

Layanan Penamaan
(Name Service)
dan
Direktori
(Directory Service)

Pengertian [1]

- Dalam sistem terdistribusi, nama digunakan untuk menunjuk ke suatu resource yang beragam dan tersebar seperti komputer, layanan (services), file dan *remote object*.
- Layanan penamaan berfungsi untuk menyimpan **naming context**.
 - Naming context yaitu himpunan keterkaitan antara nama dan atribut objek: seperti user, komputer, *services dan remote object*.

Pengertian^[2]

- Pengaksesan **resource** pada **sistem terdistribusi** memerlukan:
 - Nama resource (untuk pemanggilan),
 - Alamat (lokasi resource tsb),
 - Pemetaan antara nama dan alamat sebenarnya
 - Rute (bagaimana mencapai lokasi tsb)

Manfaat penamaan

- **Komunikasi:** misal, nama domain sebagai bagian dari email
- **Resource sharing:** proses tidak dapat mengakses suatu resource, jika resource tersebut tidak diberi nama
- **Location independence:** perubahan lokasi tidak menuntut perubahan nama, asalkan lokasi tidak menjadi bagian dari nama resource tsb.
- **Security:** jika sebuah nama dipilih secara acak, maka nama tersebut hanya bisa diketahui dari legitimate source, Jadi jika seseorang mengetahui nama obyek tersebut, maka dia memang diberitahu.

Jenis Nama

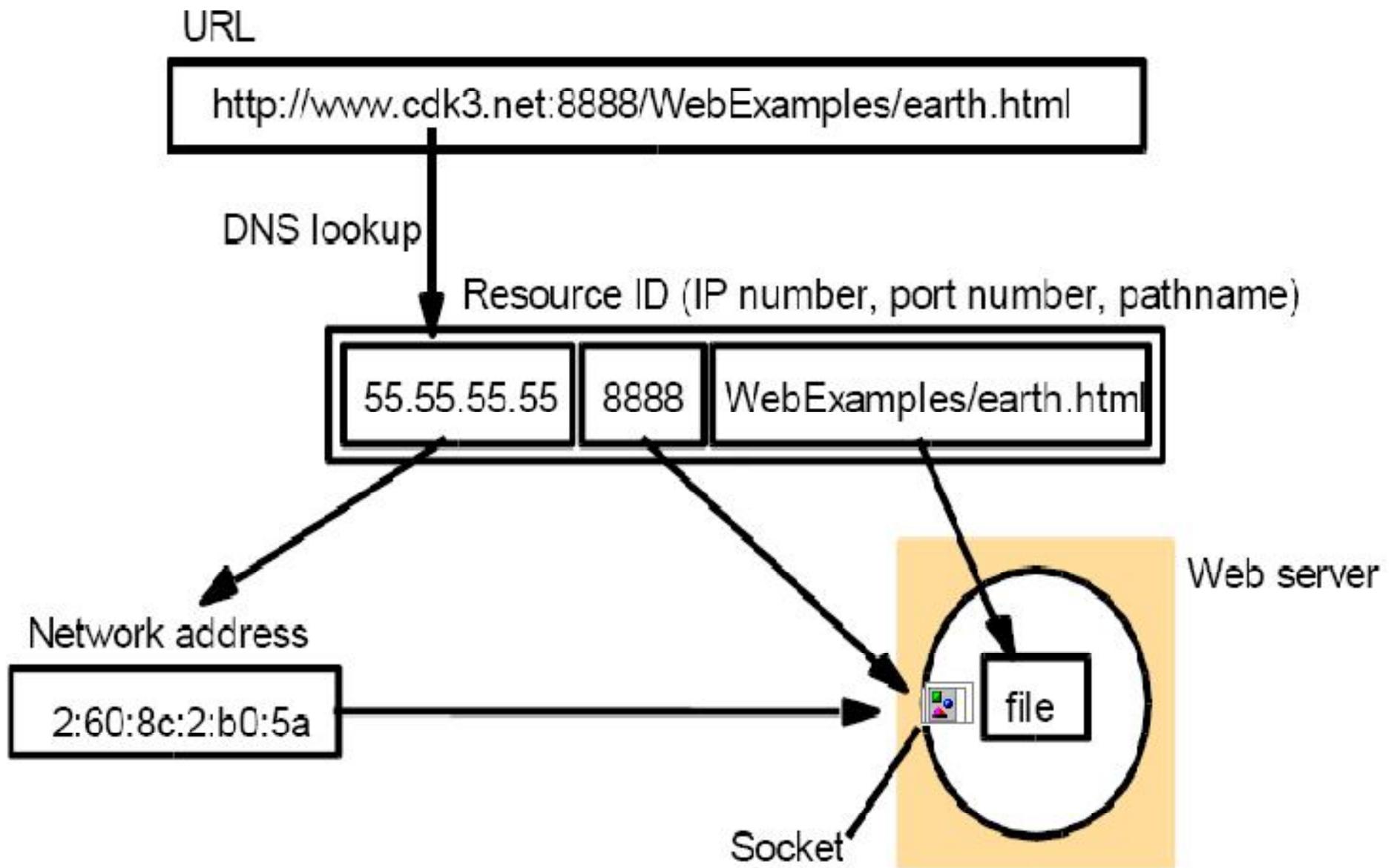
- **Pure name** : nama yang tidak perlu di terjemahkan, karena pada nama tersebut sudah menunjuk alamat objek langsung. Contoh : IP
- **non-pure name** : dalam nama mengandung suatu informasi (atribut misalnya) tentang suatu objek. Contoh : URL, alamat email

