

BAB II

LANDASAN TEORI

2.1 Server

Server, sesuai dengan namanya bisa diartikan sebagai pelayan pada suatu jaringan komputer. *Server* adalah komputer yang berfungsi untuk melayani, membatasi, dan mengontrol akses terhadap user-user dan sumber daya pada suatu jaringan komputer. *Server* merupakan suatu bagian terpenting dari sebuah jaringan. suatu jaringan komputer dengan banyak komputer memerlukan suatu server yang bertugas untuk menyediakan layanan yang dibutuhkan oleh user. [1]

Server memiliki beberapa macam jenis. Dari berbagai macam jenis *server* tersebut memiliki kegunaan dan tugasnya masing – masing. Dan setiap jenis *server* tersebut bisa dijadikan satu mesin. Jadi satu komputer *server* bisa melayani berbagai *service* atau layanan dan juga bisa menggunakan sistem *cluster server* yang membagi setiap layanan pada *node-node* yang tersedia. [2]

2.1.1 Macam-macam Server

1. Web Server

Server ini merupakan server yang mungkin tidak aneh lagi didengar. Situs seperti search engine, blog, website dll pasti membutuhkan web server agar file – file web seperti .html .php .asp bisa diproses terlebih dahulu dalam web server sebelum file tersebut dikirimkan ke client. Web server melayani request servis dengan protokol HTTP, dan ada juga IMAP ataupun FTP.

2. Database Server

Server database, seperti namanya. Server ini melayani servis bagi client yang membutuhkan layanan penyimpanan database. Dalam server database tersebut, bisa berisi ratusan ataupun ribuan database dari banyak user. Biasanya database tersebut dikelompokkan atau disimpan per user yang memakai layanan database tersebut. Agar tidak terjadinya pencurian data.

3. FTP Server

Server ini mempunyai layanan protokol FTP. Artinya dengan adanya server FTP ini, segala permintaan FTP (FTP merupakan protokol untuk layanan transfer data) akan terlaksana.

4. Mail Server

Mail server mempunyai peran untuk melayani client dalam hal mail (surat). Surat dalam hal ini, client bisa mengirimkan pesan ke client yang lain, yang dimana surat tersebut disimpan dalam server mail itu juga. Selain sebagai tempat untuk menyimpan surat, server mail juga menyediakan layanan pelengkap mail servis seperti web interface untuk memudahkan client menulis surat dan mengorganisir surat yang dipunya client.

5. Proxy Server

Proxy server, bisa digunakan untuk memperkuat security pada sebuah jaringan. Server ini bisa digunakan untuk memfilter permintaan servis masuk maupun keluar dari sebuah client. Dengan adanya proxy server ini, kita bisa memblock suatu IP agar bisa berkomunikasi misalnya.

6. DNS Server

DNS server mempunyai kegunaan untuk servis DNS. DNS berguna untuk menerjemahkan alamat domain sebuah komputer ke dalam bentuk IP. Oleh sebab itu jika kita tidak menentukan DNS server pada settingan IP Address, maka kita akan kesulitan untuk membuka situs seperti Google, karena alamat google tersebut harus diterjemahkan dulu ke dalam bentuk IP address sebelum komputer kita bisa terkoneksi ke server google tersebut.. Sedangkan alamat IP server dns belum ditentukan oleh kita.

7. Game Server

Game server merupakan server yang digunakan untuk pusat untuk menghubungkan antar pemain (client) dengan pemain yang lainnya. Game server ini selain merupakan server tersendiri, bisa juga didirikan dari komputer client yang bermain game tersebut.

8. Print Server

Server ini bisa digunakan untuk pusat layanan percetakan (print) bagi client. Dengan adanya server ini, segala aktivitas “print” bisa dilakukan secara bersama dan menghemat biaya hardware karena meskipun terdapat satu printer, printer tersebut bisa digunakan oleh banyak komputer.

9. Home Server

Server ini merupakan server yang khusus bagi pengguna rumah yang membutuhkan layanan seperti home entertainment. Komputer ini bisa juga ditambah layanan servis lainnya seperti printing servis, web servis, maupun file servis sebagai backup data. Sehingga komputer ini memiliki berbagai fungsi.

2.2 Jaringan Komputer

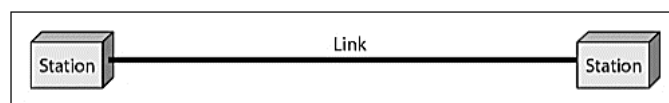
Jaringan komputer adalah sekumpulan perangkat atau *node* yang terhubung dengan jaringan komunikasi. Sebuah *node* dapat berupa komputer, printer atau perangkat apapun yang dapat mengirimkan atau dapat menerima data yang dihasilkan oleh *node* lainnya dalam jaringan. [3]

2.2.1 Struktur Fisik Jaringan

Sebuah jaringan adalah dua atau beberapa perangkat yang saling terhubung melalui sebuah *link*. *Link* adalah sebuah jalur komunikasi untuk mengirimkan data dari satu perangkat ke perangkat lainnya. Ada dua tipe koneksi, yaitu *point-to-point* dan *multipoint*. [3]

1. Point-to-point

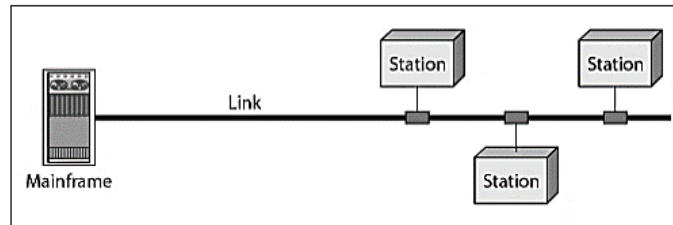
Koneksi *point-to-point* adalah dua buah perangkat yang saling berhubungan menggunakan sebuah *link*. Kebanyakan koneksi *point-to-point* menggunakan kabel, namun dapat pula menggunakan *microwave* atau *link satelit*. Berikut Gambar 2.1 merupakan gambaran dari *point-to-point*.



