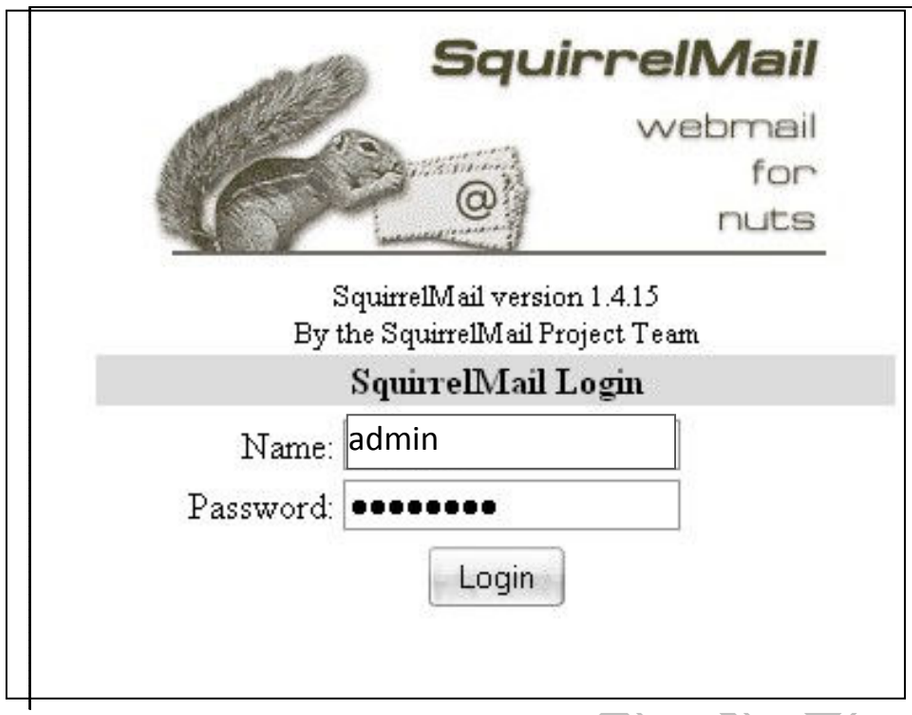
Proses pembuatan mail server dan web mail sudah selesai kita buat dan konfigurasi,, tinggal kita cek di clien…

Sekarang kita akan mengecek di clien,, gimana hasil'nya,,, dan coba kirim e'mail dari user satu ke user yang lainnya,, apa'kah ke'kirim atau tidak,, kalau tidak ada pesan eror,, berarti email sudah berjalan dengan baik,tapi kalau pas waktu pengiriman ada pesan eror berarti ada yang eror pada IMAP,POP3,atau MTA,, dan bisa juga, pesan terkirim tapi pas dilihat di inbok tujuan,, pesan tersebut tidak ada,, nah itu juga termasuk kegagalan atau ada yang eror pada system paket mail,,

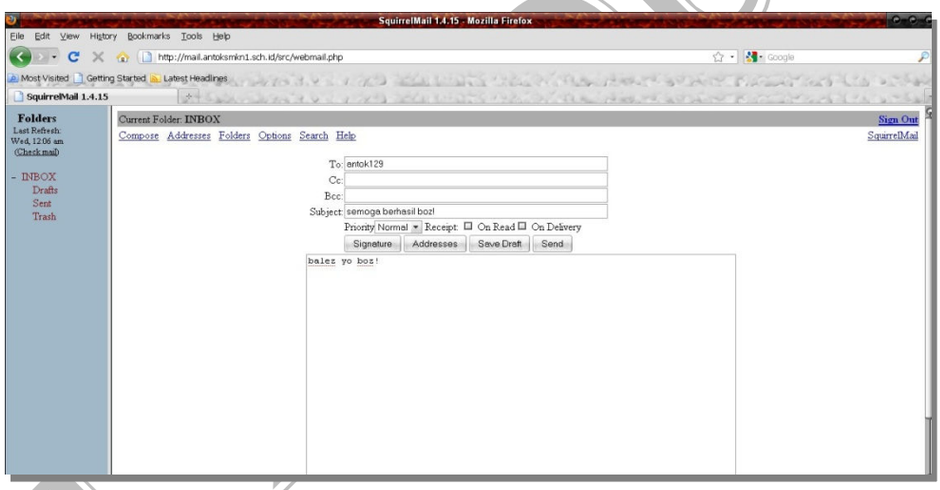
Sekarang cek di clien untuk membuktikan hasil konfigurasi tadi..

Buka di browser, ketikan mail.smknbtkl.sch.id, dan setelah muncul masuk dengan salah satu user,,,

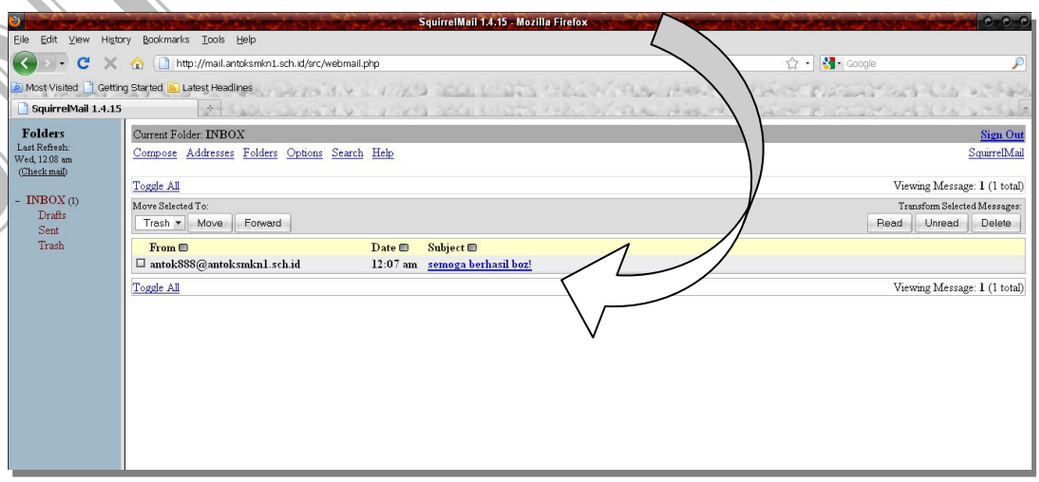
Masuk dengan salah satu user



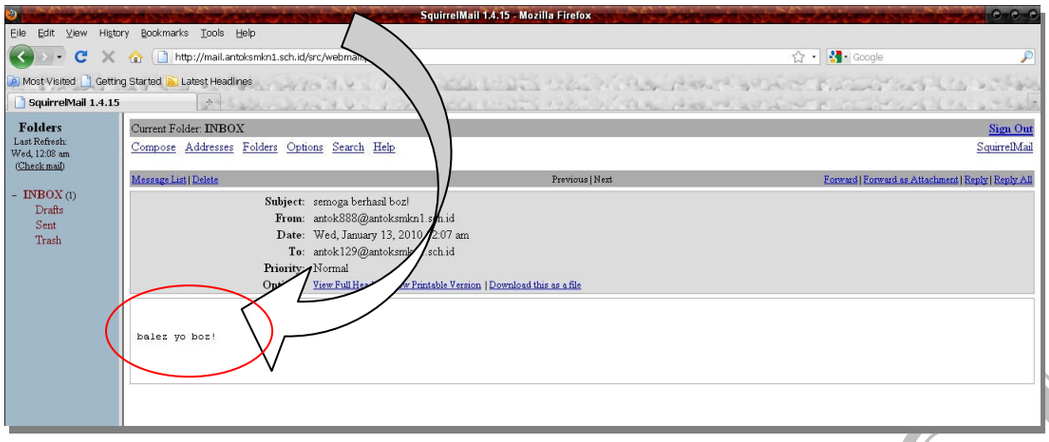
Coba kirim suatu pesan kepada user lain,,



Kalau tidak ada pesa eror pada waktu pengiriman,, berarti email berhasil kekirim,, Sekarang keluar dari user ini,, dan lihat di user tujuan,,, untuk melihat email masuk (inbox). Dan ini hasilnya..



Isi dari inbox

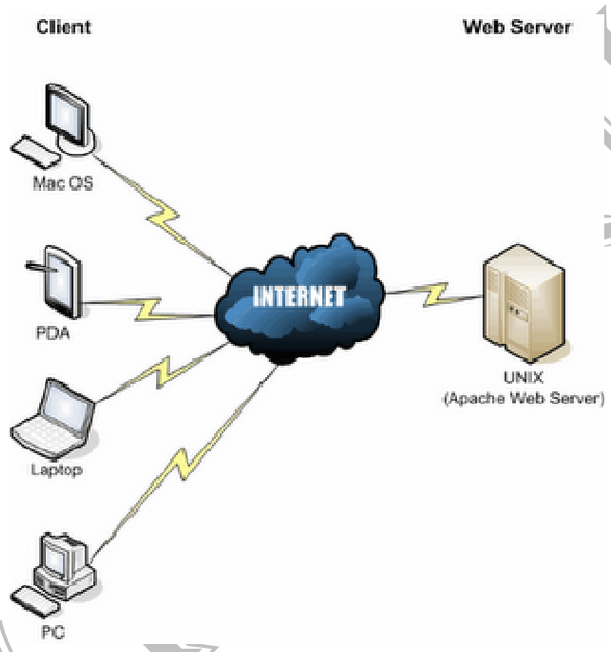


❖ **SELESAI**

Konfigurasi web server

1. Teori

Server web adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser web dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML. Server web yang terkenal diantaranya adalah Apache dan Microsoft Internet Information Service (IIS). Apache merupakan server web antar-platform, sedangkan IIS hanya dapat beroperasi di sistem operasi Windows. Server web juga dapat berarti komputer yang berfungsi seperti definisi di atas.



Web server adalah software yang menjadi tulang belakang dari *world wide web* (www). Web server menunggu permintaan dari client yang menggunakan browser seperti Netscape Navigator, Internet Explorer, Modzilla, dan program browser lainnya. Jika ada permintaan dari browser, maka *web server* akan memproses permintaan itu kemudian memberikan hasil prosesnya berupa data yang diinginkan kembali ke *browser.* Data ini mempunyai format yang standar, disebut dengan format SGML (*standar general markup language*). Data yang berupa format ini kemudian akan ditampilkan oleh browser sesuai dengan kemampuan browser tersebut. Contohnya, bila data yang dikirim berupa gambar, browser yang hanya mampu menampilkan teks (misalnya *lynx*) tidak akan mampu menampilkan gambar tersebut, dan jika ada akan menampilkan alternatifnya saja. Web server, untuk berkomunikasi dengan client-nya (*web browser*) mempunyai protokol sendiri, yaitu HTTP (*hypertext transfer protocol*).

Dengan protokol ini, komunikasi antar *web server* dengan client-nya dapat saling dimengerti dan lebih mudah. Seperti telah dijelaskan diatas, format data pada *world wide web* adalah SGML. Tapi para pengguna internet saat ini lebih banyak menggunakan format HTML (*hypertext markup language*) karena penggunaannya lebih sederhana dan mudah dipelajari. Kata *HyperText* mempunyai arti bahwa seorang pengguna internet dengan *web browser*nya dapat membuka dan membaca dokumen-dokumen yang ada dalam komputernya atau bahkan jauh tempatnya sekalipun.

Hal ini memberikan cita rasa dari suatu proses yang tridimensional, artinya pengguna internet dapat membaca dari satu dokumen ke dokumen yang lain hanya dengan mengklik beberapa bagian dari halaman-halaman dokumen (web) itu. Proses yang dimulai dari permintaan *webclient* (browser), diterima web server, diproses, dan dikembalikan hasil prosesnya oleh web server ke web client lagi dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web client*nya.