Daftar istilah

- **Name resolution:**
 - Nama ditranslasikan ke data tentang resource/object tersebut.
- **Binding:**
 - Asosiasi / Pengikatan antara nama & obyek.
 - Biasanya nama diikat (bound) ke atribut dari suatu obyek tertentu.
- **Address:** alamat dari sebuah entitas dalam sistem terdistribusi
- **Attribute:** nilai property suatu object.

Contoh Pemetaan Atribut

- **DNS:** memetakan dari nama ke atribut alamat IP host
- **X.500 directory service:** memetakan suatu nama seseorang ke beberapa atribut, seperti email, telepon, dsb.
- **CORBA Naming Service** yang memetakan nama *remote objek ke remote object reference*.



Penjelasan

- **URL (Uniform Resource Locator)** merupakan suatu tipe khusus **URI (Uniform Resource Identifier)**.
- Tipe lainnya adalah **URN (Uniform Resource Name)**. Ide dengan adanya URN adalah user dapat melakukan query berdasarkan URN untuk mendapatkan URL objek.
- **URC (Uniform Resource Characteristics)** merupakan subset dari URN untuk mendeskripsikan suatu sumber web dengan suatu atribut, contoh 'author=andi', 'keywords=sister,...'

Name Service

- Name service menyimpan kumpulan satu atau lebih *kontek penamaan* – yaitu himpunan keterkaitan antara nama dan atribut objek, seperti user, komputer, *services* dan *remote object*.
- Kebutuhan terhadap Name Services
 - Penamaan unik yang standard
 - Scalability
 - Consistency
 - Performance dan availability
 - Mudah menyesuaikan terhadap perubahan

Name Server

- *Name server* menggunakan basis data terdistribusi yang terdiri dari *tuples* <nama, lokasi, [atribut]>. Contoh atribut: jika *resource* adalah *printer*, maka atribut dapat menyatakan apakah obyek dapat melakukan pencetakan *postscripts* atau tidak.
- Hal penting yang harus dimiliki:
 - *Availability*,
 - Konsistensi,
 - Kecepatan menerima pengaruh perubahan *name lists*,
 - Kemudahan mengkompilasi *list of objects (resources)*.

