Membangun Server DNS dengan djbdns

DNS adalah sistem penamaan komputer yang terdistribusi. Sebagai alternatif pengganti BIND di mesin-mesin yang menjalankan sistem operasi turunan UNIX adalah djbdns. Meski konsep yang dijalankan berbeda, tetapi mempunyai fungsi yang sama.

DNS adalah singkatan dari *Domain Name System*. Kadang-kadang diartikan sebagai *Domain Name Server*, jika sedang membahas *server*. DNS berfungsi untuk mengatur proses penerjemahan nama domain ke nomor IP atau sebaliknya. DNS adalah juga sistem database yang terdistribusi, sehingga memungkinkan setiap bagian dari database dikelola secara terpisah.

DNS bekerja dengan konsep *client-server*. Yang menjalankan fungsi sebagai server adalah server DNS. Di dunia *nix (Unix/Linux) biasanya dipakai program yang bernama *named* dari paket BIND, sedangkan di sisi klien disebut dengan *resolver*.

Database DNS mempunyai struktur yang mirip dengan pohon yang terbalik atau direktori dalam sistem operasi *nix. Struktur seperti ini bisa dilihat pada **Gambar 1**.

Kalau *root* direktori diberi lambang dengan karakter garis miring (/), dalam DNS root dilambangkan dengan karakter titik (.). Antara nama domain dan sub-domain dipisahkan dengan karakter titik juga.

DNS memegang peranan penting dalam komunikasi melalui Internet. Karena kita mengetahui komputer saling mengenal dan berkomunikasi memakai nomor IP. Sedangkan manusia lebih mudah berkomunikasi dengan nama. Sebagai contoh, lebih mudah bagi kita mengirim e-mail dengan alamat *salman@openware.or.id*, daripada *salman@[202.159.11.155]*. Demikian juga akan lebih mudah mengingat nama host *www.salman.or.id* dari pada nomor IP-nya.

Jika Anda ingin lebih tahu lebih dalam mengenai *Domain Name System ini*, saya anjurkan untuk membaca *DNS HOWTO* tulisan **Nicolai Langfeldt, Jamie Norris,** dkk. Buku karangan **Paul Albitz** dan **Cricket Liu** yang berjudul *DNS and BIND* sangat lengkap membahas teori dan cara kerja DNS. Dalam tulisan ini pembahasan lebih ditekankan pada pemakaian praktis, dengan asumsi pembaca sudah bisa bekerja dengan Linux dalam mode text. Ya, bukan grafik.

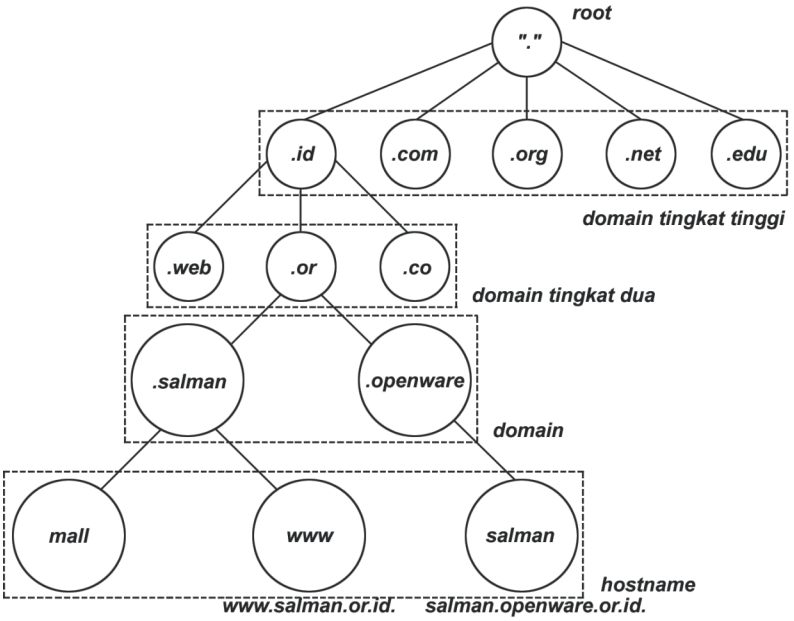
Paket djbdns adalah sekumpulan program aplikasi dengan ukuran kecil yang bisa disusun sesuai dengan kebutuhan. Peranti lunak ini dibuat dengan kerangka berpikir yang mengutamakan keamanan dan unjuk kerja yang tinggi. Untuk membatasi pembahasan supaya tidak terlalu melebar, hanya akan dibahas tiga program aplikasi dalam paket djbdns, yaitu *tinydns, dnscache,* dan *axfrdns.* Pertimbangan yang diambil karena aplikasi ini yang paling sering dipakai.

