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

Pendahuluan

- Dalam sistem terdistribusi, nama digunakan untuk menunjuk ke suatu sesumber yang beragam dan tersebar seperti komputer, layanan (services), file, *remote object*, use.
- Bayangkan : bagaimana jika file dalam sistem berkas Anda tidak diberi nama?
- Nama memfasilitasi
  - ***komunikasi*** : nama domain sebagai bagian dari email
  - ***resource sharing*** : nama domain internet. Proses tidak dapat mengakses suatu sesumber, jika sesumber tersebut tidak diberi nama
- Seberapa banyak informasi tentang suatu objek dalam suatu 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, X.500 Directory Service, IOR (Interoperability Object Reference).

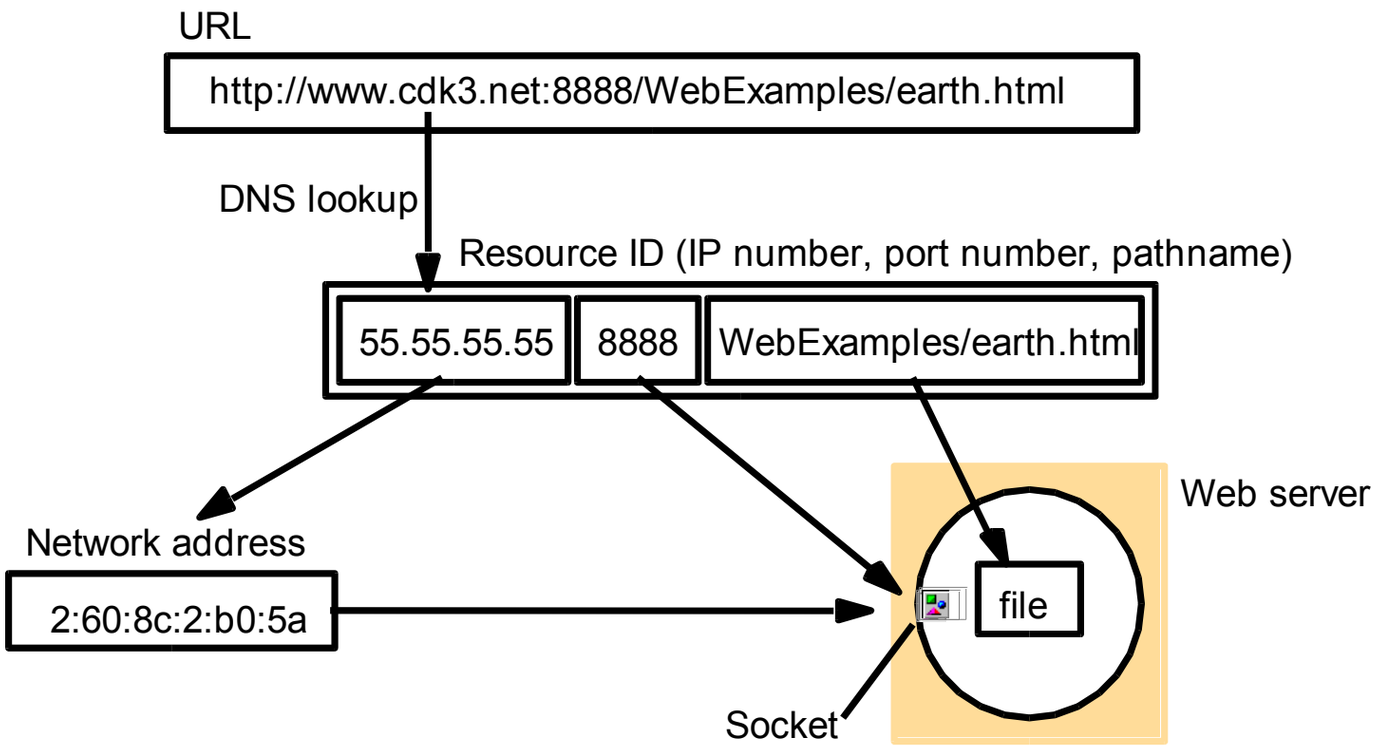
$ tnameserv &

Initial Naming Context:

IOR:000000000000002b49444c3a6f6d672e6f72672f436f734e616d696e672f4e616d696e67436f
6e746578744578743a312e3000000000000010000000000000007c000102000000000a3132372e302e
302e3100038400000035afabcb00000000205ff5185200000001000000000000000010000000d544e
616d6553657276696365000000000000000400000000a000000000000010000000100000020000
0000000100010000000205010001000100200001010900000000100010100

