BAB 2
TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja

Balai Besar Bahan dan Barang Teknik (B4T) beralamat di Jalan Sangkuriang 14 Bandung 40135.Telp. 022 – 2504088, 2504828, 2510682. Fax 022 – 2502027. Website : www.b4t.go.id dan *e-mail* : info@b4t.go.id.

2.1.1 Sejarah Instansi

B4T didirikan pada tahun 1909 di Batavia (Jakarta sekarang) oleh pemerintah Hindia Belanda dengan nama Laboraturium Voor Metal Onderzoek dan berada dibawah naungan *Burgelizke Openbake Warken* (Departemen Pekerjaan Umum sekarang).

Pada tahun 1912 diperluas menjadi Laboraturium Voor Material Onderzoek.Yang semula balai ini hanya mengurus benda-benda metal diperluas dengan benda material lainnya.

Pada tahun 1921 lokasi balai dipindahkan ke Bandung.Tepatnya ke lokasi *Technische Hogeschool* (Institut Teknologi Bandung sekarang). Tahun 1934 kedudukan balai berpindah ke naungan *Van Ekonomische* (Departemen Perekonomian dan Perdangangan)

Pada tahun 1942 dibawah kekuasaan pemerintah Jepang balai berubah nama menjadi Laboraturium Zeiro Sikendya dan kemudian berubah lagi menjadi Laboraturim Kogio Sikendya.

Tahun 1945 Indonesia merdeka dan balai ini kembali berubah nama menjadi Balai Penyelidikan Bahan-bahan yang berkedudukan dibawah kementerian Kemakmuran. Pada periode tahun 1952 – 1960 balai ini dua kali beralih naungan, yaitu dari Kementerian Perekonomian ke Kementerian Perindustrian. Pada tahun 1961 balai ini menempati Jalan

Sangkuriang Bandung dengan nama Balai Penelitian Bahan-bahan. Pada tahun 1963 kedudukan balai berada di bawah Perindustrian Rakyat.

Tahun 1971 kedudukan balai berada di bawah naungan Lembaga Penelitian dan Pendidikan Industri.Pada tahun 1974 kedudukan balai berada di bawah Puslitbang Industri Logam dan Mesin Departemen Perindustrian.

Tahun 1980 kedudukan balai dibawah Badan Litbang Industri Departemen Perindustrian dan berubah nama menjadi Balai Besar Penelitian dan Pengembangan Industri Bahan dan Barang Teknik. Dan pada akhir tahun 2002 balai ini berubah nama menjadi Balai Besar Bahan dan Barang Teknik yang disingkat dengan B4T dan dipakai hingga sekarang.

2.1.2 Logo Instansi

Berikut gambar logo instansi B4T :



Gambar 2.1 Logo Instansi B4T

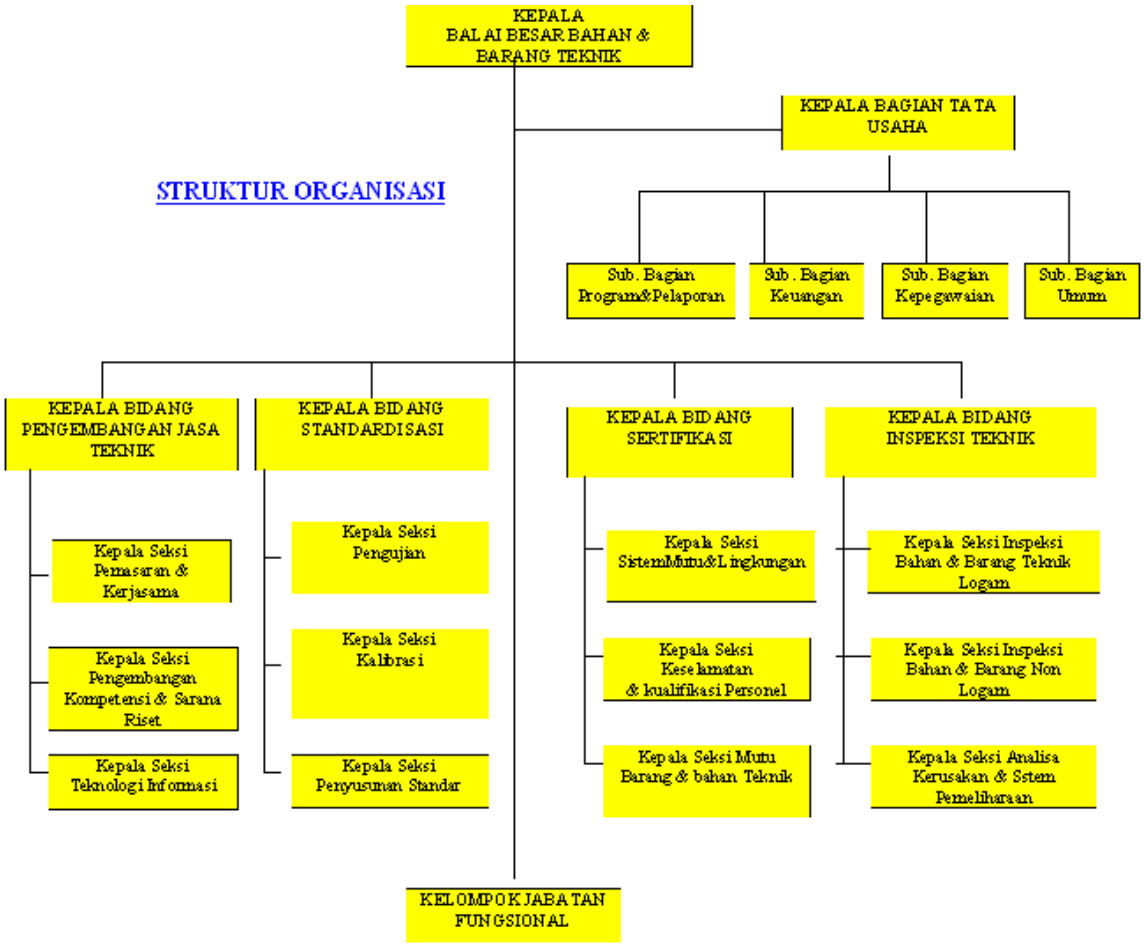
Huruf B dan T berwarna merah menandakan independen. Hurup B menandakan awal kata dari kata “Balai”, “Besar”, “Bahan” dan “Barang”. Sedangkan huruf T menandakan awal kata dari kata “Teknik”. Angka 4 yang berwarna silver menandakan bahwa kantor ini sebagian besar menangani barang barang teknik yang tidak jauh dari menangani barang berbahankan dasar logam dan menandakan dari jumlah awal kata yaitu huruf B dari kalimat “Balai Besar Bahan dan Barang Teknik”.

2.1.3 Badan Hukum Instansi

Balai Besar Bahan dan Barang Teknik merupakan balai resmi pemerintah yang berada di bawah naungan Badan Penelitian dan Pengembangan Industri, Departemen Perindustrian.Badan hukum oleh Menteri Perindustrian Republik Indonesia.Nomor : 13 / M – IMD / PER / 6 / 2006.

