




Sistem Penamaan

Pertemuan 5

Overview

- ▶ Bab ini menerangkan mengenai konsep dari penamaan sebuah entitas dalam sebuah sistem tersebar di mana konsep nama yang independen dari cara mengakses entitas tersebut melahirkan beberapa cara penamaan seperti flat naming, structured naming, dan attribute based naming.
- ▶ Pada bab ini akan diterangkan pula teknik-teknik menemukan alamat sebuah entitas dari nama yang dimiliki entitas tersebut (name resolving).
- ▶ Mahasiswa juga akan melihat implementasi nyata dari penamaan ini pada DNS (*Domain Name System*) dan LDAP (*Lightweight Directory Access Protocol*).

Tujuan

1. Memahami definisi entitas, nama, alamat, dan identifier.
 2. Memahami sistem penamaan Flat Naming dan teknik-teknik resolving pada flat naming.
 3. Memahami sistem penamaan structured naming dan teknik-teknik resolving pada structured naming.
 4. Memahami sistem penamaan attribute based naming dan teknik-teknik resolving pada attribute based naming.
- 


1. Pentingnya Masalah Penamaan

- Penamaan berperan penting dalam banyak hal mengenai sistem komputer.
- Nama merupakan identitas yang dapat membedakan suatu entitas dengan entitas lainnya.
- Nama juga dapat menunjukkan lokasi suatu entitas.
- Masalah yang perlu diperhatikan dalam hal penamaan adalah bagaimana sebuah nama dapat menunjuk ke entitas yang diwakilinya

Sebuah NIM (Nomor Induk Mahasiswa) atau nama dapat menunjuk ke mahasiswa yang diwakili NIM atau nama tersebut.




Cont...

- Untuk mewujudkan hal tersebut dibuatlah suatu sistem penamaan.
 - Pada sistem tersebar implementasi sistem penamaan tidak berpusat pada sebuah komputer namun lebih sering tersebar pada beberapa mesin dan cara pendistribusiannya mempengaruhi efisiensi dan skalabilitas dari sistem penamaan tersebut.
- 

2. Nama, Identifier, dan Alamat

- Sebuah nama dalam sistem tersebar adalah sebuah deretean karakter yang digunakan untuk mewakili sebuah entitas.
- Entitas sendiri dapat berupa secara praktis dapat berarti apapun baik yang bersifat fisik seperti: komputer, printer, media penyimpanan, ataupun modem, maupun yang bersifat abstrak (logik) seperti: berkas (file), user, proses, mailbox, dsb.
- Untuk memanfaatkan entitas–entitas dalam sistem tersebar pengguna (manusia ataupun mesin) perlu mengakses entitas–entitas tersebut melalui sesuatu yang disebut access point, address, atau alamat

Cont...

- Sebuah entitas dapat memiliki beberapa alamat seperti halnya seseorang dapat memiliki beberapa nomor telepon genggam.
 - Contoh lainnya adalah ketika seseorang berpindah tempat seperti kota atau negara maka nomor teleponnya sering harus diubah sesuai dengan sistem di kota (kode area) atau negara (kode negara) tersebut.
- 

Cont...

Jenis nama lain yang mendapat perlakuan khusus selain alamat adalah identifier dengan sifat-sifat sebagai berikut:


1. mewakili paling banyak satu entitas,
2. setiap entitas diwakili oleh paling banyak satu identifier,
3. sebuah identifier selalu mewakili entitas yang sama (tidak berubah menurut waktu dan kondisi).

Satu lagi jenis nama penting adalah nama dengan sifat *user-friendly* yang mudah dibaca dan diingat oleh manusia. Nama ini biasanya terdiri dari sederetan karakter yang dikenali manusia seperti nama pada file atau nama yang digunakan pada Domain Name System seperti www.dinus.ac.id Atau www.google.co.id, an sebagainya.


Cont...

- Solusi yang lebih sering digunakan adalah dengan menyimpan informasi pasangan nama dan alamat tersebut pada beberapa sistem terpisah yang bersifat rekursif. Contohnya untuk mengetahui alamat www.yahoo.com :
 1. Pengguna perlu menghubungi sebuah sistem (yaitu .(dot) atau root) yang akan memberikan alamat dari sistem di bawahnya yang memiliki tabel informasi mengenai alamat dari nama-nama yang berakhiran dengan com (DNS domain com).
 2. Kemudian menghubungi sistem tersebut (DNS domain com) yang akan memberikan alamat dari sistem yang memiliki tabel informasi mengenai alamat dari nama-nama yang berakhiran yahoo.com (DNS domain yahoo.com).
 3. Untuk kemudian mengulangi lagi menghubungi sistem (DNS domain yahoo.com) yang memiliki tabel berisi pasangan nama www.yahoo.com dengan alamatnya.

Cara memasang alamat dengan nama atau identifier dan menemukan alamat pasangan dari nama atau identifier dan sebaliknya ini dikenal dengan ***Naming system***


- Cara memasang alamat dengan nama atau identifier dan menemukan alamat pasangan dari nama atau identifier dan sebaliknya ini dikenal dengan naming system. Ada tiga naming system , yaitu:
 1. Flat Naming
 2. Structured Naming
 3. Attribute-Based Naming
- 

3. Flat Naming

- Flat Naming adalah sistem penamaan yang tidak memiliki struktur tertentu. Nama dalam Flat Naming terdiri dari sederetan bit karakter yang tidak mengandung informasi tentang bagaimana menemukan alamat untuk entitas yang diwakili oleh nama tersebut.
- 

3.1. Teknik Resolving

Cara penamaan ini memang sederhana tetapi menimbulkan akibat yang tidak sederhana dalam cara menemukan alamat entitas yang diwakilinya. Ada beberapa solusi dalam menyelesaikan masalah menemukan pasangan alamat/menerjemahkan nama menjadi alamat (resolving) pada sistem penamaan flat naming, yaitu:

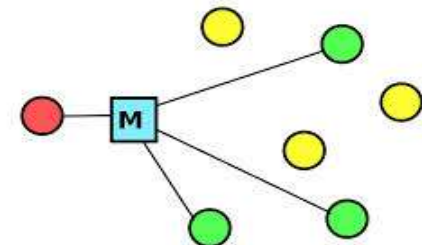
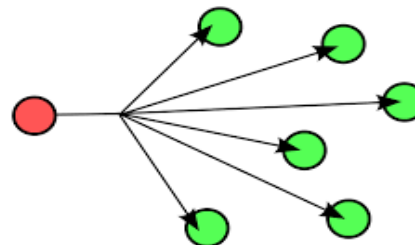
1. Broadcasting dan multicasting
 2. Forwarding pointer
 3. Home-based approach
 4. Hierarchical search tree
- 

3.1.1. Broadcasting dan Multicasting

- ▶ Cara ini cukup sederhana yaitu mengirimkan sebuah pesan berisi permintaan pasangan identifier dari sebuah alamat kepada seluruh atau sebagian anggota jaringan dan hanya entitas yang memiliki alamat tersebut yang akan menjawab dengan identifier yang ia miliki.

