



**UNIVERSITAS INDONESIA**

**ANALISA PERBANDINGAN PERFORMANSI  
JARINGAN IPV4, IPV6 DAN *TUNNELING* 6TO4  
UNTUK APLIKASI *FILE TRANSFER PROTOKOL* (FTP)  
PADA MEDIA *WIRED* DAN *WIRELESS* DI SISI *CLIENT***

**SKRIPSI**

**GILANG RAMADHAN PARAMAYUDHA  
0806365854**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2010**



**UNIVERSITAS INDONESIA**

**ANALISA PERBANDINGAN PERFORMANSI  
JARINGAN IPV4, IPV6 DAN *TUNNELING* 6TO4  
UNTUK APLIKASI *FILE TRANSFER PROTOKOL* (FTP)  
PADA MEDIA *WIRED* DAN *WIRELESS* DI SISI *CLIENT***

**SKRIPSI**

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

**GILANG RAMADHAN PARAMAYUDHA  
0806365854**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2010**

## PERNYATAAN ORISINALITAS

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

Nama : Gilang Ramadhan Paramayudha

NPM : 0806365854

Tanda Tangan :

Tanggal : 30 Juni 2010

## PENGESAHAN

Skripsi ini diajukan oleh :  
Nama : Gilang Ramadhan Paramayudha  
NPM : 0806365854  
Program Studi : Teknik Elektro  
Judul Skripsi : Analisa Perbandingan Performansi Jaringan IPv4, IPv6 dan *Tunneling 6to4* Untuk Aplikasi *File Transfer Protocol (FTP)* Pada Media *Wired* dan *Wireless* Di Sisi *Client*

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

Tanggal : 30 Juni 2010

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada :

1. Ir. A Endang Sriningsih MT., Si, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
2. Orang tua dan keluarga saya yang telah memberikan bantuan dukungan moral.
3. Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 30 Juni 2010

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
SKRIPSI UNTUK KEPENTINGAN AKADEMIS**

---

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

Nama : Gilang Ramadhan Paramayudha  
NPM : 0806365854  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

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

ANALISA PERBANDINGAN PERFORMANSI JARINGAN IPV4, IPV6 DAN  
*TUNNELING* 6TO4 UNTUK APLIKASI *FILE TRANSFER PROTOKOL* (FTP)  
PADA MEDIA *WIRED* DAN *WIRELESS* DI SISI *CLIENT*

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 30 Juni 2010

Yang menyatakan

(.....)

## ABSTRAK

Nama : Gilang Ramadhan Paramayudha  
Program Studi : Teknik Elektro  
Judul : Analisa Perbandingan Performansi Jaringan IPv4, IPv6 dan Tunneling 6to4 Untuk Aplikasi *File Transfer Protocol* (FTP) Pada Media *Wired* dan *Wireless* Di Sisi *Client*

Tujuan dari penulisan skripsi ini adalah untuk menganalisa dan membandingkan performa dari jaringan protokol IPv4, IPv6 dan *tunneling* 6to4 yang diterapkan pada aplikasi FTP (*File Transfer Protocol*) pada media *wired* dan *wireless* di sisi *client*.

Proses pengambilan data menggunakan jaringan lokal sederhana. Dalam pengujian digunakan dua buah *laptop* yang diterapkan sebagai *server* dan *client* serta sebuah PC yang diemulasikan sebagai *router* dengan aplikasi GNS3. Pengambilan data dilakukan dengan cara men-*download* file dengan ukuran yang berbeda - beda dari *server* ke *client*. Untuk *wireless*, jarak uji coba antara *access point* dan *client*  $\pm$  10 m. Parameter uji coba yang dibandingkan adalah *transfer time*, *throughput* dan *delay*.

Konfigurasi IPv4 murni memiliki nilai *transfer time*, *throughput*, dan *delay* paling baik dibandingkan konfigurasi lainnya. Untuk perbandingan *throughput*, jaringan IPv4 *wireless* memiliki nilai *throughput* lebih besar 20.63% dari jaringan IPv4 *wired*, jaringan IPv6 *wireless* juga lebih besar 3.85% dibandingkan dengan IPv6 *wired*, sedangkan jaringan *tunneling* 6to4 lebih besar 28.58% pada *wireless* daripada *wired*. Perbedaan performansi jaringan antara media *wired* dan *wireless*, dipengaruhi kemampuan perangkat *access point* dengan kecepatan 300 Mbps.

Kata Kunci :

IPv6, *File Transfer Protocol*, *Tunneling*, GNS3, *transfer time*, *throughput*, *delay*

## ABSTRACT

Name : Gilang Ramadhan Paramayudha  
Study Program : Electrical Engineering  
Title : Comparative Analysis of Network Performance IPv4, IPv6 and 6to4 Tunneling for Application File Transfer Protocol (FTP) at Wired and Wireless Media at client side

The aim of this paper is to analyze and compare the performance of network protocols IPv4, IPv6 and tunneling 6to4 is applied to the application of FTP (File Transfer Protocol) in wired and wireless media at client side.

The retrieval of data using a simple local network. In the test used by the two laptops are implemented as a server and client and a PC is emulated as a router with GNS3 application. Data collection was done by way of downloading files with different sizes-ranging from server to client. For wireless, client and access point distance  $\pm 10$  m. Test parameters compared are transfer time, throughput, and delay.

A pure IPv4 configuration has a value of transfer time, throughput, dan delay is best compared to other configurations. For comparison of throughput, IPv4 using the wireless network has a throughput of greater value 20.63% of the IPv4 network that uses wired, wireless IPv6 network is also 3.85% larger than IPv6 wired, while for 6to4 tunneling network greater than 28.58% on wireless wired. Differences in performance of the network using wired and wireless media, influence access point performance with 300 Mbps speed of bandwidth.

