BAB II

LANDASAN TEORI

2.1 Jaringan Komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-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.

Dalam komunikasi antar sistem komputer, diperlukan suatu bentuk standar dari komunikasi pada seluruh kerja jaringan komputer dan komunikasi antar komputer. Untuk itu dibuat suatu pembakuan dalam hal komunikasi data antar sistem komputer ini. *ISO* (*The Internasional Standar Organization*) sebagai organisasi standarisasi internasional memberikan suatu model standarisasi bagi seluruh fungsi kerja dan komunikasi antar sistem komputer yatiu model *OSI* (*Open System Interconnection*). (Melwin Syafrizal,2008)

Tipe-tipe jaringan komputer berdasarkan sistem koneksi antar *node* (komputer) menjadi dua, yakni:

a. *Jaringan Peer to Peer*

Peer to peer network adalah jaringan komputer yang terdiri dari beberapa komputer (biasanya tidak lebih dari sepuluh komputer dengan satu sampai dua printer). Untuk penggunaan khusus, seperti laboratorium komputer, riset, dan beberapa hal lain, maka model *peer to peer* ini bisa saja dikembangkan untuk koneksi lebih dari sepuluh hingga seratus komputer. Model *peer to peer* ini,

tiap PC dapat memakai *resource* pada PC lain atau memberikan *resourcenya* untuk dipakai PC lain. Dengan kata lain dapat berfungsi sebagai *client* maupun *server* pada periode yang sama. Metode *peer to peer* ini pada sistem Windows dikenal sebagai *Workgroup*, dimana tiap-tiap komputer dalam satu jaringan dikelompokkan dalam satu kelompok kerja.

b. *Jaringan Client-Server*

Server adalah komputer yang menyediakan fasilitas bagi komputer-komputer lain dalam jaringan dan *client* adalah komputer-komputer yang menerima atau menggunakan fasilitas yang disediakan oleh *server*. Akses dilakukan secara transparan dari *client* dengan melakukan *login* terlebih dahulu ke *server* yang dituju. *Client* hanya bisa menggunakan *resource* yang disediakan *server* sesuai dengan otoritas yang diberikan oleh administrator. Aplikasi yang dijalankan pada sisi *client* bisa saja merupakan *resource* yang tersedia di *server* atau aplikasi yang di-*install* di sisi *client* namun hanya bisa dijalankan setelah terkoneksi ke *server*.

2.2 IP Address (*Internet Protocol Address*).

IP address dirancang untuk memungkinkan terjadinya suatu komunikasi antara sebuah computer dalam suatu jaringan komputer dengan komputer-komputer lainnya baik dalam jaringan komputer yang sama atau jaringan komputer lainnya.

Dengan menentukan *IP address* berarti kita telah memberikan identitas yang universal bagi *interface* komputer. Jika suatu komputer memiliki lebih dari satu *interface* (misalkan menggunakan dua etherrnet) maka kita harus memberi

dua *IP address* untuk komputer tersebut masing-masing untuk setiap *interfacenya*. (Irvan, 2004).

2.3 IPv4 (Internet Protocol Version 4)

IP *address* yang lebih dikenal secara umum dan digunakan saat ini adalah IPversi 4 atau IPv4. Alamat IPv4 merupakan salah satu jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP menggunakan protokol IPv4.Pada dasarnya. Alamat IPv4 terdiri dari 32-bit yang di bagi menjadi empat *octet* dan setiap *octet* terdiri dari 8-bit.IPv4 sendiri terbagi menjadi beberapa kelas, yaitu kelas A, B, C, D dan E(Riza, 2002). Sistem pengalamatan IPv4 dibagi menjadi 5 kelas, berdasarkan jumlah *host* yang dapat dialokasikan, yaitu:

Tabel 2.1.*Tabel Representasi Kelas IPv4 (Rahmat, 2005).*

KELAS ALAMAT IP	OKTET PERTAMA (DESIMAL)	SUBNET MASK (DESIMAL)	PREFIX LENGHT
KELAS A	1-126	255.0.0.0	/8
KELAS B	128-191	255.255.0.0	/16
KELAS C	192-223	255.255.255.0	/24
KELAS D	224-239		
KELAS E	240-255		

Keterangan :

a. Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut bit tertinggi di dalam alamat IP kelas A selalu diset dengan nilai **0** (nol). Tujuh bit berikutnya untuk melengkapi oktet pertama akan membuat sebuah *network identifier*. 24 bit sisanya (atau tiga oktet terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 host tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme Interprocess Communication (IPC) / loopback di dalam perangkat yang bersangkutan.

b. Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua bit pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner **10**. 14 bit berikutnya (untuk melengkapi dua oktet pertama), akan membuat sebuah *network identifier*. 16 bit sisanya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki 16,384 network dan 65,534 host untuk setiap *network*-nya.

c. Kelas C

Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga bit pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner **110**. 21 bit selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bit sisanya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah *network*, dan 254 host untuk setiap *network*-nya.

d. Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner **1110**. 28 *bit* sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host. Untuk lebih jelas mengenal alamat ini, lihat pada bagian Alamat *Multicast* IPv4.