Gambar 2.1 Point-to-point

2. Multipoint

Koneksi *multipoint* adalah sebuah koneksi antara satu atau beberapa perangkat dalam satu *link* yang digunakan bersama. Berikut Gambar 2.2 merupakan gambaran dari *multipoint*.



Gambar 2.2 Multipoint

2.2.2 Kategori Jaringan

Kategori jaringan berdasarkan jangkauan geografis.

1. LAN (*Local Area Network*)

LAN merupakan jaringan komputer yang hanya mencakup wilayah kecil seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Beberapa teknologi yang digunakan pada LAN adalah ethernet, Token Ring, Token Bus, FDDI, dan ATM LAN. Namun untuk jaman sekarang, kebanyakan LAN berbasis pada teknologi IEEE 802.3 Ethernet yang mempunyai kecepatan transfer data 10, 100, 1.000, 10.000, 40.000 atau 100.000 Mbit/s.

2. MAN (*Metropolitan Area Network*)

Jaringan dalam suatu kota dengan transfer data berkecepatan tinggi, yang menghubungkan berbagai lokasi seperti kampus, perkantoran, pemerintahan, dan sebagainya. Jaringan MAN memiliki jangkauan antar 10 hingga 50 km. Biasanya MAN meliputi jaringan antar kantor-kantor dalam satu kota antara pabrik/instansi dan kantor pusat yang berada dalam jangkauannya.

3. WAN (*Wide Area Network*)

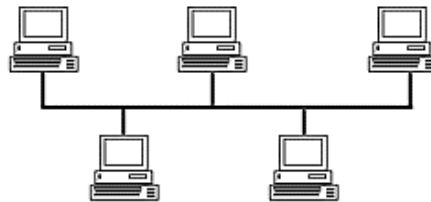
Jaringan komputer yang mencakup area lokasi yang lebih luas, dengan melibatkan kesatuan komputer yang lebih banyak. Hal ini sudah jelas berbeda dengan MAN atau mungkin LAN yang dibatasi dengan ruangan dan bangunan. WAN tidak dibatasi dengan ruangan ataupun bangunan bahkan cakupannya bisa sampai dengan antar benua.

2.2.3 Topologi Jaringan

Beberapa jenis dari topologi fisik sebagai berikut :

1. Topologi Bus

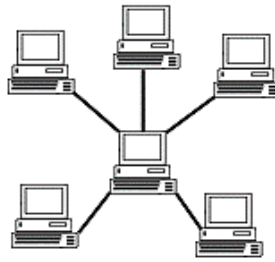
Topologi bus yaitu topologi yang di setiap terminalnya terhubung ke sebuah jalur utama komunikasi data. Informasi atau data yang akan dikirim dan diambil harus melalui jalur utama komunikasi data dan melewati workstation (komputer). Kelebihan topologi Bus adalah biaya instalasi yang murah dan pemasangannya mudah. Kekurangannya adalah jika ada satu jalur yang putus maka satu jaringan tersebut akan berhenti atau tidak dapat digunakan.



Gambar 2.3 Topologi Bus

2. Topologi Star

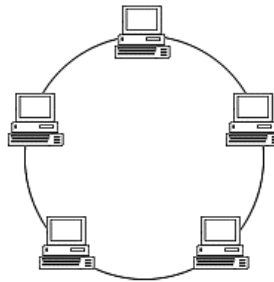
Topologi Star yaitu topologi yang memiliki pusat berupa hub atau switch yang berfungsi untuk menghubungkan dan meneruskan pengiriman data antar perangkat. Kelebihan dari topologi Star adalah mudah melakukan pelacakan ketika terjadi suatu masalah fisik, mudah untuk melakukan penambahan dan pengurangan perangkat, jika terjadi suatu masalah pada suatu jalur maka tidak mempengaruhi jalur yang lain. Kekurangannya adalah jika pusat penghubung jaringan (hub / switch) rusak atau mati maka setiap perangkat yang terhubung pada pusat tersebut tidak dapat melakukan pengiriman atau pengambilan data dan membutuhkan kabel yang lebih banyak dibandingkan dengan topologi Bus dan topologi Ring.



Gambar 2.4 Topologi Star

3. Topologi Ring

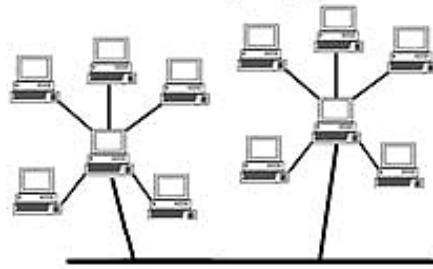
Topologi ring yaitu topologi yang mana setiap perangkat terhubung secara seri dengan yang lain hingga membentuk suatu lingkaran yang tertutup (loop). Kelebihan dari topologi ini adalah mudah dalam pembangunannya dan biaya untuk pembangunan topologi ini masih lebih murah dibandingkan dengan topologi star. Kekurangannya adalah topologi ini agak sulit dikembangkan ke skala yang lebih besar dan jika di suatu jaringan ada kabel yang putus maka jaringan tersebut akan berhenti.



Gambar 2.5 Topologi Ring

4. Topologi Tree

Topologi tree merupakan gabungan dari dua topologi fisik yaitu topologi bus dan gabungan dari beberapa topologi star. Namun pada topologi ini, jika kabel pada topologi bus terputus maka setiap jaringan pada topologi star masih berjalan dengan baik hanya saja topologi star disuatu jaringan tidak dapat terhubung dengan topologi star lainnya. Topologi ini biasanya digunakan untuk jaringan berskala besar yaitu WAN. Meskipun namanya adalah topologi bus, kabel yang digunakan tidak hanya coaxial saja, namun juga bisa berupa serat optik.