Web Server Apache

Apache merupakan *web server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya Apache mengeluarkan programnya yang dapat dijalankan di Windows NT. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya. Beberapa dukungan Apache :

1. Kontrol Akses.
   Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP
2. CGI (common Gateway Interface)
   Yang paling terkenal untuk digunakan adalah perl (*Practical Extraction and Report Language*), didukung oleh Apache dengan menempatkannya sebagai modul (mod_perl)
3. PHP (*Personal Home Page/PHP Hypertext Processor)*
   Program dengan metode semacam CGI, yang memproses teks dan bekerja di server. Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (mod_php). Hal ini membuat kinerja PHP menjadi lebih baik
4. SSI

Web server Apache mempunyai kelebihan dari beberapa pertimbangan di atas :

- Apache termasuk dalam kategori freeware

- Apache mudah sekali proses instalasinya jika dibanding web server lainnya seperti NCSA, IIS, dan lain-lain
- Mampu beroperasi pada berbagai platform sistem operasi.
- Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi
- Mudah dalam menambahkan peripheral lainnya ke dalam platform web servernya

Fasilitas atau ciri khas dari web server Apache adalah :

- Dapat dijadikan pengganti bagi NCSA web server
- Perbaikan terhadap kerusakan dan error pada NCSA 1.3 dan 1.4
- Apache merespon web client sangat cepat jauh melebihi NCSA
- Mampu di kompilasi sesuai dengan spesifikasi HTTP yang sekarang
- Kita dapat menetapkan respon error yang akan dikirim web server dengan menggunakan file atau skrip.
- Server apache dapat otomatis berkomunikasi dengan *client browser*nya untuk menampilkan tampilan terbaik pada *client browser*nya. Web server Apache secara otomatis menjalankan file *index.html*, halaman utamanya, untuk ditampilkan secara otomatis pada clientnya.
- Web server Apache mempunyai level-level pengamanan
- Apache mempunyai komponen dasar terbanyak di antara web server lain
- Ditinjau dari segi sejarah perkembangan dan prospeknya, Apache *web server* mempunyai prospek yang cerah. Apache berasal dari *web server* NCSA yang kemudian dikembangkan karena NCSA masih mempunyai kekurangan di bidang kompatibilitasnya dengan sistim operasi lain. Sampai saat ini, web server Apache terus dikembangkan oleh tim dari *apache.org*.
- Performasi dan konsumsi sumber daya dari web server Apache tidak terlalu banyak, hanya sekitar 20 MB untuk file-file dasarnya dan setiap *daemon*nya hanya memerlukan sekitar 950 KB memory per *child*

2. Instalasi dan Konfigurasi

Debian merupakan salah satu distro linux yang stabil untuk membangun sebuah web server. Mengapa memilih distro Debian untuk membangun web server? karena debian telah menyediakan instalasi otomatis beserta depedensi paket-paketnya melalui perintah apt-get milik debian.

Setelah berhasil Mengkonfigurasi DNS Server Selanjutnya kita akan mengkonfigurasi Web Server. Pada setting kali ini akan sangat mudah, karna hanya membutuhkan beberapa langkah saja. Silahkan ikuti langkah berikut: :

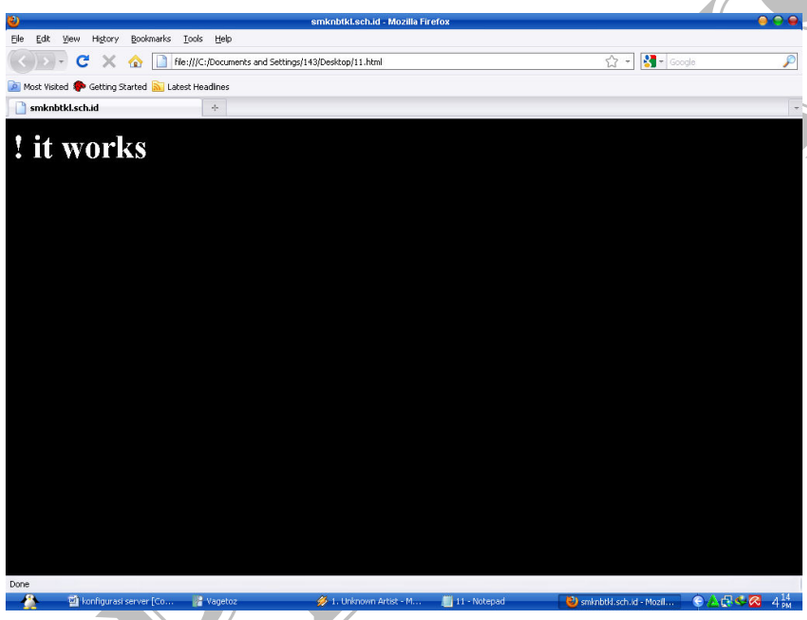
❖ Instalasi Apache2

```
# apt-get install apache2
```

Setelah install restart apache'nya..

```
# /etc/init.d/apache2  restart
```

Setelah selesai, cek di clien dengan mengetik di url http://localhost. Atau dengan domain yang sudah kita buat tadi www.smknbtkl.sch.id.



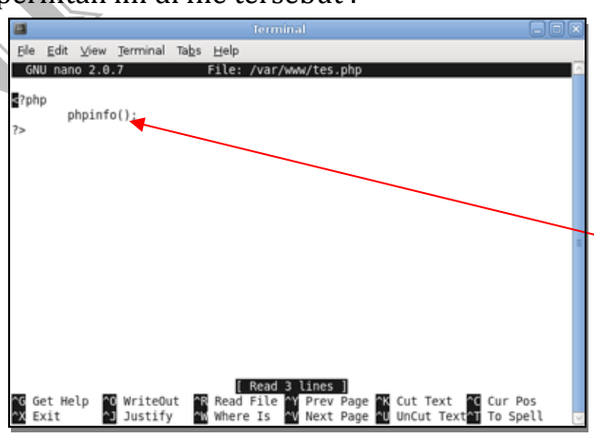
❖ Installasi php

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

Setelah selesai, kita tes dengan membuat file tes.php, caranya:

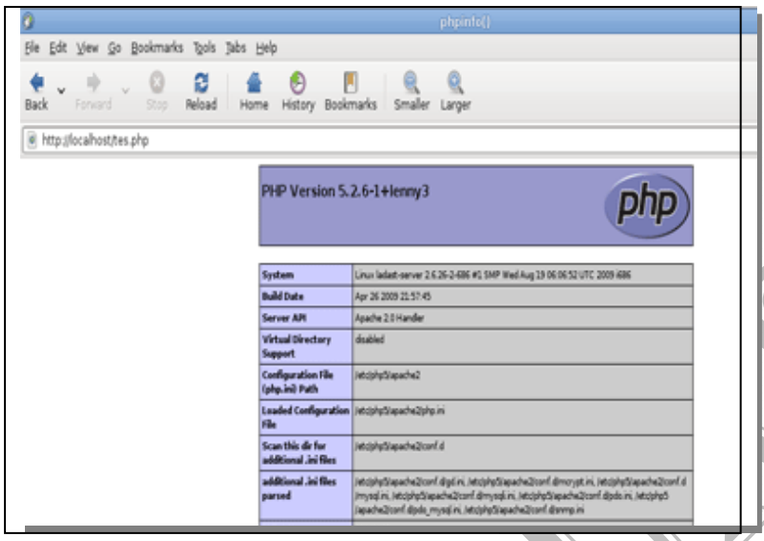
```
# nano /var/www/tes.php
```

Tulikan perintah ini di file tersebut :




```
<?php
        phpinfo();
?>
```

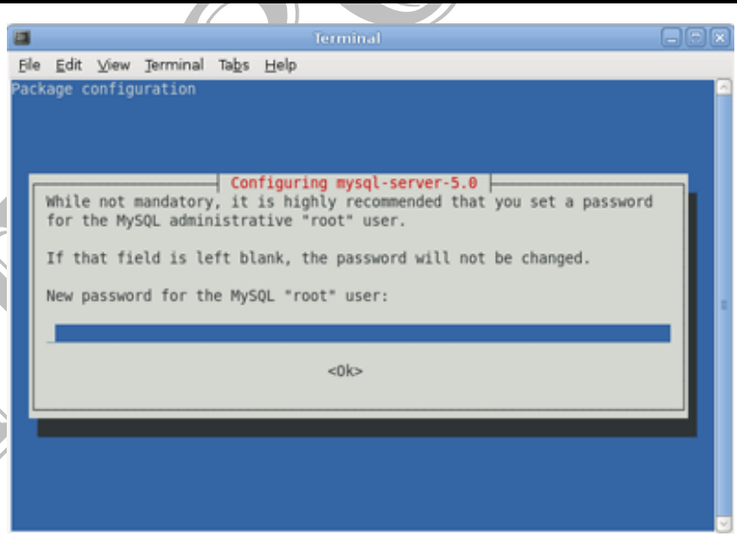
Simpan perubahan dan keluar,,, dan selanjut'nya Sekarang kita tes di browser dengan mengetikan http://smknbtkl.sch.id/tes.php Jika berhasil, hasilnya akan seperti gambar dibawah ini:



Jika tampil untuk men-unduh file tes.php tersebut, coba reboot terlebih dahulu, kemudian di tes lagi.

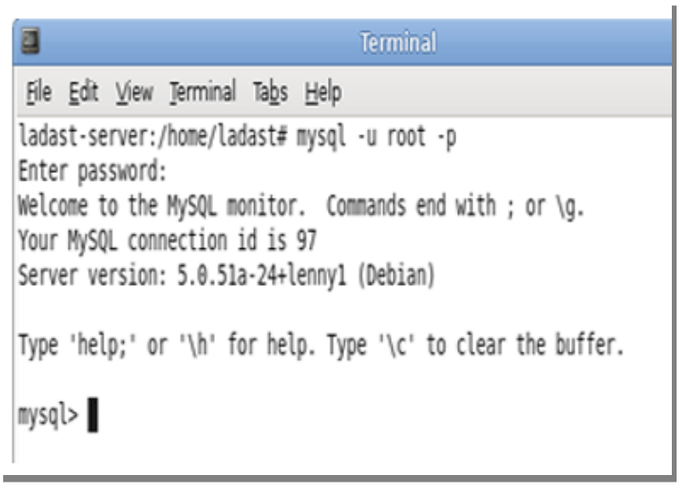
❖ Install php5-mysql di Debian

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



Silakan isikan password anda yang gampang untuk diingat.

