BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Pelaksanaan kerja praktek ini meninjau beberapa hasil penelitian sebelumnya sebagai tinjauan pustaka. Penelitian-penelitian tersebut dijadikan sebagai referensi dalam menentukan arah pelaksanaan kerja praktek sehingga kesalahan dalam pengerjaanya dapat diminimalkan.

Penelitian yang berjudul Analisa Perbandingan Manajemen Bandwidth Class Based Queue (CBQ) dan Hierarchical Fair Service Curve (HFSC) di Jaringan TCP/IP. Penelitian ini membandingkan kinerja CBQ dan HFSC serta HTB di Jaringan TCP/IP. Parameter yang diukur adalah throughput, waktu pemrosesan pesan dalam bandwidth manager, jitter dan packet loss. Hasil pengukuran untuk throughput, HTB lebih akurat dibandingkan HFSC dan CBQ. Untuk waktu pemrosesan pesan dalam bandwidth manager, CBQ lebih cepat dibandingkan HFSC kemudian HTB. Untuk jitter, HFSC lebih baik dibandingkan HTB kemudian CBQ. Untuk packet loss, pada CBQ sebesar 2,8%, pada HTB sebesar 0,04%, sedangkan pada HFSC tidak terdapat packet loss (Rumani, dkk, 2011).

Penelitian yang berjudul Analisis Perbandingan HTB (Hierarchical Token Bucket) dan CBQ (Class Based Queue) untuk Mengatur Bandwidth Menggunakan Linux. Penelitian ini membandingkan kinerja HTB dan CBQ menggunakan Linux. Parameter yang diukur adalah throughput, jitter dan packet loss. Hasil pengukuran untuk throughput, HTB mematuhi batas maksimum yang ditentukan, sedangkan CBQ melampaui batas maksimum yang ditentukan. Untuk jitter, CBQ memiliki rata-rata jitter yang lebih baik 6 dibandingkan HTB. Untuk packet loss, CBQ lebih buruk dibandingkan HTB (Pangera, 2013).

Tahap akhir dalam penelitiannya adalah pengujian bandwidth terhadap client yang terkoneksi. Pengujian ini dilakukan untuk mengetahui apakah sistem jaringan berjalan dengan baik atau tidak. Aplikasi router menggunakan PfSense dapat melakukan manajemen bandwidth sesuai dengan kebutuhan pengguna, sehingga pengguna dapat menggunakan internet dengan lancar dan nyaman sesuai dengan ketentuan yang telah ditetapkan.

2.2 Landasan Teori

2.2.1 Pengertian Jaringan Komputer

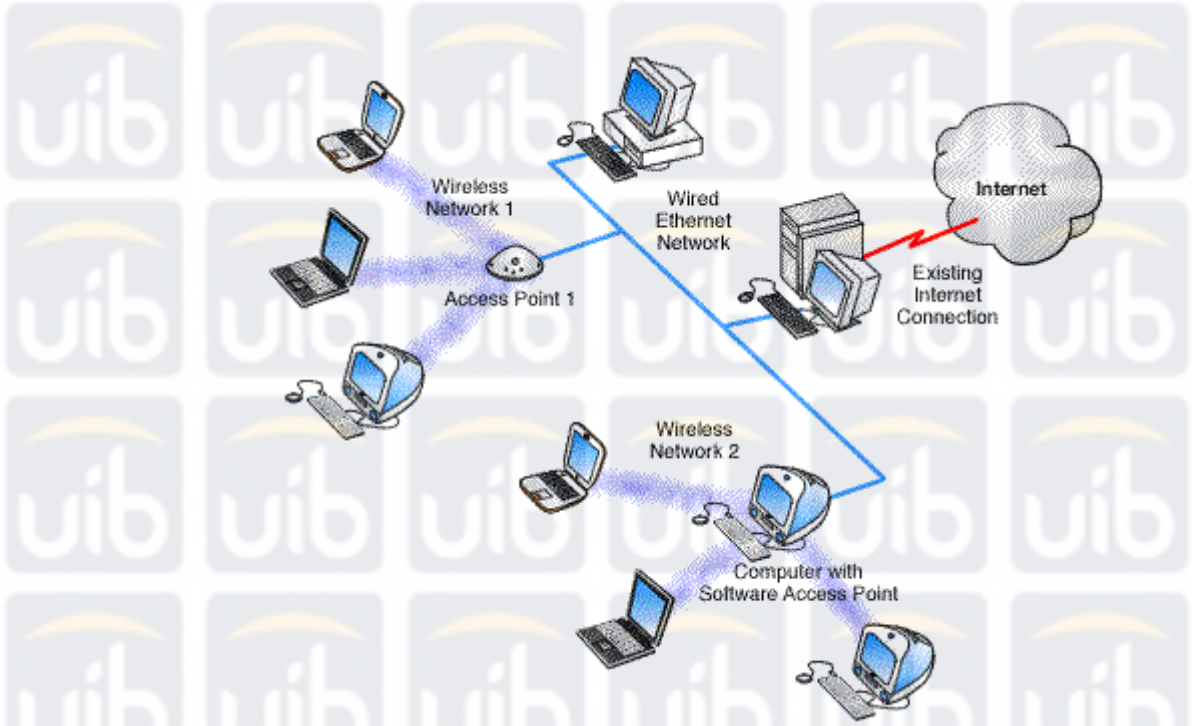
Jaringan komputer merupakan sekelompok komputer otonom yang saling dihubungkan satu sama lainnya, menggunakan suatu media dan *protocol* komunikasi tertentu, sehingga dapat saling berbagi data dan informasi. Jaringan computer memungkinkan terjadinya komunikasi yang lebih efisien antar pemakai (*mail dan teleconference*) (Tommy Pratama, 2015).

