



**UNIVERSITAS INDONESIA**

**ANALISA PERFORMA JARINGAN AUTOMATIC 6TO4  
TUNNELING DAN JARINGAN MANUALLY CONFIGURED  
IPV6 TUNNELING DENGAN MENGGUNAKAN APLIKASI  
VLC DAN HELIX STREAMING SERVER**

**SKRIPSI**

**ARDIAN PRAWIRAYUDHA  
0404030121**

**FAKULTAS TEKNIK ELEKTRO  
PROGRAM STUDI TEKNIK KOMPUTER  
DEPOK  
DESEMBER 2008**



**UNIVERSITAS INDONESIA**

**ANALISA PERFORMA JARINGAN AUTOMATIC 6TO4  
TUNNELING DAN JARINGAN MANUALLY CONFIGURED  
IPV6 TUNNELING DENGAN MENGGUNAKAN APLIKASI  
VLC DAN HELIX STREAMING SERVER**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik Elektro**

**ARDIAN PRAWIRAYUDHA  
0404030121**

**FAKULTAS TEKNIK ELEKTRO  
PROGRAM STUDI TEKNIK KOMPUTER  
DEPOK  
DESEMBER 2008**

## **HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Ardian Prawirayudha**

**NPM : 0404030121**

**Tanda Tangan :**

**Tanggal : 12 Desember 2008**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Ardian Prawirayudha  
NPM : 0404030121  
Program Studi : Teknik Komputer  
Judul Skripsi : Analisa Performa Jaringan *Automatic 6to4 Tunneling* dan Jaringan *Manually Configured IPv6 Tunneling* dengan Menggunakan Aplikasi VLC dan *HELIX Streaming Server*

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik Elektro pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia**

## DEWAN PENGUJI

Pembimbing : Ir. Endang Sriningsih, MT, Si ( )  
Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng ( )  
Penguji : Muhammad Salman ST., MIT ( )

Ditetapkan di : Kampus UI Depok  
Tanggal : 12 Desember 2008

## **KATA PENGANTAR/UCAPAN TERIMA KASIH**

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga dapat terselesaikannya skripsi ini. Shalawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW. Saya sangat menyadari bahwa, tanpa bantuan dari berbagai pihak, sangatlah sulit untuk menyelesaikan skripsi ini dengan baik dan tepat waktu. Oleh karena itu, saya ingin mengucapkan terima kasih kepada:

1. Ir. Endang Sriningsih, MT, yang bersedia meluangkan waktunya membimbing saya dalam menyusun skripsi ini;
2. Mama dan Papa serta keluarga yang lain yang telah memberikan dukungan moral dan material;
3. Sahabat saya sekaligus rekan satu tim saya, Ihtiar yang telah memberikan kerjasama yang baik selama masa penyusunan skripsi ini.
4. Sahabat-sahabat lain yang telah membantu dalam berbagai hal.

Akhir kata, saya berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan.

Depok, 10 Desember 2008

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Ardian Prawirayudha  
NPM : 0404030121  
Program Studi : Teknik Komputer  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demikian pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**Analisa Performa Jaringan *Automatic 6to4 Tunneling* dan Jaringan *Manually Configured IPv6 Tunneling* dengan Menggunakan Aplikasi VLC dan HELIX Streaming Server**

berserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok, Indonesia

Pada tanggal : 12 Desember 2008

Yang menyatakan

(Ardian Prawirayudha)

## ABSTRAK

Nama : Ardian Prawirayudha  
Program Studi : Teknik Elektro  
Judul : Analisa Performa Jaringan *Automatic 6to4 Tunneling* dan Jaringan *Manually Configured IPv6 Tunneling* dengan Menggunakan Aplikasi VLC dan HELIX *Streaming Server*

Skripsi ini membandingkan performa dari jaringan *automatic 6to4 tunneling* dengan jaringan *manually configured IPv6 tunneling*. Uji coba dilakukan pada jaringan lokal dengan menggunakan empat macam topologi jaringan, jaringan IPv4, jaringan IPv6, jaringan *automatic 6to4 tunneling* dan jaringan *manually configured IPv6 tunneling*. Aplikasi yang digunakan berupa aplikasi *video streaming* yaitu VLC dan HELIX *streaming server*. Uji coba dilakukan dengan menggunakan dua buah *laptop*, *router cisco 3700* dan *3800 series*, serta sebuah *layer-2 switch*. Parameter yang diukur adalah *packet loss* dan *throughput*. Variasi dalam pengambilan data dilakukan dengan menggunakan *file video* dengan format yang berbeda-beda. Tiap format *video* dilakukan pengambilan data sebanyak lima kali tiap topologi jaringan. Hasil pengolahan data menunjukkan bahwa jaringan *manually configured IPv6 tunneling* memiliki performa yang lebih bagus dibandingkan jaringan *automatic 6to4 tunneling* yang ditunjukkan dengan lebih kecilnya presentase *packet loss* yang dihasilkannya. Selisih presentase *packet loss* sebesar 0,38% pada *streaming* menggunakan VLC dan 1,3% pada *streaming* menggunakan HELIX.

Kata kunci: *tunneling*, IPv4, IPv6, *manually configured*, *6to4*, *video streaming*

## ABSTRACT

Name : Ardian Prawirayudha  
Study Program: Computer Engineering  
Title : Performance Analysis of Automatic 6to4 Tunneling Network and Manually Configured IPv6 Tunneling Network by Using VLC and HELIX Streaming Server

This thesis compares performances of automatic 6to4 tunneling network and manually configured IPv6 tunneling network. Testing is done on a local network by using four kinds of network topology; they are IPv4 network, IPv6 network, automatic 6to4 tunneling network and manually configured IPv6 tunneling network. Testing uses two video streaming applications, they are VLC and HELIX streaming server. The local network uses two laptops, 3800 and 3700 series Cisco routers, and a layer-2 switch. The parameters are packet loss and throughput. Variation is done by using videos in different formats. The data are collected five times each video format on each topology. The result of data processing show that manually configured IPv6 tunneling network has better performance compared with automatic 6to4 tunneling network because it has lower packet loss percentage. The packet loss difference is about 3,8% in streaming using VLC and 1,3% in streaming using HELIX.

Key words: tunneling, IPv4, IPv6, manually configured, 6to4, video streaming



## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR/UCAPAN TERIMA KASIH.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
DAFTAR SINGKATAN.....	xii
<b>1. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penulisan.....	2
1.4 Batasan Masalah.....	2
1.5 Sistematika Penulisan.....	3
<b>2. PENGENALAN IPV6 DAN METODE TRANSISI TUNNELING.....</b>	<b>4</b>
2.1 <i>Internet Protocol</i> .....	4
2.2 IPv4 dan keterbatsannya.....	5
2.3 IPv6.....	5
2.3.1 Struktur IPv6.....	6
2.3.1.1 Header IPv6.....	6
2.3.1.2 Extension Header.....	8
2.3.2 Pengalamatan pada IPv6.....	8
2.3.2.1 Alamat <i>Unicast</i> .....	9
2.3.2.2 Alamat <i>Anycast</i> .....	10
2.3.2.3 Alamat <i>Multicast</i> .....	11
2.4 Metode-metode transisi IPv6 ke IPv4.....	11
2.4.1 <i>Translation</i> .....	11
2.4.2 <i>Dual-Stack</i> .....	11
2.4.3 <i>Tunneling</i> .....	12
2.4.3.1 <i>Manually Configured IPv6 Tunneling</i> .....	13
2.4.3.2 <i>Automatic 6to4 Tunneling</i> .....	14
2.5 <i>Video Streaming</i> .....	16
<b>3. TOPOLOGI JARINGAN DAN METODE PENGAMBILAN DATA .</b>	<b>18</b>
3.1 Perangkat Jaringan.....	18
3.2 Topologi Jaringan <i>Test-bed</i> .....	21
3.2.1 Topologi Jaringan IPv4 Murni.....	22

3.2.2	Topologi Jaringan Ipv6 Murni.....	22
3.2.3	Topologi Jaringan <i>Automatic 6to4 Tunneling</i> .....	23
3.2.4	Topologi Jaringan <i>Manually Configured Ipv6 Tunneling</i> .....	24
3.3	Metode Pengambilan Data.....	25
<b>4.</b>	<b>ANALISA PERBANDINGAN PERFORMA JARINGAN.....</b>	<b>28</b>
4.1	Dokumentasi Jaringan.....	21
4.1.1	Jaringan IPv4 Murni.....	28
4.1.2	Jaringan IPv6 Murni.....	29
4.1.3	Jaringan <i>Automatic 6to4 Tunneling</i> .....	31
4.1.4	Jaringan <i>Manually Configured IPv6 Tunneling</i> .....	34
4.2	Dokumentasi <i>Server</i> dan <i>Client</i> .....	35
4.2.1	Dokumentasi <i>Server</i> dan <i>Client</i> VLC.....	35
4.2.2	Dokumentasi <i>Server</i> HELIX dan <i>Client Real player</i> .....	38
4.3	Pengolahan Data.....	41
4.4	Analisa <i>Packet Loss</i> VLC .....	43
4.5	Analisa <i>Packet Loss</i> HELIX .....	45
4.6	Analisa <i>Throughput</i> VLC .....	47
4.7	Analisa <i>Throughput</i> HELIX .....	48
4.8	Perbandingan Rata-rata Performa Jaringan.....	49
<b>5.</b>	<b>KESIMPULAN .....</b>	<b>52</b>
	DAFTAR REFERENSI.....	53

## DAFTAR GAMBAR

Gambar 2.1	Perbandingan <i>Header IPv4 dan IPv6</i> .....	7
Gambar 2.2	<i>Manually Configured IPv6 Tunneling</i> .....	13
Gambar 2.3	Format Alamat <i>6to4</i> .....	14
Gambar 2.4	<i>Automatic 6to4 Tunneling</i> .....	15
Gambar 3.1	<i>Router CISCO 3725 Series</i> .....	18
Gambar 3.2	<i>Router CISCO 3845 Series</i> .....	19
Gambar 3.3	<i>Catalyst CISCO 3550 Series</i> .....	19
Gambar 3.4	<i>VLC Player</i> .....	20
Gambar 3.5	<i>Real Player</i> .....	20
Gambar 3.6	<i>Wireshark</i> .....	21
Gambar 3.7	Topologi Umum .....	21
Gambar 3.8	Topologi Jaringan IPv4 Murni.....	22
Gambar 3.9	Topologi Jaringan IPv6 Murni.....	23
Gambar 3.10	Topologi Jaringan <i>Automatic 6to4 Tunneling</i> .....	24
Gambar 3.11	Topologi <i>Manually Configured IPv6 Tunneling</i> .....	25
Gambar 4.1	<i>Router1 IPv4</i> .....	28
Gambar 4.2	<i>Router2 IPv4</i> .....	28
Gambar 4.3	<i>Client IPv4</i> .....	29
Gambar 4.4	<i>Server IPv4</i> .....	29
Gambar 4.5	<i>Router1 IPv6</i> .....	30
Gambar 4.6	<i>Router2 IPv6</i> .....	30
Gambar 4.7	<i>Client IPv6</i> .....	31
Gambar 4.8	<i>Server IPv6</i> .....	31
Gambar 4.9	<i>Router1 6to4</i> .....	32
Gambar 4.10	<i>Router2 6to4</i> .....	32
Gambar 4.11	<i>Client 6to4</i> .....	33
Gambar 4.12	<i>Server 6to4</i> .....	33
Gambar 4.13	<i>Router1 Manual</i> .....	34
Gambar 4.14	<i>Router2 Manual</i> .....	34
Gambar 4.15	Setting <i>Server dan Client VLC Topologi IPv4</i> .....	35
Gambar 4.16	Hasil <i>Packet Sniffing</i> Menggunakan <i>Wireshark</i> pada <i>VLC Client Topologi IPv4</i> .....	36
Gambar 4.17	Setting <i>Server dan Client VLC Topologi IPv6</i> .....	37
Gambar 4.18	Hasil <i>Packet Sniffing</i> Menggunakan <i>Wireshark</i> pada <i>VLC Client Topologi IPv6</i> .....	38
Gambar 4.19	Setting <i>Real Player</i> Untuk Melakukan <i>Streaming</i> .....	39
Gambar 4.20	Hasil <i>Packet Sniffing Real Player</i> Pada topologi IPv4 .....	39
Gambar 4.21	Hasil <i>Packet Sniffing Real Player</i> Pada topologi IPv6 .....	40
Gambar 4.22	Hasil <i>Packet Sniffing Real Player</i> Pada topologi <i>6to4</i> .....	40
Gambar 4.23	Hasil <i>Packet Sniffing Real Player</i> Pada topologi manual .....	40
Gambar 4.24	<i>Summary</i> pada <i>Wireshark</i> .....	42

Gambar 4.25	Grafik Perbandingan <i>Packet Loss</i> VLC .....	44
Gambar 4.26	Grafik Perbandingan <i>Packet Loss</i> HELIX .....	46
Gambar 4.27	Grafik Perbandingan <i>Throughput</i> VLC.....	47
Gambar 4.28	Grafik Perbandingan <i>Throughput</i> HELIX.....	49

## DAFTAR TABEL

Tabel 4.1	Perbandingan Nilai Presentase <i>Packet Loss</i> VLC.....	44
Tabel 4.2	Perbandingan Nilai Presentase <i>Packet Loss</i> HELIX.....	45
Tabel 4.3	Perbandingan Nilai <i>Throughput</i> VLC .....	47
Tabel 4.4	Perbandingan Nilai <i>Throughput</i> HELIX.....	48
Tabel 4.5	Rata-rata <i>Packet Loss</i> .....	50
Tabel 4.6	Rata-rata <i>Throughput</i> .....	50

## DAFTAR SINGKATAN

HTTP	Hiper Text Transfer Protocol
IETF	Internet Engineering Task Force
IPv4	Internet Protocol versi 4
IPv6	Internet Protocol versi 6
MPEG TS	Moving Picture Experts Group Transport Stream
RDT	Real Data Transport
RTP	Real Time Protocol
TCP/IP	Transmission Control Protocol/ Internet Protocol
UDP	Unit Datagram Protocol
VLC	VideoLAN <i>Client</i>

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Saat ini Internet masih menjadi jaringan komputer terbesar di dunia yang dapat menyatukan suatu jaringan komputer dengan jaringan komputer lainnya atau komunikasi antar komputer. Dengan menggunakan kemampuan Internet untuk berkomunikasi, tingkat produktivitas manusia saat ini bertambah dengan pesatnya. Kemampuan Internet untuk berkomunikasi secara universal ini dikarenakan peran protokol yang digunakannya. Suatu protokol pengalamatan bernama *Internet Protocol* (IP) merupakan kunci dari kemampuan Internet yang fleksibel. Versi IP yang masih sering digunakan saat ini adalah *Internet Protocol version 4* (IPv4), sedangkan versi terbaru dari IP, yaitu *Internet Protocol version 6* (IPv6), sudah dikembangkan guna menutupi kekurangan dan keterbatasan yang ada pada IPv4.

Keterbatasan yang ada pada IPv4 salah satunya adalah berupa keterbatasan pengalokasian alamat yang digunakan oleh IPv4. Saat ini alamat yang tersedia tidak lebih dari tiga koma lima milyar alamat, walaupun secara teori alamat yang tersedia seharusnya sebanyak 4,294,967,296 alamat. Angka tiga koma lima milyar bukanlah angka yang cukup untuk memenuhi pengalokasian alamat pada saat ini. Besarnya kebutuhan akan alamat IPv4 membuat jumlah alamat IPv4 yang tersedia semakin menipis. Dari hasil perhitungan *Internet Assigned Numbers Authority* (IANA) memperkirakan bahwa jumlah alamat IPv4 akan habis sekitar tahun 2009 (Jordi Palet, 2007).

Tujuan utama dikembangkannya IPv6 adalah mengantisipasi habisnya alamat IPv4. Berbeda dengan IPv4 yang memiliki panjang alamat 32 bit, IPv6 memiliki panjang alamat 128 bit. Dengan panjang alamat 128 bit, seharusnya alamat IPv6 yang tersedia sebanyak  $2^{128}$  alamat. Dengan jumlah alamat yang  $2^{96}$  kali lebih banyak dari pada jumlah alamat IPv4, IPv6 tentunya dapat menjadi solusi terbatasnya alamat IPv4 saat ini.

Migrasi dari IPv4 ke IPv6 tidak mungkin dapat dilakukan dengan cepat. Diperlukan suatu metode transisi agar kedua versi IP ini bisa berjalan bersamaan. Secara garis besar, metode transisi dibagi menjadi tiga jenis, yaitu *dual-stack*, *translation*, dan *tunneling*. Metode transisi yang paling fleksibel adalah metode transisi *tunneling*, karena memungkinkan *coexistence* antara dua *cloud* IP (IPv4 dan IPv6).

## 1.2. Perumusan Masalah

Permasalahan yang dibahas di dalam skripsi ini yaitu seberapa baik performa dari metode *automatic 6to4 tunneling* dan *manually configured IPv6 tunneling* dalam menjalankan suatu aplikasi *video streaming*.

## 1.3. Tujuan Penulisan

Skripsi ini bertujuan untuk membandingkan performa dari jaringan *automatic 6to4 tunneling* dan jaringan *manually configured IPv6 tunneling* dengan menggunakan aplikasi *video streaming*. Parameter-parameter yang diamati berupa *packet loss* dan *throughput*.

## 1.4. Batasan Masalah

Permasalahan yang akan diuji dibatasi pada penggunaan *test bed* yang akan terfokus pada performa topologi jaringan *automatic 6to4 tunneling* dan topologi jaringan *manually configured IPv6 tunneling* dengan menggunakan aplikasi *video streaming*. *Test bed* juga di konfigurasi ke dalam topologi jaringan IPv4 dan jaringan IPv6 untuk menguji apakah perangkat lunak yang digunakan mendukung pengalamatan IPv4 dan IPv6.

*Test bed* yang digunakan adalah berupa jaringan lokal yang terdiri dari dua buah *laptop*. Satu *laptop* berperan sebagai *server* dan *laptop* yang lain berperan sebagai *client*. Kedua *laptop* itu dihubungkan oleh suatu *intermediary devices* berupa router CISCO 3725 dan 3845.

Aplikasi yang digunakan adalah aplikasi *video streaming* berupa VLC dan Helix streaming server. Parameter-parameter yang dianalisa berupa *packet loss* dan *throughput*.



## 1.5. Sistematika Penulisan

Skripsi ini terdiri atas lima bab dengan sistematika penulisan sebagai berikut :

### **BAB 1 PENDAHULUAN**

Hal-hal yang akan dibahas pada bab pertama ini adalah berupa latar belakang penulisan, tujuan penulisan, pembatasan masalah dan juga sistematika penulisan.

### **BAB 2 PENGENALAN IPV6 DAN METODE TRANSISI *TUNNELING***

Bab kedua ini menjelaskan teori dasar seputar IPv6. Mulai dari struktur IPv6, pengalaman IPv6, metode-metode transisi dari IPv4 ke IPv6 serta perbedaan *tunneling* manual dan *tunneling* otomatis.

### **BAB 3 TOPOLOGI JARINGAN DAN METODE PENGAMBILAN DATA**

Di dalam bab ini terdapat penjelasan mengenai keempat topologi jaringan yang digunakan serta metode-metode yang digunakan untuk pengambilan data.

### **BAB 4 ANALISA PERBANDINGAN PERFORMA JARINGAN**

Membandingkan data hasil pengujian dengan *test-bed* untuk menentukan mana yang lebih baik antara metode *automatic 6to4 tunneling* dengan *manually configured IPv6 tunneling* dalam menjalankan aplikasi *video streaming*.

### **BAB 5 KESIMPULAN**

Bab ini berisi kesimpulan dari keseluruhan isi skripsi

## BAB 2

### PENGENALAN IPV6 DAN METODE TRANSISI TUNNELING

#### 2.1. *Internet Protocol*

*Internet Protocol* adalah suatu aturan standar yang mengatur agar jaringan internet berjalan semestinya. *Internet Protocol* dikembangkan pertama kali oleh *Defense Advanced Research Projects Agency* ( DARPA) pada tahun 1970 sebagai awal dari usaha untuk mengembangkan protokol yang dapat melakukan interkoneksi berbagai jaringan komputer yang terpisah, yang masing-masing jaringan tersebut menggunakan teknologi yang berbeda. Protokol utama yang dihasilkan proyek ini adalah *Internet Protocol* (IP).

