

Apa Itu Server ?

Server adalah ibarat pelayan yang memiliki hak untuk mengatur. Kenapa server disebut sebagai pelayan ? server disebut sebagai pelayan dikarenakan fungsi server secara keseluruhan adalah memberi layanan (service) kepada client yang saling terhubung satu sama lain dalam satu jaringan.

Sedangkan fungsi server dalam mengatur adalah bagaimana server mengatur dalam memberi hak akses terhadap client yang terhubung dengan server tersebut. Contohnya hak akses internet, akses directory, dll

Selain itu server dapat berfungsi sebagai dinding keamanan (firewall). Fungsi server ini sangat penting dalam jaringan yang terhubung dengan jaringan luar seperti internet. Server dapat berfungsi untuk membatasi dan menolak suatu koneksi yang ingin merusak dan melakukan pencurian metadata.

Server dapat pula berfungsi sekaligus sebagai router yang menghubungkan antara sebuah jaringan dengan jaringan yang lain tapi berbeda segmen.

Layanan yang diberikan server kepada client bermacam-macam. Layanan tersebut dapat juga berupa service E-Mail, Domain, Web, Proxy, dll.

MEMBANGUN SERVER



➤ Pendahuluan

Setiap kali anda menggunakan internet dalam kegiatan anda sehari-hari, maka setiap kali itu pula secara tidak langsung anda menggunakan DNS (Domain Name System). Penggunaan DNS meliputi aplikasi email (electronic-mail), browsing, ssh/telnet, ftp, maupun aplikasi yang lain yang ada kaitannya dengan internet. Fungsi utama dari sebuah sistem DNS adalah menerjemahkan nama-nama host (hostnames) menjadi nomor IP (IP address) ataupun sebaliknya, sehingga nama tersebut mudah diingat oleh pengguna internet. Fungsi lainnya adalah untuk memberikan suatu informasi tentang suatu host ke seluruh jaringan internet. Sebelum kita mulai instalasi BIND kita flashback dulu dengan sejarah internet, DNS dan BIND, serta konsep DNS.

Sejarah Internet Pada akhir tahun 1960, US. Department of Defense Advanced Research Projects Administration (ARPA/DARPA) mendanai percobaan dan riset tentang jaringan komputer secara luas yang saling menghubungkan antar hampir semua organisasi di Amerika yang akhirnya dikenal dengan sebutan

ARPAnet. Hasil dari riset tersebut email (electronic-mail) mulai digunakan. Pada awal tahun 1980 protokol TCP/IP (Transmission Control Protocol/Internet Protocol) mulai dikenalkan dan akhirnya menjadi protokol standar dalam struktur jaringan ARPAnet. Jaringan ARPAnet berkembang pesat jumlahnya menjadi ribuan host dan masih menggunakan standar protokol TCP/IP, dan akhirnya jaringan tersebut dikenal dengan internet.

Pada tahun 1988, DARPA digantikan oleh National Science Foundation (NSF) dalam pendanaan riset diikuti dengan penggantian dari ARPAnet menjadi NSFnet sebagai tulang punggung (backbone) jaringan internet. Kemudian pada musim semi tahun 1995, backbone internet melakukan transisi dari NSFnet (yang didanai oleh

publik) ke beberapa backbone komersil, dimana memungkinkan interkoneksi antar jaringan bisa menjadi lebih jauh jaraknya. Penyedia backbone komersil tersebut diantaranya adalah MCI dan Sprint serta pemain lama seperti UUNet dan PSINet.

Sejarah DNS Pada tahun 1970an jaringan ARPAnet hanya terdiri dari beberapa ratus host saja. Pada waktu itu, sebuah file HOSTS.TXT yang berisi tentang semua informasi host-hosts tersebut masih bisa melayani setiap permintaan query dan menerjemahkan nama ke alamat IP (name-to-address-mapping). Pada sistem operasi berbasis UNIX, file /etc/hosts merupakan hasil dari pengolahan file HOSTS.TXT tersebut.

