

# DOMAIN NAME SYSTEM

**Oleh**  
**Tim Network Administrator PENS ITS**



**Politeknik Elektronikan Negeri Surabaya**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya**

# INTRO TO DNS

- Merupakan sistem berbentuk database terdistribusi
- Memetakan/mengkonversikan nama host/mesin/domain ke alamat IP (Internet Protocol) dan sebaliknya dari alamat IP ke nama host yang disebut dengan reverse-mapping.

- ❑ Untuk memetakan nama mesin misal [www.eeplis-its.edu](http://www.eeplis-its.edu) ke alamat IP misal 202.154.187.7
- ❑ Untuk routing e-mail, telnet, ftp, web, dan lain-lain.

# INTRO TO DNS

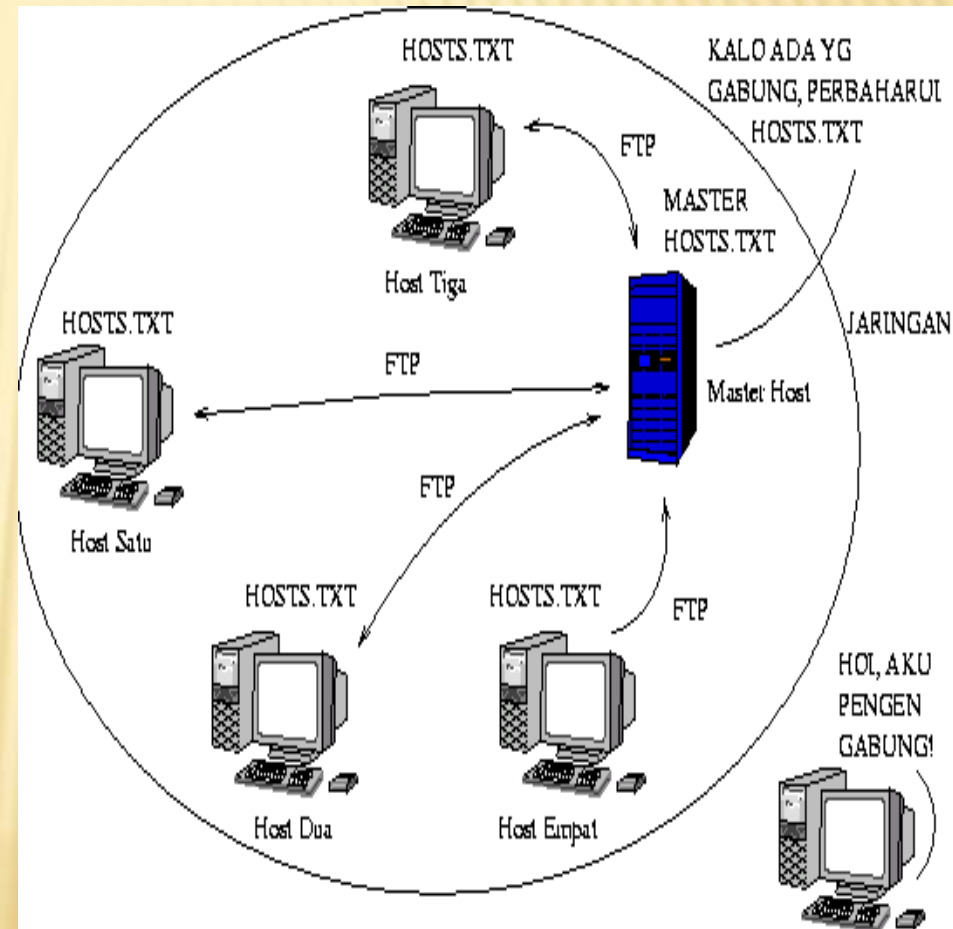
## ✗ DNS sebagai jembatan

- Manusia lebih mudah untuk mengingat nama daripada alamat IP dengan panjang 32 bit itu.
- Komputer menggunakan alamat IP untuk berkomunikasi dan berinteraksi.

## ✗ DNS tidak diperlukan jika kita bisa mengingat ratusan, ribuan, bahkan jutaan alamat IP di Internet.

# HISTORY

- ✗ Sebelum adanya DNS, tahun 1970-an ARPAnet menggunakan pemetaan dengan bentuk tabel host pada berkas HOSTS.TXT
- ✗ HOSTS.TXT berisi nama host dan alamat IP serta pemetaannya dari seluruh mesin/komputer yang terhubung dalam jaringan.
- ✗ Ketika ada komputer lain yang terhubung ke jaringan ARPAnet maka masing-masing komputer dalam jaringan tersebut harus memperbaharui berkas HOSTS.TXT-nya.
- ✗ Cara meng-update berkas HOSTS.TXT dengan menggunakan ftp setiap satu atau dua minggu sekali.
- ✗ Masalah ketika jaringan menjadi semakin besar. Kesulitan meng-update isi berkas HOSTS.TXT karena jumlah nama mesin/komputer yang dituliskan sudah terlalu besar dan tidak efisien.



# HISTORY

Muncul ide untuk membuat sistem database terdistribusi yang mempunyai data mengenai pemetaan nama host ke alamat IP dan sebaliknya.