Ada 2 jenis *Name Server*

- Ada 2 jenis *Name Server*:

A. *Primary Name Server*

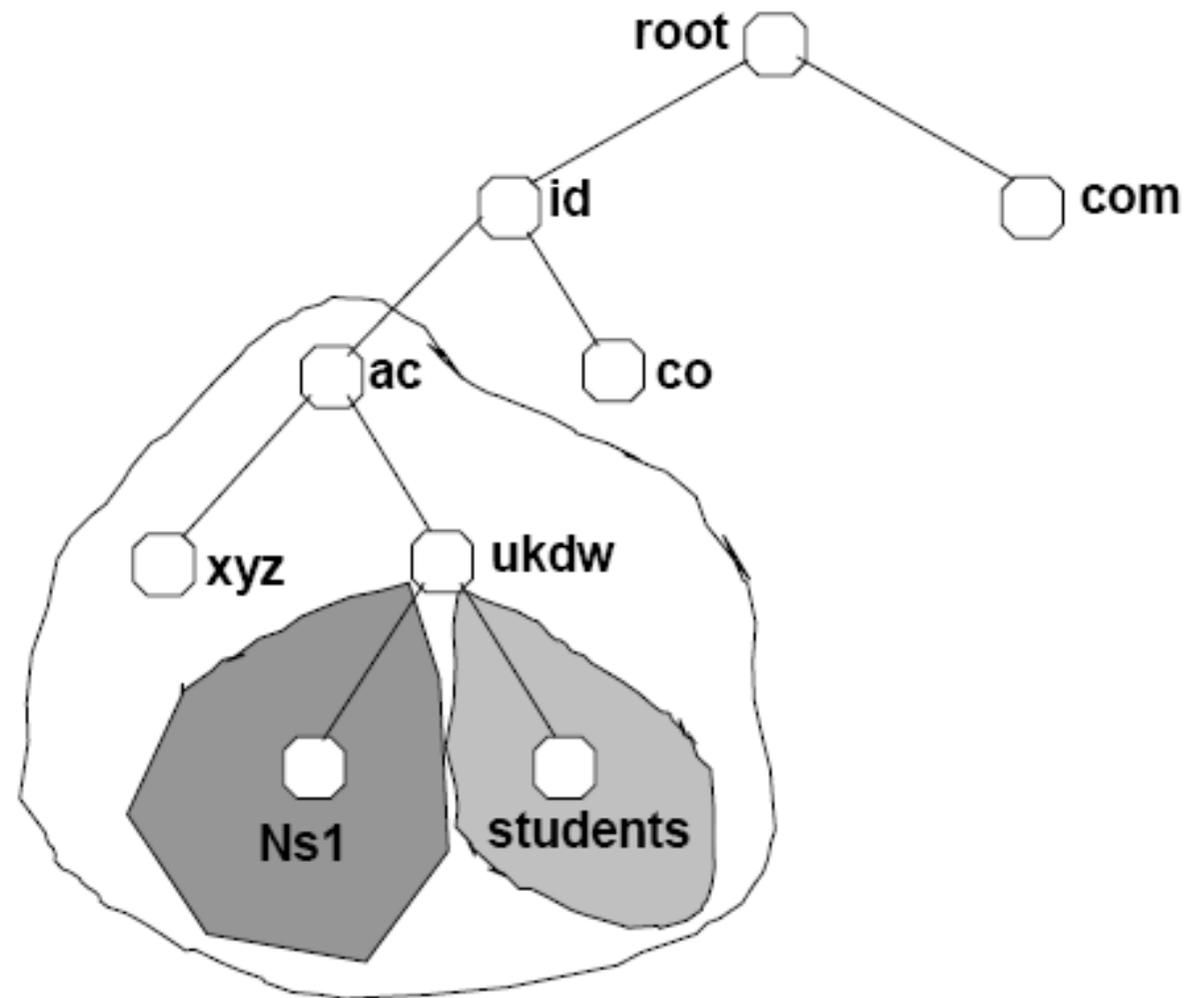
- Mendapatkan data dengan membaca file di storage. lebih dikenal dengan File Zone

B. *Secondary Name Server*

- Mendapatkan data dengan mereplikasikan data yang ada di primary server. Lebih dikenal dengan Transfer Zone

- Mekanisme name server :
 - **Partitioning**
 - Tidak ada satu name server yang dapat menyimpan seluruh nama dan atribut untuk seluruh jaringan.
 - Data nama dipartisi berdasarkan domain
 - **Replication**
 - Sebuah domain biasanya memiliki lebih dari satu name server untuk meningkatkan availability dan performance
 - **Caching**
 - Sebuah name server dapat melakukan mekanisme caching terhadap data nama dari name server lain untuk mencegah operasi permintaan yang sama secara berulang-ulang