Jaringan komputer adalah sekelompok komputer otonom yang saling menggunakan *protocol* komunikasi melalui media komunikasi sehingga dapat berbagi data, informasi, program aplikasi dan perangkat keras seperti *printer*, *scanner*, CD-*drive* maupun *harddisk* serta memungkinkan komunikasi secara elektronik (Yulianti, 2015).

Sedangkan (M.Azhar Irwansyah, 2015) pada aplikasi *home user*, memungkinkan komunikasi antar pengguna lebih efisien (*chat*), interaktif *entertainment* lebih multimedia (games, video,dan lain- lain).

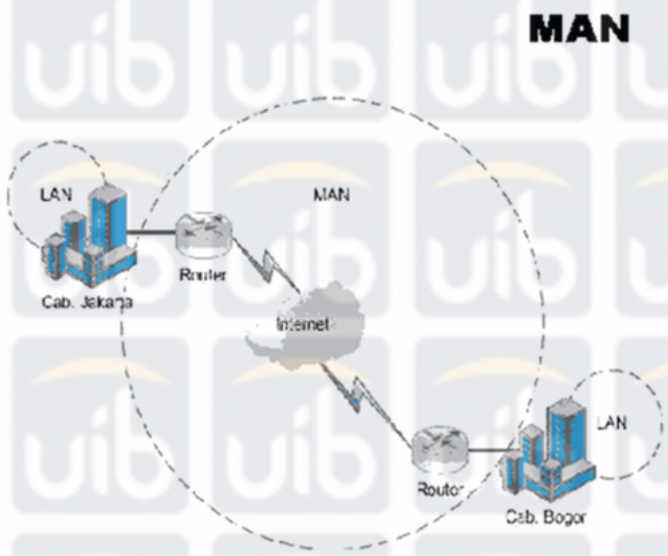
Klasifikasi jaringan komputer :

1. LAN (*Local Area Network*) : Jaringan komputer yang saling terhubung ke suatu komputer *server* dengan menggunakan suatu topologi tertentu, biasanya digunakan dalam kawasan satu gedung atau kawasan yang jaraknya tidak lebih dari 1 km.



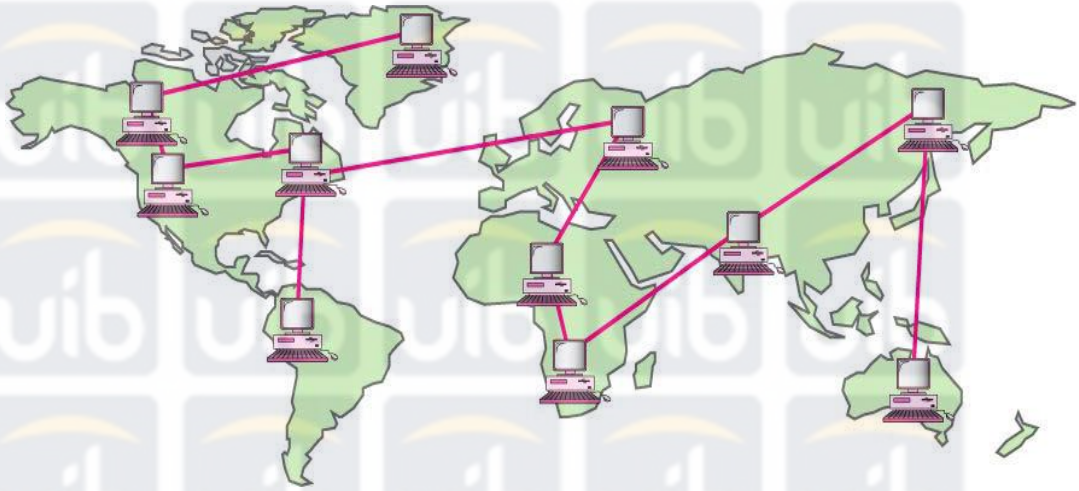
Gambar 2.1 Jaringan LAN

2. MAN (*Metropolitan Area Network*) : Jaringan komputer yang saling terkoneksi dalam suatu kawasan kota yang jaraknya bisa lebih dari 1 km. Pilihan untuk membangun jaringan komputer antar kantor dalam satu kota, kampus dalam satu kota.



Gambar 2.2 Jaringan MAN

3. WAN (*Wide Area Network*) : Jaringan komputer yang menghubungkan banyak LAN ke dalam suatu jaringan terpadu, antara satu jaringan dengan jaringan lain berjarak ribuan kilometer atau terpisahkan letak geografi dengan menggunakan metode komunikasi tertentu.