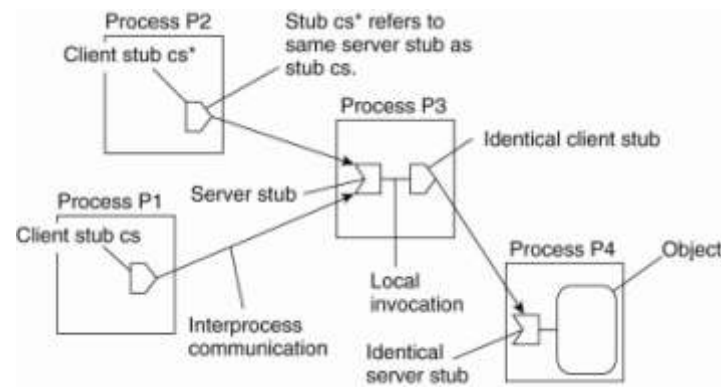
Contoh implementasinya adalah ARP (address resolution protocol) yang digunakan untuk mengetahui alamat hardware (MAC Address) berdasarkan sebuah alamat IP.

- ▶ Cara lainnya adalah dengan melakukan multicasting yaitu:
 - Mengirimkan pesan hanya ke beberapa entitas dalam jaringan. Contoh kasusnya adalah pada unit bergerak seperti laptop milik pegawai yang terkoneksi dengan jaringan nirkabel.
 - Ketika laptop pegawai terkoneksi ke jaringan khusus pegawai laptop tersebut akan mendapatkan alamat IP milik staff berdasarkan alamat hardwarenya.
 - Ketika pegawai lain mencoba menghubungi laptop tersebut maka pesan dikirimkan hanya ke komputer yang tergabung dalam jaringan pegawai



3.1.2 Forwarding Pointers

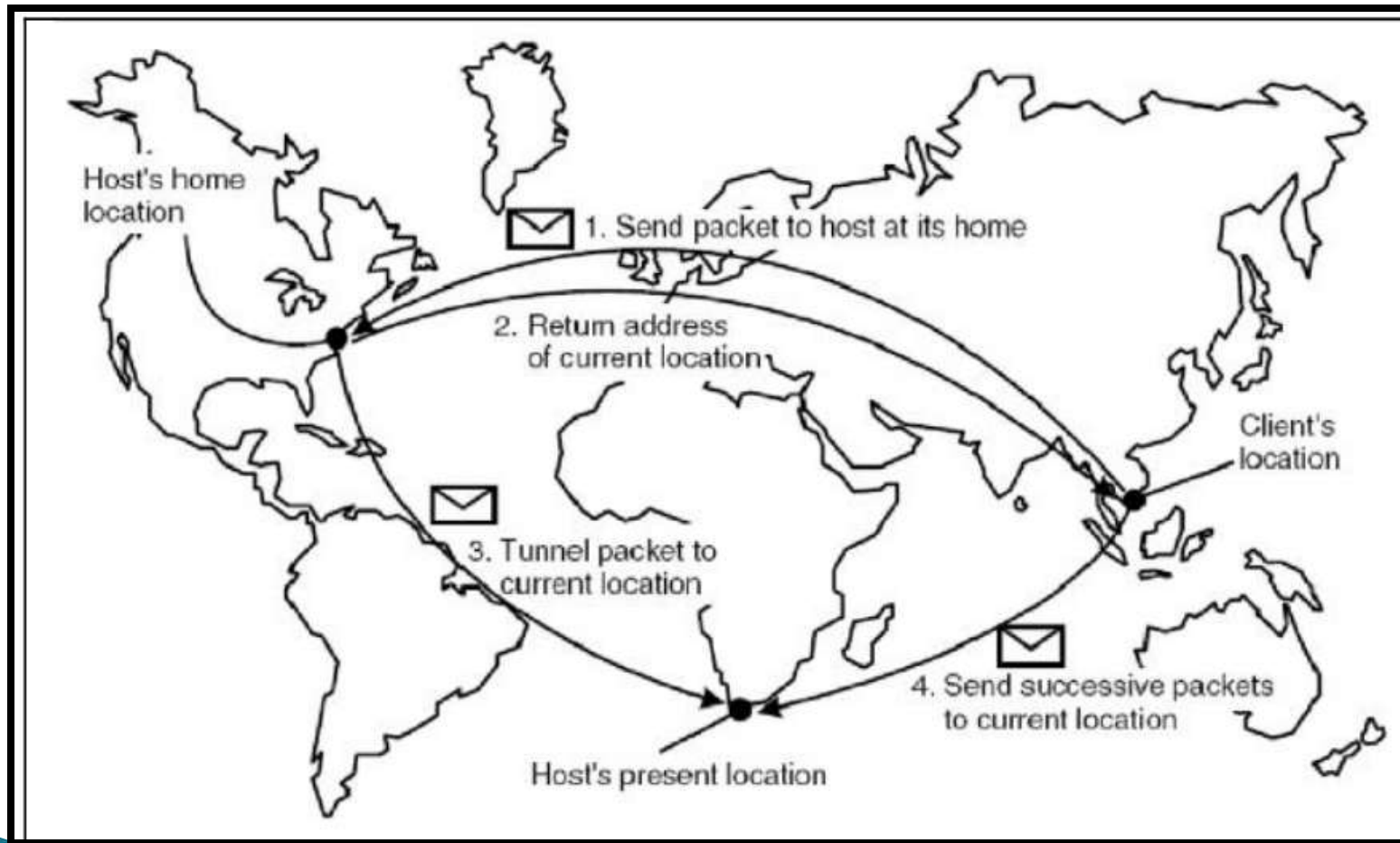
- ▶ Pada teknik ini setiap kali sebuah entitas berpindah lokasi dan mendapatkan alamat baru maka entitas tersebut meninggalkan informasi mengenai lokasi barunya di lokasi lamanya sedemikian sehingga entitas lain yang mencarinya dapat menelusuri jejak perpindahan dan berkomunikasi dengan entitas tersebut. Informasi tersebut dikenal sebagai pointer.
- ▶ Permasalahan timbul ketika entitas bergerak cukup sering dengan jarak yang semakin jauh.
- ▶ Rantai pointer akan semakin panjang sehingga penelusuran dapat memakan sumber daya lebih besar dan mengakibatkan penurunan kinerja.




3.1.3. Home-based Approach

- ▶ Alternatif teknik lain yang tidak mengalami permasalahan ketika entitas bergerak dalam jaringan skala besar adalah pendekatan home-based, yaitu sebuah teknik resolving sedemikian sehingga sebuah entitas bergerak memiliki entitas agen dengan sebuah alamat tetap yang menjadi 'alamat rumah'.
- ▶ Ketika entitas tersebut bergerak ke lokasi lain entitas tersebut akan mendapatkan alamat baru yang kemudian didaftarkan entitas agen yang berada di 'alamat rumah'.
- ▶ Ketika sebuah pesan dikirimkan untuk entitas bergerak tersebut tujuan dari pesan adalah 'alamat rumah'. Entitas agen juga akan memberitahu pengirim pesan alamat baru dari entitas bergerak tersebut sehingga komunikasi berikutnya terjadi antara pengirim dengan entitas bergerak di lokasi barunya.