Dalam lapisan-lapisan TCP/IP, tinydns bekerja di lapisan UDP bukan di lapisan TCP. Karena itu tinydns mempunyai respons yang lebih cepat, akan tetapi mempunyai kelemahan panjang data yang bisa disampaikan sangat terbatas. Hal ini disebabkan protokol UDP tidak berorientasi pada koneksi, jadi agar bagian-bagian data tidak hilang harus dikirim dalam paket yang kecil.

Kesederhanaan dan optimasi database tinydns juga menyumbang pada unjuk kerja. Semua aplikasi dalam djbdns memakai database statik dalam format cdb. Akibatnya Anda harus selalu mengonversi file text data DNS menjadi format cdb ini jika ada perubahan yang Anda lakukan terhadap data DNS. Akan tetapi, Anda tidak perlu khawatir, yang perlu dilakukan untuk melakukan konversi ini hanyalah menjalankan perintah *make.*

Karena layanan DNS sangat penting, maka perlu adanya server cadangan yang melayani sebuah domain. Paket BIND menyediakan mekanisme penggandaan layanan dengan nama zone transfer. Layanan ini sudah dimasukkan dalam program utama yang bernama *named.*

Layanan yang sama, diimplementasikan di djbdns dengan menjalankan aplikasi terpisah. Program tersebut bernama *axfrdns*. Kalau tinydns berjalan di lapisan UDP dan mendengarkan pada port 53, maka axfrdns berjalan pada lapisan TCP dengan port yang sama, sehingga kedua layanan



Struktur Database DNS

ini bisa berjalan pada saat bersamaan. Dan biasanya kedua program ini mengambil data dari sumber yang sama, yaitu file */etc/tinydns/root/data.cdb.*

Sebenarnya ada cara lain yang lebih bagus dan aman untuk membuat server DNS cadangan, yaitu dengan cara menyalin setiap perubahan file data di server utama ke server-server cadangan tersebut. Anda bisa menggunakan program *rsync* atau *scp.* Dengan cara ini konsep zone transfer sudah tidak berlaku lagi.

Program ketiga yang akan kita bahas adalah *dnscache*. Program ini akan kita perlukan jika kita tidak ingin membangun dns server sendiri. Ia akan menyimpan setiap permintaan penerjemahan DNS yang didapatkan dari server-server DNS di Internet. Jika dnscache menerima permintaan dari *resolver,* dia akan meneruskan ke server-server DNS di Internet. Setelah jawaban yang benar didapatkan, kemudian akan disimpan ke dalam *cache*. Jika nanti ada resolver meminta penerjemahan nama dengan nama domain yang sudah ada dalam cache, maka akan diambilkan dari cache.