Name Service



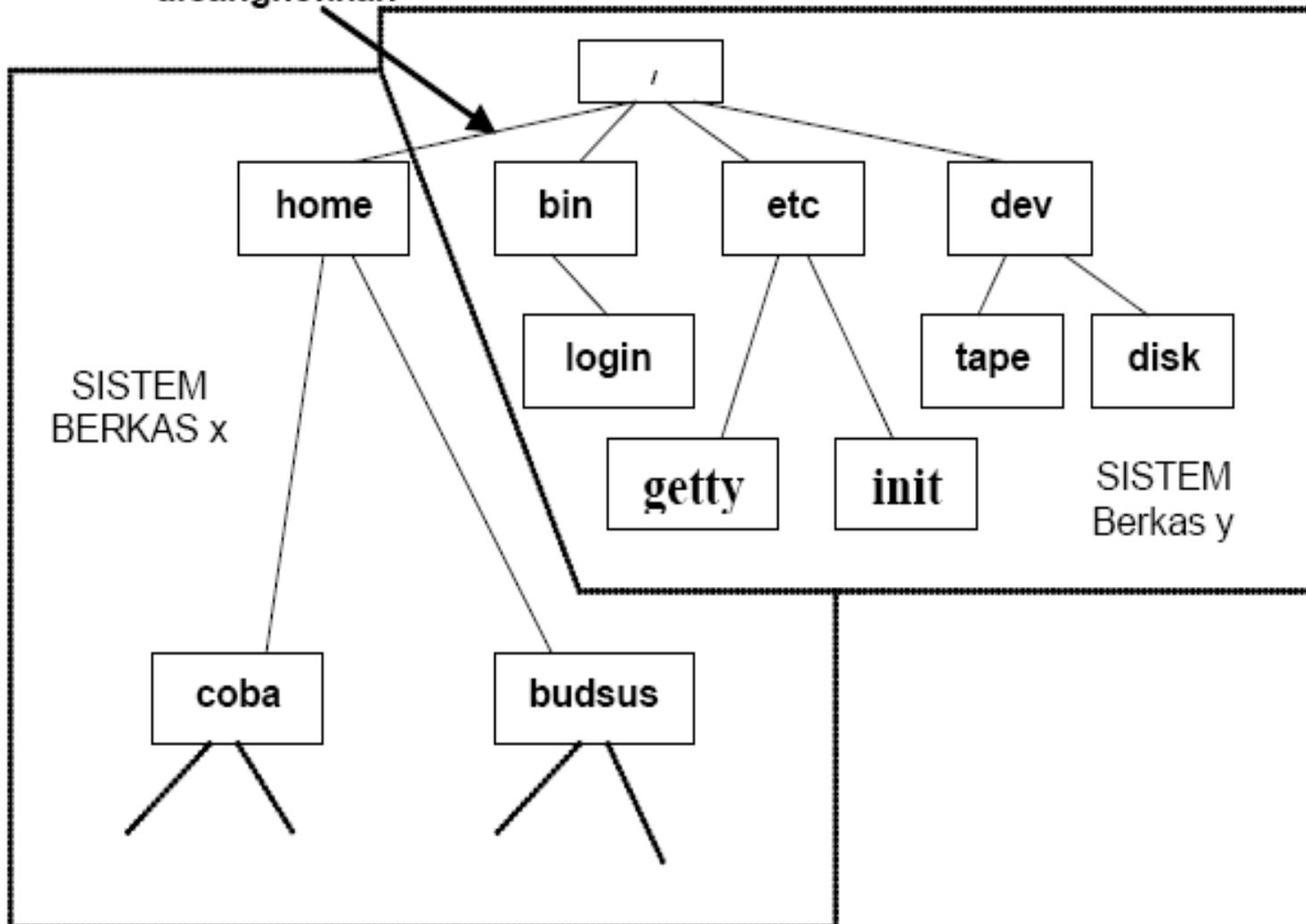
Name Space

- ***Name Space*** adalah kumpulan nama yang sah, yang dikenal oleh suatu layanan yang sesuai.
- Suatu nama dapat digambarkan dengan menggunakan generative grammar, seperti BNF (Backus-Naur Form).
Contoh BNF untuk email