TransientNameServer: setting port for initial object references to: 900

- Kebutuhan akan penamaan tersebut memicu munculnya layanan penamaan (*Naming Services*) yang menyediakan mekanisme dan struktur penamaan objek itu sendiri. Contoh : DNS (Domain Name Service). Dan juga kebutuhan untuk pencarian objek berdasar nama dan juga atribut objek itu sendiri (*Directory Service*)
- Suatu nama akan diterjemahkan ke dalam suatu data tentang sesumber atau objek yang dimaksudkan. Gabungan antara objek dan nama disebut *binding*.
- Dalam nama objek terdapat beberapa ***atribut*** yang merupakan properti suatu objek. Contoh :
  - DNS : memetakan dari nama ke atribut alamat IP host
  - X.500 : memetakan suatu nama seseorang ke beberapa atribut, seperti email, telepon, dsb.
  - CORBA Naming Service yang memetakan nama *remote objek* ke *remote object reference*.



- **URL (Uniform Resource Locator)** merupakan suatu tipe khusus **URI (Uniform Resource Identifier)**. Tipe lain lainnya adalah **URN (Uniform Resource Name)**. Ide dengan adanya URN adalah user dapat melakukan query berdasar URN untuk mendapatkan URL objek. **URC (Uniform Resource Characteristics)** merupakan subset dari URN untuk

mendeskripsikan suatu sesumber Web dengan suatu atribut, contoh 'author=budsus', 'keywords=sister,...'

- Untuk informasi selengkapnya, silahkan kunjungi http://ftp.ics.uci.edu/pub/ietf/uri/

- Contoh bentuk **URL**
  - Pada RFC 1738, penamaan dengan URL dapat mendukung beberapa protokol berikut :

ftp	File Transfer Protocol
http	Hypertext Transfer Protocol
gopher	The Gopher Protocol
mailto	Email address
news	USENET news
nntp	USENET news using NNTP access
telnet	Reference to interactive sessions
wais	Wide-Area Information Servers
file	Host-specific file name
prospero	Prospero Directory Service

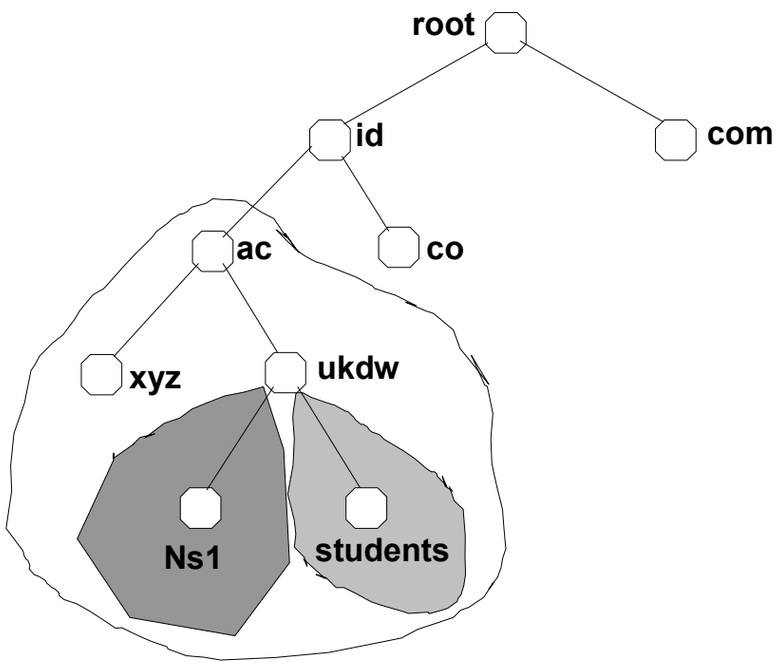
  - "http://" host [":" port] ["/" path] ["?" search]
  - "ftp://" [user ":" password "@" host] [":" port] *["/" directoryname] ["/" filename]
- Contoh Bentuk **URN**
  - *urn:nameSpace:nameSpace-specificName*.
  - Contoh : urn:ISBN:0-201-62433-8
  - Contoh : urn:dcs:gormenghast.ac.uk:TR2000-56

