

## BAB II

### LANDASAN TEORI

#### 2.1. Dasar Teori

##### 2.1.1. Pengertian *Internet*, *Intranet* dan *Extranet*

*Internet* merupakan singkatan dari *Interconnected Network*. *Internet* adalah kumpulan komputer yang terhubung satu dengan yang lain dalam sebuah jaringan. Disebut jaringan yang saling terhubung karena *internet* menghubungkan komputer satu dengan komputer lain yang ada di seluruh dunia. *Internet* dapat digunakan sebagai sarana pertukaran informasi maupun berkomunikasi antar komputer yang saling terhubung. Selain kita dapat memperoleh banyak data yang diperlukan, lewat *internet* proses untuk mendapatkan data atau informasi menjadi lebih cepat dan efisien.

*Intranet* adalah suatu jaringan komputer pribadi yang menggunakan *internet* protokol dan konektivitas jaringan yang memungkinkan untuk berbagi perangkat dalam sebuah organisasi. *Intranet* berbeda dengan *website* pada umumnya, *website* pada umumnya berorientasi untuk umum sehingga tidak membutuhkan autentifikasi untuk menampilkan isi yang ada di dalamnya. Sedangkan *intranetsite* berisi tentang sesuatu yang lebih bersifat rahasia dan memerlukan autentifikasi ketika ingin menampilkan isinya.

*Extranet* adalah suatu jaringan komputer pribadi yang menggunakan *internet* protokol, konektivitas jaringan dan memungkinkan komputer untuk dapat diakses oleh luar organisasi seperti pelanggan, rekan bisnis,

*vendor* maupun *supplier*. *Extranet* dapat juga diartikan sebagai *intranet* perusahaan yang diperluas fungsinya untuk kebutuhan pengguna di luar perusahaan.

### **2.1.2. Pengertian Jaringan Komputer**

Perkembangan teknologi jaringan komputer semakin berkembang dengan pesatnya. Proses perkembangan ini didukung pula oleh perkembangan teknologi komunikasi yang tidak kalah pesatnya. Hal ini disebabkan oleh manusia yang semakin dinamis sehingga ia tidak mau terkungkung dalam waktu dan ruang. Manusia menginginkan segala yang akan dikerjakan dapat dikerjakan kapan dan dimanapun juga. Selain itu manusia juga merasa tidak puas jika hanya bekerja sendirian karena tidak efisien, maka ia juga ingin agar komputer yang digunakan dapat saling berkomunikasi dan berbagi informasi ataupun data.

Oleh karena itu, agar lebih efisien maka komputer-komputer tersebut kemudian dihubungkan satu dengan yang lain. Sistem inilah yang kini dinamakan sebagai jaringan komputer. Jaringan komputer dapat didefinisikan sebagai sekelompok komputer otonom yang dihubungkan satu dengan yang lainnya menggunakan protokol komunikasi melalui media transmisi atau media komunikasi sehingga dapat saling berbagi data informasi, program-program, penggunaan bersama perangkat keras seperti *printer*, *harddisk* dan memberikan layanan komunikasi antar pemakai (Andrew S. Tanenbaum, 1997).

### **2.1.3. Pemrograman Web**

Pemrograman *web* dibagi menjadi 2 yaitu :

1. Pemrograman *Server Side*.
2. Pemrograman *Client Side*.

#### **2.1.3.1. Pemrograman *Server Side***

Pada pemrograman *Server Side*, perintah-perintah program dijalankan di *server web*. Biasanya merupakan komputer *high-end* yang memiliki kemampuan tinggi, dimana di dalamnya terinstall program *server* yang siap melayani permintaan layanan dari *client*. Pada pemrograman ini, *server* tidak pernah mengirimkan permintaan. Proses yang terjadi adalah menunggu permintaan dan menanggapi permintaan serta mengirimkan data kepada *client*. Contoh pemrograman *server* antara lain *web server*, *mail server* dan *database server*.

#### **2.1.3.2. Pemrograman *Client Side***

Pada pemrograman *Client Side*, perintah program dijalankan pada *browser web* sehingga ketika *client* meminta dokumen *script* maka *script* dapat *download* dari *server* kemudian dijalankan pada *browser* yang bersangkutan. Pada pemrograman ini, *client* mengirimkan permintaan kepada *server*. Setelah mengirimkan permintaan, *client* akan menunggu balasan dari *server*. Contoh pemrograman *client* antara lain *web browsers* dan *email client*.