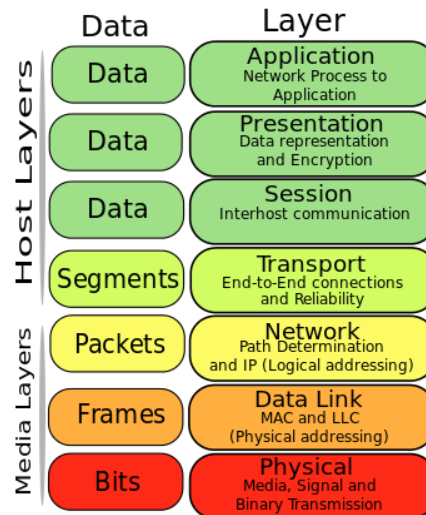
Gambar 2.6 Topologi Tree

2.3 Protokol Jaringan Komputer

Pada jaringan komputer terdapat aturan-aturan yang bertujuan untuk mengatur komunikasi data. Peraturan tersebut dibentuk dalam sebuah model yang disebut protokol. Protokol adalah sebuah aturan atau standar yang mengatur komunikasi data. Sebuah protokol mendefinisikan tentang apa, bagaimana dan kapan data dikomunikasikan. [3]

2.3.1 Model OSI (Open Systems Interconnection)

Model OSI bukanlah suatu protokol melainkan suatu model yang digunakan untuk memahami dan mendesain arsitektur jaringan yang fleksibel, robust, dan dapat dioperasikan. Model OSI juga menggambarkan lapisan kerangka kerja untuk desain sistem jaringan yang memungkinkan komunikasi antara semua tipe sistem komputer. Terdiri dari tujuh lapisan yang terpisah tetapi memiliki relasi. Tujuan dibuatnya model OSI adalah untuk memperlihatkan bagaimanana mempermudah komunikasi antara sistem yang berbeda tanpa harus merubah logika dasar *hardware* dan *software*.



Gambar 2.7 Model OSI

Berikut penjelasan untuk setiap lapisan :

1. *Application Layer* (Lapisan Aplikasi)

Lapisan ini menyediakan antar muka bagi *user* (pengguna) untuk menggunakan layanan seperti e-mail, unduh dan unggah *file*, *remote* perangkat, dan layananlainnya.

2. *Presentation Layer* (Lapisan Presentasi)

Lapisan ini berhubungan dengan format data yang digunakan antara dua sistem yang saling bertukar data. Memiliki tugas utama yaitu :

- Translasi yang berguna untuk menerjemahkan kode data dari setiap sistem yang berbeda-beda. Pengkodean tersebut diperlukan karena setiap sistem menggunakan metode pengkodean yang berbeda.
- Enkripsi yang berguna untuk mengubah informasi asli yang akan dipertukarkan menjadi bentuk lain agar aman ketika melalui jalur pengiriman.
- Kompresi yang berguna untuk mengurangi jumlah bit pada informasi yang akan dipertukarkan.

3. *Session Layer* (Lapisan Sesi)

Lapisan sesi berguna untuk melakukan pembangunan, menjaga, dan mensinkronisasikan interaksi antara sistem yang akan berkomunikasi. Tanggung jawabnya adalah sebagai *Dialog Control* yaitu lapisan sesi yang memungkinkandua sistem untuk membangun dialog serta bertanggung

jawab untuk *Synchronization* yaitu proses untuk menambahkan titik-titik pemeriksaan pada aliran data.

4. *Transport Layer* (Lapisan Transport)

Lapisan ini bertanggung jawab untuk pengiriman proses ke proses dari seluruh pesan. Tugas lain dari lapisan ini sebagai berikut :

- *Service-point addressing* yang berarti untuk memberikan alamat port atau untuk membedakan proses(program berjalan) yang akan digunakan.
- *Segmentation and Reassembly*. Berguna memastikan seluruh pesan yang sampai di bagian penerima dalam keadaan utuh dan sesuai urutan dan melakukan identifikasi serta penggantian paket jika terjadi kehilangan pada saat proses pengiriman.
- *Connection Control*, Lapisan transport bisa berupa connectionless atau connection-oriented. Jika Connectionless, maka setiap segmen diperlakukan sebagai paket bebas dan langsung dikirimkan ke perangkat tujuan. Jika Connection-oriented, sebelum mengirimkan paket maka terlebih dahulu membuat koneksi ke perangkat tujuan. Setelah data terkirim, koneksi akan diputus.
- *Flow Control*, untuk mengatur kecepatan pengiriman data agar tidak dibuang akibat ketidakmampuan penerima(tujuan) dalam menerima data dari sumber pengirim.
- *Error Control*. Memastikan seluruh pesan sampai pada lapisan transport di penerima tanpa ada masalah.

5. *Network Layer* (Lapisan Jaringan)

Lapisan jaringan bertanggung jawab untuk pengiriman paket dari sumber pengirim ke penerima (tujuan akhir) meskipun melewati beberapa jaringan. Tanggung jawab lainnya di lapisan ini yaitu melakukan fungsi pengalamatan logika yang digunakan untuk pengiriman paket ketika keluar melewati jaringan lokal serta fungsi Routing untuk mengarahkan pengiriman paket hingga ketujuan melalui berbagai jaringan.

6. *Data Link Layer* (Lapisan Data Link)

Lapisan ini bertanggung jawab untuk perpindahan *frame* dari satu *node* ke node selanjutnya. Tanggung jawab lainnya dari lapisan ini sebagai berikut:

- *Framing* yaitu membagi aliran bit-bit yang diterima dari lapisan jaringan menjadi *frame*
- *Physical Addressing* (Pengalamatan Fisik) yaitu alamat yang digunakan sebagai sumber dan tujuan ketika melakukan pertukaran data pada jaringan lokal. Jika *frame* ditujukan pada perangkat diluar jaringan lokal maka alamat tujuannya adalah berupa alamat perangkat yang menghubungkan jaringan lokal dengan jaringan luar tersebut.
- *Flow Control* yang fungsinya mirip dengan lapisan transport hanya saja ini pengendalian antara node ke node.
- *Error Control* yang berfungsi untuk mekanisme mendeteksi dan mengirimkan *frame* yang rusak, hilang, duplikasi.
- *Access Control* untuk menentukan perangkat mana yang memiliki kendali terhadap koneksi dengan perangkat lainnya pada waktu tertentu.

