

Peranan Graf/Tree dalam sejarah perkembangan DNS Internet

Habibie Faried (13511069)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
¹13511069@std.stei.itb.ac.id

Abstract—Makalah ini membahas tentang sejarah perkembangan DNS dengan teorema graf/tree memiliki peranan penting di dalam sejarah tersebut. DNS adalah *domain name system* yang digunakan untuk mengkonversi nama sebuah mesin kedalam IP Address. DNS ini digunakan untuk mempermudah pemetaan dari nama mesin.

Index Terms—DNS, IPAddress, Cara Kerja, Aplikasi.

I. PENDAHULUAN

Pada tahun 1962, departemen pertahanan Amerika (DARPA) memutuskan untuk mengadakan riset tentang bagaimana caranya menghubungkan beberapa computer dari berbagai wilayah. Hal ini digunakan untuk menghindari pemusatan Informasi di satu titik (computer) karena jika satu computer hancur akibat peperangan, maka hilang sudah informasi yang berharga tersebut.

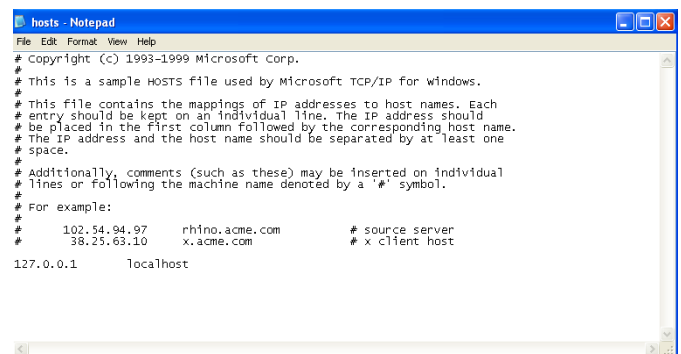
Untuk itu, diperlukan beberapa computer dari berbagai wilayah untuk menyimpan informasi tersebut dan meng-*update* informasi secara langsung jika salah satu informasi di computer wilayah lain ditambahkan/diubah. Untuk itu, perlu adanya koneksi antara computer-computer tersebut.

Pada tahun 1970, DARPA berhasil membuat jaringan computer antara 10 computer dengan wilayah berbeda. Lalu, sampai pada tahun 1984, jumlah pemakai computer yang tergabung dalam jaringan sekitar 10.000 user. Pada tahun ini juga sudah ada fitur bernama e-mail, dan TCP/IP Protocol dan IP Address.

Semakin berkembang dan banyaknya computer yang tergabung dalam jaringan, maka muncul satu masalah bahwa tidak mungkin seorang user dapat menghafalkan IP Address user lain yang sangat banyak. Untuk itu, diperlukan sebuah mekanisme untuk membuat “penyingkatan” dan “pemudahan” alamat IP Address tersebut.

Lalu, dibuat dan dikenalkan sebuah system yang bernama DNS (Domain name system). DNS digunakan untuk memetakan semua nama computer ke dalam IP Address.

Sehingga, pada saat itu nama-nama computer yang tergabung dalam jaringan tersebut diinput secara manual oleh *operator* disimpan secara persisten di dalam `/etc/hosts`. Berikut adalah format file tersebut yang ada di windows.



```
File Edit Format View Help
# Copyright (c) 1993-1999 Microsoft Corp.
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       192.54.94.97       rhino.acme.com       # source server
#       38.25.63.10      x.acme.com         # x client host
127.0.0.1                localhost
```

Misalkan waktu itu saya memiliki computer dengan IP Address 192.168.1.10. Maka, agar seorang *user* lain dapat mengingat computer milik saya, maka saya merequest kepada operator DNS dengan nama “habibieoke”.

Sehingga, *user* tentu lebih mudah mengingat “habibieoke” dibandingkan mengingat 192.168.1.10 dengan syarat *user* tersebut sudah mendownload file hosts dari operator DNS.

Namun, system DNS seperti ini tentu hal yang sangat merepotkan bagi *user* dan *operator* karena user harus mendownload file hosts setiap ada penambahan computer. Di sisi operator, dia juga harus menambahkan secara manual setiap ada penambahan computer.