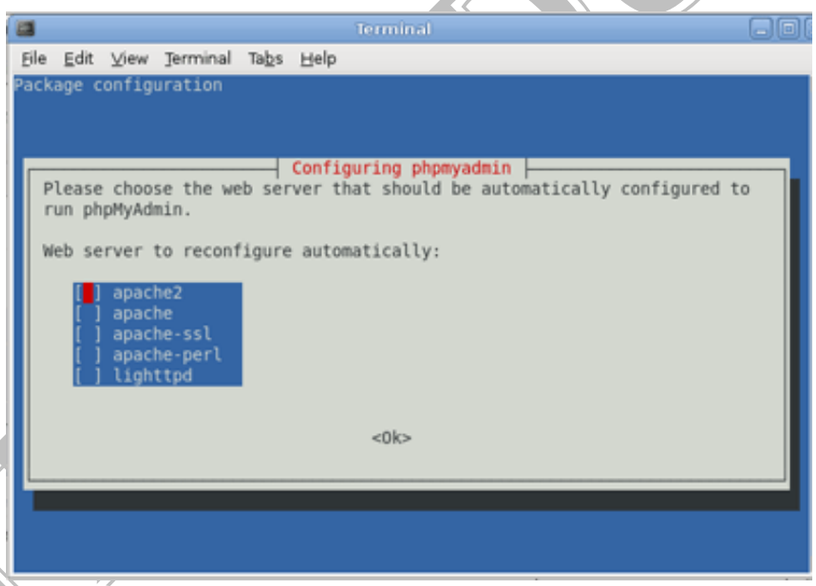
Kita tes dengan cara mengetikan perintah ini di terminal **#mysql -u root -p** kemudian isikan password root mysql anda yang telah anda isikan sebelumnya. Jika berhasil akan muncul form sebagai berikut:



- Install phpmyadmin

```
# apt-get install phpmyadmin
```

Setelah itu akan muncul form untuk mengkonfigurasi phpmyadmin. Disini saya pilih apache2. Gambarnya sebagai berikut :



Setelah itu kita buka file konfigurasi apache untuk phpmyadmin dengan mengetikan

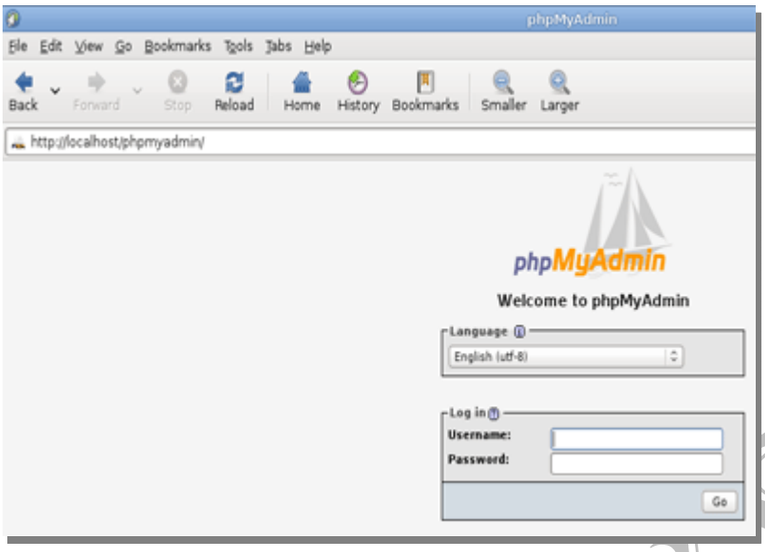
```
# nano /etc/phpmyadmin/apahce.conf
```

copykan isi seluruh perintah di file tersebut ke file **/etc/apache2/apache2.conf di** baris paling akhir

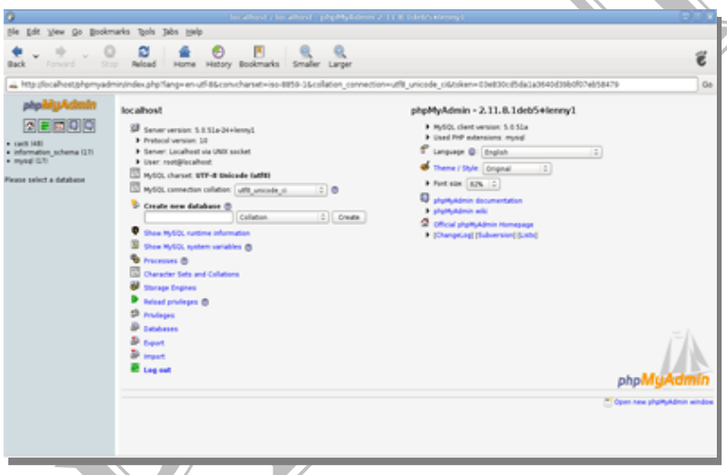
restart apache'nya

```
# /etc/init.d/apache.conf
```

- Kita tes dengan membuka browser dan mengetikan http://localhost/phpmyadmin Akan muncul form sebagai berikut:



- Masukan password root anda dan hasilnya terlihat seperti gambar dibawah ini:



- Konfugurasi Default apache2

  Secara default file web dari apache2 terletak di /var/www/index.html, jadi kalau kita mau merubah hamaman web kita kita tinggal rubah pada file index.html,

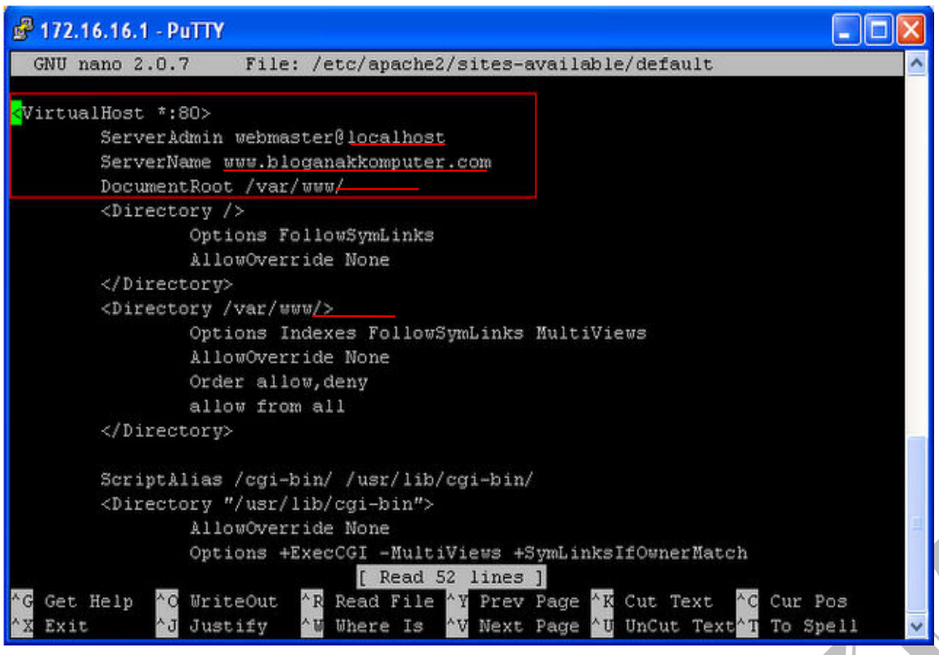
Setelah itu kita buat sub domain di Web kita Edit apache2

```
# cd /etc/apache2/sites-available/
# ls
#cp default download
```

Ubah dan tambahkan dengan domain yang kita buat pada bagian paling bawah.

Karena saya akan membuat download.smknbtkl.sch.id maka konfigurasinya :

1. Untuk local host yang di beri garis bawah di bawah maka ganti dengan nama domain contoh : smknbtkl.sch.id.
2. Untuk di bawahnya kita tuliskan ServerName download.smknbtkl.sch.id
3. Dan untuk di bawahnya setelah /var/www/download, mengapa download karena subdomain yang akan kita buat adalah download



Buat juga folder untuk subdomain download (sebab saya membuat domain download.smknbtkl.sch.id jika subdomain anda berbeda, tinggal sesuikan saja).

```
# mkdir /var/www/download
# cp index.html /var/www/download
# nano /var/www/download/index.html
```

Isi script html sesuai yang kita inginkan (dapat juga dengan php script). Contohnya

Ini adalah halaman Download Bloganakkomputer

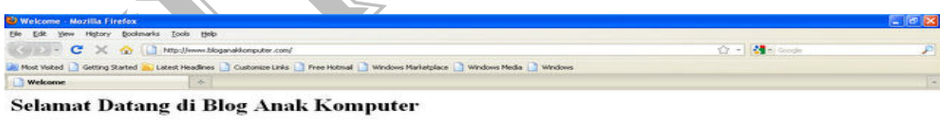
Restart apache2

```
# /etc/init.d/apache2 restart
```

2. Pengujian

Untuk pengujian, akan saya coba dari computer client dengan browser Mozilla Firefox (atau dapat dengan browser yang lain).

> Pertama buka www.smknbtkl.sch.id maka akan keluar:



Jika telah berhasil keluar halaman sesuia yang kita buat, berarti Virtual Host kita telah bekerja dengan baik. Namun jika belum keluar, cek sekali lagi mungkin konfigurasi apache2 atau file index yang kita buat salah tempat.

Sekian.

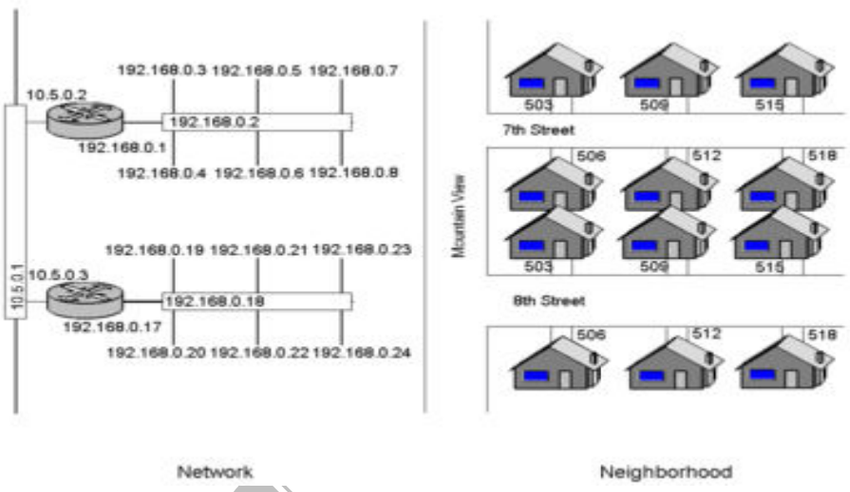
Konfigurasi router

1. Teori

Router

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai *routing*. Proses *routing* terjadi pada lapisan 3 (Lapisan jaringan seperti Internet Protocol) dari stack protokol tujuh-lapis OSI.

Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. Router berbeda dengan *switch*. *Switch* merupakan penghubung beberapa alat untuk membentuk suatu *Local Area Network* (LAN).



Analogi Router dan Switch

Sebagai ilustrasi perbedaan fungsi dari *router* dan *switch* merupakan suatu jalanan, dan *router* merupakan penghubung antar jalan. Masing-masing rumah berada pada jalan yang memiliki alamat dalam suatu urutan tertentu. Dengan cara yang sama, *switch* menghubungkan berbagai macam alat, dimana masing-masing alat memiliki alamat IP sendiri pada sebuah LAN.

Router sangat banyak digunakan dalam jaringan berbasis teknologi protokol TCP/IP, dan router jenis itu disebut juga dengan *IP Router*. Selain *IP Router*, ada lagi *AppleTalk Router*, dan masih ada beberapa jenis *router* lainnya. Internet merupakan contoh utama dari sebuah jaringan yang memiliki banyak *router IP*. *Router* dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan

internetwork, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya.

Router juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda (seperti halnya *router wireless* yang pada umumnya selain ia dapat menghubungkan komputer dengan menggunakan radio, ia juga mendukung penghubungan komputer dengan kabel UTP), atau berbeda arsitektur jaringan, seperti halnya dari Ethernet ke Token Ring.

Router juga dapat digunakan untuk menghubungkan LAN ke sebuah layanan telekomunikasi seperti halnya telekomunikasi *leased line* atau *Digital Subscriber Line* (DSL). *Router* yang digunakan untuk menghubungkan LAN ke sebuah koneksi *leased line* seperti T1, atau T3, sering disebut sebagai *access server*. Sementara itu, *router* yang digunakan untuk menghubungkan jaringan lokal ke sebuah koneksi DSL disebut juga dengan *DSL router*. *Router-router* jenis tersebut umumnya memiliki fungsi *firewall* untuk melakukan penapisan paket berdasarkan alamat sumber dan alamat tujuan paket tersebut, meski beberapa router tidak memilikinya. *Router* yang memiliki fitur penapisan paket disebut juga dengan *packet-filtering router*. *Router* umumnya memblokir lalu lintas data yang dipancarkan secara *broadcast* sehingga dapat mencegah adanya *broadcast storm* yang mampu memperlambat kinerja jaringan.