File HOSTS.TXT pada waktu itu dikelola oleh Stanford Research Institute Network Information Center (SRINIC) di Menlo Park, California. File tersebut tersebut didistribusikan ke semua host dan penggunaanya hanya 2 dengan menggunakan satu buah host (mesin/komputer) saja. Petugas administrasi dari ARPAnet biasanya mengirimkan email kepada SRI-NIC tentang perubahan (termasuk penambahan maupun pengurangan) tentang informasi suatu host, dan dalam periode tertentu, mereka melakukan transfer file HOSTS.TXT yang paling baru (biasanya diperbaharui sekali dalam seminggu) dengan menggunakan protokol ftp. Seiring dengan berkembangnya jaringan ARPAnet dan penggunaan protokol TCP/IP, ukuran dari file HOSTS.TXT menjadi besar dengan bertambahnya jumlah host yang bergabung dengan jaringan ARPAnet. Kemudian timbul beberapa masalah dengan penggunaan file HOSTS.TXT ini, misalnya :

- Trafik dan Beban (Traffic and load) Beban mesin dan trafik (bandwidth) di SRI-NIC dalam mendistribusikan file menjadi lebih berat dan besar
- Penamaan yang saling bentrok (name collisions) Pada file HOSTS.TXT tidak diperkenankan adanya dua buah nama host yang sama. Namun pada prakteknya, tidak ada cara untuk mencegah seseorang untuk menambahkan nama yang sama sehingga kemungkinan bisa menjadi bentrok dan pada akhirnya merusak skema yang telah ada
- Keaslian (consistency) Mengelola keaslian dan keutuhan sebuah file antar beberapa jaringan yang sedang berkembang pesat merupakan sesuatu hal yang sulit dilakukan Berangkat dari masalah-masalah tersebut diatas, ARPAnet membentuk suatu sistem alternatif pengganti dari sistem lama yang menggunakan

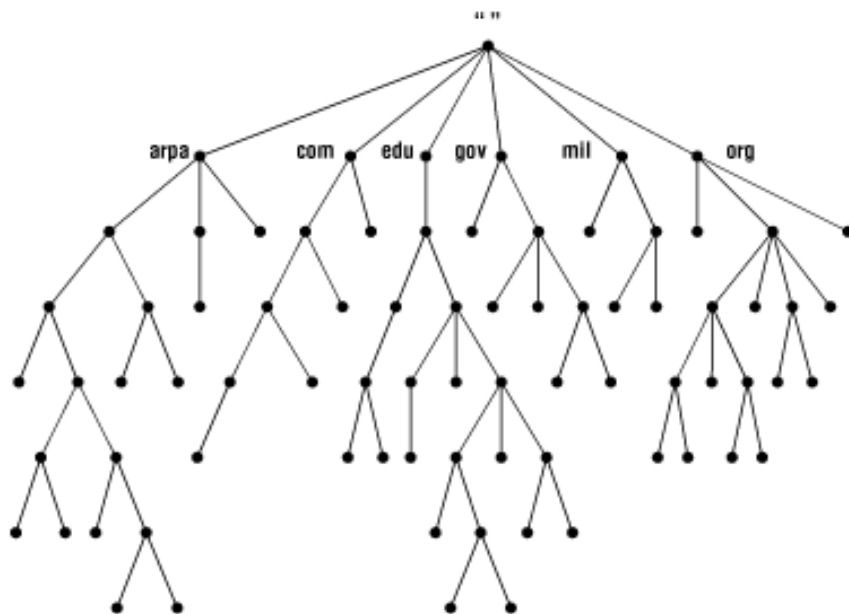
file HOSTS.TXT. Tujuannya adalah untuk memecahkan masalah dalam pengelolaan tabel host yang sangat beraneka ragam dan masih menggunakan metode sentralisasi. Pada sistem yang baru, seorang sistem administrator memungkinkan untuk mengelola data secara lokal, namun akan selalu update secara global di internet. Sistem yang menggunakan metode desentralisasi ini diharapkan akan mengurangi beban dan trafik, serta pengelolaan data dan proses update dari sebuah informasi akan menjadi lebih mudah.