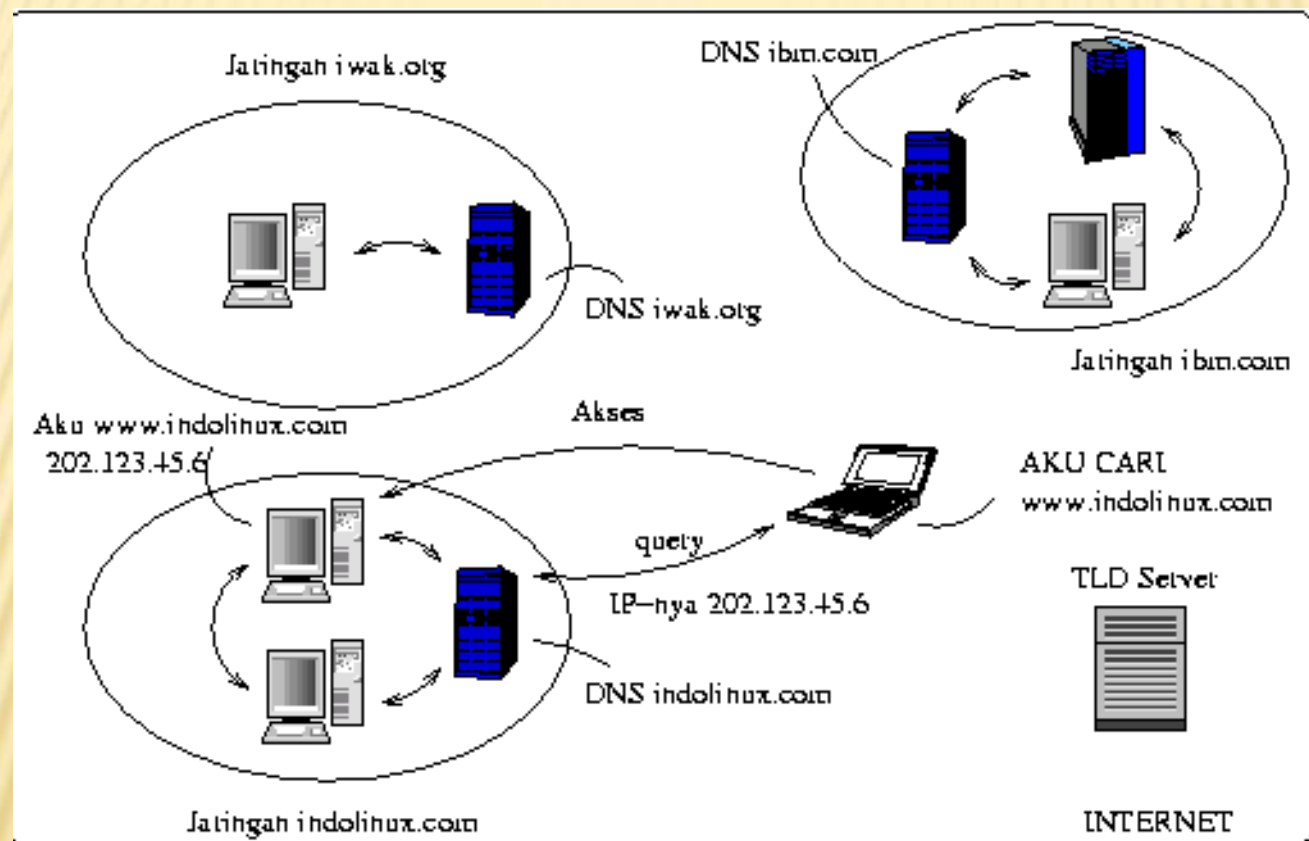
Dengan adanya pendistribusian database nama host dan alamat IP, maka tiap organisasi yang memiliki jaringan di dalam domain tertentu hanya bertanggung jawab terhadap database informasi pemetaan nama host dan alamat IP pada jaringannya saja yang biasa disebut zone.

Administrasi domain tersebut dilakukan secara lokal tetapi informasi itu dapat diakses oleh semua komputer di Internet.

Karena sifat database yang terdistribusi ini, maka dibutuhkan suatu mekanisme pengaksesan informasi bagi host lain pada database yang terdistribusi untuk menemukan informasi host atau jaringan yang dipunyai oleh suatu organisasi.

Dan pada tahun 1984, Paul Mockapetris mengusulkan sistem database terdistribusi ini dengan Domain Name System (DNS) yang dideskripsikan dalam RFC 882 dan 883. Sistem ini digunakan sampai sekarang pada jaringan khususnya Internet.