Gambar 2.3 Jaringan WAN

Secara tahapan ada beberapa garis besar dalam membangun jaringan LAN, diantaranya :

1. Menentukan teknologi tipe jaringannya (*Ethernet, Fast Ethernet, Token Ring*, FDDI).
2. Memilih model perkabelan (*Fiber*, UTP, *Coaxial*).
3. Menentukan bentuk topologi jaringan (*Bus*, *Ring*, dan *Star*).
4. Menentukan teknologi *Client/Server* atau *Peer to Peer*.
5. Memilih Sistem Operasi *Server* (*Windows, Linux,* atau yang lainnya).

2.2.2 Topologi Jaringan

Pengertian topologi jaringan komputer adalah suatu cara atau konsep untuk menghubungkan beberapa atau banyak komputer sekaligus menjadi suatu jaringan yang saling terkoneksi. Dan setiap macam topologi jaringan komputer akan berbeda dari segi kecepatan pengiriman data, biaya pembuatan, serta kemudahan dalam proses *maintenance* nya. Dan juga setiap jenis topologi jaringan komputer memiliki kelebihan serta kekurangannya masing-masing. Ada banyak macam topologi seperti topologi *ring, star, bus, mesh, dan tree* (Ryan Oktavian, 2010).

2.2.2.1 Topologi Star

Topologi ini membentuk seperti bintang karena semua komputer dihubungkan ke sebuah hub atau switch dengan kabel UTP, sehingga hub/switch lah pusat dari jaringan dan bertugas untuk mengontrol lalu lintas data, jadi jika komputer 1 ingin mengirim data ke komputer 4, data akan dikirim ke switch dan langsung di kirimkan ke komputer tujuan tanpa melewati komputer lain. Topologi jaringan komputer inilah yang paling banyak digunakan sekarang karena kelebihannya lebih banyak.

Gambar 2.4 Topologi Star

- Kelebihan topologi ini adalah sangat mudah mendeteksi komputer mana yang mengalami gangguan, mudah untuk melakukan penambahan atau pengurangan komputer tanpa mengganggu yang lain, serta tingkat keamanan sebuah data lebih tinggi, .
- Kekurangannya topologi jaringan komputer ini adalah, memerlukan biaya yang tinggi untuk pemasangan, karena membutuhkan kabel yang banyak serta switch/hub, dan kestabilan jaringan sangat tergantung pada terminal pusat, sehingga jika switch/hub mengalami gangguan, maka seluruh jaringan akan terganggu.

2.2.3 Router

Router adalah perangkat yang akan melewatkan paket IP dari suatu jaringan ke jaringan yang lain, menggunakan metode *addressing* dan *protocol* tertentu untuk melewatkan paket data tersebut (Simon Siregar, 2011).

Router memiliki kemampuan melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur diantara keduanya. *Router-router* yang saling terhubung dalam jaringan internet turut serta dalam sebuah algoritma *routing* terdistribusi untuk menentukan jalur terbaik yang dilalui paket IP dari *system* ke *system* lain. Proses *routing* dilakukan secara *hop by hop*. IP tidak mengetahui jalur keseluruhan menuju tujuan setiap paket. IP *routing*

hanya menyediakan IP *address* dari *router* berikutnya yang menurutnya lebih dekat ke *host* tujuan. Berikut merupakan fungsi *router* secara umum :

1. Membaca alamat logika / IP *address source and destination* untuk menentukan *routing* dari suatu LAN ke LAN lainnya.
2. Menyimpan *routing table* untuk menentukan rute terbaik antara LAN ke WAN.
3. Perangkat di layer 3 OSI *Layer*.
4. Bisa berupa "box" atau sebuah OS yang menjalankan sebuah *daemon routing.*
5. *Interfaces Ethernet,* Serial, ISDN BRI.

2.2.4 *Router dan Gateway*

Untuk menghubungkan *user* dengan *server* agar *user* dapat terkoneksi dalam suatu jaringan, maka server dibuat sebagai pintu gerbang (*router/gateway*).

Router dan *gateway* sendiri sebenarnya secara teori mempunyai filosopi arti yang berbeda. *Gateway* sebenarnya mengacu pada alat yang difungsikan untuk menjembatani dua jaringan yang mempunyai topologi yang berbeda, *subnet* yang berbeda, dan lain sebagainya. Sedangkan *router* untuk mengatur pengalamatan paket-paket data dalam jaringan yang berbeda sehingga komunikasi dapat terlaksana (Simon Siregar, 2011).

Akan tetapi dalam kenyataannya sehari-hari, *router* dan *gateway* seringkali ditangani oleh sebuah alat saja. Hal inilah yang menyebabkan *router* selalu diidentikan sebagai *gateway*, begitu pula sebaliknya.

Router memiliki kemampuan untuk melewatkan paket IP dari satu jaringan ke jaringan lain yang mungkin memiliki banyak jalur di antara keduanya.