Instalasi djbdns

Untuk menjalankan DNS dengan djbdns diperlukan tiga paket peranti lunak, yaitu:

- **daemontools,** peranti lunak ini bisa Anda dapatkan dari URL:
  - ✓ *http://cr.yp.to/daemontools/daemontools-0.76.tar.gz*
  - ✓ *http://sas.docspages.com/software/daemontools-0.76.tar.gz*
- **ucspi-tcp,** peranti lunak ini bisa Anda dapatkan dari URL:
  - ✓ *http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz*
  - ✓ *http://sas.docspages.com/software/ucspi-tcp-0.88.tar.gz*
- **djbdns,** peranti lunak ini bisa Anda dapatkan dari URL:
  - ✓ *http://cr.yp.to/djbdns/djbdns-1.05.tar.gz*
  - ✓ *http://sas.docspages.com/software/djbdns-1.05.tar.gz*

daemontools

Paket program **daemontools** adalah sekumpulan program untuk mengadministrasi program-program yang akan dijalankan sebagai server/daemon. Dalam paket ini juga terdapat program untuk mencatat aktivitas server.

Ikuti langkah-langkah berikut untuk meng-instalasi paket program daemontools secara manual:

- Buat direktori /package/

```
# mkdir -p /package
# chmod 1755 /package
```

- Urai file paket program dalam direktori /package/

```
# tar -xzv -C /package/ -f daemontools-0.76.tar.gz
```

- Kompilasi dan *setup* daemontools

```
# cd /package/admin/daemontools-0.76/
# package/install
```

Proses setup akan membuat direktori /service/ dan */command/*. Selain itu, pada file */etc/inittab* akan ditambahkan baris:

```
SV:123456:respawn:/command/svscanboot
```

Itu untuk distribusi Linux yang menganut System V. Tapi jika file tersebut tidak ada, maka akan ditambahkan pada file */etc/rc.local* baris:

```
csh -cf '/command/svscanboot &'
```

Proses ini juga akan mengirimkan pesan pada init untuk membaca ulang file konfigurasinya, yang juga berarti langsung menjalankan program *svscanboot*. Program ini akan selalu diawasi oleh init supaya tidak mati.

svscanboot menjalankan program svscan, yang akan selalu mengawasi direktori */service/*. Jika program ini melihat keberadaan sebuah subdirektori dalam direktori /service/, ia akan menjalankan *script* yang bernama *run* di subdirektori tersebut. Tentu saja script tersebut harus bisa dieksekusi.

ucspi-tcp

Paket program ucspi-tcp terdiri dari program server dan beberapa program aplikasi kecil lainnya. Paket progam ini berfungsi untuk membangun aplikasi client-server dengan mudah. Sekadar Anda tahu saja, UCSPI sendiri adalah kependekan dari *UNIX Client-Server Program Interface.*

Paket peranti lunak ini bisa menggantikan posisi *inetd/xinetd* yang biasanya dijadikan komponen standar dalam semua distribusi linux. ucspi-tcp diklaim mempunyai kelebihan dibandingkan dengan inetd/xinetd. Salah satu kelebihannya adalah proses koneksi yang terjadi bersamaan bisa dibatasi dengan jumlah tertentu. Kelebihan lain bisa Anda lihat di situs resminya di *http://cr.yp.to.*

Ikuti langkah-langkah berikut secara bertahap:

- Download file dan salin ke direktori /usr/local/src/

```
# cp ucspi-tcp-0.88.tar.gz /usr/local/src/
```

- Urai file.

```
# cd /usr/local/src/
# tar -xzvf ucspi-tcp-0.88.tar.gz
```

- Kompilasi dan setup paket program

```
# cd ucspi-tcp-0.88
# make
# make setup check
```

djbdns

Instalasi paket djbdns dapat dilakukan dengan langkah-langkah berikut:

- Setelah Anda download file *djbdns-1.05.tar.gz,* salin ke direktori */usr/local/src/*, dengan perintah:

```
# cp djbdns-1.05.tar.gz /usr/local/src/
```

- Uraikan file tersebut.

```
# cd /usr/local/src/
# tar -xzvf djbdns-1.05.tar.gz
```

- Kemudian lakukan proses kompilasi dan instalasi, dengan perintah:

```
# cd djbdns-1.05
# make
# make setup check
```

Setup tinydns

Setelah djbdns diinstalasi, ia belum berfungsi sebagai DNS. Untuk bisa melayani permintaan dari program-program yang ingin menerjemahkan nama menjadi nomor IP, djbdns harus dikonfigurasi sesuai dengan kebutuhan.