Untuk itu, perlu adanya sebuah system DNS baru untuk mengatasi masalah ini. Lalu, lahirlah konsep

“otomatisasi” dalam system DNS ini. Jadi, user tidak perlu mendownload file hosts tersebut, user hanya perlu mengarahkan komputernya ke IP Address computer yang menangani DNS Server secara otomatis ini. Sehingga, user tidak perlu mengakses file hosts karena hal ini sudah ditangani oleh Server. Sampai saat ini, DNS yang sekarang dipakai adalah DNS dengan metode otomatisasi ini.

II. KOSAKATA DAN ISTILAH

Struktur Data ialah salah satu cabang ilmu informatika yang berguna untuk menyusun data-data sedemikian rupa agar *readable* dan *scalable*.

IP Address adalah suatu mekanisme yang digunakan untuk pengalamatan suatu mesin/computer dalam jaringan. Hal ini berfungsi persis seperti alamat rumah kita.

Array Traversal adalah mekanisme pencarian data di dalam *array of tuple* dengan cara traversal, yaitu memeriksa satu per satu sampai ketemu atau akhir dari *array of tuple*. Sangat tidak efektif jika datanya berjumlah banyak.

Authoritative DNS adalah DNS primer yang diwajibkan menjawab sebuah *request* computer yang ada dalam zona *authoritative DNS* tersebut.

DNS Zone adalah zona-zona wilayah yang memiliki *Authoritative DNS* masing-masing di setiap zona. Berikut adalah contoh zona-zona DNS.

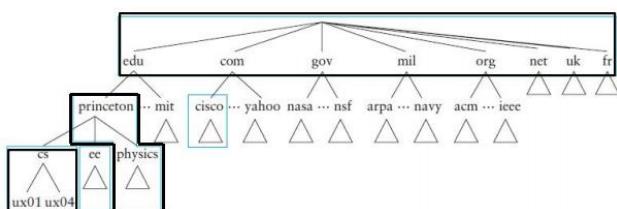


Figure 9.3 Domain hierarchy partitioned into zones.

Domain adalah sebuah nama unik yang berfungsi untuk menyingkatkan IP Address. Sehingga *user* tidak perlu menghafalkan IP Address suatu computer, akan tetapi cukup menghafalkan nama domain computer tersebut.

Root level domain adalah sebuah server pusat DNS yang merupakan *bapak/root* dari semua *Authoritative DNS* yang berada di zonanya masing-masing. Server ini menampung semua pemetaan dari domain ke IP Address seluruh dunia. Sehingga apabila *Authoritative DNS* suatu zona tidak bias menjawab request *user*, maka *user* akan diarahkan ke root level domain.

Top level domain adalah sebuah zona DNS yang berada langsung dibawah/anak dari root level domain. Contohnya domain .com, .id, .uk, .net, dll.

Secondary level domain adalah zona domain yang berada dibawah/anak dari top-level domain. Contohnya ialah facebook.com, google.com, etc.

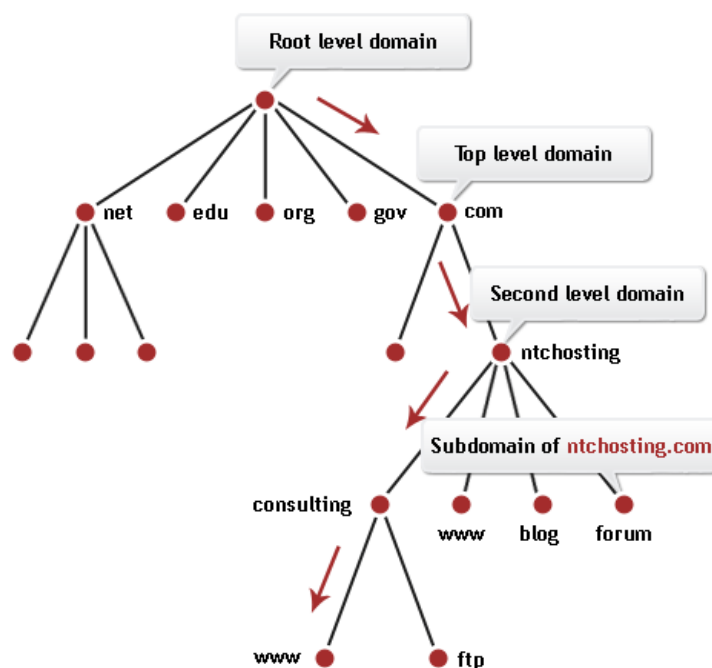
Sub-domain adalah seluruh domain yang berada dibawah/anak dari secondary level domain. Anak dari sub-domain adalah sub-domain. Sebagai contoh cs.princeton.edu, mail.cs.princeton.edu.