2.1.4 Struktur Organisasi dan Job Description

Berikut gambar Struktur Organisasi B4T:



Gambar 2.2 Struktur Organisasi B4T

Berikut gambar Struktur Organisasi Sistem pelelangan di B4T :



Gambar 2.3 Organisasi Sistem pelelangan di B4T

Berikut merupakan deskripsi pekerjaan dari struktur organisasi Pelelangan (*Job Description*) di B4T sesuai dengan jabatannya masing-masing.

1. Kepala B4T

Mengerjakan hal-hal sebagai berikut :

   a. Sebagai pimpinan di B4T.
   b. Mengawasi jalannya aktivitas di B4T
   c. Menetapkan tugas dari kepala bidang yang ada.
   d. Mengawasi dan mengkoordinir kegiatan-kegiatan dari setiap kepala bidang yang ada.

2. Kepala Bidang Pengembangan Jasa dan Teknik

   Mengerjakan hal- hal berikut

a. Sebagai pimpinan di divisi pengembangan jasa dan teknik.
b. Menetapkan kegiatan lelang.
c. Membuat surat laporan kegiatan lelang.
d. Menyetujui senua dokumen yang masuk ke divisi pengembangan jasa dan teknik.
e. Menentukan pemenang kegiatan lelang.

3. Kepala seksi Pengembangan Kompetensi dan Sarana Riset

   Mengerjakan hal-hal berikut

   a. Menyediakan barang-barang yang akan dilelang.
   b. Mengawasi kegiatan lelang
   c. Mengecek data perusahaan peserta lelang.
   d. Mengecek data perusahaan pemenang lelang

4. Kepala Seksi Informasi

   Mengerjakan hal-hal berikut

   a. Mengawasi jaringan *server* di B4T
   b. Mengawasi semua sistem yang berjalan di B4T
   c. Admin di sistem pelelangan.

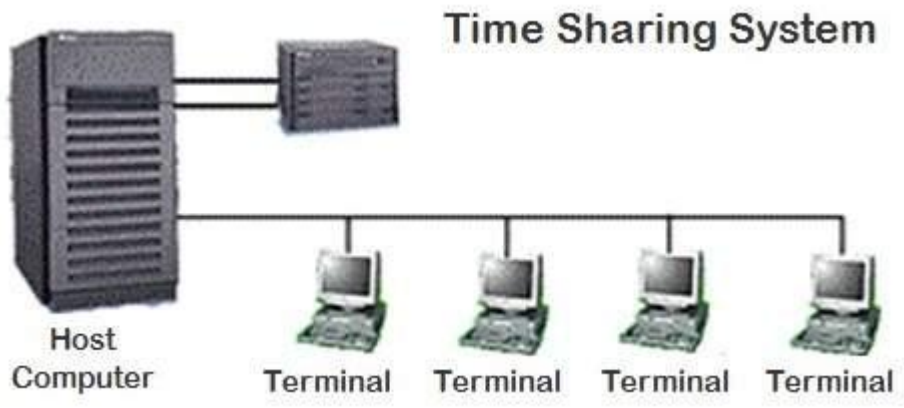
2.2 Landasan Teori

2.2.1 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada *printer* yang sama dan bersama-sama menggunakan *hardware*/*software* yang terhubung dengan jaringan. Setiap komputer, printer atau periferal yang terhubung dengan jaringan disebut *node*. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan *node*.

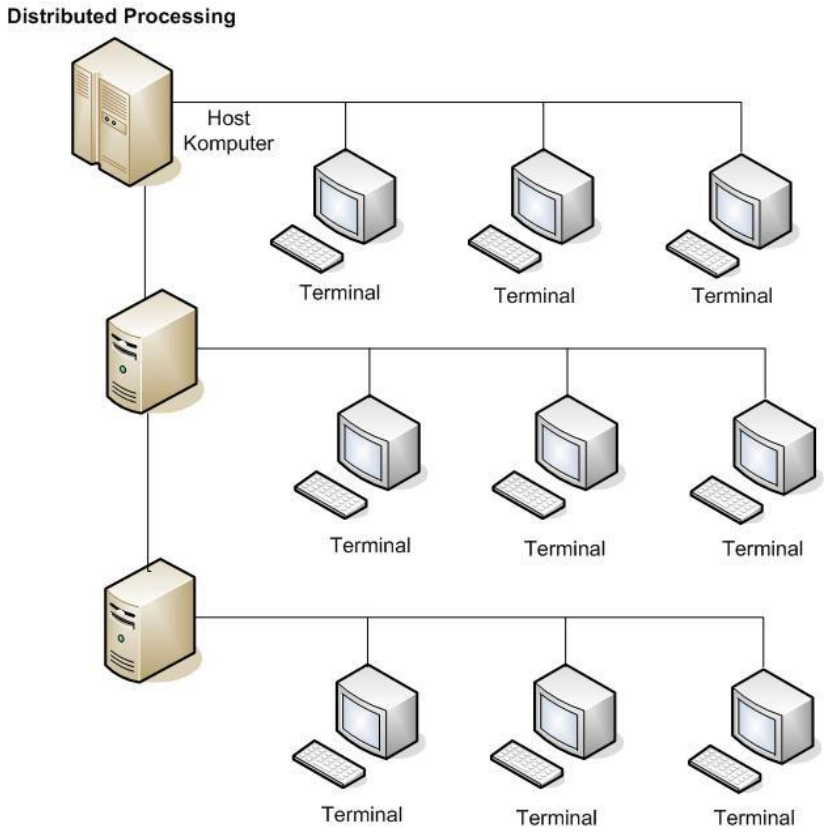
Konsep jaringan komputer lahir pada tahun 1940-an di Amerika dari sebuah proyek pengembangan komputer MODEL I di laboratorium Bell dan group riset Harvard

University yang dipimpin profesor H. Aiken. Pada mulanya proyek tersebut hanyalah ingin memanfaatkan sebuah perangkat komputer yang harus dipakai bersama. Untuk mengerjakan beberapa proses tanpa banyak membuang waktu kosong dibuatlah proses beruntun (*Batch Processing*), sehingga beberapa program bisa dijalankan dalam sebuah komputer dengan dengan kaidah antrian.