Berikut beberapa karakter name space

- Memiliki struktur internal
 - Flat
 - Hirarki yang mempresentasikan posisi (contoh sistem berkas UNIX)
 - Hirarki yang mempresentasikan struktur organisasi (contoh Internet Domain)
- Karena hirarki, memiliki potensi tak terbatas, suatu nama dapat memiliki suatu nama alias.
 - `www IN CNAME ns1.dani.com`
- Suatu name space dapat berupa gabungan dari beberapa name space lain. Contoh mount pada UNIX/Linux

dicangkokkan



Domain Name Services

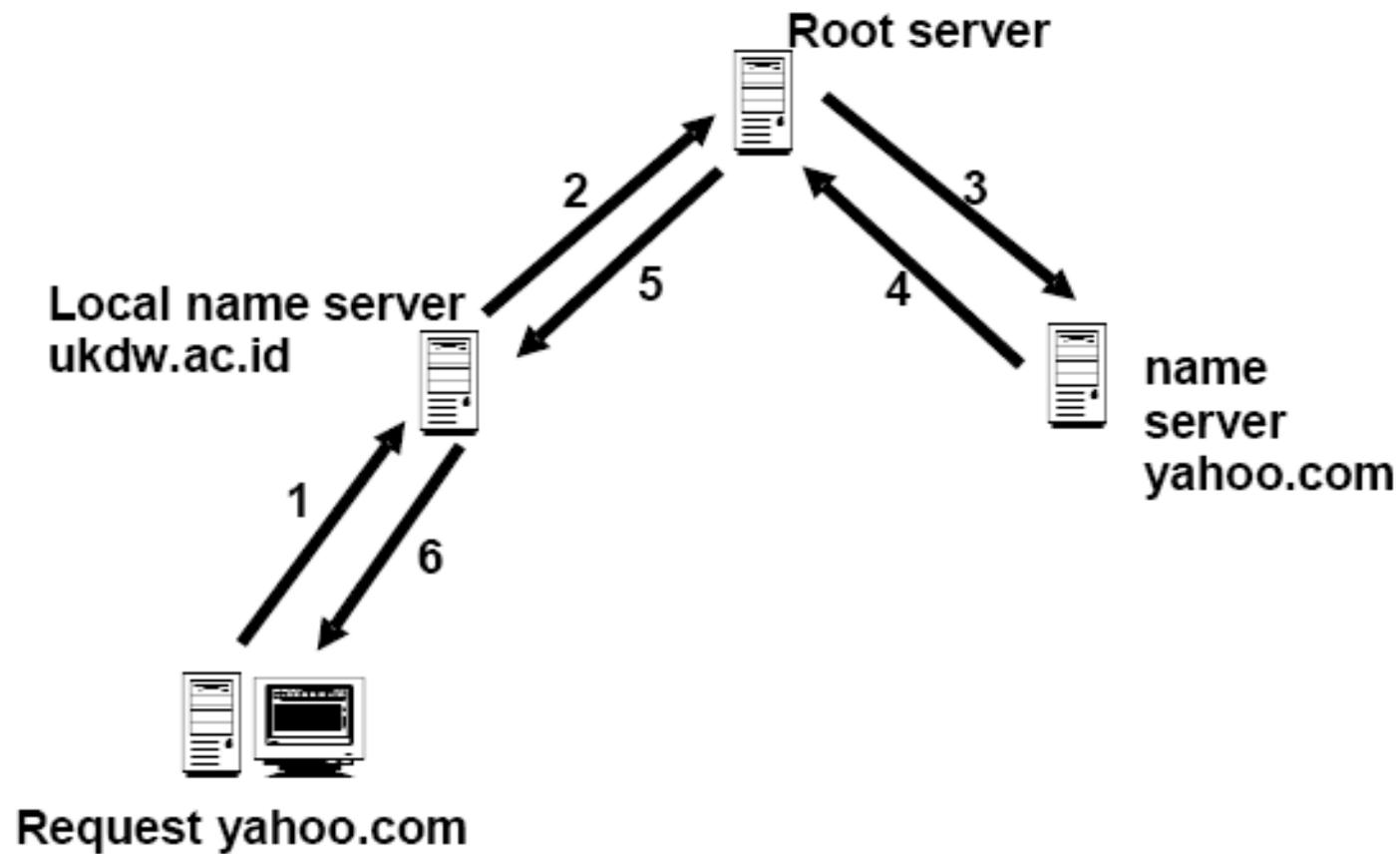
- Merupakan standard penamaan pada Internet
- Database DNS diterapkan dengan sistem partitioning yang terbagi-bagi dalam suatu zone berdasarkan domainnya.
- Pada tiap zone memiliki 2 name server yang menyediakan authoritative naming information (replikasi)
- Suatu authoritative DNS Server ada 2 jenis
 - Primary server : membaca data langsung dari database nama utama untuk zone tersebut
 - Secondary Server : secara periodik mendownload data zone dari database utama