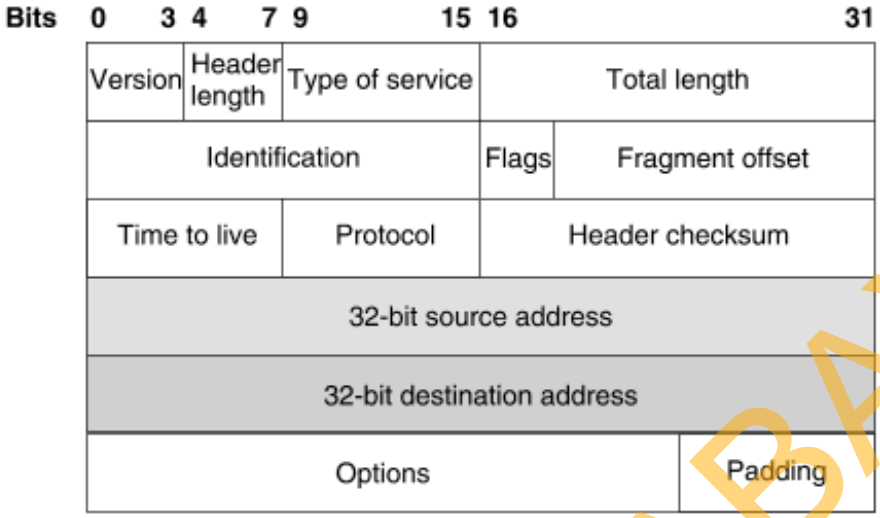
e. Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat "eksperimental" atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat bit pertama selalu diset kepada bilangan biner **1111**. 28 bit sisanya digunakan sebagai alamat yang dapat digunakan untuk mengenali host.

Ada 2 kelas yang ditujukan untuk pemakaian khusus, yakni kelas D dan kelas E *IP Address* Kelas D digunakan untuk multicasting, yaitu pemakaian aplikasi secara bersama-sama oleh sejumlah komputer.Salah satu penggunaan multicast *address* pada internet saat ini adalah aplikasi real time video conference yang melibatkan lebih dari dua host (multipoint) dengan menggunakan Mbone (Multicast Backbone).

Pada jaringan *IP Address* kelas E,merupakan kelas *IP address* yang bersifat "eksperimental" atau percobaan. Eksperimen tersebut dipersiapkan untuk penggunaan *IP Address* di masa yang akan datang.

Format *header* dari IPv4 dapat dilihat pada **Gambar 2.1**:



Gambar 2.1. *Format Header IPv4. (Rahmat, 2005).*

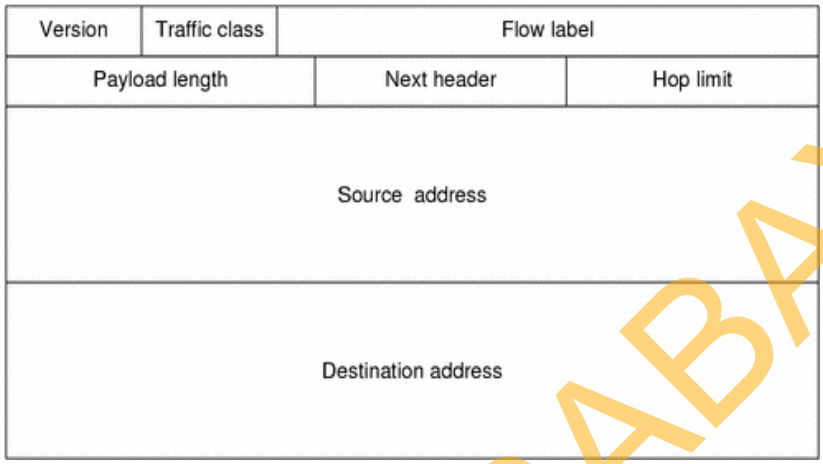
Secara teori IPv4 ini mampu mencakup hingga 4 miliar *host* komputer yang di alamatkannya. Sehingga bila suatu saat batas kuota tersebut melebihi *host* yang ada diseluruh dunia maka akan terjadi kekurangan pengalamatan untuk *host-host* baru yang bermunculan, sehingga dikembangkanlah pengalamatan jenis baru yang sekarang dikenal dengan IP versi 6 atau IPv6. (Rahmat, 2003).

2.4 IPv6 (Internet Protocol Version 6)

Alamat IPv6 atau di kenal dengan *Next Generation Internet Protocol* atau IPng. Pengalamatan jenis ini mulai dikenalkan pada pertengahan tahun 1994 oleh *Ipng Area Detector dari Internet Engineering Task Force* (IETF). IPv6 adalah salah satu jenis pengalamatan jaringan yang juga di pergunakan dalam lingkup *protocol* jaringan TCP/IP yang menggunakan protokol IP versi 6. *IP address* ini memiliki ukuran 128-bit (16-byte), dan secara teoritis dapat mengalamati hingga 2^{128}=3,4 x 10^{38} *host* komputer di seluruh dunia. Sehingga begitu besar jumlah

pengalamatan *host* yang dapat dicakup oleh IP jenis ini. Contoh alamat IP versi 6 adalah 2002:c0a8:b1::/64. (Rahmat, 2005).

Format *header* dari IPv6 dapat dilihat pada **Gambar 2.2** :



Gambar 2.2. *Format header IPv6. (Robert, 1995).*

Jika dilihat dari cakupan alamatnya, alamat IPv6 terbagi beberapa jenis alamat berikut:

1. ***Link-Local,*** merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam satu subnet. Contoh: **FE80::/10 (FE8, FE9, FEA, FEB)**
2. ***Site-Local,*** merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam sebuah intranet. Contoh : **FEC0::/10 (FEC, FED, FEE, FEF)**
3. ***Global Address,*** merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam Internet berbasis IPv6. Contoh: **2001::/32 (2002, 2003, 2400, 2600, 2A00, 2E00, 3000)**

Pada implementasi integrasi jaringan IPv4 dan IPv6 menggunakan sistem *tunneling*. Penulis memilih IPv6 karena hal ini merupakan suatu langkah baru

untuk meminimalisir permasalahan kekurangan pengalamatan *host* yang terjadi. Versi IP baru ini dirancang untuk suatu tindakan *evolusiner* dari IPV4. Secara langsung IPv4 dengan IPv6 tidak dapat dihubungkan, maka dibutuhkan suatu sistem *tunneling* untuk mengintergrasiakan keduanya. *Tunnel* di dalam dunia jaringan komputer diartikan sebagai suatu cara untuk mengenkapsulasikan atau membungkus *packet* IP didalam *packet* IP yang lain. (Hendra, 2007).