7. *Physical Layer* (Lapisan Fisik)

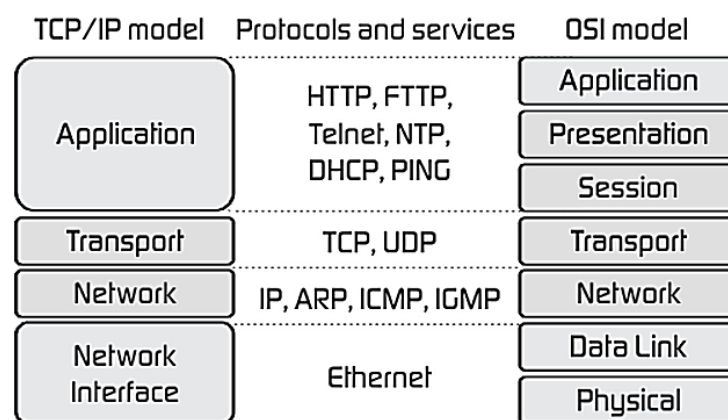
Lapisan ini bertanggung jawab terhadap perpindahan bit dari satu node ke node selanjutnya. Lapisan fisik juga bertanggung jawab dengan hal sebagai berikut :

- Karakteristik fisik dari antarmuka dan media perantara.
- Bit-bit representasi yaitu untuk menentukan jenis pengkodean bagaimana bit 0 dan bit 1 diubah ke bentuk sinyal.
- *Data Rate* yaitu untuk menentukan jumlah bit yang dikirimkan setiap detik.
- Bit Sinkronisasi yaitu untuk menyamakan clock antara pengirim dan penerima.
- Topologi Fisik yaitu untuk menggambarkan secara fisik atau nyata bagaimana perangkat-perangkat terhubung satu sama lain hingga menjadi suatu jaringan.

- Mode Transimisi yaitu untuk menentukan arah transmisi dari perangkat yang sedang melakukan pertukaran data. *Simplex* yaitu hanya satu perangkat yang bisa mengirimkan data dan yang lainnya hanya menerima. *Half-Duplex* yaitu dua perangkat bisa mengirim atau menerima data tetapi tidak dalam waktu yang bersamaan. *Full-Duplex* yaitu setiap perangkat bisa mengirim dan menerima data di waktu yang bersamaan.

2.3.2 Protokol TCP/IP

Protokol TCP/IP lebih dulu dikembangkan dari pada model OSI. Memiliki empat susun lapisan yang mana jumlahnya tidak sesuai dengan model OSI. TCP/IP adalah protokol berhirarki yang dibuat secara bermodul dan masing-masing memiliki fungsi yang spesifik. Jika dibandingkan antara model OSI dan protokol TCP/IP, terlihat pada gambar 2.8 bahwa pada TCP/IP tidak memiliki lapisan Presentasi dan lapisan Sesi seperti model OSI karena fungsi dari kedua lapisan tersebut sudah dijadikan satu pada protokol TCP/IP di lapisan Aplikasi. Terdapat alasan mengapa hal tersebut dilakukan. Pertama, TCP/IP memiliki lebih dari satu lapisan Transport dimana beberapa fungsi lapisan Sesi juga terdapat di lapisan Transport. Alasan kedua, lapisan Aplikasi bukan hanya sekedar perangkat lunak. Di Lapisan ini, aplikasi bisa dikembangkan dengan banyak cara yang mana terkadang bisa memasukkan fungsi dari lapisan Presentasi dan lapisan Sesi. Dibawah ini adalah gambar perbandingan antara model OSI dan protokol TCP/IP. [4]



Gambar 2.8 Perbandingan Model OSI dan Protokol TCP/IP

Berikut penjelasan dari setiap lapisan pada protokol TCP/IP :

1. *Physical Layer* (Lapisan Fisik)

TCP/IP tidak menjelaskan tentang protokol spesifik apapun pada lapisan ini. Pada tingkatan inilah komunikasi antara *node* atau komputer terjadi secara fisik dengan bentuk data berupa bit yang mana aliran bit dikirimkan antara kedua *node* ketika koneksi telah dibangun.

2. *Data Link Layer* (Lapisan Data Link)

Di lapisan ini TCP/IP juga tidak menjelaskan tentang protokol spesifik apapun. Tetapi bentuk data yang digunakan berupa *frame*. *Frame* adalah paket yang mengenkapsulasi data dari lapisan Jaringan dengan tambahan *header* dan *trailer*. Pada dibagian depan *frame* terdapat alamat tujuan dan alamat sumber yang mana alamat tujuan diperlukan untuk menentukan penerima *frame* yang akan dikirimkan sedangkan alamat sumber diperlukan untuk kemungkinan respon atau pengenalan yang dibutuhkan oleh beberapa protokol lainnya.

3. *Network Layer* (Lapisan Jaringan)

Pada lapisan ini terdapat suatu protokol yang digunakan TCP/IP sebagai mekanisme transmisi yang disebut *Internet Protocol* (IP). IP mengirimkan data dalam bentuk paket yang terpisah-pisah, biasa disebut dengan *datagrams*. Terdapat perbedaan antara komunikasi paket di lapisan jaringan dan lapisan fisik atau lapisan data *Link*. Pada lapisan jaringan, komunikasinya bersifat *end-to-end* yaitu antara sumber pengirim dan penerima akhir (tujuan), sedangkan pada lapisan fisik atau lapisan data *Link* hanya bersifat *hop-to-hop* yaitu hanya antara dua *node* yang saling terhubung secara langsung.

4. *Transport Layer* (Lapisan Transport)

Terdapat perbedaan utama antara lapisan jaringan dan lapisan *transport*. Setiap *node* memang memerlukan lapisan Jaringan, tetapi hanya komputer dengan komunikasi *end-to-end* saja yang memerlukan lapisan *transport*. Lapisan Jaringan bertanggung jawab untuk pengiriman masing-masing setiap datagram dari komputer A ke kekomputer B, sedangkan lapisan *transport* bertanggung jawab untuk pengiriman seluruh pesan dari

komputer A ke komputer B yang mana bentuk data itu disebut dengan *Segment*. Di lapisan ini pula terdapat protokol yang biasa digunakan untuk pengiriman data yaitu *User Datagram Protocol*(UDP), *Transmission Control Protocol* (TCP), *Stream Control Transmission Protocol* (SCTP).