3.1.3. Home-based Approach

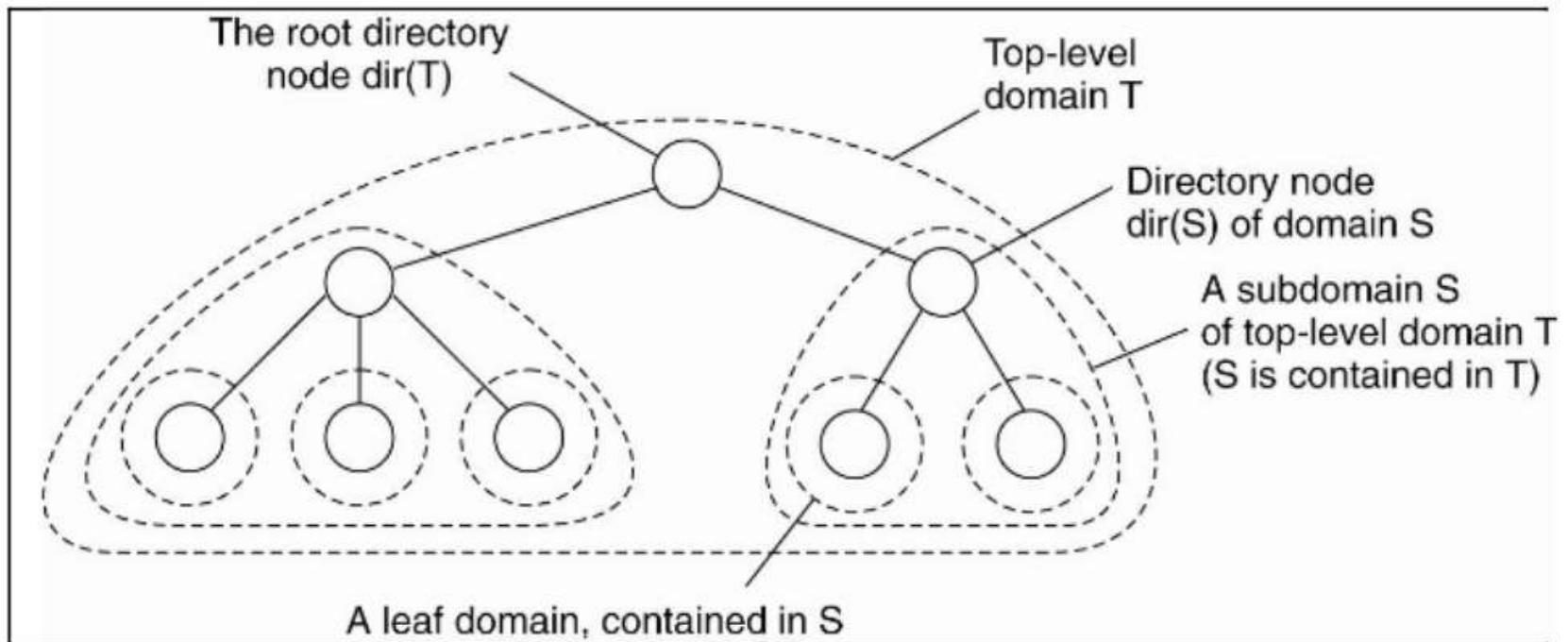


Gambar 4.2 Prinsip Mobile IP

- Kekurangan teknik ini adalah penggunaan lokasi rumah yang tetap sehingga rumah ini harus dipastikan selalu tersedia.
 - Jika entitas bergerak hendak pindah untuk jangka waktu lama atau menetap di lokasi lain maka ada baiknya lokasi rumah juga ikut berpindah.
 - Solusi masalah ini adalah dengan mendaftarkan lokasi rumah pada sebuah layanan penerjemahan alamat (***naming service***) jika lokasi rumah dapat berpindah sehingga lokasi rumah dapat ditemukan oleh entitas yang memerlukannya melalui naming Service tersebut.
- 