2.5 Tunneling

Tunneling merupakan suatu sistem yang digunakan untuk proses peng-engkapsulisasian *IP address*, baik peng-enkapsulasian IPv6 dalam *packet* IPv4 atau sebaliknya. Sistem *tunneling* ini digunakan mengintegrasikan kedua IP tersebut yaitu dengan cara membawa data IPv6 melalui jaringan IPv4 yang masih existing. (Wahidi, 2003).

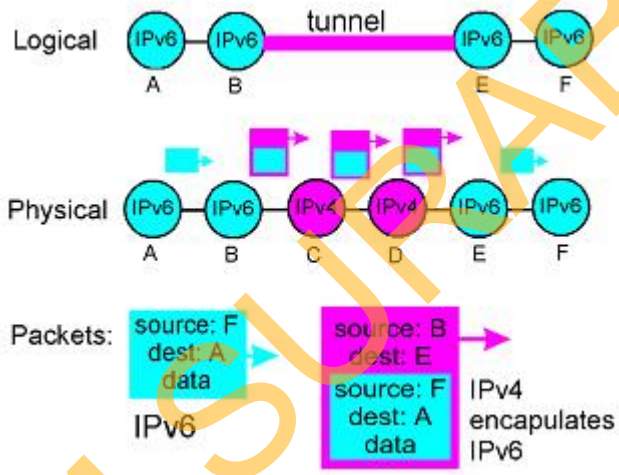
Di dalam sistem *tunnelling* terdapat suatu aspek yang paling penting, yaitu *payload* atau biasa disebut dengan *packet* data asli yang bisa jadi merupakan suatu *unsupported protocol* atau protokol yang tidak dikenal. Pada *tunneling* terdapat *header* yang diperlukan sehingga *packet* data tersebut dapat dikirim melalui infrastruktur jaringan dan diterima oleh tujuan. (Wahidi, 2003).

Packet tunnel yang dikirim melalui jaringan dengan menggunakan *tunnel*. Saat *node* tujuan menerima *packet tunnel*, maka *packet tunnel* tersebut akan di enkapsulasikan kedalam *packet* data hasil. (Wahidi, 2003).

Sistem Tunneling merupakan suatu sistem *tunneling* yang berfungsi untuk melewatkan *packet* IPv6 melalui jaringan IPv4 tanpa merubah infrastruktur dari jaringan tersebut. *Tunneling* jenis ini memilki prinsip kerja yang mengengkapsulasikan *packet* IPv6 ke *header* IPv4 yang kemudian langsung

dikirim ke jaringan IPv4. Fungsi dari enkapsulasi *packet* IPv6 tersebut adalah supaya *packet* tersebut dapat di*routing*-kan oleh *router* IPv4 tersebut. Namun dengan adanya penambahan *header* IPv4 ini, *packet* tersebut akan bertambah besar sesuai dengan panjang dari *header* IPv4. Pertambahan panjang *packet* ini akan mengakibatkan pertambahan waktu *delay* pada proses pengiriman *packet* tersebut. (Jonas, 2008).

Sistem *tunneling* dapat dilihat pada **Gambar 2.3.**



Gambar 2.3. *Tunneling.* (James, 2000).

2.6 *Server* Komputer

Server komputer adalah suatu sistem komputer yang dibuat untuk menjalankan aplikasi *server*. Sebuah komputer *server* yang di fungsikan untuk menjalankan salah satu aplikasi *server* yang spesifik sering kali komputer *server* tersebut dikenal dengan nama dari aplikasinya. Sebagai contoh, misalkan pada komputer *server* digunakan *software Apache HTTP server* biasanya di sebut *WebServer* saja. Pada dasarnya aplikasi *server* adalah *fleksibe*l, dalam artian aplikasi *server* dapat dibagi menjadi beberapa komputer tergantung pada kebutuhan dan beban. (Dedi, 2010).

2.7 *Web Server*

Webserver merupakan suatu aplikasi yang berfungsi untuk menproses permintaan dari *client* dalam bentuk *web* atau *world wide web* (www). *Webserver* bertugas menunggu permintaan dari *client* yang menggunakan *browser* seperti *Netscape Navigator, Internet Explorer, Modzilla*, dan program *browser* lainnya. Jika ada permintaan dari *browser*, maka *webserver* akan mengeksekusi permintaan tersebut dan kemudian memberikan hasil dari proses yang dilakukan kepada *browser*. Data ini mempunyai format yang standar, disebut dengan format SGML (*Standar General Markup Language*). (Feit Sidney, 1996).

2.8 LAN (*Local Area Networ*).

Local Area Network biasa disingkat LAN adalah jaringan komputer yang jaringannya hanya mencakup wilayah kecil; seperti jaringan komputer kampus, gedung, kantor, dalam rumah, sekolah atau yang lebih kecil. Saat ini, kebanyakan LAN berbasis pada teknologi IEEE 802.3 Etherrnet menggunakan perangkat switch, yang mempunyai kecepatan transfer data 10, 100, atau 1000 Mbit/s.