Name Resolution adalah suatu proses untuk mengkonversi dari Nama domain menjadi IP Address dengan bantuan DNS Server.

III. KELEBIHAN DNS SAAT INI

Kelebihan DNS saat ini adalah dikenalkannya istilah “*Authoritative DNS*”. Maksudnya ialah DNS Server terbagi di beberapa wilayah/zona, sehingga apabila ada computer yang ingin mengakses sebuah *hostname*, maka hanya perlu mengontak “*Authoritative DNS*” pada zona si *user*. Sehingga proses translasi ke IP Address berjalan lebih cepat.

Kelebihan lain ialah *hierarchical*. Maksudnya ialah DNS saat ini memiliki susunan untuk beberapa DNS dan Domain yang tergabung. Susunan ini berbentuk *n-ary tree*. Berikut adalah gambar susunan seluruh domain pada internet.



Jadi, sudah ada susunan tersendiri tentang penamaan DNS dengan mengimplementasikan *n-ary tree*. Sehingga pencarian sebuah domain akan lebih terstruktur dan lebih cepat. Penjelasan tentang cara kerja DNS akan dibahas nanti.

Kelebihan lainnya adalah pengelolaan terdistribusi. Maksudnya ialah DNS Server tidak terpusat pada root level domain, akan tetapi tersebar di beberapa zona. Bahkan, kita sendiri pun bisa membuat DNS Server Zone yang baru dengan mudahnya.

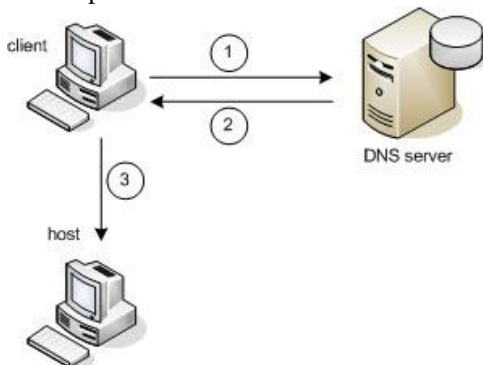
Kelebihan lainnya ialah apabila suatu domain mengubah IP Addressnya, maka nama domain tersebut tetap bisa diakses, misalkan *itb.ac.id* memiliki IP Address 167.205.3.3 berganti menjadi 167.205.1.3, maka hal yang perlu dilakukan hanyalah mengontak *Authoritative DNS Server (Zona ITB)* untuk mengganti *record databasenya* dan menghubungi root level domain untuk melakukan hal yang sama.

Jadi, dengan system baru seperti ini, tentu DNS sekarang memiliki kinerja yang lebih baik daripada DNS yang lama.

IV. OPERASI DAN CARA KERJA DNS SERVER

IV.1 Name Resolution

Berikut adalah cara kerja seorang user merequest sebuah pemetaan dari domain ke IP Address.



Jadi, didalam DNS itu sendiri ada yang bernama Client yang biasa disebut *resolver* dan DNS Server yang bernama *name server*. Jadi, cara kerjanya sebagai berikut:

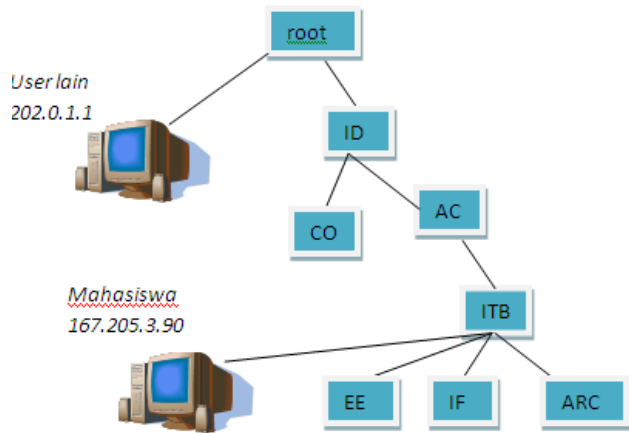
- 1) Pada computer Client, sebuah program aplikasi misalnya http, meminta pemetaan IP Address (*forward lookup query*) kepada DNS Server.
- 2) Name server mengecek apakah ada *record* tentang domain yang direquest oleh Client, jika ada *record* tentang domain tersebut, maka DNS Server akan

memberikan IP Address host yang dimaksud.