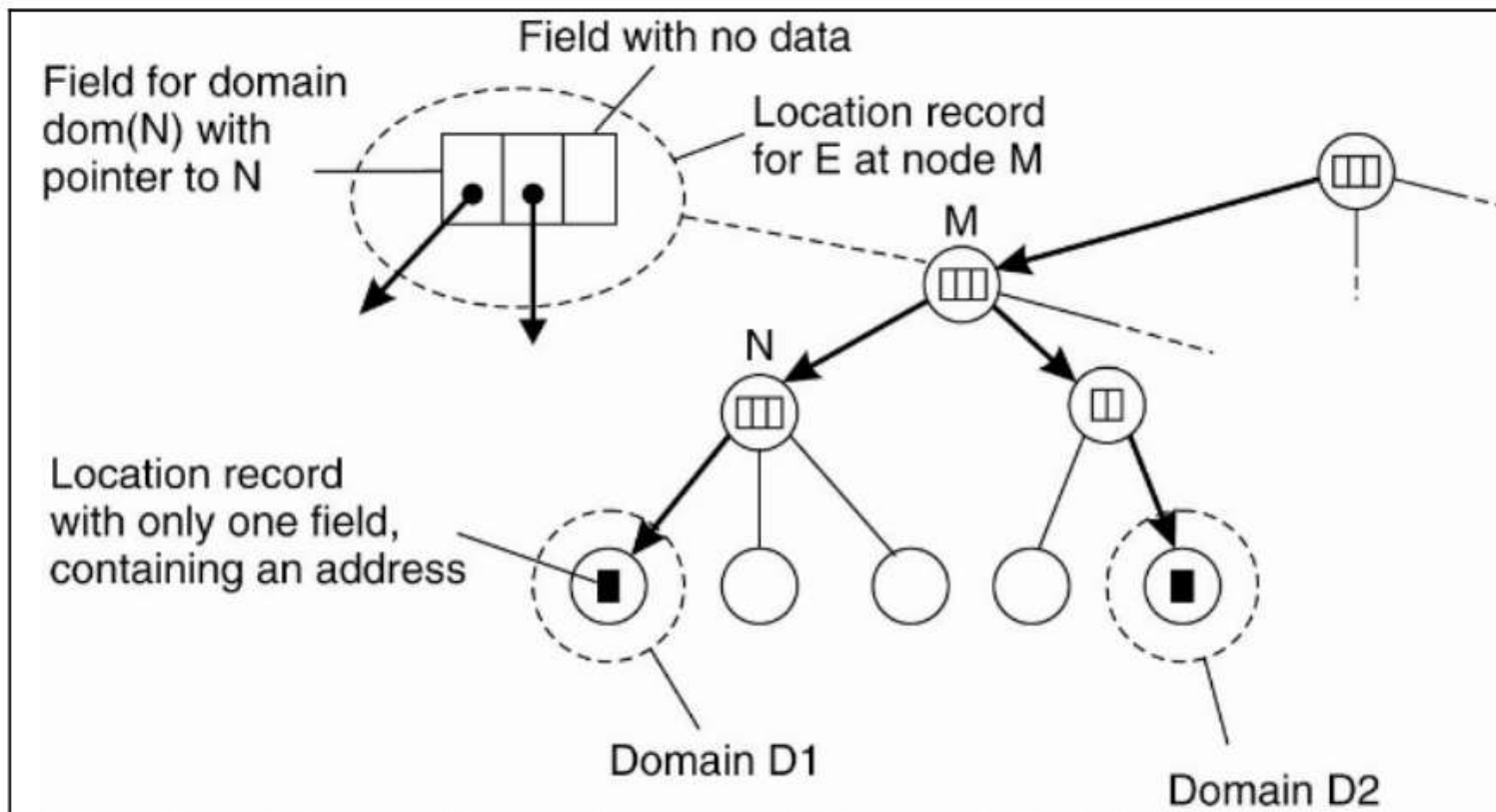
3.1.4 Hierarchical Search Tree

- ▶ Pada teknik ini jaringan dibagi-bagi menjadi beberapa bagian yang dikenal sebagai domain. Domain-domain ini berada di dalam sebuah domain yang berada di tingkat paling atas yang meliputi seluruh jaringan (top-level domain).
 - ▶ Sebuah domain yang berada di bawah/dalam sebuah domain lain dikenal sebagai sub-domain. Domain yang berada pada posisi paling bawah dan tidak memiliki sub-domain lagi dikenal sebagai leaf-domain.
 - ▶ Setiap domain D memiliki simpul direktori, $\text{dir}(D)$, yang menyimpan informasi lokasi setiap entitas dalam domain tersebut. Akibatnya terbentuk sebuah pohon direktori dengan simpul paling atas disebut simpul direktori akar (root directroy node) yang mengetahui semua entitas.
- 




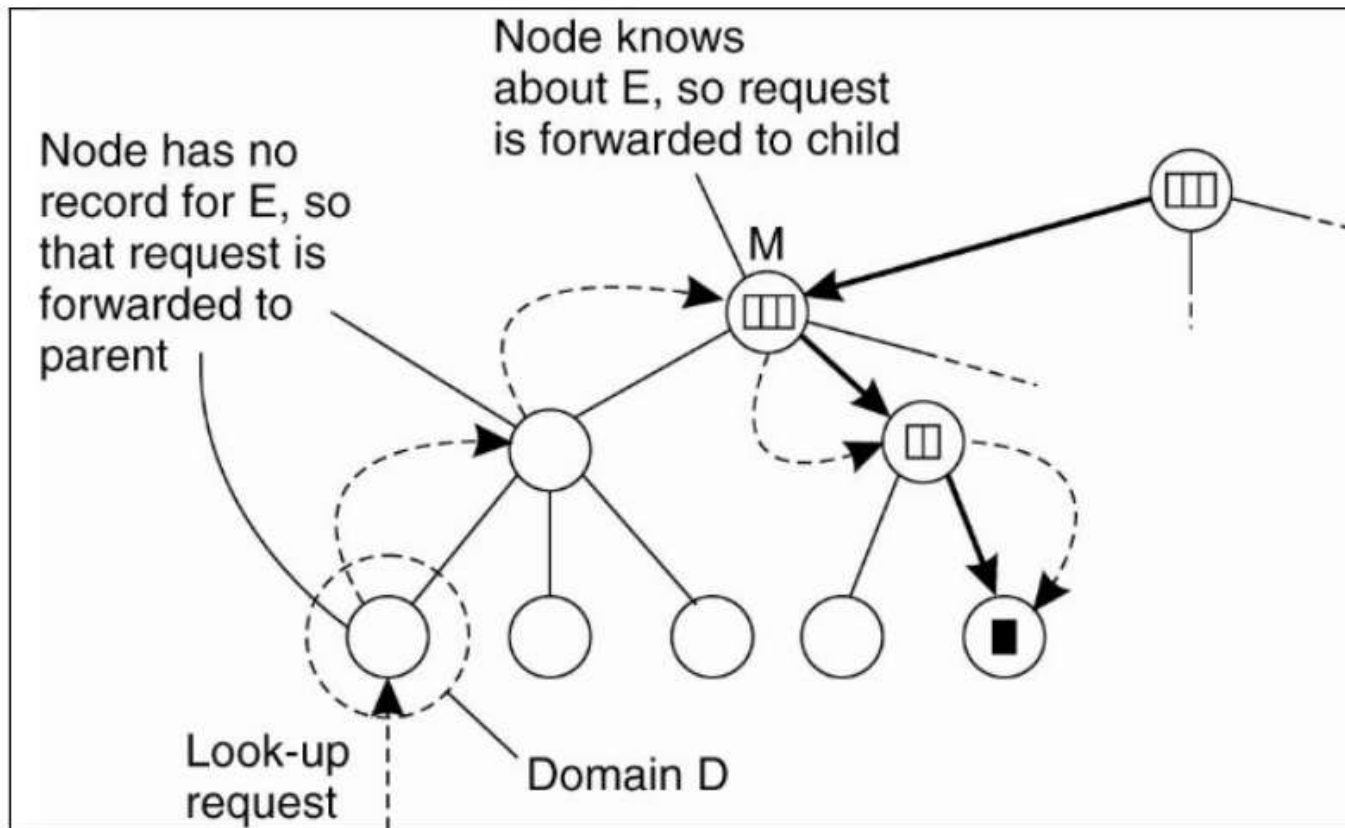
Gambar 4.3 Hirarki organisasi dari layanan lokasi di dalam domain, masing-masing berasosiasi dengan node direktori

- Untuk melacak lokasi sebuah entitas dalam sebuah domain setiap entitas dalam domain terdaftar di dalam simpul direktori domain tersebut. Catatan lokasi(location record) sebuah entitas E pada simpul direktori N dari domain D berisi alamat entitas tersebut di dalam domain D.
- Sebaliknya, simpul direktori N' dari domain D' yang menjadi domain induk dari domain D hanya akan menyimpan catatan lokasi E berupa penunjuk (pointer) ke simpul direktori N dan demikian juga induk dari N' hanya menyimpan penunjuk ke simpul direktori N'. Pada akhirnya simpul direktori root berisi penunjuk ke simpul direktori dari semua sub-domain di bawahnya.
- Sebuah entitas dapat memiliki beberapa alamat sebagai contoh: jika entitas tersebut direplikasi. Jika sebuah entitas memiliki alamat di leaf-domain D1 dan D2 maka simpul direktori dari domain induk dari domain D1 dan D2 menyimpan dua penunjuk untuk setiap sub-domain.




Gambar 4.4 contoh penyimpanan informasi dari tiap entitas yang memiliki 2 alamat pada cabang yang berbeda

- ▶ Jika sebuah entitas E1 ingin mengetahui alamat entitas E2 maka pertama kali ia akan mencari catatan lokasi entitas tujuan tersebut di simpul direktori domain tempat ia berada. Jika pada simpul direktori ini tidak terdapat catatan lokasi entitas E2 berarti entitas E2 tidak berada pada domain yang sama dengan entitas E1.
 - ▶ Permintaan entitas E1 akan diteruskan ke simpul direktori dari simpul direktori dari induk domain tempatnya berada dan seterusnya ke simpul direktori induk dari induk domainnya hingga menemukan catatan lokasi untuk E2. Pada kasus terburuk pencarian akan berlanjut hingga simpul direktori domain akar.
 - ▶ Jika pada salah satu domain ditemukan catatan lokasi untuk E2 maka permintaan pencarian akan diteruskan ke sub domain yang ditunjuk oleh catatan lokasi untuk E2 tersebut seterusnya hingga ditemukan alamat dari E2.
- 