Selain teknologi Etherrnet, saat ini teknologi 802.11b (atau biasa disebut *Wi-fi*) juga sering digunakan untuk membentuk LAN. Tempat-tempat yang menyediakan koneksi LAN dengan teknologi *Wi-fi* biasa disebut *hotspot*.

Pada sebuah LAN, setiap node atau komputer mempunyai daya komputasi sendiri, berbeda dengan konsep *dump terminal*. Setiap komputer juga dapat mengakses sumber daya yang ada di LAN sesuai dengan hak akses yang telah diatur. Sumber daya tersebut dapat berupa data atau perangkat seperti printer.

Pada LAN, seorang pengguna juga dapat berkomunikasi dengan pengguna yang lain dengan menggunakan aplikasi yang sesuai.

Berbeda dengan Jaringan Area Luas atau *Wide Area Network* (WAN), maka LAN mempunyai karakteristik sebagai berikut :

1. Mempunyai pesat data yang lebih tinggi.
2. Meliputi wilayah geografi yang lebih sempit.
3. Tidak membutuhkan jalur telekomunikasi yang disewa dari operator telekomunikasi.

Biasanya salah satu komputer di antara jaringan komputer itu akan digunakan menjadi *server* yang mengatur semua sistem di dalam jaringan tersebut. (Iwan Sofana, 2011).

2.9 Mikrotik Router OS

MikroTik *Router*OS™ adalah sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan computer menjadi *router network* yang handal, mencakup berbagai fitur yang dibuat untuk ip *network* dan jaringan wireless, cocok digunakan oleh ISP dan *provider hotspot.*

Untuk instalasi Mikrotik tidak dibutuhkan piranti lunak tambahan atau komponen tambahan lain. Mikrotik didesain untuk mudah digunakan dan sangat baik digunakan untuk keperluan administrasi jaringan komputer seperti merancang dan membangun sebuah sistem jaringan komputer skala kecil hingga yang kompleks sekalipun.

Mikrotik dibuat oleh MikroTikls sebuah perusahaan di kota Riga, Latvia. Latvia adalah sebuah Negara yang merupakan “pecahan” dari negara Uni Soviet dulunya atau Rusia sekarang ini. Dengan nama merek dagang Mikrotik mulai

didirikan tahun 1995 yang pada awalnya ditujukan perusahaan jasa layanan Internet (PJI) atau *Internet Service Provider* (ISP) yang melayani pelanggannya menggunakan teknologi nirkabel atau *wireless*. Saat ini MikroTikls memberikan layanan kepada banyak ISP nirkabel untuk layanan akses Internet dibanyak negara di dunia dan juga sangat populer di Indonesia.

Mikrotik pada standar perangkat keras berbasiskan Personal Computer (PC) dikenal dengan kestabilan, kualitas kontrol dan fleksibilitas untuk berbagai jenis *packet* data dan penanganan proses rute atau lebih dikenal dengan istilah *routing*. Mikrotik yang dibuat sebagai *router* berbasiskan PC banyak bermanfaat untuk sebuah ISP yang ingin menjalankan beberapa aplikasi mulai dari hal yang paling ringan hingga tingkat lanjut. Contoh aplikasi yang dapat diterapkan dengan adanya Mikrotik selain *routing* adalah aplikasi kapasitas akses (*bandwidth*) manajemen, firewall, wireless access point (WiFi), backhaul link, sistem hotspot, *Virtual Private Network* (VPN) *server* dan masih banyak lainnya. (Aziz Catur, 2005).

2.10 Router (RouterBoard 750)

Router merupakan salah satu perangkat dalam dunia jaringan komputer. Pengertian *Router* adalah perangkat jaringan yang berfungsi untuk menghubungkan beberapa jaringan atau *network*, baik jaringan yang menggunakan teknologi sama atau yang berbeda, misalnya menghubungkan jaringan topologi Bus, topologi Star atau topologi Ring.

Karena *router* ini menghubungkan beberapa jaringan tentunya *router* berbeda dengan *Switch*. *Switch* hanya perangkat yang digunakan untuk menghubungkan beberapa komputer sehingga membentuk LAN atau *local area*

network. Sedangkan *router* adalah perangkat yang menghubungkan satu LAN dengan banyak LAN lainnya.

Router dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang disebut dengan *internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya. *Router* juga kadang digunakan untuk mengoneksikan dua buah jaringan yang menggunakan media yang berbeda atau berbeda arsitektur jaringan, seperti halnya dari Etherrnet ke Token Ring.

Router umumnya dipakai untuk jaringan berbasis teknologi protokol TCP/IP, *router* jenis ini dinamakan IP *Router*. Internet merupakan contoh utama dari jaringan yang memiliki IP *Router*.

Umumnya *router* ada dua jenis konfigurasi *router*, yaitu *router statis* dan *router dinamis*, *Router statis* atau *static router* merupakan *router* yang memiliki tabel routing statis yang disetting dengan cara manual oleh para administrator jaringan. Sedangkan *router dinamis* atau *dynamic router* merupakan *router* yang memiliki dan membuat tabel routing dinamis dengan membaca lalu lintas jaringan dan juga dengan saling berhubungan dengan *router* lainnya. (Azis Catur,2005)

Pada implementasi kali ini penulis menggunakan *RouterBoard* (RB) 750 keluaran dari Produsen *Router* dunia yaitu Mikrotik, alasan menggunakan RB 750 karena *Router* dengan tipe ini di dalamnya sudah terdapat paket *file* IPv6 bawaan dari *Router*nya yang tentunya dengan paket file tersebut dapat dimanfaatkan untuk pemakaian serta konfigurasi IPv6 itu sendiri.