Jenis-jenis router

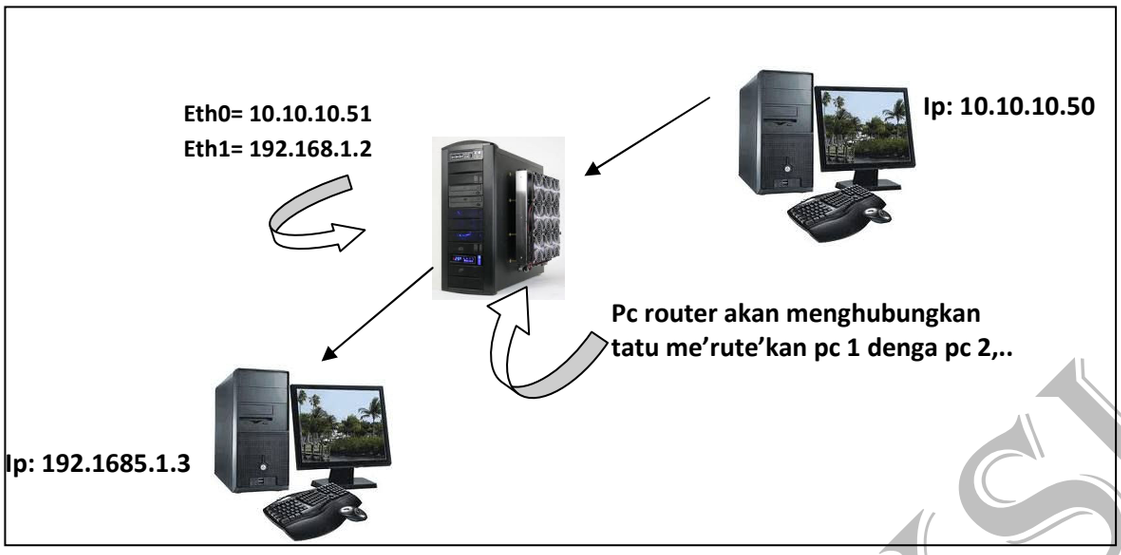
Secara umum, *router* dibagi menjadi dua buah jenis, yakni:

- *static router* (*router* statis): adalah sebuah *router* yang memiliki tabel *routing* statis yang di setting secara manual oleh para administrator jaringan.
- *dynamic router* (*router* dinamis): adalah sebuah *router* yang memiliki dab membuat tabel *routing* dinamis, dengan mendengarkan lalu lintas jaringan dan juga dengan saling berhubungan dengan *router* lainnya.

PC Router

Pc router adalah sebuah router yang dibuat dengan menggunakan system operasi berbasis server, yang dikonfigurasi bertujuan menjalankan 2 network dengna address yang berbeda,, cara kerja pc router ini sama seperti router,,

Misilakan kita mempunya 1 pc dengan ip address 10.10.10.50 (kls A) dan 1 pc lagi dengan ip address 192.168.1.3 (kls C),, kalau kita hubungkan pc tersebut pasti gak akan terhubung,, sebab hostname, kelas dan net id'nya berbeda,, nah untuk itu kita bisa membuat suat pc yang berguna sebagai router atau yang merute'kan, atau menyambungkan,, jadi otomatis di pc router harus ada 2 lan, missal yang satu on board (eth0, klo pd linux) dan yang satu lagi lan card (eth1, klo pd linux),, liha gambar berikut



2. Konfigurasi

Misalkan :

Pc 1 = 192.168.1.3/24 (pc server smknbtkl.sch.id)

Pc2 = 10.10.10.5/8 (pc router)

Konfigurasi untuk pc router'nya adalah

- Lan on board (eth0) ip'nya 10.10.10.5
- Lan card (eth1) ip'nya 192.168.1.2

Langkah2 & konfigurasi

❖ Edit networking

```
# nano /etc/network/interfaces
```

Tabahkan atau ketikan di baris yang paling bawah setingan untuk ip eth1, supaya dapat terhubung dengan pc yang terhubung ke eth1,,

Tambahakan scrip berikut dan ketka dibari yang paling bawah di bawah kata *dns-search smknbtkl.sch.id*

```
auto eth1
iface eth1 inet static
     address 192.168.1.2
     netmask 255.255.255.0
     network 192.168.1.0
     broadcast 192.165.1.255
     gateway   192.168.1.1
```

Sesuaikan dengan network yang ada di pc yang akan dihubungkan'nya

❖ Edit file rc.local yang ada di /etc/

```
# nano /etc/rc.local
```

Sebelum baris terakhir yaitu exit 0, tambahkan diatas'nya sebagai berikut:

iptables –A POSTROUTING –t nat –o eth1 –j MASQUERADE

simpan perubahan…

- Edit file sysctl.conf yang ada di /etc/

```
# nano /etc/sysctl.conf
```

Bertujuan untu pengaktipa ip forward,,, hilangkah tanda # pada

```
#net.ipv4-ip_forward=1
```

- Restart network'nya…

```
# /etc/init.d/networking restart
```

- Restar system

```
# reboot
```

- Setelah hidup lagi,, lakukan perintahh pada konsol

```
# iptables –A POSTROUTING –t nat –o eth1 –j MASQUERADE
```

- Restar lagi network dan system'nya..
- Selanjutnya Masukan Perintah

```
# iptables-save
```

- Cek di semua pc,,, apakah terhubung ke pc yang lain'nya tidak,, kalu di ping konek, maka router sudah jalan,,,

Konfigurasi proxy SERVER

1. Teori

Proxy

Proxy server (peladen proxy) adalah sebuah komputer server atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan *request* terhadap content dari Internet atau intranet.

Proxy Server bertindak sebagai *gateway* terhadap dunia ini Internet untuk setiap komputer klien. *Proxy server* tidak terlihat oleh komputer klien: 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 *request* itu datang secara langsung dari komputer klien, bukan dari *proxy server*.

Proxy server juga dapat digunakan untuk mengamankan jaringan pribadi yang dihubungkan ke sebuah jaringan publik (seperti halnya Internet). *Proxy server* memiliki lebih banyak fungsi daripada *router* yang memiliki fitur *packet filtering* karena memang *proxy server* beroperasi pada level yang lebih tinggi dan memiliki kontrol yang lebih menyeluruh terhadap akses jaringan. *Proxy server* yang berfungsi sebagai sebuah "agen keamanan" untuk sebuah jaringan pribadi, umumnya dikenal sebagai *firewall*.

Squid proxy

Squid adalah sebuah daemon yang digunakan sebagai *proxy server* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat server web dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, caching situs web, dan caching pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga pada membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Meskipun seringnya digunakan untuk protokol HTTP dan FTP, Squid juga menawarkan dukungan terbatas untuk beberapa protokol lainnya termasuk *Transport Layer Security* (TLS), Secure Socket Layer (SSL), Internet Gopher, dan HTTPS. Versi Squid 3.1 mencakup dukungan protokol IPv6 dan Internet Content Adaptation Protocol (ICAP).

Squid pada awalnya dikembangkan oleh Duane Wessels sebagai "Harvest object cache", yang merupakan bagian dari proyek Harvest yang dikembangkan di University of Colorado at Boulder. Pekerjaan selanjutnya dilakukan hingga selesai di University of California, San Diego dan didanai melalui National Science Foundation. Squid kini hampir secara eksklusif dikembangkan dengan cara usaha sukarela.Squid umumnya

didesain untuk berjalan di atas sistem operasi mirip UNIX, meski Squid juga bisa berjalan di atas sistem operasi Windows. Karena dirilis di bawah lisensi GNU General Public License, maka Squid merupakan perangkat lunak bebas

Web proxy

Caching merupakan sebuah cara untuk menyimpan objek-objek Internet yang diminta (seperti halnya data halaman web) yang bisa diakses melalui HTTP, FTP dan Gopher di dalam sebuah sistem yang lebih dekat dengan situs yang memintanya. Beberapa penjelajah web dapat menggunakan *cache* Squid lokal untuk sebagai *server proxy HTTP*, sehingga dapat mengurangi waktu akses dan juga tentu saja konsumsi *bandwidth*. Hal ini sering berguna bagi para penyedia layanan Internet untuk meningkatkan kecepatan kepada para pelanggannya, dan LAN yang membagi saluran Internet. Karena memang bentuknya sebagai *proxy* (ia berlaku sebagaimana layaknya klien, sesuai dengan permintaan klien), *web cache* bisa menyediakan anonimitas dan keamanan. Tapi, *web cache* juga bisa menjadi masalah yang signifikan bila melihat maslah privasi, karena memang ia dapat mencatat banyak data, termasuk URL yang diminta oleh klien, kapan hal itu terjadi, nama dan versi penjelajah web yang digunakan klien serta sistem operasinya, dan dari mana ia mengakses situs itu.

Selanjutnya, sebuah program klien (sebagai contoh adalah penjelajah web) bisa menentukan secara ekplisit *proxy server* yang digunakan bila memang hendak menggunakan proxy (umumnya bagi para pelanggan ISP) atau bisa juga menggunakan proxy tanpa konfigurasi ekstra, yang sering disebut sebagai "*Transparent Caching*", di mana semua permintaan HTTP ke jaringan luar akan diolah oleh *proxy server* dan semua respons disimpan di dalam *cache*. Kasus kedua umumnya dilakukan di dalam perusahaan dan korporasi (semua klien berada di dalam LAN yang sama) dan sering memiliki masalah privasi yang disebutkan di atas.

Squid memiliki banyak fitur yang bisa membantu melakukan koneksi secara anonim, seperti memodifikasi atau mematikan beberapa *field header* tertentu dalam sebuah permintaan HTTP yang diajukan oleh klien. Saat itu terpenuhi, apa yang akan dilakukan oleh Squid adalah tergantung orang yang menangani komputer yang menjalankan Squid. Orang yang meminta halaman web melalui sebuah jaringan yang secara transparan yang menggunakan biasanya tidak mengetahui bahwa informasi semua permintaan HTTP yang mereka ajukan dicatat oleh Squid.