Untuk mempermudah pembahasan, anggap kita akan mensetup sebuah layanan dns yang melayani domain *salman.or.id* dan penerjemahan nomor IP 192.168.1.*.

Setup tinydns sangat mudah. Jika Anda mengikuti langkah-langkah yang saya berikan di sini, Anda tidak akan menemukan masalah yang berarti.

- **Langkah pertama,** Anda harus menyiapkan *user* yang akan menjalankan server dns dan *log*-nya. Standard user-user ini adalah tinydns dan dnslog. Akan tetapi, Anda boleh memakai nama user yang lainnya.

```
# useradd -d /etc/tinydns -s /dev/null tinydns
# useradd -d /etc/tinydns -s /dev/null dnslog
```

User-user tersebut tidak perlu memiliki *privilege* apapun, tetapi harus bisa menjalankan program.

- **Kedua,** jalankan perintah berikut untuk menyalin file-file yang diperlukan oleh tinydns.

```
# tinydns-conf tinydns dnslog /etc/tinydns
192.168.1.20
```

/etc/tinydns adalah root direktori tinydns, sedangkan 192.168.1.20 adalah nomor IP komputer termpat kita akan menjalankan layanan tinydns.

- **Ketiga,** informasikan pada program svscan bahwa kita akan menjalankan layanan dns. Program ini adalah bagian dari paket daemontools. Jalankan perintah:

```
# ln -s /etc/tinydns /service
```

- **Keempat,** buat database DNS. Tambahkan domain salman.or.id dan 1.168.192 untuk permintaan *reverse*-nya.

```
# cd /service/tinydns/root/
# ./add-ns salman.or.id 192.168.1.20
# ./add-ns .1.168.192.in-addr.arpa 192.168.1.20
# make
```

Hasil dari perintah-perintah tersebut adalah dua buah file bernama data dan data.cdb. File yang terakhir disebut ini adalah hasil dari perintah make yang berfungsi untuk membuat file database dengan format cdb dari file text. Isi file text tersebut adalah:

```
.salman.or.id:192.168.1.20:a:259200
.1.168.192.in-addr.arpa:192.168.1.20:a:259200
```

Anda tidak perlu pusing-pusing memikirkan format file text yang agak eksentrik ini. Nanti kita akan bahas kembali untuk memahami apa artinya.

- **Langkah terakhir,** kita harus mencoba name server kita apakah sudah berfungsi, sebelum

mempelajari lebih dalam lagi djbdns.

```
# dnsq a salman.or.id 192.168.1.20
1 salman.or.id:
82 bytes, 1+0+1+0 records, response, authoritative, noerror
query: 1 salman.or.id
authority: salman.or.id 2560 SOA a.ns.salman.or.id hostmaster.salman.or.id 1045229519 16384 2048 1048576 2560
```

Tentu saja ini harus Anda lakukan setelah Anda mengisi file */etc/resolv.conf* dengan nomor IP dari komputer yang sedang Anda setup sebagai name server. Dan selamat, Anda sudah memiliki dns server yang siap menjalankan tugasnya!

Format perintah dnsq adalah:

```
dnsq type name server
```

di mana, type adalah *resource record* apa yang kita inginkan misalnya, a, ns, any, mx, txt, soa. Sedangkan *name* adalah nama host lengkap atau nama domain dan server adalah name server tujuan.

Menambahkan data

Dalam paket djbdns sudah disediakan script-script untuk menambahkan data ke dalam database. Pekerjaan yang sering dikerjakan dalam mengelola dns adalah:

- Menambahkan nama host.
- Menambahkan alias host.
- Menambahkan informasi *mail server* untuk domain.
- Menambahkan informasi name server untuk domain.
- Mendelegasikan sebuah sub-domain kepada server dns yang lain.