Paul Mockapertis dari University of Southern California Information Science Institute di Marina del Rey, California, dipilih sebagai orang yang bertanggung jawab terhadap rancangan, desain, arsitektur dan implementasi dari sistem pengelolaan data host yang baru. Pada tahun 1984 beliau merilis RFC (Request For Comment) 882 dan RFC 883 yang menjelaskan tentang Domain Name System (DNS). Kemudian disusul dengan RFC 1034 dan RFC 1035 yang juga menambahkan tentang masalah keamanan DNS, penerapan (implementasi), pengelolaan (administrative), mekanisme pembaharuan data secara dinamis, serta keamanan data dalam sebuah domain dan lain-lainnya.

(DNS and BIND 3rd Edition, Paul Albitz & Cricket Liu, 1998 Hal 3)

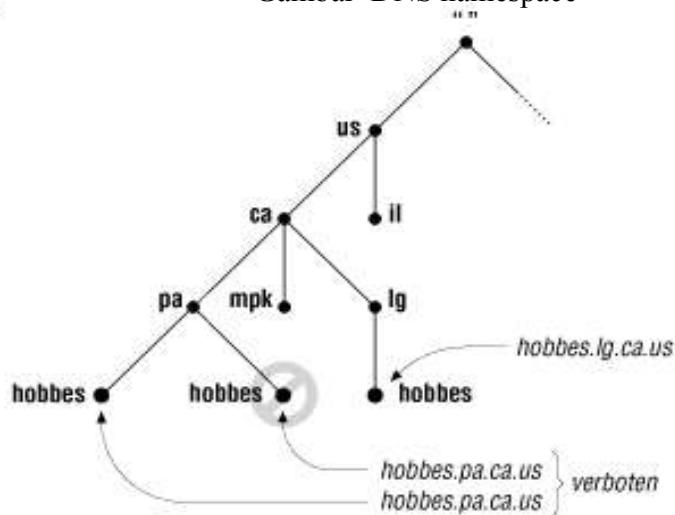
Konsep dan hirarki DNS DNS adalah suatu bentuk database yang terdistribusi, dimana pengelolaan secara lokal terhadap suatu data akan segera diteruskan ke seluruh jaringan (internet) dengan menggunakan skema client-server. Suatu program yang dinamakan name server, mengandung semua segmen informasi dari database dan juga merupakan resolver bagi client-client yang berhubungan ataupun menggunakannya.

Struktur dari database DNS bisa diibaratkan dengan dengan struktur file dari sebuah sistem operasi UNIX. Seluruh database digambarkan sebagai sebuah struktur terbalik dari sebuah pohon (tree) dimana pada puncaknya disebut dengan root node. Pada setiap node dalam tree tersebut mempunyai keterangan (label) misalnya, .org, .com, .edu, .net, .id dan lain-lainnya, yang relatif terhadap puncaknya (parent). Ini bisa diibaratkan dengan relative pathname pada sistem file UNIX, seperti direktori bin, usr, var, etc dan lain sebagainya. Pada puncak root node dalam sebuah sistem DNS dinotasikan dengan "." atau "/" pada sistem file UNIX.