5. *Application Layer* (Lapisan Aplikasi).

Lapisan aplikasi pada TCP/IP setara dengan tiga kombinasi lapisan sesi, lapisan presentasi, lapisan aplikasi pada model OSI. Di lapisan Aplikasi, pengguna diperbolehkan mengakses layanan Internet pribadi kita atau Internet global. Banyak protokol yang digunakan pada lapisan untuk menyediakan bermacam layanan seperti *electronic mail*(e-mail), pertukaran data, mengakses situs, dan lain-lain. Bentuk data pada lapisan ini biasa disebut dengan *message*. Komunikasi pada lapisan ini juga sama seperti lapisan *transport* yaitu *end-to-end*. Beberapa protokol yang ada pada lapisan ini sebagai berikut :

- HTTP (*Hyper Text Transfer Protocol*) yang digunakan untuk layanan *web*.
- Telnet yang berfungsi untuk melakukan *remote* pada suatu *host*.
- FTP (*File Transfer Protocol*) yang digunakan untuk keperluan *file sharing*.

2.4 Pengalamatan IPv4 (Internet Protocol version 4)

Alamat IPv4 adalah pengalamatan 32 bit yang secara unik dan universal menjelaskan hubungan antar perangkat (contoh: komputer atau router) ke Internet. [3] IPv4 bersifat unik yang berarti, tidak boleh ada dua atau lebih perangkat disaat yang bersamaan memiliki alamat yang sama. IPv4 bersifat universal yang berarti ketentuan sistem pengalamatan harus diterimaoleh setiap host yang ingin terhubung dengan jaringan. Pada IPv4 menggunakan pengalamatan 32 bit yang berarti memiliki spasi alamat sebanyak 2^{32} atau sekitar 4,294,967,296 (lebih dari 4 milyar) alamat.

2.4.1 Notasi IPv4

Terdapat dua notasi yang umum digunakan untuk memperlihatkan alamat IPv4 yaitu notasi biner dan notasi desimal bertitik.

1. Notasi Biner.

Pada notasi biner, alamat IPv4 di tampilkan dengan 32 bit yang dibagi menjadi empat oktet bagian atau biasa disebut dengan alamat 4 *byte*.

2. Notasi Desimal Bertitik.

Notasi ini dibuat agar IPv4 terlihat lebih pendek dan mudah dibaca. Pada pengalamatan di jaringan, notasi ini biasanya ditulis dalam bentuk desimal dengan pemisah berupa *dot decimal* (notasi desimal bertitik) untuk memisahkan angka disetiap *byte*. Dibawah adalah contoh notasi pengalamatan IPv4 :

$$\begin{array}{ccc}
 & \text{Notasi Biner} & \text{Notasi Desimal Bertitik} \\
 \hline
 \text{011100100 00001010 00000001 00000010} & = & \text{100.10.1.2}
 \end{array}$$

Gambar 2.9 Contoh Notasi Pengalamatan IPv4

2.4.2 Classful Addressing

Pada *classful addressing*, pengalamatan dibagi menjadi 5 kelas yaitu A, B, C, D, dan E. Dibawah ini adalah gambar pembagian alamat ke 5 kelas tersebut.

Tabel 2.1 Skema Pengalamatan Classful untuk IPv4

	First bits of address	Number of bits of network address	Number of bits of host address	Number of network address ranges available	Number of hosts per network address
Class A	0	7	24	126	16.8 million
Class B	10	14	16	16 383	65 536
Class C	110	21	8	2 097 151	256
Class D (Multicast)	1110	28	Non-aggregatable Multicast address	268 million	0
Class E	1111	Experimental use		268 million	0

Untuk penggunaan, pada umumnya hanya kelas A, B, dan C saja yang biasa digunakan, karena kelas D merupakan kelas untuk pengalamatan Multicast serta E merupakan kelas untuk keperluan tertentu seperti riset dan sebagainya. Untuk pengelompokan pengalamatan, selain nomor IP dikenal juga netmask atau

subnetmask. Yang besarnya sama dengan nomor IP yaitu 32 bit. Subnetmask tersebut dikelompokkan kedalam tiga kelas yaitu :

1. Kelas A, adalah semua nomor IP yang mempunyai subnetmask 255.0.0.0
2. Kelas B, adalah semua nomor IP yang mempunyai subnetmask 255.255.0.0
3. Kelas C, adalah semua nomor IP yang mempunyai subnetmask 255.255.255.0

Konsep mask tidak berlaku untuk pengalamatan kelas D dan E.

Tabel 2.2 Default Mask Untuk Classful Addressing

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

2.4.3 Classless Addressing

Pada tahun 1992 menjadi jelas bahwa ruang alamat IPv4 akan habis, dan metode alokasi alamat *classful* tidak bisa digunakan lagi. Hal ini menyebabkandikembangkannya metode baru alokasi umum kisaran alamat IP berdasarkan *Classless Inter-Domain Routing* (CIDR). Skema alamat CIDR (diperkenalkan oleh RFC 1517-9 pada tahun 1993) adalah skema alokasi alamat yang lebih fleksibel yang memungkinkan untuk *host* atau *subnet* (subnetwork) rentang alamat dari setiap panjang (bukan ketat 8-bit, 16-bit atau 24 - bit host dan alamat *subnetwork* panjang dari skema alamat *classful*). Dengan memperkenalkan CIDR, kelas B dan kelas C alamat belum dialokasikan tersisa dapat dibagi antara perusahaan bisnis lebih daripada yang akan mungkin terjadi.

Fitur utama dari CIDR adalah pemisahan dari alamat jaringan dan alamat *subnet* (misalnya, alamat host) dengan cara *subnet mask*. Pada bagian *subnet mask*, semua bit 1 merupakan alamat *subnetwork*.