Gambar 4.5 memandang suatu lokasi di dalam layanan lokasi secara hirarkis

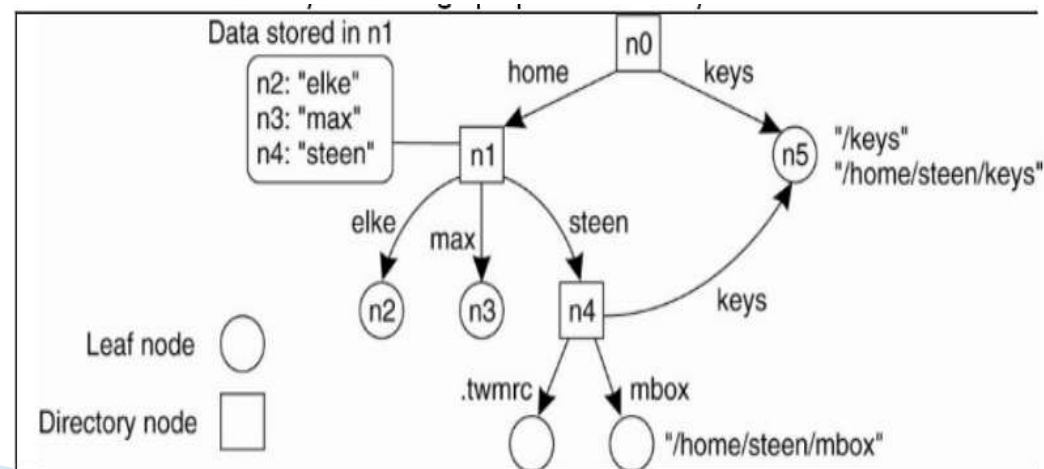
4. Structured Naming

- Flat name baik untuk menamai mesin tetapi biasanya tidak cukup nyaman untuk digunakan oleh manusia. Sebagai alternatifnya sistem penamaan biasanya mendukung penggunaan nama yang terstruktur yang dibentuk dari beberapa nama yang sederhana dan mudah dikenali manusia. Penamaan pada sistem file dan host pada internet mengikuti kaidah ini.
- 

4.1 Name Spaces


- Nama-nama biasanya diatur menjadi sesuatu yang dikenal sebagai ruang nama (name space). Name space untuk nama yang terstruktur dapat direpresentasikan dalam bentuk graph (sebuah diagram yang terdiri dari simpul (node) dan sisi (edge) yang menghubungkan dua simpul) yang memiliki label (nama) dan arah.
 - Ada dengan dua macam simpul pada graph yang merepresentasikan name space, yaitu:
 1. Simpul daun yang merepresentasikan entitas yang memiliki nama dan tidak menjadi induk dari simpul lainnya.
 2. Simpul direktori yang memiliki ujung-ujung yang bernama dan menunjuk pada dari simpul daun lain.
- 

- ▶ Simpul direktori menyimpan sebuah tabel yang disebut directory table
- ▶ yang berisi pasangan nama sisi (edge) yang mengarah ke luar menuju pada sebuah simpul anak dan identifier dari simpul anak tersebut. Simpul yang menjadi induk dan tidak memiliki induk disebut sebagai simpul root (akar).
- ▶ Sebuah graph penamaan dapat memiliki beberapa root namun demi kesederhanaan biasanya sebuah graph penamaan hanya memiliki satu root.



Gambar 4.6 Grafik penamaan yang umum dengan satu *root* node

- Setiap jalur (path) menuju suatu entitas dapat ditulis sebagai urutan label-label dari setiap sisi yang mengarah pada entitas tersebut seperti $N:\langle \text{label-1}, \text{label-2}, \dots, \text{label-n} \rangle$ di mana N adalah simpul di awal jalur. Jika sebuah jalur diawali dengan simpul root maka jalur tersebut disebut absolute path dan jika tidak diawali simpul root maka dinamakan relative path.
- Contoh mudahnya adalah jalur ke sebuah file pada banyak sistem berkas (file System) ditulis sebagai urutan label dari sisi-sisi (melambangkan direktori-jangan tertukar dengan istilah simpul direktori) yang mengarah pada file tersebut seperti `c:\windows\System32\cmd.exe` atau `/home/faculty/test.sh`


- Ada beberapa cara berbeda untuk mengatur sebuah name space namun kebanyakan hanya memiliki satu simpul root dan pada banyak kasus bersifat hierarkis dalam artian name space tersebut memiliki struktur seperti pohon (tree).
 - Pada pengorganisasian seperti itu semua simpul memiliki tepat satu sisi yang mengarah ke simpul tersebut (kecuali simpul root) dan dapat memiliki beberapa sisi yang mengarah ke luar dari simpul tersebut ke simpul di bawahnya sehingga setiap simpul memiliki tepat satu absolute path.
- 

4.2 Name resolution

- Name resolution merupakan istilah untuk proses pencarian (Looking up) sebuah nama ketika kita mendapatkan path dari nama tersebut. Proses name resolution ini akan menghasilkan identifier dari sebuah simpul yang dilalui pada proses tersebut.

4.2.1 Mekanisme Penutupan (Closure)

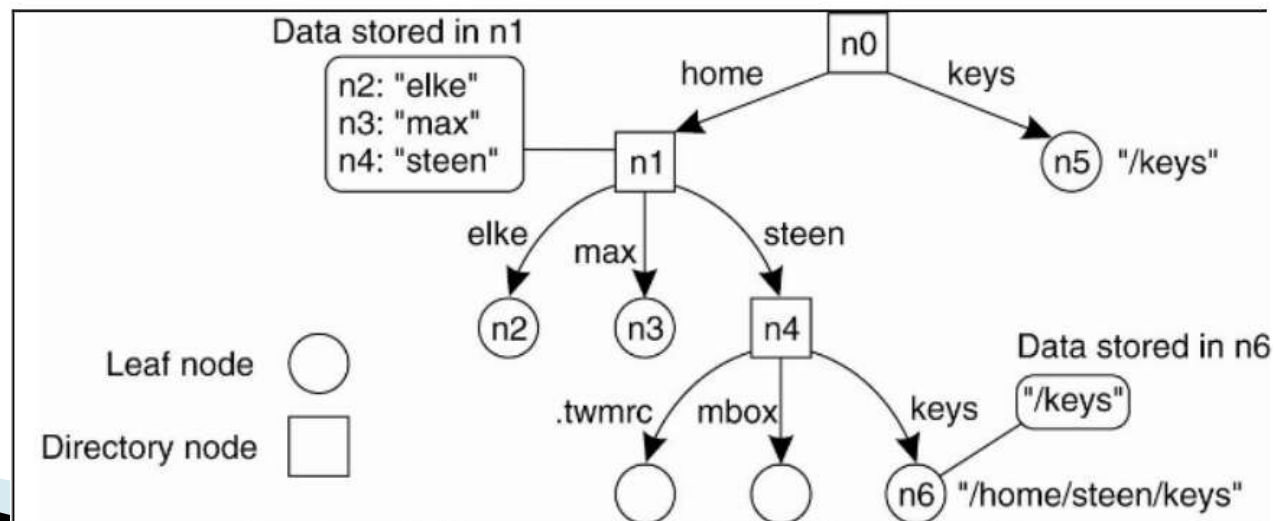
- ▶ Proses name resolution hanya dapat terjadi apabila pencari mengetahui bagaimana dan di mana untuk memulai proses. Hal ini dikenal sebagai **closure mechanism** yang secara esensi berurusan dengan menentukan simpul awal dalam name space di mana proses pencarian bisa dimulai.
- ▶ Closure mechanism terkadang tidak tampak (implicit) dan sudah menjadi bagian dari sebuah sistem dan dapat berbeda-beda pada setiap sistem. Sebagai contoh proses name resolution pada graph penamaan di file System UNIX menggunakan aturan bahwa inode (simpul) dari direktori root adalah inode (simpul) pertama.
- ▶ Jika sebuah file bernama /home/steen/mbox ingin ditemukan maka sistem operasi harus sudah dapat mengakses inode root untuk memulai pencarian.