IP mempunyai sifat yang independen, baik independen terhadap *network interface*, independen terhadap *physical layer* dan independen terhadap data *link layer*. Pengiriman data pada IP bersifat *connectionless*, yang artinya pengiriman yang dilakukan tidak melalui proses *acknowledgement* (*How IPv6 Works*, 2003). Data yang dikirimkan oleh IP berbentuk *datagram*. Sebuah *datagram* tersusun atas sebuah muatan IP dan sebuah *header* IP. *Header* dari IP mengandung informasi yang dibutuhkan untuk menentukan rute paket, oleh karena itu IP biasa disebut juga sebagai protokol yang *routable*. *Header* IP biasanya tersusun dari dua puluh sampai enam puluh byte, tergantung dari versi IP yang digunakannya. Saat ini, ada dua versi IP yang digunakan, yaitu *Internet Protocol version 4* (IPv4) dan *Internet Protocol version 6* (IPv6). IPv4 pertama kali diperkenalkan pada tahun 1981 oleh DARPA. Sedangkan IPv6 pertama kali diperkenalkan pada tahun 1998. Tujuan utama dikembangkannya IPv6 adalah untuk mengatasi kekurangan-kekurangan yang ada pada IPv4. Selain itu, keterbatasan alamat yang tersedia pada IPv4 juga menjadi faktor dikembangkannya IPv6.

## 2.2. IPv4 dan Keterbatasannya

*Internet Protocol version 4* atau yang biasa disebut dengan IPv4 memiliki panjang total 32-bit yang dibagi menjadi empat buah oktet dalam penulisannya. Tiap oktet berukuran 8-bit dan nilainya berkisar antara 0 sampai 225. Format alamat IPv4 terdiri dari 2 bagian, *netid* dan *hostid*. *Netid* sendiri menyatakan alamat jaringan sedangkan *hostid* menyatakan alamat lokal (*host/router*). Alamat IPv4 dibedakan menjadi lima kelas, yaitu kelas A, kelas B, kelas C, kelas D, dan kelas E. Perbedaan tiap kelasnya terletak pada oktet pertamanya. Pembagian alamat ke dalam lima kelas ini memiliki beberapa tujuan, yaitu:

- Memudahkan pengelolaan dan pengaturan alamat-alamat.
- Memanfaatkan jumlah alamat yang ada secara maksimal.
- Memudahkan pengorganisasian jaringan di seluruh dunia dengan membedakan jaringan tersebut ke dalam kategori besar, menengah, atau kecil.
- Membedakan antara alamat untuk jaringan dan alamat untuk *host/router*.

Secara teoritis, IPv4 dapat mengalami hingga empat milyar lebih *host* di seluruh dunia, yaitu  $2^{32} = 4.294.967.296$  alamat. Karena pertumbuhan jaringan internet yang semakin cepat, kebutuhan akan alamat IP yang digunakan pun semakin meningkat. Pertumbuhan jaringan internet ini telah membuat terlampauinya kapasitas jaringan berbasis IP untuk mensuplai layanan dan fungsi yang diperlukan. Sebuah lingkungan seperti jaringan internet membutuhkan dukungan pada lalu-lintas data secara *real-time* maupun fungsi sekuriti yang memadai. Kebutuhan akan fungsi sekuriti saat ini sangat sulit dipenuhi oleh IPv4. Hal ini mendorong para ahli untuk merumuskan *Internet Protocol* baru untuk menanggulangi keterbatasan *resource Internet Protocol* yang sudah mulai habis serta menciptakan *Internet Protocol* yang memiliki fungsi sekuriti yang memadai.

## 2.3. IPv6

*Internet Protocol version 6*, biasa disebut juga IPv6 pertama kali diperkenalkan pada tanggal 25 Juli tahun 1994 di Toronto pada saat pertemuan *Internet Engineering Task Force* (IETF). Pengembangan IPv6 dilatarbelakangi

oleh keterbatasan IPv4 yang saat ini memiliki panjang 32 bit, akibat ledakan pertumbuhan jaringan. IPv6 atau ada yang menyebutnya dengan nama *Internet Protocol next generation* (IPng), merupakan versi baru dari IP yang merupakan pengembangan dari IPv4.

Perbedaan yang sangat signifikan dari versi pendahulunya terlihat dari panjang alamat yang digunakannya. Jika IPv4 memiliki panjang alamat 32-bit, maka IPv6 memiliki panjang alamat 128-bit. Dengan panjang alamat 128-bit, total alamat yang mungkin tersedia adalah  $2^{128} = 3,4 \times 10^{38}$  alamat. Total alamat yang sangat besar ini bertujuan untuk menyediakan ruang alamat yang tidak akan habis untuk beberapa tahun ke depan, dan membentuk infrastruktur *routing* yang disusun secara rapih, sehingga mengurangi rumitnya proses *routing* dan *tabel routing*. Karena kebutuhan akan suatu *Internet Protocol* baru itulah alasan dikembangkannya IPv6, versi penerus dari IPv4.

### 2.3.1. Struktur IPv6

Suatu paket IPv6 memiliki komponen yang berbeda dari versi pendahulunya, antara lain adalah:

- *Header IPv6*
- *Extension header*

#### 2.3.1.1. *Header IPv6*

*Header IPv6* mempunyai ukuran yang tetap yaitu 40 bytes. *Header* ini merupakan penyederhanaan dari *header IPv4* dengan menghilangkan bagian yang tidak diperlukan atau jarang digunakan dan menambahkan bagian yang menyediakan dukungan yang lebih bagus untuk komunikasi masa depan yang sebagian besar adalah trafik *real-time*. Berikut ini adalah beberapa perbandingan kunci dari *header IPv4* dan IPv6 :

- Jumlah *header field* berkurang dari 12 bytes (termasuk *option*) pada *header IPv4* menjadi 8 bytes pada *header IPv6*.





- Lalu, setiap blok berukuran 16-bit tersebut dikonversikan ke dalam bilangan heksadesimal dan setiap bilangan heksadesimal tersebut dipisahkan dengan menggunakan tanda titik dua. Hasil konversinya adalah sebagai berikut:

2001:0d30:0003:0242:0000:0000:0000:00001

- Untuk penyederhanaan bisa menghilangkan angka nol pada awal setiap blok, yaitu menjadi seperti berikut :

2001:d30:3:242:0:0:0:1

Atau,

2001:d30:3:242::1

Sama seperti pada IPv4, IPv6 juga mempunyai pengalamatan ke dalam *subnet-subnet*. Notasi untuk *subnet* biasanya ditulis ke dalam dalam blok alamat dengan panjang *prefix* tertentu dengan notasi *Classless Inter-Domain Routing* (CIDR). CIDR adalah metode pengalamatan baru yang lebih efisien dalam pengalokasian alamat IP dibandingkan dengan metode pengalamatan lama seperti pembagian kelas A, B, dan C. Contoh alamat IPv6 dengan menggunakan *subnet* adalah sebagai berikut:

2001:d30:3:242::1/56

Pengalamatan pada IPv6 secara umum dibagi menjadi tiga jenis pengalamatan, antara lain:

- Alamat *Unicast*
- Alamat *Anycast*
- Alamat *Multicast*

#### 2.3.2.1 Alamat *Unicast*

Alamat *unicast* adalah suatu identitas dari suatu *interface* tunggal yang sifatnya unik. Menurut fungsinya, alamat *unicast* dapat dibagi lagi menjadi beberapa jenis, yaitu:

- Alamat *unicast global*, mirip dengan alamat pada Ipv4, alamat *unicast global* dapat secara *global* diakses oleh *host-host* yang ada di *internet*. Alamat ini ditandai dengan *prefix* 001. Alamat ini juga sering disebut

sebagai *Aggregatable Global Unicast Address* karena memang jenis alamat ini dirancang agar bisa diagregasi guna menghasilkan pengalamatan yang lebih efisien.

- Alamat *unicast site-local*, alamat ini mirip dengan alamat privat pada Ipv4. Ciri dari jenis alamat ini adalah *prefix* FE80.
- Alamat *unicast link-local*, jenis alamat ini digunakan oleh *host-host* dalam satu *subnet* yang sama atau bisa digunakan juga pada jaringan dengan *link* yang sama. *Router* tidak akan melewati trafik dari alamat-alamat ini keluar *link*. Alamat ini ditandai dengan *prefix* FE80.
- Alamat unicast yang belum ditentukan (*unicast unspecified address*), sesuai dengan namanya, alamat ini memang belum ditentukan oleh administrator karena tidak ada *DHCP server* yang bisa memberikan alamat. Alamat ini bernilai nol semua dalam penulisannya, yaitu 0:0:0:0:0:0:0:0 atau bisa juga ditulis dengan titik dua saja (::)
- Alamat *unicast loopback*, alamat ini mirip dengan alamat 127.0.0.1 pada Ipv4. Pada Ipv6 alamat ini ditulis 0:0:0:0:0:0:0:1 atau bisa disingkat menjadi ::1. Alamat ini tidak boleh dikonfigurasi pada *interface* dan tidak boleh menjadi tujuan *routing* karena alamat ini dipergunakan hanya untuk *Inter-Process Communication* (IPC) dalam sebuah *host*. IPC adalah suatu proses pertukaran data, baik di dalam satu *host* saja maupun antar *host* (dua *host*).
- Alamat *unicast compatibility addresses*, alamat ini dibuat untuk mempermudah proses migrasi dari Ipv4 ke Ipv6. Alamat ini dapat dibagi lagi menjadi beberapa jenis, yaitu alamat *Ipv4-compatible*, alamat *Ipv4-mapped*, Alamat *6over4*, Alamat *6to4* dan alamat ISATAP (S. Deering & R. Hinden, 2003).

#### 2.3.2.2. Alamat *Anycast*

Sama halnya dengan alamat *anycast* pada IPv4, pada IPv6 alamat jenis ini biasanya digunakan oleh *Internet Service Provider* (ISP) untuk melayani klien-kliennya. Alamat *anycast* akan menyampaikan paket ke salah satu dari banyak



penerima, rutenya pun menggunakan rute yang paling dekat atau rute yang terbaik sesuai dengan yang telah dikonfigurasi (*Introduction to IP Version 6*, 2008).

#### 2.3.2.3. Alamat *Multicast*

Jika pada alamat *anycast* pengiriman paket hanya kepada satu penerima saja, pada alamat *multicast* pengirimannya dilakukan ke banyak penerima, biasanya penerima-penerima bergabung dalam suatu grup. Ciri dari alamat ini adalah menggunakan *prefix* FF00 (*Introduction to IP Version 6*, 2008).

### 2.4. Metode-metode Transisi IPv4 ke IPv6

Cepat atau lambat seluruh dunia akan beralih ke IPv6 dan meninggalkan IPv4 karena keterbatasan-keterbatasan yang dimiliki IPv4. Dalam masa peralihan ini sangat dibutuhkan suatu metode transisi agar IPv6 dan IPv4 dapat digunakan bersama-sama. Hal ini dikarenakan perangkat-perangkat komputer atau elektronik yang ada di seluruh dunia belum semua mampu menggunakan IPv6 karena memang dirancang untuk menggunakan IPv4 dan tidaklah mungkin untuk mengganti semua perangkat yang sudah ada dengan perangkat yang mampu menggunakan IPv6 (Lorenzo, 2004).

Secara garis besar, metode transisi dari IPv4 ke IPv6 dibagi menjadi tiga metode, yaitu metode *translation*, metode *dual-stack*, dan metode *tunneling*.

#### 2.4.1. *Translation*

Metode *translation* adalah metode yang dapat menerjemahkan paket IPv6 untuk digunakan dalam jaringan IPv4. Penerjemahan paket ini juga berlaku sebaliknya dimana suatu paket IPv4 diterjemahkan untuk digunakan di dalam jaringan IPv6, sehingga komunikasi *host* IPv4 dengan *host* IPv6 dapat dilakukan. Ketentuan-ketentuan mengenai metode ini terdapat pada RFC 2765 dan 2766

#### 2.4.2. *Dual-Stack*

Dengan menggunakan metode *dual-stack*, kedua versi protokol, baik IPv4 maupun IPv6, dapat digunakan bersama-sama di dalam satu perangkat jaringan yang sama. Metode ini memang memiliki beberapa kelebihan seperti

pengaturannya yang mudah karena tidak perlu menkonfigurasi *tunnel*, hanya menggunakan suatu *node* yang mampu menjalankan aplikasi untuk IPv4 dan IPv6. Adapun kelemahan dari metode ini adalah proses *routing* yang lama, karena menggunakan dua *routing table*

#### 2.4.3. Tunneling

Metode ini sering juga disebut sebagai enkapsulasi karena suatu paket IPv6 akan dienkapsulasi dengan menggunakan IPv4 yang kemudian dikirim melewati jaringan IPv4. Di dalam aplikasinya, ada tiga aplikasi dengan menggunakan metode *tunneling* ini, yaitu *router to router*, *host to host*, dan *host to router* [2]. Di dalam aplikasi *router to router*, proses enkapsulasi hanya dilakukan di antara dua buah *router* yang menjembatani dua buah jaringan IPv6. Pada aplikasi *host to host*, proses enkapsulasi dilakukan antara dua buah *host* yang menggunakan jaringan IPv4, pada aplikasi ini tidak dibutuhkan adanya *router*. Pada aplikasi yang terakhir, yaitu aplikasi *host to router*, proses enkapsulasi dilakukan di antara suatu *host* dan sebuah *router*, dimana pada aplikasi ini sebuah *router* menjembatani dua buah jaringan yang berbeda, yaitu jaringan IPv4 dan jaringan IPv6.

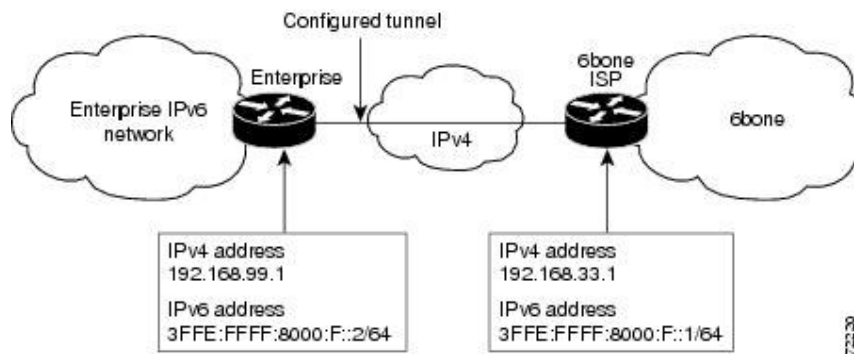
Bila ditinjau dari cara mengkonfigurasinya, *tunneling* dibagi kedalam dua jenis, yaitu *tunneling* manual dan *tunneling* otomatis. Di dalam *tunneling* manual, baik *host* maupun *router*, dikonfigurasi agar dapat berkomunikasi dengan menggunakan *dual-stack node* pada ujung *tunnel*-nya. *Tunneling* manual ini bisa dilakukan antar *host*, antar *router* dan juga komunikasi antara *host* dengan *router*.

Berbeda dengan *tunneling* manual, pada *tunneling* otomatis tidak diperlukan adanya konfigurasi *host* atau *router* secara manual. *Tunneling* otomatis ini menggunakan *dynamic routing* untuk mengirim paket-paket IPv6 dalam suatu jaringan IPv4. Kedua ujung dari *tunnel* ini juga menggunakan *dual-stack node* (Jean-Francois, 2006).

Contoh dari *tunneling* manual adalah IPv6 *manually configured tunneling*, sedangkan contoh dari *tunneling* otomatis adalah *automatic 6to4 tunneling*.

#### 2.4.3.1. Manually Configured IPv6 Tunneling

*Manually Configured IPv6 Tunneling* adalah metode *tunneling* dimana semua pengalamatannya dikonfigurasi secara manual. *Tunnel interface* menggunakan alamat IPv6 dalam konfigurasinya, sedangkan *tunnel source* dan *tunnel destination* menggunakan alamat IPv4 dalam konfigurasinya. Dalam konfigurasinya, metode ini memerlukan lebih banyak waktu ketimbang metode *tunneling* otomatis karena harus menentukan alamat sumber dan alamat tujuan (*source address* dan *destination address*) terlebih dahulu (*IPv6: Connecting*, 2003).



Gambar 2.2. Manually Configured IPv6 Tunneling

(sumber: <http://www.cisco.com>)

Gambar 2.2 adalah contoh implementasi dari metode *tunneling* ini. Cara kerja dari metode *tunneling* ini adalah dengan melewati semua trafik IPv6 (trafik antara IPv6 *host* pada *Enterprise IPv6 network* dengan 6bone) melewati *cloud* IPv4. Kedua *border router* (*Enterprise router* dan 6bone ISP *router*) pada metode *tunneling* ini harus mendukung *dual-stack addressing*.

Keuntungan dari menggunakan metode tunneling jenis ini adalah:

- didukung oleh Cisco IOS *Software*
- menyediakan koneksi yang aman dan *private* antara *host* dan ISP
- memiliki statistik yang unik pada semua *tunnel traffic*

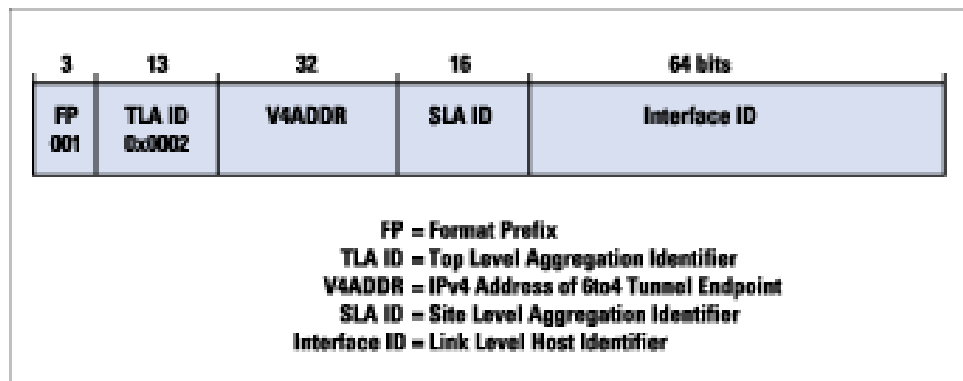
- dapat menkonfigurasi banyak *tunnel* pada ISP yang berbeda dengan menggunakan *multihoming* dengan tujuan meningkatkan reliabilitas jaringan (*IPv6: Connecting*, 2003).

Kekurangan dari metode tunneling jenis ini adalah:

- Metode tunneling jenis ini hanya mendukung koneksi *point-to-point*, sedangkan untuk koneksi lebih dari satu *point* harus menambahkan *tunnel* baru.
- Memerlukan waktu yang lama untuk menkonfigurasi *router* dan komponen jaringan lainnya bila menggunakan banyak *tunnel*.
- Tidak diperbolehkan untuk menkonfigurasi *Network Address Translation* (NAT) secara independen (*IPv6: Connecting*, 2003).

#### 2.4.3.2. Automatic 6to4 Tunneling

Dasar pengalaman metode tunneling 6to4 menggunakan *address prefix*, yaitu hanya menginformasikan satu alamat unik IPv6 pada *node* yang sekurang-kurangnya memiliki satu alamat global IPv4. Pada jenis *tunneling* ini tidak melalui proses enkapsulasi seluruh paket IPv6 ke dalam format IPv4, sehingga *router* tidak terbebani oleh tambahan informasi bila melalui proses enkapsulasi seluruh paket. gambar dari format alamat 6to4 dapat dilihat pada Gambar 2.3.

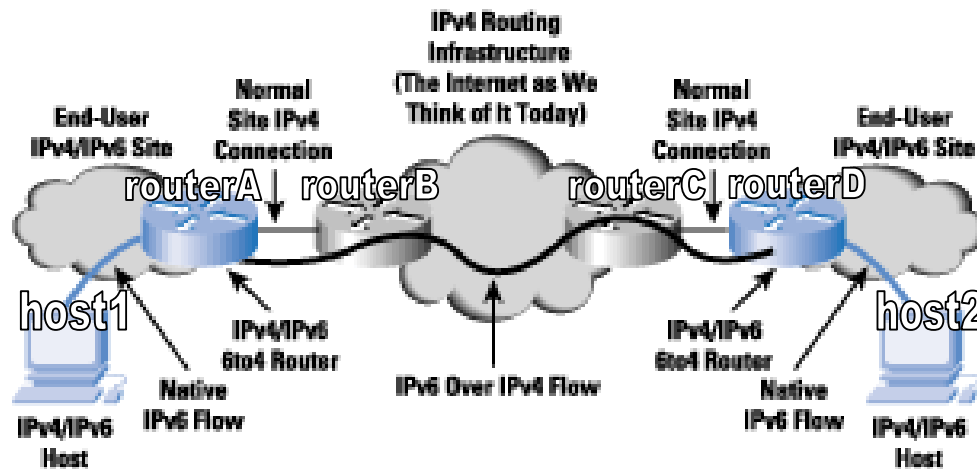


Gambar 2.3. Format Alamat 6to4

(sumber: <http://www.cisco.com>)

Alamat yang digunakan oleh jenis *tunneling* ini memiliki ukuran 128-bit, sama seperti alamat IPv6 murni. Perbedaan antara alamat *6to4* dengan alamat IPv6 murni terletak pada *bit* ke tujuh belas. *Bit* ke tujuh belas pada *6to4* berisi satu buah alamat IPv4. Ciri dari jenis alamat ini adalah menggunakan *prefix* 2002.