2.5 Mini PC

Mini PC adalah komputer dengan bentuk yang kecil dan murah serta didesain untuk mampu melakukan pekerjaan yang ringan seperti berselancar di Internet, mengakses aplikasi web, memproses dokumen, dan memutar musik /video. Selain kecil dan murah, mini PC pada umumnya juga memiliki konsumsi daya yang rendah yaitu dibawah 50 Watt bahkan bisa bekerja dengan daya yang lebih rendah lagi yaitu hingga 8 Watt yang mana tergantung dari perangkat keras yang digunakan. Jika dibandingkan komputer desktop, maka akan terlihat jauh sekali perbedaannya mengingat konsumsi daya pada komputer desktop pada umumnya menggunakan daya 100 Watt keatas.

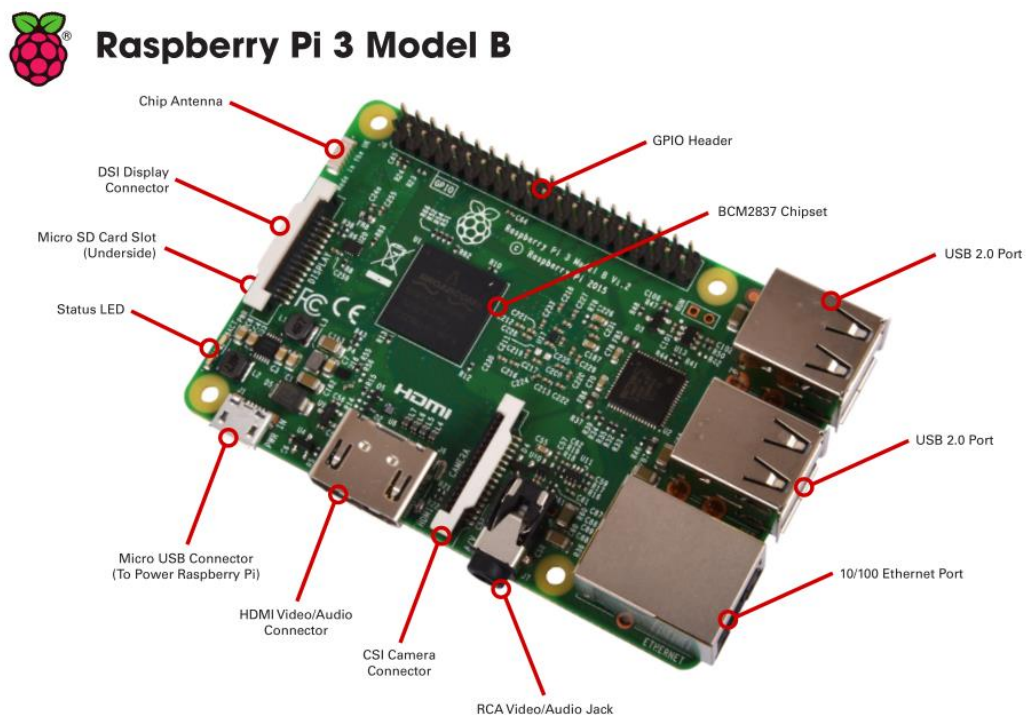
2.5.1 Raspberry Pi

Raspberry Pi adalah sebuah *single board computer* (Mini PC) pertama diluncurkan pada tahun 2012. *Raspberry Pi* dibuat/dikembangkan oleh sebuah perusahaan yang bernama *Raspberry Pi Foundation* dari UK. Perusahaan ini membuat *Raspberry Pi* bertujuan untuk mempromosikan pengajaran ilmu komputer di sekolah dasar. *Raspberry Pi* menggunakan *system on a chip* (SoC) dari Broadcom BCM2837 dan tidak menggunakan hard disk, namun menggunakan SD Card untuk proses booting dan penyimpanan data jangka panjang.

Raspberry Pi berjalan diatas sistem operasi berbasis linux seperti *Raspbian Wheezy*, *Pidora*, *Risc OS*, *Rasp BMC*, *Arch Linux*, dan lain-lain. untuk bahasa pemrogramannya menggunakan bahasa pemrogramana *Phyton*.

Raspberry Pi 3 Model B adalah generasi ketiga dari *Raspberry*. *Single board computer* berukuran sebesar kartu kredit. menggantikan model sebelumnya yaitu *Raspberry Pi Model B+* dan *Raspberry Pi 2 Model B*. Dengan ukuran yang masih sama dengan pendahulunya *Raspberry Pi 3 Model B* memiliki performa 10x lebih cepat di banding dengan *Raspberry Pi* generasi pertama, selain itu pada generasi ketiga di tambahkan *wireless LAN* dan *Bluetooth* yang membuat *Raspberry Pi 3* ini menjadi lebih baik di dibandingkan dengan generasi sebelumnya. Spesifikasi *Raspberry Pi 3 Model B* :

SoC	: Broadcom BCM2837
CPU	: 4× ARM Cortex-A53, 1.2GHz
GPU	: Broadcom VideoCore IV
RAM	: 1GB LPDDR2 (900 MHz)
Networking	: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage	: microSD
GPIO	: 40-pin header, populated
Ports	: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
Dimensions	: 85 x 56 x 17mm
Power	: Micro USB socket 5V1, 2.5A
OS	: Raspbian (Official OS), Ubuntu Mate, Windows 10 IoT Core, RISC OS, PINET, dll.



Gambar 2.10 Raspberry Pi 3 Model B

2.6 Web Server

Web Server merupakan perangkat lunak yang memiliki fungsi menerima permintaan berupa halaman *web* melalui protokol *HTTP* atau *HTTPS* dari suatu klien yang lebih dikenal dengan nama *browser*, kemudian mengirimkan kembali dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen *HTML*. Fungsi utama sebuah *server web* adalah untuk mentransfer berkas atas permintaan pengguna melalui protokol komunikasi yang telah ditentukan. Disebabkan sebuah halaman *web* dapat terdiri atas berkas teks, gambar, video, dan lainnya pemanfaatan *server web* berfungsi pula untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web yang terkait termasuk di dalamnya teks, gambar, video, atau lainnya. Penggunaan biasanya melalui aplikasi pengguna seperti peramban *web*, meminta layanan atas berkas ataupun halaman *web* yang terdapat pada sebuah *server web*, kemudian *server* sebagai manajer layanan tersebut akan merespon balik dengan mengirimkan halaman dan berkas-berkas pendukung yang dibutuhkan, atau menolak permintaan tersebut jika halaman yang diminta tidak tersedia. Saat ini pada umumnya *server web* telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan *server web* menyediakan layanan situs *web* dinamis dengan memanfaatkan pustaka tambahan seperti *PHP*, dan *ASP*.