Name Service

- Suatu name service menyimpan kumpulan satu atau lebih *kontek penamaan* – yaitu sehimpunan 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
- perlindungan kegagalan

- Untuk memenuhi kebutuhan tersebut, sebuah name server setidaknya dapat menerapkan mekanisme berikut :
  - **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
    - hal ini dilakukan untuk mencegah operasi permintaan sama berulang-ulang



Name Space

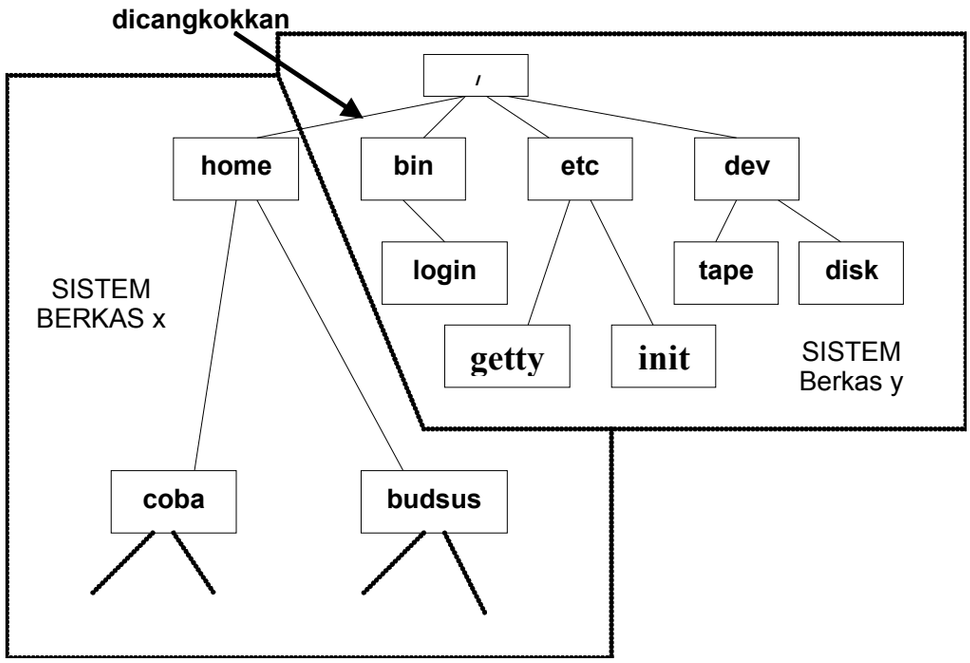
- ***Name Space*** adalah kumpulan nama sah yang dikenal oleh suatu layanan yang sesuai.

- Suatu nama dapat digambarkan dengan menggunakan generative grammar, seperti BNF (Backus-Naur Form). Contoh BNF untuk email

```
address    =  mailbox /  group              ; one addressee or named list
group      =  phrase ":" [#mailbox] ";"
mailbox    =  addr-spec / phrase route-addr ; simple addr or name&adddr-spec
route-addr =  "<" [route] addr-spec ">"
route      =  ("@" domain) *(", @" domain) ":" ; path-relative
addr-spec  =  local-part "@" domain         ; global address
local-part =  word *("." word)              ; uninterpreted case-preserved
domain     =  sub-domain *("." sub-domain)
sub-domain =  domain-ref / domain-literal
domain-ref =  atom                          ; symbolic reference
phrase     =  1*word                        ; sequence of words
atom       =  1*<any CHAR except specials, SPACE and CTLs>
word       =  atom / quoted-string
```