Keywords :  
IPv6, File Transfer Protocol, Tunneling, GNS3, transfer time, throughput, delay

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
DAFTAR LAMPIRAN .....	xii
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Penulisan .....	2
1.3 Batasan Masalah .....	2
1.4 Metodologi Penulisan .....	3
1.5 Sistematika Penulisan .....	3
<b>BAB 2 PROTOKOL IPV6 DAN FILE TRANSFER</b>	
<b>PROTOKOL (FTP) .....</b>	<b>5</b>
2.1 <i>Internet Protocol Version 6 (IPv6)</i> .....	5
2.1.1 Sejarah dan Latar Belakang Perkembangan IPv6 .....	5
2.1.2 Struktur IPv6 .....	5
2.1.2.1 Fitur-fitur dalam IPv6 .....	5
2.1.2.2 Format <i>Header</i> .....	7
2.1.2.3 Format Alamat .....	9
2.1.2.4 Format Prefiks .....	11
2.1.3 Perbandingan IPv6 dengan IPv4 .....	11
2.2 Mekanisme <i>Tunneling</i> .....	14
2.2.1 <i>Tunneling Manual</i> .....	15
2.2.2 <i>Tunneling Otomatis</i> .....	15
2.2.2.1 <i>Tunneling 6to4</i> .....	16
2.3 Media Transmisi .....	17
2.3.1 <i>Twisted-Pair Cable</i> .....	17
2.3.1.1 <i>Unshield Twisted Pair (UTP) Cable</i> .....	18
2.3.2 <i>Wireless</i> .....	19
2.3.2.1 <i>Standarisasi Wireless</i> .....	19
2.4 <i>File Transfer Protocol (FTP)</i> .....	20
2.4.1 <i>Anonymous</i> .....	21
2.4.2 <i>User Legal (Authenticated User)</i> .....	22

<b>BAB 3 PERANCANGAN TOPOLOGI JARINGAN</b> .....	23
3.1 Perancangan Topologi Jaringan .....	23
3.2 Konfigurasi Jaringan <i>Wired</i> .....	25
3.2.1 Konfigurasi IPv4 <i>Wired</i> .....	26
3.2.2 Konfigurasi IPv6 <i>Wired</i> .....	27
3.2.3 Konfigurasi IPv6 <i>Tunneling 6to4 Wired</i> .....	28
3.3 Konfigurasi Jaringan <i>Wireless</i> .....	29
3.3.1 Konfigurasi IPv4 <i>Wireless</i> .....	29
3.3.2 Konfigurasi IPv6 <i>Wireless</i> .....	30
3.3.3 Konfigurasi IPv6 <i>Tunneling 6to4 Wireless</i> .....	31
3.4 Metode Pengambilan Data .....	32
<b>BAB 4 ANALISA DATA</b> .....	34
4.1 Analisa Konfigurasi Jaringan .....	34
4.1.1. Konfigurasi IPv4 Murni .....	34
4.1.2. Konfigurasi IPv6 Murni .....	35
4.1.3. Konfigurasi <i>Tunneling 6to4</i> .....	35
4.2 Analisa Performa Jaringan Pada FTP .....	36
4.2.1. Analisa <i>Transfer Time</i> .....	39
4.2.2. Analisa <i>Throughput</i> .....	44
4.2.3. Analisa <i>Delay</i> .....	49
<b>BAB 5 KESIMPULAN</b> .....	54
<b>DAFTAR REFERENSI</b> .....	55

## DAFTAR GAMBAR

Gambar 2.1	Struktur Dari Paket IPv6 .....	7
Gambar 2.2	Perbandingan Format <i>Header</i> IPv4 Dengan IPv6.....	7
Gambar 2.3	<i>Tunneling</i> IPv6 Over IPv4 .....	14
Gambar 2.4	Enkapsulasi Paket Pada Proses <i>Tunneling</i> .....	15
Gambar 2.5	Contoh Konfigurasi Pengalamatan 6to4 .....	16
Gambar 2.6	Format Alamat 6to4 .....	17
Gambar 2.7	Konektor RJ-45 <i>Cable</i> .....	18
Gambar 3.1	Tampilan Topologi Umum Jaringan <i>Test-bed</i> Pada GNS3 .....	23
Gambar 3.2	Topologi Jaringan <i>Test-bed</i> IPv4 <i>Wired</i> .....	26
Gambar 3.3	Topologi Jaringan <i>Test-bed</i> IPv6 <i>Wired</i> .....	27
Gambar 3.4	Topologi Jaringan <i>Test-bed</i> IPv6 <i>Tunneling</i> 6to4 <i>Wired</i> .....	28
Gambar 3.5	Topologi Jaringan <i>Test-bed</i> IPv4 <i>Wireless</i> .....	29
Gambar 3.6	Topologi Jaringan <i>Test-bed</i> IPv6 <i>Wireless</i> .....	30
Gambar 3.7	Topologi Jaringan <i>Test-bed</i> IPv6 <i>Tunneling</i> 6to4 <i>Wireless</i> .....	31
Gambar 4.1	Tampilan <i>Traceroute</i> Pada Jaringan IPv4 Murni.....	35
Gambar 4.2	Tampilan <i>Traceroute</i> Pada Jaringan IPv6 Murni .....	35
Gambar 4.3	Tampilan <i>Traceroute</i> Pada Jaringan <i>Tunneling</i> 6to4.....	36
Gambar 4.4	Aplikasi Xlight FTP <i>Server</i> .....	37
Gambar 4.5	Tampilan Koneksi FTP <i>Server</i> Dengan <i>Client</i> .....	38
Gambar 4.6	Pengambilan Nilai <i>Transfer Time</i> .....	40
Gambar 4.7	Diagram Perbandingan <i>Transfer Time</i> Untuk <i>File Size</i> 5M ....	42
Gambar 4.8	Diagram Perbandingan <i>Transfer Time</i> Untuk <i>File Size</i> 10M ...	42
Gambar 4.9	Diagram Perbandingan <i>Transfer Time</i> Untuk <i>File Size</i> 15M ...	43
Gambar 4.10	Diagram Perbandingan <i>Transfer Time</i> Untuk <i>File Size</i> 20M ...	43
Gambar 4.11	Pengambilan Data <i>Throughput</i> .....	45
Gambar 4.12	Diagram Perbandingan <i>Throughput</i> Untuk <i>File Size</i> 5M .....	47
Gambar 4.13	Diagram Perbandingan <i>Throughput</i> Untuk <i>File Size</i> 10M .....	47
Gambar 4.14	Diagram Perbandingan <i>Throughput</i> Untuk <i>File Size</i> 15M .....	48
Gambar 4.15	Diagram Perbandingan <i>Throughput</i> Untuk <i>File Size</i> 20M .....	48
Gambar 4.16	Diagram Perbandingan <i>Delay</i> Untuk <i>File Size</i> 5M.....	51
Gambar 4.17	Diagram Perbandingan <i>Delay</i> Untuk <i>File Size</i> 10M .....	51
Gambar 4.18	Diagram Perbandingan <i>Delay</i> Untuk <i>File Size</i> 15M .....	52
Gambar 4.19	Diagram Perbandingan <i>Delay</i> Untuk <i>File Size</i> 20M .....	52