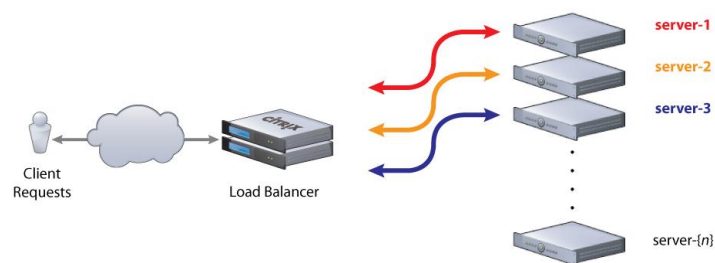
Salah satu contoh dari *Web Server* adalah Apache. Apache (*Apache Web Server – The HTTP Web Server*) merupakan *web server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

2.7 Cluster Server

Cluster server adalah sekumpulan *server* independen yang beroperasi serta bekerja secara erat dan terlihat oleh klien jaringan seolah-olah *server-server* tersebut adalah satu buah unit *server*. Kelebihan dari *cluster server* adalah kecepatan akses data yang lebih baik daripada *server* tunggal sehingga *cluster server* ini mempunyai kemampuan komputasi yang relatif baik. Kemudian kelebihannya lainnya dibanding dengan server biasa adalah pemroses dalam hal

ini prosesor pada *cluster* dapat terus bertambah sesuai dengan jumlah prosesor yang diclusterkan sehingga dapat dipastikan bahwa *server* yang sudah diclusterkan mempunyai kemampuan yang relatif lebih baik dibandingkan dengan *server* biasa. Sebuah *server cluster* terdiri dari seperangkat komputer yang bekerja sama sebagai komputasi yang terintegrasi. Tidak hanya mampu melakukan komputasi yang kompleks, *cluster server* juga menyediakan ketersediaan yang maximal. Selain itu, sistem *cluster server* juga dapat mengatasi kegagalan sistem yang disebabkan oleh faktor yang tidak direncanakan.

2.8 Load Balancing



Gambar 2.11 Load Balancer

Load balancing adalah teknik untuk mendistribusikan beban trafik pada dua atau lebih jalur koneksi secara seimbang, agar trafik dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi. *Load balancing* digunakan pada saat sebuah *server* telah memiliki jumlah *user* yang telah melebihi maksimal kapasitasnya. *Load balancing* juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, link jaringan, CPU, hard drive, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal.

Load balancer (perangkat *load balancing*) menggunakan beberapa peralatan yang sama untuk menjalankan tugas yang sama. Hal ini memungkinkan pekerjaan dilakukan dengan lebih cepat dibandingkan apabila dikerjakan oleh hanya satu peralatan saja dan dapat meringankan beban kerja peralatan, serta mempercepat waktu *respons*. *Load balancer* bertindak sebagai penengah di antara layanan utama dan pengguna, dimana layanan utama merupakan sekumpulan *server*/mesin yang siap melayani banyak pengguna. Disaat *load balancer* menerima permintaan layanan dari *user*, maka permintaan tersebut akan

diteruskan ke *server* utama. Biasanya *load balancer* dengan pintar dapat menentukan *server* mana yang memiliki load yang lebih rendah dan respon yang lebih cepat. Bahkan bisa menghentikan akses ke *server* yang sedang mengalami masalah dan hanya meneruskannya ke *server* yang dapat memberikan layanan. Hal ini salah satu kelebihan yang umumnya dimiliki *load balancer*, sehingga layanan seolah olah tidak ada gangguan di mata pengguna.

2.9 QoS (Quality of Service)

Quality of Service atau kualitas layanan adalah sebuah standar untuk memastikan bahwa suatu performa pada suatu jaringan berjalan dengan baik. Tujuan QoS adalah untuk menghemat penggunaan *bandwidth*, mengurangi kemungkinan *dataloss* dan untuk mengendalikan *latency* serta *jitter*. [5] Pada umumnya ada empat karakteristik dari QoS yaitu:

1. *Reliability* (Kehandalan)

Reliability merupakan kurangnya kehandalan pada proses pengiriman mengakibatkan paket harus dikirim kembali. Tetapi kepekaan setiap program tentu saja tidak sama. Contohnya saja seperti ketika menggunakan layanan streaming kamera lebih membutuhkan kecepatan. Tidak seperti e-mail dan file transfer yang lebih mengutamakan kehandalan.

2. *Delay*

Delay merupakan waktu tunda suatu paket ketika paket sedang dikirimkan dari satu titik ke titik lain. Pada kasus ini, streaming tentu saja membutuhkan *delay* yang minimal. Untuk mengetahui nilai *delay* rata - rata adalah dengan melihat lama waktu yang digunakan dan total paket yang diterima. Baik buruknya suatu delay ditunjukkan pada tabel 2.3. Secara matematis nilai delay dapat dirumuskan pada persamaan 2.1 :

$$Delay = \frac{Total\ waktu}{Total\ paket\ yang\ diterima} \dots (2.1)$$

Tabel 2.3 Rekomendasi ITU-T G.114 untuk delay

Range dalam Millisecond	Penjelasan
0-150	Dapat diterima untuk banyak pengguna aplikasi
150 – 400	Dapat diterima dengan ketentuan bahwa administrator mengetahui waktu transmisi dan dampaknya pada kualitas transmisi pengguna aplikasi.
> 400	Tidak dapat diterima untuk tujuan perencanaan jaringan umum, diakui bahwa dalam beberapa kasus khusus batas ini akan terlampaui

3. *Jitter*

Jitter merupakan suatu kejadian dimana adanya variasi delay ketika paket dikirimkan. Peningkatan trafik secara tiba-tiba dan kecepatan terima atau pengiriman paket dari setiap node merupakan salah satu penyebab terjadinya *jitter*.