Gambar 2.4 Jaringan Komputer Model TSS

Ditahun 1950-an ketika jenis komputer mulai membesar sampai terciptanya super komputer, maka sebuah komputer mesti melayani beberapa terminal (lihat Gambar 1) Untuk itu ditemukan konsep distribusi proses berdasarkan waktu yang dikenal dengan nama TSS (*Time Sharing System*), maka untuk pertama kali bentuk jaringan (*network*) komputer diaplikasikan. Pada sistem TSS beberapa terminal terhubung secara seri ke sebuah *host* komputer. Dalam proses TSS mulai nampak perpaduan teknologi komputer dan teknologi telekomunikasi yang pada awalnya berkembang sendiri-sendiri



Gambar 2.5 Jaringan komputer model distributed processing

Memasuki tahun 1970-an, setelah beban pekerjaan bertambah banyak dan harga perangkat komputer besar mulai terasa sangat mahal, maka mulailah digunakan konsep proses distribusi (*Distributed Processing*). Seperti pada Gambar 2.2, dalam proses ini beberapa *host* komputer mengerjakan sebuah pekerjaan besar secara paralel untuk melayani beberapa terminal yang tersambung secara seri disetiap *host* komputer. Dalam proses distribusi sudah mutlak diperlukan perpaduan yang mendalam antara teknologi komputer dan telekomunikasi, karena selain proses yang harus didistribusikan, semua *host* komputer wajib melayani terminal-terminalnya dalam satu perintah dari komputer pusat.

Selanjutnya ketika harga-harga komputer kecil sudah mulai menurun dan konsep proses distribusi sudah matang, maka penggunaan komputer dan jaringannya sudah mulai beragam dari mulai menangani proses bersama maupun komunikasi antar komputer (*Peer to Peer System*) saja tanpa melalui komputer pusat. Untuk itu mulailah berkembang

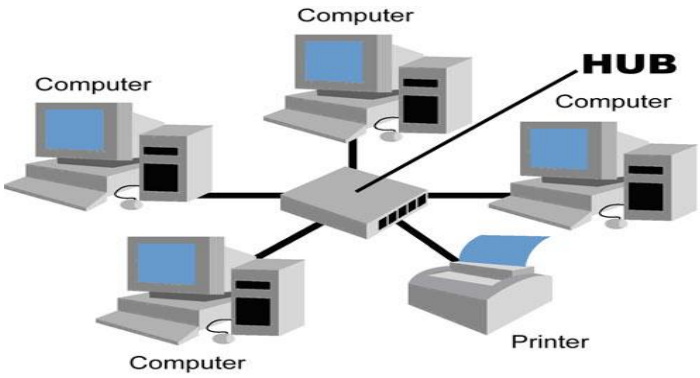
teknologi jaringan lokal yang dikenal dengan sebutan LAN. Demikian pula ketika Internet mulai diperkenalkan, maka sebagian besar LAN yang berdiri sendiri mulai berhubungan dan terbentuklah jaringan raksasa WAN

2.2.2 Topologi Jaringan

Topologi Jaringan adalah susunan lintasan aliran data didalam jaringan yang secara fisik menghubungkan simpul yang satu dengan simpul lainnya. Berikut ini adalah beberapa topologi jaringan yang ada dan dipakai hingga saat ini, yaitu:

2.2.2.1 Topologi Star

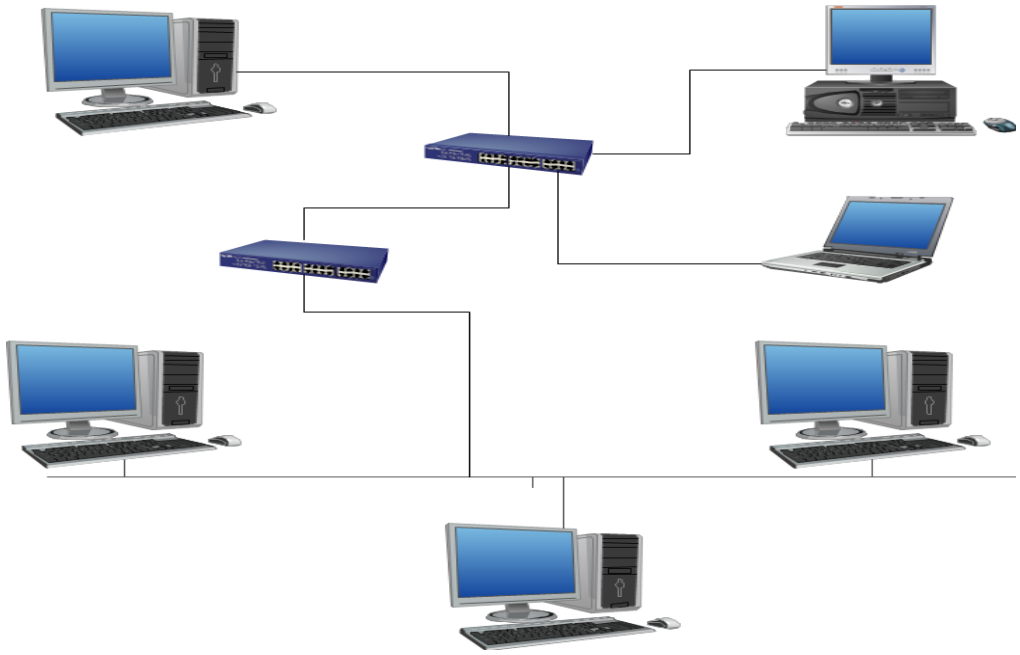
Beberapa simpul/*node* yang diswitchkan dengan simpul pusat/*host*, yang membentuk jaringan fisik seperti bintang, semua komunikasi ditangani langsung dan dikelola oleh *host* yang berupa mainframe komputer seperti Switch hub.Topologi star digunakan dalam jaringan yang padat, ketika *endpoint* dapat dicapai langsung dari lokasi pusat, kebutuhan untuk perluasan jaringan, dan membutuhkan kehandalan yang tinggi. Topologi ini merupakan susunan yang menggunakan lebih banyak kabel daripada bus dan karena semua komputer dan perangkat terhubung ke *central point*. Jadi bila ada salah satu komputer atau perangkat yang mengalami kerusakan maka tidak akan mempengaruhi yang jaringan yang lainnya. Berikut ini gambar topologi star sebagai gambaran.



