BAB III

DOMAIN NAME SYSTEM (DNS)

UNIT PELAKSANA TEKNIS PERPUSTAKAAN

3.1 Sejarah DNS

Sebelum dipergunakannya DNS, jaringan komputer menggunakan HOSTS files yang berisi informasi dari nama komputer dan IP address-nya. Di Internet, file ini dikelola secara terpusat dan di setiap loaksi harus di copy versi terbaru dari HOSTS files, dari sini bisa dibayangkan betapa repotnya jika ada penambahan 1 komputer di jaringan, maka kita harus copy versi terbaru file ini ke setiap lokasi.

Dengan makin meluasnya jaringan internet, hal ini makin merepotkan, akhirnya dibuatkan sebuah solusi dimana DNS di desain menggantikan fungsi HOSTS files, dengan kelebihan unlimited database size, dan performace yang baik. DNS adalah sebuah aplikasi services di Internet yang menerjemahkan sebuah domain name ke IP address. Sebagai contoh, www untuk penggunaan di Internet, lalu diketikan nama domain, misalnya: yahoo.com maka akan di petakan ke sebuah IP mis 202.68.0.134. Jadi DNS dapat di analogikan pada pemakaian buku telepon, dimana orang yang kita kenal berdasarkan nama untuk menghubunginya kita harus memutar nomor telepon di pesawat telepon. Sama persis, host komputer mengirimkan queries berupa nama komputer dan domain name server ke DNS, lalu oleh DNS dipetakan ke IP address.

3.2 Pengertian DNS

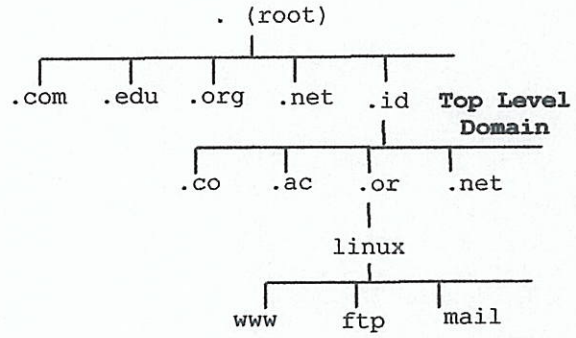
Domain Name System (DNS) adalah distribute database system yang digunakan untuk pencarian nama komputer (name resolution) di jaringan yang mengunakan TCP/IP (Transmission Control Protocol/Internet Protocol). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti web atau e-mail, dimana DNS membantu memetakan host name sebuah komputer ke IP address.Selain digunakan di Internet, DNS juga dapat di implementasikan ke private network atau intranet dimana DNS memiliki keunggulan seperti:

1. **Mudah**, DNS sangat mudah karena user tidak lagi direpotkan untuk mengingat IP address sebuah komputer cukup host name (nama Komputer).
2. **Konsisten**, IP address sebuah komputer bisa berubah tapi host name tidak berubah.
3. **Simple**, user hanya menggunakan satu nama domain untuk mencari baik di Internet maupun di Intranet.

DNS dapat disamakan fungsinya dengan buku telepon. Dimana setiap komputer di jaringan Internet memiliki host name (nama komputer) dan Internet Protocol (IP) address. Secara umum, setiap client yang akan mengkoneksikan komputer yang satu ke komputer yang lain, akan menggunakan host name. Lalu komputer anda akan menghubungi DNS server untuk mencek host name yang anda minta tersebut berapa IP

address-nya. IP address ini yang digunakan untuk mengkoneksikan komputer anda dengan komputer lainnya.

DNS merupakan suatu *database* tersebar yang berisi nama (alfanumerik) dan alamat IP dari setiap sistem yang terdaftar di Internet (Yager 1995). Tujuan DNS adalah untuk menyediakan mekanisme tata penamaan suatu *resource* jaringan sedemikian hingga dapat digunakan secara bersamaan dalam *host*, jaringan, famili protokol, Internet dalam suatu keseragaman (Mockapetris 1987b).



Gambar 1. Struktur *tree* DNS

DNS bertugas untuk menterjemahkan nama *host* menjadi alamat IP yang unik dan sebaliknya. Pada puncak *database* berhirarki ini adalah domain *root* yang dikelola oleh *Internet Assigned Number Association* (IANA, **http://www. iana.org**). Di bawahnya, terdapat *Top Level Domain* (TLD) seperti **org** (*Organization*), **com** (*Commercial*), **au** (*Australia*), **id** (Indonesia), dan sebagainya (Samik-Ibrahim 1997).