- 3) Jika tidak, maka name server akan mengembalikan IP Address root level domain, agar Client tersebut mengakses DNS Server di root level domain.

Tadi sudah dijelaskan bagaimana antara *resolver* dan *name server* bekerja, sekarang akan dijelaskan bagaimana pencarian domain pada struktur *n-ary tree* DNS yang telah terbentuk.

Misalkan ada seorang *user* dari zona wilayah lain (misalkan dari koneksi hotspot UI) ingin mengakses *arc.itb.ac.id* dengan struktur pohon *n-ary* DNSnya sebagai berikut :



Asumsikan IP Address user tersebut adalah 202.0.1.1 (bukan di dalam zona ITB (167.205.x.x)). Maka, langkah-langkah yang dilakukan computer tersebut ialah sebagai berikut :

- 1) Mengecek apakah domain tersebut ada didalam cache local computer nya. Apabila ada, maka langsung dapat diketahui IP Addressnya dan bisa diakses.
- 2) Apabila tidak ada, maka computer akan menghubungi *Authoritative DNS Server* pada zona dia berada (zona UI).
- 3) Jika tidak ditemukan *record database* domain *arc.itb.ac.id*, maka si *Authoritative DNS* akan menghubungkan computer tersebut ke root level domain.
- 4) Pada root level domain, tidak ditemukan *record* *arc.itb.ac.id*, akan tetapi ditemukan alamat top level domain DNS "id". Sehingga, computer *user* akan diarahkan ke computer DNS "id".
- 5) Pada DNS "id", tidak ditemukan adanya record sub-domain *arc.itb.ac.id*, akan tetapi ditemukan alamat secondary level domain "ac.id". Sehingga, computer *user*

akan diarahkan ke computer DNS “ac.id”

6) Begitu seterusnya sehingga ditemukan computer arc.itb.ac.id.

7) record disimpan didalam cache local komputernya, sehingga jika ingin mengakses arc.itb.ac.id kembali, maka computer hanya perlu membaca file cache milik kita untuk melakukan pemetaan IP Address arc.itb.ac.id.

Pada kasus selanjutnya (menggunakan skema yang sama), misalkan ada mahasiswa ITB ingin mengakses arc.itb.ac.id dengan computer yang terhubung ke Wi-fi ITB, maka hal yang dilakukan ialah

1) Mengecek apakah domain tersebut ada di cache local komputernya, apabila ada, maka dapat diketahui secara langsung IP Addressnya dan di akses langsung.

2) Jika tidak, maka computer mahasiswa ITB akan mengirimkan request untuk name resolution arc.itb.ac.id kepada *Authoritative DNS Server ITB*.

3) *Authoritative DNS Server ITB* mengenali arc.itb.ac.id karena masih dalam satu zona dan memang ditemukan adanya *record* yang memuat situs tersebut, sehingga IP Address dapat langsung diberikan dan diterima oleh computer mahasiswa.

4) Disimpan didalam cache local komputernya, sehingga jika ingin mengakses arc.itb.ac.id kembali, maka computer hanya perlu membaca file cache milik kita untuk melakukan pemetaan IP Address arc.itb.ac.id.

Dengan melihat 2 kasus diatas, kita dapat melihat bahwa cara kerja DNS Server mencari domain di internet mirip dengan algoritma *Binary Search*. Jadi, dengan menelusuri tiap simpul yang sesuai dengan urutan domain tersebut.

Jika domain yang dicari misalkan arc.itb.ac.id, maka urutan pencarian domainnya melalui simpul sebagai berikut :