### **2.1.4. Skala Jaringan Komputer**

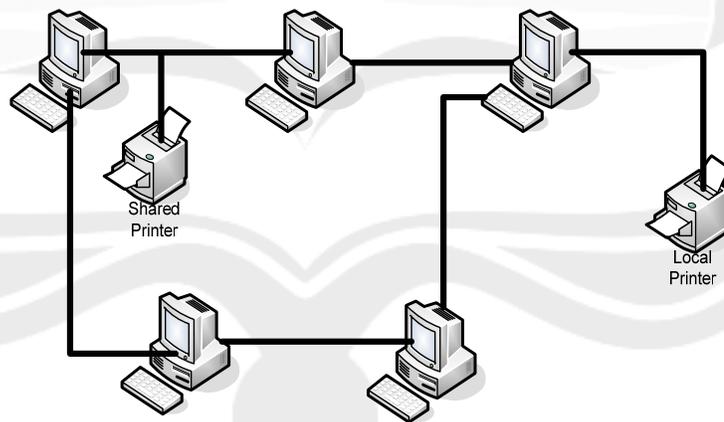
Jaringan komputer dibedakan menjadi 3 jenis yaitu :

## 1. Local Area Network (LAN)

Jaringan yang menghubungkan komputer-komputer dalam satu ruangan atau suatu gedung. LAN sendiri dapat dibagi menjadi 2 jenis yaitu *Peer-To-Peer* dan *Client-Server*.

### Peer-To-Peer

Adalah jaringan yang semua komputer atau mesin yang terhubung mempunyai kedudukan dan fungsi yang sama (tidak ada komputer yang menjadi pengendali utama). Fasilitas yang tersedia untuk model ini adalah berbagi *file* dan berbagi *printer* sehingga setiap mesin yang terhubung dapat membuka file atau menggunakan sumber daya di mesin yang lain tergantung dari hak akses dari mesin tersebut.

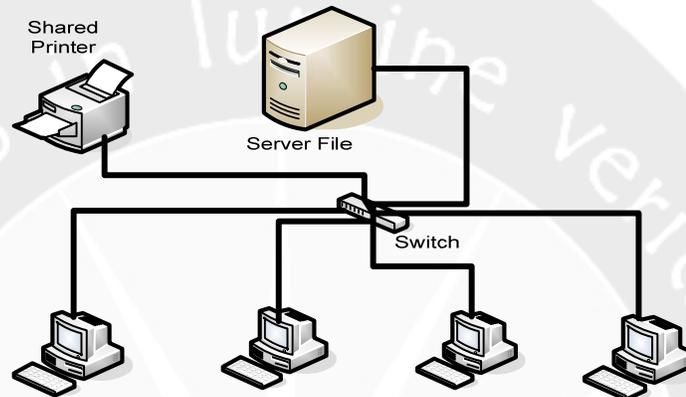


2.1 Gambar Jaringan *Peer-To-Peer*

### Client-Server

Adalah jaringan yang komputer-komputer di dalamnya bertindak sebagai *file server* dan mesin yang lain sebagai *client*. Komunikasi pada model *client-*

*server* pada umumnya berbentuk pesan permintaan untuk melaksanakan berbagai pekerjaan dari *client* kepada *server*, dan setelah *server* melaksanakan tugasnya, kemudian hasilnya akan dikirim kembali ke *client*.



2.2 Gambar Jaringan *Client-Server*

## 2. Metropolitan Area Network (MAN)

Jaringan yang menghubungkan komputer-komputer dalam satu kota, sampai dengan beberapa puluh kilometer. Cakupan daerahnya lebih luas dibandingkan dengan LAN.

## 3. Wide Area Network (WAN)

Jaringan yang menghubungkan komputer-komputer dalam satu negara atau benua, sampai beberapa ratus kilometer.

### 2.1.4. Sistem kerja HTTP

HTTP yang merupakan kependekan dari *HyperText Transfer Protocol* adalah protokol yang dipergunakan untuk mentransfer dokumen dalam *World Wide Web (WWW)*. Protokol ini adalah protokol ringan, tidak berstatus

yang dapat dipergunakan untuk berbagai macam tipe dokumen. Sebuah *client* HTTP seperti *web browser*, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke *port* tertentu (biasanya *port* 80). Sebuah *server* HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tertentu. Beberapa kepala (*header*) juga bebas ditulis atau tidak, sementara lainnya (seperti *tuas rumah*) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), *server* mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan error atau pesan lainnya.

Komunikasi antara *web client* dengan *server* dilakukan menggunakan protokol HTTP (*Hypertext Transfer Protocol*). Dalam tulisan ini kita gunakan istilah "HTTP server" yang mengacu pada HTTP daemon itu sendiri, dan bukannya "web server" yang mengacu pada *server* secara keseluruhan di mana termasuk juga operasi berikut *back-end database* jika digunakan. HTTP biasanya menggunakan TCP sebagai *transport* protokolnya, dan termasuk protokol tingkat tinggi seperti halnya SMTP, POP3, FTP, dan NNTP. Saat ini *server-server* HTTP yang utama (seperti Apache) telah mendukung HTTP versi 1.1.