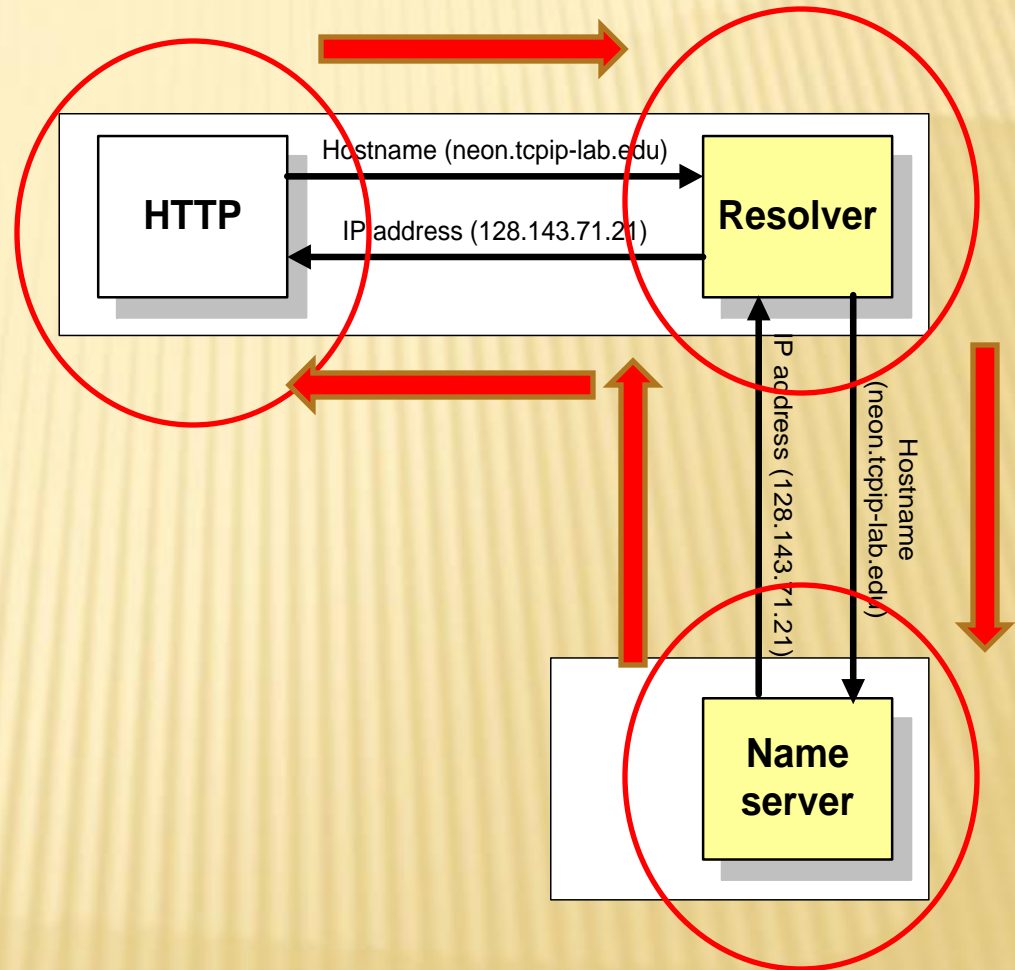
# HISTORY





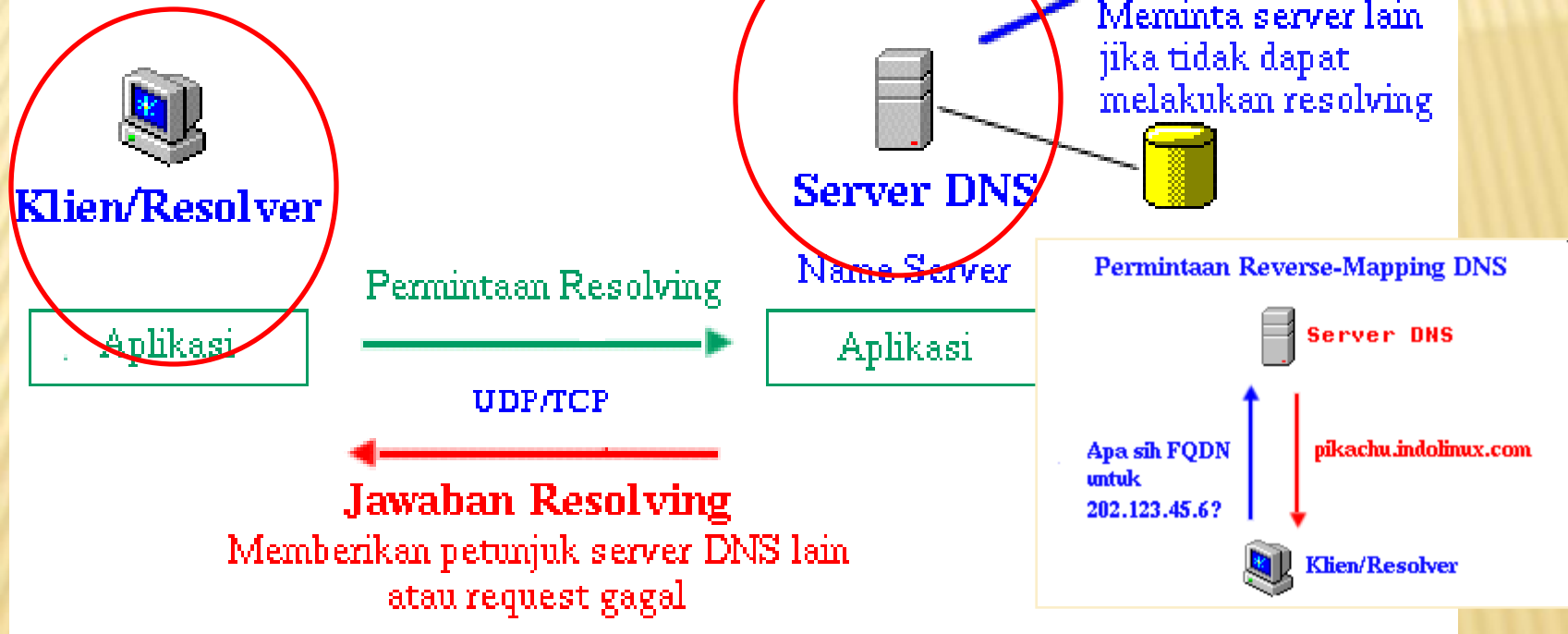
# RESOLVER AND NAME SERVER

1. Sebuah program aplikasi pada host yang mengakses domain system disebut sebagai **resolver**
  2. Resolver mengontak DNS server, yang biasa disebut name server
  3. DNS server mengembalikan IP address ke resolver yang meneruskan ke aplikasi yang membutuhkan IP address
- ✗ Reverse lookups are also possible, i.e., find the hostname given an IP address



# KERJA DNS

## Bagaimana DNS Bekerja

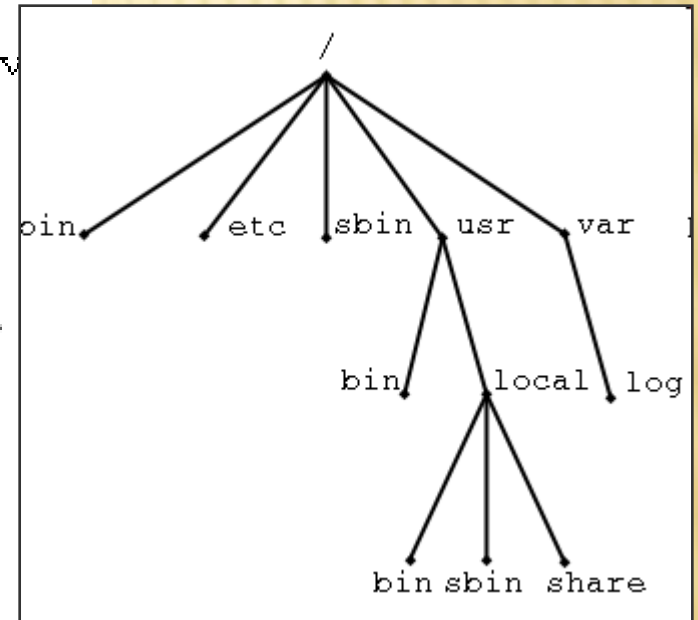
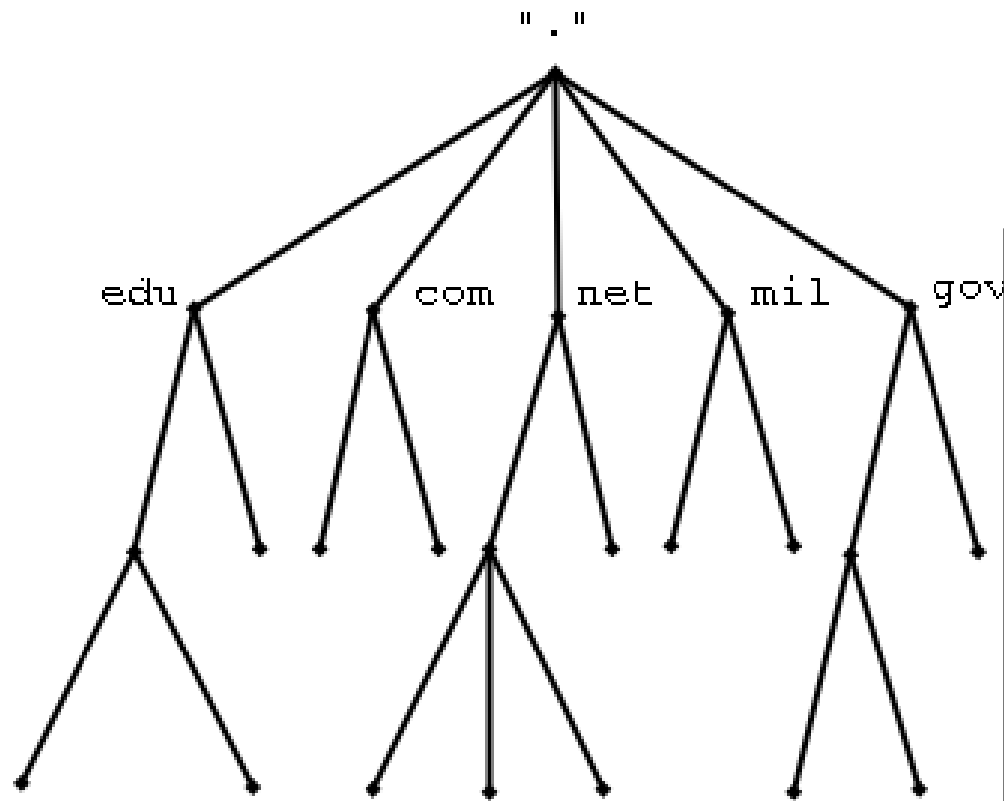