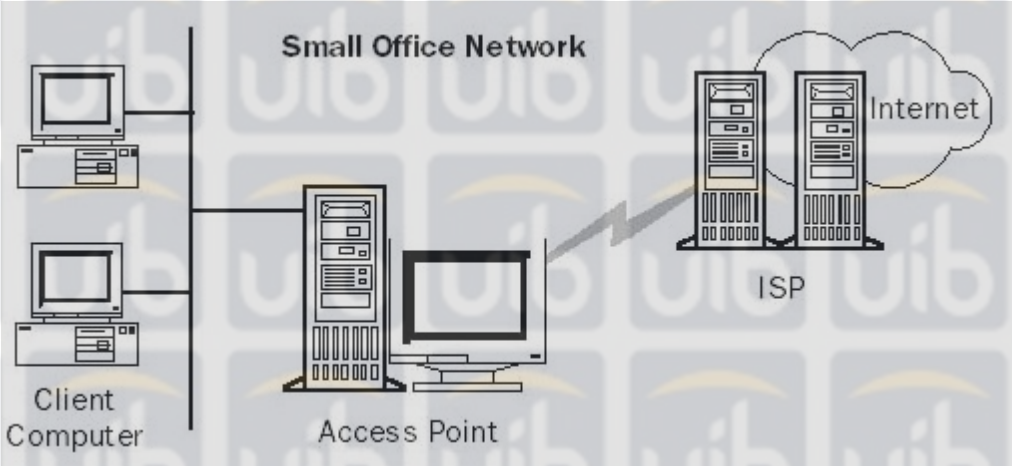
Router-router yang terhubung di Internet memiliki algoritma *routing* terdistribusi yang digunakan untuk memilih jalur terbaik yang dilalui paket IP dari satu jaringan ke jaringan lain.

Router umumnya digunakan untuk menghubungkan sejumlah LAN, sekaligus mengisolasikan trafik data antara LAN satu dengan lainnya. Jika dua atau lebih LAN terhubung dengan satu *router*, maka setiap LAN akan dianggap memiliki *subnetwork* yang berbeda.

2.2.5 *Network Address Translator* (NAT)

Ada dua tipe alamat IP: umum dan pribadi. Alamat umum diberikan kepada kita oleh *Internet Service Provider* (ISP) yang kita pakai untuk berhubungan ke *internet*. Bagi *host* di dalam organisasi yang tidak memerlukan akses langsung ke *internet*, alamat IP yang tidak menduplikasi alamat umum yang sudah diberikan memang dibutuhkan. Untuk memecahkan persoalan alamat ini, para desainer *internet* mencadangkan suatu bagian dari ruang alamat IP dan menamai ruang ini sebagai ruang alamat pribadi. Suatu alamat IP pada ruang alamat pribadi tidak pernah diberikan sebagai alamat umum. Alamat IP di dalam ruang alamat pribadi dikenal sebagai alamat pribadi. Dengan memakai alamat IP pribadi, kita dapat memberikan proteksi dari para *hacker* jaringan (Tedy Gunawan, 2014).

Karena alamat IP pada ruang alamat pribadi tidak akan pernah diberikan oleh *Internet Network Information Center* (InterNIC) sebagai alamat umum, maka *route* di dalam *internet router* untuk alamat pribadi takkan pernah ada. Alamat pribadi tidak dapat dijangkau di dalam *internet*. Oleh karena itu, saat memakai alamat IP pribadi, kita membutuhkan beberapa tipe *proxy* atau *server* untuk mengonversi sejumlah alamat IP pribadi pada jaringan lokal kita menjadi alamat IP umum yang dapat di-*routed*. Pilihan lain adalah menerjemahkan alamat pribadi menjadi alamat umum yang *valid* dengan *Network Address Translator* (NAT) sebelum dikirimkan di *internet*. Dukungan bagi NAT untuk menerjemahkan alamat umum dan alamat pribadi memungkinkan terjadinya koneksi jaringan-jaringan kantor-rumah atau kantor yang kecil ke *internet* seperti ditampilkan gambar 2.5 berikut ini.



Gambar 2.5 Menghubungkan sebuah jaringan kantor yang kecil ke internet

Sebuah NAT menyembunyikan alamat-alamat IP yang dikelola secara internal dari jaringan-jaringan eksternal dengan menerjemahkan alamat internal pribadi menjadi alamat eksternal umum. Hal ini mengurangi biaya registrasi alamat IP dengan cara membiarkan para pelanggan memakai alamat IP yang tidak terdaftar secara internal melalui suatu terjemahan ke sejumlah kecil alamat IP yang terdaftar secara eksternal. Hal ini juga menyembunyikan struktur jaringan internal, mengurangi resiko penolakan serangan layanan terhadap sistem internal.

2.2.6 OSI (Open System Interconnection) Layer

Menurut Beasley (2012 p86), model referensi jaringan terbuka OSI atau *reference model for open systems interconnect* dikembangkan oleh *Internastional Standard Organization* (ISO) pada tahun 1984 untuk memungkinkan berbagai jenis ajringan dapat terhubung. OSI model terdiri dari 7 lapisan. Lapisan-lapisan ini menggambarkan fungsi jaringan antarmuka, jaringan fisik ke antarmukan perangkat lunak aplikasi.