Gambar 2.6 Topologi Star

2.2.2.2 Topologi Tree

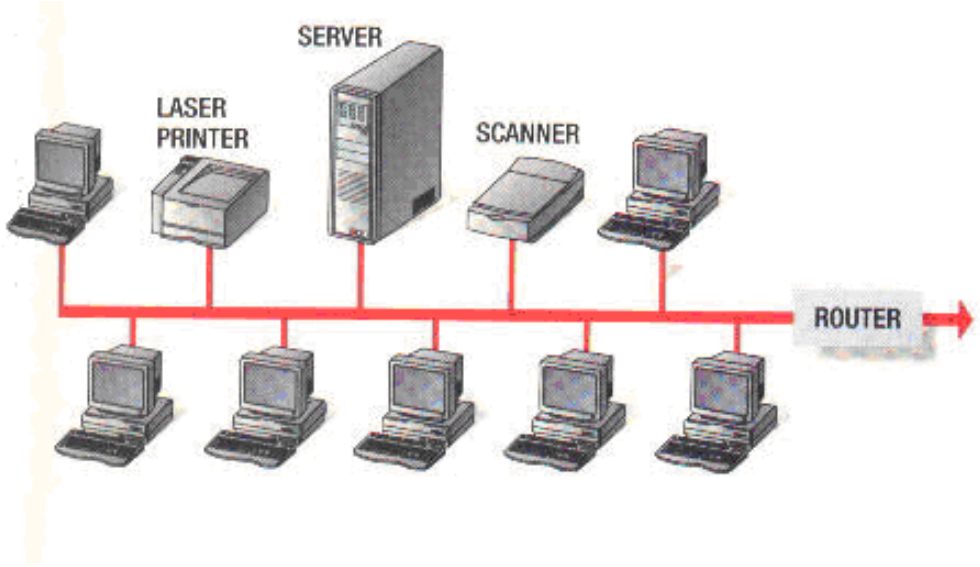
Berbentuk seperti pohon bercabang yang terditi dari komputer induk (*host*) yang diswitchkan dengan simpul atau *node* lain secara berjenjang, jenjang yang lebih tinggi berfungsi sebagai pengetur kerja jenjang dibawahnya, biasanya topologi ini digunakan oleh perusahaan besar atau lembaga besar yang mempunyai beberapa cabang daerah, sehingga data dari pusat bisa didistribusikan ke cabang atau sebaliknya. Berikut ini gambar topologi hierarkis sebagai gambaran.



Gambar 2.7 Topologi Tree

2.2.2.3 Topologi Bus

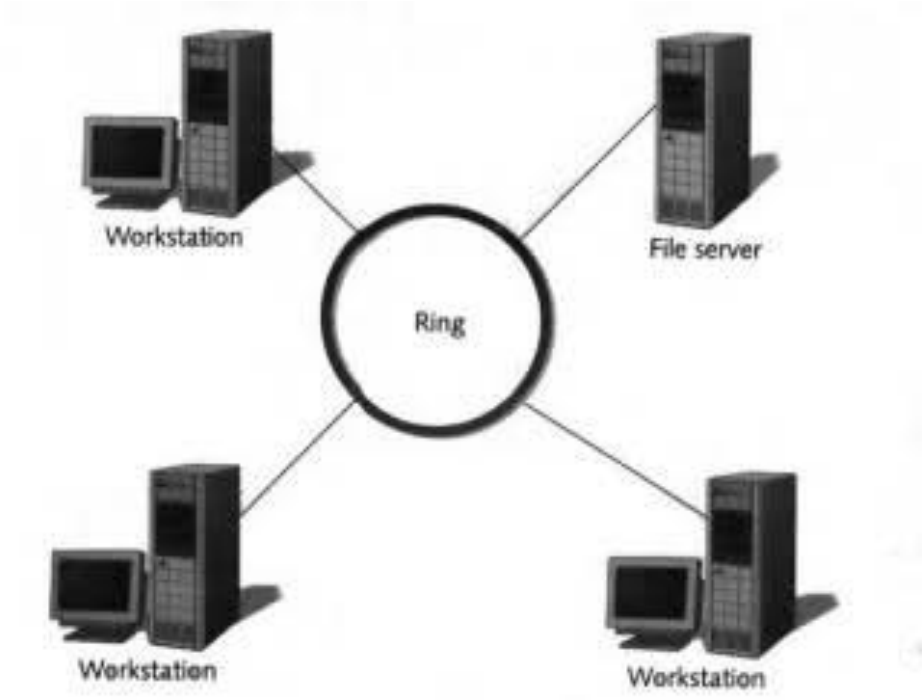
Beberapa simpul/*node* diswitchkan dengan jalur data (bus). Masing-masing *node* dapat melakukan tugas-tugas dan operasi yang berbeda namun semua mempunyai hierarki yang sama. Topologi ini biasanya menggunakan kabel Coaxial, yang sekarang sudah sangat jarang digunakan atau di implementasikan. Berikut ini gambar topologi bus sebagai gambaran.



Gambar 2.8 Topologi Bus

2.2.2.4 Topologi Ring

Di dalam topologi Ring semua *workstatio*n dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat- alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan. Kelemahan dari topologi ini adalah setiap *node* dalam jaringan akan selalu ikut serta mengelola informasi yang dilewatkan dalam jaringan, sehingga bila terdapat gangguan di suatu *node* maka seluruh jaringan akan terganggu. Keunggulan topologi Ring adalah tidak terjadinya *collision* atau tabrakan pengiriman data seperti pada topologi Bus, karena hanya satu *node* dapat mengirimkan data pada suatu saat, dan yang lainnya menggu hingga pengiriman data selesai. Berikut ini gambar topologi ring.