# STRUKTUR

- ✖ Struktur database DNS mirip dengan sistem-berkas/filesystem UNIX yaitu berbentuk hierarki atau pohon.
- ✖ Tingkat teratas pada DNS adalah root yang disimbolkan dengan titik/dot (.) sedangkan pada sistem berkas UNIX, root disimbolkan dengan slash (/).
- ✖ Setiap titik cabang mempunyai label yang mengidentifikasikannya relatif terhadap root (.).
- ✖ Tiap titik cabang merupakan root bagi sub-tree/tingkat bawahnya.
- ✖ Tiap sub-tree merupakan domain dan dibawah domain terdapat sub-tree lagi bernama subdomain.
- ✖ Setiap domain mempunyai nama yang unik dan menunjukkan posisinya pada pohon DNS, pengurutan/penyebutan nama domain secara penuh dimulai dari domain paling bawah menuju ke root (.).
- ✖ Masing-masing nama yang membentuk suatu domain dipisahkan dengan titik/dot (.) dan diakhiri dengan titik yang merupakan nama absolut relatif terhadap root (.).

# STRUKTUR



# STRUKTUR

- ✗ Contoh: `www.its.ac.id`.
- ✗ "." merupakan root domain
- ✗ `id` merupakan Top Level Domain
- ✗ `ac` merupakan Second Level Domain
- ✗ `its` merupakan Third Level Domain
- ✗ `www` merupakan nama komputer/mesin yang bersangkutan
- ✗ Sistem penulisan nama secara absolut dan lengkap ini disebut FQDN (Fully Qualified Domain Name) - `www.its.ac.id`.

# HIRARKI

- ✖ Tiap organisasi yang telah mendaftar ke Network Information Center(NIC) akan mendapatkan nama domain sesuai dengan organisasi tersebut.
- ✖ Nama domain tersebut bisa dibagi menjadi subdomain sesuai kebutuhan organisasi.
  - + Contoh: InterNIC mempunyai semua Top Level Domain termasuk edu,
  - + Lembaga pendidikan PENS akan mendaftarkan nama domain eepis-its.edu (education), maka PENS diberikan/didelegasikan oleh InterNIC untuk mengelola domain eepis-its.edu yang merupakan sub domain dari edu.
  - + PENS dapat membagi lagi domain eepis-its.edu ke beberapa sub domain misal [www.eepis-its.edu](http://www.eepis-its.edu) , [ies.eepis-its.edu](http://ies.eepis-its.edu), [eis.eepis-its.edu](http://eis.eepis-its.edu), [elearning.eepis-its.edu](http://elearning.eepis-its.edu).

# HIRARKI

- ✘ Dengan adanya sistem berbentuk hierarki/pohon ini maka tidak ada nama host yang sama pada domain/subdomain yang sama, karena masing-masing dari node/titik-cabang mempunyai nama unik dan tidak boleh ada yang menyamainya kecuali berbeda sub-tree/sub pohon.
- ✘ Tidak akan ada konflik antar organisasi karena masing-masing organisasi mempunyai domain yang berbeda-beda dan ini diatur oleh InterNIC untuk TLD (Top Level Domain).
- ✘ Kedalaman pohon dibatasi sampai level 127

# TOP LEVEL DOMAIN (TLD)

## ✖ Domain Generik

- + com , net , gov , mil , org , edu , int
- + Selain 7 domain di atas ada lagi 7 domain baru dari ICANN ([www.icann.org](http://www.icann.org)) yaitu: aero, biz , coop , info , museum , name , pro

## ✖ Domain Negara

- + Contoh: id untuk Indonesia, au untuk Australia, uk untuk Inggris, dan lain-lain.
- + Domain negara ini dapat dan umumnya diturunkan lagi ke level-level di bawahnya yang diatur oleh NIC dari masing-masing negara, untuk Indonesia yaitu IDNIC. Contoh level bawah dari id yaitu net.id, co.id, web.id.