## DAFTAR TABEL

Tabel 2.1	Penyederhanaan Bentuk Alamat Pada IPv6 .....	10
Tabel 2.2	Perbandingan Antara IPv4 Dengan IPv6 .....	12
Tabel 2.3	Ekivalensi Pada IPv6 Dengan IPv4.....	13
Tabel 2.4	Spesifikasi WLAN.....	20
Tabel 2.5	<i>Assigned Port</i> .....	20
Tabel 4.1	Nama File dan Kapasitasnya .....	39
Tabel 4.2	Data Percobaan Nilai Rata – rata <i>Transfer Time</i> .....	40
Tabel 4.3	Nilai Rata – rata Percobaan <i>Throughput</i> .....	45
Tabel 4.4	Nilai Rata – rata Percobaan <i>Delay</i> .....	50

## DAFTAR LAMPIRAN

Lampiran 1	Konfigurasi <i>Router</i> Pada Jaringan IPv4 Murni .....	56
Lampiran 2	Konfigurasi <i>Router</i> Pada Jaringan IPv6 Murni .....	58
Lampiran 3	Konfigurasi <i>Router</i> Pada Jaringan <i>Tunneling</i> 6to4 .....	60
Lampiran 4	Hasil <i>Capture</i> Wireshark Konfigurasi IPv4 Murni .....	63
Lampiran 5	Hasil <i>Capture</i> Wireshark Konfigurasi IPv6 Murni .....	64
Lampiran 6	Hasil <i>Capture</i> Wireshark Konfigurasi <i>Tunneling</i> 6to4 .....	64
Lampiran 7	Data Hasil Uji Coba Konfigurasi IPv4 Murni Dengan Media <i>Wired</i> .....	66
Lampiran 8	Data Hasil Uji Coba Konfigurasi IPv6 Murni Dengan Media <i>Wired</i> .....	67
Lampiran 9	Data Hasil Uji Coba Konfigurasi <i>Tunneling</i> 6to4 Dengan Media <i>Wired</i> .....	68
Lampiran 10	Data Hasil Uji Coba Konfigurasi IPv4 Murni Dengan Media <i>Wireless</i> .....	69
Lampiran 11	Data Hasil Uji Coba Konfigurasi IPv6 Murni Dengan Media <i>Wireless</i> .....	70
Lampiran 12	Nilai Data Hasil Uji Coba Konfigurasi <i>Tunneling</i> 6to4 Dengan Media <i>Wireless</i> .....	71

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Seiring dengan berkembangnya teknologi pada zaman sekarang ini, terutama teknologi komunikasi berbasis *Internet Protocol* (IP). Sehingga memungkinkan banyaknya penggunaan *Internet Protocol Version 4* (IPv4) yaitu teknologi yang digunakan saat ini akan mengalami masalah pada pengalamatan yang semakin terbatas. Karena keterbatasan jumlah pengalamatan pada IPv4 ini, sehingga dibutuhkan suatu standar baru pada *routing protocol* yang mampu mengakomodasi jumlah pengalamatan yang lebih banyak. Berangkat dari masalah ini, *Internet Protocol version 6* (IPv6) kemudian dikembangkan dan dijadikan standar baru yang kelak akan mengakomodasi pengalamatan yang lebih banyak.

IPv6 adalah protokol generasi baru yang didesain oleh *Internet Engineering Task Force* (IETF) untuk menggantikan protokol internet saat ini, yaitu IPv4. Alamat IPv4 pada dasarnya menggunakan metode pengalamatan berbasis 32 bit, yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan  $2^{32}$  atau sekitar  $4,294 \times 10^9$ . Sedangkan IPv6 menggunakan metode pengalamatan berbasis 128 bit, yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan  $2^{128}$  atau sekitar  $3,402 \times 10^{38}$ . Dengan perbandingan jumlah pengalamatan yang begitu besar inilah, yang mendasari perubahan dari IPv4 menjadi IPv6.

IPv6 merupakan IP generasi berikutnya atau disebut juga *Internet Protocol Next Generation* (IPng). IPv6 dirancang sedemikian rupa agar memiliki kinerja yang lebih handal bila dibandingkan dengan IPv4 seperti dalam pengiriman paket, *security, authentication* dan *QoS (Quality Of Service)*. Selain itu diharapkan IPv6 juga mampu memberikan fitur-fitur lain yang lebih kompleks yang akan dikembangkan lagi.

Sampai saat ini, secara umum jaringan masih menggunakan IPv4 sehingga implementasi jaringan IPv6 dilakukan secara bertahap dan diusahakan tidak akan mengganggu jaringan IPv4 yang sudah ada saat ini. Oleh karena itu, diperlukan suatu mekanisme transisi untuk mengganti penggunaan jaringan IPv4 menjadi

jaringan IPv6 secara keseluruhan. Untuk melakukan proses transisi dari IPv4 ke IPv6 maka diperlukan suatu metode yang mampu menunjang mekanisme transisi tersebut. Beberapa metode telah diteliti dan diantaranya yaitu metode *Dual Stack*, *Tunneling* dan *Translation*.

Salah satu aplikasi yang akan diimplementasikan sekaligus menjadi uji coba dalam IPv6 adalah *transfer file* yang menggunakan *protocol* yang disebut *File Transfer Protocol* (FTP). Kebutuhan akan proses *upload* dan *download* ke suatu *server* menjadikan aplikasi ini sering dipergunakan.

### 1.2 Tujuan Penulisan

Penulisan skripsi ini bertujuan untuk menganalisa dan membandingkan performa dari aplikasi *File Transfer Protocol* (FTP) yang diterapkan pada jaringan protokol IPv4 murni, IPv6 murni, dan *tunneling* 6to4 pada media *wired* dan *wireless* di sisi *client*.

### 1.3 Batasan Masalah

Pengujian dilakukan menggunakan jaringan sederhana yang difokuskan pada aplikasi *transfer file* dan diterapkan pada konfigurasi jaringan IPv4 murni, jaringan IPv6 murni dan metode *tunneling* 6to4.

Rancangan jaringan sederhana yang akan dibangun adalah jaringan lokal yang menggunakan aplikasi GNS3 sebagai perangkat *emulator* yang berfungsi sebagai *router*. Dimana jaringan tersebut terdiri dari dua buah *Mobile Computer* sebagai *client* dan *server* yang terhubung dengan *PC Router* yang menggunakan *emulator* GNS3.

Percobaan dilakukan pada media yang berbeda yaitu media yang menggunakan *wired* secara keseluruhan dan media yang menggunakan *wireless* pada sisi *client*. Pada percobaan media *wireless* digunakan *access point* pada sisi *client* dengan jarak antara *access point* dengan *client*  $\pm 10$  m.