- Contoh lainnya dari closure mechanism ialah jika diberikan sebuah deretan angka “00301231751” banyak yang tidak mengetahui apa yang harus dilakukan kecuali ada informasi tambahan bahwa deretan angka tersebut adalah nomor telepon.
 - Informasi tersebut cukup untuk memulai sebuah proses name resolution dengan cara memanggil nomor tersebut melalui pesawat telepon dan sistem telekomunikasi telepon akan melakukan proses lookup.
- 

4.2.2 Linking dan Mounting

Terkadang suatu entitas memiliki beberapa nama atau dikenal sebagai alias. Pada sebuah graph penamaan terdapat dua cara untuk mengimplementasikan konsep alias, yaitu:

1. Dengan mengizinkan adanya lebih dari satu absolute path menuju suatu simpul (pada dunia file System UNIX ini dikenal sebagai hard link)
2. atau dengan merepresentasikan sebuah entitas dengan sebuah simpul daun yang berisi informasi berupa absolute path dari entitas tersebut. Cara ini ke-dua dikenal dengan istilah symbolic link.

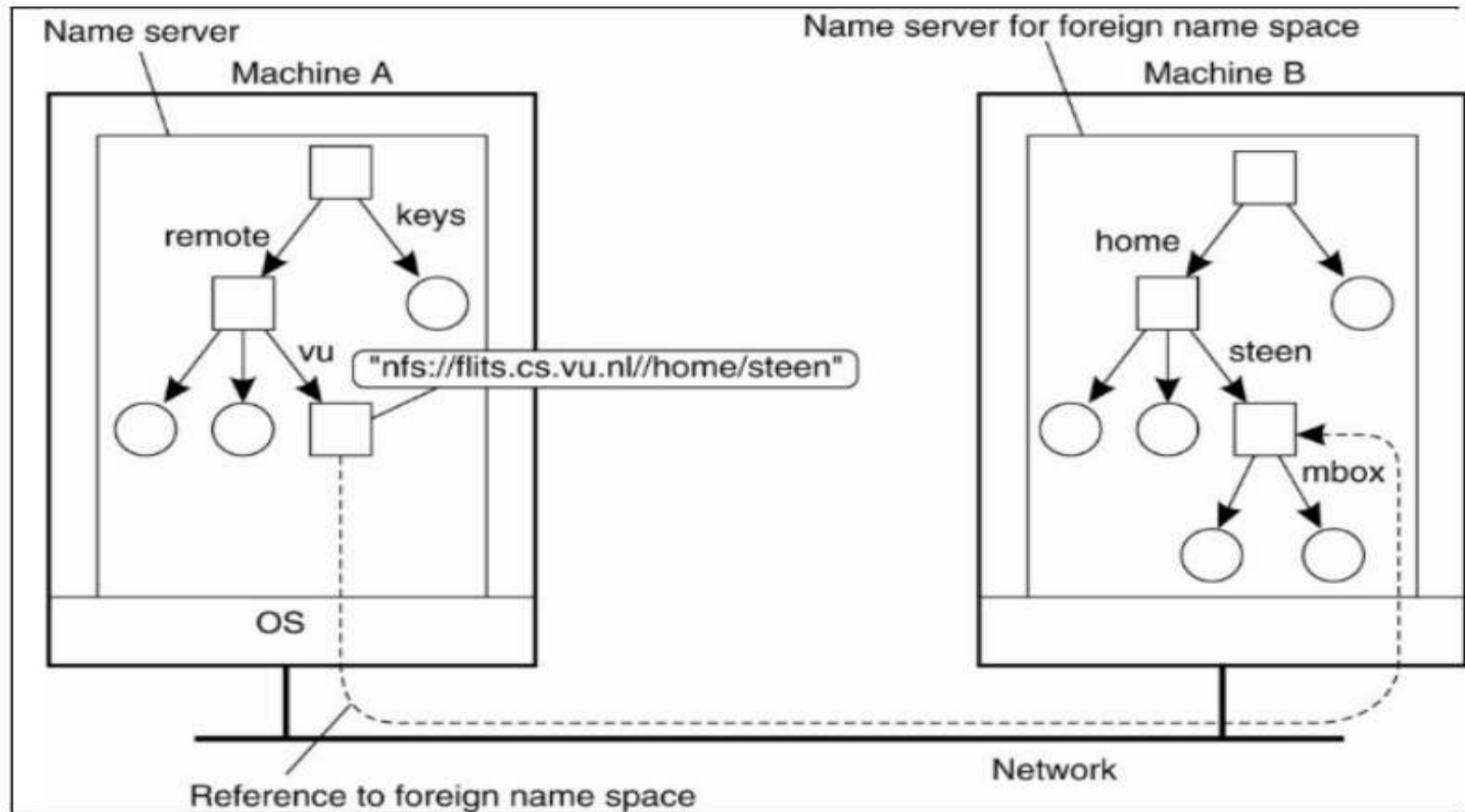


Gambar 4.7 kosep secara simbolik proses link


Cont...

- Proses name resolution yang dicontohkan sejauh ini berada dalam sebuah name space. Name resolution dapat juga melibatkan dua atau lebih name space.
- Contohnya adalah mounted file System pada UNIX. Proses mounting terjadi ketika sebuah simpul direktori menyimpan identifier dari simpul direktori yang berada di naming space yang berbeda atau dikenal sebagai simpul direktori asing.
- Simpul direktori yang menyimpan identifier simpul direktori asing disebut sebagai mount point sedangkan simpul direktori asing dikenal sebagai mounting point.
- Pada sebuah sistem tersebar proses mounting dapat dilakukan lintas komputer jika didukung oleh tiga informasi berikut ini, yaitu:
 1. nama dari protocol aksesnya,
 2. nama dari komputer lain,
 3. Nama mounting point yang berada pada name space yang dimiliki komputer tersebut.
- Contoh implementasinya adalah NFS (Network File System) yang dikembangkan oleh Sun Microsystems.