Platform yang didukung

Squid dapat berjalan di atas sistem-sistem operasi berikut:

- AIX
- BSDI
- Digital Unix
- FreeBSD
- HP-UX
- IRIX
- Linux
- Mac OS X
- NetBSD
- NeXTStep
- OpenBSD
- SCO OpenServer
- Solaris
- UnixWare
- windows

2. Instalasi

❖ Install squid

```
# apt-get install squid
```

❖ Tunggu hingga proses installasi selesai,,

3. Konfigurasi

Setelah instalasi selesai selanjut'nya ktia konfigurasi paket squid tersebut,,, sebelum ke konfiguarasi kita harus punya tujuan kita dalam membuat sebuah proxy server,, yaitu apakah transfarent atau non transfarent,, transparent artinya sebuah proxy yang otomatis bekerja di computer clien, tanpa melakukan pengaturan secara menual di kompuer clien,, sedangkan proxy non transfarent adalah proxy yang tidak otomatis bekerja di clien apabila kita tidak melakukan pengaturan ulang secara manual di computer clien,,

Nah disini saya mengkonfigurasi poxy yang transfarenn,, berikut langkah-langkah'nya...,,

❖ Stop dulu paket squid'nya…

```
# /etc/init.d/squid stop
```

❖ Edit file squid.conf yang berada di /etc/squid/squid.conf

```
# nano /etc/squid/squid.conf
```

- Cari baris2 sebagai berikut,, Hilangkan semua tanda pagar atau dan tambahkan pada baris2 yang diharus'kan untuk ditambahkan. perhatikan sesudah dan sebelum konfigurasi,,

```
#http_port 3128
#cache_mem 8 mb
#cache_dir ufs /var/spool/squid 500 16 256
#      auth_param basic children 5
#      auth_param basic realm squid proxy.caching web server
#       auth_param basic credentalsttl 2 hours
#      auth_param basic casensitive off
```

Tambahkan

```
http_port 3128 transparent
cache_mem 16 mb
cache_dir ufs /var/spool/squid 500 16 256
cahe_mgr admin@smknbtkl.sch.id
visible_hostname proxy.smknbtkl.sch.id
auth_param basic children 5
auth_param basic realm squid proxy.caching web server
auth_param basic creaden tal sttl 2 hours
auth_param basic casensitive off
```

Buat baru

- Lalu simpan konfigurasi..,
- Konfigurasi untuk pemblokan situs, (masih di squid.conf)
- `# nano /etc/squid/squid.conf`
- Cari barsi acl CONNECT . . . . . dan tambahkan dibawah'nya scrip berikut,,

```
Acl situs url_regex -I
"/etc/situsterlarang.txt"
http_access deny  situs
acl lan src 10.10.10.5
http_access allow lan
http_access allow all
```

Ip pc router,, semisal squid di jalankan di cp router

- Simpan konfigurasi..,
- Buat file untuk menyimpan alamat-alamat situs yang akan kita blok,, semisal nama file'nya situsterlarang.txt,, dan simpan di /etc/, .

```
# touch situs terlarang.txt
# nano situsterlarang.txt
```

- Kita masukan nama situs yang akan kita blok,, masukan nama dari domain yang sudah kita buat,, semisal pc router'nya terhubung ke server..,, kitaa isi

  www.smknbtkl.sch.id.

  Mail.smknbtkl.sch.id.

  Facebook.com

- Simpan perubahan..
- Baut swap

```
# squid -z
```

- Edit file rc.local,, bertujuan untuk mengaktifkan ip_forward dan membuat table routing,, perintah'nya sebagai berikut,,

```
# nano /etc/rc.local
```

```
icho "1" >> /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -s 10.10.10.5/8 -p tcp -dport 80 -j REDIRECT -to-port 3128
```

Tambahkan ke-2 bari tersebut tepat diatas'nya exit 0

- Simpan perubahan
- Setelah di simpan,, lalu jalan'kan perintah berikut:

```
# iptables -t nat -A PREROUTING -s 10.10.10.5/8 -p tcp -dport
  80 -j REDIRECT -to-port 3128
```

&

```
# iptables -t nat -A POSTROUTING -s 10.10.10.5/8 -j MASQUERADE
```

&

```
# iptables -t nat -A POSTROUTING -s 10.10.10.5/8 -j MASQUERADE
```

- Restar squd'nya...

```
# /etc/init.d/squid restart
```

- Cek di clienn,, ketikan di url, dengan domain yang sudah kita blok tadi,, misalkan disini www.mail.smknbtkl.sch.id

Konfigurasi file sharing

1. **Teori**
2. **Instalasi**

   - Instalasi samba

     ```
     # apt-get install samba
     ```

   - Tunggu hingga proses instalasi selesai

3. **Konfigurasi**

   - Edit file smb.conf yang ada di /etc/samba/

     ```
     # nano /etc/samba/smb.conf
     ```

   - Cari baris ***security = user***,
   - Ganti user'nya dengan ***share*** , sehingga menjadi ***security = share***,
   - Supaya tidak meminta username dan password saat mengecek di windows,, bila memang ktia tidak mau pake password..,

     security = user, terdapat di baris atau di dalam

     ##### authentication #####, jadi cari bari yang seperti itu, dan ***security = user***, ada didalam'nya atau dibawah'nya...
   - Buat konfigurasi untuk pengaturan file atau folder yang akan kita sharing'kan,, missal di home kita mempunya folder **TKJ** dan folder **TKJ** tersebut ada di user **smknbtkl,** maksud'nya (/home/smknbtkl/tkj/)**,** dan folder **TKJ** tersebut akan kita sharingkan., maka konfigurasi'nya adalah sebagai berikut,,
   - Tambah'kan scrip berikut pada smb.conf yang kita edit pada bari yang paling bawah atau (ctrl+w+v).,

```
[share-smknbtkl]
     Comment          = sharing from smknbtkl
     Path             = /home/smknbtkl/tkj/
     Browserable      = yes
     Red only         = no
     Gues ok          = yes
```

Supaya log in tanpa pasword

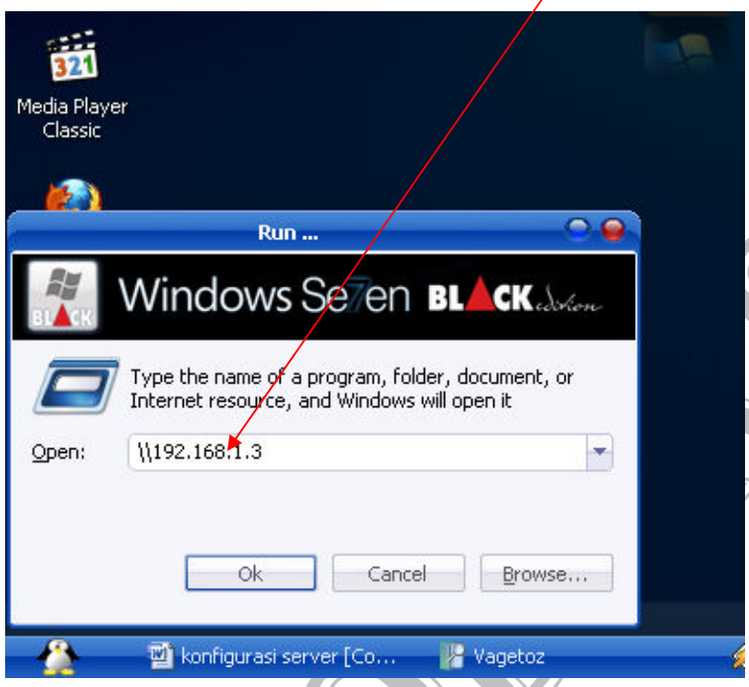
Agar yang di sharing dpt di buka di windows

Simpan perubahan

- Restar samba'nya,,,

```
# /etc/init.s/samba restart
```

- Cek di windows,, misal'kan pake run, dan ketikan \\192.168.1.3 (ip yang kita install samba, yaitu server)



- Maka akan tampil polder yang kita sharing'kan dari pc server

Konfigurasi firewall

1. Teori

Firewall atau tembok-api adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas jaringan yang dianggap aman untuk melaluinya dan mencegah

lalu lintas jaringan yang tidak aman. Umumnya, sebuah tembok-api diterapkan dalam sebuah mesin terdedikasi, yang berjalan pada pintu gerbang (gateway) antara jaringan lokal dan jaringan lainnya. Tembok-api umumnya juga digunakan untuk mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan pribadi dari pihak luar. Saat ini, istilah firewall menjadi istilah lazim yang merujuk pada sistem yang mengatur komunikasi antar dua jaringan yang berbeda. Mengingat saat ini banyak perusahaan yang memiliki akses ke Internet dan juga tentu saja jaringan berbadan hukum di dalamnya, maka perlindungan terhadap modal digital perusahaan tersebut dari serangan para peretas, pemata-mata, ataupun pencuri data lainnya, menjadi hakikat.

Jenis-jenis Firewall

Firewall terbagi menjadi dua jenis, yakni sebagai berikut

- Personal Firewall: Personal Firewall didesain untuk melindungi sebuah komputer yang terhubung ke jaringan dari akses yang tidak dikehendaki. Firewall jenis ini akhir-akhir ini berevolusi menjadi sebuah kumpulan program yang bertujuan untuk mengamankan komputer secara total, dengan ditambahkannya beberapa fitur pengaman tambahan semacam perangkat proteksi terhadap virus, anti-spyware, anti-spam, dan lainnya. Bahkan beberapa produk firewall lainnya dilengkapi dengan fungsi pendeteksian gangguan keamanan jaringan (Intrusion Detection System). Contoh dari firewall jenis ini adalah Microsoft Windows Firewall (yang telah terintegrasi dalam sistem operasi Windows XP Service Pack 2, Windows Vista dan Windows Server 2003 Service Pack 1), Symantec Norton Personal Firewall, Kerio Personal Firewall, dan lain-lain. Personal Firewall secara umum hanya memiliki dua fitur utama, yakni Packet Filter Firewall dan Stateful Firewall.
- Network Firewall: Network “”Firewall didesain untuk melindungi jaringan secara keseluruhan dari berbagai serangan. Umumnya dijumpai dalam dua bentuk, yakni sebuah perangkat terdedikasi atau sebagai sebuah perangkat lunak yang diinstalasikan dalam sebuah server. Contoh dari firewall ini adalah Microsoft Internet Security and Acceleration Server (ISA Server), Cisco PIX, Cisco ASA,

  IPTables dalam sistem operasi GNU/Linux, pf dalam keluarga sistem operasi Unix BSD, serta SunScreen dari Sun Microsystems, Inc. yang dibundel dalam sistem operasi Solaris. Network Firewall secara umum memiliki beberapa fitur utama, yakni apa

yang dimiliki oleh personal firewall (packet filter firewall dan stateful firewall), Circuit Level Gateway, Application Level Gateway, dan juga NAT Firewall. Network Firewall umumnya bersifat transparan (tidak terlihat) dari pengguna dan menggunakan teknologi routing untuk menentukan paket mana yang diizinkan, dan mana paket yang akan ditolak.

Fungsi Firewall

Secara fundamental, firewall dapat melakukan hal-hal berikut:

» Mengatur dan mengontrol lalu lintas jaringan
» Melakukan autentikasi terhadap akses
» Melindungi sumber daya dalam jaringan privat
» Mencatat semua kejadian, dan melaporkan kepada administrator

Mengatur dan Mengontrol Lalu lintas jaringan

Fungsi pertama yang dapat dilakukan oleh firewall adalah firewall harus dapat mengatur dan mengontrol lalu lintas jaringan yang diizinkan untuk mengakses jaringan privat atau komputer yang dilindungi oleh firewall. Firewall melakukan hal yang demikian, dengan melakukan inspeksi terhadap paket-paket dan memantau koneksi yang sedang dibuat, lalu melakukan penapisan (filtering) terhadap koneksi berdasarkan hasil inspeksi paket dan koneksi tersebut.

Melakukan autentikasi terhadap akses

Fungsi fundamental firewall yang kedua adalah firewall dapat melakukan autentikasi terhadap akses.

Protokol TCP/IP dibangun dengan premis bahwa protokol tersebut mendukung komunikasi yang terbuka. Jika dua host saling mengetahui alamat IP satu sama lainnya, maka mereka diizinkan untuk saling berkomunikasi. Pada awal-awal perkembangan Internet, hal ini boleh dianggap sebagai suatu berkah. Tapi saat ini, di saat semakin banyak yang terhubung ke Internet, mungkin kita tidak mau siapa saja yang dapat berkomunikasi dengan sistem yang kita miliki. Karenanya, firewall dilengkapi dengan fungsi autentikasi dengan menggunakan beberapa mekanisme autentikasi, sebagai berikut:

- Firewall dapat meminta input dari pengguna mengenai nama pengguna (user name) serta kata kunci (password). Metode ini sering disebut sebagai extended authentication atau xauth. Menggunakan xauth pengguna yang mencoba untuk membuat sebuah koneksi akan diminta input mengenai nama dan kata kuncinya sebelum akhirnya diizinkan oleh firewall. Umumnya, setelah koneksi diizinkan oleh kebijakan keamanan dalam firewall, firewall pun tidak perlu lagi mengisikan

input password dan namanya, kecuali jika koneksi terputus dan pengguna mencoba menghubungkan dirinya kembali.