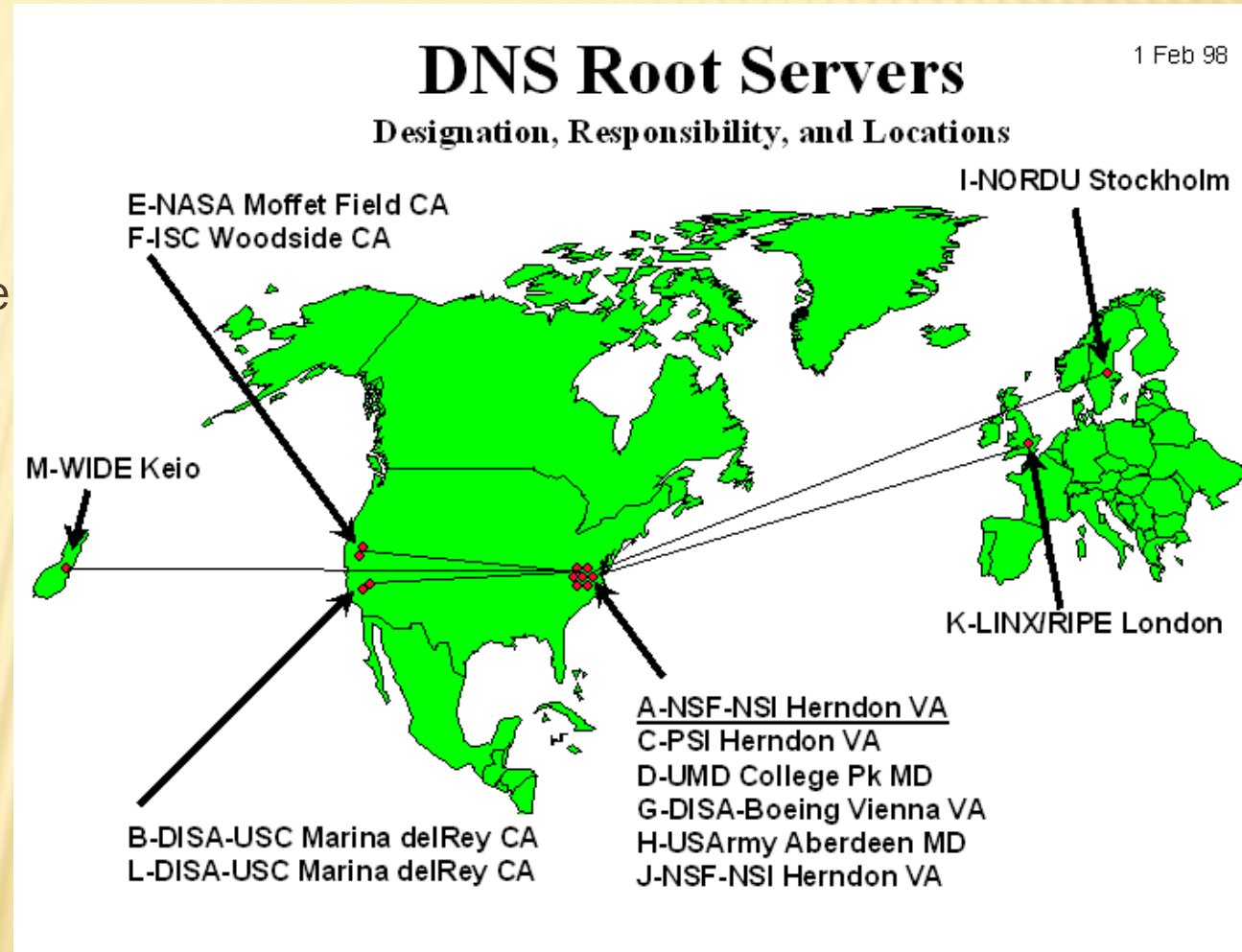
## ✖ Domain Arpa

- + Merupakan domain untuk jaringan ARPAnet. Tiap domain yang tergabung ke Internet berhak memiliki name-space .in-addr.arpa sesuai dengan alamat IP-nya.



# ROOT NAME SERVERS

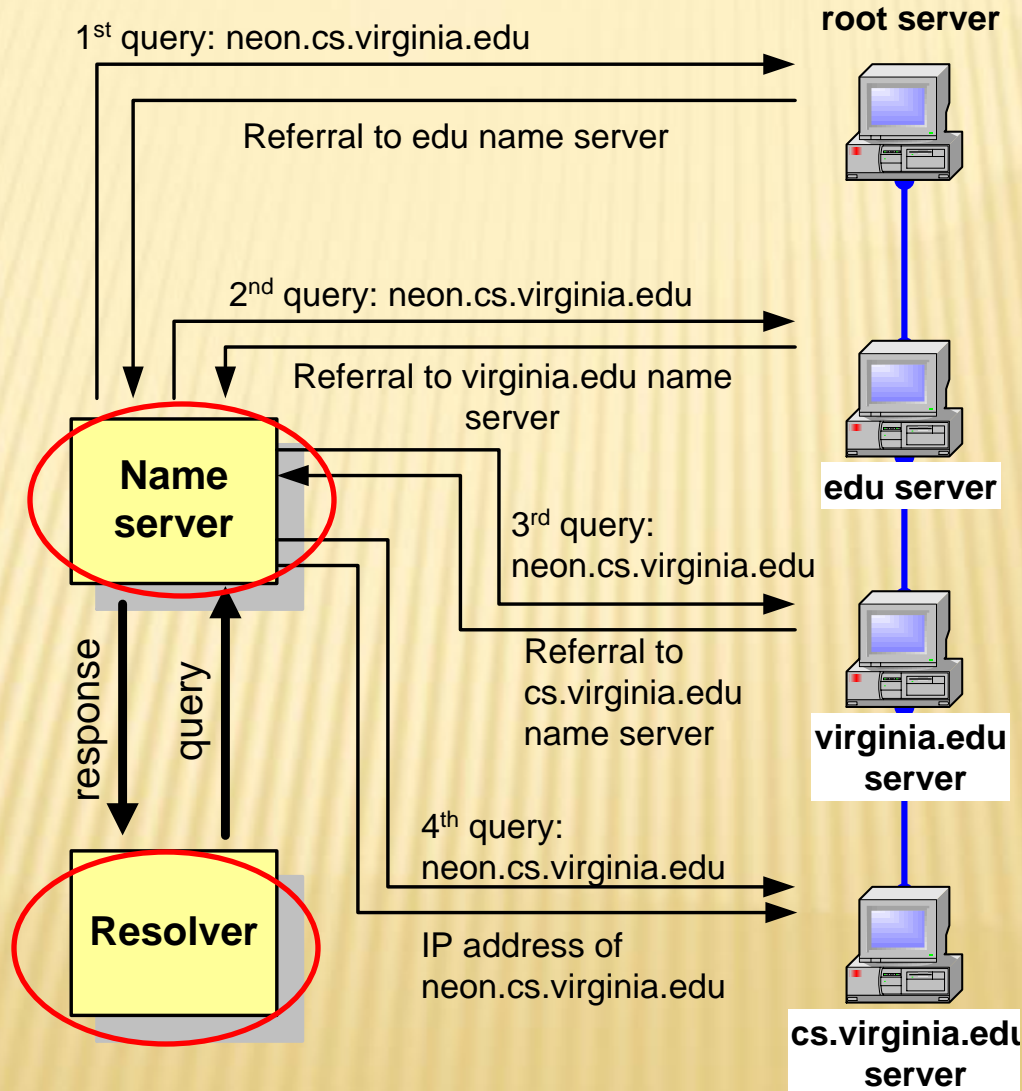
- ✗ Server root digunakan untuk menemukan authoritative name servers untuk semua zona top-level.
- ✗ Ada 13 server root
- ✗ Digunakan untuk name resolution