Aplikasi yang akan diimplementasikan pada jaringan *test bed* adalah aplikasi Xlight pada sisi *server*. Parameter-parameter yang akan dianalisa untuk menguji kualitas jaringan meliputi *transfer time*, *throughput*, dan *delay*.

#### 1.4 Metodologi Penulisan

1. Studi literatur

Mengumpulkan dan mempelajari referensi tentang jaringan IPv6, *tunneling*, FTP, aplikasi Wireshark, Xlight.

2. Perancangan sistem

Pada skripsi ini dirancang sistem agar perangkat keras yang diperlukan dapat berjalan lancar dengan perangkat lunak yang diaplikasikan ke dalam perangkat keras tersebut.

3. Pembuatan sistem

Pembuatan sistem dapat dilakukan setelah semua perangkat keras dan lunak telah terpenuhi. Permasalahan perangkat keras berupa *router*, dapat diselesaikan dengan aplikasi GNS3 sebagai *emulator*.

4. Pengambilan dan analisa data

Setelah dilakukan implementasi, akan dicatat data-data yang berhubungan dengan *file transfer protocol* (ftp) pada masing-masing jaringan. Dan parameter yang diambil adalah *transfer time*, *throughput*, dan *delay*.

5. Penarikan kesimpulan

Penarikan kesimpulan dapat diambil dari topologi masing – masing jaringan yang akan dibuat.

#### 1.5 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut :

BAB 1 Pendahuluan

Bab ini berisi penjelasan singkat mengenai Latar Belakang, Tujuan, Batasan Masalah, Metodologi dan Sistematika Penulisan.

BAB 2 Protokol IPv6 Dan *File Transfer Protocol* (FTP)

Bab ini berisi penjelasan mengenai konsep IPv6, *tunneling* 6to4, *wireless*, FTP.

BAB 3 Perancangan Topologi Jaringan

Bab ini berisi penjelasan mengenai perancangan topologi jaringan dan konfigurasi jaringan.

BAB 4 Analisa Data

Bab ini berisi proses pengambilan dan analisa data pada masing – masing jaringan untuk aplikasi *file transfer protocol* (ftp).

BAB 5 Kesimpulan

Bab ini berisi kesimpulan dari skripsi yang dilakukan.



## BAB 2

### PROTOKOL IPV6 DAN *FILE TRANSFER PROTOCOL* (FTP)

#### 2.1 *Internet Protocol Version 6* (IPv6)

##### 2.1.1 Sejarah dan Latar Belakang Perkembangan IPv6

Aplikasi internet saat ini kebanyakan masih menggunakan *Internet Protocol version 4* (IPv4) yang menggunakan metode pengalamatan berbasis 32 bit, yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan  $2^{32}$  atau sekitar  $4,294 \times 10^9$ . *Internet Protocol version 4* (IPv4) yang dijadikan standar *routing protocol* pada tahun 1981 (RFC 791) sebenarnya telah terbukti tangguh, mudah diimplementasikan dan dioperasikan, dan telah berhasil melalui tes skalabilitas dalam jaringan internet secara global sampai saat ini. Namun *design* dari IPv4 tidak mengantisipasi dampak yang mungkin ditimbulkan akibat perkembangan jaringan internet yang semakin pesat, diantaranya :

- a. Membutuhkan jumlah pengalamatan yang lebih banyak untuk mendukung kebutuhan perkembangan jaringan internet dimasa yang akan datang.
- b. Membutuhkan kemampuan dari *router backbone* internet dalam mengelola tabel *routing* yang besar.
- c. Membutuhkan konfigurasi yang lebih sederhana.
- d. Membutuhkan keamanan pada level IP (*Internet Protocol*).
- e. Membutuhkan dukungan yang lebih baik untuk pengiriman data secara *real-time*, disebut juga dengan *Quality Of Service* (QoS). [1]

##### 2.1.2 Struktur IPv6

###### 2.1.2.1 Fitur – fitur dalam IPv6

IPv6 didukung oleh fitur-fitur yang diharapkan dapat mengatasi kekurangan-kekurangan yang ada pada *routing protocol* IPv4, beberapa diantaranya :

- a. Format *header* yang baru

*Header* IPv6 dua kali lebih besar dari *header* IPv4, sedangkan pengalamatan IPv6 empat kali lebih besar dari pengalamatan IPv4.

Walaupun demikian, format *header* IPv6 justru mengalami penyederhanaan dimana ada beberapa *field* yang dihilangkan karena dianggap tidak efisien. Sebagai gantinya ditambahkan *header* tambahan yang disebut *extension header*.

b. Perluasan format pengalamatan

IPv6 mempunyai format pengalamatan 128-bit (16-byte) yang berarti mampu mengakomodasi jumlah pengalamatan sampai dengan  $2^{128}$  atau sekitar  $3,402 \times 10^{38}$ .

c. Infrastruktur *routing* dan pengalamatan yang lebih efisien dan berbentuk hierarki

d. Konfigurasi alamat *stateless* dan *stateful*

Sama seperti halnya IPv4, pada IPv6 juga mengizinkan adanya DHCP *server* sebagai pengatur alamat otomatis. Jika dalam IPv4 terdapat *dynamic address* dan *static address*, maka dalam IPv6, konfigurasi alamat dengan menggunakan DHCP *server* dinamakan dengan *stateful address configuration*, sementara jika konfigurasi alamat IPv6 tanpa DHCP *server* dinamakan dengan *stateless address configuration*

e. Keamanan yang lebih baik

IPv6 telah dirancang untuk mendukung IPSec sehingga bisa dikatakan IPv6 memiliki keamanan yang lebih baik bila dibandingkan dengan IPv4

f. Mendukung Quality of Services (QoS)

*Field* baru yang berada pada IPv6 *header* mendefinisikan bagaimana trafik khusus ditangani dan diidentifikasi. Identifikasi trafik yang memerlukan penanganan khusus menggunakan *field flow lable*

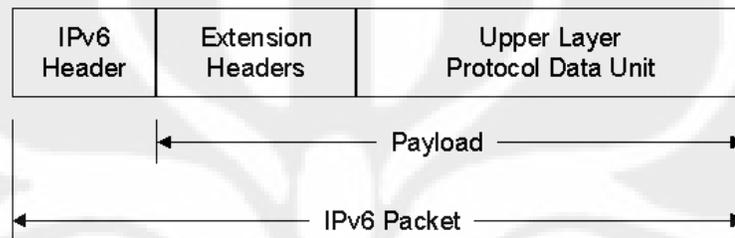
g. Ekstensibilitas

IPv6 dapat dengan mudah memperluas fitur baru dengan menambahkan *extention header* setelah IPv6 *header*. [1]