Berikut ini adalah contoh konfigurasinya:



Gambar 2.4. Automatic 6to4 Tunneling

(sumber: <http://www.cisco.com>)

Contoh konfigurasi pada gambar 2.4. menggunakan empat buah *router* dan dua buah *host*. RouterA dan routerD adalah *router* yang mendukung fitur 6to4, sedangkan routerB dan routerC adalah *router* biasa untuk mengirimkan paket IPv4 murni. Untuk mengirimkan sebuah paket dari host1 menuju host2 harus melewati empat buah *router*. Dari host1 menuju routerA paket dikirim dengan menggunakan pengalamatan IPv6 murni. Sesampainya di routerA, paket mengalami proses penambahan *address prefix*. *Address prefix* ini membuat paket dialamatkan sebagai IPv4. Lalu dari routerA menuju routerD (melewati routerB dan routerC) paket dikirim dengan menggunakan pengalamatan IPv4 murni. Sesampainya di routerD, *address prefix* akan dibaca oleh router dan dikenali sebagai alamat IPv6 lagi. Setelah itu paket dikirim menuju host2 dengan menggunakan pengalamatan IPv6 murni.

Keuntungan dari menggunakan metode tunneling jenis ini adalah:

- Didukung oleh Cisco IOS *Software*.
- Konfigurasi *host* yang sederhana.
- Tidak diperlukan adanya konfigurasi tambahan pada jaringan bila menambah *host* baru.
- Menggunakan *dynamic IP addressing* pada sisi *host*.
- Tunnel hanya terbentuk pada satu sesi durasi (*IPv6: Connecting*, 2003).

Kekurangan dari menggunakan tunneling jenis ini adalah:

- Tidak dapat menkonfigurasi NAT secara independen.
- Tidak dapat mengimplementasikan *multihoming* dengan mudah.
- Pengalamatan *host* terbatas karena harus mengikuti format pengalamatan 6to4.
- Tidak dapat menggunakan *adaptive routing* (*IPv6: Connecting*, 2003).

## 2.5. Video Streaming

Pengertian dari *video streaming* adalah suatu proses pengiriman video melalui internet dari suatu *server* ke *client* dimana pengiriman video tersebut dilakukan secara *frame-by-frame* dan bersifat *real-time* sehingga video tersebut tidak perlu di-*download* oleh pihak *client*.

Oleh karena sifat *real-time* yang dimiliki oleh suatu aplikasi *video streaming*, maka aplikasi ini memerlukan performa jaringan yang dapat menunjang kelancaran dari aplikasi ini. Performa jaringan yang dimaksud adalah kestabilan jaringan dengan sedikitnya *packet loss* dan *throughput* yang semakin besar. Semakin besarnya *packet loss* akan mengakibatkan semakin banyaknya jumlah *frame* yang hilang karena *packet* yang dikirimkan dalam suatu proses *video streaming* adalah dalam bentuk *frame*. Semakin banyaknya *frame* yang hilang akan berimbas pada semakin buruknya kualitas gambar yang diterima oleh pihak *client* sehingga akan mengurangi kenyamanan *client*.

Protokol yang digunakan oleh suatu *video streaming* berbeda dengan protokol dalam pengiriman dokumen. Pada *video streaming* tidak menggunakan

*File Transfer Protocol* (FTP) karena cara kerja dari protokol tersebut adalah dengan memecah *file* menjadi suatu paket-paket kecil lalu mengirimkannya secara acak setelah itu baru mengaturnya kembali saat semua paket sudah terkirim. Di dalam *video streaming* menggunakan protokol yang bersifat *real-time* yaitu diantaranya adalah *Real-time Transfer Protocol* (RTP), *Real-time Streaming Protocol* (RTSP) dan *Real-time Transport Control Protocol* (RTCP). Cara kerja dari protokol *real-time* tersebut adalah dengan memecah *file* ke dalam pecahan *file* yang sangat kecil lalu mengirimkannya sesuai urutan tertentu (Wilson, n.d.). Selain juga ada protokol *real-time* lain yang hanya digunakan pada perangkat lunak tertentu semisal *Real Data Transport* (RDT) yang digunakan oleh aplikasi yang dikembangkan oleh Realnetworks.

## BAB 3

### TOPOLOGI JARINGAN DAN METODE PENGAMBILAN DATA

#### 3.1. Perangkat Jaringan

Jaringan *test-bed* yang digunakan terdiri dari empat macam topologi, adapun secara umum kesemua topologi menggunakan perangkat keras dan perangkat lunak yang sama.

Perangkat keras yang digunakan adalah sebagai berikut:

- *Laptop* dengan spesifikasi: Intel Centrino Mobile CPU 1.7GHz, 2 Gbytes RAM, Ethernet 10/100 Mbps, menggunakan sistem operasi Windows server 2003. *Laptop* ini digunakan sebagai *Server*.
- *Laptop* dengan spesifikasi: Intel Centrino Mobile CPU 1.7GHz, 2 Gbytes RAM, Ethernet 10/100 Mbps, menggunakan sistem operasi Windows XP service pack 2. *Laptop* ini digunakan sebagai *Client*.
- *Router* CISCO 3725 dengan spesifikasi: versi IOS 12.2(15)ZJ3, total RAM 262144 bytes, dan *Flash memory* 38862848 bytes. *Router* ini digunakan sebagai *Router1* di dalam topologi. Gambar dari *router* ini bisa dilihat pada Gambar 3.1.



Gambar 3.1. *Router* CISCO 3725 Series

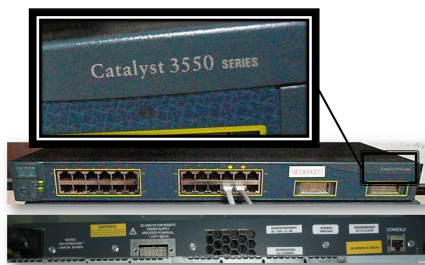
- *Router* CISCO 3845 dengan spesifikasi: versi IOS 12.3(11r)T2, total RAM 1048576 bytes, dan *Flash memory* 262144 bytes. *Router* ini digunakan sebagai *router2* di dalam topologi. Gambar dari *router* ini bisa dilihat pada Gambar 3.2.





Gambar 3.2. Router CISCO 3845 Series

- *Switch CISCO Catalyst 3550*, satu-satunya *switch* di dalam topologi. Gambar dari *switch* ini bisa dilihat pada Gambar 3.3.

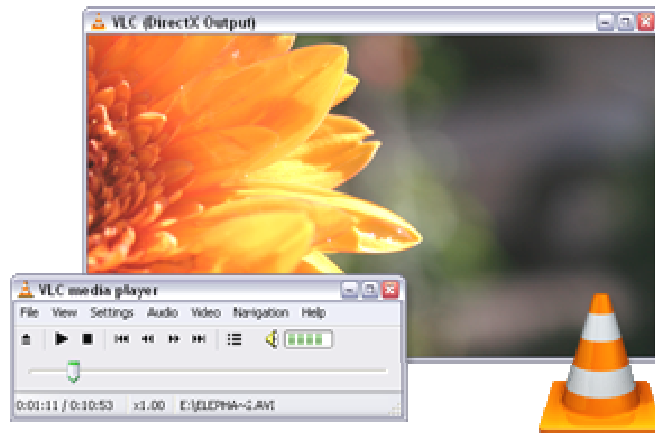


Gambar 3.3. Catalyst CISCO 3550 Series

- Kabel UTP yang dikonfigurasi dalam bentuk *cross* dan *straight*. Kabel dengan konfigurasi *cross* digunakan untuk menghubungkan *router1* ke *router2* dan *router2* ke *client*. Kabel dengan konfigurasi *straight* digunakan untuk menghubungkan *router1* ke *switch* dan *switch* ke *server*.

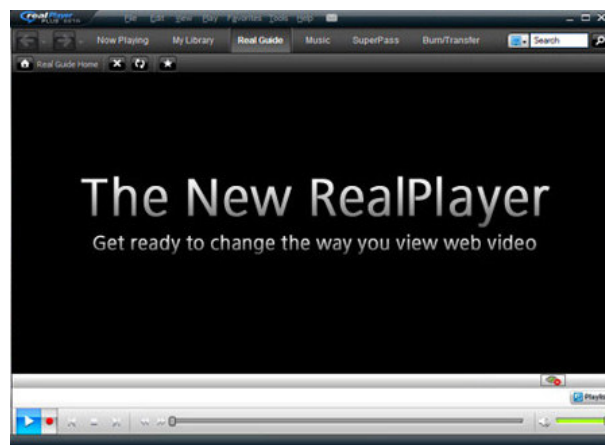
Perangkat lunak yang digunakan adalah sebagai berikut:

- VideoLAN *Client* (VLC), perangkat lunak ini bisa didapat dengan gratis karena status *freeware* yang dimilikinya. Fungsi dari perangkat lunak ini adalah memutar *file video* baik pada sisi *server* maupun pada sisi *client*. Perangkat lunak ini dapat memutar berbagai macam *video codec* dan juga dapat mengatur besar *bit rate* dan *frame rate* sesuai dengan yang diinginkan. Perangkat lunak ini bisa dijalankan pada IPv4 dan IPv6. Gambar 3.4 merupakan tampilan dari VLC *player*.



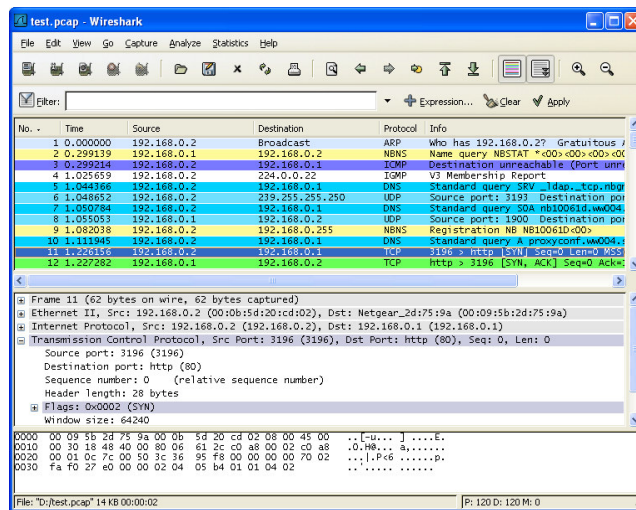
Gambar 3.4. VLC Player

- *HELIX Streaming Server*, perangkat lunak ini berfungsi sebagai *server streaming* yang dijalankan di dalam sistem operasi *Windows Server 2003*. Untuk dapat menggunakan perangkat lunak ini dengan semua fitur yang dimilikinya maka harus membeli *license* terlebih dahulu, namun *Helix Streaming Server* dengan fitur terbatas dapat digunakan dengan gratis.
- *Helix Mobile Producer*, perangkat lunak yang digunakan untuk melakukan proses *encoding* agar file video menjadi bervariasi dalam hal format sehingga bisa dijalankan oleh *Helix Streaming Server*.
- *Real Player*, perangkat lunak yang dijalankan pada sisi *client* sebagai pemutar *file video* dalam berbagai format *video*, dapat dijalankan pada IPv4 maupun IPv6. Perangkat lunak ini dikembangkan oleh *Realnetworks* sehingga menggunakan protokol *RDT* dalam menjalankan proses video streaming. Gambar 3.5 merupakan tampilan dari *Real player*.



Gambar 3.5. Real Player

- Wireshark, suatu perangkat lunak yang digunakan untuk memonitor suatu jaringan. Gambar 3.6 merupakan tampilan dari wireshark.

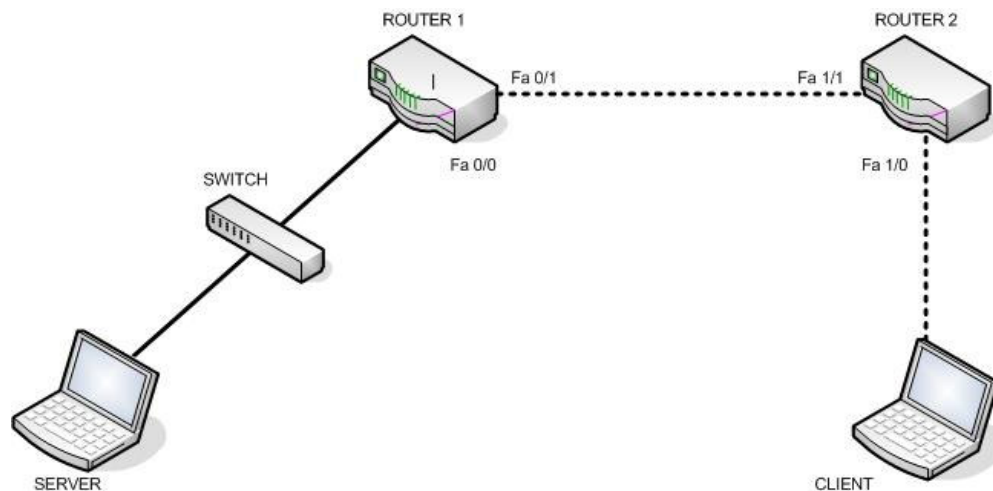


Gambar 3.6. Wireshark

- Putty, perangkat lunak yang digunakan untuk menkonfigurasi *router*.

### 3.2. Topologi Jaringan *Test-bed*

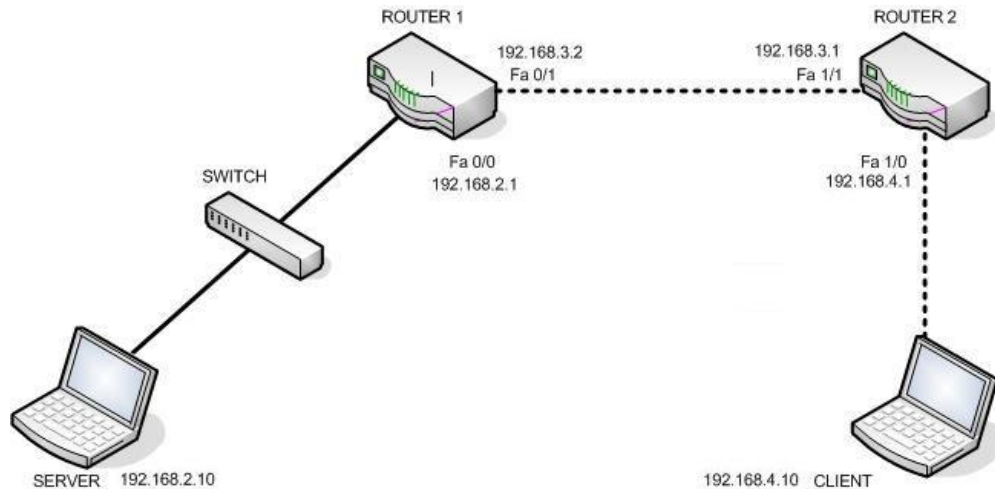
Jaringan *test-bed* yang digunakan terdiri dari empat macam topologi, yaitu topologi jaringan IPv4 murni, topologi jaringan IPv6 murni, topologi jaringan *automatic 6to4 tunneling* dan topologi jaringan *manually configured IPv6 tunneling*. Gambar 3.7 adalah gambar topologi secara umum yang dapat mewakili kesemua topologi yang ada.



Gambar 3.7. Topologi Umum

### 3.2.1 Topologi Jaringan IPv4 Murni

Pada jenis topologi ini ke semua komponen jaringan, baik *router* maupun *laptop*, diberikan alamat IPv4 saja. Pemberian alamat ini dilakukan secara manual. Pemberian alamat *router* dan konfigurasi jaringan dilakukan di dalam *Hyper Terminal*. Gambar topologi ini dapat dilihat pada Gambar 3.8.

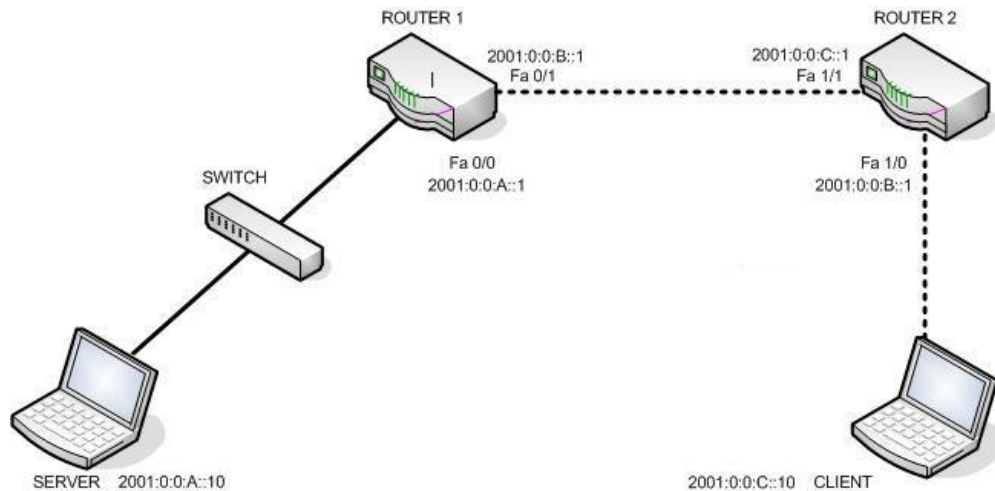


Gambar 3.8. Topologi IPv4 Murni

Pengalamatan dapat dilihat pada gambar di atas, sementara konfigurasi jaringannya dapat dilihat pada lampiran 1. *Routing protocol* yang digunakan pada jaringan ini adalah *routing protocol RIP version 2* karena dapat melakukan *classes routing*. Uji coba perangkat lunak dilakukan pada jenis topologi ini, seperti cara mengkonfigurasi *VLC player*, *helix streaming server*, dan *real player*. Setelah perangkat lunak dapat berjalan pada topologi ini, baru selanjutnya dicoba pada Jaringan IPv6 murni.

### 3.2.2 Topologi Jaringan IPv6 Murni

Pada jenis topologi ini semua komponen jaringannya menggunakan alamat IPv6 saja, tetapi tidak menonaktifkan pengalamatan IPv4 pada server dan client karena perangkat lunak yang digunakan masih menggunakan beberapa fitur IPv4 yang merupakan fitur *default* dari perangkat lunak tersebut. *Routing protocol* yang digunakan pada topologi ini adalah RIPng. Gambar beserta pengalamatan topologi ini dapat dilihat pada Gambar 3.9.

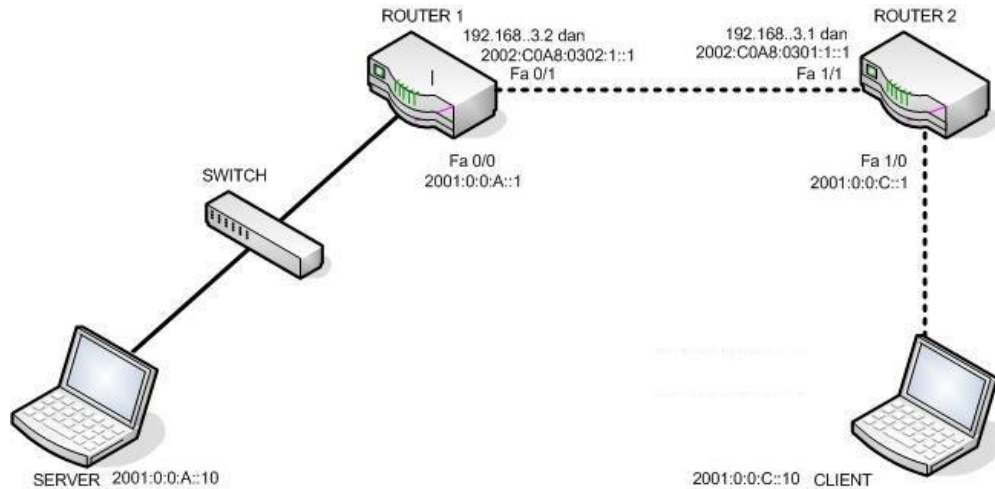


Gambar 3.9. Topologi IPv6 Murni

Pengujian apakah perangkat lunak yang digunakan mendukung fitur IPv6 atau tidak dilakukan pada jenis topologi ini. Bila ternyata semua perangkat lunak dapat berjalan pada jenis topologi ini, maka secara teori perangkat lunak tersebut pun dapat berjalan pada dua topologi berikutnya, yaitu topologi dengan menggunakan *tunneling*. Konfigurasi pengalaman mengenai topologi ini dapat dilihat pada Lampiran2.