# ADDRESS ROOT SERVERS (2004)

A.ROOT-SERVERS.NET.	(VeriSign, Dulles, VA)	198.41.0.4
B.ROOT-SERVERS.NET.	(ISI, Marina Del Rey CA)	192.228.79.201
C.ROOT-SERVERS.NET.	(Cogent Communications)	192.33.4.12
D.ROOT-SERVERS.NET.	(University of Maryland)	128.8.10.90
E.ROOT-SERVERS.NET.	(Nasa Ames Research Center)	192.203.230.10
F.ROOT-SERVERS.NET.	(Internet Systems Consortium)	192.5.5.241
G.ROOT-SERVERS.NET.	(US Department of Defense)	192.112.36.4
H.ROOT-SERVERS.NET.	(US Army Research Lab)	128.63.2.53
I.ROOT-SERVERS.NET.	(Autonomica/NORDUnet)	192.36.148.17
J.ROOT-SERVERS.NET.	(Verisign, multiple cities)	192.58.128.30
K.ROOT-SERVERS.NET.	(RIPE, Europe multiple cities)	193.0.14.129
L.ROOT-SERVERS.NET.	(IANA, Los Angeles)	198.32.64.12
M.ROOT-SERVERS.NET.	(WIDE, Tokyo, Seoul, Paris)	202.12.27.33

# RECURSIVE QUERIES



# CACHING

---

- ✖ Penyimpanan informasi mapping antara IP dan name server untuk mengurangi traffic
- ✖ Ketika ada permintaan/ query server tidak perlu lagi menghubungi server lain



# RESOURCE RECORDS

- ✗ Record database pada Database DNS terdistribusi disebut **resource records (RR)**
- ✗ Resource records disimpan pada file konfigurasi (zone files) pada name servers.

Berikut ini contoh sebuah zone Resource record →

**db.mylab.com**

```
$TTL 86400
mylab.com. IN SOA PC4.mylab.com.
                        hostmaster.mylab.com. (
                        1 ; serial
                        28800 ; refresh
                        7200 ; retry
                        604800 ; expire
                        86400 ; ttl
                        )
```

```
;
mylab.com. IN NS PC4.mylab.com.
;
localhost A 127.0.0.1
PC4.mylab.com. A 10.0.1.41
PC3.mylab.com. A 10.0.1.31
PC2.mylab.com. A 10.0.1.21
PC1.mylab.com. A 10.0.1.11
```

# RESOURCE RECORDS

**db.mylab.com**

\$TTL 86400 ←

mylab.com. IN SOA PC4.mylab.com.

hostmaster@mylab.com. (

1 ; serial  
28800 ; refresh  
7200 ; retry  
604800 ; expire  
86400 ; ttl  
)

;

mylab.com. IN NS PC4.mylab.com.

;

localhost A 127.0.0.1

PC4.mylab.com. A 10.0.1.41

PC3.mylab.com. A 10.0.1.31

PC2.mylab.com. A 10.0.1.21 ←

PC1.mylab.com. A 10.0.1.11

Maksimum umur data  
cache dalam detik

• Record Start of authority  
(SOA) arti : “Zona  
authoritative Name server-  
nya Mylab.com”

• PC4.mylab.com adalah  
name server

• Email address PICnya  
hostmaster@mylab.com

Record Name server (NS).

Address (A) records.  
Satu entry untuk setiap  
hostaddress



# SOFTWARE

---

- ✖ Pada Redhat Linux yang sudah terinstall BIND (name server daemon) akan dijumpai beberapa file sebagai berikut :
- ✖ Di dalam /var/named akan ada 2 file yaitu :
  - + named.ca
  - + named.local
- ✖ Di dalam /etc akan terdapat file named.conf

# FILE-FILE KONFIGURASI

---

## Standard

- ✗ `named.conf` di dalam `/etc`
- ✗ `named.ca` di dalam `/var/named`
- ✗ `named.local` di dalam `/var/named`

Jika ingin membuat master server maka harus ada:

- ✗ file zone -> mapping dari nama ke IP
- ✗ file reverse zone -> mapping dari IP ke nama

---

**NAMED.CONF**

NAMED.CONF

# BLOK DALAM NAMED.CONF

---

- ✗ **options** — List konfigurasi global dan default
- ✗ **include** — berisi path file lain yang diperlukan
- ✗ **acl** — IP address dalam access control list
- ✗ **server** — properties khusus untuk remote servers
- ✗ **zone** — informasi khusus untuk zona



✖ // generated by named-bootconf.pl

1. Directory untuk  
menempatkan file zone

✖ options {

✖     directory "/var/named";

✖     /\*

✖     \* If there is a firewall between you and nameservers you want

✖     \* to talk to, you might need to uncomment the query-source

✖     \* directive below. Previous versions of BIND always asked

✖     \* questions using port 53, but BIND 8.1 uses an unprivileged

✖     \* port by default.

✖     \*/

✖     // query-source address \* port 53;

✖ };

✖ //

✖ // a caching only nameserver config

✖ //

✖ controls {

✖     inet 127.0.0.1 allow { localhost; };

✖ };

2. Blok untuk mengatur akses

### 3. Zone untuk root

- ✗ zone "." IN {
- ✗     type hint;
- ✗     file "named.ca";
- ✗ };

### 4. Zone untuk localhost

- ✗ zone "localhost" IN {
- ✗     type master;
- ✗     file "localhost.zone";
- ✗     allow-update { none; };
- ✗ };

### 5. Zone untuk reverse address

- ✗ zone "0.0.127.in-addr.arpa" IN {
- ✗     type master;
- ✗     file "named.local";
- ✗     allow-update { none; };
- ✗ };



# OPTIONS

---

- ✗ Biasanya ditaruh pada baris pertama named.conf

- ✗ Sintak :

```
options {  
    value "property";  
}
```

# OPTIONS : ALLOW-QUERY

- ✘ Menerima query hanya dari host dalam address yang sudah (default any host).
- ✘ Penggunaan: allow-query {"address-list"};.

# OPTIONS : ALLOW-TRANSFER

- ✗ Zone transfers menerima query hanya dari host dalam address yang sudah (default all host).
- ✗ Penggunaan : allow-transfer {"address list"};.

# OPTIONS : DIRECTORY

---

- ✗ Tempat dimana file konfigurasi server berada.
- ✗ Penggunaan: directory “path to directory”; (specify path).

# OPTIONS : FORWARDERS

- ✗ Menunjukkan IP addresses server untuk memforward query (default is none).
- ✗ Penggunaan: forwarders “IP addresses of servers”; (specify IP addresses).