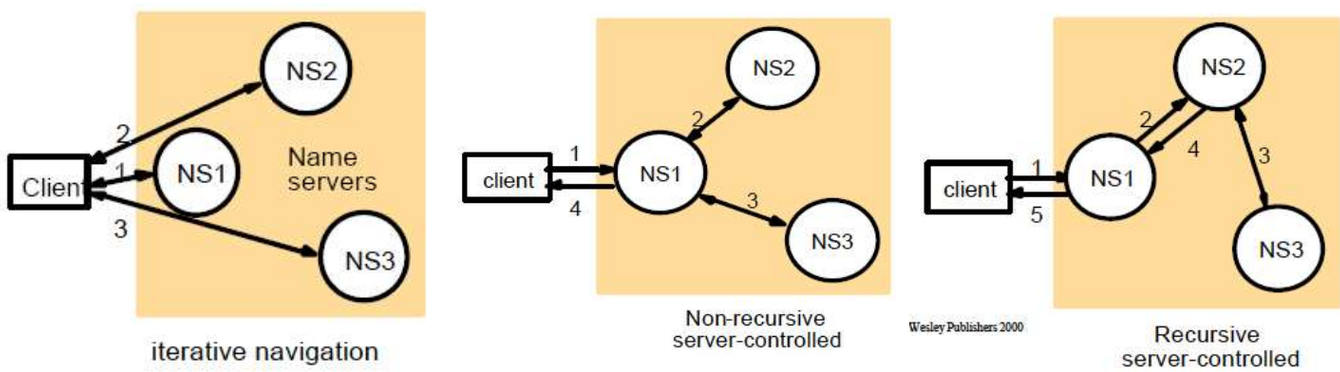
Berikut beberapa karakter name space

- 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.budsus.com
- suatu name space dapat diatur secara terdistribusi (*naming domain*)
- suatu name space dapat berupa gabungan dari beberapa name space lain. Contoh mount pada UNIX/Linux



Navigasi Name Server

- Navigasi di sini memiliki arti petunjuk pengaksesan data nama dari lebih dari satu name server untuk menyelesaikan pemetaan nama (*resolve a name*)
  - **Iterative Navigation**
  - **non-recursive, server-controlled**
  - **recursive, server-controlled**



Domain Name Services

- merupakan standard penamaan pada Internet
- database DNS diterapkan dengan sistem partitioning yang terbagi-bagi dalam suatu zone berdasar 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 disimpa pada direktori /var/named
- Untuk mendefinisikan authoritative DNS server dapat diberikan data berikut pada file domain.db (contoh : budsus.com.db)

```
@ IN SOA  ns1.budsus.com.  root.ns1.budsus.com. (
            20020401000   ;serial number
            7200          ;refresh - 2 jam
            3600          ;Retry - 1 jam
            43200         ;Expire - 12 jam
            3600 )        ;Minimum - 1 hour
```

- Beberapa tipe record pada DNS yang dikenal antara lain :

Tipe record NS (Name Server)

```
IN NS ns1.budsus.com
```

Tipe record MX (Mail eXchange)

```
IN MX 10 ns1.budsus.com
```

Tipe record A (Address)

```
ns1    IN A  192.168.0.2
```

Tipe record PTR (Domain Name Pointer)

```
192.168.0.2  IN PTRwww.budsus.com
```

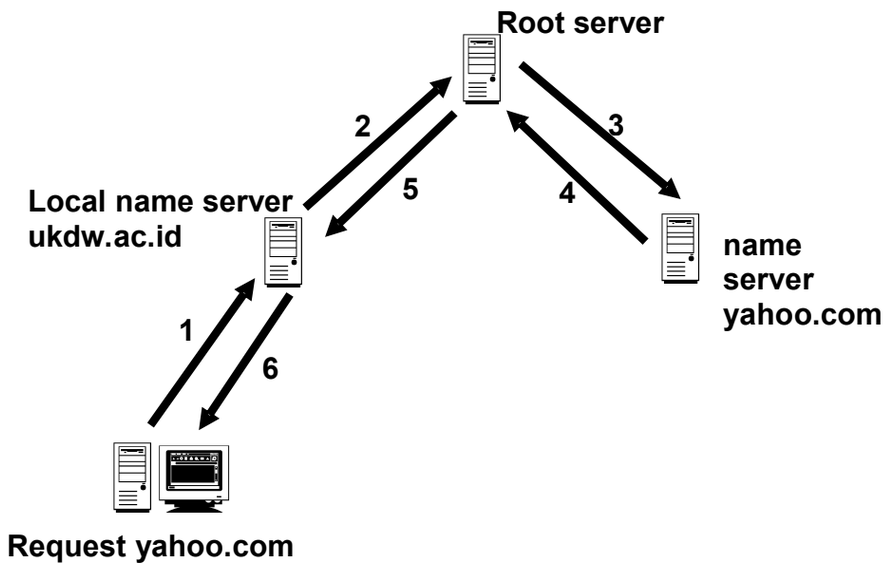
CNAME (Canonical Name)