- Metode kedua adalah dengan menggunakan sertifikat digital dan kunci publik. Keunggulan metode ini dibandingkan dengan metode pertama adalah proses autentikasi dapat terjadi tanpa intervensi pengguna. Selain itu, metode ini lebih cepat dalam rangka melakukan proses autentikasi. Meskipun demikian, metode ini lebih rumit implementasinya karena membutuhkan banyak komponen seperti halnya implementasi infrastruktur kunci publik.
- Metode selanjutnya adalah dengan menggunakan Pre-Shared Key (PSK) atau kunci yang telah diberitahu kepada pengguna. Jika dibandingkan dengan sertifikat digital, PSK lebih mudah diimplenentasikan karena lebih sederhana, tetapi PSK juga mengizinkan proses autentikasi terjadi tanpa intervensi pengguna. Dengan menggunakan PSK, setiap host akan diberikan sebuah kunci yang telah ditentukan sebelumnya yang kemudian digunakan untuk proses autentikasi. Kelemahan metode ini adalah kunci PSK jarang sekali diperbarui dan banyak organisasi sering sekali menggunakan kunci yang sama untuk melakukan koneksi terhadap host-host yang berada pada jarak jauh, sehingga hal ini sama saja meruntuhkan proses autentikasi. Agar tercapai sebuah derajat keamanan yang tinggi, umumnya beberapa organisasi juga menggunakan gabungan antara metode PSK dengan xauth atau PSK dengan sertifikat digital.

Dengan mengimplementasikan proses autentikasi, firewall dapat menjamin bahwa koneksi dapat diizinkan atau tidak. Meskipun jika paket telah diizinkan dengan menggunakan inspeksi paket (PI) atau berdasarkan keadaan koneksi (SPI), jika host tersebut tidak lolos proses autentikasi, paket tersebut akan dibuang.

Melindungi sumber daya dalam jaringan privat

Salah satu tugas firewall adalah melindungi sumber daya dari ancaman yang mungkin datang. Proteksi ini dapat diperoleh dengan menggunakan beberapa peraturan pengaturan akses (access control), penggunaan SPI, application proxy, atau kombinasi dari semuanya untuk mencegah host yang dilindungi dapat diakses oleh host-host yang mencurigakan atau dari lalu lintas jaringan yang mencurigakan. Meskipun demikian, firewall bukanlah satu-satunya metode proteksi terhadap sumber daya, dan mempercayakan proteksi terhadap sumber daya dari ancaman terhadap firewall secara eksklusif adalah salah satu kesalahan fatal. Jika sebuah host yang menjalankan sistem operasi tertentu yang memiliki lubang keamanan yang belum ditambal dikoneksikan ke Internet, firewall mungkin tidak dapat mencegah dieksploitasinya host tersebut oleh host-

host lainnya, khususnya jika exploit tersebut menggunakan lalu lintas yang oleh firewall telah diizinkan (dalam konfigurasinya). Sebagai contoh, jika sebuah acket-inspection firewall mengizinkan lalu lintas HTTP ke sebuah web server yang menjalankan sebuah layanan web yang memiliki lubang keamanan yang belum ditambal, maka seorang pengguna yang "iseng" dapat saja membuat exploit untuk meruntuhkan web server tersebut karena memang web server yang bersangkutan memiliki lubang keamanan yang belum ditambal. Dalam contoh ini, web server tersebut akhirnya mengakibatkan proteksi yang ditawarkan oleh firewall menjadi tidak berguna. Hal ini disebabkan oleh firewall yang tidak dapat membedakan antara request HTTP yang mencurigakan atau tidak. Apalagi, jika firewall yang digunakan bukan application proxy. Oleh karena itulah, sumber daya yang dilindungi haruslah dipelihara dengan melakukan penambalan terhadap lubang-lubang keamanan, selain tentunya dilindungi oleh firewall.

Mencatat semua kejadian, dan melaporkan kepada administrator

2. Konfigurasi

Untuk firewall disini, kita membangun firewall dengan menggunakan iptables, jadi tidak ada paket yang di install,, untuk konfigurasi'nya ikutil langkah2 berikut..

- Edit file rc.local yang ada di /etc/

```
# nano /etc/rc.local
```

- Tambahkan konfigurasi berikut di atas exit 0

```
#flashing
iptables -F
iptables -t nat -F
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

# paket ( ini adalah filtering paket yang masuk )
iptables -A INPUT -m state --state NEW -i eth1 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state NEW,INVALID -i eth1 -j DROP

iptables -A FORWARD -m state --state NEW -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW,INVALID -j DROP

iptables -A OUTPUT -m state --state NEW -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state NEW,INVALID -j DROP

# netbios ( DROP untuk netbios )
iptables -A FORWARD -p udp --sport 137:139 -j DROP
```

```
# ping ( action ACCEPT berarti dapat ping, kalau action DROP berati tidak
dapat di ping)

iptables -A INPUT -p icmp -j ACCEPT
iptables -A FORWARD -p icmp -j ACCEPT
iptables -A OUTPUT -p icmp -j ACCEPT

# browser ( action ACCEPT berarti dapat mengakses web, kalau action DROP
berati tidak dapat access web )
iptables -A INPUT -p tcp -m multiport --ports 80 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport --ports 80 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 80 -j ACCEPT

# dns ( action ACCEPT berarti dns bisa di nslookup, kalau actions DROP maka
dns tdk bisa di nslookup )
iptables -A INPUT -m state --state NEW -p tcp -m multiport --ports 53 -j ACCEPT
iptables -A FORWARD -p udp -m multiport --ports 53 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 53 -j ACCEPT
iptables -A INPUT -p udp -m multiport --ports 53 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport --ports 53 -j ACCEPT
iptables -A OUTPUT -p udp -m multiport --ports 53 -j ACCEPT

# smtp ( actions ACCEPT berarti dapat mengirim email, kalau action DROP maka
tidak dapat )
iptables -A INPUT -p tcp -m multiport --ports 25 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport --ports 25 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 25 -j ACCEPT

# pop3 ( action ACCEPT berarti dapat melihat masuk pesan email, kalau action
DROP maka tidak )
iptables -A INPUT -p tcp -m multiport --ports 110 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport --ports 110 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 110 -j ACCEPT

#imap ( action ACCEPT berarti port smtp dapat diakses, dan action DROP
sebaliknya )
iptables -A INPUT -p tcp -m multiport --ports 143 -j ACCEPT
iptables -A FORWARD -p tcp -m multiport --ports 143 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --ports 143 -j ACCEPT

#ntp server ( action ACCEPT berarti port network time protokol dapat dilalui
#untuk update time, dan action DROP sebailnya)
iptables -A INPUT -p udp -m multiport --ports 123 -j ACCEPT
iptables -A FORWARD -p udp -m multiport --ports 123 -j ACCEPT
iptables -A OUTPUT -p udp -m multiport --ports 123 -j ACCEPT

# routing table
iptables -t nat -A POSTROUTING -o eth0 -s 172.16.1.0/29 -j MASQUERADE
```

```
# redirect ke server proxy
iptables -t nat -A PREROUTING -p tcp -m multiport --dport 80 -i eth1 -s
172.16.4.0/24 -j DNAT --to-destination 192.168.4.1:8080


# catatan : semua file rc.local kita taruh saja di router, karena router berfungsi
untuk merouting semua paket.
```

- Simpan perubahan
- Lalu ketikan perintah

```
# iptables-save
```

- Restart system
- Cek di clean,, cek apakah jalan atau tidak port yang sudah kita blok tadi,,

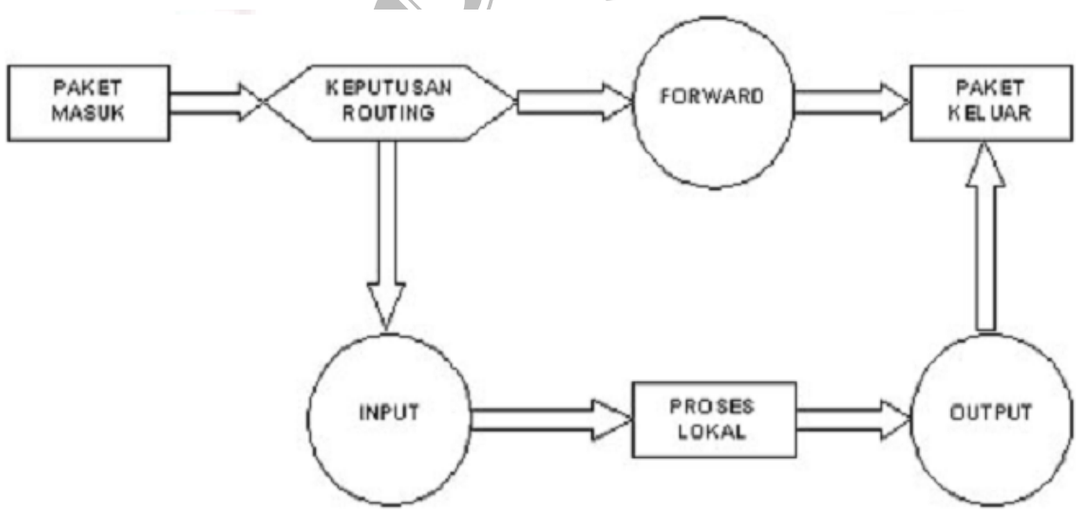
TUTORIAL IPTABLES

Sebelum mulai, diharapkan pembaca sudah memiliki pengetahuan dasar mengenai TCP/IP karena hal ini merupakan dasar dari penggunaan IPTables. Ada (sangat) banyak resource yang mendokumentasikan konsep dasar tentang TCP/IP, baik itu secara online maupun cetak. Silahkan googling untuk mendapatkannya.

Hal berikutnya yang harus anda persiapkan adalah sebuah komputer yang terinstall Linux. Akan lebih baik jika komputer anda memiliki 2 buah *network interface card*, sebab bisa menjalankan fungsi *packet forwarding*. Disarankan anda menggunakan linux dengan kernel 2.4 ke atas, karena (setahu saya) linux dengan kernel 2.4 ke atas sudah memiliki dukungan IPTables secara default, sehingga anda tidak perlu mengkompilasi ulang kernel anda. Bagi anda yang menggunakan kernel 2.2 atau sebelumnya, anda harus melakukan kompilasi kernel untuk memasukkan dukungan IPTables. Silahkan lihat tutorial Kompilasi kernel 2.4.x di Linux oleh mas Asfik.

Pendahuluan

IPTables memiliki tiga macam daftar aturan bawaan dalam tabel penyaringan, daftar tersebut dinamakan rantai firewall (*firewall chain*) atau sering disebut *chain* saja. Ketiga chain tersebut adalah INPUT, OUTPUT dan FORWARD.



Pada diagram tersebut, lingkaran menggambarkan ketiga rantai atau chain. Pada saat sebuah paket sampai pada sebuah lingkaran, maka disitulah terjadi proses penyaringan. Rantai akan memutuskan nasib paket tersebut. Apabila keputusannnya adalah DROP, maka paket tersebut akan di-drop. Tetapi jika rantai memutuskan untuk ACCEPT, maka paket akan dilewatkan melalui diagram tersebut.