No.	LAPISAN	TCP/IP	NAMA PROTOCOL
7.	APLIKASI	APLIKASI	• DHCP (Dynamic Host Configuration Protocol) • DNS (Domain Name Server) • FTP (File transfer Protocol) • HTTP (hyper text transfer protocol) • MIME (Multiperpose internet mail extension) • NNTP (network news transfer prtocol) • POP (post office protocol) • SMB (server message block)
6.	PRESENTASI	APLIKASI	• SMTP (simple mail transfer protocol) • SNMP (simple network management) • TELNET • TFTP (trivial FTP)
5.	SESSI	APLIKASI	• NETBIOS (network basic input output system) • RPC (remote procedure call • Socket
4.	TRANSPORT	TRANSPORT	• TCP (transmission control protocol) • UDP (user datagram protocol)
3.	NETWORK	INTERNET	• IP (internet protocol) • RIP (routing information protocol) • ARP (addres resolution protocol) • RARP (revers addres resolution protocol)
2.	DATA LINK LLC	NETWORK INTERFACE	• PPP (point to point protocol)
	DATA LINK MAC		• SLIP (serial line internet protocol)
1.	FISIK		ETHERNET, FDDI, ISDN, ATM

Gambar 2.6 OSI Layer

model OSI *Layer* tersebut adalah:

1. *Physical layer*

Merupakan lapisan paling bawah dari OSI *Layer*, memiliki fungsi mendefinisikan media transmisi jaringan fisik yang digunakan mentransmisi sinyal yang digunakan untuk mengirimkan data melalui kabel dari satu node ke node yang lain. Lapisan ini menggunakan *network interface card* (NIC) yang merupakan alat lapisan pertama.

2. *Data Link layer*

Menanganani pemulihan kesalahan, merupakan media akses kontrol dimana *mac address* didefinisikan serta berfungsi menentukan bagian

mnetransmisikan bit-bit data melalui jaringan. Protokol *data link* mengidentifikasikan besarnya tipe paket data yang akan dikirim.

3. *Network layer*

Berfungsi menerima pesan keluar dan menggabungkan pesan atau segmen menjadi paket, menambahkan *header* yang berisi informasi *routing*. Bertindak sebagai *controller* jaringan.

4. *Transport layer*

Berkaitan dengan integritas pesan antara sumber dan tujuan serta berfungsi meemcah data ke dalam paket-paket data yang lebih kecil serta memberikan nomor urut pada paket-paket tersebut sehingga pengiriman data akan efisien dan tidak ada data yang hilang. Lapisan ini akan mengakui penerimaan paket dan pengiriman ulang untuk paker yang tidak sampai atau hilang.

5. *Session layer*

Menyediakan fungsi kontrol untuk mendefinisikan bagaimana koneksi dapat dibuat, diperlihara, atau dihancurkan. Selain itu, dilevel ini juga dilakukan resolusi nama.

6. Presentation layer

Lapisan ini bertangguna jawab untuk kompresi data serta berfungsi untuk mentransmisikan data untuk aplikasi. Format pengiriman yang digunakan umumnya adalah ASCII (*American Standard Code for Information Interchange*).

7. Aplication layer

Merupakan lapisan paling atas dari OSI *layer* yang berfungsi sebagai antarmuka dengan aplikasi menyangkut pada fungsionalitas jaringan, mengatur bagaimana aplikasi dapat mengakses jaringan dan kemudian membuat pesan-pesan yang menjadi notifikasi terjadinya suatu kesalahan.

2.2.7 *TCP* /IP

Gabungan dari protokol *TCP* (*Transmission Control Protocol*) dan IP (*Internet Protocol*) yang mengatur komunikasi data dari satu komputer ke komputer lain di dalam jaringan internet dan akan memastikan pengiriman data sampai ke alamat yang dituju. Protokol ini tidak dapat berdiri sendiri, karena protokol ini berupa kumpulan protokol (*protocol s*uite). Protokol ini mampu bekerja dan diimplementasikan pada lintas perangkat lunak (*software*) di berbagai sistem operasi.Istilah yang diberikan kepada perangkat lunak ini adalah *TCP* /IP stack.

Protokol *TCP* /IP dikembangkan di akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). *TCP* /IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme *transport* jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat routable yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti

Microsoft Windows dan keluarga *UNIX*) untuk membentuk jaringan yang heterogen.

Pengembangan pada protokol *TCP* /IP dilakukan oleh beberapa badan, seperti *Internet Society* (ISOC), *Internet Architecture Board* (IAB), dan *Internet Engineering Task Force* (IETF). Macam-macam protokol yang berjalan di atas *TCP* /IP, skema pengalamatan, dan konsep *TCP* /IP didefinisikan dalam dokumen yang disebut sebagai *Request for Comments* (RFC) yang dikeluarkan oleh IETF.

TCP /IP merupakan protokol jaringan komputer terbuka dan bisa terhubung dengan berbagai jenis perangkat keras dan lunak. *TCP* terdiri dari beberapa *layer* atau lapisan yang memiliki fungsi tertentu dalam komunikasi data. Setiap fungsi dari *layer* selain dapat bekerjasama dengan *layer* pada tingkat lebih rendah atau lebih tinggi, juga bisa berkomunikasi dengan *layer* sejenis pada *remote host* (*peering*). IP adalah jantung *TCP* /IP yang memiliki peran sebagai pembawa data yang independen. Semua dokumen *TCP* /IP dalam bentuk *public document* IEN dan RFC. IP dibagi atas kelas *network* A,B,C,D, dan E.

IP ditulis dalayang membentuk bilangan desimal dari 0 sampai 255. Data yang mengalir antar *layer* atau antar *host* dienkapsulasi dan diberi *header* agar tiap *layer* bisa memprosesnya. Sebuah *host* tidak tahu alamat IP *gateway* di *network* tetapi data mengalir ke *host* tujuan di *network* lain melalui *gateway network* setelah diberi penentuan *routing* alamat IP (Indriyawati, 2010).