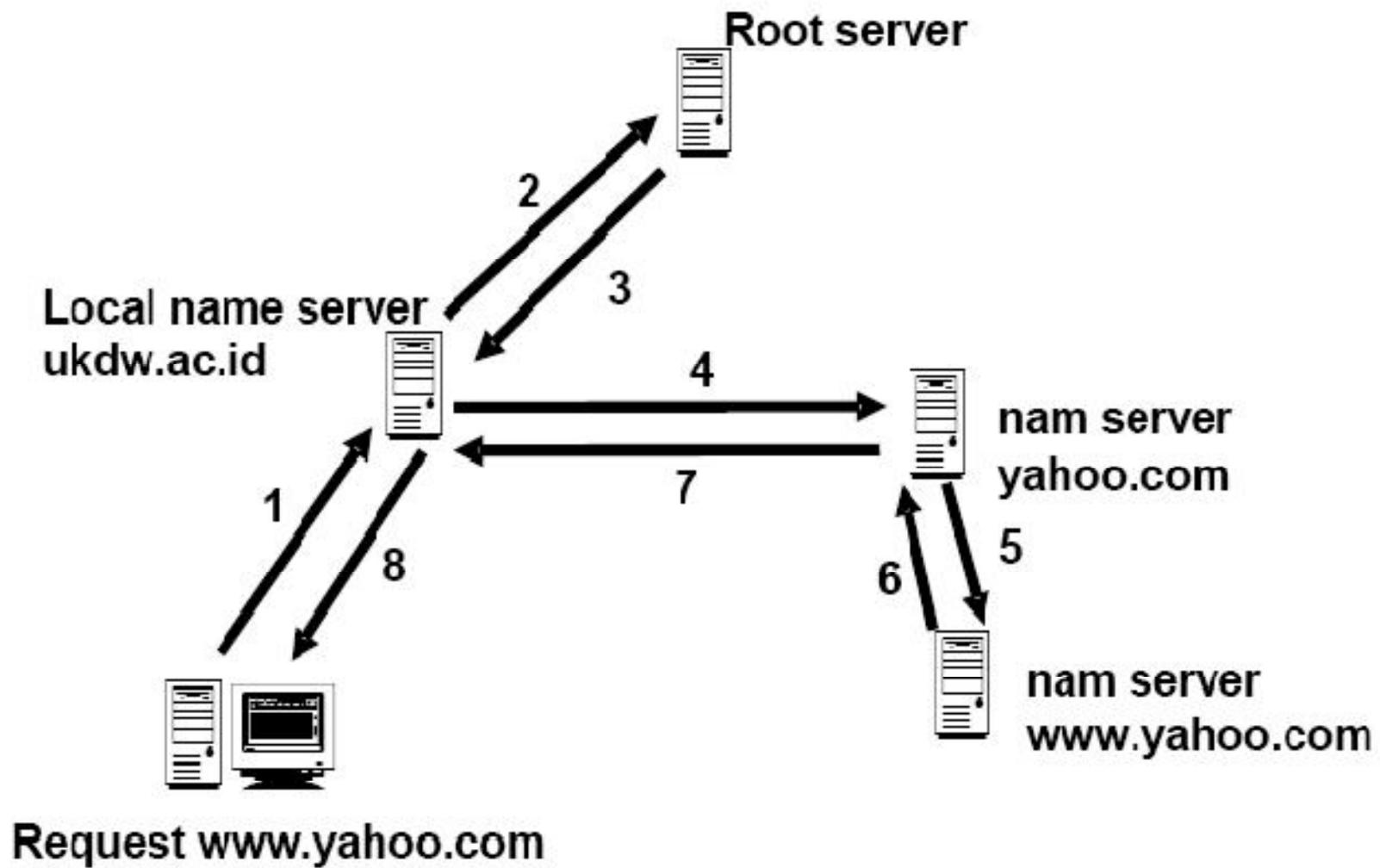
- Selain itu DNS juga menerapkan mekanisme caching yang selalu menyimpan informasi hasil resolve name yang sudah dilakukan
- Contoh definisi zone dengan BIND pada file /etc/named.conf

```
zone "budsus.com" {  
    type master;  
    file "budsus.db";  
};
```