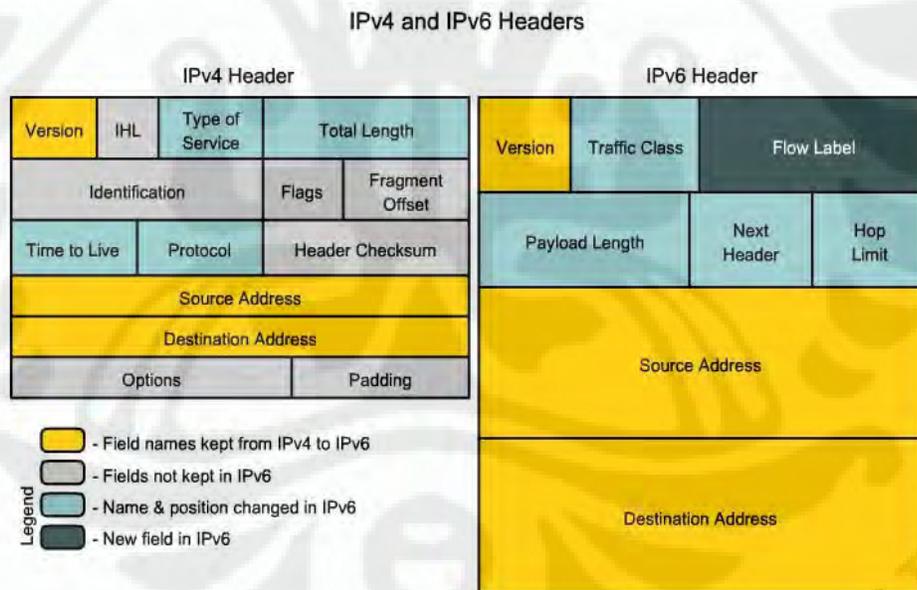
Fitur-fitur diatas hanya sedikit contoh dari fitur-fitur yang dimiliki oleh IPv6 yang mampu mendukung IPv6 agar dapat menjadi sebuah *routing protocol* yang handal dimasa yang akan datang.

### 2.1.2.2 Format Header

Format *header* IPv6 terbagi menjadi 2 bagian, yaitu *header standart (default)* dan *header tambahan (extension)*. *Header standart (default)* merupakan *field-field* yang selalu ada dalam setiap paket IPv6, sedangkan *header tambahan (extension)* merupakan *field-field* yang ditambahkan hanya apabila dibutuhkan saja, terletak diantara *header standart* IPv6 dengan *upper-layer header*. Pada Gambar 2.1 berikut ini ditunjukkan struktur dari paket IPv6.



Gambar 2.1 Struktur Dari Paket IPv6 [1]



Gambar 2.2 Perbandingan Format Header IPv4 Dengan IPv6 [2]

Dari Gambar 2.2 ditunjukkan bahwa *header* IPv6 mempunyai ukuran yang lebih besar bila dibandingkan dengan *header* IPv4 sehingga mampu mendukung jumlah pengalamatan sampai dengan 128 bit. Meskipun demikian, format *header*

IPv6 lebih sederhana bila dibandingkan dengan format *header* IPv4 karena ada beberapa *field* pada *header* IPv4 yang dihilangkan pada *header* IPv6.[2]

Di bawah ini akan dijelaskan format *header* IPv6 pada masing – masing bagiannya :

☒ *Version*

Merupakan 4 bit *field* yang menunjukkan versi dari suatu protokol IP dimana untuk paket IPv6 diinisialisasi dengan nilai 6.

☒ *Traffic Class*

Merupakan 8 bit *field* yang mengindikasikan skala prioritas pada paket atau mengidentifikasi paket – paket yang membutuhkan penanganan khusus.

☒ *Flow Label*

Bagian yang terdiri dari 20 bit yang berfungsi mengidentifikasi paket – paket *real time* yang dianggap memiliki alur data yang sama.

☒ *Payload Length*

Merupakan 16 bit *field* yang berfungsi untuk menunjukkan panjang bit data yang dibawa pada paket IPv6.

☒ *Next Header*

Merupakan 8 bit *field* yang menunjukkan jenis protokol dari paket IPv6 yang dikirimkan tersebut. Selain itu, bisa berfungsi untuk identifikasi adanya *extension header* pada paket IPv6.

☒ *Hop Limit*

Merupakan 8 bit *field* yang menunjukkan jumlah hop yang akan dilewati oleh suatu paket IPv6 yang dikirimkan.

☒ *Source Address*

Merupakan 128 bit *field* yang menunjukkan alamat pengirim atau sumber dari paket IPv6 yang telah dikirimkan.

☒ *Destination Address*

Merupakan 128 bit *field* yang menunjukkan alamat penerima dari paket IPv6 yang telah dikirimkan.

☒ *Optional Extension Headers*

Merupakan *header* tambahan yang berfungsi sebagai informasi tambahan.

### 2.1.2.3 Format Alamat

Alamat IPv6 berbeda dengan IPv4 yang hanya memiliki panjang 32-bit (jumlah total alamat yang dapat dicapainya mencapai 4,294,967,296 alamat), alamat IPv6 memiliki panjang 128-bit. IPv6, yang memiliki panjang 128-bit, memiliki total alamat yang mungkin hingga  $2^{128}=3,4 \times 10^{38}$  alamat. Total alamat yang sangat besar ini bertujuan untuk menyediakan ruang alamat yang tidak akan habis (hingga beberapa masa ke depan), dan membentuk infrastruktur *routing* yang disusun secara hierarkis, sehingga mengurangi kompleksitas proses *routing* dan *tabel routing*. [1]

Dalam IPv6, alamat 128-bit akan dibagi ke dalam 8 blok berukuran 16-bit, yang dapat dikonversikan ke dalam bilangan heksadesimal berukuran 4-digit. Setiap blok bilangan heksadesimal tersebut akan dipisahkan dengan tanda titik dua (:). Karenanya, format notasi yang digunakan oleh IPv6 juga sering disebut dengan *colon-hexadecimal format*, berbeda dengan IPv4 yang menggunakan *dotted-decimal format*.