# OPTIONS : FORWARD

---

- ✖ Jika diset pertama kali, Server akan didaftar pada query forwarders pertama,
- ✖ Penggunaan: forward “first or only”; (pilih salah satu).



# OPTIONS : LISTEN-ON

- ✘ Port dimana server listen dari query yang ada (default is port 53).
- ✘ Penggunaan : listen-on “port {address list}”;

# OPTIONS : RECURSION

---

- ✗ Server secara recursive mencari jawaban query (default is yes).
- ✗ Penggunaan: recursion “ yes or no”; (choose one).

# INCLUDE

---

- ✘ Berisi path dan file yang dibutuhkan yang berada diluar direktori yang sudah ditentukan pada `named.conf`.

# ACL

- ✗ IP address dalam access control list. Hanya host yang terdaftar yang boleh akses ke server
- ✗ `acl "transferdns" {`
- ✗     `{ 216.65.64.146/32; };`
- ✗     `{ 209.25.238/24; };`
- ✗     `{ 202.154.63.3/32; };`
- ✗ `};`

# NAMED.CA

---

- ✗ Dikenal sebagai cache file untuk DNS
- ✗ Berisikan daftar world root servers



# NAMED.CA

x ; This file holds the information on root name servers needed to  
x ; initialize cache of Internet domain name servers  
x ; (e.g. reference this file in the "cache . <file>"  
x ; configuration file of BIND domain name servers).

x ;  
x ; This file is made available by InterNIC  
x ; under anonymous FTP as  
x ; file /domain/named.cache  
x ; on server FTP.INTERNIC.NET

x ;  
x ; last update: Nov 5, 2002  
x ; related version of root zone: 2002110501

x ;  
x ;  
x ; formerly NS.INTERNIC.NET

x ;  
x . 3600000 IN NS A.ROOT-SERVERS.NET.  
x A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4

x ;  
x ; formerly NS1.ISI.EDU

x ;  
x . 3600000 NS B.ROOT-SERVERS.NET.  
x B.ROOT-SERVERS.NET. 3600000 A 128.9.0.107

x ;  
x ; formerly C.PSI.NET

x ;  
x . 3600000 NS C.ROOT-SERVERS.NET.

# NAMED.LOCAL

---

- ✕ Berisikan informasi tentang localhost
- ✕ Berisikan info untuk me-resolv loopback address untuk localhost

# NAMED.LOCAL

```
× @      IN      SOA      localhost. root.localhost. (
×                               1997022700 ; Serial
×                               28800    ; Refresh
×                               14400    ; Retry
×                               3600000  ; Expire
×                               86400 )  ; Minimum
×                IN      NS      localhost.
× 1            IN      PTR      localhost.
```

# NAMED.REV

---

- ✗ Menyediakan informasi untuk reverse lookups.
- ✗ Digunakan untuk mengetahui nama dari suatu host berdasarkan IP

# NAMED.REV

```
x 63.154.202.in-addr.arpa. IN    SOA    ns1.pens-its.edu. admin.pens-its.edu.
(
x          2000081012 ; Serial
x          28800      ; Refresh
x          14400      ; Retry
x          3600000    ; Expire
x          86400 )    ; Minimum
x          IN    NS    ns1.pens-its.edu.
x          IN    NS    ns2.pens-its.edu.

x 4          IN    PTR    www.pens-its.edu.
x 5          IN    PTR    ies.pens-its.edu.
x 6          IN    PTR    elerning.pens-its.edu.
```



# FILE ZONE

---

- ✗ File zone berisikan resource record (RR) tentang IP address
- ✗ File ZONE akan diawali oleh SOA yang merupakan penanda bahwa name server tersebut adalah merupakan sumber yang sah untuk domain tersebut
- ✗ SATU zone file HANYA akan punya SATU SOA

# SOA

- ✖ @ IN SOA main.tactechonology.com. mail.tactechonology.com. (
- ✖ 2000052101 ; Serial
- ✖ 8h ;Refresh
- ✖ 2h ;Retry
- ✖ 1w ;Expire
- ✖ 1d) ;Minimum TTL
  
- ✖ SOA seperti ini adalah Start Of Authority untuk domain yang di spesifikasikan di named.conf
- ✖ Nama server yang sah adalah main.technology.com
- ✖ Mail-address dari administratornya adalah mail.tatechonology.com

# SOA

- ✗ Serial : Serial number dari file zone tersebut
- ✗ Refresh : waktu yang dibutuhkan untuk me-refresh data
- ✗ Retry : waktu yang dibutuhkan untuk menunggu sebelum berusaha mengontak server utama jika ada kegagalan
- ✗ Expire : jika secondary master gagal mengontak server utama dalam waktu ini maka database tentang domain tersebut akan dibuang
- ✗ TTL: Time to live untuk menentukan berapa lama data disimpan dalam cache

# RESOURCE RECORD

---

- × **NS — NAME SERVERS**

- + Menunjukkan nama “name server”.

- × **A — THE IP ADDRESS FOR THE NAME**

- + Menunjukkan nomor IP “name server”.

- × **PTR — POINTER FOR ADDRESS NAME MAPPING**

- + Digunakan untuk menunjuk name server

- × **CNAME — CANONICAL NAME**

- + Menunjukkan nama real host.

- × **MX — MAIL EXCHANGE RECORD**

- + Menunjukkan sebagai mail server pada domain tersebut.

# DYNAMIC DNS

- ✗ Suatu cara melakukan update DNS server tanpa harus melakukan restart terhadap konfigurasi DNS kita.
- ✗ Pada waktu konfigurasi DNS harus ada cara untuk mengupdate, Pada waktu suatu host hidup kita bisa menyediakan address via DHCP, kemudian DHCP meminta DNS untuk merubah record A dan PTR sesuai kebutuhan.
- ✗ Kolaborasi antara DNS dan DHCP
- ✗ Membutuhkan bind9 dan DHCP3
- ✗ Konfigurasi file utama : dhcpd.conf dan named.conf
- ✗ Dijelaskan lebih lanjut pada Bagian DHCP Server