Jalankan perintah-perintah di bagian ini dari root direktori tinydns, */service/tinydns/root/.*

Untuk menambah *hostname* bisa dipakai perintah seperti ini:

```
# ./add-host fariz.salman.or.id 192.168.1.192
# ./add-host salma.salman.or.id 192.168.1.20
# make
```

Jika Anda mencoba menambahkan nama baru dengan nomor IP yang sudah pernah didaftarkan akan ditolak. Pada dasarnya sebuah server hanya memiliki satu nama, tapi jika ingin sebuah server/host bisa memiliki nama lain Anda bisa menambahkan nama alias ini ke file *data.*

```
# ./add-alias mail.salman.or.id 192.168.1.20
# make
```

Untuk menambahkan mail server untuk domain tertentu gunakan perintah *add-mx*.

```
# ./add-mx salman.or.id 192.168.1.20
# make
```

Apabila perintah-perintah di atas Anda lakukan, file /etc/tinydns/root/data, akan seperti ini:

```
.salman.or.id:192.168.1.20:a:259200
.1.168.192.in-addr.arpa:192.168.1.20:a:259200
=fariz.salman.or.id:192.168.1.192:86400
=salma.salman.or.id:192.168.1.20:86400
+mail.salman.or.id:192.168.1.20:86400
@salman.or.id:192.168.1.20:a::86400
```

Format data tinydns

Tinydns mencari informasi tentang domain dari file database dengan format cdb, namanya *data.cdb.* Seperti telah disebutkan sebelumnya, file ini berasal dari file yang bernama data. File yang disebutkan terakhir inilah yang akan kita bahas lebih lanjut.

Kalau Anda jeli melihat contoh file data yang dihasilkan dari script-script yang kita jalankan di bagian-bagian sebelumnya, bisa Anda lihat ada empat hal:

- Informasi nama name server diawali dengan karakter titik (.).
- Informasi nama host diawali dengan karakter sama-dengan (=).
- Informasi nama alias selalu diawali dengan karakter plus (+).
- Informasi nama mail server selalu diawali dengan karakter at-sign (@).

Mari kita lihat satu per satu isi dari file data tersebut. Lihat dua baris yang berisi informasi mengenai name server berikut ini:

```
.salman.or.id:192.168.1.20:a:259200
.1.168.192.in-addr.arpa:192.168.1.20:a:259200
```

Pola keduanya sama, yaitu:

```
.nama-domain:nomor-ip:name-server:ttl
```

Nama-domain harus dalam bentuk *Full Qualified Domain Name* (FQDN) dan diawali dengan karakter titik. Name server utama untuk domain ini adalah *name-server.ns.nama-domain*, jika name-server tidak mengandung tanda titik. Akan tetapi, jika tidak ada tanda titik maka name-server dipakai sebagai name server tanpa ditambahkan nama-domain di belakangnya. *nomor-ip* adalah nomor IP dari *name-server.ns.nama-domain*. Setiap domain boleh diberikan name server lebih dari satu, akan tetapi

yang dianggap server utama hanya satu saja. Alamat kontak administrator adalah *hostmaster@nama-domain.*

Pada contoh ini, jika name-server adalah ns..salman.or.id, maka nama ini yang akan dipakai sebagai NS resource record.

Time to live atau disingkat ttl adalah batas waktu—dalam detik—data DNS bisa disimpan dalam cached.

Dua baris berikutnya berisi informasi tentang hostname, seperti ini:

```
=fariz.salman.or.id:192.168.1.192:86400
=salma.salman.or.id:192.168.1.20:86400
```

Pola keduanya sama, yaitu:

```
=nama-host:nomor-ip:ttl
```

Informasi ini menentukan nomor-ip diasosiasikan dengan nama-host. Nama host harus disebutkan lengkap dengan domainnya.