- Biasanya pada BIND, database name zone disimpan pada direktori /var/named
- Untuk mendefinisikan authoritative DNS server dapat diberikan data berikut pada file domain.db (contoh : budsus.com.db)

Berikut cara kerja DNS secara iterative,
non-recursive atau recursive

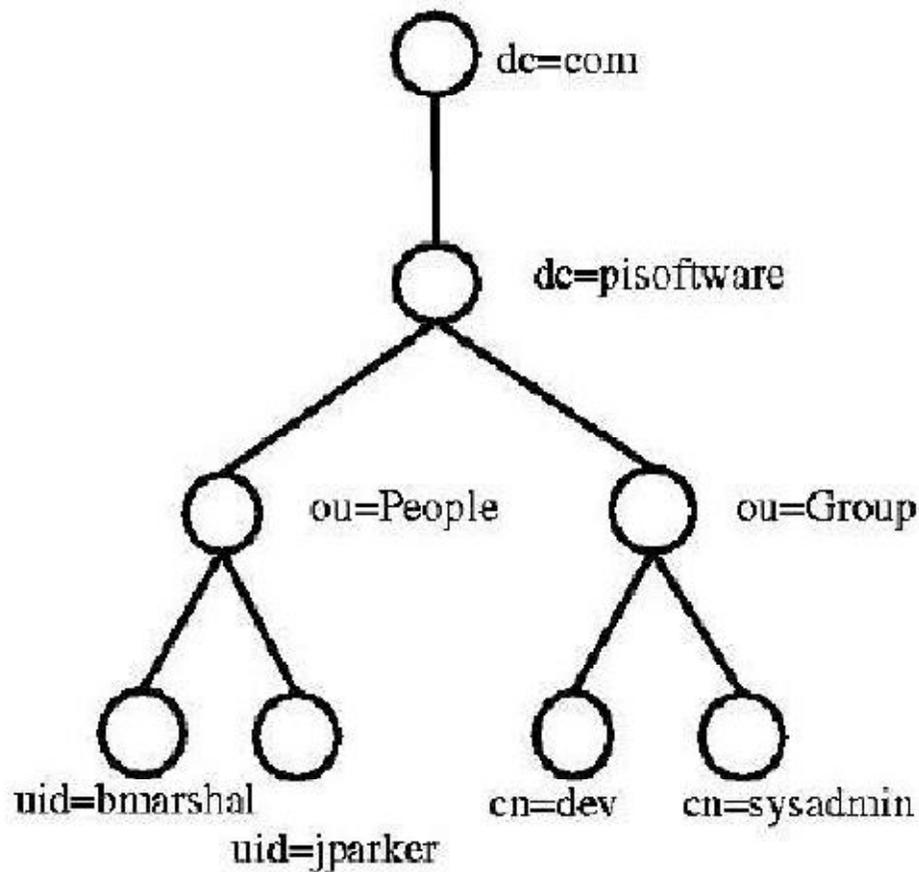




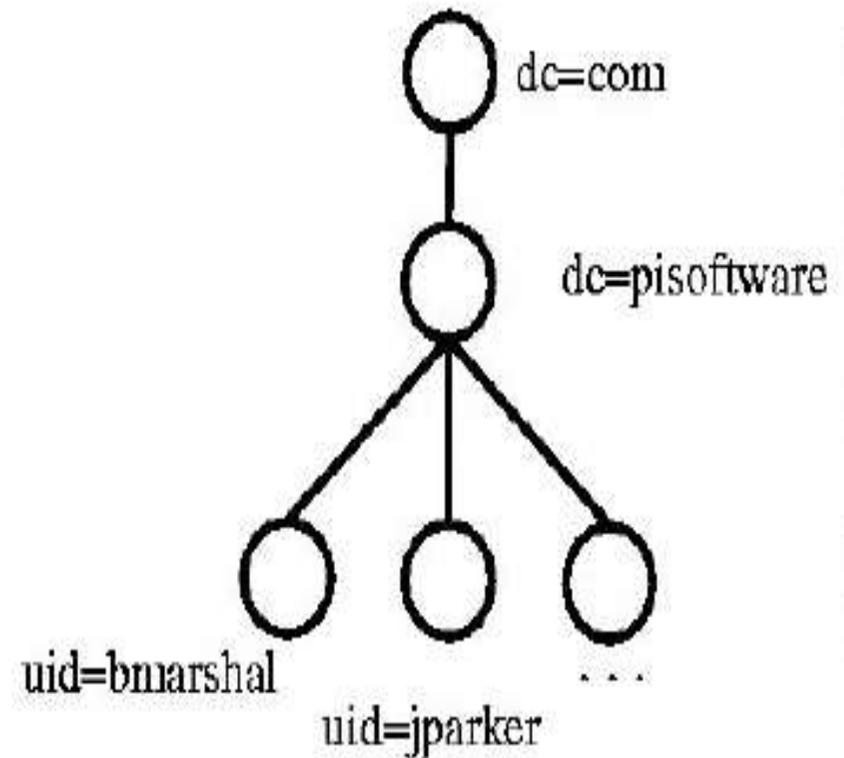
Directory Services

- Sebuah direktori merupakan sebuah database, yang pengelolaan informasinya di dasarkan pada atribut setiap item datanya.
- Informasi pada direktori lebih banyak akan dibaca daripada operasi update (add, modify, ataupun delete). Sehingga pada suatu directory service tidak menerapkan transaksi atau skema roll-back yang kompleks seperti halnya yang kita temui pada database service.
- Pengubahan informasi pada direktori terjadi pada semua atau beberapa atribut suatu item direktori. Dengan sifatnya yang sebagian besar diterapkan operasi pembacaan, maka suatu directory service akan menerapkan suatu model replikasi untuk dapat memberikan kehandalan yang lebih baik.

Suatu directory service akan menerapkan protokol LDAP (Lightweight Directory Access Protocol) dengan format atribut untuk tiap itemnya di dasarkan pada format standard X.500. LDAP disepakati dengan RFC 1777



Flat



- Secara prinsip struktur database pada directory service adalah hirarki seperti yang ditunjukkan pada gambar di atas.
- Seperti pada struktur hirarki pada umumnya, pada suatu database directory service akan memiliki suatu item yang dijadikan sebagai root.
- Untuk sebuah titik root, secara umum ditunjukkan dengan suatu atribut dc (Domain Component), namun dapat juga langsung ditunjuk dengan atribut cn (Common Name) atau ou(Organization Unit) ataupun o (Organization).
- Kemudian pada titik daun (leaf) biasanya akan berisi suatu item dengan atribut uid (User ID) ataupun cn. Berikut beberapa atribut untuk sebuah titik pada directory service (lihat RFC 2256) :

- uid → User id
- cn → Common Name
- sn → Surname
- L → Location
- ou → Organisational Unit
- o → Organisation
- dc → Domain Component
- st → State
- c → Country

SEKIAN