Gambar 4.8 penggunaan remote name space s melalui akses terhadap protokol secara spesifik



4.3 Implementasi Name Space

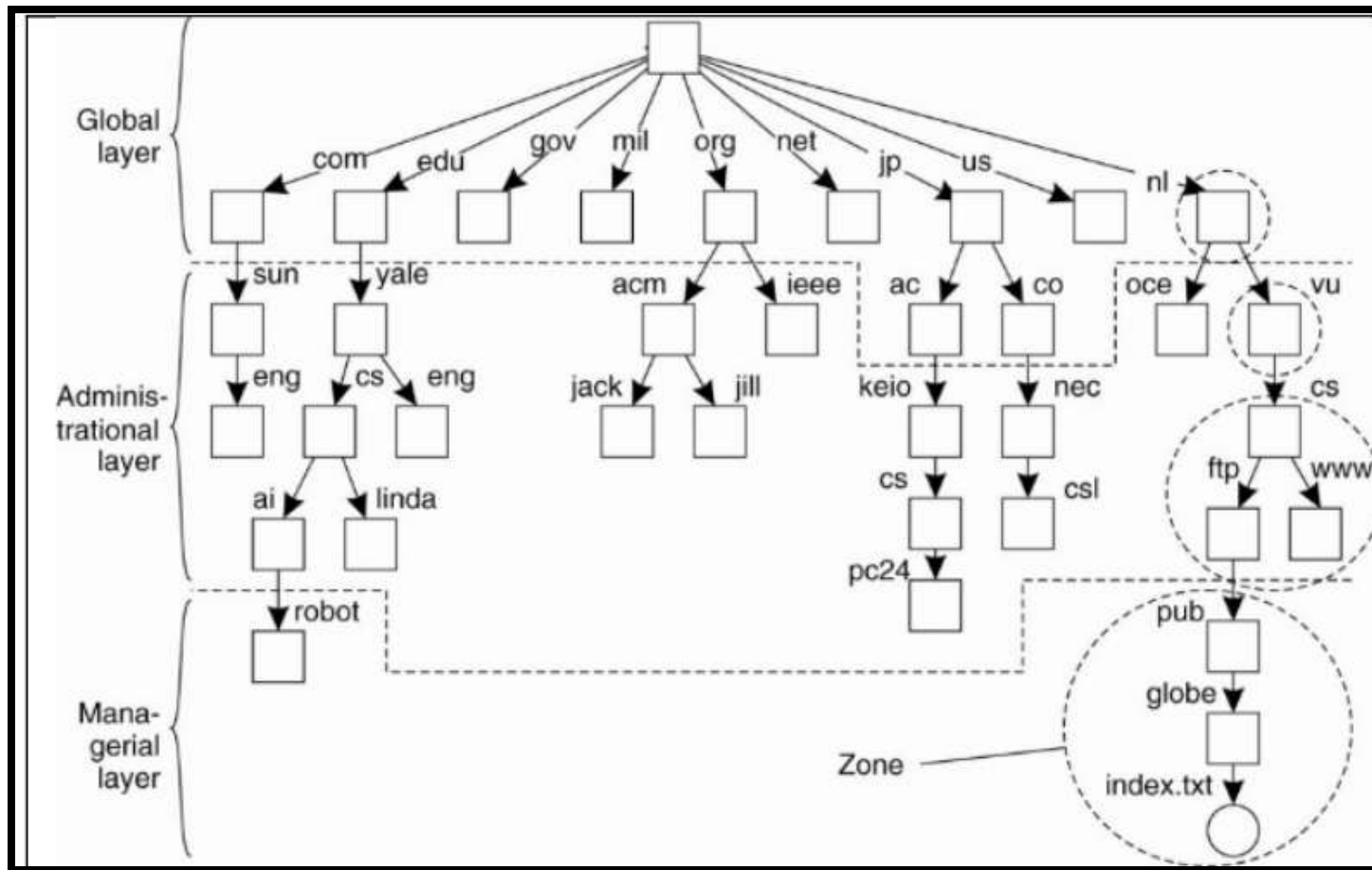
- Name space adalah inti dari layanan penamaan, yaitu sebuah layanan yang digunakan oleh pengguna (manusia maupun komputer) untuk menambah, mengurangi, dan mencari nama-nama.
 - Pada sebuah local area network layanan ini biasanya disediakan oleh sebuah server (name server) saja namun pada sistem terdistribusi skala besar layanan ini melibatkan beberapa name server.
- 

4.3.1 Distribusi Name Space

Name space yang digunakan pada sistem terdistribusi skala besar dan mungkin melibatkan seluruh dunia (contohnya: internet) biasanya disusun dalam bentuk hierarkis dengan sebuah simpul root. Pada implementasinya dengan alasan efektivitas name space dibagi menjadi tiga lapisan logik, yaitu:

1. lapisan global yang terdiri dari simpul root dan anak-anaknya yang merepresentasikan organisasi atau sekumpulan organisasi dan isinya jarang berubah,
2. lapisan administrasional yang dibentuk oleh sekumpulan simpul direktori yang diatur dalam sebuah organisasi (berada dalam organisasi yang sama) seperti: simpul yang menggambarkan sebuah departemen dalam sebuah organisasi,
3. lapisan manajerial merupakan lapisan terakhir yang berisi simpul-simpul yang sering berubah. Biasanya berisi simpul-simpul inimerrepresentasikan komputer-komputer dalam sebuah LAN atau file-file biner dan pustaka yang dibagi pakai dalam jaringan.

Gambar 4.9 Distribusi Name Space



4.3.2 Implementasi Name Resolution

- ▶ Distribusi name space pada beberapa server berimplikasi pada cara melakukan name resolution atau name lookup. Untuk memperjelas uraian digunakan contoh absolute path name berikut: root :

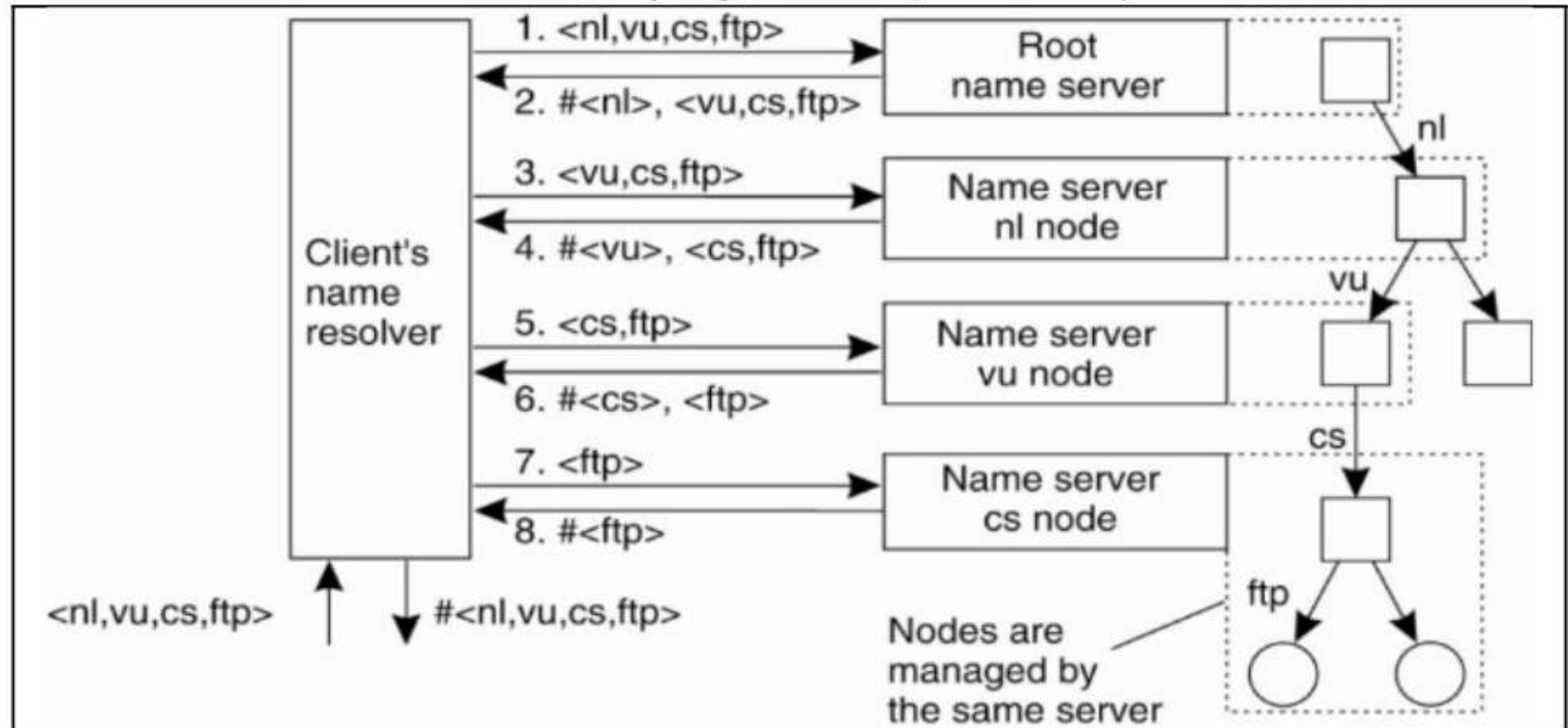
`<id, ac, poltek, ftp, pub, index.html>`