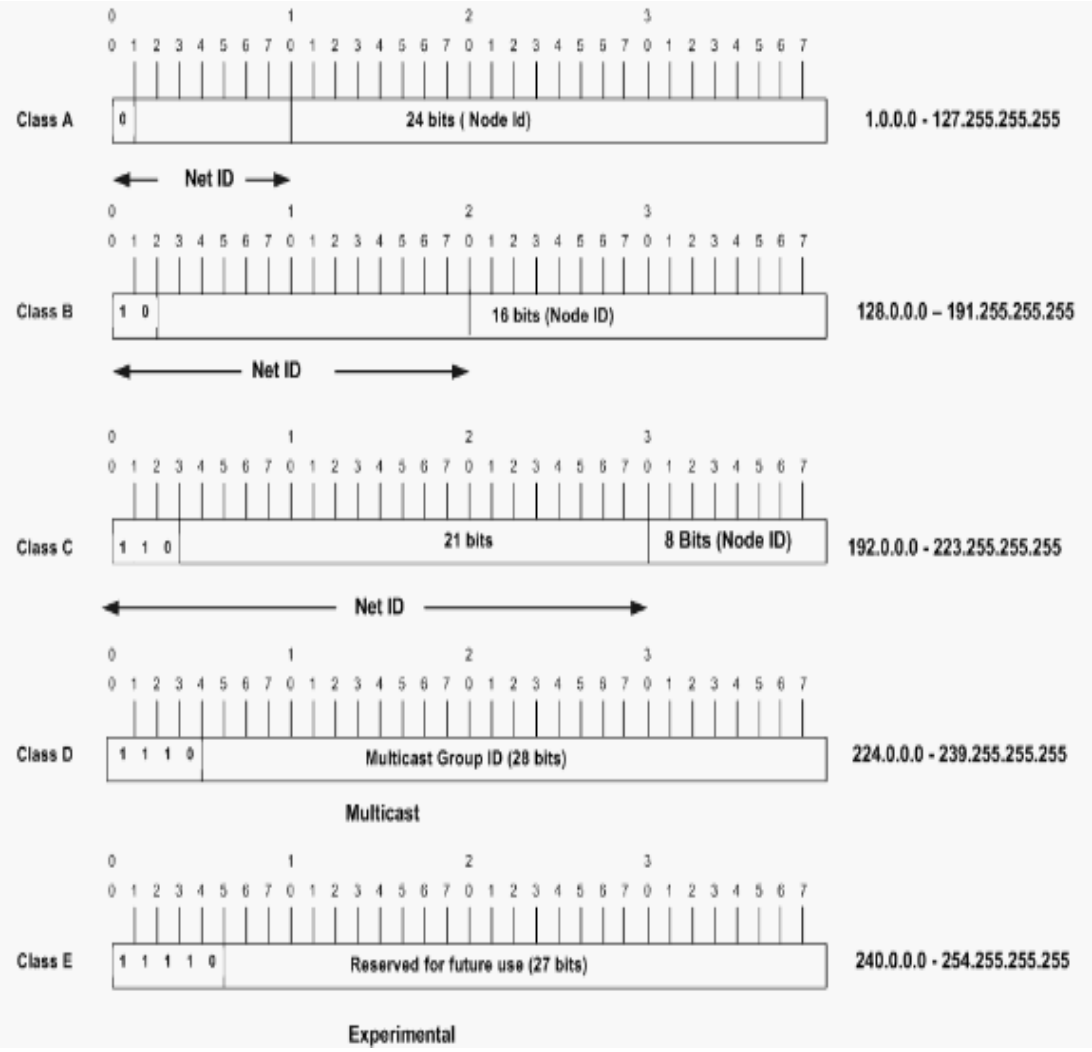


Gambar 2.9 Topologi Ring

2.2.3 *IP address*

IP address adalah alamat unik yang digunakan untuk mengidentifikasikan perangkat pada suatu jaringan IP. *IP address* terdiri dari 32 bit biner yang dapat dibagi ke dalam bagian jaringan dan bagian *host* dengan bantuan sebuah subnet mask. 32 bit biner dibagi menjadi empat *oktal* (1 *oktal* = 8 bit). Setiap *oktal* dikonversi ke decimal dan dipisahkan oleh titik. Nilai di setiap *oktal* berkisar dari 0 sampai dengan 255 desimal atau 00000000-11111111 dalam biner. Contoh *IP address* misalnya 172.16.81.100

Oktal oktal dalam *IP address* dipecah untuk menentukan skema pengalamatan yang dapat mengakomodasi jaringan besar dan jaringan kecil. Ada lima kelas jaringan yang berbeda, dari kelas A sampai kelas E. Setiap kelas dapat ditentukan dalam tiga *hight-order* bit. Gambar di bawah ini akan menunjukan secara signifikan dalam tiga *hight-order* bit dalam kisaran alamat yang masuk dalam masing masing kelas.



Gambar 2.10 Class IP

Dalam Kelas A, *oktal* pertama adalah bagian jaringan, sehingga Kelas A contoh pada diatas memiliki alamat jaringan utama 1.0.0.0 – 127.255.255.255. Oktet 2, 3, dan 4 (24 bit berikutnya) adalah untuk pengelola jaringan dibagi menjadi *subnet* dan *host*. Kelas A digunakan untuk jaringan yang memiliki lebih dari 65.536 *host* (sebenarnya, sampai dengan 16.777.214 *host*!).

Dalam Kelas B, dua oktet pertama adalah bagian jaringan, sehingga contoh Kelas B pada Gambar diatas memiliki alamat jaringan utama dari 128.0.0.0 – 191.255.255.255.

Oktet 3 dan 4 (16 bit) adalah untuk subnet lokal dan *host*. Kelas B alamat yang digunakan untuk jaringan yang memiliki 256 dan 65.534 *host*.

Dalam Kelas C, tiga oktet pertama adalah bagian jaringan. Contoh Kelas C pada Gambar 1 memiliki alamat jaringan utama dari 192.0.0.0 – 233.255.255.255. *Oktal* 4 (8 bit) adalah untuk subnet lokal dan *host*. Kelas ini sempurna untuk jaringan dengan kurang dari 254 *host*.

2.2.4 Subneting *IP address*

Subnetting *IP address* memungkinkan Anda untuk membuat multiple jaringan yang ada dalam jaringan A, B, atau C Class. Jika Anda tidak melakukan subneting, Anda hanya dapat menggunakan satu jaringan dari Kelas Anda A, B, atau C network, yang tidak realistis.

Setiap data link pada jaringan harus memiliki ID jaringan yang unik, dengan setiap *node* pada link menjadi anggota jaringan yang sama. Jika Anda memecah jaringan utama (Kelas A, B, atau C) menjadi *subnetwork* yang lebih kecil, memungkinkan Anda untuk membuat jaringan interkoneksi *subnetwork*. Setiap data link pada jaringan ini kemudian akan memiliki jaringan yang unik / *subnetwork* ID. Setiap perangkat, atau gateway, menghubungkan jaringan n / *subnetwork* memiliki n alamat IP yang berbeda, satu untuk setiap jaringan / *subnetwork* bahwa interkoneksi.

Untuk subnet jaringan, memperpanjang *natural mask* dengan menggunakan beberapa bit dari bagian *host* ID dari alamat untuk membuat *subnetwork* ID. Sebagai contoh, diberi jaringan Kelas C 204.17.5.0 dari yang memiliki *natural mask* 255.255.255.0, Anda bisa membuat subnet dengan cara ini:

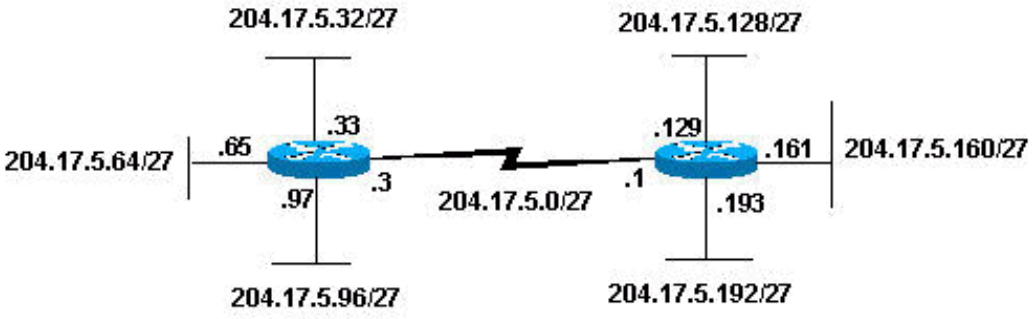
```
204.17.5.0 -      11001100.00010001.00000101.00000000
255.255.255.224 – 11111111.11111111.11111111.11100000
                            -------------------------|sub|----
```

Dengan memperluas masker menjadi 255.255.255.224, Anda telah mengambil tiga bit (ditandai dengan “sub”) dari bagian *host* asli dari alamat dan menggunakannya untuk membuat subnet. Dengan tiga bit, memungkinkan untuk menciptakan delapan subnet.

Dengan lima bit sisa *host* ID, subnet masing-masing dapat memiliki hingga 32 alamat *host*, 30 dari yang sebenarnya dapat diberikan ke perangkat karena *host* id dari semua nol atau semua yang tidak diperbolehkan (sangat penting untuk mengingat hal ini). Jadi, subnet telah diciptakan.