### 3.2.3 Topologi Jaringan *Automatic 6to4 Tunneling*

Semua topologi memiliki perangkat keras yang sama dengan penempatan yang sama, perbedaan hanya terdapat pada konfigurasi *router* saja. Pada jenis topologi ini *router* dikonfigurasi menjadi *router 6to4*, dan diberi pengalamatan IPv4 dan IPv6. Sementara itu kedua *laptop*, baik *server* maupun *client* hanya diberi pengalamatan IPv6 saja. Gambar topologi serta pembagian alamatnya dapat dilihat pada Gambar 3.10.

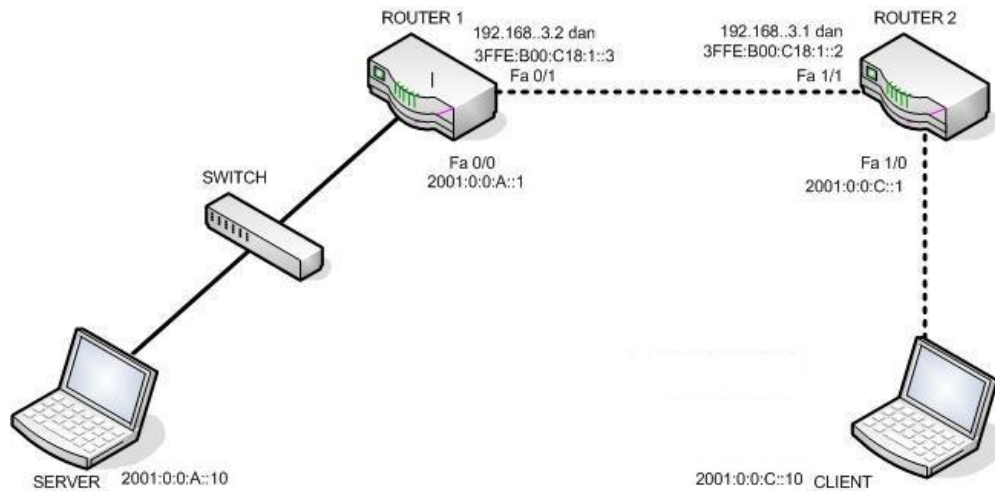


Gambar 3.10 Topologi *Automatic 6to4 Tunneling*

Alamat IPv6 dari Fa 0/1 dan Fa 1/1 adalah alamat hasil konversi dari alamat IPv4, dimana alamat IPv4 dikonversi kedalam bentuk heksadesimal lalu diberi *header* 2002 yang merupakan format *header* yang digunakan pada metode *tunneling 6to4*. *Routing protocol* yang digunakan adalah *routing protocol RIP version 2*. Proses enkapsulasi terjadi di dalam *router1* dan proses *tunneling* terjadi di sepanjang *router1* dan *router2*.

### 3.2.4 Topologi Jaringan *Manually Configured IPv6 Tunneling*

Pada jenis topologi ini pengalamatan pada *router* juga menggunakan dua jenis alamat baik alamat IPv4 maupun alamat IPv6. Perbedaannya dengan *automatic 6to4 tunneling* adalah pada *tunneling* jenis ini tidak ada proses konversi dan penyisipan alamat IPv4.



Gambar 3.11 Topologi *Manually Configured IPv6 Tunneling*

Untuk dapat mengetahui konfigurasi dan pengalaman yang lebih jelas dapat dilihat pada lampiran 4. Pada topologi ini juga menggunakan *routing protocol RIP version 2*.

### 3.3 Metode Pengambilan Data

Pengambilan data dilakukan dalam empat macam topologi jaringan dan dua cara pengiriman video. Topologi jaringan tersebut adalah topologi jaringan IPv4, topologi jaringan IPv6, topologi jaringan *automatic 6to4 tunneling* dan topologi jaringan *manually configured IPv6 tunneling*. Cara pengiriman video yang pertama adalah dengan menggunakan VLC pada sisi *server* sebagai *streaming server* maupun pada sisi *client* sebagai *streaming client*. Cara ini menggunakan dua jenis video dengan format yang berbeda tapi memiliki durasi yang sama, yaitu:

1. TestSkripsi.mpg dengan durasi 60 detik dan mempunyai ukuran file sebesar 2,114 KB
2. TestSkripsi.mp4 dengan durasi 60 detik dan mempunyai ukuran file sebesar 4,216 KB

Perbedaan format dimaksudkan untuk dapat melihat perbedaan kinerja jaringan terhadap proses *encoding* yang berbeda pula. Tiap-tiap video tersebut nantinya akan melewati proses *encoding* ke dalam tiga macam variasi *video bit rate* dan

*audio bit rate* dengan menggunakan VLC. Dengan menggunakan ketiga macam variasi tersebut, maka keseluruhan video yang digunakan untuk proses *streaming* dengan menggunakan VLC adalah sebagai berikut:

1. TestSkripsi.mpg dengan menggunakan *video bit rate* sebesar 256 kb/s dan *audio bit rate* sebesar 32 kb/s.
2. TestSkripsi.mpg dengan menggunakan *video bit rate* sebesar 512 kb/s dan *audio bit rate* sebesar 96 kb/s.
3. TestSkripsi.mpg dengan menggunakan *video bit rate* sebesar 1024 kb/s dan *audio bit rate* sebesar 192 kb/s.
4. TestSkripsi.mp4 dengan menggunakan *video bit rate* sebesar 256 kb/s dan *audio bit rate* sebesar 32 kb/s.
5. TestSkripsi.mp4 dengan menggunakan *video bit rate* sebesar 512 kb/s dan *audio bit rate* sebesar 96 kb/s.
6. TestSkripsi.mp4 dengan menggunakan *video bit rate* sebesar 1024 kb/s dan *audio bit rate* sebesar 192 kb/s.

Cara pengiriman video yang kedua adalah dengan menggunakan Helix *Streaming Server* pada sisi *server* dan *Real Player* pada sisi *client*. Ada sembilan macam *file* video yang digunakan, perbedaan terdapat pada format, *bit rate* dan *frame rate*. Berikut ini adalah video-video yang digunakan:

1. TestSkripsi80-10.mp4, berformat mp4 memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 10 fps.
2. TestSkripsi80-20.mp4, berformat mp4 memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 20 fps.
3. TestSkripsi80-30.mp4, berformat mp4 memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 30 fps.
4. TestSkripsi160-10.mp4, berformat mp4 memiliki *bit rate* 160 kb/s dan *frame rate* sebesar 10 fps.
5. TestSkripsi160-20.mp4, berformat mp4 memiliki *bit rate* 160 kb/s dan *frame rate* sebesar 20 fps.
6. TestSkripsi160-30.mp4, berformat mp4 memiliki *bit rate* 160 kb/s dan *frame rate* sebesar 30 fps.



7. TestSkripsi(mp4)80,10.rm, berformat rm memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 10 fps.
8. TestSkripsi(mp4)80,20.rm, berformat rm memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 20 fps.
9. TestSkripsi(mp4)80,30.rm, berformat rm memiliki *bit rate* 80 kb/s dan *frame rate* sebesar 30 fps.

*File* video yang berformat rm sebenarnya adalah *file* video berformat mp4 yang sudah melewati proses *encoding* dengan menggunakan *Helix Mobile Producer*.

Masing-masing *file* video diatas dimainkan sebanyak lima kali tiap topologi, sehingga akan menghasilkan lima data tiap *file* video tiap topologi jaringannya. Wireshark akan merekam semua aktivitas jaringan termasuk jumlah *packet* yang keluar atau masuk dan besarnya *throughput*. Pengambilan data dilakukan dengan melihat fitur *summary* yang ada pada Wireshark. Parameter *packet loss* didapat dari selisih jumlah *packet* yang dikirimkan *server* dengan jumlah *packet* yang diterima *client* setelah itu barulah dibagi dengan jumlah *packet* yang dikirimkan *server* untuk mendapatkan nilai persentasenya. Parameter *throughput* yang didapat dengan melihat *summary* pada wireshark menggunakan satuan MBit/sec.

Secara keseluruhan hanya ada tiga format video yang digunakan, yaitu mp4, mpg dan rm. Mp4 atau biasa disebut MPEG-4 adalah generasi terbaru dari MPEG (*Moving Pictures Expert Group*) dimana pada versi terbarunya ini menghasilkan kompresi video yang lebih efektif, sehingga dengan ukuran *file* yang relatif kecil dapat menghasilkan kualitas video yang lebih bagus dari versi sebelumnya, MPEG-1 atau biasa disebut mpg, yang juga digunakan di dalam skripsi ini. Sedangkan format yang terakhir yaitu rm (*Real Media*), adalah suatu format multimedia yang dibuat oleh *RealNetworks* yang hanya dapat diputar dengan menggunakan *Real Player*. Alasan digunakannya format-format tersebut adalah karena perangkat lunak yang digunakan mendukung format-format tersebut.

## BAB 4

### ANALISA PERBANDINGAN PERFORMA JARINGAN

#### 4.1. Dokumentasi Jaringan

##### 4.1.1. Jaringan IPv4 Murni

Gambar 4.1 dan Gambar 4.2 adalah hasil dokumentasi dari *router1* dan *router2* dengan menggunakan *command show ip route* pada Putty.

```
router1#
router1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    192.168.4.0/24 [120/1] via 192.168.3.1, 00:00:03, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
C    192.168.3.0/24 is directly connected, FastEthernet0/1
```

Gambar 4.1. Router1 IPv4

```
Router#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS lev
       ia - IS-IS inter area, * - candidate default, U - per-user static
       te
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.4.0/24 is directly connected, FastEthernet1/0
R    192.168.2.0/24 [120/1] via 192.168.3.2, 00:00:06, FastEthernet1/1
C    192.168.3.0/24 is directly connected, FastEthernet1/1
```

Gambar 4.2. Router2 IPv4

Terlihat bahwa rute yang akan dilewati paket melalui *gateway* 192.168.2.0, 192.168.3.0 dan 192.168.4.0 dimana *router1* dan *router2* terhubung melewati *interface FastEthernet0/1* dan *FastEthernet1/1*.

Dokumentasi berikutnya adalah pada sisi *server* dan *client*. Gambar 4.3 dan Gambar 4.4 adalah hasil dokumentasi dengan menggunakan *command tracert* pada *command prompt*.

```
C:\Documents and Settings\tiar>tracert 192.168.2.10
Tracing route to 192.168.2.10 over a maximum of 30 hops
  1  <1 ms    <1 ms    <1 ms    192.168.4.1
  2  <1 ms    <1 ms    <1 ms    192.168.3.2
  3  <1 ms    <1 ms    <1 ms    192.168.2.10
Trace complete.
```

Gambar 4.3. *Client IPv4*

```
D:\Documents and Settings\Administrator>tracert 192.168.4.10
Tracing route to PRESARIO-6F6832 [192.168.4.10]
over a maximum of 30 hops:
  1  <1 ms    <1 ms    <1 ms    192.168.2.1
  2  <1 ms    <1 ms    <1 ms    192.168.3.1
  3  <1 ms    <1 ms    <1 ms    PRESARIO-6F6832 [192.168.4.10]
Trace complete.
```

Gambar 4.4. *Server IPv4*

Alamat *router1* dapat dilihat pada gambar di atas yaitu 192.168.3.2 dan alamat *router2* yaitu 192.168.3.1, keduanya terhubung dengan *network* 192.168.3.0.

#### 4.1.2. Jaringan IPv6 Murni

Dokumentasi pada jenis jaringan ini hampir sama dengan pada jaringan IPv4 murni, hanya saja alamat yang digunakan adalah alamat IPv6. Perbedaan lainnya adalah pada *routing protocol* yang digunakannya, jenis jaringan ini menggunakan *routing protocol* RIPng. Gambar 4.5. dan gambar 4.6. merupakan hasil dokumentasi topologi jaringan IPv6 murni dengan menggunakan *command show ipv6 route* pada Putty.

```

router1#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B -
BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1,
OE2 - OSPF ext 2
C 2001:0:0:A::/64 [0/0]
  via ::, FastEthernet0/0
L 2001:0:0:A::1/128 [0/0]
  via ::, FastEthernet0/0
C 2001:0:0:B::/64 [0/0]
  via ::, FastEthernet0/1
L 2001:0:0:B::2/128 [0/0]
  via ::, FastEthernet0/1
R 2001:0:0:C::/64 [120/2]
  via FE80::217:5AFF:FE40:A082, FastEthernet0/1
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0

```

Gambar 4.5. Router1 IPv6

```

Router#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B -
BGP
        U - Per-user Static route
        I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS
- ISIS summary
        O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1,
OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R 2001:0:0:A::/64 [120/2]
  via FE80::20F:23FF:FE02:2541, FastEthernet1/1
C 2001:0:0:B::/64 [0/0]
  via ::, FastEthernet1/1
L 2001:0:0:B::1/128 [0/0]
  via ::, FastEthernet1/1
C 2001:0:0:C::/64 [0/0]
  via ::, FastEthernet1/0
L 2001:0:0:C::1/128 [0/0]
  via ::, FastEthernet1/0
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0

```

Gambar 4.6. Router2 IPv6

Gambar 4.7 dan Gambar 4.8 adalah dokumentasi dari sisi *server* dan *client* dengan menggunakan *command tracert* pada *command prompt*:

```
C:\Documents and Settings\tiar>tracert 2001:0:0:a::10
Tracing route to 2001:0:0:a::10 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms    2001:0:0:c::1
  2    <1 ms    <1 ms    <1 ms    2001:0:0:b::2
  3    <1 ms    <1 ms    <1 ms    2001:0:0:a::10
Trace complete.
```

Gambar 4.7. *Client IPv6*

```
D:\Documents and Settings\Administrator>tracert 2001:0:0:c::10
Tracing route to 2001:0:0:c::10 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms    2001:0:0:a::1
  2    <1 ms    <1 ms    <1 ms    2001:0:0:b::1
  3    <1 ms    <1 ms    <1 ms    2001:0:0:c::10
Trace complete.
```

Gambar 4.8. *Server IPv6*

*Router1* dengan alamat IP 2001:0:0:B::2 dan *router2* dengan alamat IP 2001:0:0:B::1 terhubung oleh *network* 2001:0:0:A::0 pada *interface FastEthernet1/1* dan *network* 2001:0:0:C::0 pada *interface FastEthernet0/1*.

#### 4.1.3. Jaringan *Automatic 6to4 Tunneling*

Berbeda dengan dua jenis topologi jaringan sebelumnya, yaitu bila pada topologi jaringan IPv4 murni menggunakan *routing protocol* RIP versi2 dan pada topologi jaringan IPv6 murni menggunakan *routing protocol* RIPng, pada jenis topologi jaringan 6to4 ini menggunakan kedua *routing protocol* tersebut. Penggunaan dua jenis *routing protocol* ini dikarenakan penggunaan dua jenis pengalamatan, yaitu pengalamatan IPv4 dan pengalamatan IPv6. Pada bagian pengalamatan IPv4-nya, jaringan ini menggunakan *routing protocol* RIP versi2; sedangkan pada bagian pengalamatan IPv6-nya, jaringan ini menggunakan *routing protocol* RIPng. Gambar 4.9. dan gambar 4.10. merupakan dokumentasi dari topologi jaringan *automatic 6to4 tunneling* yang dilakukan dengan *capture* tampilan pada Putty dengan menggunakan *command show ipv6 route*.

```

router1#show ipv6 route
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
S  ::/0 [1/0]
   via 2002:COA8:301:1::1, Null0
C  2001:0:0:A::/64 [0/0]
   via ::, FastEthernet0/0
L  2001:0:0:A::1/128 [0/0]
   via ::, FastEthernet0/0
S  2002::/16 [1/0]
   via ::, Tunnel0
C  2002:COA8:302:1::/64 [0/0]
   via ::, FastEthernet0/1
L  2002:COA8:302:1::1/128 [0/0]
   via ::, FastEthernet0/1
L  FE80::/10 [0/0]
   via ::, Null0
L  FF00::/8 [0/0]
   via ::, Null0

```

Gambar 4.9. Router1 6to4

```

Router#show ipv6 route
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S  ::/0 [1/0]
   via 2002:COA8:302:1::1
C  2001:0:0:C::/64 [0/0]
   via ::, FastEthernet1/0
L  2001:0:0:C::1/128 [0/0]
   via ::, FastEthernet1/0
S  2002::/16 [1/0]
   via ::, Tunnel0
C  2002:COA8:301:1::/64 [0/0]
   via ::, FastEthernet1/1
L  2002:COA8:301:1::1/128 [0/0]
   via ::, FastEthernet1/1
L  FE80::/10 [0/0]
   via ::, Null0
L  FF00::/8 [0/0]
   via ::, Null0

```

Gambar 4.10. Router2 6to4

Selanjutnya Gambar 4.11 dan Gambar 4.12 dibawah ini adalah dokumentasi yang dilakukan dengan meng-capture tampilan pada *command prompt* dengan menggunakan *command tracert*:

```
C:\Documents and Settings\tiar>tracert 2001:0:0:a::10
Tracing route to 2001:0:0:a::10 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms    2001:0:0:c::1
  2     1 ms    <1 ms    <1 ms    2002:c0a8:302:1::1
  3     1 ms    <1 ms    <1 ms    2001:0:0:a::10
Trace complete.
```

Gambar 4.11. *Client 6to4*

```
D:\Documents and Settings\Administrator>tracert 2001:0:0:c::10
Tracing route to 2001:0:0:c::10 over a maximum of 30 hops
  1    <1 ms    <1 ms    <1 ms    2001:0:0:a::1
  2     1 ms    <1 ms    <1 ms    2002:c0a8:301:1::1
  3     1 ms    <1 ms    <1 ms    2001:0:0:c::10
Trace complete.
```

Gambar 4.12. *Server 6to4*

Terlihat dari dokumentasi di atas bahwa kedua *router*, terhubung pada satu *tunnel* dengan IP dari *router1* adalah 2002:c0a8:301:1::1 dan IP dari *router2* adalah 2002:c0a8:302:1::1. Pada dokumentasi di atas memang sama sekali tidak terlihat penggunaan dari alamat IPv4, namun bukan berarti alamat IPv4 tidak berperan pada jenis jaringan ini. Tidak terlihatnya alamat-alamat IPv4 dikarenakan kecenderungan dari *client* dan *server* yang hanya akan mencari alamat IPv6. Peran dari alamat IPv4 di sini adalah menjadikan *router1* dan *router2* menjadi satu *network* yang sama, dimana pengalamatan IPv6 pada jaringan ini tidak dapat melakukan hal tersebut. Alamat IPv4 yang digunakan oleh *router1* adalah 192.168.3.2 dengan *subnet* 255.255.255.0 dan alamat IPv4 yang digunakan oleh *Router2* adalah 192.168.3.1 dengan *subnet* yang sama dengan *router1*. Hal ini menunjukkan bahwa *router1* dan *router2* sudah mempunyai satu *network* yang sama yaitu *network* 192.168.3.0. Untuk lebih jelasnya bisa dilihat pada Lampiran3.



#### 4.1.4. Jaringan *Manually Configured IPv6 Tunneling*

Sama halnya dengan topologi jaringan 6to4, pada topologi jenis ini menggunakan dua buah jenis *routing protocol*. Pada pengalamtan IPv4-nya menggunakan *routing protocol* RIP versi2 sedangkan pada pengalamatan IPv6-nya menggunakan *routing protocol* RIPng. Gambar 4.13 dan Gambar 4.14 adalah hasil dokumentasinya dengan menggunakan perintah *show ipv6 route* pada Putty:

```
Router#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
S 2001::/16 [1/0]
  via ::, Tunnel0
C 2001:0:0:A::/64 [0/0]
  via ::, FastEthernet0/0
L 2001:0:0:A::1/128 [0/0]
  via ::, FastEthernet0/0
C 3FFE:B00:C18:1::/64 [0/0]
  via ::, Tunnel0
L 3FFE:B00:C18:1::3/128 [0/0]
  via ::, Tunnel0
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0
```

Gambar 4.13. *Router1* Manual

```
Router#show ipv6 route
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S 2001::/16 [1/0]
  via ::, Tunnel0
C 2001:0:0:C::/64 [0/0]
  via ::, FastEthernet1/0
L 2001:0:0:C::1/128 [0/0]
  via ::, FastEthernet1/0
C 3FFE:B00:C18:1::/64 [0/0]
  via ::, Tunnel0
L 3FFE:B00:C18:1::2/128 [0/0]
  via ::, Tunnel0
L FE80::/10 [0/0]
  via ::, Null0
L FF00::/8 [0/0]
  via ::, Null0
```

Gambar 4.14. *Router2* Manual

Terlihat dari hasil dokumentasi di atas bahwa *router1* (2001:0:0:A::/64) dan *Router2* (2001:0:0:C::/64) terhubung melalui *tunnel0* (3FFE:B00:C18:1::/64).