Syntax atau perintah – perintah konfgiurasi pada *router* yang digunakan untuk membangun implementasi integrasi jaringan IPv6 dengan jaringan IPv4 pada *Local Area Network* (LAN) menggunakan Sistem *Tunneling*, seperti pada **Tabel 2.2.**

Tabel 2.2. Tabel keterangan *syntax* atau perintah-perintah konfigurasi pemasangan alamat IPv4 *address* pada *Router*.

Fungsi	**Syntax / Perintah**
Pemasangan alamat IPv4 *address*.	Ip *address add address* = [alamat IPv4] *interface* = [pilihan *interface/ethernet*] *disabled* = [pilihan aktivasi IP *address* (*yes/no*)]
Detil keterangan syntax/perintah	
Syntax	**Keterangan**
Ip *address add address* = [alamat IPv4]	Perintah untuk memasang alamat Ipv4 *address* ke dalam suatu *interface/ethernet* pada *router*.
interface = [pilihan *interface/ethernet*]	Perintah untuk *setting* penempatan alamat IPv4 *address* di *interface/ethernet* mana yang akan dipasangkan alamat IPv4 *address*.
disabled = [pilihan aktivasi IP *address* (*yes/no*)]	Perintah untuk mengaktifkan alamat IPv4 *address* yang telah dipasang.

Tabel 2.3. Tabel keterangan *syntax* atau perintah-perintah konfigurasi pembuatan *interface/ethernet* jalur *tunneling 6to4* pada *Router.*

Fungsi	***Syntax* / Perintah**
Pembuatan *interface /ethernet* jalur *tunneling 6to4.*	*/interface 6to4 add mtu* = [besar pembagian paket] *name*=[nama *interface/ethernet* baru] *local-address* = [alamat lokal IPv4] *disabled*= [pilihan aktivasi alamat IP *address* (*yes/no*)]
Detil keterangan *syntax*/perintah	
Syntax	**Keterangan**
/interface 6to4 add mtu = [besar pembagian paket]	Perintah untuk membuat *interface*/*ethernet tunnel* 6to4 dan membagi besaran paket data.
Name = [nama *interface/ethernet* baru]	Perintah untuk memberi penamaan pada *interface* / *tunnel* 6to4 yang sudah dibuat
local-address = [alamat lokal IPv4]	Perintah untuk menentukan *local–address* IPv4 untuk menentukan titik enkapsulasi dan dekapsulasi pada *Router.*
Disabled = [pilihan aktivasi alamat IP *address* (*yes/no*)]	Perintah untuk mengaktifkan *interface/ethernet tunneling* yang telah dibuat.

Tabel 2.4. Tabel keterangan *syntax* atau perintah-perintah konfigurasi pemasangan alamat IPv6 *address* pada *Router*.

Fungsi	***Syntax* / Perintah**
Pemasangan alamat IPv6 *address*.	/IPv6 *address add address* = [alamat IPv6] *interface* = [pilihan *interface/ethernet*]
Detil keterangan *syntax*/perintah	
Syntax	**Keterangan**
IPv6 *address add address* = [alamat IPv6 *address*]	Perintah untuk menambahkan alamat IPv6 *address*
interface = [pilihan *interface/ethernet*]	Perintah untuk *setting* penempatan alamat IPv6 *address* di *interface/ethernet* mana yang akan dipasangkan alamat IPv6 *address*.

Tabel 2.5. Tabel keterangan *syntax* atau perintah-perintah konfigurasi *gateway tunnel* dan *routing* IPv6 pada *Router*.

Fungsi	***Syntax* / Perintah**
Konfigurasi *gateway tunnel* dan *routing* IPv6.	IPv6 *route add dst-address* = [alamat *prefix* jaringan yang akan dituju] *gateway* = [IP *compatible* untuk jalur *tunnel*] % [*interface/ethernet* 6to4 yang telah dibuat]
Detil keterangan *syntax*/perintah	
Syntax	**Keterangan**
IPv6 *route add dst-address* = [alamat *prefix* jaringan yang akan dituju]	Perintah untuk menambahkan routing IPv6 dan penetapan alamat entri tabel routingnya
gateway = [IP *compatible* untuk jalur *tunnel*]	Perintah untuk konfigurasi *gateway* dengan tujuan ke *Router*2 berupa IPv4 dari Eth1 di *Router*2 dirubah menjadi IP *Compatible* dengan ditambahkan tanda titik dua **(::)** sebanyak 2 kali gabung dengan angka yang paling depan, contoh : *(::30.20.20.3)*
% [*interface/ethernet* 6to4 yang telah dibuat]	Konfigurasi untuk melewatkan routing IPv6 melalui *interface*/*ethernet tunneling* 6to4 yang telah dibangun.

2.11 Konversi Desimal ke Hexadesimal

Pada implementasi integrasi jaringan IPv6 dengan jaringan IPv4 menggunakan Sistem *Tunneling* terdapat beberapa tahapan konfigurasi yang salah satu tahapannya adalah melakukan konversi dari bilangan desimal ke bilangan hexadesimal.

Salah satu tahap konfigurasi untuk mendaparkan alamat IPv6 untuk *gateway tunnel* yaitu dengan mengkonversi alamat IPv4 yang sudah ada dan merupakan bilangan desimal kemudian dikonversi/dirubah ke dalam alamat IPv6 yang merupakan bilangan hexadesimal.