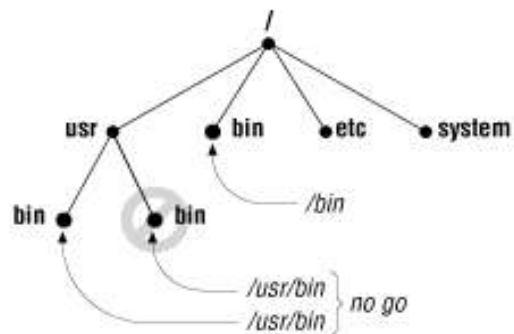


DNS database

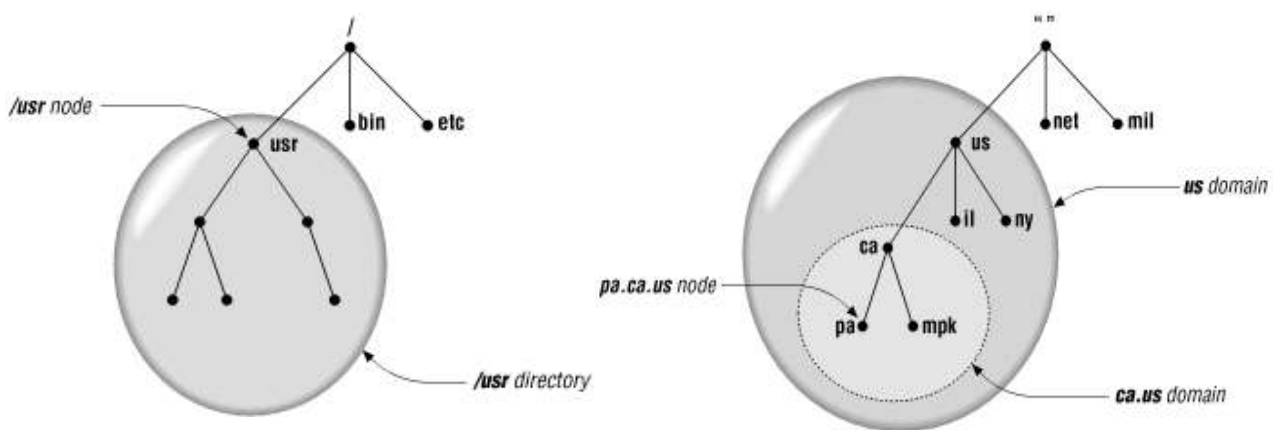
Gambar DNS namespace



Unix filesystem



Pada setiap node juga merupakan root dari subtree, atau pada sistem file UNIX merupakan root direktori dari sebuah direktori. Hal ini pada sistem DNS disebut dengan nama domain. Pada tiap domain juga memungkinkan nama subtree dan bisa berbeda pula, hal ini disebut subdomain atau subdirektori pada sistem file UNIX. Pada bagian subdomain juga memungkinkan adanya subtree lagi yang bisa dikelola oleh organisasi yang berbeda dengan domain utamanya.



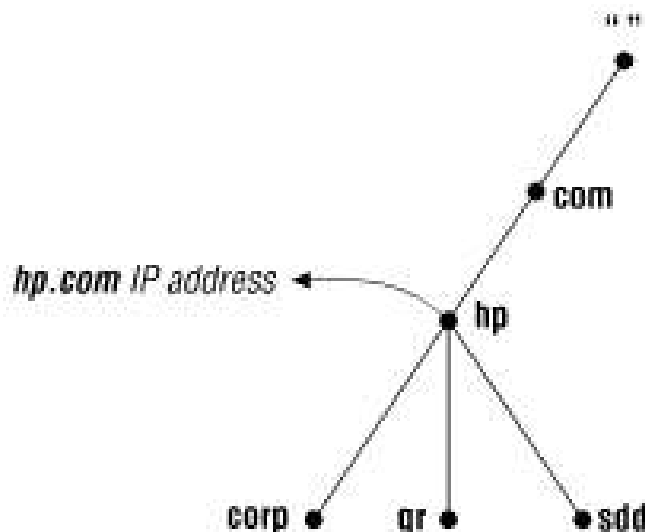
(DNS and BIND 3rd Edition, Paul Albitz & Cricket Liu, 1998 Hal 4)

Sejarah BIND Program DNS yang bernama JEEVES pertama kali diimplementasikan dan ditulis sendiri oleh Paul Mockapertis. Kemudian diteruskan oleh BIND (versi 4.8.3) yang diimplementasikan pada sistem operasi 4.3 BSD UNIX yang ditulis oleh Douglas Terry, Mark Painter, David Riggle dan Songnian Zhou dari Computer Systems Research Group (CSRG) pada Universitas California di Berkeley. Pada tahun antara 1985-1987, Kevin Dunlap seseorang dari Digital Equipment Corporation (DEC) bergabung dengan CSRG yang kemudian diikuti oleh Doug Kingston, Craig Partridge, Smoot Carl- Mitchell, Mike Muuss, Jim Bloom dan Mike Schwartz. Pemimpin dari proyek ini adalah Mike Karels dan O. Kure.