Alias dari hostname direpresentasikan dengan baris,

```
+mail.salman.or.id:192.168.1.20:86400
```

Format ini sama dengan informasi hostname hanya disini diawali dengan tanda plus (+).

Sedangkan, informasi tentang mail server sebuah domain diberikan dengan

```
@salman.or.id:192.168.1.20:a::86400
```

Polanya adalah:

```
@nama-domain:nomor-ip:mail-server:bobot:ttl
```

Aturan yang berlaku di sini sama dengan aturan yang berlaku untuk name server. Tetapi ada informasi tambahan di sini, yaitu bobot. Anda bisa memberikan beberapa mail server untuk satu nama domain. Mail server akan diakses sesuai dengan urutan bobot ini.

Setup dnscache

Tanpa menggunakan cache, setiap kita mencari informasi tentang sebuah domain maka server DNS akan sibuk menjawab pertanyaan yang sama berkali-kali. Jika proses ini dilakukan untuk banyak client, akibatnya beban kerja dari DNS server akan berat. Seyogyanya setiap administrator sistem menyiapkan paling tidak sebuah server cache DNS. dnscache adalah bagian dari paket djbdns. Ia bekerja pada port 53 dengan protokol TCP dan UDP.

Ada dua jenis cache yang bisa kita buat, yaitu:

- Cache internal.
- Cache external.

Perbedaan kedua jenis cache ini hanya terletak pada tempat meletakkan cache DNS. Cache internal biasanya dipakai untuk akses melewati nomor IP 127.0.0.1, yang akan diterjemahkan sebagai *localhost*. Sedangkan layanan cache DNS eksternal terletak pada nomor IP ethernet card.

Cara men-setup cache DNS server seperti ini:

- Tambahkan user yang akan menjalankan cachedns:

```
# useradd -d /dev/null -s /dev/null dnscache
# useradd -d /dev/null -s /dev/null dnslog
```

- Konfigurasikan dnscache dengan perintah:

```
# dnscache-conf dnscache dnslog /etc/dnscache/
```

- Aktifkan server baru ini dengan daemontools, dengan cara membuat *link* di direktori *service.*

```
# ln -s /etc/dnscache/ /service/
```

Untuk melihat bahwa server dnscache kita sudah berjalan, perintahkan:

```
# netstat -pln | grep dnscache
```

Hasilnya akan ditampilkan seperti berikut:

```
(Not all processes could be identified, non-owned
process info
 will not be shown, you would have to be root to see it
all.)
 tcp    0   0 127.0.0.1:53  0.0.0.0:*   LISTEN  5038/
dnscache
 udp    0   0 127.0.0.1:53  0.0.0.0:*           5038/
dnscache
```

Terlihat di sini bahwa dnscache local bekerja di nomor IP 127.0.0.1 port 53 dengan protokol UDP dan TCP.

Untuk membuat server cache DNS eksternal prosesnya sama dengan cache internal hanya ada sedikit perbedaan saat menjalankan program *dnscache-conf* dan saat membuat *soft link.*

```
# dnscache-conf dnscache dnslog /etc/dnscachex
192.168.1.192
# ln -s /etc/dnscachex /service/
```

Setup axfrdns

Untuk memberi kemudahan proses *back up* data dari sebuah name server ke name server yang kedua, djbdns menyertakan program axfrdns untuk menjaga kompatibilitas dengan BIND. Program ini berguna jika Anda ingin menjalankan back up name server Anda dengan konsep *zone-transfer* milik BIND.

Akan tetapi, jika Anda menjalankan semua DNS server Anda, baik *primary* maupun *second-*

ary, dengan djbdns sangat dianjurkan tidak menggunakan konsep zone-transfer ini. Lebih baik gunakanlah program bantu lain yang lebih aman dan cepat, misalnya scp atau rsync. scp adalah bagian dari paket openSSH.