HTTP disebut sebagai *stateless protocol* karena setiap perintah di dalamnya akan di eksekusi satu persatu, jadi koneksi antara *browser* dan *server* akan

berakhir begitu transaksi sudah dijalankan. Sebuah transaksi HTTP dimulai dengan *client* mengirimkan *request* ke *server* HTTP, yang kemudian menjawab dengan mengirimkan respons balik ke *client*. Transaksi berikutnya *independent* dari transaksi sebelumnya, proses inilah yang dinamakan sebagai *stateless protocol*. Kita dapat menggunakan *telnet* untuk membuat koneksi ke *server* HTTP serta mengirim *request* dan menerima *respons* HTTP sebagaimana yang dilakukan oleh *web browser*.

#### **2.1.5. Pengertian PHP**

PHP adalah singkatan dari *Hypertext Preprocessor*. PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah *server*. Dengan menggunakan PHP, sebuah *web sites* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *web sites* akan diolah dan disimpan pada *database web server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP, file harus di *upload* ke dalam *server*. *Upload* adalah proses mentransfer data atau file dari komputer *client* ke *web server*. *Download* adalah proses mentransfer atau memindah data dari *web server* ke *client*. PHP termasuk *server-side scripting* artinya semua perintah PHP dieksekusi oleh *server* bukan oleh *browser*. Tujuan dari PHP adalah untuk membuat aplikasi yang dijalankan pada teknologi *web*. Jadi seluruh aplikasi yang berbasis *web* dapat dibuat dengan menggunakan PHP.

Keunggulan utama menggunakan PHP adalah *script* yang sederhana untuk pemula, tetapi juga menyediakan fitur-fitur tambahan untuk *programmer professional*. Selain itu PHP juga mempunyai kecepatan akses yang tinggi, dapat berjalan dalam *server web* yang berbeda dan dalam sistem operasi yang berbeda, termasuk dalam bahasa yang *embedded* maupun diletakkan dalam *tag HTML* dan bersifat *open source*.

## **2.2. Konsep Dasar**

Untuk mempelajari LDAP, sangatlah penting untuk kita memahami arti dari direktori dan untuk apa suatu direktori digunakan. Mungkin tanpa sadar kita sudah terbiasa dengan direktori. Direktori dapat berupa *personal address book*, *yellow pages* bahkan *web direktori* seperti Yahoo. Sebagai contoh *yellow pages*, kita dapat menemukan alamat lengkap, nomor telepon, alamat *website* dan *e-mail* dari suatu perusahaan hanya dengan mencarinya berdasarkan nama dari perusahaan yang sudah disusun secara alfabetis pada direktori *yellow pages*.

Dalam komputer, *directory service* bisa dikatakan sebagai suatu *database* tempat penyimpanan data yang dapat digunakan untuk memberikan informasi-informasi yang berkaitan dengan objeknya. Bagian direktori mungkin dapat berisi kumpulan informasi tentang user seperti *user name*, *first name*, *phone number* dan lain sebagainya. *Directory service* biasanya menyimpan informasi dalam bentuk struktur *tree* yang dinamakan *Directory Information Tree (DIT)*.

### 2.2.1. Pengertian LDAP

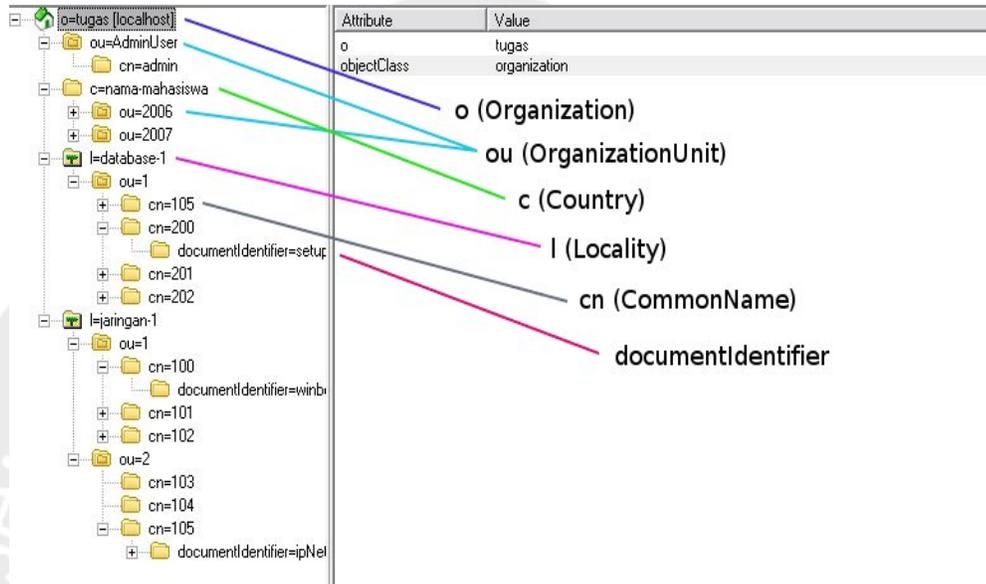
LDAP merupakan singkatan dari *Light Weight Directory Access Protocol*. LDAP adalah sebuah protokol yang mengatur mekanisme pengaksesan layanan direktori (*directory service*) yang dapat digunakan untuk mendeskripsikan banyak informasi seperti informasi tentang *people, organizations, roles, services* dan banyak entitas lainnya. LDAP menggunakan *model client-server* dimana *client* mengirimkan *identifier* data kepada *server* menggunakan protokol TCP/IP dan *server* mencoba mencarinya pada DIT yang tersimpan di *server*. Bila ditemukan maka hasilnya akan dikirimkan ke *client* tersebut namun bila tidak maka hasilnya berupa *pointer* ke *server* lain yang menyimpan data yang dicari. Dan untuk memfasilitasi pengisian dan perubahan informasi data dalam suatu direktori maka digunakanlah LDIF (*LDAP Data Interchange Format*).