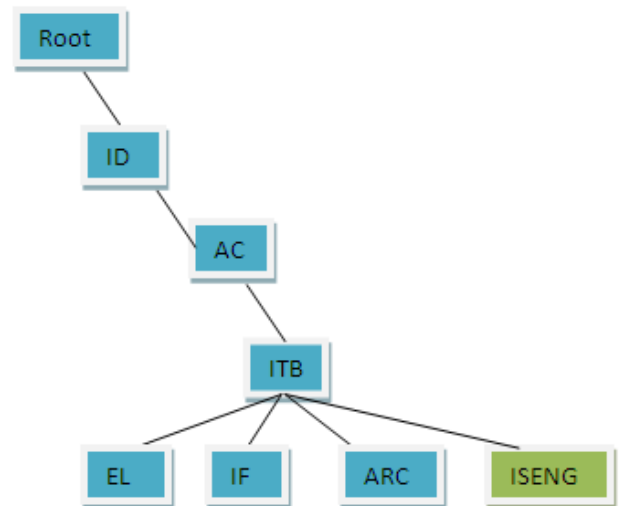
id → ac → itb → arc

IV.II Penambahan Domain baru

Pada zaman internet masih menggunakan system DNS yang lama, maka hal yang dilakukan adalah langsung mengontak *operator* agar menambahkan domain dan IP Addressnya/etc/hosts. Tentu ini merupakan hal yang cukup merepotkan bagi *user* maupun *operator*, bayangkan saja jika setiap hari *operator* menerima ratusan *request* untuk melakukan input record pada

/etc/hosts, tentu *operator* akan merasa kesulitan.

Untuk itu, pada system DNS yang baru, dikenalkan hal yang bernama zona DNS (*dns zone*). Jadi, misalkan saya ingin membuat domain baru bernama iseng.itb.ac.id dengan IP Address 167.205.3.16, maka saya tinggal mengontak *Authoritative DNS Server ITB* dan meminta agar domain dengan IP Address tersebut dimasukkan kedalam database DNS Server. Jika disetujui, maka struktur pohon DNSnya akan seperti ini :



Sehingga, dengan mekanisme seperti ini, operator/admin pada *root* tidak perlu kerepotan setiap ada penambahan domain yang baru. Karena yang bertanggung jawab terhadap domain baru tersebut ialah *Authoritative DNS Server* pada suatu zona DNS tertentu.

Analogi membuat domain baru mirip sekali dengan algoritma untuk menambah daun pada tree dari simpul yang telah ditentukan. Misalkan ada domain baru yang akan didaftarkan dengan nama tekno.com, maka hal yang dilakukan adalah mencari alamat simpul “com”, lalu tambahkan daun pada simpul tersebut, sehingga terbentuk pohon DNS yang baru.

V. KESIMPULAN

Sebelum adanya DNS, *user* internet diharuskan menghafal IP Address semua computer yang dianggap perlu olehnya. Hal ini tentu merupakan kesulitan sendiri bagi seorang *netter*.

Lalu, dibuatlah penyingkatan IP Address dengan membuat nama-nama mesin untuk mengganti IP Address tersebut, sehingga *user* hanya perlu menuliskan nama mesin tersebut untuk mengaksesnya.

Sistem DNS ini menggunakan/etc/hosts. *Operator* DNS menginput secara manual di file ini. Lalu, setiap periode tertentu *operator* akan membagikan file /etc/hosts/ ini kepada *user* internet. *User* mendownload file tersebut lalu ditaruh di /etc/.

Akan tetapi, lama kelamaan hal ini akan merepotkan *user* dan *operator juga*. Lalu dikembangkan lah system DNS baru. Ciri khas dari system ini adalah de-sentralisasi DNS Server, dan juga beberapa DNS Server ini memiliki bapak/parent bernama *root DNS Server* (pusat DNS) dan juga tadi juga membentuk n-ary tree.

Dengan cara ini, maka pencarian domain akan menjadi lebih mudah dikarenakan kita mencari domain seperti mencari daun melewati beberapa simpul (*DNS Server*) yang berkaitan dengan domain tersebut.

Jadi, model dan teorema dari graf/tree dapat meengubah system DNS agar menjadi lebih baik seperti saat ini.

REFERENCES

- [1] Peterson and Davie, *Computer Networks A System Approach*, Dartmouth: University of Massachussets, 2007, pp. 634–642.
- [2] <http://www.duasatu.web.id/2012/05/sejarah-perkembangan-internet-di.html>, Sabtu 08 Desember 2012 07.30.
- [3] <http://goopiet.blogspot.com/2008/07/cara-kerja-dns-server.html>, Sabtu 08 Desember 2012 08.00.
- [4] <http://ahmad-pravitno.com/2011/09/pengertian-top-level-domain-dan-secondary-level-domain/>, Sabtu 08 Desember 2012 08.20.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 09 Desember 2012



Habibie Faried dan 13511069