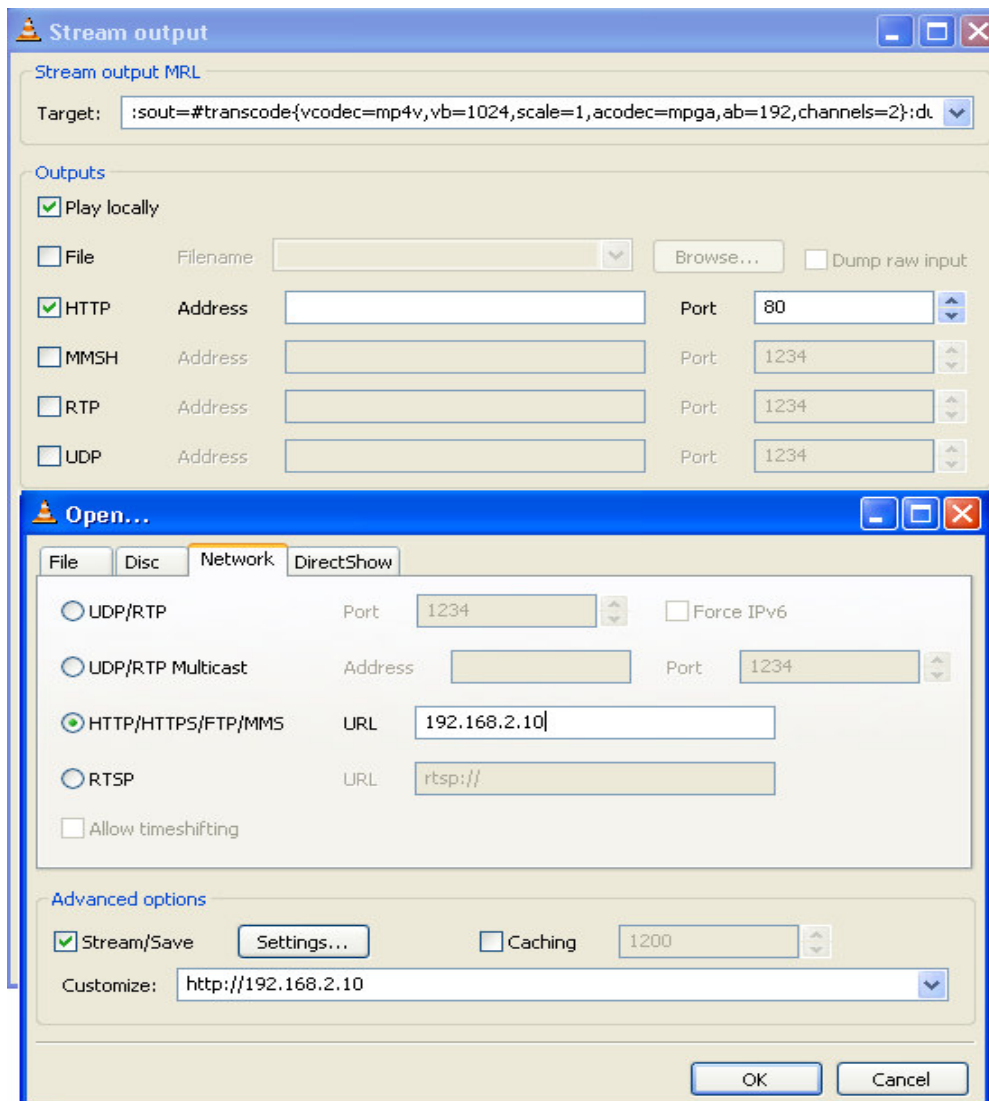


Huruf C pada kiri alamat adalah keterangan yang menunjukkan sudah tersambung nya *interface* yang dimaksud.

## 4.2. Dokumentasi *Server* dan *Client*

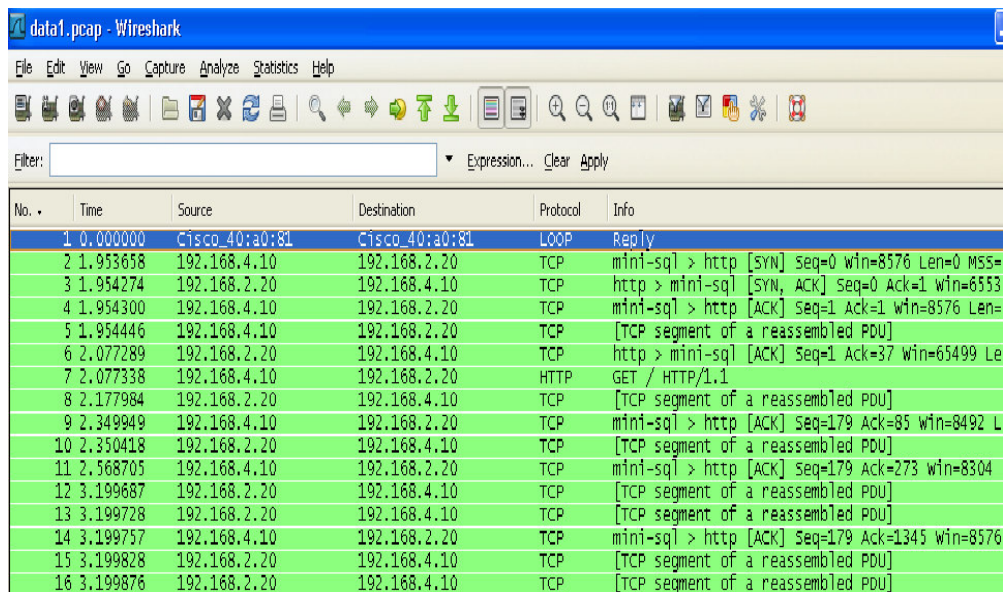
### 4.2.1. Dokumentasi *Server* dan *Client* VLC

Dari hasil percobaan yang dilakukan, perangkat lunak VLC berhasil dijalankan baik dengan menggunakan pengalamatan IPv4 maupun dengan menggunakan pengalamatan IPv6. Gambar 4.15 berikut ini adalah dokumentasi dari penggunaan VLC sebagai *streaming server* dan juga sebagai *client*:



Gambar 4.15. Setting *Server* dan *Client* VLC Topologi IPv4

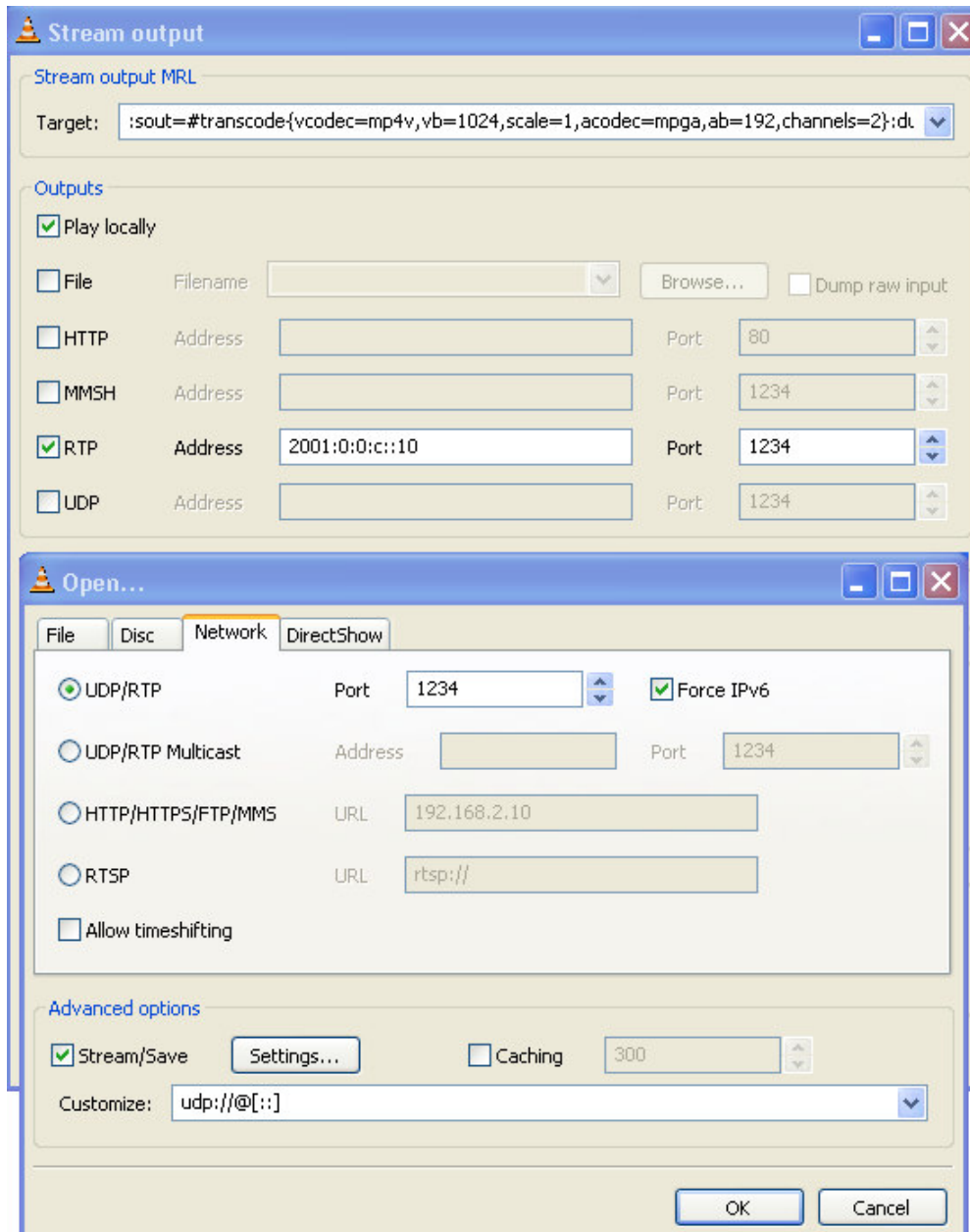
Terlihat ada dua *window* pada Gambar 4.15. *Window* yang atas adalah *window* VLC pada *server* sedangkan *window* yang bawah adalah *window* VLC pada *client*. Pada *window server* terlihat penggunaan *port* 80 yaitu *port* HTTP. Penggunaan *port* HTTP ini dimaksudkan agar *file video* dapat diakses oleh *client* mana saja yang ingin mengakses. Pada *window client* terlihat bahwa *client* sedang memanggil *file video* melalui *port* HTTP dengan menggunakan alamat IP *server* sebagai alamat tujuannya. Protokol yang digunakan adalah protokol TCP yang bisa dilihat dengan menggunakan Wireshark pada Gambar 4.16.



No. ·	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_40:a0:81	Cisco_40:a0:81	LOOP	Reply
2	1.953658	192.168.4.10	192.168.2.20	TCP	mini-sql > http [SYN] Seq=0 win=8576 Len=0 MSS=
3	1.954274	192.168.2.20	192.168.4.10	TCP	http > mini-sql [SYN, ACK] Seq=0 Ack=1 win=6553
4	1.954300	192.168.4.10	192.168.2.20	TCP	mini-sql > http [ACK] Seq=1 Ack=1 win=8576 Len=
5	1.954446	192.168.4.10	192.168.2.20	TCP	[TCP segment of a reassembled PDU]
6	2.077289	192.168.2.20	192.168.4.10	TCP	http > mini-sql [ACK] Seq=1 Ack=37 win=65499 Le
7	2.077338	192.168.4.10	192.168.2.20	HTTP	GET / HTTP/1.1
8	2.177984	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]
9	2.349949	192.168.4.10	192.168.2.20	TCP	mini-sql > http [ACK] Seq=179 Ack=85 win=8492 L
10	2.350418	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]
11	2.568705	192.168.4.10	192.168.2.20	TCP	mini-sql > http [ACK] Seq=179 Ack=273 win=8304
12	3.199687	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]
13	3.199728	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]
14	3.199757	192.168.4.10	192.168.2.20	TCP	mini-sql > http [ACK] Seq=179 Ack=1345 win=8576
15	3.199828	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]
16	3.199876	192.168.2.20	192.168.4.10	TCP	[TCP segment of a reassembled PDU]

Gambar 4.16. Hasil *Packet Sniffing* Menggunakan Wireshark pada VLC *Client* Topologi IPv4

Penggunaan *port* HTTP seperti di atas hanya dapat dilakukan dengan pada jaringan IPv4 murni saja karena fitur HTTP pada VLC belum mendukung pengalamatan IPv6. Untuk menggunakan pengalamatan IPv6 pada VLC dapat digunakan *port* 1234 yaitu *port* RTP. Gambar 4.17 adalah dokumentasi penggunaan *port* RTP yang diterapkan pada jaringan IPv6 murni, jaringan *automatic 6to4 tunneling* dan juga jaringan *manually configured IPv6 tunneling*.



Gambar 4.17. *Setting Server dan Client VLC Topologi IPv6*

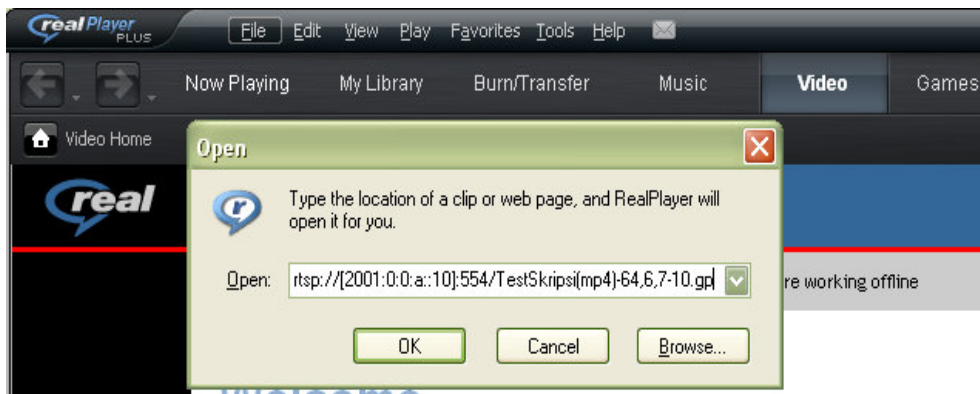
Berbeda dengan pengalaman IPv4 yang menggunakan protokol TCP, pada IPv6 ini menggunakan protokol UDP yang bisa ditunjukkan pada Gambar 4.18.

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_40:a0:81	Cisco_40:a0:81	LOOP	Reply
2	1.540950	169.254.212.133	169.254.255.255	NBNS	Name query NB ::<00>
3	2.286695	169.254.212.133	169.254.255.255	NBNS	Name query NB ::<00>
4	3.036726	169.254.212.133	169.254.255.255	NBNS	Name query NB ::<00>
5	5.683834	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
6	5.710045	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
7	5.736406	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
8	5.762768	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
9	5.790084	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
10	5.816571	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
11	5.842800	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
12	5.869252	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
13	5.894591	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
14	5.919047	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
15	5.943443	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
16	5.967869	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
17	5.992475	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
18	6.016763	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
19	6.040162	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
20	6.064593	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a
21	6.088983	2001:0:0:a::20	2001:0:0:c::10	UDP	Source port: kiosk Destination port: search-a

Gambar 4.18. Hasil *Packet Sniffing* Menggunakan Wireshark pada VLC *Client* Topologi IPv6

#### 4.2.2. Dokumentasi *Server HELIX* dan *Client Real Player*

Dari hasil percobaan yang dilakukan, perangkat lunak *HELIX streaming server* dan *Real Player* berhasil dijalankan baik dengan menggunakan pengalamatan IPv4 maupun dengan menggunakan pengalamatan IPv6. Protocol yang digunakan oleh *HELIX server* dan *real player* ini adalah RTSP atau *Real Time Streaming Protocol*. Perbedaannya dengan *streaming* menggunakan VLC tidak hanya terletak pada protokolnya saja, tetapi juga dengan cara kerjanya. Jika pada *streaming* menggunakan VLC sebuah *file video* dikirim dengan menggunakan sebuah *server* lalu ditampilkan menggunakan VLC *player* pada sisi *client*-nya, pada *streaming* menggunakan *HELIX server* dan *real player* *file video*-nya tidak dikirim oleh *server* melainkan tersimpan pada suatu *folder* di dalam *server* tersebut. *HELIX server* menyimpan *file video* yang ingin digunakan untuk proses *streaming* lalu *real player* berfungsi meminta *file video* tersebut. Gambar 4.19 adalah contoh *real player* yang sedang ingin membuka *file video* yang tersimpan pada *server HELIX*.



Gambar 4.19. Setting Real Player Untuk Melakukan Streaming

Nama *file video* yang ingin dibuka pada contoh di atas adalah “TestSkripsi(mp4)-64,6,7-10.gp” sedangkan alamat *server* yang dituju adalah 2001:0:0:a::10 dengan menggunakan *port* 554 yaitu *port* RTSP. Protokol *streaming* yang digunakan adalah protokol RDT pada semua jenis *file video* maupun semua jenis konfigurasi jaringan baik jaringan IPv4 murni, jaringan IPv6 murni, jaringan *automatic 6to4 tunneling*, maupun jaringan *manually configured IPv6 tunneling* seperti yang terlihat pada Gambar 4.20, Gambar 4.21, Gambar 4.22 dan Gambar 4.23.

No.	Time	Source	Destination	Protocol	Info
677	8.161982	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
678	8.161986	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
679	8.161991	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
680	8.162135	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
681	8.162344	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=00
682	8.162355	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
683	8.162360	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
684	8.211403	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
685	8.211431	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
686	8.211438	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
687	8.211444	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
688	8.271827	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
689	8.271858	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
690	8.271866	192.168.2.10	192.168.4.10	RDT	DATA: stream-id=01
691	8.271872	192.168.2.10	192.168.4.10	RDT	TRANSPORT-INFO-REQ

Gambar 4.20. Hasil Packet Sniffing Real Player Pada topologi IPv4



data1.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
69	0.781330	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
70	0.781350	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
71	0.803369	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
72	0.803391	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
73	0.821157	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
74	0.852729	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
75	0.852758	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
76	0.867699	2001:0:0:a::10	2001:0:0:c::10	RDT	TRANSPORT-INFO-REQ
77	0.867718	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
78	0.880606	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
79	0.880622	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
80	0.881496	2001:0:0:c::10	2001:0:0:a::10	RDT	RESPONSE:
81	0.903466	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
82	0.929532	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01

Gambar 4.21. Hasil *Packet Sniffing Real Player* Pada topologi IPv6

data1.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
69	0.781330	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
70	0.781350	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
71	0.803369	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
72	0.803391	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
73	0.821157	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
74	0.852729	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
75	0.852758	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
76	0.867699	2001:0:0:a::10	2001:0:0:c::10	RDT	TRANSPORT-INFO-REQ
77	0.867718	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
78	0.880606	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
79	0.880622	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
80	0.881496	2001:0:0:c::10	2001:0:0:a::10	RDT	RESPONSE:
81	0.903466	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
82	0.929532	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01

Gambar 4.22. Hasil *Packet Sniffing Real Player* Pada topologi 6to4

data1.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
39	3.146003	2001:0:0:c::10	2001:0:0:a::10	RDT	RESPONSE:
40	3.177514	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
41	3.223005	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
42	3.223060	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
43	3.223186	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
44	3.223302	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
45	3.223397	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
46	3.267008	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
47	3.267055	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
48	3.267184	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
49	3.321706	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=01
50	3.365237	2001:0:0:a::10	2001:0:0:c::10	RDT	DATA: stream-id=00
51	3.365280	2001:0:0:a::10	2001:0:0:c::10	RDT	TRANSPORT-INFO-REQ

Gambar 4.23. Hasil *Packet Sniffing Real Player* Pada topologi manual

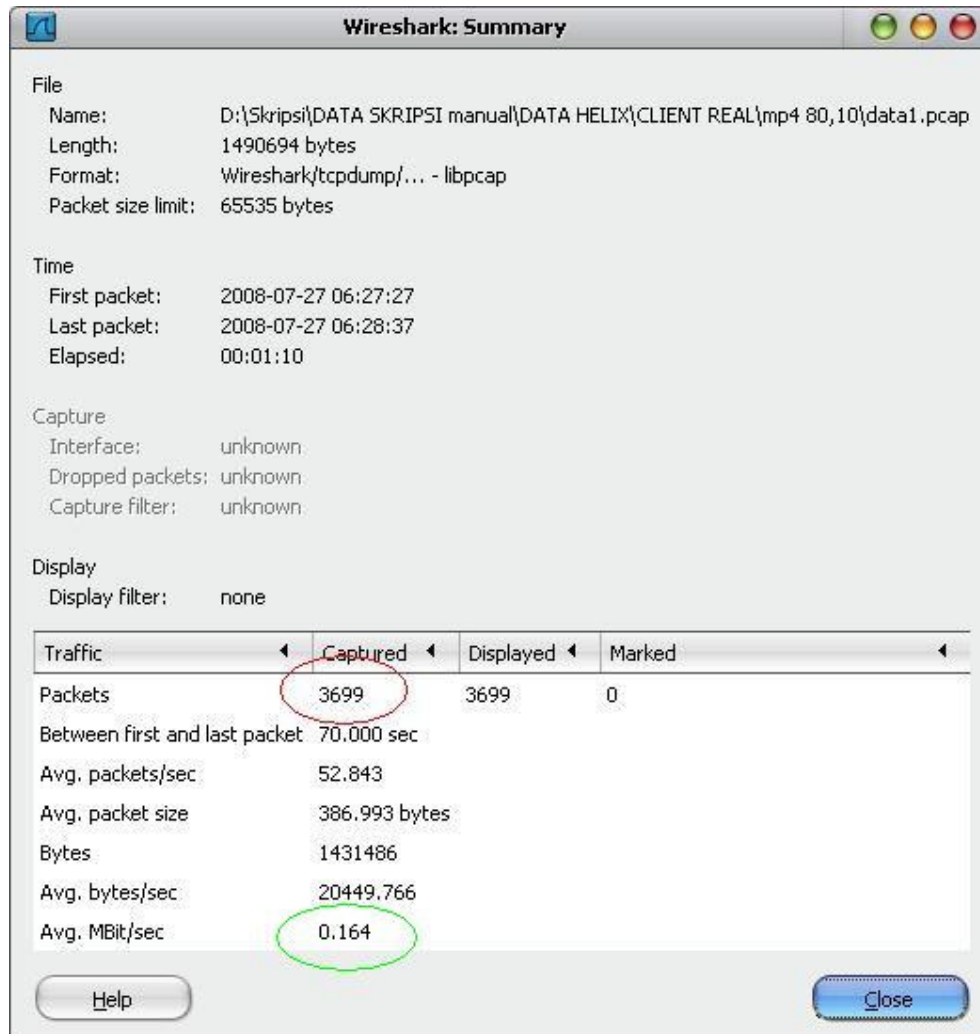
### 4.3. Pengolahan Data

Pengolahan data dilakukan guna membantu dalam menganalisa parameter-parameter pada setiap topologi jaringan. Dengan menggunakan berbagai macam jenis *video* dengan format yang berbeda-beda, seperti yang telah dijelaskan pada subbab metode pengambilan data pada Bab 3, akan menghasilkan data sebanyak lima buah data tiap *file video* tiap topologinya. Kemudian dari kelima data itu diambil rata-ratanya. Hasil rata-rata itu kemudian ditampilkan ke dalam tabel untuk nantinya dianalisa lebih lanjut.