DNS mendefinisikan prosedur untuk mengakses data dan memberikan *pointer* ke *name server* lain. DNS juga mendefinisikan

prosedur dalam melakukan *caching* pada data yang pernah diambil dan juga mendefinisikan prosedur dalam melakukan *refresh* data secara berkala dalam waktu yang telah ditentukan oleh *system administrator*.

Dalam sistem ini diperlukan *system administrator* yang bertugas untuk :

- mendefinisikan batasan zone pada domain
- membuat *master file,*
- melakukan update pada master file, dan
- menentukan waktu refresh

Sementara itu pada DNS harus telah didefinisikan:

- format standar *master file*,
- metode standar query pada *database*, dan
- metode standar pada name *server* untuk melakukan *refresh* data lokal dari name *server* lain

(Mockapetris 1987a).

3.3 Struktur DNS

Domain Name Space merupakan sebuah hirarki pengelompokan domain berdasarkan nama, yang terbagi menjadi beberapa bagian diantaranya:

3.3.1 Root-Level Domains

Domain ditentukan berdasarkan tingkatan kemampuan yang ada di struktur hirarki yang disebut dengan level. Level paling atas di hirarki disebut dengan root domain. Root domain di ekspresikan berdasarkan periode dimana lambang untuk root domain adalah (".").

3.3.2 Top-Level Domains

Pada bagian dibawah ini adalah contoh dari top-level domains:

com Organisasi Komersial

edu Institusi pendidikan atau universitas

org Organisasi non-profit

net Networks (backbone Internet)

gov Organisasi pemerintah non militer

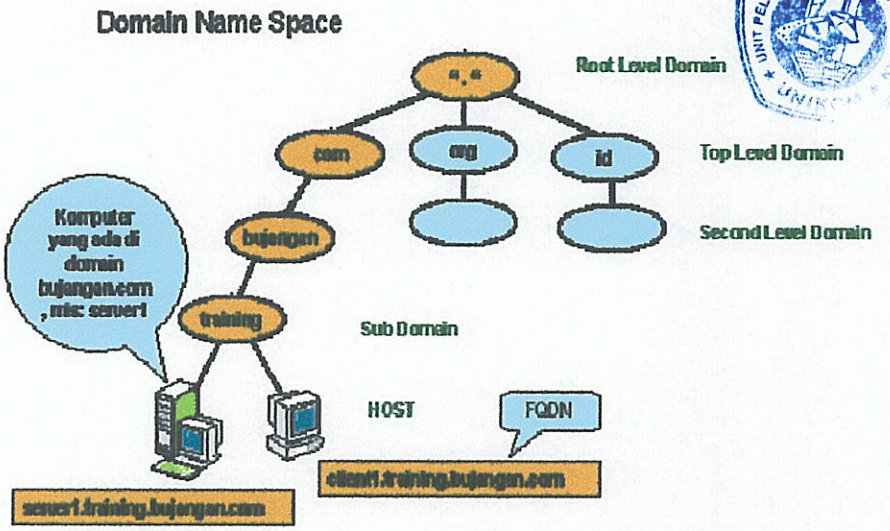
mil Organisasi pemerintah militer

num No telpon

arpa Reverse DNS

xx dua-huruf untuk kode negara (id:Indonesia,sg:singapura,au:australia,dll)

Top-level domains dapat berisi second-level domains dan hosts.



Second-Level Domains

Second-level domains dapat berisi host dan domain lain, yang disebut dengan subdomain. Untuk contoh: Domain Bujangan, bujangan.com terdapat komputer (host) seperti server1.bujangan.com dan subdomain training.bujangan.com. Subdomain training.bujangan.com juga terdapat komputer (host) seperti client1.training.bujangan.com.

Host Names

Domain name yang digunakan dengan host name akan menciptakan fully qualified domain name (FQDN) untuk setiap komputer. Sebagai contoh, jika terdapat fileserver1.detik.com, dimana fileserver1 adalah host name dan detik.com adalah domain name.

3.4 Sistem Kerja DNS

Konsep DNS disusun oleh Mockapetris (1987a) dalam STD013 (RFC1034 dan RFC1035). DNS terdiri dari tiga komponen, yaitu:

1. *DOMAIN NAME SPACE* dan *RESOURCE RECORD* (RR) yang merupakan spesifikasi penamaan domain dan data yang berhubungan dengan domain tersebut dalam bentuk struktur *tree*.
2. Program *NAME SERVER* yang menyimpan struktur *tree* dan beberapa informasi dari domain yang dikelola. Program ini dapat melakukan *cache* terhadap beberapa bagian dari struktur *tree* domain (Gambar 1), namun harus memiliki struktur *tree* dari subdomain yang dikelola secara lengkap. Bila name *server* ini memiliki struktur *tree* yang lengkap pada suatu bagian domain, maka name *server* ini disebut sebagai *AUTHORITY* dari bagian domain tersebut dan informasinya disimpan dalam unit-unit yang disebut dengan *ZONE*
3. Program *RESOLVER* yang bertugas untuk mengekstrak informasi yang diberikan *name server* sesuai dengan permintaan *client*. Program ini harus mampu berhubungan dengan minimal satu *name server* dan langsung menggunakan *name server* tersebut untuk melakukan pemetaan nama *host* ke alamat IP secara langsung, ataupun melakukan *forward query*, yaitu meminta *resolver* lain untuk mencari alamat IP tersebut.