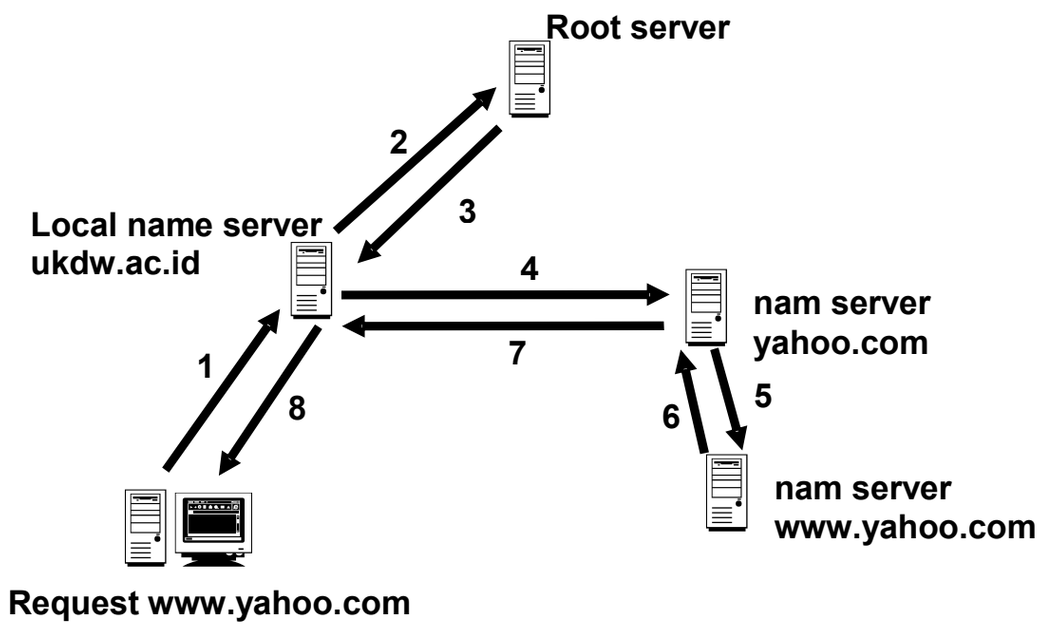
```
www    IN CNAME  ns1.budsus.com
```

- Agar mesin Anda dapat melakukan resolve ke DNS Server, pada file /etc/resolv.conf dapat diisi dengan informasi

```
domain wayga.net
nameserver 192.33.4.10
```

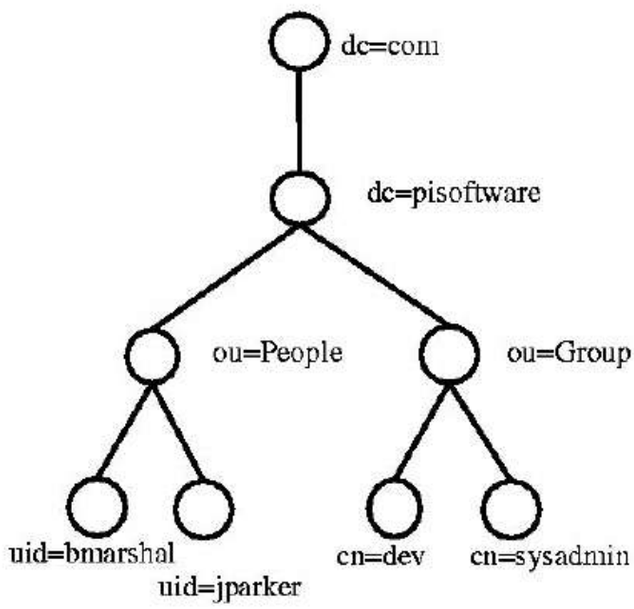
Berikut cara kerja DNS secara iterative, non-recursive atau recursive

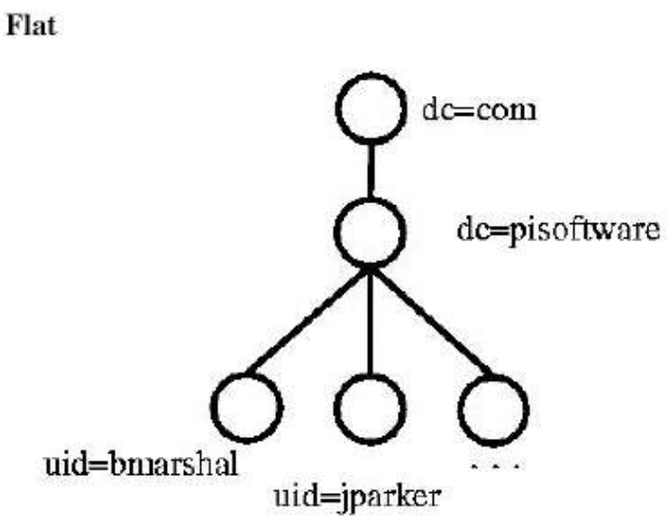




Directory Services

- 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 komplek 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.