204.17.5.0 255.255.255.224	*host address* range 1 to 30
204.17.5.32 255.255.255.224	*host address* range 33 to 62
204.17.5.64 255.255.255.224	*host address* range 65 to 94
204.17.5.96 255.255.255.224	*host address* range 97 to 126
204.17.5.128 255.255.255.224	*host address* range 129 to 158
204.17.5.160 255.255.255.224	*host address* range 161 to 190
204.17.5.192 255.255.255.224	*host address* range 193 to 222
204.17.5.224 255.255.255.224	*host address* range 225 to 254

Dari subneting diatas akan diperoleh skema jaringan digambarkan oleh gambar di bawah ini:



Gambar 2.11 Contoh Model Skema *Subneting*

2.2.5 Pengertian *Server*

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentudalam sebuah jaringan komputer. *Server* didukung dengan prosesor yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebutsebagai sistem operasi jaringan atau *network operating system*. *Server* juga menjalankanperangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (printer),

danmemberikan akses kepada *workstation* anggota jaringan. Umumnya, di atas sistem operasi *server* terdapat aplikasi- aplikasi yang menggunakan arsitektur *client*/*server*.

Setiap sistem operasi *server* umumnya membundel layanan-layanan tersebut atau layanan tersebut juga dapat diperoleh dari pihak ketiga. Setiap layanan tersebut akan merespons terhadap *request* dari *client*. Sebagai contoh, *client* DHCP akan memberikan *request* kepada *server* yang menjalankan *server* DHCP; ketika sebuah *client* membutuhkan alamat IP, *client* akan memberikan perintah/ *request* kepada *server*, dengan bahasa yang dipahami oleh *server* DHCP, yakni protokol DHCP itu sendiri. Contoh sistem operasi *server* adalah Windows NT 3.51, dan dilanjutkan dengan Windows NT 4.0. Saat ini sistem yang cukup populer adalah Windows 2000 *Server* dan Windows *Server* 2003, kemudian Sun Solaris, Unix, dan GNU/Linux. *Server* biasanya terhubung dengan *client* dengan kabel UTP dan sebuah Network Card. Kartu jaringan ini biasanya berupa kartu PCI atau ISA. Fungsi *server* sangat banyak, misalnya untuk situs internet, ilmu pengetahuan, atau sekedar penyimpanan data. Namun yang paling umum adalah untuk mengkoneksikan komputer *client* ke Internet. *Server* adalah suatu komputer yang menjadi pengelola dan pusat bagi komputer lainnya

Oleh karena itu komputer *server* haruslah memiliki spesifikasi yang lebih tinggi dari pada *client-client* nya. Selain itu *server* memiliki macam-macam jenis, yaitu diantaranya :

1. FTP *server*
2. DNS *server*
3. Web *Server*
4. *E-mail server*
5. Proxy *Server*

2.2.5.1 FTP *Server*

File Transfer Protocol (FTP) adalah suatu protokol yang berfungsi untuk tukar-menukar file dalam suatu network yang menggunakan TCP koneksi bukan UDP. Dua hal yang penting dalam FTP adalah FTP *Server* dan FTP *Client*. FTP *server* adalah suatu *server* yang menjalankan software yang berfungsi untuk memberikan layanan tukar

menukar file dimana *server* tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan (*request*) dari FTP *client*.

FTP *client* adalah komputer yang me-*request* koneksi ke FTP *server* untuk tujuan tukar menukar file. Setelah terhubung dengan FTP *server*, maka *client* dapat *men-download, meng-upload, merename, men-delete*, dll sesuai dengan ijin yang diberikan oleh FTP *server*.

Tujuan dari FTP *server* adalah sebagai berikut :

a. Untuk tujuan *sharing* data.
b. Untuk menyediakan *indirect atau implicit remote computer*.
c. Untuk menyediakan tempat penyimpanan bagi user.
d. Untuk menyediakan transfer data yang *reliable* dan efisien.

2.2.5.2 DNS *Server*

Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang mengunakan TCP/IP (Transmission Control Protocol/*Internet Protocol*). DNS biasa digunakan pada aplikasi yang terhubung ke Internet seperti web browser atau *e-mail*, dimana DNS membantu memetakan *host* name sebuah komputer ke *IP address*. DNS dapat disamakan fungsinya dengan buku telepon. Dimana setiap komputer di jaringan Internet memiliki *host* name (nama komputer) dan *Internet Protocol* (IP) *address*. Secara umum, setiap *client* yang akan mengkoneksikan komputer yang satu ke komputer yang lain, akan menggunakan *host* name. Lalu komputer anda akan menghubungi DNS *server* untuk mencek *host* name yang anda minta tersebut berapa *IP address*-nya. *IP address* ini yang digunakan untuk mengkoneksikan komputer anda dengan komputer lainnya.

2.2.5.3 Web *Server*

Web server adalah *server* yang melayani permintaan *client* terdapat halaman web seperti *apache, IIS (Internet Information Server)* dan berkomunikasi dengan Middleware

untuk menterjemahkan kode-kode tertentu, menjalankan kode-kode tersebut dan memungkinkan berinteraksi dengan basis data, PHP atau ASP.

Web *Server* merupakan sebuah perangkat lunak *server* yang berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan web browser dan mengirimkan kembali hasilnya dalam bentuk halaman- halaman web yang umumnya berbentuk dokumen HTML. Selain itu Web *Server* juga merupakan sebuah komputer yang menyediakan layanan untuk internet. *Server* disebut juga dengan *host*.

Agar anda dapat memasukkan web yang anda rancang ke dalam internet, maka anda harus memiliki ruangan terlebih dahulu dalam internet, dan ruangan ini disediakan oleh *server*.

Adapun arsitektur aplikasi *server* adalah sebagai berikut :

1. Browser atau *client* berinteraksi dengan web *server*.
2. Secara internal web *server* berinteraksi dengan *middleware.*
3. *Middleware* yang berhubungan dengan database.

Adapun teknologi yang berjalan di *server* antara lain : CGI *(Common Gateway Interface),* ASP *(Aktive Server Page*), JSP *(Java Server Page)* dan *PHP*

2.2.5.4 *E-mail Server*

E-mail merupakan sebuah layanan pengiriman surat elektronik yang dikirim melalui internet. *E-mail* dikirim dari suatu alamat *e-mail* yang terdapat pada sebuah mail *server* kepada alamat *e-mail* yang lainnya yang terdapat pada mail *server* yang sama maupun pada mail *server* yang berbeda. Fungsi *e-mail server* untuk melakukan penyimpanan (*storing*) dan distribusi yang berupa pengiriman (*sending*), penjaluran (*routing*), dan penerimaan (*receiving) e-mail*

E-mail dapat dianalogikan dengan kotak surat yang ada di kantor POS sedangkan *server e-mail* dapat diibaratkan sebagai kantor POS. dengan analogi ini sebuah mail *server* dapat memiliki banyak *akun e-mail* yang ada didalamnya.