Parameter yang digunakan adalah *packet loss* dan *throughput*. *Packet loss* yang ditampilkan di tabel adalah *packet loss* dalam bentuk presentase. Nilai presentase ini didapat dari mengurangi jumlah paket yang dikirimkan *server* dengan jumlah paket yang diterima oleh *client* lalu dibagi dengan jumlah paket yang dikirimkan *server*. Berikut ini adalah persamaan yang digunakan untuk mencari presentase *packet loss*:

$$\%packet\ loss = \frac{(\text{paket yang dikirim server} - \text{paket yang diterima client})}{\text{paket yang dikirimkan server}} \times 100\%$$

Untuk mendapatkan jumlah paket yang dikirim *server* dan jumlah paket yang diterima *client* digunakan *tools summary* pada wireshark. Gambar 4.24 adalah *screenshot* dari *tools summary* pada Wireshark.



Gambar 4.24. Summary pada Wireshark

Angka pada lingkaran yang atas pada Gambar 4.24 adalah nilai dari paket yang dikirim atau yang diterima, jika dilihat dari *summary* milik *client* maka nilai tersebut adalah jumlah paket yang diterima oleh *client* dan jika dilihat dari *summary* milik *server* maka nilai tersebut adalah jumlah paket yang dikirim oleh *server*.

Parameter kedua yang digunakan adalah *throughput*. Sama halnya dengan *packet loss*, nilai *throughput* didapat dengan melihat *tools summary* pada wireshark. Angka pada lingkaran bawah pada gambar 4.24. merupakan nilai dari *throughput* jaringan. Jumlah nilai *throughput* tiap *file video* sebanyak lima buah tiap topologi jaringan. Kelima nilai ini akan diambil nilai rata-ratanya terlebih dahulu, kemudian setelah itu nilai rata-rata inilah yang akan ditampilkan di tabel.



#### 4.4. Analisa Packet Loss VLC

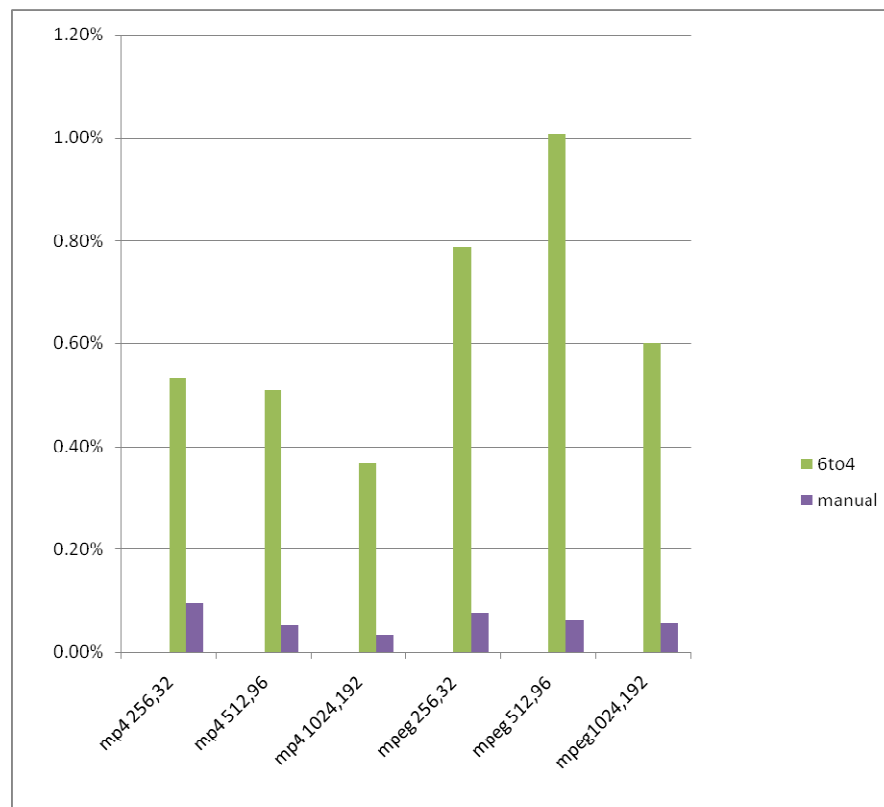
*Packet loss* menunjukkan banyaknya jumlah paket yang hilang dalam suatu proses pengiriman data dari satu titik ke titik yang lain. Tiap paket di dalam *video streaming* dapat berisi suatu *frame* gambar atau suatu *bit audio*, sehingga di dalam *video streaming packet loss* menunjukkan banyaknya jumlah *frame* yang hilang. Semakin banyak *frame* yang hilang akan menyebabkan semakin buruk kualitas gambarnya. Dalam mengamati performa sebuah jaringan, *packet loss* bisa menjadi salah satu parameternya, yaitu semakin besarnya jumlah *packet loss* dalam sebuah jaringan maka semakin buruk pula performa dari jaringan tersebut.

Adanya *packet loss* pada suatu jaringan disebabkan oleh adanya *congestion*. *Congestion* adalah kemacetan data sehingga menyebabkan adanya tumpukan atau antrian data dalam jumlah yang besar karena pengaruh efek *bottleneck* pada jaringan. Efek *bottleneck* pada jaringan disebabkan oleh tidak samanya kemampuan komponen-komponen jaringan dalam mengolah data. Komponen jaringan itu berupa dapat *laptop* yang digunakan oleh *client*, *laptop* yang digunakan pada *server* dan kedua *router* yang digunakan. Perbedaan kemampuan komponen jaringan terlihat pada perbedaan spesifikasi kedua *router* yaitu *router1* dan *router2*. Pada *router1* menggunakan *flash memory* sebesar 38862848 *bytes* sedangkan pada *router2* menggunakan *flash memory* 262144 *bytes*.

Ada enam macam *file video* yang digunakan pada *streaming* menggunakan VLC, keenam *file video* itu melewati proses *streaming* dengan menggunakan empat macam topologi jaringan yang berbeda. Adapun keempat macam topologi jaringan tersebut mempunyai karakteristik dan cara kerja yang berbeda-beda pula sehingga jumlah *packet loss* tiap topologi jaringan akan berbeda pula. Besarnya jumlah *packet loss* tiap topologi jaringan bisa dibandingkan dengan melihat data yang sudah diolah dan ditampilkan pada Tabel 4.1.

Tabel 4.1. Perbandingan Nilai Presentase *Packet Loss* VLC

	% <i>Packet Loss</i> VLC	
	6to4	manual
mp4 256,32	0.53%	0.09%
mp4 512,96	0.51%	0.05%
mp4 1024,192	0.37%	0.03%
mpeg 256,32	0.79%	0.08%
mpeg 512,96	1.01%	0.06%
mpeg1024,192	0.60%	0.06%

Gambar 4.25. Grafik Perbandingan *Packet Loss* VLC

Setelah melihat Tabel 4.1 dan Gambar 4.25 akan dapat diketahui bahwa performa terbaik ditunjukkan oleh topologi manual atau *manually configured* IPv6 *tunneling*. Rata-rata jumlah *packet loss* yang dimiliki topologi manual tidak lebih dari 0,1%. Topologi 6to4 atau *automatic 6to4 tunneling* memiliki rata-rata jumlah *packet loss* yang lebih besar dari pada topologi IPv6 dan topologi manual.

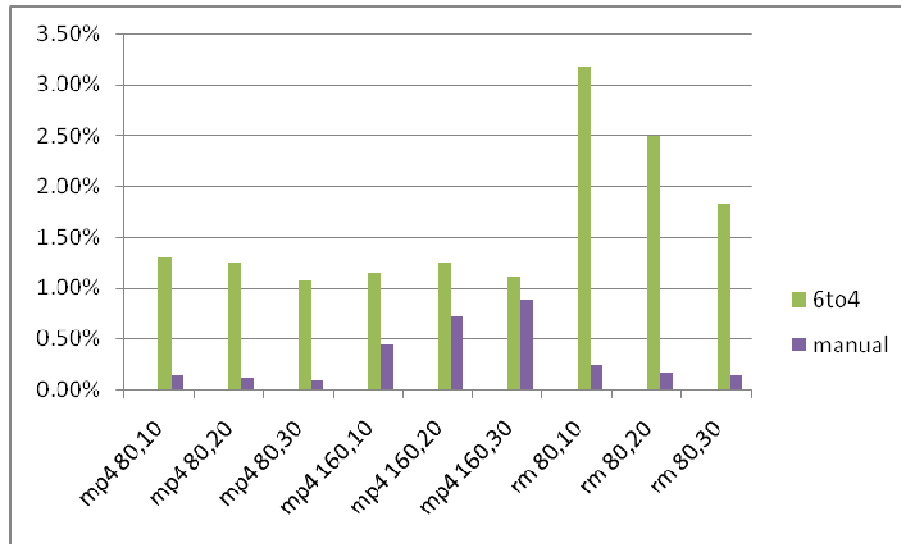
Hal ini disebabkan proses enkapsulasi yang digunakan oleh topologi 6to4, yaitu dengan menambahkan *address prefix* pada paketnya lalu mengirimkannya dengan menggunakan format IPv4. Berbeda dengan topologi manual yang tidak menggunakan *address prefix* dalam pengiriman paketnya. Pada topologi manual menggunakan format pengiriman IPv6 melalui *tunnel* yang mampu bekerja pada *cloud* IPv4. Hal inilah yang menyebabkan topologi manual memiliki performa jaringan yang lebih baik dibandingkan dengan topologi 6to4.

#### 4.5. Analisa Packet Loss Helix

HELIX merupakan aplikasi yang dikembangkan oleh *Realnetworks* sehingga protokol yang digunakannya berbeda dengan proses *streaming* menggunakan VLC. Pada proses *streaming* dengan menggunakan HELIX, protokol yang digunakan pada *transport layer*-nya adalah protokol *Real Data Transport* (RDT). Penggunaan protokol ini berlaku untuk semua topologi jaringan baik topologi IPv4, topologi IPv6, topologi 6to4, maupun topologi manual. Berbeda dengan *transport layer protocol* yang digunakan oleh VLC yaitu berupa protokol TCP dan UDP. Perbandingan jumlah *packet loss* dengan menggunakan HELIX dapat dilihat pada Tabel 4.2.

Tabel 4.2. Perbandingan Nilai Presentase *Packet Loss* HELIX

	<b>% Packet Loss HELIX</b>	
	<b>6to4</b>	<b>manual</b>
mp4 80,10	1.30%	0.12%
mp4 80,20	1.25%	0.10%
mp4 80,30	1.06%	0.08%
mp4 160,10	1.14%	0.44%
mp4 160,20	1.25%	0.71%
mp4 160,30	1.11%	0.87%
rm 80,10	3.18%	0.24%
rm 80,20	2.49%	0.15%
rm 80,30	1.83%	0.13%



Gambar 4.26. Grafik Perbandingan *Packet Loss* HELIX

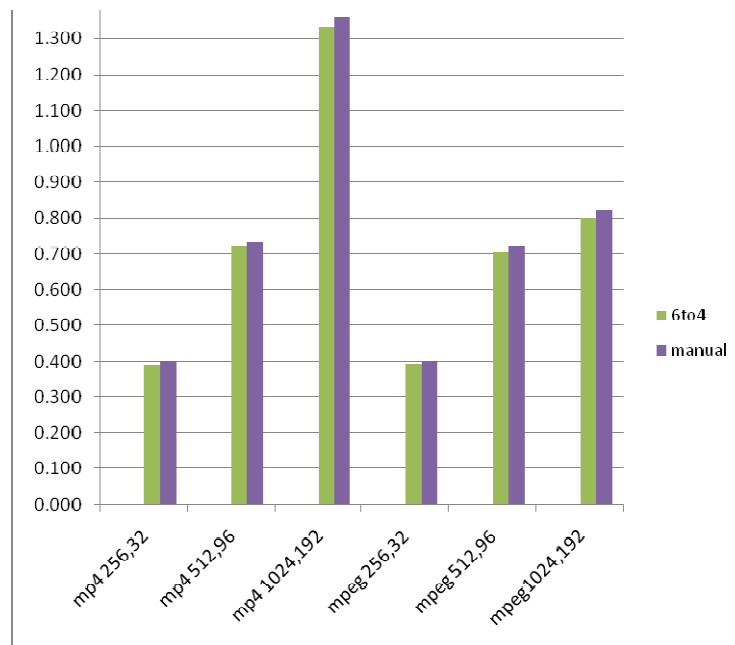
Dari Gambar 4.26 secara keseluruhan terlihat bahwa rata-rata jumlah *packet loss* dari kedua topologi tergolong bagus karena jumlah *packet loss* terbesar dicapai pada angka 3,18%. Jumlah *packet loss* ini masih termasuk dalam range 0 – 10 % dimana streaming memiliki kualitas yang halus dan bagus (Ramli, 2003). Presentase *packet loss* yang dihasilkan oleh topologi 6to4 lebih besar daripada presentase yang dihasilkan oleh topologi manual. Hal ini dikarenakan pada topologi 6to4 terjadi enkapsulasi paket, yaitu penambahan *address prefix* pada *header* agar paket mampu melewati *could* IPv4. Proses enkapsulasi dan penambahan *address prefix* inilah yang menyebabkan semakin besarnya peluang terjadinya *overhead* pada *router* sehingga topologi 6to4 memiliki jumlah *packet loss* yang lebih besar dibandingkan topologi manual. Baik *streaming* menggunakan HELIX maupun *streaming* menggunakan VLC, jumlah *packet loss* yang dihasilkan oleh topologi manual selalu lebih sedikit dibandingkan jumlah *packet loss* yang dihasilkan oleh topologi 6to4.

#### 4.6. Analisa *Throughput* VLC

*Throughput* menyatakan jumlah data yang berhasil dikirimkan dari satu titik ke titik lain dalam periode tertentu. Dapat dikatakan juga bahwa *throughput* adalah *bandwidth* efektif yang terukur pada suatu waktu tertentu untuk kondisi tertentu. Semakin besar nilai *throughput* dari suatu jaringan maka performa dari jaringan tersebut akan semakin bagus. Perbandingan nilai *throughput* dari antara topologi 6to4 dan topologi manual dapat dilihat pada Tabel 4.3.

Tabel 4.3. Perbandingan Nilai *Throughput* VLC

	<i>Throughput</i> VLC (MBit/sec)	
	6to4	Manual
mp4 256,32	0.3906	0.401
mp4 512,96	0.7226	0.7314
mp4 1024,192	1.333	1.3612
mpeg 256,32	0.3928	0.401
mpeg 512,96	0.7054	0.721
mpeg1024,192	0.7998	0.8224



Gambar 4.27. Grafik Perbandingan *Throughput* VLC

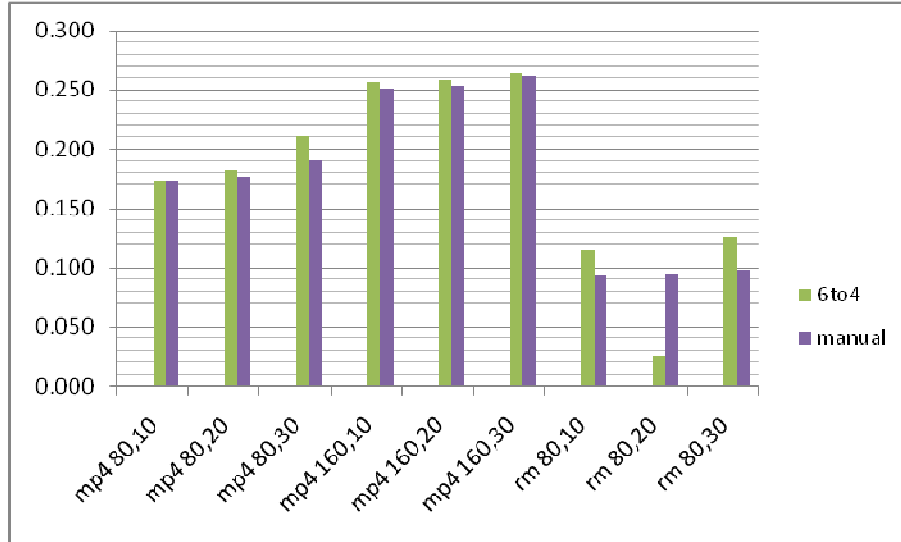
Setelah melihat Tabel 4.3 dan Gambar 4.27 dapat dikatakan bahwa nilai keseluruhan *throughput* dari kedua topologi tidak jauh berbeda. Diantara tipisnya perbedaan itu, nilai *throughput* manual terlihat sedikit lebih besar dibandingkan topologi 6to4. Perbedaan nilai *throughput* terbesar antara topologi 6to4 dengan topologi manual dicapai pada *streaming* dengan menggunakan *video* mp4 1024,192 yaitu sebesar 0,0282 MBit/sec. Perbedaan nilai *throughput* topologi 6to4 dengan topologi manual ini memang tidak signifikan sehingga tidak dapat ditentukan topologi mana yang mempunyai performa yang lebih baik, atau dengan kata lain performa yang dihasilkan oleh kedua topologi adalah sama.

#### 4.7. Analisa *Throughput* HELIX

Sama halnya dengan nilai *throughput* yang dihasilkan dari proses *streaming* dengan menggunakan VLC, nilai *throughput* yang dihasilkan dari proses *streaming* dengan menggunakan HELIX tidak jauh berbeda antara kedua topologi jaringan. Pada *streaming* dengan HELIX menggunakan sembilan jenis *video* yang berbeda-beda. Dapat dilihat pada Tabel 4.4 dan Gambar 4.28 bahwa hampir semua nilai *throughput* memiliki nilai yang tidak jauh berbeda antara topologi 6to4 dan topologi manual.

Tabel 4.4. Perbandingan Nilai *Throughput* HELIX

	<b><i>Throughput</i> HELIX (MBit/sec)</b>	
	<b>6to4</b>	<b>manual</b>
mp4 80,10	0.173	0.173
mp4 80,20	0.182	0.176
mp4 80,30	0.212	0.191
mp4 160,10	0.256	0.252
mp4 160,20	0.258	0.253
mp4 160,30	0.264	0.262
rm 80,10	0.115	0.093
rm 80,20	0.024	0.094
rm 80,30	0.125	0.098



Gambar 4.28. Grafik Perbandingan *Throughput* HELIX

Nilai *throughput* antara topologi 6to4 dengan topologi manual tidak menunjukkan adanya indikasi topologi mana yang memiliki performa yang lebih baik, karena nilai *throughput* topologi 6to4 tidak selalu lebih besar dibandingkan topologi manual begitu juga sebaliknya. Nilai *throughput* kedua topologi tersebut memiliki perbedaan yang tidak signifikan. Perbedaan nilai *throughput* terbesar dicapai pada proses *streaming* dengan menggunakan *video* rm 80, 20 yaitu sebesar 0,07Mbit/sec.