- Secara prinsip struktur database pada directory service adalah hierarchy 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 denga 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

- Lalu sekarang pertanyaannya adalah bagaimana caranya kita menunjuk ke suatu item dari database directory service? Cara penunjukkan suatu item titik pada directory service sama seperti kita memperlakukan struktur hirarki DNS. Sebuah item titik pada directory service diberi suatu alamat baik secara relatif maupun absolut.
- Untuk suatu alamat relatif sering disebut sebagai RDN (Relative Distinguish Name), sedangkan alamat yang absolut disebut sebagai DN (Distinguish Name). Pengalamatan ini disepakati dengan RFC 1779. Contoh pada gambar di atas DN untuk uid=jparker adalah “dn=uid=jparker, ou=People, dc=pisoftware, dc=com”.
- Secara keseluruhan, sebuah item dapat diakses dari directory client dengan mematuhi aturan URI (Uniform Resource Identifier) seperti yang tertulis pada RFC 1959. Berikut beberapa sintak URI:

<ldapurl> ::= "ldap://" [<hostport>] "/" <dn> ["?" <attributes>
["?" <scope> "?" <filter>]]

<hostport> ::= <hostname> [":" <portnumber>]

<dn> ::= a string as defined in RFC 1485

<attributes> ::= NULL | <attributelist>

<attributelist> ::= <attributetype>
| <attributetype> ["," <attributelist>]

<attributetype> ::= a string as defined in RFC 1777

<scope> ::= "base" | "one" | "sub"

<filter> ::= a string as defined in RFC 1558

- Scope :
  - **base** : pencarian dilakukan dimulai dari titik yang ditunjuk
  - **one** : pencarian dilakukan hanya pada satu level saja
  - **sub** : pencarian dilakukan hanya pada subtree dari titik yang ditunjuk
- Sedangkan operator yang dapat Anda berikan pada pola pencarian (filtering) adalah sebagai berikut :
  - & = and
  - | = or
  - ! = not
  - ~= = serupa
  - >= = lebih besar atau sama dengan
  - <= = lebih kecil atau sama dengan
  - * = sembarang

- Contoh :
  - `(objectclass=posixAccount)`
  - `(cn=Mickey M*)`
  - `(|(uid=fred)(uid=bill))`
  - `(&(|(uid=jack)(uid=jill))(objectclass=posixAccount))`

- Contoh URI LDAP :
  - ldap://foo.bar.com/dc=bar,dc=com
  - ldap://argle.bargle.com/dc=bar,dc=com??sub?uid=barney

- Contoh pengaksesan directory services (OpenLDAP)

```
$ ldapsearch -vLx -h 127.0.0.1  -b "ou=linux,o=budsus.id"
"(mail=*)" cn
ldap_init( 127.0.0.1, 0 )
filter: (mail=*)
requesting: cn
version: 1

#
# filter: (mail=*)
# requesting: cn
#

# Budi Susanto, linux, budsus.id
dn: cn=Budi Susanto, ou=linux, o=budsus.id
cn: Budi Susanto

# Umi Proboyekti, linux, budsus.id
dn: cn=Umi Proboyekti, ou=linux, o=budsus.id
cn: Umi Proboyekdi

# search result

# numResponses: 3
# numEntries: 2
```

- Contoh program Perl untuk pengaksesan directory services

```
#!/usr/bin/perl -w
use strict;
use Net::LDAP;
my($ldap) = Net::LDAP->new('localhost') or
       die "Tidak dapat terkoneksi ke ldap: $!\n";
$ldap->bind;
my($mesg) = $ldap->search(
                base => "ou=linux,o=budsus.id",
                filter => '(objectclass=*)');

$mesg->code && die $mesg->error;
my($entry);

foreach $entry ($mesg->all_entries) {
  $entry->dump;
}
$ldap->unbind;
```