Contoh cara melakukan konversi dari bilangan desimal IPv4 dirubah menjadi bilangan hexadesimal IPv6, sebagai berikut :

Cara Konversi IPv4 ke IPv6 :

1. Langkah pertama mengubah bilangan desimal ke bilangan *binary.*

30.30.20.2 =

30 =	**30 =**	**20 =**	**2 =**
30 / 2 = 15 sisa 0	30 / 2 = 15 sisa 0	20 / 2 = 10 sisa 0	2 / 2 = 1 sisa 0
15 / 2 = 7.5 sisa 1	15 / 2 = 7.5 sisa 1	10 / 2 = 5 sisa 0	1/ 2 = 0.5 sisa 1
7 / 2 = 3.5 sisa 1	7 / 2 = 3.5 sisa 1	5 / 2 = 2 sisa 1	
3 / 2 = 1.5 sisa 1	3 / 2 = 1.5 sisa 1	2 / 2 = 1 sisa 0	
1 / 2 = 0.5 sisa 1	1 / 2 = 0.5 sisa 1	1 / 2 = 0,5 sisa 1	
à 11110	**à 11110**	**à 10100**	**à 10**

2. Langkah ke 2 merubah format *binary* yang sudah dikonversi tadi ke dalam *binary* yang sudah siap untuk dirubah ke hexadesimal dengan cara menambah angka 0 didepan *binary* yang sudah dikonversi tadi sampai berjumlah 8 digit

karena 1 hexa terdiri dari 4 *binary* digit.

000**11110 .** 000**11110 .** 000**10100 .** 00000**10**

3. Selanjutnya adalah mengubah bilangan *binary* ke hexadesimal dengan cara merubah 4 bilangan *binary* digit ke hexadesimal yang sesuai dengan melihat **Tabel 3.1.**

***Tabel 3.1**. Tabel Konversi Hexadesimal.*

Hex	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1
Binari	1111	1110	1101	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Decimal	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

→ 0001 1110 . 0001 1110 . 0001 0100 . 0000 0010

1 E 1 E 1 4 0 2

à 1E1E:1402

2002 = prefix global

1e1e:1402= Alamat IPv4 dalam hexa (30.30.20.2 = 1e1e:1402)

Jadi alokasi alamat dan tabel routing IPv6 di jaringan di ***Router*** 1, adalah *2002:1e1e:1402::*

2.12 Kabel UTP (*Unshielded Twisted Pair*)

Kabel UTP merupakan salah satu media transmisi yang paling banyak digunakan untuk membuat sebuah jaringan local *(Local Area Network)*, selain karena harganya relative murah, mudah dipasang dan cukup bisa diandalkan.

Sesuai namanya *Unshielded Twisted Pair* berarti kabel pasangan berpilin/terbelit *(twisted pair)* tanpa pelindung *(unshielded)*. Fungsi lilitan ini adalah sebagai eleminasi terhadap induksi dan kebocoran.(Dede Sopandi,2005)

Terdapat beberapa jenis kategori kabel UTP ini yang menunjukkan kualitas, jumlah kerapatan lilitan *pair*nya, semakin tinggi katagorinya semakin rapat lilitannya dan parameter lainnya seperti berikut ini:

a. Kabel UTP *Category* 1

Digunakan untuk komunikasi telepon (mentransmisikan data kecepatan rendah), sehingga tidak cocock untuk mentransmisikan data.

b. Kabel UTP *Category* 2

Mampu mentransmisikan data dengan kecepatan sampai dengan 4 Mbps *(Megabits per second)*

c. Kabel UTP *Category* 3

Digunakan pada **10BaseT** *network*, mampu mentransmisikan data dengan kecepatan sampai 1Mbps. 10BaseT kependekan dari 10 Mbps, *Baseband*, *Twisted pair*.

d. Kabel UTP *Category* 4

Sering digunakan pada topologi token ring, mampu mentransmisikan data dengan kecepatan sampai 16 Mbps.

e. Kabel UTP *Category* 5

Mampu mentransmisikan data dengan kecepatan sampai 100 Mbps.

f. Kabel UTP *Category* 5e

Mampu mentransmisikan data dengan kecepatan sampai 1000 Mbps *(1Gbps)*, frekwensi signal yang dapat dilewatkan sampai 100 MHz.

g. Kabel UTP *Category* 6

Mampu mentransmisikan data dengan kecepatan sampai 1000 Mbps *(1Gbps)*, frekwensi signal yang dapat dilewatkan sampai 200 MHz. Secara fisik terdapat separator yg terbuat dari plastik yang berfungsi memisahkan keempat *pair* di dalam kabel tersebut.

h. Kabel UTP *Category* 7 gigabit Etherrnet (1Gbps), *frekwensi signal* 400 MHz

Untuk pemasangan kabel UTP, terdapat dua jenis pemasangan kabel UTP yang umum digunakan pada jaringan komputer terutama LAN, yaitu:

1. ***Straight Through Cable***

Kabel *straight* merupakan kabel yang memiliki cara pemasangan yang sama antara ujung satu dengan ujung yang lainnya. Kabel ***straight*** digunakan untuk menghubungkan 2 device yang berbeda. Urutan standar kabel *straight* adalah seperti **Gambar 2.4** yaitu sesuai dengan standar TIA/EIA 368B (yang paling banyak dipakai) atau kadang-kadang juga dipakai sesuai standar TIA/EIA 368A.