Secara umum LDAP dapat dibagi menjadi 2 mode operasi yaitu *mode stand alone* dan *mode gateway*. Pada *mode stand alone*, LDAP berfungsi menyediakan pelayanan direktori sekaligus sebagai penyimpan data. Sedangkan pada *mode gateway*, LDAP berfungsi sebagai perantara dari data direktori / informasi yang disimpan dengan protokol lain. Pada tugas ini, LDAP yang digunakan adalah mode LDAP *stand alone*.

Model operasi LDAP secara garis besar dapat dibagi menjadi 4 mode pendekatan yaitu model informasi, model penamaan, model fungsi dan model keamanan. Model informasi didefinisikan sebagai unit dasar dari LDAP yang digunakan untuk menyimpan informasi. Setiap unit data yang disebut *entry data*,

memiliki nama yang unik dan juga memiliki OID (*unique object identifier*) yang berdasarkan standar operasi LDAP. Model penamaan didefinisikan sebagai struktur direktori, dalam implementasinya disebut sebagai DN (*Distinguished Name*). Struktur DN didefinisikan dengan standar tertentu yang telah ditentukan. Struktur DN dapat berfungsi antara lain sebagai struktur data bagaimana data di atur, di urutkan dan mencegah data yang sama, duplikat dan keambiguan data. Model fungsi didefinisikan sebagai fungsi LDAP itu sendiri dalam mengakses, memelihara dan mengatur data yang ada. LDAP memberikan beberapa kemungkinan untuk melakukan pencarian, perbandingan dan perubahan data. Model keamanan didefinisikan dengan proses autentifikasi dan otorisasi. Proses autentifikasi dilakukan sebelum pengguna menggunakan pelayanan di LDAP. Sedangkan proses otorisasi memberikan izin akses/tingkat penggunaan kepada pengguna yang telah melakukan autentifikasi.

Pada tugas ini digunakan *software OpenLDAP* yang berfungsi sebagai penyedia servis LDAP. *OpenLDAP* menyediakan skema standard yang secara *default terinstall*. Adapun *ObjectClass* yang digunakan adalah *Organization(o)*, *OrganizationUnit(ou)*, *Country(c)*, *Locality(l)*, *CommonName(cn)* dan *documentIdentifier*. Berikut dibawah ini gambar struktur penyimpanan data :



2.3 Gambar Struktur Penyimpanan Data

Terdapat dua servis utama pada LDAP yaitu yang pertama adalah *slapd* yang merupakan *LDAP daemon*, melayani *request* dari *client*, *query* dan berkomunikasi dengan *backend database* dan yang kedua adalah *sharpd* yang merupakan *replication daemon*, melayani replikasi data agar terus terjadi sinkronisasi data antara *client-server*.

### 2.2.2. Alasan penggunaan LDAP

Beberapa alasan kenapa akhirnya para pengguna lebih memilih menggunakan LDAP yaitu:

1. *Query* terhadap data atau informasi yang dibutuhkan relatif mudah.
2. Metode Pengaksesan LDAP tidak terbatas pada 1 bahasa pemrograman atau *compiler*.

3. LDAP mudah untuk di extends baik dalam tipe data maupun struktur data.
4. Sangat efisien dalam pengaksesan data atau informasi.
5. Subtree dapat didelegasikan ke *server* lain tanpa mengubah konfigurasi *client*.