Sebuah rantai adalah aturan-aturan yang telah ditentukan. Setiap aturan menyatakan "jika paket memiliki informasi awal (header) seperti ini, maka inilah yang harus dilakukan

terhadap paket". Jika aturan tersebut tidak sesuai dengan paket, maka aturan berikutnya akan memproses paket tersebut. Apabila sampai aturan terakhir yang ada, paket tersebut belum memenuhi salah satu aturan, maka kernel akan melihat kebijakan bawaan (default) untuk memutuskan apa yang harus dilakukan kepada paket tersebut. Ada dua kebijakan bawaan yaitu default DROP dan default ACCEPT.

Jalannya sebuah paket melalui diagram tersebut bisa dicontohkan sebagai berikut:

Perjalanan paket yang diforward ke host yang lain.

1. Paket berada pada jaringan fisik, contoh internet.
2. Paket masuk ke interface jaringan, contoh eth0.
3. Paket masuk ke chain PREROUTING pada table Mangle. Chain ini berfungsi untuk me-mangle menghaluskan) paket, seperti merubah TOS, TTL dan lain-lain.
4. Paket masuk ke chain PREROUTING pada tabel nat. Chain ini berfungsi utamanya untuk melakukan DNAT (Destination Network Address Translation).
5. Paket mengalami keputusan routing, apakah akan diproses oleh host lokal atau diteruskan ke host lain.
6. Paket masuk ke chain FORWARD pada tabel filter. Disinlah proses pemfilteran yang utama terjadi.
7. Paket masuk ke chain POSTROUTING pada tabel nat. Chain ini berfungsi utamanya untuk melakukan SNAT (Source Network Address Translation).
8. Paket keluar menuju interface jaringan, contoh eth1.
9. Paket kembali berada pada jaringan fisik, contoh LAN.

Perjalanan paket yang ditujukan bagi host local

1. Paket berada dalam jaringan fisik, contoh internet.
2. Paket masuk ke interface jaringan, contoh eth0.
3. Paket masuk ke chain PREROUTING pada tabel mangle.
4. Paket masuk ke chain PREROUTING pada tabel nat.
5. Paket mengalami keputusan routing.
6. Paket masuk ke chain INPUT pada tabel filter untuk mengalami proses penyaringan.
7. Paket akan diterima oleh aplikasi lokal.

Perjalanan paket yang berasal dari host local.

1. Aplikasi lokal menghasilkan paket data yang akan dikirimkan melalui jaringan.
2. Paket memasuki chain OUTPUT pada tabel mangle.
3. Paket memasuki chain OUTPUT pada tabel nat.
4. Paket memasuki chain OUTPUT pada tabel filter.
5. Paket mengalami keputusan routing, seperti ke mana paket harus pergi dan melalui interface mana.

6. Paket masuk ke chain POSTROUTING pada tabel NAT.
7. Paket masuk ke interface jaringan, contoh eth0.
8. Paket berada pada jaringan fisik, contoh internet.

1. Sintaks IPTables

iptables [-t table] command [match] [target/jump]

1. Table

IPTables memiliki 3 buah tabel, yaitu NAT, MANGLE dan FILTER. Penggunannya disesuaikan dengan sifat dan karakteristik masing-masing. Fungsi dari masing-masing tabel tersebut sebagai berikut :

NAT : Secara umum digunakan untuk melakukan Network Address Translation. NAT adalah penggantian field alamat asal atau alamat tujuan dari sebuah paket.

a. MANGLE : Digunakan untuk melakukan penghalusan (mangle) paket, seperti TTL, TOS dan MARK.
b. FILTER : Secara umum, inilah pemfilteran paket yang sesungguhnya.. Di sini bisa dintukan apakah paket akan di-DROP, LOG, ACCEPT atau REJECT

2. Command

Command pada baris perintah IPTables akan memberitahu apa yang harus dilakukan terhadap lanjutan sintaks perintah. Umumnya dilakukan penambahan atau penghapusan sesuatu dari tabel atau yang lain.

Command	Keterangan
-A **--append**	Perintah ini menambahkan aturan pada akhir chain. Aturan akan ditambahkan di akhir baris pada chain yang bersangkutan, sehingga akan dieksekusi terakhir
-D **--delete**	Perintah ini menghapus suatu aturan pada chain. Dilakukan dengan cara menyebutkan secara lengkap perintah yang ingin dihapus atau dengan menyebutkan nomor baris dimana perintah akan dihapus.
-R **--replace**	Penggunaannya sama seperti **--delete**, tetapi *command* ini menggantinya dengan entry yang baru.
-I **--insert**	Memasukkan aturan pada suatu baris di chain. Aturan akan dimasukkan pada baris yang disebutkan, dan aturan awal yang menempati baris tersebut akan digeser ke bawah. Demikian pula baris-baris selanjutnya.

-L **--list**	Perintah ini menampilkan semua aturan pada sebuah tabel. Apabila tabel tidak disebutkan, maka seluruh aturan pada semua tabel akan ditampilkan, walaupun tidak ada aturan sama sekali pada sebuah tabel. *Command* ini bisa dikombinasikan dengan option –v (verbose), -n (numeric) dan –x (exact).
-F **--flush**	Perintah ini mengosongkan aturan pada sebuah chain. Apabila chain tidak disebutkan, maka semua chain akan di-*flush*.
-N **--new-chain**	Perintah tersebut akan membuat chain baru.
-X **--delete-chain**	Perintah ini akan menghapus chain yang disebutkan. Agar perintah di atas berhasil, tidak boleh ada aturan lain yang mengacu kepada chain tersebut.
-P **--policy**	Perintah ini membuat kebijakan default pada sebuah chain. Sehingga jika ada sebuah paket yang tidak memenuhi aturan pada baris-baris yang telah didefinisikan, maka paket akan diperlakukan sesuai dengan kebijakan default ini.
-E **--rename-chain**	Perintah ini akan merubah nama suatu chain.

3. Option

Option digunakan dikombinasikan dengan command tertentu yang akan menghasilkan suatu variasi perintah.

Option	Command Pemakai	Keterangan
-v **--verbose**	**--list** **--append** **--insert** **--delete** **--replace**	Memberikan output yang lebih detail, utamanya digunakan dengan --list. Jika digunakan dengan --list, akan menampilkam K (x1.000), M (1.000.000) dan G (1.000.000.000).
-x **--exact**	**--list**	Memberikan output yang lebih tepat.
-n **--numeric**	**--list**	Memberikan output yang berbentuk angka. Alamat IP dan nomor port akan ditampilkan dalam bentuk angka dan bukan hostname ataupun nama aplikasi/servis.

--line-number	**--list**	Akan menampilkan nomor dari daftar aturan. Hal ni akan mempermudah bagi kita untuk melakukan modifikasi aturan, jika kita mau meyisipkan atau menghapus aturan dengan nomor tertentu.
--modprobe	**All**	Memerintahkan IPTables untuk memanggil modul tertentu. Bisa digunakan bersamaan dengan semua *command*.

4. Generic Matches

Generic Matches artinya pendefinisian kriteria yang berlaku secara umum. Dengan kata lain, sintaks generic matches akan sama untuk semua protokol. Setelah protokol didefinisikan, maka baru didefinisikan aturan yang lebih spesifik yang dimiliki oleh protokol tersebut. Hal ini dilakukan karena tiap-tiap protokol memiliki karakteristik yang berbeda, sehingga memerlukan perlakuan khusus.

Match	Keterangan
-p **--protocol**	Digunakan untuk mengecek tipe protokol tertentu. Contoh protokol yang umum adalah TCP, UDP, ICMP dan ALL. Daftar protokol bisa dilihat pada **/etc/protocols**. Tanda inversi juga bisa diberlakukan di sini, misal kita menghendaki semua protokol kecuali icmp, maka kita bisa menuliskan **--protokol ! icmp** yang berarti semua kecuali icmp.
-s **--src** **--source**	Kriteria ini digunakan untuk mencocokkan paket berdasarkan alamat IP asal. Alamat di sini bisa berberntuk alamat tunggal seperti 192.168.1.1, atau suatu alamat network menggunakan netmask misal 192.168.1.0/255.255.255.0, atau bisa juga ditulis 192.168.1.0/24 yang artinya semua alamat 192.168.1.x. Kita juga bisa menggunakan inversi.
-d **--dst** **--destination**	Digunakan untuk mecocokkan paket berdasarkan alamat tujuan. Penggunaannya sama dengan *match* **–src**
-i **--in-interface**	*Match* ini berguna untuk mencocokkan paket berdasarkan interface di mana paket datang. *Match* ini hanya berlaku pada chain INPUT, FORWARD dan PREROUTING

-o **--out-interface**	Berfungsi untuk mencocokkan paket berdasarkan interface di mana paket keluar. Penggunannya sama dengan **--in-interface**. Berlaku untuk chain OUTPUT, FORWARD dan POSTROUTING

5. Implicit Matches

Implicit Matches adalah match yang spesifik untuk tipe protokol tertentu. Implicit Match merupakan sekumpulan rule yang akan diload setelah tipe protokol disebutkan. Ada 3 Implicit Match berlaku untuk tiga jenis protokol, yaitu TCP matches, UDP matches dan ICMP matches.

a. TCP matches

Match	Keterangan
--sport **--source-port**	*Match* ini berguna untuk mecocokkan paket berdasarkan port asal. Dalam hal ini kia bisa mendefinisikan nomor port atau nama *service*-nya. Daftar nama service dan nomor port yang bersesuaian dapat dilihat di **/etc/services**. **--sport** juga bisa dituliskan untuk range port tertentu. Misalkan kita ingin mendefinisikan range antara port 22 sampai dengan 80, maka kita bisa menuliskan **--sport 22:80**. Jika bagian salah satu bagian pada range tersebut kita hilangkan maka hal itu bisa kita artikan dari port 0, jika bagian kiri yang kita hilangkan, atau 65535 jika bagian kanan yang kita hilangkan. Contohnya **--sport :80** artinya paket dengan port asal nol sampai dengan 80, atau **--sport 1024:** artinya paket dengan port asal 1024 sampai dengan 65535.Match ini juga mengenal inversi.
--dport **--destination-port**	Penggunaan *match* ini sama dengan match **--source-port**.
--tcp-flags	Digunakan untuk mencocokkan paket berdasarkan TCP *flags* yang ada pada paket tersebut. Pertama, pengecekan akan mengambil daftar *flag* yang akan diperbandingkan, dan kedua, akan memeriksa paket yang di-*set* 1, atau *on*. Pada kedua *list*, masing-masing entry-nya harus dipisahkan oleh koma dan tidak boleh ada spasi antar entry, kecuali spasi antar kedua *list. Match* ini mengenali SYN,ACK,FIN,RST,URG, PSH. Selain itu kita juga menuliskan ALL dan NONE. Match ini juga bisa menggunakan inversi.

--syn	*Match* ini akan memeriksa apakah flag SYN di-*set* dan ACK dan FIN tidak di-*set*. Perintah ini sama artinya jika kita menggunakan *match* **--tcp-flags SYN,ACK,FIN SYN** Paket dengan *match* di atas digunakan untuk melakukan *request* koneksi TCP yang baru terhadap server

b. UDP Matches

Karena bahwa protokol UDP bersifat connectionless, maka tidak ada flags yang mendeskripsikan status paket untuk untuk membuka atau menutup koneksi. Paket UDP juga tidak memerlukan acknowledgement. Sehingga Implicit Match untuk protokol UDP lebih sedikit daripada TCP.