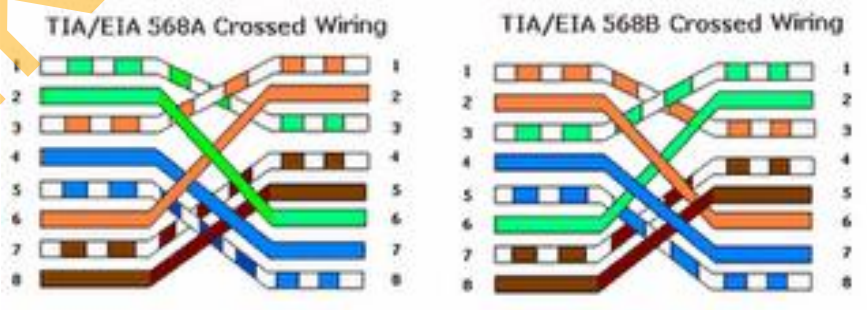
Gambar 2.4. *Susunan Standar Kabel Straight dengan Standar TIA/EIA 568A dan TIA/EIA 568B.*

Contoh penggunaan kabel *straight* adalah sebagai berikut :

1. Menghubungkan antara computer dengan switch
2. Menghubungkan computer dengan LAN pada modem *cable*/DSL
3. Menghubungkan router dengan LAN pada modem *cable*/DSL
4. Menghubungkan switch ke *router*
5. Menghubungkan hub ke *router*

2. ***Cross Over Cable***

Kabel *cross over* merupakan kabel yang memiliki susunan berbeda antara ujung satu dengan ujung dua. Kabel *cross over* digunakan untuk menghubungkan 2 device yang sama. **Gambar 2.5** adalah susunan standar kabel *cross over*.



Gambar 2.5. *Susunan Standar Kabel Cross Over dengan Standar TIA/EIA 568A dan TIA/EIA 568B.*

Contoh penggunaan kabel *cross over* adalah sebagai berikut :

1. Menghubungkan 2 buah komputer secara langsung
2. Menghubungkan 2 buah *switch*
3. Menghubungkan 2 buah *hub*
4. Menghubungkan *switch* dengan *hub*
5. Menghubungkan komputer dengan *router*

Dari 8 buah kabel yang ada pada kabel UTP ini (baik pada kabel *straight* maupun *cross over*) hanya 4 buah saja yang digunakan untuk mengirim dan menerima data, yaitu kabel pada pin no 1, 2, 3 dan 6. (Iwan Sofana, 2011).

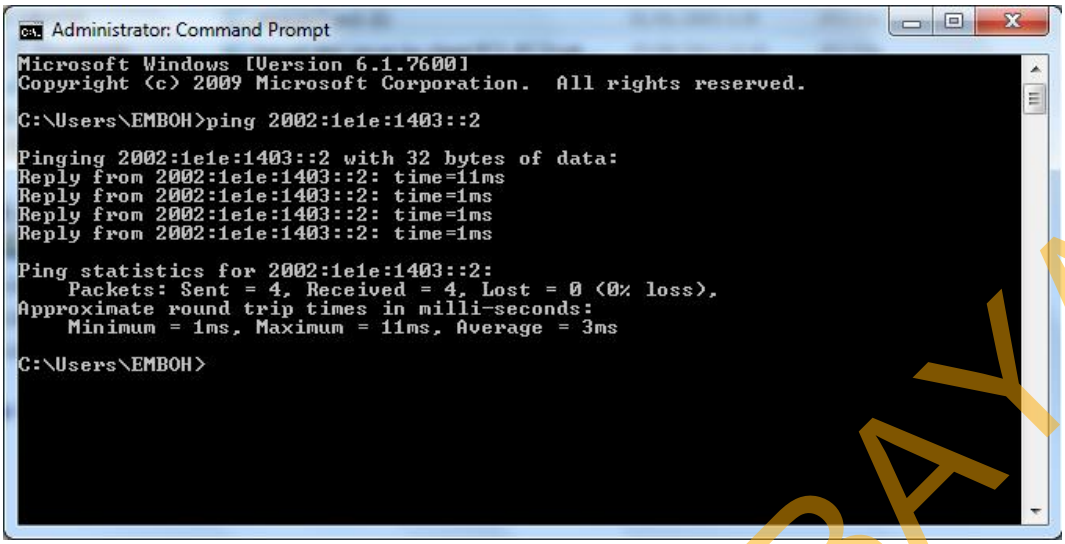
2.13 PING (Packet Internet Gopher)

PING merupakan salah satu program yang digunakan untuk mengecek komunikasi antar komputer dalam sebuah jaringan melalui protokol TCP/IP. *PING* akan mengirimkan *Internet Control Message Protocol* (ICMP) *Echo Request messages* pada *ip address* komputer yang dituju dan meminta respons dari komputer tersebut pada implementasi integrasi jaringan Ipv6 dengan jaringan IPv4 menggunakan Sistem *Tunneling* dilakukan 50 kali percobaan PING untuk mengetahui kualitas jaringan yang dihasilkan.

Berikut ini adalah beberapa pesan ICMP yang biasa disampaikan oleh program *Ping*:

1. Echo Reply.

Pesan ini digunakan untuk merespon ping ketika sistem masih hidup, dan pesan ini menandakan bahwa sudah terjadi koneksi antara pengirim dan penerima paket.



Gambar 2.6. Tampilan pesan Echo Reply.

Jika komputer target memberikan respons maka komputer tersebut memberikan informasi seperti contoh *PING report* yang anda berikan yaitu:

bytes=32 time=30ms TTL=123.

Bytes menunjukkan besar request *packet* yang dikirimkan. *Time* menunjukkan nilai "*round trip delay*" (disebut juga sebagai delay atau *latency*) yang menunjukkan waktu yang diperlukan *packet* yang anda kirimkan untuk mencapai komputer yang dituju. Nilai ini dihitung dengan membagi dua selisih waktu *PING packet* mulai dikirimkan dengan waktu response dari *PING packet* diterima.