4. *Bandwith*

Bandwith merupakan banyaknya ketersediaan sumber daya yang dapat mengalir pada trafik dalam jaringan. Satuan yang digunakan pada *bandwith* adalah bit per second atau disingkat bps.

5. *Throughput*

Merupakan kemampuan suatu jaringan dalam melakukan pengiriman paket data. *Throughput* merupakan jumlah total paket data terkirim yang dibagi dengan waktu pengiriman paket data (detik).

6. *Packet Loss*

Packet Loss merupakan suatu parameter yang menunjukkan jumlah total paket yang hilang. Hal ini dapat terjadi karena adanya collision dan congestion pada jaringan.

2.10 Software uji kinerja web server

Terdapat beberapa software yang dapat digunakan untuk menganalisa performa dari sebuah web server yang dikembangkan secara open source yang dapat digunakan secara gratis, diantaranya: [6]

1. ab (*Apache Benchmarking*)

ab(*Apache HTTP Server Benchmarking tool*) adalah sebuah software yang dapat digunakan untuk mengukur kinerja *Apache HTTP server*. Ini dirancang khusus oleh Apache untuk mengukur berapa banyak permintaan yang mampu dilayani per detik. beberapa contoh perintah nya adalah -n (jumlah permintaan yang dapat dilayani per detik), -c (nomor konkurensi yaitu jumlah permintaan yang dilakukan dalam satu waktu), -t (batas waktu tunggu maksimum atau respons). Salah satu hasil dari output ab yang paling penting adalah Time per request (waktu respons untuk permintaan dalam milidetik). termasuk *Concurrency Level*, waktu yang dibutuhkan dalam pengujian, *Complete requests*, *Failed requests*, *Write errors*, *Total transferred*, *HTML transferred*, *Requests per second*, *Time per request*, *Transfer rate*.

2. httpperf (*http performance measurement tool*)

httpperf (http performance measurement tool): Dalam pengujian sistem web terdiri dari web server, klien, dan jaringan yang menghubungkan klien ke server. beberapa contoh perintah nya adalah --server (nama atau alamat ip web server), --rate (Menentukan berapa banyak koneksi HTTP total yang akan di buat selama pengujian, semakin tinggi angka maka semakin lama waktu pengujian)

3. siege

siege: Software ini dapat memberikan pengguna kemampuan untuk menguji konsumsi *resource* server dalam lingkungan yang realistis. siege dapat membantu *web developers* untuk menguji bagaimana web server bertahan dalam menangani request. siege mendukung otentikasi dasar, cookie, HTTP dan protokol HTTPS. siege sangat berguna untuk menguji *Availability* dan *Concurrency* server. jika *Availability* nya tidak 100%, user harus memberikan perhatian khusus karena itu berarti beberapa

pengguna tidak dapat memiliki akses ke server. *Concurrency* adalah hasil dari waktu pemrosesan setiap transaksi (jumlah klik termasuk semua otentikasi) dibagi dengan waktu yang telah berlalu. *Concurrency* menunjukkan tingkat rata-rata koneksi konkuren — semakin tinggi tingkat *Concurrency*, semakin banyak beban yang dimiliki server. Dengan kata lain, socket dibuka untuk memproses lalu lintas baru, lebih banyak waktu diperlukan untuk server untuk memproses transaksi, dan ada lalu lintas tambahan untuk server untuk memproses secara bersamaan. Oleh karena itu, kinerja server dapat menurun.

Ada tiga item yang dapat digunakan untuk analisis kinerja dari sebuah web server adalah: *Connection rate*, *connection time*, *request rate* dan *reply time* di *httpperf* item-item ini adalah yang paling relevan untuk mengukur kinerja sebuah *web server*. Semakin baik nilai yang dihasilkan maka kinerja server semakin cepat.

2.11 Ansible

Ansible adalah *software* otomatisasi *open source* yang mengotomatisasi penyediaan perangkat lunak, manajemen konfigurasi, dan pemasangan aplikasi. Ansible disertakan sebagai bagian dari distro Fedora Linux, yang dimiliki oleh Red Hat Inc., dan juga tersedia untuk Red Hat Enterprise Linux, CentOS, dan Scientific Linux melalui Paket Ekstra untuk Enterprise Linux (EPEL) dan juga untuk sistem operasi lain. Seperti kebanyakan manajemen konfigurasi perangkat lunak, Ansible memiliki dua tipe *server*: mesin pengontrol dan *node*. Pertama, terdapat satu mesin pengontrol tunggal dimana mesin ini akan melakukan orkestrasi atau mengatur setiap *node* yang ada. Node dikelola oleh mesin pengontrol melalui SSH. Mesin pengontrol memetakan lokasi dari setiap *node* melalui "inventory". Untuk mengatur node, Ansible menyebarkan modul-modul ke node yang ada melalui SSH. Modul-modul ini disimpan sementara di setiap *node* dan berkomunikasi dengan mesin pengontrol melalui protokol JSON. Ketika Ansible tidak sedang mengelola *node*, ia tidak mengkonsumsi sumber daya karena tidak ada daemon atau program yang dijalankan untuk Ansible di latar belakang. Berbeda dengan manajemen konfigurasi perangkat lunak populer - seperti Chef,

Puppet, dan CFEngine - Ansible menggunakan arsitektur tanpa agen. Dengan arsitektur berbasis agen, *node* harus memiliki daemon terpasang secara lokal yang berkomunikasi dengan mesin pengendali. Dengan arsitektur tanpa agen, node tidak diharuskan memasang dan menjalankan daemon untuk terhubung dengan mesin jaringan dengan mencegah *node* untuk terus mengecek apakah mesin pengendali telah siap atau tidak. [7]