Jika Anda masih berkeras mau menggunakan zone-transfer, langkah-langkah berikut bisa dijalankan untuk mengaktifkan axfrdns:

- Buatlah user axfrdns dan dnslog jika belum ada.

```
# useradd -d /dev/null -s /dev/null axfrdns
# useradd -d /dev/null -s /dev/null dnslog
```

- Setup axfrdns dengan perintah:

```
# axfrdns-conf axfrdns dnslog /etc/axfrdns /etc/
tinydns 192.168.1.20
```

- Tambahkan no IP host yang boleh melakukan zone transfer beserta domain apa saja. Untuk menambahkan edit file /etc/axfrdns/tcp. Sebagai contoh isi file ini adalah:

```
# sample line:
1.2.3.4:allow,AXFR="heaven.af.mil/3.2.1.in-
addr.arpa"
192.168.1.192:allow,AXFR="salman.or.id"
:deny
```

Kemudian onversi file text ini menjadi format cdb dengan perintah:

```
# make
```

Dengan isi file seperti ini host yang boleh melakukan zone transfer hanyalah host dengan IP 192.168.1.192. Dan zone yang boleh ditransfer hanyalah salman.or.id. Jika Anda menginginkan zone lain juga bisa ditransfer dari host yang sama, tambahkan zone itu ke dengan format yang ditunjukkan pada baris yang diberi remark di atas. Baris ketiga menetapkan host lain akan ditolak jika melakukan transfer zone.

- Langkah terakhir jalankan service.

```
# ln -s /etc/axfrdns/ /service/
```

Sekian pembahasan singkat dari instalasi sampai setup name server menggunakan djbdns. Segala kritik, komentar, koreksi, alamatkan ke *sas@salman.or.id*. Versi terbaru dari dokumen ini bisa Anda dapatkan dari url *http://www.salman.or.id/djbdns/*.

Menggunakan utilitas paket djbdns

Dalam paket djbdns selain berisi peranti lunak dns server-nya sendiri, juga terdapat *software-software client* untuk proses *troubleshooting*.

- **dnnsip,** dipakai untuk mencari nomor IP dari sebuah full qualified domain name atau disingkat fqdn. Format dan contoh:

```
dnsip [fqdn] [fqdn] ...
# dnsip xyz.openware.or.id salman.or.id
```

- **dnsmx,** dipakai untuk mencari mail server/MX record dari sebuah domain. Format dan contoh:

```
dnsmx fqdn
# dnsmx salman.or.id
```

- **dnsname,** berfungsi untuk melakukan reverse lookup sebuah nomor IP untuk mengetahui nama host-nya. Format dan contoh:

```
dnsname nomor_ip
# dnsname 202.159.11.156
```

- **dnsq,** berguna untuk melakukan query nonrekursif terhadap sebuah dns server. Jadi query tidak akan dilacak dari name server root. Format dan contoh:

```
dnsq type name server
# dnsq a salman.or.id ns3.sslguarded.com
```

Catatan:

Server boleh berupa nomor IP atau nama server. Sedangkan *type* boleh berisi type RR, misalnya a, mx, ns, soa, dan lain sebagainya.

- **dnsqr,** untuk mencari type RR tertentu di bawah domain. Format dan contoh:

```
dnsqr type name
# dnsqr a google.com
```

- **dnstrace,** berguna untuk melacak name server apa saja yang mempengaruhi resolusi sebuah type RR tertentu dari sebuah domain. Proses pelacakan dimulai dari satu atau lebih name server root. Format dan contoh:

```
dnstrace t fqdn r
# dnstrace a salman.or.id  a.root-servers.net
```

- **dnstxt,** dipakai untuk mencari RR TXT dari sebuah domain. Format dan contoh:

```
dnstxt fqdn
# dnstxt openware.or.id
```

Daftar rujukan

- DNS HOWTO, Nicolai Langfeldt, Jamie Norris dkk.
- DNS and BIND, Paul Albitz dan Cricket Liu.
- http://cr.yp.to/djbdns/djbdns.html.

Salman Agus Supriadi *(http://salman.or.id)*