Mail *Server* berjalan dengan beberapa protokol pada TCP/IP, yakni SMTP (port 25), POP3 (port 110), dan IMAP (port 143). Mail *Server* memiliki tiga komponen utama, yakni Mail Transfer Agent (MTA), Mail Delivery Agent (MDA), dan Mail User Agent (MUA).

Untuk mengirim sebuah *e-mail* dari alamat *e-mail* yang satu ke alamat *e-mail* yang lain digunakan sebuah protocol (aturan) yaitu *Simple Mail Transfer Protocol SMTP*. Protokol SMTP telah menjadi aturan dasar yang disepakati untuk pengiriman *e-mail*. Dengan demikian semua software *e-mail server* pasti mendukung protokol ini.

SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman surat elektronik di Internet. SMTP merupakan protokol yang digunakan untuk berkomunikasi dengan *client*. Sedangkan untuk *client*, digunakan protokol *imap imaps pop3 pop3s.*

POP3 (*Post Office Protocol version 3*) adalah protokol yang digunakan untuk mengambil surat elektronik (*e-mail*) dari *server e-mail*. Protokol POP3 dibuat karena desain dari sistem surat elektronik yang mengharuskan adanya *server* surat elektronik yang menampung surat eletronik untuk sementara sampai surat elektronik tersebut diambil oleh penerima yang berhak.

Supaya sebuah mail *server* dapat diakses oleh *client*, dikembangkan sebuah aplikasi dimana *client* dapat mengakses *e-mail* dari sebuah *e-mail server*. IMAP (*Internet Message Access Protocol*) adalah protokol standar untuk mengakses/mengambil *e-mail* dari *server*. IMAP memungkinkan pengguna memilih pesan *e-mail* yang akan diambil, membuat folder di *server*, mencari pesan *e-mail* tertentu, bahkan menghapus pesan *e-mail* yang ada. IMAP dan POP3 memiliki fungsi yang sama, akan tetapi memiliki karakteristik yang berbeda dalam cara pengaksesan pada *server*. Untuk dapat melakukan tugasnya, sebuah mail *server* harus dapat melayani *e-mail* yang mempergunakan protokol SMTP serta harus mampu melayani *client* yang ingin mengakses *e-mail* dengan menyediakan IMAP atau POP3.

MDA adalah komponen yang bertanggung jawab untuk pengiriman pesan ke kotak masuk surat penerima lokal. MDA bekerjasam dengan MTA untuk menangani pesa-pesan

e-mail yang datang untuk di;etakan/di distribusikan sesuia pada *mailbox user* masing masing.

MTA adalah sebuah aplikasi untuk mengantarkan *e-mail* dan mengatur penerimaan *e-mail*. MTA melakukan fungsi fungsi sebagai berikut :

a. Pertukaran *e-mail* menggunakan protokol TCP
b. Menerima *e-mail* masuk (*incoming*)
c. Meneruskan *e-mail* yang akan keluar (*outgoing*)
d. Mengatur antrian bila ada *e-mail* masuk, keluar dan yang tertunda pengirimannya.

MTA yang umun dipakai adalah send mail dan qmail serta postfix untuk di unix serta untuk di ms windows menggunakn mdaemon.

MUA adalah aplikasi yang berfungsi sebagai antar muka (*interface*) antara *e-mail* (dalam hal ini berhubungan denga user yang memiliki *e-mail* tersebut) dengan MTA yang mendukungnya. MUA berfungsi :

a. Menulis *e-mail* dan membaca *e-mail* yang masuk.
b. Mengatur konfigurasi *e-mail* sehingga sesuai dengan MTA yang mendukungnya.
c. Memberikan kenyamanan kepada user dalam menerima dan mengirim *e-mail*.

Beberapa agen *e-mail* yang populer saat ini adalah roundcube , eudora, netscape, outlook, evolution mail, thunderbird dan pegassus.

2.2.5.5 Proxy *Server*

Proxy *server* adalah sebuah komputer *server* atau program komputer yang dapat bertindak sebagai komputer lainnya untuk melakukan *request* terhadap content dari Internet atau intranet. Proxy *Server* bertindak sebagai gateway terhadap dunia Internet untuk setiap komputer *client*. Proxy *server* tidak terlihat oleh komputer *client*: seorang pengguna yang berinteraksi dengan Internet melalui sebuah proxy *server* tidak akan mengetahui bahwa

sebuah proxy *server* sedang menangani *request* yang dilakukannya. Web *server* yang menerima *request* dari proxy *server* akan menginterpretasikan *request-request* tersebut seolah-olah *request* itu datang secara langsung dari komputer *client*, bukan dari proxy *server*.

Proxy *server* juga dapat digunakan untuk mengamankan jaringan pribadi yang dihubungkan ke sebuah jaringan publik (seperti halnya Internet). Proxy *server* memiliki lebih banyak fungsi daripada router yang memiliki fitur packet filtering karena memang proxy *server* beroperasi pada level yang lebih tinggi dan memiliki kontrol yang lebih menyeluruh terhadap akses jaringan. Proxy *server* yang berfungsi sebagai sebuah "agen keamanan" untuk sebuah jaringan pribadi, umumnya dikenal sebagai *firewall*.

2.2.6 Linux

Linux adalah sebuah sistem operasi open source dan bebas (free) di bawah lisensi GNU (GNU is Not Unix) GPL (General Public License). Arti *open source* adalah kode sumber (source code) diikutsertakan dalam program Linux sehingga dapat dilihat oleh siapa saja tanpa harus mendatangi suatu perjanjian khusus seperti NDA (*Non Disclosure Agreement*). Arti kata bebas atau *free* mengacu pada kebebasan bukan pada bebas biaya.

2.2.7 LAMP Server

LAMP adalah istilah yang merupakan singkatan dari Linux, Apache, MySQL dan Perl/PHP/Phyton. Merupakan sebuah paket perangkat lunak bebas yang digunakan untuk menjalankan sebuah aplikasi secara lengkap.

Komponen-komponen dari LAMP:

1. Linux sebagai sistem operasi
2. Apache HTTP Server sebagai web server
3. MySQL sebagai sistem basis data
4. Perl atau PHP atau Pyton sebagai bahasa pemrograman yang dipakai

Beberapa perangkat lunak yang menggunakan konfigurasi LAMP antara lain MediaWiki dan Bugzilla.