Ada dua macam match untuk UDP:

--sport atau --source-port

--dport atau --destination-port

ICMP Matches Paket ICMP digunakan untuk mengirimkan pesan-pesan kesalahan dan kondisi-kondisi jaringan yang lain. Hanya ada satu implicit match untuk tipe protokol ICMP, yaitu :

--icmp-type

6. Explicit Matches

a. MAC Addres

Match jenis ini berguna untuk melakukan pencocokan paket berdasarkan MAC source address. Perlu diingat bahwa MAC hanya berfungsi untuk jaringan yang menggunakan teknologi ethernet.

iptables –A INPUT –m mac –mac-source 00:00:00:00:00:01

c. Multiport Matches

Ekstensi Multiport Matches digunakan untuk mendefinisikan port atau port range lebih dari satu, yang berfungsi jika ingin didefinisikan aturan yang sama untuk beberapa port. Tapi hal yang perlu diingat bahwa kita tidak bisa menggunakan port matching standard dan multiport matching dalam waktu yang bersamaan.

iptables –A INPUT –p tcp –m multiport --source-port 22,53,80,110

d. Owner Matches

Penggunaan match ini untuk mencocokkan paket berdasarkan pembuat atau pemilik/owner paket tersebut. Match ini bekerja dalam chain OUTPUT, akan tetapi penggunaan match ini tidak terlalu luas, sebab ada beberapa proses tidak memiliki owner (??).

iptables –A OUTPUT –m owner --uid-owner 500

Kita juga bisa memfilter berdasarkan group ID dengan sintaks --gid-owner. Salah satu penggunannya adalah bisa mencegah user selain yang dikehendaki untuk mengakses internet misalnya.

e. State Matches

Match ini mendefinisikan state apa saja yang cocok. Ada 4 state yang berlaku, yaitu NEW, ESTABLISHED, RELATED dan INVALID. NEW digunakan untuk paket yang akan memulai koneksi baru. ESTABLISHED digunakan jika koneksi telah tersambung dan paket-paketnya merupakan bagian dari koneki tersebut. RELATED digunakan untuk paket-paket yang bukan bagian dari koneksi tetapi masih berhubungan dengan koneksi tersebut, contohnya adalah FTP data transfer yang menyertai sebuah koneksi TCP atau UDP. INVALID adalah paket yang tidak bisa diidentifikasi, bukan merupakan bagian dari koneksi yang ada.

iptables –A INPUT –m state --state RELATED,ESTABLISHED

6. Target/Jump

Target atau jump adalah perlakuan yang diberikan terhadap paket-paket yang memenuhi kriteria atau match. Jump memerlukan sebuah chain yang lain dalam tabel yang sama. Chain tersebut nantinya akan dimasuki oleh paket yang memenuhi kriteria. Analoginya ialah chain baru nanti berlaku sebagai prosedur/fungsi dari program utama. Sebagai contoh dibuat sebuah chain yang bernama tcp_packets. Setelah ditambahkan aturan-aturan ke dalam chain tersebut, kemudian chain tersebut akan direferensi dari chain input.

iptables –A INPUT –p tcp –j tcp_packets

Target	Keterangan
-j ACCEPT **--jump ACCEPT**	Ketika paket cocok dengan daftar *match* dan target ini diberlakukan, maka paket tidak akan melalui baris-baris aturan yang lain dalam chain tersebut atau chain yang lain yang mereferensi chain tersebut. Akan tetapi paket masih akan memasuki chain-chain pada tabel yang lain seperti biasa.
-j DROP **--jump DROP**	Target ini men-*drop* paket dan menolak untuk memproses lebih jauh. Dalam beberapa kasus mungkin hal ini kurang baik, karena akan meninggalkan *dead socket* antara *client* dan *server*. Paket yang menerima target DROP benar-benar mati dan target tidak akan mengirim informasi tambahan dalam bentuk apapun kepada client atau server.

-j RETURN **--jump RETURN**	Target ini akan membuat paket berhenti melintasi aturan-aturan pada chain dimana paket tersebut menemui target RETURN. Jika chain merupakan *subchain* dari chain yang lain, maka paket akan kembali ke *superset chain* di atasnya dan masuk ke baris aturan berikutnya. Apabila *chain* adalah chain utama misalnya INPUT, maka paket akan dikembalikan kepada kebijakan default dari *chain* tersebut.
-j MIRROR	Apabila kompuuter A menjalankan target seperti contoh di atas, kemudian komputer B melakukan koneksi http ke komputer A, maka yang akan muncul pada browser adalah website komputer B itu sendiri. Karena fungsi utama target ini adalah membalik *source address* dan *destination address.* Target ini bekerja pada chain INPUT, FORWARD dan PREROUTING atau chain buatan yang dipanggil melalui chain tersebut.

Beberapa target yang lain biasanya memerlukan parameter tambahan:

a.LOG Target

Ada beberapa option yang bisa digunakan bersamaan dengan target ini. Yang pertama adalah yang digunakan untuk menentukan tingkat log. Tingkatan log yang bisa digunakan adalah debug, info, notice, warning, err, crit, alert dan emerg.Yang kedua adalah -j LOG --log-prefix yang digunakan untuk memberikan string yang tertulis pada awalan log, sehingga memudahkan pembacaan log tersebut.

iptables –A FORWARD –p tcp –j LOG --log-level debug

iptables –A INPUT –p tcp –j LOG --log-prefix “INPUT Packets”

b. REJECT Target

Secara umum, REJECT bekerja seperti DROP, yaitu memblok paket dan menolak untuk memproses lebih lanjut paket tersebut. Tetapi, REJECT akan mengirimkan error message ke host pengirim paket tersebut. REJECT bekerja pada chain INPUT, OUTPUT dan FORWARD atau pada chain tambahan yang dipanggil dari ketiga chain tersebut.

iptables –A FORWARD –p tcp –dport 22 –j REJECT --reject-with icmp-host-unreachable

Ada beberapa tipe pesan yang bisa dikirimkan yaitu icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unrachable, icmp-net-prohibited dan icmp-host-prohibited.

c. **SNAT Target**

Target ini berguna untuk melakukan perubahan alamat asal dari paket (Source Network Address Translation). Target ini berlaku untuk tabel nat pada chain POSTROUTING, dan hanya di sinilah SNAT bisa dilakukan. Jika paket pertama dari sebuah koneksi mengalami

SNAT, maka paket-paket berikutnya dalam koneksi tersebut juga akan mengalami hal yang sama.

```
iptables –t nat –A POSTROUTING –o eth0 –j SNAT --to-source
194.236.50.155-194.236.50.160:1024-32000
```

d. DNAT Target

Berkebalikan dengan SNAT, DNAT digunakan untuk melakukan translasi field alamat tujuan (Destination Network Address Translation) pada header dari paket-paket yang memenuhi kriteria match. DNAT hanya bekerja untuk tabel nat pada chain PREROUTING dan OUTPUT atau chain buatan yang dipanggil oleh kedua chain tersebut.

```
iptables –t nat –A PREROUTING –p tcp –d 15.45.23.67 --dport 80 –j DNAT --
to-destination 192.168.0.2
```

e. MASQUERADE Target

Secara umum, target MASQUERADE bekerja dengan cara yang hampir sama seperti target SNAT, tetapi target ini tidak memerlukan option --to-source. MASQUERADE memang didesain untuk bekerja pada komputer dengan koneksi yang tidak tetap seperti dial-up atau DHCP yang akan memberi pada kita nomor IP yang berubah-ubah.

Seperti halnya pada SNAT, target ini hanya bekerja untuk tabel nat pada chain POSTROUTING.

```
iptables –t nat –A POSTROUTING –o ppp0 –j MASQUERADE
```

f. REDIRECT Target

Target REDIRECT digunakan untuk mengalihkan jurusan (redirect) paket ke mesin itu sendiri. Target ini umumnya digunakan untuk mengarahkan paket yang menuju suatu port tertentu untuk memasuki suatu aplikasi proxy, lebih jauh lagi hal ini sangat berguna untuk membangun sebuah sistem jaringan yang menggunakan transparent proxy. Contohnya kita ingin mengalihkan semua koneksi yang menuju port http untuk memasuki aplikasi http proxy misalnya squid. Target ini hanya bekerja untuk tabel nat pada chain PREROUTING dan OUTPUT atau pada chain buatan yang dipanggil dari kedua chain tersebut.

```
iptables –t nat –A PREROUTING –i eth1 –p tcp --dport 80 –j REDIRECT --to-port
3128
```

Perintah Dasar

Perintah dasar yang sering digunakan dalam linux adalah sebagai berikut :

1. ls : melihat isi direktori yang aktif

   Contoh : debian:/# cd /root

   debian :/root# ls → menampilkan isi direktori root

2. cat : melihat isi file secara keseluruhan

   Contoh : debian:/# cat [nama_file]

3. more : menampilkan isi file per layer

   Contoh : debian:/#more [nama_file]

4. tail : menampilkan sepuluh baris terakhir

   Contoh : debian:/#tail [nama_file]

5. less : melihat isi file tetapi yang bisa discroll

   Contoh : debian:/#less [nama_file]

6. cp : mengkopi file

   Contoh : debian:/#cp file1 /home → mengkopi file1 dari root ke direktori home

7. mv : memindahkan file

   Contoh : debian:/#mv file1 /home

8. rm : menghapus file

   Contoh : debian:/#rm [nama_file]

9. mkdir : membuat direktori

   contoh: debian:/#mkdir [nama_direktori]

10. rmdir : menghapus direktori

    contoh: debian:/#rmdir [nama_direktori]

11. cd : pindah direktori

    contoh: debian:/#cd root → pindah ke direktori root

III. Filesystem Hierarchy Standard

Filesystem Hierarchy Standard (FHS) adalah standar yang digunakan oleh perangkat lunak dan pengguna untuk mengetahui lokasi dari file atau direktori yang berada pada komputer. Hal ini dilakukan dengan cara menetapkan prinsip-prinsip dasar pada setiap daerah pada sistem file, menetapkan file dan direktori minimum yang dibutuhkan, mengatur banyaknya pengecualian dan mengatur kasus yang sebelumnya pernah mengalami konflik secara spesifik.

Dokumen FHS ini digunakan oleh pembuat perangkat lunak untuk menciptakan suatu aplikasi yang compliant dengan FHS. Selain itu, dokumen ini juga digunakan oleh para pembuat sistem operasi untuk menyediakan sistem yang compliant dengan FHS.

Direktori	Deskripsi
/etc	Berisi file administrative (konfigurasi dll) dan file executable atau script yang berguna untuk administrasi system.
/dev	Berisi file khusus yang merepresentasikan peralatan hardware seperti memori, disk, printer, tape, floppy, jaringan dll.
/bin	Berisi program standar Linux (binary).
/sbin	Berisi perintah-perintah yang berhubungan dengan dengan system (hanya super user).
/lib	Berisi program library yang diperlukan untuk kompilasi program (misalnya C). Berisi instruksi (command) misalnya untuk Print Spooler (lpadmin) dll.
/tmp	Berisi file sementara, yang pada saat Bootstrap akan dihapus
/boot	Berisi file yang sangat penting untuk proses bootstrap. Kernel vmlinuz disimpan di direktori ini.
/proc	Berisi informasi tentang kernel Linux, proses dan virtual system file.
/var	Direktori variable, artinya tempan penyimpanan LOG (catatan hasil output program), file ini dapat membengkak dan perlu dimonitor perkembangannya.
/home	Berisi direktori untuk pemakai Linux (pada SCO diletakkan pada /usr)
/mnt	Direktori untuk mounting system file
/root	Home direktori untuk superuser (root)
/usr/bin/X11	Symbolic link ke /usr/X11R6/bin, program untuk X-Window
/usr/src	Source code untuk Linux
/opt	Option, direktori ini biasanya berisi aplikasi tambahan (“add-on”) seperti Netscape Navigator, kde, gnome, applix dll.
/usr	Berisi subdirectory yang bisa di execute oleh semua user
/sys	Berisi system, driver-driver yang aktif dan lebih tertata
lost+found	Berisi informasi jika kita melakukan command fsck

CATATAN