2.2.7.1 IP address

IP *Address* merupakan alamat logika yang diberikan ke semua perangkat jaringan yang menggunakan protokol *TCP* /IP. IP *address* memungkinkan *host* pada jaringan yang berbeda maupun pada jaringan yang sama untuk bisa saling berkomunikasi walaupun dalam *platform* yang berbeda (Mubarak, 2014). IP *address* merupakan bilangan biner 32 bit yang terbagi menjadi 4 kelompok sehingga masing-masing kelompok terdiri dari bilangan biner 8 bit, yang merupakan implementasi IPv4.

Kelas-kelas alamat IP *address* terdiri dari:

1. Kelas A

Memiliki struktur 0nnnnnn xxxxxxxx xxxxxxxx xxxxxxxx dimulai dari 0 sampai 127, memiliki 16.777.216 alamat.

2. Kelas B

Memiliki struktur 10nnnnnn nnnnnnnn xxxxxxxx xxxxxxxx dimulai dari 128 sampai 191, memiliki 65.536 alamat.

3. Kelas C

Memiliki struktur 110nnnnn nnnnnnnn nnnnnnnn xxxxxxxx dimulai dari 192 sampai 223, memiliki nilai 256 alamat.

4. Kelas D

Dimulai dari 224 sampai 239, digunakan untuk pengalamatan *multicast.*

5. Kelas E

Dimulai dari 240 sampai 254, digunakan untuk keperluan eksperimen IP *address* terbagi menjadi 2 bagian yaitu bagian *network* (*net* ID) dan bagian *host* (*host* ID):

1. *Network* ID

Network ID digunakan untuk mengenali suatu *network* suatu *network* pada jaringan *internet*. Tujuannya adalah untuk menyederhanakan informasi *routing* pada *internet*. Sebagai contoh; router hanya akan melihat *network address* (192.168) untuk menentukan ke router mana *datagram* tersebut harus dikirimkan. Analogi seperti ini mirip dengan proses pengantaran surat, petugas kantor pos cukup melihat kota tujuan pada alamat surat untuk menentukan jalur mana yang harus ditempuh surat tersebut (Saraswati, 2011).

2. *Host* ID

Host ID digunakan untuk mengirim/menerima informasi yang harus diketahui oleh seluruh *host* yang ada pada suatu *network*. Seperti diketahui, setiap *datagram* IP memiliki *header* alamat tujuan berupa IP *address* dari *host* yang akan dituju oleh *datagram* tersebut. Dengan adanya alamat ini, maka hanya *host* tujuan saja yang memproses *datagram* tersebut (Saraswati, 2011).

2.2.7.2 Jenis-jenis IP address

1. *IP address private*

IP address private merupakan alamat-alamat *IP* yang disediakan untuk digunakan melakukan komunikasi pada jaringan yang tidak terhubung langsung dengan *internet*. *IP*

address private hanya dapat dipakai untuk komunikasi pada jaringan *intranet* dan tidak dapat digunakan pada jaringan *internet* (Budi, 2011).

2. *IP address public*

IP address public merupakan alamat-alamat *IP* yang disediakan untuk digunakan pada jaringan *internet*. *IP public* dapat diperoleh melalui ISP (*Internet Service Provider*) atau penyedia layanan *internet*, alamat *IP* ini telah ditetapkan oleh InterNIC dan berisi beberapa buah *network* ID yang tidak mungkin ada yang sama (Budi, 2011).

2.2.8 PfSense

PfSense adalah FreeBSD berbasis sistem operasi, diturunkan dari m0n0wall, OS yang menggunakan penyaring paket OpenBSD pf. pfSense dirancang untuk digunakan sebagai firewall dan router. Selain menjadi kuat, fleksibel firewall dan routing platform, ini meliputi daftar panjang fitur terkait dan sistem paket yang memungkinkan upgrade lebih lanjut tanpa menambah gembung dan potensi kerentanan keamanan ke basis distribusi (Agus Sarifin, 2012).

pfSense adalah proyek yang populer dengan lebih dari 1 juta download sejak awal, dan terbukti dalam instalasi yang tak terhitung mulai dari jaringan rumah kecil melindungi PC dan Xbox untuk perusahaan besar, universitas dan organisasi-organisasi lain melindungi ribuan perangkat jaringan.

Proyek ini dimulai pada tahun 2004 sebagai pencabangan dari proyek monowall, tetapi fokus terhadap instalasi PC lengkap daripada perangkat keras yang tertanam fokus m0n0wall. pfSense juga menawarkan gambar tertanam untuk instalasi berbasis Compact Flash, namun itu bukan fokus utamanya.

Pfsense merupakan distro linux turunan free bsd, "akan tetapi disesuaikan untuk digunakan sebagai firewall dan router. Selain menjadi, platform yang kuat fleksibel firewall dan routing, itu termasuk daftar panjang fitur terkait dan sistem paket yang memungkinkan upgrade lebih lanjut tanpa menambahkan dan kerentanan keamanan potensial untuk distribusi dasar. pfSense adalah proyek populer dengan lebih dari 1 juta download sejak awal, dan terbukti dalam instalasi yang tak terhitung jumlahnya mulai dari jaringan rumah kecil melindungi PC dan Xbox untuk perusahaan besar, universitas dan organisasi lainnya melindungi ribuan perangkat jaringan."