2.2.8 SSH (*Secure Shell*)

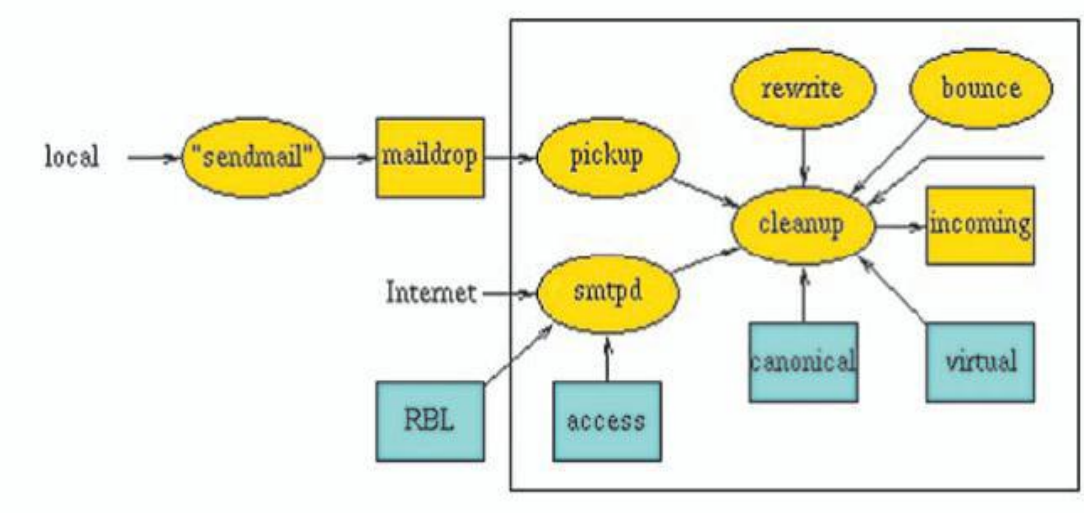
SSH menggunakan kriptografi kunci publik untuk mengotentikasi komputer *remote* dan biarkan komputer *remote* untuk mengotentikasi pengguna, jika perlu. SSH biasanya digunakan untuk login ke mesin *remote* dan mengeksekusi berbagai perintah, tetapi juga mendukung *tunneling, forwarding TCP port* dan X11 connections untuk mentransfer file menggunakan SFTP atau SCP protocols. SSH menggunakan *client-server* model. Yang standar TCP port 22 telah ditetapkan untuk menghubungi *server* SSH. Sebuah klien program SSH ini biasanya digunakan untuk membangun koneksi ke SSH daemon untuk dapat di-remote. Keduanya biasanya terdapat pada sistem operasi modern, termasuk Mac OS X, Linux, FreeBSD, Solaris dan OpenVMS. Tersedia versi berpemilik, freeware dan open source untuk berbagai tingkat kerumitan dan kelengkapan.

2.2.9 WinSPC

WinSPC adalah *software file transport client* yang berfungsi utamanya menyalin file secara aman antara lokal dan untuk meremote komputer lain serta dapat mengelola beberapa tindakan file manajement lainnya. WinSPC menggunakan teknologi SSH (*Secure Shell*) yang memungkinkan aman dalam menyalin file antar lokal dan meremote PC menggunakan FTP (*File Transfer Protocol*), SSH FTP atau SCP Protocol (*Secure Copy protocol*) yang menawarkan fitur manajemen file dasar.

2.2.10 Postfix

Postfix berasal dari IBM *Secure Mailer* yang dirilis sekitar tahun 1998, yang kemudian diberi nama VMailer. Tapi karena alasan kemiripan dengan merek dagang yang telah ada, pemakaian nama ini tidak dilanjutkan dan diganti dengan postfix. Proyek postfix dimulai oleh Wietse Venema, pakar keamanan komputer dan periset di IBM. Postfix berusaha memberikan *alternative* untuk program *mail* Sendmail yang telah luas memasyarakat. Postfix dirancang lebih cepat, lebih mudah dirawat, dan lebih aman.



Gambar 2.12 Cara kerja postfix

2.2.11 PostfixAdmin

PostfixAdmin adalah antarmuka berbasis *web* yang digunakan untuk mengelola surat, domain virtual dan alias. *PostfixAdmin* digunakan sebagai *management user, domain virtual, alias* dan *mailbox* dari *database postfix* yang sebelumnya di buat pada saat konfigurasi *postfix*

2.2.12 Dovecot

Dovecot merupakan sebuah aplikasi yang dijalankan untuk mengikuti protokol imap imaps pop3 pop3s. Dengan mempergunakan protokol tersebut, kita dapat mengakses *e-mail* yang ada di server. Sehingga kita dapat melakukan pengaksesan *e-mail* dimanapun juga dan kapan saja

2.2.13 Email Security

Spam atau *junk mail* adalah penyalahgunaan dalam pengiriman berita elektronik untuk menampilkan berita iklan dan keperluan lainnya yang mengakibatkan ketidaknyamanan bagi para pengguna web. Penanganan *spam* dapat ditangani dengan aplikasi seperti :

1. Amavisd-New
2. Clam AntiVirus
3. Spamassassin

2.2.13.1 Amavisd-New

Amavisd-New adalah *content filter* yang ditulis menggunakan perl (*Net Server)* dan berjalan secara *daemonized* sehingga memungkinkan penanganan lalu lintas email besar tanpa harus banyak memakan *resource* komputer yang tersedia. Selain itu, amavisd-new dapat menerima email yang akan diperiksa melalui SMTP sehingga dapat dipadukan dengan mudah dengan hampir semua MTA yang ada. Amavis dapat juga dapat difungsikan sebagai mail gateway.

Dalam memeriksa *e-mail*, amavisd-new menyimpan *hash* tiap *e-mail* yang masuk. Bila hash *e-mail* yang masuk belum tersimpan maka amavis akan memeriksa email tersebut secara utuh. Bila hashnya sudah ditemukan maka email tersebut tidak diperiksa secara utuh, melainkan keputusannya sesuai dengan hash yang sudah ada. Bila sebelumnya email dengan jenis *hash* itu di-drop, maka email baru ini akan di-drop, dan sebaliknya. Amavis dapat digabungkan dengan beberapa antivirus seperti ClamAV, f-prot, grisoft, kaspersky, dll

2.2.13.2 Clam AntiVirus

Clam AntiVirus adalah aplikasi antivirus yang open source untuk UNIX, didesain khusus untuk melakukan pemeriksaan email pada *mail gateway*. Juga menyediakan beberapa fasilitas termasuk fungsi *multi-thread* yang fleksibel dan handal, terdapat pula fungsi update database secara otomatis.

2.2.13.3 Spamassassin

SpamAssassin adalah program komputer yang dijalankan pada Apache dan dipakai untuk menyaring spam terhadap surat elektronik yang masuk. Prinsip kerja dari SpamAssassin ini adalah berdasarkan pengecekan isi e-mail (*content-matching rules*),

pengecekan alamat DNS, pengecekan *checksum* isi, dan berdasarkan pada penyaringan secara statistik, yang didukung oleh basis data *online*. SpamAssassin ini secara umum dianggap paling efektif dalam memerangi spam, terutama jika dikombinasikan dengan basisdata spam.

2.2.14 Roundcube

Roundcube adalah *web-based* IMAP *e-mail client* yang ditulis dalam bahasa pemrograman PHP. Roundcube sudah menggunakan teknologi Ajax untuk menyajikan antarmuka pengguna yang lebih interaktif dan responsif dibandingkan dengan *client e-mail* berbasis web biasa. Rilis pertama kali oleh Roundcube pada awal tahun 2008.