Namespace, RR, dan *name server* berada di suatu mesin yang khusus bertugas sebagai *server* DNS. *Resolver* merupakan *server* khusus di suatu jaringan yang bertugas menghubungi *name server*. *Resolver* biasanya juga bertugas sebagai *name server*. Proses pencarian (*resolving*) alamat IP suatu *hostname* ditampilkan pada Lampiran 3.

Pada saat seorang user ingin melakukan telnet ke mesin **www** pada domain **linux.or.id**, maka ia akan mengetikkan FQDN milik mesin tersebut, misalnya:

[palui $] telnet www.linux.or.id
trying 202.146.0.244

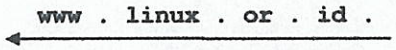
Pada contoh di atas, FQDN **www.linux.or.id** diterjemahkan menjadi alamat IP **202.146.0.244** oleh *name server*.

FQDN adalah nama lengkap suatu *host*, yaitu gabungan nama *host* dengan domain tempat *host* itu berada.

Proses yang sesungguhnya terjadi pada contoh di atas adalah:

1. Protokol TCP/IP akan menghubungi *resolver* untuk mencari alamat IP dari FQDN **www.linux.or.id**. (Pada sistem Unix dan klone, daftar *resolver* ditentukan pada file /etc/resolv.conf).
2. Kemudian *resolver* akan menghubungi *name server* untuk mencari alamat IP mesin **www** pada domain **linux.or.id** dengan urutan sebagai berikut (lihat juga Gambar 2 dan Lampiran 3) :

1. mencari alamat IP *name server* domain **root** pada *cache*. Domain **root** adalah domain tertinggi pada struktur *tree* DNS di dunia. Domain ini dilambangkan dengan karakter titik (.). Daftar *name server* domain root harus sudah dituliskan pada file *cache* dan harus dimiliki oleh setiap *resolver*.
2. mencari alamat IP *name server* domain **id** pada *server* **root.**
3. mencari alamat IP *name server* domain **or** pada *name server* **id.**
4. mencari alamat IP *name server* domain **linux** pada *name server* **or.id.**
5. mencari alamat IP *host* **www** pada *name server* domain **linux.or.id.**



Arah penelusuran

Gambar 2. Arah penelusuran FQDN www.linux.or.id

Penelusuran di atas tidak selalu berlaku mengingat pada *name server* pada tiap tahap terdapat *cache* yang berisi *request* yang pernah dilakukan sehingga proses *request* menjadi lebih cepat dilakukan. Bila ternyata *request* mengalami kegagalan maka dianggap FQDN tersebut tidak memiliki alamat IP dan dianggap FQDN yang tidak sah.

Pada DNS, dibutuhkan lebih dari satu *server* khusus untuk mengelola suatu domain. Tujuannya adalah agar apabila *server* utama

(*Primary*) sedang berhalangan (*down*), maka terdapat *server* penggantinya (*secondary*, *tertiary*, dan seterusnya) untuk menangani *request* dari *resolver*. *Server secondary* akan men*download database* zone domain dari *server primary* dan terus melakukan *update database* pada periode *refresh* tertentu. Kegiatan men*download database* zone domain disebut juga dengan transfer zone.

3.5 Master File

Master File adalah sebuah file teks yang berisi *database* RR dalam bentuk teks (Mockapetris 1987b). File ini digunakan oleh *name server* sebagai penyimpan struktur *tree* dari domain. Pada penelitian ini, master file merupakan file yang selalu di*update* bila terdapat transaksi *update* alamat IP pada *server* DNS Dinamis.

Tiap RR memiliki label, kelas, tipe, dan data (Elz & Bush 1997). Dan tidak boleh ada duplikasi RR pada master file.