Dengan tampilan yang sederhan dengan web gui administrator memudahkan kita mengoprasikan pfsense, meskipun kita yang baru belajar routing dan firewall pada jaringan local ataupun internet. dan di ingat pfsense adalah opensource alias GPL GNU, sebuah software yang layak digunakan sebagai alternatif router, firewall, load balancing, ataupun web proxy dan masih banyak lagi fitur yang diberikan.

2.2.9 Manajemen Bandwidth

Management bandwidth Management bandwidth adalah suatu cara yang dapat digunakan untuk management dan mengoptimalkan berbagai jenis jaringan dengan menerapan layanan Quality of service (QoS) untuk menetapkan tipe-tipe lalulintas jaringan. Sedangkan QoS adalah kemampuan untuk menggambarkan suatu tingkatan pencapaian didalam suatu sistem komunikasi data. Maksud dari manajemen bandwidth ini adalah bagaimana kita menerapkan pengalokasian atau pengaturan bandwidth dengan menggunakan sebuah PC Router PfSense. Manajemen bandwidth memberikan kemampuan untuk mengatur Bandwidth jaringan dan memberikan level layanan sesuai dengan kebutuhan dan prioritas sesuai dengan permintaan pelanggan (Bhekti Ratna Timur Astuti, 2012)Selain itu juga diperoleh keuntungan sebagai berikut :

a. Semua komputer dapat menggunakan internet dengan lancar dan stabil walaupun semua unit komputer menggunakan internet dalam waktu yang bersamaan.

b. Semua bagian unit komputer mendapatkan *bandwidth* sesuai dengan kebutuhan koneksi internet.

c. Membantu admin dalam mengontrol *bandwidth*.

2.2.9.1 *Throughput*

Throughput adalah kecepatan rata-rata data yang diterima oleh suatu suatu node dalam selang waktu pengamatan tertentu. *Throughput* merupakan bandwidth aktual saat itu juga dimana kita sedang melakukan koneksi. Satuan yang dimilikinya sama dengan *bandwidth* yaitu *Bit per second* (*bps)* (Setiawan, 2012).

Rumus untuk menghitung nilai *throughput* adalah sebagai berikut:

$$\textit{Throughput} = \frac{\text{Jumlah data yang dikirim}}{\text{Waktu pengiriman data}}$$

2.2.9.2 *Delay*

Delay adalah waktu tunda saat paket yang diakibatkan oleh proses transmisi dari satu titik menuju titik lain yang menjadi tujuannya. *Delay* diperoleh dari selisih waktu kirim antara satu paket TCP dengan paket lainnya yang direpresentasikan dalam satuan *seconds* (Setiawan, 2012).

Rumus untuk menghitung nilai *delay* adalah :

$$\text{Rata-rata } \textit{delay} = \frac{\text{Total } \textit{delay}}{\text{Total paket yang diterima}}$$

2.2.9.3 *Jitter*

Jitter merupakan variasi *delay* antar paket yang terjadi pada jaringan IP. Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya*congestion* dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai *QoS* akan semakin turun. Untuk mendapatkan nilai *QoS* jaringan yang baik, nilai *jitter* harus dijaga seminimum mungkin (Setiawan, 2012).

Rumus untuk menghitung nilai *jitter* adalah :

$$\text{Jitter} = \frac{\text{Total variasi delay}}{\text{Total paket yang diterima}}$$

2.2.10 Bandwidth

Bandwidth adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari suatu tempat ke tempat lain dalam suatu waktu tertentu. *Bandwidth* dapat dipakaikan untuk mengukur baik aliran data analog mau pun aliran data digital. Sekarang telah menjadi umum jika kata *bandwidth* lebih banyak dipakaikan untuk mengukur aliran data digital (Wahyu Patrya, 2013).

Satuan yang dipakai untuk *bandwidth* adalah *bits per second* atau sering disingkat sebagai bps. Seperti kita tahu bahwa bit atau *binary digit* adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit(angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melalui suatu media. Konversi satuan bandwidth sebagai berikut:

- 1 byte = 8 **bit**
- 1 kilobyte = 1.024 byte
- 1 megabyte = 1.024 kilobyte
- 1 gigabyte = 1.024 megabyte
- 1 terabyte = 1.024 gigabyte
- 1 exabyte = 1.024 terabyte

Sehingga 1 kilo byte = 1024 x 8 bit = 8192 bit, dan seterusnya.

Bandwidth adalah konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara mengirimkan informasi mau pun media apa yang dipakai dalam penghantaran informasi. Hal ini karena adanya hukum fisika maupun batasan teknologi. Ini akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, mau pun perlakuan khusus terhadap media yang dipakai (J. Triyono, 2011).

Sedangkan batasan terhadap perlakuan atau cara pengiriman data misalnya dengan pengiriman secara paralel (*synchronous*), serial (*asynchronous*), perlakuan terhadap media yang spesifik seperti media yang tidak boleh ditekuk (serat optis), pengirim dan penerima harus berhadapan langsung (*line of sight*), kompresi data yang dikirim, dan lain-lain.

Jika dilihat dari sudut pandang proses pengriman data, *bandwidth* dapat dibagi menjadi 2 kategori, yaitu *download* dan *upload*.