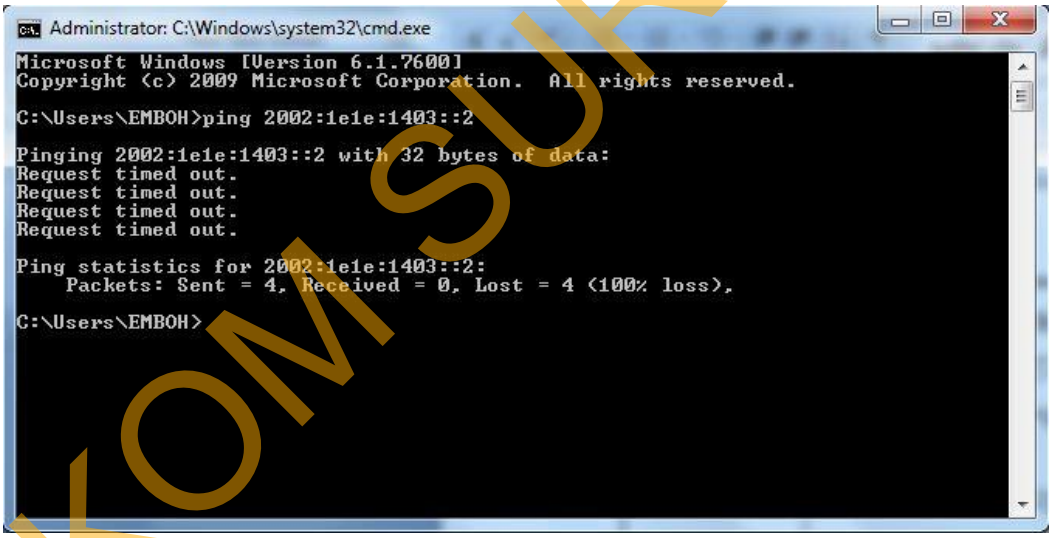
Sedangkan TTL merupakan nilai "*Time-To-Live*" yang digunakan untuk mencegah adanya *circular routing* pada suatu jaringan. Dengan mengurangi nilai TTL awal yaitu 128 dengan nilai TTL akhir maka bisa dihitung banyaknya hop yang dilalui dari komputer asal ke komputer tujuan. Setiap kali *PING packet* melalui sebuah *ip address* maka nilai TTL nya akan dikurangi satu. Sehingga jika

TTL mencapai nilai nol, *PING packet* akan di-*discard* / didrop dan hasil *PING* menunjukkan: TTL *expired in transit*

2. Request Time Out.

Ketika komputer server tidak merespon permintaan koneksi dari klien setelah beberapa lama (jangka waktu timeout bervariasi) antara lain karena:

1. Utilisasi/pemakaian bandwidth sudah penuh. solusi harus upgrade kecepatan.
2. Kualitas akses jaringan (wireless/wireline) kurang bagus.
3. Website yang dituju memiliki delay yang tinggi, sehingga ping timeout.
4. Koneksi ke IP tersebut putus, atau
5. Port di komputer tersebut ditutup.



Gambar 2.7. Tampilan pesan Request Time Out (RTO).

Kegunaan *PING* antara lain adalah sebagai berikut :

a. Mengetahui status up/down komputer dalam jaringan. Kita dapat mengecek apakah sebuah komputer up/down menggunakan perintah *PING*, jika komputer tersebut memberikan response terhadap perintah

PING yang kita berikan maka dikatakan bahwa komputer tersebut up atau hidup.

b. Memonitor availability status komputer dalam jaringan. *PING* dapat digunakan sebagai *tool monitoring availibilitas* komputer dalam jaringan yang merupakan salah satu indikator kualitas jaringan yaitu dengan melakukan *PING* secara periodik pada komputer yang dituju. Semakin kecil *downtime*, semakin bagus kualitas jaringan tersebut.

c. Mengetahui responsifitas komunikasi sebuah jaringan. Besarnya nilai *delay* atau *latency* yang dilaporkan oleh *PING* menjadi indikasi seberapa responsif komunikasi terjadi dengan komputer yang dituju. Semakin besar nilai *delay* menunjukkan semakin lamban respons yang diberikan. Sehingga nilai *delay* ini juga bisa digunakan sebagai indikator kualitas jaringan.

Banyak aplikasi hanya bisa dijalankan dengan maksimal *delay* tertentu, sehingga sangat penting untuk mengukur *delay* pada jaringan untuk memastikan aplikasi tersebut dapat dijalankan. Aplikasi yang memerlukan *delay* kecil dikatakan sebagai *delay-sensitive application* dan memerlukan jaminan agar maksimal *delay* selalu terjaga dalam komunikasi data yang dilakukan, contohnya adalah *network game*, *voice* dan *video conference application*. (Iwan Sofana, 2009).

2.14 Sinyal Kontrol

Sinyal control adalah suatu sinyal yang berfungsi mengatur jaringan dan menetapkan panggilan, mempertahankan panggilan, serta menghentikan panggilan. (Amzar,2003).

Salah satu kerja Sinyal kontrol yang ada pada uji coba kali ini yaitu pada saat akan melakukan uji coba download file via web server yang akan dilakukan oleh client. Client melakukan request/permintaan untuk dapat mengakses web server yang telah disediakan oleh server dengan cara mentransmisikan perintah berupa sinyal kontrol agar dapat berkomunikasi dengan server, kemudian oleh server secara otomatis akan melakukan feedback ke client dengan melakukan pengiriman hasil request/permintaan yang diminta oleh client tadi berupa halaman web server atau file sesuai yang perintah dari request/permintaan yang dilakukan oleh Client.