BIND versi 4.9 dan 4.9.1 kemudian dirilis oleh DEC (yang sekarang diakusisi oleh Compaq Computer Corporation). Pemimpin dari proyek ini adalah Paul Vixie yang merupakan karyawan dari DEC serta dibantu oleh Phil Almquist, Robert Elz, Alan Barrett, Paul Albitz, Bryan Beecher, Andrew Partan, Andy Cherenon, Tom Limoncelli, Berthold Paffrath, Fuat Baran, Anant Kumar, Art Harkin, Win Treese, Don

Lewis, Christophe Wolfhugel, dan lain-lainnya. BIND versi 4.9.2 kemudian diambil alih oleh Vixie Enterprises, dan Paul Vixie menjadi arsitek dan programmernya. BIND mulai dari versi 4.9.3 dan seterusnya kemudian diambil alih oleh Internet Software Consortium (ISC) dan akhirnya untuk pertama kalinya, pada tanggal 8 Mei 1997 Bob Halley dan Paul Vixie merilis versi BIND untuk keperluan produksi. Sekarang BIND versi 4 sudah mulai jarang digunakan, dan sebagai penggantinya adalah BIND versi 8 dan versi 9. (<http://www.isc.org/products/BIND/bind-history.html>)

Cara kerja DNS Ketika anda melakukan query (bisa berupa ping, ssh, dig, host, nslookup, email, dan lain sebagainya) ke sebuah host misalnya local.bpgupg.go.id maka name server akan memeriksa terlebih dahulu apakah ada record host tersebut di cache name server lokal. Jika tidak ada, name server lokal akan melakukan query kepada root server dan mereferensikan name server untuk TLD .edu , name server lokal kembali melakukan query kepada name server .edu dengan jenis query yang sama dan mereferensikan local.bpgupg.go.id . Name server lokal kembali melakukan query ke name server local.bpgupg.go.id dan mereferensikan query selanjutnya ke name server lokal yaitu local.bpgupg.go.id . Kemudian name server lokal melakukan query kepada name server lokal yaitu ee.linux dan akhirnya mendapatkan jawaban address yang diminta.



➤ Instalasi DNS Server

Apabila anda melakukan instalasi linux melalui media instalasi CD, maka pada saat pemilihan paket-paket instalasi anda dapat menginstall sekaligus paket BIND yang dibutuhkan untuk DNS Server. Yaitu dengan memberi tanda centang pada Network Server, Simple Web Server, LDAP Server and Tools. Perhatikan kembali proses instalasi pada Section Software. Jadi pada bagian ini tidak dijelaskan bagaimana cara menginstall BIND. Dikarenakan linux yang dinstall telah menginstall paket-paket BIND.

➤ Instalasi DNS Server

Secara Default setelah linux anda terinstall maka direktori yang perlu anda edit adalah `/etc/named.conf` , `/etc/resolv.conf` , `/var/lib/named` . contoh kali ini kita akan membuat server dengan nama domain **linux.net** dan nama host adalah **server**. Domain dan host berkaitan erat pada pembuatan DNS Server. Sekarang kita akan memulai pembuatan DNS Server.

Masuklah kedalam direktori `/var/lib/named` dan lihat isi direktori tersebut.

```
root:~# cd /var/lib/named
root:/var/lib/named# ls
```

Setelah anda menjalankan perintah `ls` maka linux akan menampilkan isi default direktori `/var/lib/named` yaitu :

```
root:/var/lib/named# ls
.      ..      slave      127.0.0.zone      localhost.zone      root.hint
```

Salinlah file `127.0.0.zone` didalam direktori `/var/lib/named/` dengan nama file `192.168.0.zone` . file ini sesuai dengan segmen IP yang disetting pada Ethernet card yaitu `192.168.0.1`

```
root:/var/lib/named# cp 127.0.0.zone 192.168.0.zone /var/lib/named
```

Setelah anda menyalin file `127.0.0.zone` maka editlah `192.168.0.zone` dengan menggunakan perintah `vi` atau `pico`

```
root:/var/lib/named# pico 192.168.0.zone
```