- ▶ Teknik name resolution pertama adalah iterative name resolution di mana entitas yang ingin melakukan name resolution pada sebuah nama menyerahkan absolute path name tersebut kepada name resolver lokal.
- ▶ Name resolver lokal kemudian menghubungi server root yang hanya dapat melakukan resolution hingga label “id” saja. Server root akan mengembalikan alamat name server simpul “id” kepada name resolver lokal yang kemudian akan menghubungi name server simpul “id” untuk mendapatkan alamat name server simpul “ac” dengan mengirimkan:

`id:<ac, poltek, ftp, pub, index.html>`

- ▶ Proses resolving akan terus berulang hingga pada name server pada simpul “poltek” yang akan mengembalikan alamat server FTP. Pada gilirannya server FTP akan melakukan resolving terhadap sisa path name, yaitu: pub dan index.html dan mentransfer file yang diminta (index.html).

Gambar 4.10 Iterative name resolution




5. Attribute Based Naming

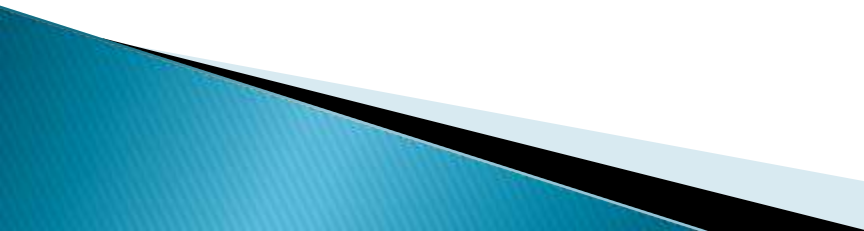
- ▶ Penamaan secara datar (flat) dan terstruktur (structured) secara umum memberikan sebuah cara untuk mereferensi suatu entitas–yang unik dan tidak bergantung pada lokasi (location independent).
- ▶ Penamaan terstruktur juga didesain untuk memberikan kemudahan bagi manusia untuk mengingat dan menggunakannya. Seiring bertambahnya informasi yang dapat diakses dari sebuah entitas maka diperlukan sebuah cara agar pengguna dapat mencari dan mengakses suatu entitas dengan memberikan keterangan mengenai entitas tersebut.
- ▶ Salah satu cara yang populer adalah menggunakan attribute based naming (penamaan berbasiskan atribut). Dengan teknik ini sebuah entitas diasosiasikan dengan sejumlah atribut yang memiliki nilai tertentu. Pengguna melakukan pencarian berdasarkan kriteria tertentu berupa jenis dan nilai atribut yang dimiliki entitas yang dicarinya.

5.1 Directory Service

- Sistem penamaan berdasarkan atribut lebih sering dikenal sebagai directory service sedangkan sistem penamaan terstruktur lebih dikenal sebagai naming System.
- Pada directory service entitas memiliki beberapa atribut yang dapat digunakan sebagai kriteria pencarian. Pada beberapa kasus pemilihan atribut-atribut yang akan dilekatkan pada sebuah entitas relatif sederhana.
- Contohnya sebuah e-mail dapat memiliki atribut pengirim, penerima, judul, tanggal, dan sebagainya yang dapat menjadi kriteria pencarian. Pemilihan atribut untuk e-mail menjadi lebih rumit ketika fitur penyaringan (filtering) ditambahkan di mana penyaring akan memilih e-mail yang boleh masuk berdasarkan atribut-atribut yang berkenaan dengan konten seperti: tidak bervirus, bukan spam, dan lain sebagainya.

- Contoh di atas memperlihatkan bahwa pemilihan atribut untuk sebuah entitas bukanlah hal yang dapat dianggap enteng.
 - Meskipun dibuat konsensus tertentu mengenai atribut apa yang harus dipilih untuk sebuah entitas pengalaman membuktikan bahwa pengisian nilai untuk suatu atribut secara konsisten oleh pihak-pihak yang berbeda ternyata membawa masalah tersendiri seperti pada kasus pencarian musik atau video di database internet.
 - RDF atau Resource Description Framework adalah salah satu cara yang dikembangkan untuk menyelesaikan masalah
- 

5.2 Implementasi Hierarkis: Lightweight Directory Access Protocol

- ▶ LDAP atau Lightweight Directory Access Protocol adalah contoh nyata dari implementasi sistem directory service. Para pengguna sistem operasi jaringan dari Microsoft seperti Microsoft Windows NT, Windows 2000, dan Windows 2003 lebih mengenal Active Directory sebagai implementasi dari LDAP sedangkan para pengguna sistem operasi turunan UNIX seperti Free BSD dan Linux lebih mengenal open LDAP sebagai implementasi dari LDAP.
 - ▶ Pada LDAP terdapat sejumlah record yang dikenal sebagai directory entries. Directory entries dapat dibandingkan dengan resource record pada DNS. Setiap record berisi sekumpulan pasangan atribut dan nilai atribut tersebut. Setiap atribut juga diasosiasikan dengan tipe tertentu dan dibedakan antara atribut bernilai tunggal dan atribut bernilai banyak.
- 

- ▶ Sekumpulan directory entries dalam LDAP disebut sebagai directory information base (DIB). Pentingnya DIB adalah bahwa setiap record memiliki nama yang unik dan dapat dicari yang dikenal dengan relative distinguished name (RDN).

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Servers	—	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	—	130.37.20.20
WWW_Server	—	130.37.20.20

- Seperti halnya pada DNS penggunaan RDN menghasilkan hirarki kumpulan directory entries yang dikenal sebagai directory information tree (DIT)



**Sekian
Terima Kasih
DAN
Apakah ada
PERTANYAAN?**