2.2.10.1 *Download* (Unduh)

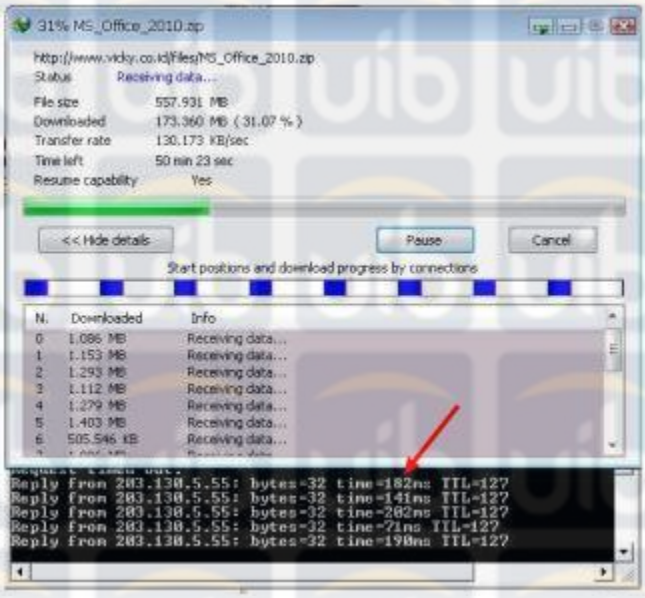
Download adalah proses menerima data (umumnya berbentuk berkas) dari sebuah sistem seperti *web server*, FTP *server*, *mail server* atau sistem serupa lainnya. *Download* juga merupakan kegiatan dimana seseorang dapat memperoleh *file-file* tertentu yang ada di internet. File yang dapat diunduh bermacam-macam, ada yang berupa *file video*, mp3, *document*, dan lain-lain (J. Triyono, 2011).

2.2.10.2 *Upload* (Unggah)

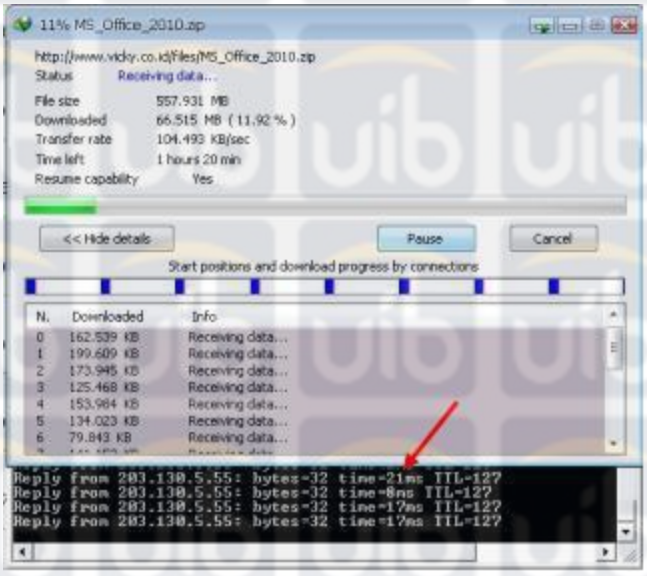
Upload adalah proses mengirim data (umumnya berbentuk berkas) dari komputer pribadi ke suatu sistem seperti *web server*, FTP *server* atau sistem serupa lainnya yang kemudian akan dipublikasikan di internet baik secara pribadi atau umum (dapat di nikmati oleh semua pengguna internet)(J. Triyono, 2011).

2.2.11 *Traffic Shaping*

Traffic shaping adalah suatu uji coba untuk mengendalikan *traffic* jaringan komputer untuk mengoptimalkan kemampuan, *latency* yang rendah dan *bandwidth*. *Traffic shaping* digunakan untuk mengatur *traffic* yang keluar ke *interface* agar alirannya sesuai dengan kecepatan dari target *interface* dan menjamin bahwa *traffic* memberitahukan ulang kebijakan yang dibuat untuknya (M. Freed, 2014).Berikut gambar penjelasan tentang *traffic shaping*:



Gambar 2.7 Kondisi *ping request* sebelum diterapkan *Traffic Shaping*



Gambar 2.8 Kondisi *ping request* setelah diterapkan *Traffic Shaping*

2.2.11.1 *Priority Queueing* (PRIQ)

Priority Queueing (PRIQ) adalah metode tersimple dari *traffic shaping* dan terkadang yang paling efektif. Metode ini melakukan prioritas dari *traffic* saja tanpa

memperhatikan *bandwidth*. Sebuah hirarki daftar tingkatan prioritas dibuat, maka semua paket dengan prioritas lebih tinggi akan selalu diproses terdahulu (M. Freed, 2014).

2.2.11.2 *Hierarchical Fair Service Curve* (HFSC)

Hierarchical Fair Service Curve (HFSC) adalah metode yang paling kompleks dan ribet dari *traffic shaping* jika belum pernah mencobanya. Metode ini memiliki hirarki dari antrian dan mampu menjamin *real-time traffic* dalam arti dapat menjamin kestabilan *traffic* ketika digunakan pada saat bersamaan, sehingga prioritas bawah pun tetap dapat mendapat *traffic*. Hal ini dapat sangat efektif untuk VOIP pada *link* yang mendegradasi cepat, seperti 3G/4G, tetapi dapat menjadi kompleks untuk mengkonfigurasi dan tweak untuk operasi yang tepat (M. Freed, 2014).