Isi file tersebut masih sama dengan `127.0.0.zone` karena file `192.168.0.zone` disalin dari file `127.0.0.zone`. maka anda harus mengedit file `192.168.0.zone` dengan isi seperti berikut.


```

$TTL 86400
@ IN SOA server.linux.net. root.linux.net. (
                                11 ; serial
                                28800 ; refresh
                                14400 ; retry
                                3600000 ; expiry
                                86400 ) ; minimum

      IN NS      linux.net.
      IN NS      server.linux.net.
1     IN PTR     server.linux.net.

```

Lalu save file tersebut dan keluarlah dari text editor pico. Salinlah file 192.168.0.zone didalam direktori /var/lib/named/ dengan nama file db/linux.net. file

```
root:/var/lib/named# cp 192.168.0.zone db.linux.net /var/lib/named
```

Setelah anda menyalin file 192.168.0.zone maka editlah db.linux.net dengan menggunakan perintah vi atau pico

```
root:/var/lib/named# pico db.linux.net
```

Isi file dari db.linux.net adalah :

```

$TTL 86400
@ IN SOA server.linux.net. root.linux.net. (
                                12 ; serial
                                28800 ; refresh
                                14400 ; retry
                                3600000 ; expiry
                                86400 ) ; minimum

      IN NS      linux.net.
      IN NS      server.linux.net.
1     IN PTR     server.linux.net.
      IN MX      10  server.linux.net.
localhost IN A      127.0.0.1
server    IN A      192.168.0.1
www       IN CNAME   server.linux.net.
mail      IN CNAME   server.linux.net.
smtp      IN CNAME   server.linux.net.
pop3      IN CNAME   server.linux.net.
ftp       IN CNAME   server.linux.net.

```

Isi file tersebut masih sama dengan 127.0.0.zone karena file 192.168.0.zone disalin dari file 127.0.0.zone. maka anda harus mengedit file 192.168.0.zone dengan isi seperti berikut.

Simpan file diatas lalu keluar dari text editor pico. Pada file diatas terdapat resources record yaitu IN NS, IN PTR, IN MX, IN A, IN CNAME. Maksud record db file tersebut adalah :

SOA record

Mengindikasikan otoritas dari sebuah zone file

NS record

Mengindikasikan daftar nama server

Other records

Menjelaskan mengenai data dari zone file tersebut, Record ini berupa :

A

Mengubah Nama menjadi Alamat IP

PTR

Mengubah Nama IP menjadi Nama

CNAME

Menjelaskan nama Alias

Setelah itu editlah file */etc/named.conf*

```
root:/var/lib/named# cd /etc/  
root:/var/lib/named# pico named.conf
```

Default dari isi file *named.conf* sebelum diedit adalah :

```
# Copyright (c) 2001 SuSE GmbH Nuernberg, Germany  
#  
# Author: Frank Bodammer <feedback@suse.de>  
#  
# /etc/named.conf  
#  
# This is a sample configuration file for the name server  
# BIND9.  
# It works as a caching only name server without  
# modification.  
#  
# A sample configuration for setting up your own domain can  
# be found in /usr/share/doc/packages/bind8/sample-config.  
#  
# A description of all available options can be found in  
# /usr/share/doc/packages/bind8/html/options.html  
  
options {  
  
    # The directory statement defines the name server's  
    # working directory  
  
    directory "/var/lib/named";  
  
    # The forwarders record contains a list of servers to  
    # which queries should be forwarded. Enable this line  
    # and  
    # modify the IP-address to your provider's name server.  
    # Up to three servers may be listed.  
  
    #forwarders { 10.11.12.13; 10.11.12.14; };  
  
    # Enable the next entry to prefer usage of the name
```

```

# server declared in the forwarders section.

#forward first;

# The listen-on record contains a list of local network
# interfaces to listen on. Optionally the port can be
# specified. Default is to listen on all interfaces
# found
# on your system. The default port is 53.

#listen-on port 53 { 127.0.0.1; };

# The next statement may be needed if a firewall stands
# between the local server and the internet.

#query-source address * port 53;

# The allow-query record contains a list of networks or
# IP-addresses to accept and deny queries from. The
# default is to allow queries from all hosts.

#allow-query { 127.0.0.1; };

# The cleaning-interval statement defines the time
# interval
# in minutes for periodic cleaning. Default is 60
# minutes.
# By default, all actions are logged to
# /var/log/messages.

cleaning-interval 120;

# Name server statistics will be logged to
# /var/log/messages
# every <statistics-interval> minutes. Default is 60
# minutes.
# A value of 0 disables this feature.

statistics-interval 0;

# If notify is set to yes (default), notify messages
# are
# sent to other name servers when the the zone data is
# changed. Instead of setting a global 'notify'
# statement
# in the 'options' section, a separate 'notify' can be
# added to each zone definition.