Berikut ini adalah contoh alamat IPv6 dalam bentuk bilangan biner :

```
00100001110110100000000011010011000000000000000001011110011101100
0000101010101000000000111111111111110001010001001110001011010
```

Untuk menerjemahkannya ke dalam bentuk notasi *colon-hexadecimal format*, angka-angka biner di atas harus dibagi ke dalam 8 buah blok berukuran 16-bit:

```
0010000111011010 0000000011010011 0000000000000000 0010111100111011
0000001010101010 0000000011111111 1111111000101000 1001110001011010
```

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

```
21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A [1]
```

Alamat di atas juga dapat disederhanakan lagi dengan membuang angka 0 pada awal setiap blok yang berukuran 16-bit di atas, dengan menyisakan satu digit terakhir. Dengan membuang angka 0, alamat di atas disederhanakan menjadi:

```
21DA:D3:0:2F3B:2AA:FF:FE28:9C5A
```

Konvensi pengalamatan IPv6 juga mengizinkan penyederhanaan alamat lebih jauh lagi, yakni dengan membuang banyak karakter 0, pada sebuah alamat yang banyak angka 0-nya. Jika sebuah alamat IPv6 yang direpresentasikan dalam notasi *colon-hexadecimal format* mengandung beberapa blok 16-bit dengan angka 0, maka alamat tersebut dapat disederhanakan dengan menggunakan tanda dua buah titik dua (::). Untuk menghindari kebingungan, penyederhanaan alamat IPv6 dengan cara ini sebaiknya hanya digunakan sekali saja di dalam satu alamat, karena kemungkinan nantinya pengguna tidak dapat menentukan berapa banyak bit 0 yang direpresentasikan oleh setiap tanda dua titik dua (::) yang terdapat dalam alamat tersebut. Tabel 2.1 berikut mengilustrasikan cara penggunaan hal ini.

Tabel 2.1 Penyederhanaan Bentuk Alamat Pada IPv6 [1]

Alamat asli	Alamat asli yang disederhanakan	Alamat setelah dikompres
FE80:0000:0000:0000:02BB:00A D:FE9A:4CB7	FE80:0:0:0:2BB:AD:FE 9A:4CB7	FE80::2BB:AD:FE9 A:4CB7
FF02:0000:0000:0000:0000:0000: 0000:0002	FF02:0:0:0:0:0:2	FF02::2

Untuk menentukan berapa banyak bit bernilai 0 yang dibuang (dan digantikan dengan tanda dua titik dua) dalam sebuah alamat IPv6, dapat dilakukan dengan menghitung berapa banyak blok yang tersedia dalam alamat tersebut, yang kemudian dikurangkan dengan angka 8, dan angka tersebut dikalikan dengan 16. Sebagai contoh, alamat FF02::2 hanya mengandung dua blok alamat (blok FF02 dan blok 2). Maka, jumlah bit yang dibuang adalah  $(8-2) \times 16 = 96$  buah bit. [1]

IPv6 mendukung beberapa jenis format *prefix*, yaitu sebagai berikut:

1. Alamat *Unicast*, yang menyediakan komunikasi secara *point-to-point*, secara langsung antara dua *host* dalam sebuah jaringan.

2. Alamat *Multicast*, yang menyediakan metode untuk mengirimkan sebuah paket data ke banyak *host* yang berada dalam *group* yang sama. Alamat ini digunakan dalam komunikasi *one-to-many*.
3. Alamat *Anycast*, yang menyediakan metode penyampaian paket data kepada anggota terdekat dari sebuah *group*. Alamat ini digunakan dalam komunikasi *one-to-one-of-many*. Alamat ini juga digunakan hanya sebagai alamat tujuan (*destination address*) dan diberikan hanya kepada *router*, bukan kepada *host-host* biasa. [1]

#### 2.1.2.4 Format Prefiks

Dalam IPv4, sebuah alamat dalam notasi *dotted-decimal format* dapat direpresentasikan dengan menggunakan angka prefiks yang merujuk kepada *subnet mask*. IPv6 juga memiliki angka prefiks, tapi tidak digunakan untuk merujuk kepada *subnet mask*, karena memang IPv6 tidak mendukung *subnet mask*.

Prefiks adalah sebuah bagian dari alamat IP, di mana bit-bit memiliki nilai-nilai yang tetap atau bit-bit tersebut merupakan bagian dari sebuah *route* atau *subnet identifier*. Prefiks dalam IPv6 direpresentasikan dengan cara yang sama seperti halnya prefiks alamat IPv4, yaitu **[alamat]/[angka panjang prefiks]**. Panjang prefiks menentukan jumlah bit terbesar paling kiri yang membuat prefiks *subnet*. Sebagai contoh, prefiks sebuah alamat IPv6 dapat direpresentasikan sebagai berikut:

3ABE:2700:DA05:C22C::/64

Pada contoh di atas, 64 bit pertama dari alamat tersebut dianggap sebagai prefiks alamat, sementara 64 bit sisanya dianggap sebagai *interface ID*. [1]

#### 2.1.3 Perbandingan IPv6 dengan IPv4

Dengan adanya *routing protocol* baru IPv6 diharapkan mampu mengatasi kekurangan-kekurangan yang ada pada IPv4. Secara garis besar, perbandingan yang mendasar antara IPv4 dengan IPv6 ditunjukkan pada Tabel 2.2 berikut ini :

Tabel 2.2 Perbandingan Antara IPv4 Dengan IPv6 [1]

IPv4	IPv6
Format pengalamatan 32 bit (4 byte).	Format pengalamatan 128 bit (16 byte).
Dukungan IPsec merupakan pilihan.	Mendukung IPsec.
Tidak ada identifikasi aliran paket untuk penanganan QoS oleh <i>router</i> yang dimasukkan ke dalam <i>header</i> IPv4.	Identifikasi aliran paket untuk penanganan QoS oleh <i>router</i> dimasukkan ke dalam <i>header</i> IPv6 menggunakan <i>Flow Label field</i> .
Fragmentasi dilakukan oleh kedua <i>router</i> dan <i>host</i> pengirim.	Fragmentasi tidak dilakukan oleh <i>router</i> , hanya oleh <i>host</i> pengirim.
<i>Header</i> mengandung <i>checksum</i> .	<i>Header</i> tidak mengandung <i>checksum</i> .
<i>Header</i> mengandung <i>options</i> .	Semua <i>options</i> dipindahkan ke <i>extension headers</i> .
<i>Address Resolution Protocol</i> (ARP) menggunakan <i>frame Broadcast ARP Request</i> untuk mengubah alamat IPv4 menjadi alamat <i>link layer</i> .	ARP diganti dengan <i>Multicast Neighbor Solicitation messages</i> .
<i>Internet Group Management Protocol</i> (IGMP) digunakan untuk mengatur kumpulan <i>local subnet group</i> .	IGMP diganti dengan <i>Multicast Listener Discovery (MLD) messages</i> .
ICMP <i>Router Discovery</i> digunakan untuk mencari <i>default gateway</i> yang terbaik dan hal ini merupakan pilihan	ICMP <i>Router Discovery</i> diganti dengan ICMPv6 <i>Router Solicitation</i> dan <i>Router Advertisement messages</i> , dan hal ini memang dibutuhkan.
Terdapat alamat <i>broadcast</i> .	Tidak ada alamat <i>broadcast</i> .
Harus dikonfigurasi secara manual ataupun melalui DHCP.	Tidak membutuhkan konfigurasi secara manual maupun DHCP.
Menggunakan alamat <i>host</i> (A) yang tercatat pada <i>Domain Name System</i> (DNS) untuk memetakan <i>hostname</i> pada alamat IPv4.	Menggunakan alamat <i>host</i> (AAAA) yang tercatat pada <i>Domain Name System</i> (DNS) untuk memetakan <i>hostname</i> pada alamat IPv6.