#### 4.8. Perbandingan Rata-rata Performa Jaringan

Dari hasil pengolahan data pada subbab sebelumnya dapat diketahui bahwa topologi *manually configured IPv6 tunneling* (manual) memiliki performa yang lebih baik dibandingkan topologi *Automatic 6to4 tunneling* dalam melakukan *streaming video* baik dengan menggunakan VLC maupun dengan menggunakan HELIX. Perbedaan performa terlihat pada perbedaan jumlah *packet loss* pada kedua topologi tersebut yang dapat dilihat pada Tabel 4.5.

Tabel 4.5. Rata-rata *Packet Loss*

rata-rata <i>packet loss</i>		
	<b>6to4</b>	<b>manual</b>
VLC	0.42%	0.04%
HELIX	1.62%	0.32%

Tabel 4.6. Rata-rata *Throughput*

rata-rata <i>throughput</i> (Mbit/sec)		
	<b>6to4</b>	<b>manual</b>
VLC	0.7240	0.7397
HELIX	0.1768	0.1649

Dari Tabel 4.5 dapat dilihat bahwa perbedaan rata-rata jumlah *packet loss* antara topologi 6to4 dengan topologi manual adalah sebesar 0,38% pada streaming dengan menggunakan VLC, sedangkan pada *streaming* menggunakan HELIX perbedaannya sebesar 1,3%, dengan kondisi jumlah *packet loss* topologi manual selalu lebih sedikit dibandingkan topologi 6to4 baik dengan menggunakan VLC maupun dengan menggunakan HELIX. Ini dapat mengindikasikan bahwa performa topologi manual di atas topologi 6to4 bila dilihat menggunakan parameter *packet loss*. Jika dilihat menggunakan parameter *throughput* perbedaan performa antara kedua topologi tersebut tidak jauh berbeda. Dapat dilihat pada Tabel 4.6 bahwa perbedaan rata-rata nilai *throughput* antara topologi 6to4 dengan topologi manual adalah sebesar 0,0157Mbit/sec pada *streaming* dengan menggunakan VLC, sedangkan pada *streaming* dengan menggunakan HELIX perbedaannya adalah sebesar 0,0119Mbit/sec, dengan kondisi nilai *throughput* topologi manual lebih besar pada streaming menggunakan VLC dan nilai *throughput* topologi 6to4 lebih besar pada *streaming* menggunakan HELIX. Nilai ini tentu tidak dapat menunjukkan topologi mana yang mempunyai performa terbaik.

Pengujian performa jaringan dengan menggunakan jaringan ini dilakukan pada skala kecil yaitu hanya sebatas *point-to-point*. Di dalam penerapan jaringan berskala kecil topologi manual mempunyai performa yang lebih bagus dibandingkan topologi 6to4, namun topologi manual memiliki kekurangan bila diterapkan pada jaringan berskala besar. Pada jaringan berskala besar,



penambahan komponen jaringan bisa lebih banyak dilakukan sehingga bila menggunakan topologi manual akan memerlukan waktu yang banyak untuk mengkonfigurasi ulang jaringan apabila terjadi penambahan komponen jaringan baru. Dengan menggunakan topologi 6to4 hal yang demikian tidak perlu dilakukan sehingga penerapan topologi 6to4 lebih cocok diterapkan pada jaringan berskala besar.

## BAB 5

### KESIMPULAN

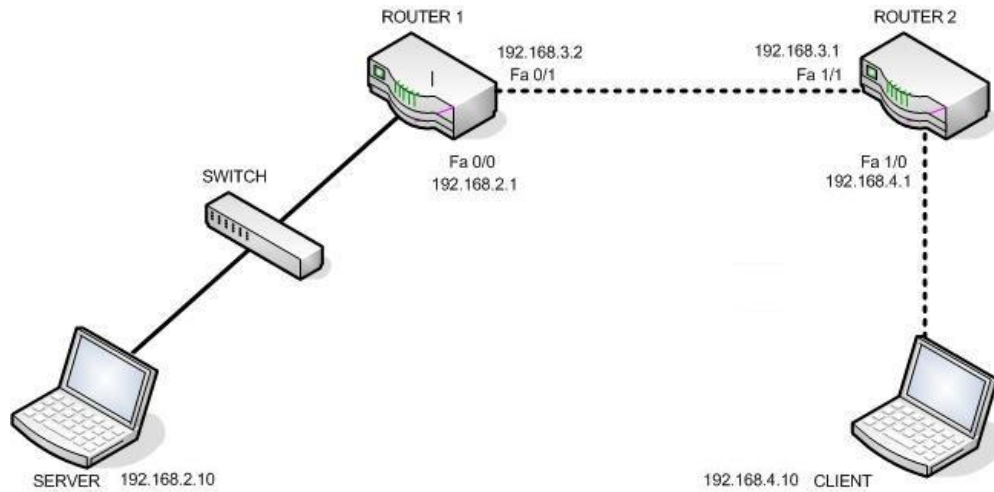
1. Dengan menggunakan parameter *packet loss*, topologi jaringan *manually configured IPv6 tunneling* mempunyai performa yang lebih baik dibandingkan topologi jaringan *automatic 6to4 tunneling*, dengan selisih presentase *packet loss* sebesar 0,38% pada *streaming* menggunakan VLC dan selisih presentase *packet loss* sebesar 1,3% pada *streaming* menggunakan HELIX.
2. Dengan menggunakan parameter *throughput*, kesemua topologi yang digunakan menunjukkan performa yang sama karena selisih nilai *throughput* kedua topologi tidak signifikan.
3. Berdasarkan penelitian yang didapat, Topologi jaringan *manually configured IPv6 tunneling* lebih dianjurkan untuk digunakan pada jaringan berskala kecil karena performanya yang lebih baik yang ditunjukkan kecilnya presentase *packet loss*.
4. Berdasarkan penelitian yang didapat, Topologi jaringan *automatic 6to4 tunneling* lebih dianjurkan untuk digunakan pada jaringan berskala besar karena tidak diperlukan adanya konfigurasi ulang pada tunnel jika ada penambahan komponen jaringan baru.

## DAFTAR REFERENSI

- Jordi Palet. (2007, April). *The Choise: IPv4 Exhaustion or Transition to IPv6*.
- Microsoft Technet. (2003). *How IPv6 Works*.
- S. Deering & R. Hinden. (2003, April). *Internet Protocol Version 6 (IPv6) Addressing Architecture*. IETF, RFC 3513.
- Microsoft Corporation. (2008). *Introduction to IP Version 6*.
- Lorenzo Colitti, *et al.* (2004, April). *IPv6-in-IPv4 unnel discovery: methods and experimental results*. IEEE Transactions on Network and Service Management, vol. 1, no. 1.
- Jean-Francois Tremblay & Mikael Lind. (2006, September). *IPv6 Tunneling Techniques*. Hexago Inc.
- Ramli, Kalamullah. (2003, Oktober). *An improved Active Network Concept and Architechture for Distributed and Dynamic Streaming Multimedia Environments with Heterogenous Bandwidths*. Universitat Duisburg-Essen.
- (2003). *IPv6: Connecting to the 6bone Using Manually Configured Tunnels*. 12 September, 2008.  
[http://www.cisco.com/en/US/docs/ios/solutions\\_docs/ipv6/6bmctnl.html](http://www.cisco.com/en/US/docs/ios/solutions_docs/ipv6/6bmctnl.html)
- (2003). *IPv6: Connecting to the 6bone Using 6to4 Tunnels*. 12 September, 2008.  
[http://www.cisco.com/en/US/docs/ios/solutions\\_docs/ipv6/6b624tnl.html](http://www.cisco.com/en/US/docs/ios/solutions_docs/ipv6/6b624tnl.html)
- Wilson, Tracy V. (n.d.). *How Streaming Video and Audio Work*. 12 September, 2008. <http://computer.howstuffworks.com/streaming-video-and-audio3.htm>

## LAMPIRAN 1

### KONFIGURASI JARINGAN IPv4 MURNI



#### **Server**

```
C:\> netsh interface ip add address "Local Area Connection" 192.168.2.10 255.255.255.0
C:\> netsh interface ip add address "Local Area Connection" gateway=192.168.4.1 gwmetric=0
```

#### **Client**

```
C:\> netsh interface ip add address "Local Area Connection" 192.168.4.10 255.255.255.0
C:\> netsh interface ip add address "Local Area Connection" gateway=192.168.4.1 gwmetric=0
```

#### **Router1**

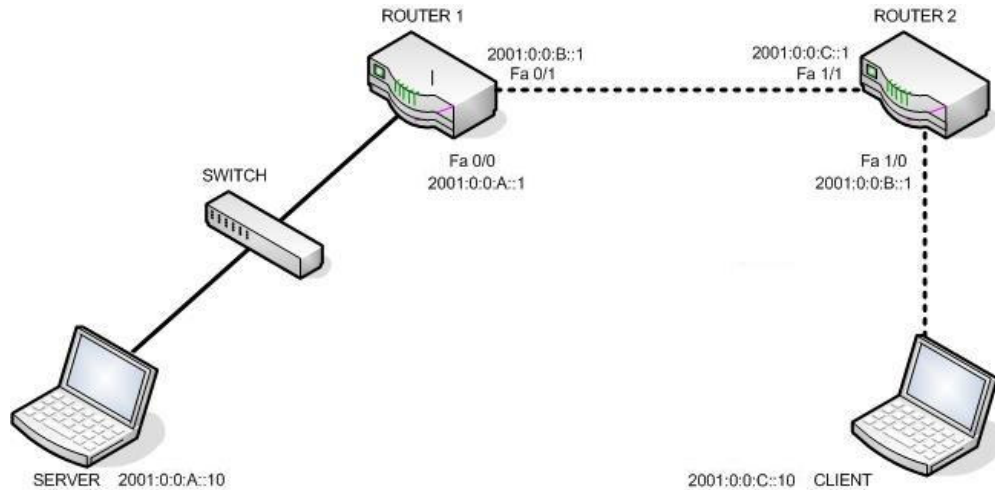
```
Router1>enable
Router1#configure terminal
Router1(config)#interface fastethernet 0/0
Router1(config-if)#ip address 192.168.2.1 255.255.255.0
Router1(config-if)#no shutdown
Router1(config)#interface fastethernet 0/1
Router1(config-if)#ip address 192.168.3.2 255.255.255.0
Router1(config-if)#no shutdown
Router1(config)#router rip
Router1(config- router)#version 2
Router1(config- router)#network 192.168.2.0
Router1(config- router)#network 192.168.3.0
Router1(config- router)#passive-interface fastethernet 0/0
Router1(config- router)#no auto-summary
```

***Router2***

```
Router2>enable
Router2#configure terminal
Router2(config)#interface fastethernet 1/0
Router2(config-if)#ip address 192.168.4.1 255.255.255.0
Router2(config-if)#no shutdown
Router2(config)#interface fastethernet 1/1
Router2(config-if)#ip address 192.168.3.1 255.255.255.0
Router2(config-if)#no shutdown
Router2(config)#router rip
Router2(config- router)#version 2
Router2(config- router)#network 192.168.3.0
Router2(config- router)#network 192.168.4.0
Router2(config- router)#passive-interface fastethernet 1/0
Router2(config- router)#no auto-summary
```

## LAMPIRAN 2

### KONFIGURASI JARINGAN IPV6 MURNI



#### Server

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:A::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:A::1
```

#### Client

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:C::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:C::1
```

#### Router1

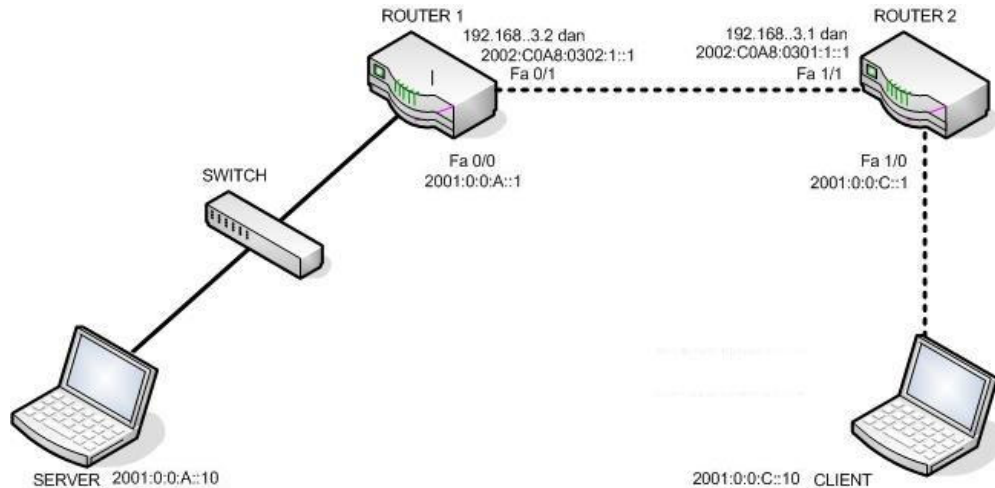
```
Router1>enable
Router1#configure terminal
Router1(config)#ipv6 unicast routing
Router1(config)#interface fastethernet 0/0
Router1(config-if)#ipv6 enable
Router1(config-if)#ipv6 address 2001:0:0:A::1/64
Router1(config-if)#ipv6 rip process1 enable
Router1(config-if)#no shutdown
Router1(config)#interface fastethernet 0/1
Router1(config-if)#ipv6 enable
Router1(config-if)#ipv6 address 2001:0:0:B::2/64
Router1(config-if)#ipv6 rip process1 enable
Router1(config-if)#no shutdown
```

**Router2**

```
Router2>enable
Router2#configure terminal
Router2(config)#ipv6 unicast routing
Router2(config)#interface fastethernet 1/0
Router2(config-if)#ipv6 enable
Router2(config-if)#ipv6 address 2001:0:0:C::1/64
Router2(config-if)#ipv6 rip process1 enable
Router2(config-if)#no shutdown
Router2(config)#interface fastethernet 1/1
Router2(config-if)#ipv6 enable
Router2(config-if)#ipv6 address 2001:0:0:B::2/64
Router2(config-if)#ipv6 rip process1 enable
Router2(config-if)#no shutdown
```

### LAMPIRAN 3

#### KONFIGURASI JARINGAN 6to4



#### Server

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:A::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:A::1
```

#### Client

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:C::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:C::1
```

#### Router1

```
Router1>enable
Router1#configure terminal
Router1(config)#ipv6 unicast routing
Router1(config)#interface fastethernet 0/0
Router1(config-if)#ipv6 enable
Router1(config-if)#ipv6 address 2001:0:0:A::1/64
Router1(config-if)#ipv6 rip process1 enable
Router1(config-if)#no shutdown
Router1(config)#interface fastethernet 0/1
Router1(config-if)#ipv6 enable
Router1(config-if)#ip address 192.168.3.2 255.255.255.0
Router1(config-if)#ipv6 address 2002:C0A8:0302:1::1/64
Router1(config)#interface tunnel 0
```



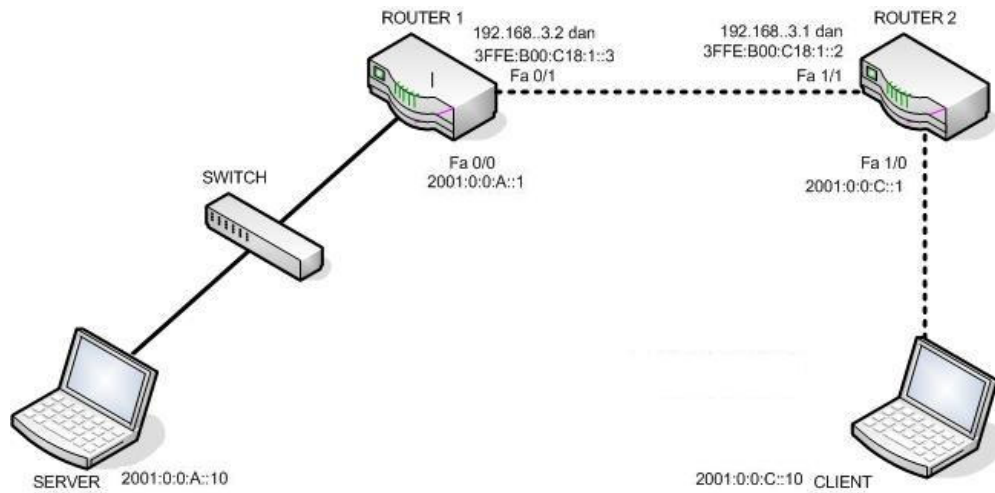
```
Router1(config-if)#ipv6 enable
Router1(config-if)#no ip address
Router1(config-if)#ipv6 unnumbered fastethernet 0/1
Router1(config-if)#tunnel source fastethernet 0/1
Router1(config-if)#tunnel mode ipv6ip 6to4
Router1(config)#router rip
Router1(config- router)#version 2
Router1(config- router)#network 192.168.3.0
Router1(config)#ipv6 route 2002::/16 tunnel 0
Router1(config)#ipv6 route ::/0 2002:C0A8:0301:1::1
```

## Router2

```
Router2>enable
Router2#configure terminal
Router2(config)#ipv6 unicast routing
Router2(config)#interface fastethernet 1/0
Router2(config-if)#ipv6 enable
Router2(config-if)#ipv6 address 2001:0:0:C::1/64
Router2(config-if)#ipv6 rip process1 enable
Router2(config-if)#no shutdown
Router2(config)#interface fastethernet 1/1
Router2(config-if)#ipv6 enable
Router2(config-if)#ip address 192.168.3.1 255.255.255.0
Router2(config-if)#ipv6 address 2002:C0A8:0301:1::1/64
Router2(config)#interface tunnel 0
Router2(config-if)#ipv6 enable
Router2(config-if)#no ip address
Router2(config-if)#ipv6 unnumbered fastethernet 1/1
Router2(config-if)#tunnel source fastethernet 1/1
Router2(config-if)#tunnel mode ipv6ip 6to4
Router2(config)#router rip
Router2(config- router)#version 2
Router2(config- router)#network 192.168.3.0
Router2(config)#ipv6 route 2002::/16 tunnel 0
Router1(config)#ipv6 route ::/0 2002:C0A8:0302:1::1
```

## LAMPIRAN 4

### KONFIGURASI JARINGAN MANUAL



#### Server

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:A::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:A::1
```

#### Client2

```
C:\> ipv6 install
C:\> netsh interface ipv6 add address "Local Area Connection" 2001:0:0:C::10
C:\> netsh interface ipv6 add route ::/0 "Local Area Connection" 2001:0:0:C::1
```

#### Router1

```
Router0>enable
Router0#configure terminal
Router0(config)#ipv6 unicast routing
Router0(config)#interface fastethernet 0/0
Router0(config-if)#ipv6 enable
Router0(config-if)#ipv6 address 2001:0:0:A::1/64
Router0(config-if)#ipv6 router rip process1
Router0(config-if)#no shutdown
Router0(config)#interface fastethernet 0/1
Router0(config-if)#ip address 192.168.30.2 255.255.255.0
```

```
Router0(config-if)#no shutdown
Router0(config)#interface tunnel0
Router0(config-if)#ipv6 address 3FFE:B00:C18:1::3/64
Router0(config-if)#tunnel source fastethernet 0/1
Router0(config-if)#tunnel destination 192.168.30.1
Router0(config-if)#tunnel mode ipv6ip
Router0(config-if)#no shutdown
```

## Router2