```

```

        notify no;
};

# The following three zone definitions don't need any
# modification.
# The first one defines localhost while the second defines
# the
# reverse lookup for localhost. The last zone "." is the
# definition of the root name servers.

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};

# You can insert further zone records for your own domains
below.

```

Edit file named.conf dan tambahkan baris berikut pada bagian akhir file named.conf

```
root:/etc# pico named.conf
```

Bagian yang ditambahkan adalah :

```

zone "0.168.192.in-addr.arpa" in {
    type master;
    file "192.168.0.zone";
};

zone "linux.net" in {
    type master;
    file "db.linux.net";
};

```

Simpan file tersebut, lalu keluar dari editor pico. Sekarang kita edit file resolv.conf pada direktori /etc/resolv.conf

```
root:/etc# pico resolv.conf
```

Sedangkan isi dari file resolv.conf adalah :

```
domain linux.net
nameserver 192.168.0.1
nameserver 127.0.0.1
search linux.net
```

Setelah semuanya telah selesai maka jalankan service named

```
root:/etc# /etc/init.d/named start
```

Atau anda bias menggunakan perintah.

```
root:/etc# rcnamed start
```

Lalu anda bisa melihat apakah hasil konfigurasi DNS server anda telah berjalan apa tidak jalankan perintah dibawah ini untuk melihat zone file yang error.

```
root:/etc# less /var/log/messages
Feb 15 12:54:18 named[25139]: starting BIND 9.2.2rc1 -u named -c
/usr/local/named/etc/named.conf
Feb 15 12:54:18 named[25139]: using 1 CPU
Feb 15 12:54:18 named[25139]: loading configuration from
'/usr/local/named/etc/named.conf'
Feb 15 12:54:18 named[25139]: no IPv6 interfaces found
Feb 15 12:54:18 named[25139]: listening on IPv4 interface lo, 127.0.0.1#53
Feb 15 12:54:18 named[25139]: listening on IPv4 interface eth0, 192.168.0.1#53
Feb 15 12:54:18 named[25139]: command channel listening on 127.0.0.1#53
Feb 15 12:54:18 named[25139]: zone 0.168.192.in-addr.arpa/IN: loaded serial 9
Feb 15 12:54:18 named[25139]: zone 0.0.127.in-addr.arpa/IN: loaded serial 10
Feb 15 12:54:18 named[25139]: zone localhost/IN: loaded serial 11
Feb 15 12:54:18 named[25139]: zone db.linux.net/IN: loaded serial 12
Feb 15 12:54:18 named[25139]: running
```

Jalankan perintah dig untuk mengecek apa server DNS yang anda buat sudah berjalan dengan baik apa tidak.

```
root:/etc# dig -x 192.168.0.1
; <<>> DiG 9.2.2rc1 <<>> -x 192.168.0.1
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30843
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;0.168.192.in-addr.arpa. IN PTR

;; ANSWER SECTION:
0.168.192.in-addr.arpa. 86400 IN PTR server.linux.net.

;; AUTHORITY SECTION:
0.168.192.in-addr.arpa. 86400 IN NS server.linux.net.

;; ADDITIONAL SECTION:
server.linux.net. 86400 IN A 192.168.0.1
;; Query time: 1 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
```