Menggunakan <i>pointer</i> (PTR) yang tercatat pada <i>domain</i> IN-ADDR.ARPA DNS untuk memetakan alamat IPv4 ke <i>hostname</i> .	Menggunakan <i>pointer</i> (PTR) yang tercatat pada <i>domain</i> IP6.ARPA DNS untuk memetakan alamat IPv6 ke <i>hostname</i> .
Mendukung sampai 576-byte ukuran paket.	Mendukung sampai 1280-byte ukuran paket.

Sumber: Microsoft Corporation, 2008

Walaupun ada perbedaan antara IPv6 dengan IPv4, namun ada beberapa konsep pengalamatan yang masih mempunyai kesamaan diantara keduanya. Beberapa hal yang ekuivalen antara IPv6 dengan IPv4 ditunjukkan pada Tabel 2.3 berikut ini.

Tabel 2.3 Ekuivalensi Pada IPv6 Dengan IPv4 [1]

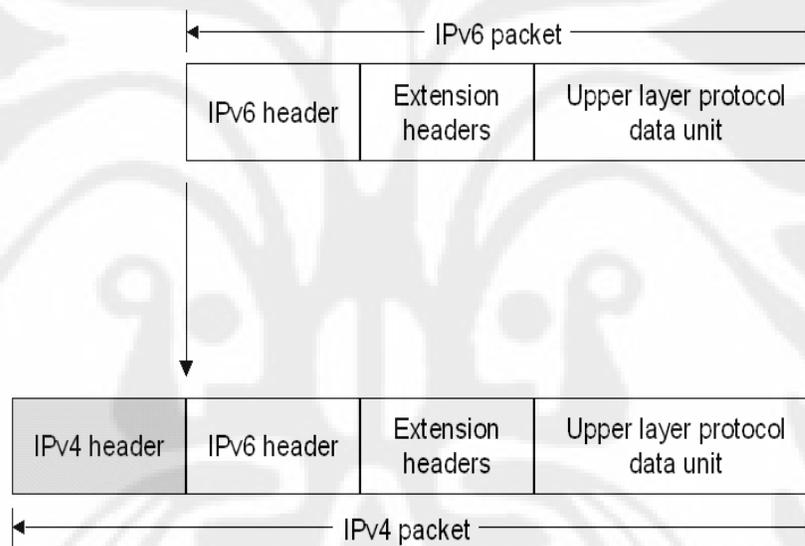
IPv4 Address	IPv6 Address
<i>Internet address classes</i> (Class A, B, C, D, E)	Dihilangkan pada IPv6
<i>Multicast addresses</i> (224.0.0.0/4)	<i>Multicast addresses</i> (FF00::/8)
<i>Broadcast addresses</i>	Tidak digunakan pada IPv6
<i>Unspecified address</i> (0.0.0.0)	<i>Unspecified address</i> (::)
<i>Loopback address</i> (127.0.0.1)	<i>Loopback address</i> (::1)
<i>Public IP addresses</i>	<i>Global unicast addresses</i>
<i>Private IP addresses</i> (10.0.0.0/8, 172.16.0.0/12, dan 192.168.0.0/16)	<i>Site-local addresses</i> (FEC0::/10)
<i>Autoconfigured addresses</i> (169.254.0.0/16)	<i>Link-local addresses</i> (FE80::/64)
Penulisan IP menggunakan <i>format dotted decimal</i> (.)	Penulisan IP menggunakan <i>format colon hexadecimal</i> (:)
Penulisan <i>subnet mask</i> menggunakan <i>format dotted decimal</i> atau <i>prefix length</i>	Penulisan <i>subnet mask</i> hanya menggunakan <i>format prefix length</i>

Sumber: Microsoft Corporation, 2008

## 2.2 Mekanisme *Tunneling*

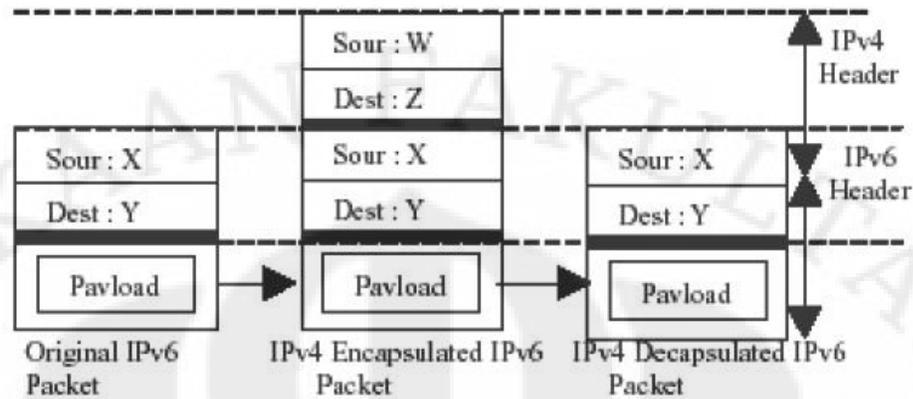
IPv6 mempunyai format alamat dan *header* yang berbeda dengan IPv4 sehingga tidak bisa melakukan interkoneksi dengan IPv4 secara langsung. Oleh karena itu, diperlukan suatu mekanisme transisi IPv6 agar paket IPv6 dapat dilewatkan pada jaringan IPv4 yang telah ada ataupun sebaliknya. Salah satu contoh mekanisme transisi adalah metode *Tunneling*.

*Tunneling protocol* merupakan mekanisme proses enkapsulasi suatu *network protocol* yang disebut *payload protocol* kedalam *delivery protocol* yang berbeda. [4] *Tunneling IPv6 over IPv4* merupakan suatu proses enkapsulasi paket IPv6 dengan *header* IPv4 sehingga paket IPv6 dapat dikirim melalui jaringan IPv4. Struktur *tunneling* IPv6 pada IPv4 ditunjukkan pada Gambar 2.3 berikut ini.