# PRAKTIKUM DNS

```
root@ubuntu: /var/cache/bind
root@ubuntu:/var/cache/bind# dhclient
There is already a pid file /var/run/dhclient.pid with pid 8508
killed old client process, removed PID file
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/

Listening on LPF/pan0/1a:ad:94:c2:62:a5
Sending on   LPF/pan0/1a:ad:94:c2:62:a5
Listening on LPF/eth0/00:0c:29:4e:0b:db
Sending on   LPF/eth0/00:0c:29:4e:0b:db
Sending on   Socket/fallback
DHCPREQUEST of 10.252.12.244 on eth0 to 255.255.255.255 port 67
DHCPACK of 10.252.12.244 from 10.252.12.1
  * Reloading /etc/samba/smb.conf smbd only
  ...done.
bound to 10.252.12.244 -- renewal in 1598 seconds.
root@ubuntu:/var/cache/bind#
```



root@ubuntu: /var/cache/bind



```
root@ubuntu:/var/cache/bind# apt-cache search bind9|grep ^bind|more
bind9 - Internet Domain Name Server
bind9-doc - Documentation for BIND
bind9-host - Version of 'host' bundled with BIND 9.X
bind9utils - Utilities for BIND
bindgraph - DNS statistics RRDtool frontend for BIND9
root@ubuntu:/var/cache/bind# apt-get install bind9
Reading package lists... Done
Building dependency tree
Reading state information... Done
bind9 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:/var/cache/bind#
```

root@ubuntu: /etc/bind

```
//di lab jarkom 10.252.108.179
zone "training.com" IN {
type master;
file "/var/cache/bind/db.training1";
};
```

```
zone "108.252.10.in-addr.arpa" IN {
type master;
file "/var/cache/bind/db.179:";
};
```

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind# ls

db.0	db.empty	named.conf	named.conf.local_copy	zones.rfc1918
db.127	db.local	named.conf.local	named.conf.options	
db.255	db.root	named.conf.local~	rndc.key	

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

root@ubuntu:/etc/bind#

```
root@ubuntu: /var/cache/bind
root@ubuntu:/etc/bind# cd /var/cache/bind/
root@ubuntu:/var/cache/bind# ls
db.179 db.72 db.training db.training1
root@ubuntu:/var/cache/bind# cat db.training
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      training.com. root.training.com. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       ns1.training.com.
@         IN      MX       10      mail.training.com.
;
www       IN      A        10.252.108.179
mail      IN      A        10.252.108.179

root@ubuntu:/var/cache/bind#
```

root@ubuntu: /var/cache/bind

```
root@ubuntu:/var/cache/bind# cat db.179
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      training.com. root.training.com. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL

@         IN      NS       ns1.training.com.
179       IN      PTR      ns1.training.com.
179       IN      PTR      www.training.com.
179       IN      PTR      mail.training.com.
```

root@ubuntu:/var/cache/bind#

root@ubuntu: /var/cache/bind

```
root@ubuntu:/var/cache/bind# cat /etc/resolv.conf
nameserver 10.252.108.179
domain eepis-its.edu
search eepis-its.edu
nameserver 202.9.85.3
nameserver 202.9.85.4
root@ubuntu:/var/cache/bind#
```



```
root@ubuntu: /var/cache/bind
root@ubuntu:/var/cache/bind# dig www.training.com

; <<>> DiG 9.5.1-P2 <<>> www.training.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15781
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;www.training.com.                IN      A

;; ANSWER SECTION:
www.training.com.                604800  IN      A      10.252.12.244

;; AUTHORITY SECTION:
training.com.                    604800  IN      NS      ns1.training.com.

;; Query time: 23 msec
;; SERVER: 10.252.12.244#53(10.252.12.244)
;; WHEN: Sun Aug 28 17:12:54 2011
;; MSG SIZE  rcvd: 68

root@ubuntu:/var/cache/bind#
```

root@ubuntu: /etc/bind

root@ubuntu:/etc/bind# cat /etc/bind/named.conf.options

```
options {  
    directory "/var/cache/bind";  
  
    // If there is a firewall between you and nameservers you want  
    // to talk to, you may need to fix the firewall to allow multiple  
    // ports to talk.  See http://www.kb.cert.org/vuls/id/800113  
  
    // If your ISP provided one or more IP addresses for stable  
    // nameserver service (this is highly recommended), uncomment the following  
    // Uncommen  
    // the all
```

root@ubuntu: /var/cache/bind

root@ubuntu:/var/cache/bind# cat /etc/resolv.conf

```
nameserver 10.252.12.244  
domain training.com  
  
forwarder  
// 0.  
200  
20  
};  
domain eepis-its.edu  
search eepis-its.edu  
nameserver 202.9.85.3  
nameserver 202.9.85.4  
root@ubuntu:/var/cache/bind# nslookup www.eepis-its.edu  
Server: 202.9.85.3  
Address: 202.9.85.3#53
```

```
Name: www.eepis-its.edu  
Address: 202.9.85.16
```

root@ubuntu:/var/cache/bind# nslookup www.training.com

```
Server: 10.252.12.244  
Address: 10.252.12.244#53
```

```
Name: www.training.com  
Address: 10.252.12.244
```

root@ubuntu:/var/cache/bind#

root@ubuntu: /etc/bind

```
root@ubuntu:/etc/bind# cat /etc/resolv.conf
nameserver 10.252.12.244
domain training.com
```

```
root@ubuntu:/etc/bind# nslookup www.training.com
Server:      10.252.12.244
Address:     10.252.12.244#53
```

```
Name:   www.training.com
Address: 10.252.12.244
```

```
root@ubuntu:/etc/bind# nslookup www.eepis-its.edu
Server:      10.252.12.244
Address:     10.252.12.244#53
```

```
Non-authoritative answer:
Name:   www.eepis-its.edu
Address: 202.9.85.16
```

```
root@ubuntu:/etc/bind#
```

root@ubuntu: /var/cache/bind

```
; <<>> DiG 9.5.1-P2 <<>> www.eepis-its.edu
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2277
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;www.eepis-its.edu.                IN      A

;; ANSWER SECTION:
www.eepis-its.edu.                86400   IN      A      202.9.85.16

;; AUTHORITY SECTION:
eepis-its.edu.                    86400   IN      NS      ns1.eepis-its.edu.
eepis-its.edu.                    86400   IN      NS      ns2.eepis-its.edu.

;; ADDITIONAL SECTION:
ns1.eepis-its.edu.                86400   IN      A      202.9.85.3
ns1.eepis-its.edu.                86400   IN      AAAA    2001:470:1f05:6bf::2
ns2.eepis-its.edu.                86400   IN      A      202.9.85.4
ns2.eepis-its.edu.                86400   IN      AAAA    2001:470:1f05:6bf::3

;; Query time: 4 msec
;; SERVER: 202.9.85.3#53(202.9.85.3)
;; WHEN: Sun Aug 28 17:16:13 2011
;; MSG SIZE rcvd: 175
```

root@ubuntu: /var/cache/bind#