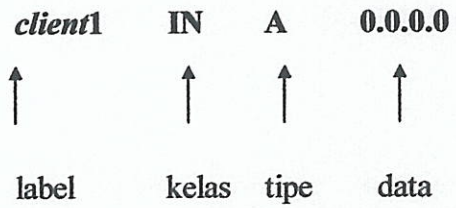
Menurut Mockapetris (1987b) dalam master file ada beberapa tipe RR yang biasa digunakan, yaitu :

- ***Start of Authority*** (SOA). RR ini berisi keterangan mengenai zone yang bersangkutan. RR ini terdiri dari :
    1. MNAME. Merupakan *primary name server* pada zone yang bersangkutan.
    2. RNAME. Berisi alamat email pengelola zone. Karakter @ dikodekan sebagai karakter titik.

3. SERIAL yang merupakan pengenal unik zone tersebut. Serial number merupakan angka 32 bit. Serial number harus dirubah bila master file dirubah. *Secondary name server* berpatokan pada *serial number* zone pada saat melakukan transfer zone. Bila *master file* dirubah maka *serial number* harus dirubah pula agar ditransfer oleh *secondary name server*.
4. REFRESH yang merupakan konstanta waktu interval bagi *secondary name server* untuk melakukan refresh dalam satuan detik. Dibentuk dalam format angka 32 bit.
5. RETRY yang merupakan konstanta waktu interval lamanya waktu tunggu bagi *secondary name server* untuk melakukan *refresh* bila mengalami kegagalan pada *refresh* sebelumnya. Dibentuk dalam format angka 32 bit.
6. EXPIRE. Merupakan batas waktu maksimum bagi *secondary name server* untuk menghapus *database* zone bila *primary name server* sudah tidak lagi aktif.

- ***Internet Address*** (IN A). RR ini berisi keterangan alamat IP milik *host* yang didefinisikan pada label.
- ***Name Server*** (IN NS). RR ini berisi daftar name *server* yang menyimpan struktur *tree* lengkap dari zone yang didefinisikan.

Sebuah contoh master file dapat dilihat pada Lampiran 2. File ini merupakan master file zone domain **linux.or.id**. *Primary name server* untuk zone ini ditunjukkan oleh RR SOA yaitu **merpati.indoglobal.com** dengan alamat IP **202.146.0.241**. *Secondary name server*nya adalah **ns1.web-dns.net** dengan alamat IP **202.137.3.50**. Sedangkan alamat IP host **linux.or.id** ditunjukkan oleh RR IN A yaitu **202.146.0.244.** (Lihat Gambar 3)



Gambar 3. Contoh Format Master File

Master file akan dibaca dan di*load* ke *database name server* pada saat dijalankan atau di*restart* ulang. Kemudian name *server* akan siap melayani *query* berdasarkan *master file* tersebut.

3.6 OSI-OSI Layer

International Organization for Standardi-zation (ISO) telah membakukan standard abstrak untuk menjelaskan interkoneksi network dalam berbagai lapisan (*layer*) operasi jaringan. Standard ini dikenal dengan *Open System Interconnection* (*OSI*) *Reference Model* dan diberi nomor ISO-7498 (Jungmaier 1996).

Dalam ISO-7498 ini dijelaskan tujuh lapisan operasi jaringan:

- ***Physical Layer***: menjelaskan media fisik yang digunakan dalam menyelenggarakan komunikasi data. Dalam layer ini dijelaskan sifat-sifat fisik mekanik, elektronik/optik, fungsi, dan prosedur pengiriman data pada alat yang digunakan.
- ***Data Link Layer***: menjelaskan bagaimana paket data dihantarkan dari suatu titik dalam jaringan ke titik yang lain. Dalam layer ini ini juga dijelaskan penggunaan pengecekan terhadap kesalahan pengiriman data.
- ***Network Layer***: menjelaskan cara-cara tersambungnya suatu jaringan dengan jaringan yang lain, penggunaan alamat dalam menyampaikan data, dan menentukan rute yang ditempuh dalam mengalirkan data.
- ***Transport Layer***: menjelaskan bagaimana agar data yang dialirkan dapat dengan sempurna sampai ke tujuan. Dalam layer ini, data dipaket-paket dalam kelompok yang disebut dengan *datagram* dan setelah sampai di tujuan *datagram* tersebut diperiksa kembali dan bila terjadi kesalahan maka pengiriman *datagram* diulangi.
- ***Session Layer***: menjelaskan bagaimana suatu hubungan komunikasi data dapat terlaksana antara dua node.
- ***Presentation Layer***: menjelaskan bagaimana data direpresentasikan ketika terjadi komunikasi data. Dalam layer ini data asli dapat dienkripsi atau dibentuk menjadi format yang berbeda dan dikembalikan menjadi bentuk asalnya setelah sampai ke tujuan.
- ***Application Layer***: dalam layer ini dijelaskan penerapan penggunaan komunikasi data dan protokol lainnya.


UNIVERSITAS KOMPUTER INDONESIA
UNIKOM