```
Router0>enable
Router0#configure terminal
Router0(config)#ipv6 unicast routing
Router0(config)#interface fastethernet 1/0
Router0(config-if)#ipv6 enable
Router0(config-if)#ipv6 address 2001:0:0:C::1/64
Router0(config)#ipv6 router rip process1
Router0(config-if)#no shutdown
Router0(config)#interface fastethernet 1/1
Router0(config-if)#ip address 192.168.30.1 255.255.255.0
Router0(config-if)#no shutdown
Router0(config)#interface tunnel0
Router0(config-if)#ipv6 address 3FFE:B00:C18:1::2/64
Router0(config-if)#tunnel source fastethernet 1/1
Router0(config-if)#tunnel destination 192.168.30.2
Router0(config-if)#tunnel mode ipv6ip
Router0(config-if)#no shutdown
```

**LAMPIRAN 5**  
**DATA PERCOBAAN VLC JARINGAN IPV4**

IPV4 - VLC - mp4 256, 32			
server packet	client packet	% packet loss	throughput
10102	8224	18.59%	0.368
10451	8236	21.19%	0.381
10539	8327	20.99%	0.402
10498	8267	21.25%	0.43
10453	8226	21.30%	0.429
		103.33%	2.01
	rata-rata	20.67%	0.402

IPV4 - VLC - mp4 512, 96			
server packet	client packet	% packet loss	throughput
18805	14816	21.21%	0.782
18796	14805	21.23%	0.783
18801	14812	21.22%	0.783
18792	14793	21.28%	0.772
18806	14812	21.24%	0.774
		106.18%	3.894
	rata-rata	21.24%	0.7788

IPV4 - VLC - mp4 1024, 192			
server packet	client packet	% packet loss	throughput
34026	26674	21.61%	1.344
34536	27176	21.31%	1.469
34531	27183	21.28%	1.424
34533	27186	21.28%	1.434
34527	27179	21.28%	1.439
		106.75%	7.11
	rata-rata	21.35%	1.422

IPV4 - VLC - mpeg 256, 32			
server packet	client packet	% packet loss	throughput
10222	8006	21.68%	0.371
10251	8042	21.55%	0.425
10212	8014	21.52%	0.41
10252	8057	21.41%	0.408
10258	8062	21.41%	0.414
		107.57%	2.028
	rata-rata	21.51%	0.4056

IPV4 - VLC - mpeg 512, 96			
server packet	client packet	% packet loss	throughput
18192	14297	21.41%	0.685
18226	14334	21.35%	0.746
18237	14344	21.35%	0.711
18188	14306	21.34%	0.728
18219	14334	21.32%	0.736
		106.78%	3.606
	rata-rata	21.36%	0.7212

IPV4 - VLC - mpeg 1024, 192			
server packet	client packet	% packet loss	throughput
20799	16322	21.53%	0.809
21169	16734	20.95%	0.801
20724	16297	21.36%	0.824
20694	16263	21.41%	0.787
20353	15922	21.77%	0.852
		107.02%	4.073
	rata-rata	21.40%	0.8146

**LAMPIRAN 6**  
**DATA PERCOBAAN VLC JARINGAN IPV6**

IPV6 - VLC - mp4 256, 32			
server packet	client packet	% packet loss	throughput
2138	2130	0.37%	0.395
2138	2125	0.61%	0.397
2138	2127	0.51%	0.398
2138	2130	0.37%	0.391
2138	2126	0.56%	0.397
		2.43%	1.978
	rata-rata	0.49%	0.3956

IPV6 - VLC - mp4 512, 96			
server packet	client packet	% packet loss	throughput
3900	3878	0.56%	0.69
3900	3876	0.62%	0.718
3900	3891	0.23%	0.726
3898	3875	0.59%	0.718
3900	3884	0.41%	0.72
		2.41%	3.572
	rata-rata	0.48%	0.7144

IPV6 - VLC - mp4 1024, 192			
server packet	client packet	% packet loss	throughput
7230	7230	0.00%	1.278
7253	7212	0.57%	1.347
7254	7227	0.37%	1.328
7253	7226	0.37%	1.295
7252	7234	0.25%	1.327
		1.56%	6.575
	rata-rata	0.31%	1.315

IPV6 - VLC - mpeg 256, 32			
server packet	client packet	% packet loss	throughput
2110	2110	0.00%	0.401
2110	2110	0.00%	0.401
2110	2110	0.00%	0.401
2109	2109	0.00%	0.401
2110	2110	0.00%	0.401
		0.00%	2.005
	rata-rata	0.00%	0.401

IPV6 - VLC - mpeg 512, 96			
server packet	client packet	% packet loss	throughput
3794	3773	0.55%	0.674
3795	3772	0.61%	0.708
3795	3778	0.45%	0.703
3795	3773	0.58%	0.703
3795	3781	0.37%	0.704
		2.56%	3.492
	rata-rata	0.51%	0.6984

IPV6 - VLC - mpeg 1024, 192			
server packet	client packet	% packet loss	throughput
4315	4315	0.00%	0.791
4327	4305	0.51%	0.809
4309	4309	0.00%	0.771
4328	4310	0.42%	0.796
4330	4293	0.85%	0.806
		1.78%	3.973
	rata-rata	0.36%	0.7946

**LAMPIRAN 7**  
**DATA PERCOBAAN VLC JARINGAN AUTOMATIC 6TO4 TUNNELING**

6to4 - VLC - mp4 256, 32			
server packet	client packet	% packet loss	throughput
2138	2123	0.70%	0.401
2138	2126	0.56%	0.387
2138	2132	0.28%	0.38
2138	2127	0.51%	0.391
2138	2125	0.61%	0.394
		2.67%	1.953
	rata-rata	0.53%	0.3906

6to4 - VLC - mp4 512, 96			
server packet	client packet	% packet loss	throughput
3899	3883	0.41%	0.726
3900	3882	0.46%	0.725
3899	3874	0.64%	0.718
3900	3878	0.56%	0.718
3899	3881	0.46%	0.726
		2.54%	3.613
	rata-rata	0.51%	0.7226

6to4 - VLC - mp4 1024, 192			
server packet	client packet	% packet loss	throughput
7252	7223	0.40%	1.34
7252	7220	0.44%	1.361
7251	7231	0.28%	1.339
7252	7231	0.29%	1.297
7253	7222	0.43%	1.328
		1.83%	6.665
	rata-rata	0.37%	1.333



6to4 - VLC - mpeg 256, 32			
server packet	client packet	% packet loss	throughput
2110	2096	0.66%	0.401
2110	2093	0.81%	0.388
2110	2093	0.81%	0.393
2110	2093	0.81%	0.389
2110	2092	0.85%	0.393
		3.93%	1.964
	rata-rata	0.79%	0.3928

6to4 - VLC - mpeg 512, 96			
server packet	client packet	% packet loss	throughput
3795	3772	0.61%	0.708
3794	3692	2.69%	0.7
3795	3777	0.47%	0.703
3795	3774	0.55%	0.721
3795	3768	0.71%	0.695
		5.03%	3.527
	rata-rata	1.01%	0.7054

6to4 - VLC - mpeg 1024, 192			
server packet	client packet	% packet loss	throughput
4330	4302	0.65%	0.809
4329	4305	0.55%	0.788
4329	4299	0.69%	0.799
4329	4308	0.49%	0.802
4329	4302	0.62%	0.801
		3.00%	3.999
	rata-rata	0.60%	0.7998

**LAMPIRAN 8**  
**DATA PERCOBAAN VLC JARINGAN MANUALLU CONFIGURED IPV6**  
**TUNNELING**

manual - VLC - mp4 256, 32			
server packet	client packet	% packet loss	throughput
2143	2141	0.09%	0.401
2143	2143	0.00%	0.401
2148	2144	0.19%	0.401
2143	2143	0.00%	0.401
2145	2141	0.19%	0.401
		0.47%	2.005
	rata-rata	0.09%	0.401

manual - VLC - mp4 512, 96			
server packet	client packet	% packet loss	throughput
3908	3904	0.10%	0.731
3904	3904	0.00%	0.731
3901	3901	0.00%	0.732
3908	3904	0.10%	0.731
3905	3903	0.05%	0.732
		0.26%	3.657
	rata-rata	0.05%	0.7314

manual - VLC - mp4 1024, 192			
server packet	client packet	% packet loss	throughput
7263	7259	0.06%	1.361
7271	7267	0.06%	1.361
7248	7244	0.06%	1.362
7252	7252	0.00%	1.361
7252	7252	0.00%	1.361
		0.17%	6.806
	rata-rata	0.03%	1.3612

manual - VLC - mpeg 256, 32			
server packet	client packet	% packet loss	throughput
2117	2113	0.19%	0.401
2117	2117	0.00%	0.401
2113	2113	0.00%	0.401
2117	2113	0.19%	0.401
2113	2113	0.00%	0.401
		0.38%	2.005
	rata-rata	0.08%	0.401

manual - VLC - mpeg 512, 96			
server packet	client packet	% packet loss	throughput
3804	3800	0.11%	0.721
3802	3798	0.11%	0.721
3801	3801	0.00%	0.721
3802	3798	0.11%	0.721
3798	3798	0.00%	0.721
		0.32%	3.605
	rata-rata	0.06%	0.721

manual - VLC - mpeg 1024, 192			
server packet	client packet	% packet loss	throughput
4343	4339	0.09%	0.822
4336	4332	0.09%	0.822
4332	4332	0.00%	0.822
4337	4333	0.09%	0.823
4333	4333	0.00%	0.823
		0.28%	4.112
	rata-rata	0.06%	0.8224

**LAMPIRAN 9**  
**DATA PERCOBAAN HELIX JARINGAN IPV4**

IPV4 - HELIX - mp4 80,10			
server packet	client packet	% packet loss	throughput
3823	3785	0.99%	0.176
3733	3690	1.15%	0.173
3712	3673	1.05%	0.139
3683	3627	1.52%	0.163
3739	3694	1.20%	0.159
		5.92%	0.81
	rata-rata	1.18%	0.162

IPV4 - HELIX - mp4 80,20			
server packet	client packet	% packet loss	throughput
4293	4254	0.91%	0.169
4303	4241	1.44%	0.173
4317	4272	1.04%	0.161
4309	4266	1.00%	0.166
4297	4250	1.09%	0.146
		5.48%	0.815
	rata-rata	1.10%	0.163

IPV4 - HELIX - mp4 80,30			
server packet	client packet	% packet loss	throughput
4961	4903	1.17%	0.185
4953	4908	0.91%	0.185
4944	4902	0.85%	0.172
4969	4920	0.99%	0.174
4945	4898	0.95%	0.184
		4.86%	0.9
	rata-rata	0.97%	0.18

IPV4 - HELIX - mp4 160,10			
server packet	client packet	% packet loss	throughput
4257	4211	1.08%	0.255
4259	4216	1.01%	0.253
4242	4196	1.08%	0.252
4243	4201	0.99%	0.253
4239	4199	0.94%	0.243
		5.11%	1.256
	rata-rata	1.02%	0.2512

IPV4 - HELIX - mp4 160,20			
server packet	client packet	% packet loss	throughput
4469	4427	0.94%	0.248
4470	4424	1.03%	0.254
4484	4436	1.07%	0.255
4492	4448	0.98%	0.232
4481	4434	1.05%	0.256
		5.07%	1.245
	rata-rata	1.01%	0.249

server packet	client packet	% packet loss	throughput
5100	5057	0.84%	0.251
5111	5062	0.96%	0.257
5102	5057	0.88%	0.258
5123	5074	0.96%	0.235
5097	5055	0.82%	0.25
		4.46%	1.251
	rata-rata	0.89%	0.2502

IPV4 - HELIX - rm 80,10			
server packet	client packet	% packet loss	throughput
1674	1629	2.69%	0.111
1679	1634	2.68%	0.105
1673	1633	2.39%	0.107
1675	1635	2.39%	0.104
1674	1632	2.51%	0.111
		12.66%	0.538
	rata-rata	2.53%	0.1076

server packet	client packet	% packet loss	throughput
2178	2134	2.02%	0.112
2179	2133	2.11%	0.111
2178	2133	2.07%	0.112
2176	2135	1.88%	0.115
2177	2133	2.02%	0.109
		10.10%	0.559
	rata-rata	2.02%	0.1118

IPV4 - HELIX - rm 80,30			
server packet	client packet	% packet loss	throughput
2499	2459	1.60%	0.116
2498	2458	1.60%	0.109
2502	2458	1.76%	0.118
2499	2457	1.68%	0.115
2503	2458	1.80%	0.119
		8.44%	0.577
	rata-rata	1.69%	0.1154

**LAMPIRAN 10**  
**DATA PERCOBAAN HELIX JARINGAN IPV6**

IPV6 - HELIX - mp4 80,10			
server packet	client packet	% packet loss	throughput
3693	3689	0.11%	0.182
3698	3694	0.11%	0.182
3683	3679	0.11%	0.181
3759	3655	2.77%	0.18
3683	3679	0.11%	0.18
		3.20%	0.905
	rata-rata	0.64%	0.181

IPV6 - HELIX - mp4 80,20			
server packet	client packet	% packet loss	throughput
4317	4261	1.30%	0.183
4245	4241	0.09%	0.182
4249	4245	0.09%	0.183
4260	4256	0.09%	0.183
4263	4259	0.09%	0.183
		1.67%	0.914
	rata-rata	0.33%	0.1828

IPV6 - HELIX - mp4 80,30			
server packet	client packet	% packet loss	throughput
4914	4910	0.08%	0.197
4906	4902	0.08%	0.196
4744	4711	0.70%	0.194
4909	4905	0.08%	0.197
4909	4905	0.08%	0.196
		1.02%	0.98
	rata-rata	0.20%	0.196

IPV6 - HELIX - mp4 160,10			
server packet	client packet	% packet loss	throughput
4202	4198	0.10%	0.262
4196	4192	0.10%	0.263
4200	4196	0.10%	0.262
4218	4214	0.09%	0.264
4200	4200	0.00%	0.265
		0.38%	1.316

IPV6 - HELIX - mp4 160,20			
server packet	client packet	% packet loss	throughput
4420	4416	0.09%	0.262
4416	4414	0.05%	0.263
4430	4430	0.00%	0.246
4450	4450	0.00%	0.266
4433	4429	0.09%	0.264
		0.23%	1.301
	rata-rata	0.05%	0.2602

IPV6 - HELIX - mp4 160,30			
server packet	client packet	% packet loss	throughput
5077	5071	0.12%	0.249
5049	5047	0.04%	0.269
5053	5051	0.04%	0.269
5062	5062	0.00%	0.27
5058	5056	0.04%	0.269
		0.24%	1.326
	rata-rata	0.05%	0.2652



IPV6 - HELIX - rm 80,10			
server packet	client packet	% packet loss	throughput
1634	1634	0.00%	0.115
1631	1631	0.00%	0.115
1632	1632	0.00%	0.115
1635	1635	0.00%	0.115
1634	1634	0.00%	0.115
		0.00%	0.575
	rata-rata	0.00%	0.115

server packet	client packet	% packet loss	throughput
4317	4261	1.30%	0.183
4310	4241	1.60%	0.182
4249	4245	0.09%	0.183
4260	4256	0.09%	0.183
4263	4259	0.09%	0.183
		3.18%	0.914
	rata-rata	0.64%	0.1828

IPV6 - HELIX - rm 80,30			
server packet	client packet	% packet loss	throughput
2460	2460	0.00%	0.125
2457	2457	0.00%	0.125
2457	2457	0.00%	0.125
2457	2457	0.00%	0.125
2460	2460	0.00%	0.125
		0.00%	0.625
	rata-rata	0.00%	0.125

**LAMPIRAN 11**  
**DATA PERCOBAAN HELIX JARINGAN AUTOMATIC 6TO4**  
**TUNNELING**

6to4 - HELIX - mp4 80,10			
server packet	client packet	% packet loss	throughput
3741	3687	1.44%	0.172
3711	3691	0.54%	0.162
3755	3700	1.46%	0.172
3745	3691	1.44%	0.18
3745	3685	1.60%	0.177
		6.49%	0.863
	rata-rata	1.30%	0.1726

6to4 - HELIX - mp4 80,20			
server packet	client packet	% packet loss	throughput
4309	4249	1.39%	0.182
4315	4260	1.27%	0.183
4309	4260	1.14%	0.183
4311	4262	1.14%	0.183
4318	4262	1.30%	0.177
		6.24%	0.908
	rata-rata	1.25%	0.1816

6to4 - HELIX - mp4 80,30			
server packet	client packet	% packet loss	throughput
4971	4918	1.07%	0.185
4946	4893	1.07%	0.191
4962	4909	1.07%	0.192
4983	4926	1.14%	0.197
4952	4904	0.97%	0.294
		5.32%	1.059
	rata-rata	1.06%	0.2118

6to4 - HELIX - mp4 160,10			
server packet	client packet	% packet loss	throughput
4269	4265	0.09%	0.257
4261	4209	1.22%	0.263
4275	4220	1.29%	0.248
4268	4187	1.90%	0.255
4257	4205	1.22%	0.259
		5.72%	1.282
	rata-rata	1.14%	0.2564

server packet	client packet	% packet loss	throughput
4472	4408	1.43%	0.26
4488	4435	1.18%	0.258
4501	4449	1.16%	0.255
4499	4439	1.33%	0.264
4491	4440	1.14%	0.253
		6.24%	1.29
	rata-rata	1.25%	0.258

6to4 - HELIX - mp4 160,30			
server packet	client packet	% packet loss	throughput
5112	5059	1.04%	0.268
5104	5050	1.06%	0.266
5124	5061	1.23%	0.255
5098	5047	1.00%	0.264
5125	5063	1.21%	0.268
		5.53%	1.321
	rata-rata	1.11%	0.2642

6to4 - HELIX - rm 80,10			
server packet	client packet	% packet loss	throughput
1688	1637	3.02%	0.114
1697	1639	3.42%	0.114
1694	1636	3.42%	0.115
1689	1637	3.08%	0.115
1686	1636	2.97%	0.115
		15.91%	0.573
	rata-rata	3.18%	0.1146

6to4 - HELIX - rm 80,20			
server packet	client packet	% packet loss	throughput
2192	2137	2.51%	0.121
2192	2137	2.51%	0.121
2194	2139	2.51%	0.121
2194	2137	2.60%	0.12
2188	2137	2.33%	
		12.45%	0.12
	rata-rata	2.49%	0.024

6to4- HELIX - rm 80,30			
server packet	client packet	% packet loss	throughput
2514	2460	2.15%	0.125
2513	2461	2.07%	0.125
2489	2460	1.17%	0.125
2509	2464	1.79%	0.125
2509	2460	1.95%	0.125
		9.13%	0.625
	rata-rata	1.83%	0.125

**LAMPIRAN 12**  
**DATA PERCOBAAN HELIX JARINGAN MANUALLY CONFIGURED**  
**IPV6 TUNNELING**

manual - HELIX - mp4 80,10			
server packet	client packet	% packet loss	throughput
3694	3689	0.14%	0.175
3687	3683	0.11%	0.174
3680	3674	0.16%	0.174
3696	3692	0.11%	0.166
3686	3682	0.11%	0.176
		0.62%	0.865
	rata-rata	0.12%	0.173

manual - HELIX - mp4 80,20			
server packet	client packet	% packet loss	throughput
4266	4262	0.09%	0.17
4277	4273	0.09%	0.178
4256	4252	0.09%	0.178
4263	4257	0.14%	0.178
4257	4253	0.09%	0.178
		0.52%	0.882
	rata-rata	0.10%	0.1764

manual - HELIX - mp4 80,30			
server packet	client packet	% packet loss	throughput
4907	4903	0.08%	0.182
4915	4911	0.08%	0.191
4899	4895	0.08%	0.193
4916	4912	0.08%	0.194
4908	4904	0.08%	0.193
		0.41%	0.953
	rata-rata	0.08%	0.1906

manual - HELIX - mp4 160,10			
server packet	client packet	% packet loss	throughput
4205	4201	0.10%	0.258
4210	4206	0.10%	0.243
4242	4186	1.32%	0.25
4197	4194	0.07%	0.253
4224	4198	0.62%	0.254
		2.20%	1.258
	rata-rata	0.44%	0.2516

manual - HELIX - mp4 160,20			
server packet	client packet	% packet loss	throughput
4491	4435	1.25%	0.257
4450	4450	0.00%	0.245
4493	4439	1.20%	0.251
4493	4443	1.11%	0.257
4435	4435	0.00%	0.257
		3.56%	1.267
	rata-rata	0.71%	0.2534

manual - HELIX - mp4 160,30			
server packet	client packet	% packet loss	throughput
5128	5070	1.13%	0.265
5070	5069	0.02%	0.261
5139	5080	1.15%	0.262
5130	5076	1.05%	0.261
5131	5079	1.01%	0.259
		4.36%	1.308
	rata-rata	0.87%	0.2616

manual - HELIX - rm 80,10			
server packet	client packet	% packet loss	throughput
1590	1583	0.44%	0.093
1584	1582	0.13%	0.093
1586	1584	0.13%	0.093
1586	1584	0.13%	0.093
1592	1586	0.38%	0.093
		1.20%	0.465
	rata-rata	0.24%	0.093

manual - HELIX - rm 80,20			
server packet	client packet	% packet loss	throughput
2050	2048	0.10%	0.097
1899	1897	0.11%	0.096
1890	1888	0.11%	0.096
1873	1871	0.11%	0.095
1737	1731	0.35%	0.087
		0.76%	0.471
	rata-rata	0.15%	0.0942

manual- HELIX - rm 80,30			
server packet	client packet	% packet loss	throughput
2123	2121	0.09%	0.099
2123	2121	0.09%	0.099
1978	1976	0.10%	0.092
2132	2126	0.28%	0.099
2121	2119	0.09%	0.099
		0.67%	0.488
	rata-rata	0.13%	0.0976