Gambar 2.3 *Tunneling IPv6 Over IPv4*

Selama proses *tunneling* IPv6 pada IPv4 berlangsung maka akan terjadi proses enkapsulasi dan dekapsulasi paket IPv6 oleh IPv4. Proses enkapsulasi dan dekapsulasi paket IPv6 ditunjukkan pada Gambar 2.4 berikut ini.



Gambar 2.4 Enkapsulasi Paket Pada Proses *Tunneling*

### 2.2.1 *Tunneling* Manual

*Tunneling* manual dapat didefinisikan sebagai suatu teknik *tunneling* dimana ujung – ujung *interface tunnel* dikonfigurasi secara eksplisit, baik oleh manusia (administrasi jaringan) maupun melalui sebuah layanan otomatis yang disebut *tunnel broker*.

*Tunneling* manual merupakan suatu teknik *tunneling* yang menggunakan enkapsulasi UDP (*User Datagram Protocol*). Teknik *tunneling* manual dapat dilakukan dengan cara mengatur *dual address* pada tiap-tiap *endpoints router* (*dual stack*). Artinya *router* mampu meneruskan paket baik paket IPv4 ataupun paket IPv6.

*Tunneling* manual mudah untuk diimplementasikan pada suatu jaringan namun keterbatasannya terdapat pada keamanannya. Selain itu pada teknik *tunneling* manual sangat bergantung pada peran administrator untuk mengkonfigurasinya ketika terjadi perubahan topologi jaringan.

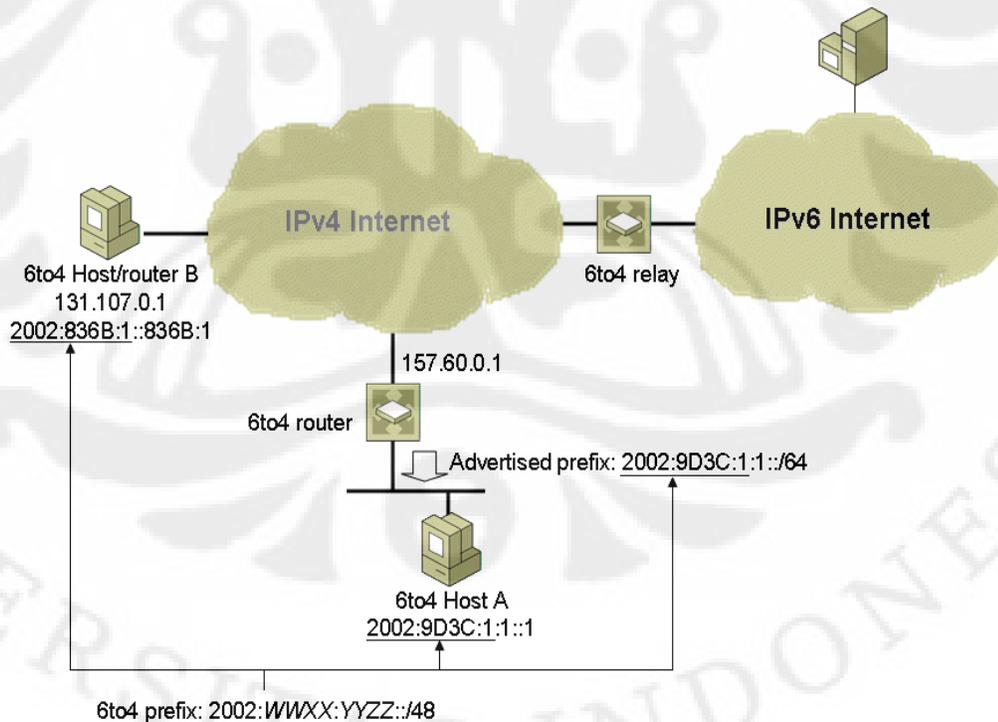
### 2.2.2 *Tunneling* Otomatis

Pada *tunneling* otomatis tiap-tiap *interface tunnel* memperoleh alamat atau *prefix* IPv6 berdasarkan format alamat IPv4 yang telah sesuai dengan konfigurasi *networknya*. Artinya *prefix* alamat IPv6 merupakan identitas unik yang dapat diperoleh dengan cara pengintegrasian alamat IPv4 dengan cara perubahan format bilangan dari biner menjadi heksadesimal. [3]

Koneksi *tunneling* terjadi secara efektif karena proses *tunneling* hanya bekerja saat dibutuhkan. Artinya ketika paket-paket dengan alamat IPv6 memerlukan jalur untuk mencapai alamat tujuan melalui jaringan IPv4 maka disaat inilah proses *tunneling* secara otomatis akan berjalan dan berakhir saat tidak dibutuhkan. Contoh *tunneling* otomatis yang biasa digunakan adalah *tunneling* 6to4, ISATAP, Teredo dll.

### 2.2.2.1 Tunneling 6to4

*Tunneling* 6to4 merupakan salah satu jenis sistem *tunneling* yang memperbolehkan paket dari IPv6 lewat pada jaringan protokol IPv4 dengan melakukan proses enkapsulasi dan dekapsulasi paket. Jenis *tunneling* ini dapat digunakan pada *individual host* ataupun *local IPv6 network*. Ketika digunakan pada *individual host*, *host* tersebut harus memiliki koneksi ke jaringan IPv4 dan alamat IPv4. *Host* tersebut bertanggung jawab dalam enkapsulasi paket IPv6 yang keluar dan dekapsulasi paket dari 6to4 yang masuk. [5] Contoh dari topologi dan konfigurasi *tunneling* 6to4 dapat dilihat pada Gambar 2.5.

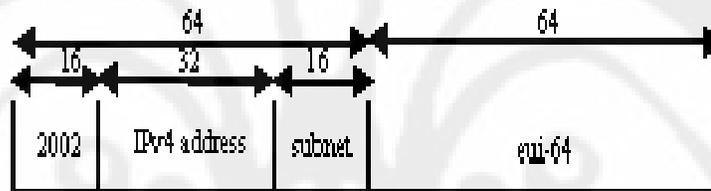


Gambar 2.5 Contoh Konfigurasi Pengalamatan 6to4

*Tunneling* 6to4 melakukan tiga fungsi utama, yaitu :

- ⊗ Menentukan blok dari tempat alamat IPv6 pada *host* atau jaringan yang mempunyai alamat *global* IPv4.
- ⊗ Enkapsulasi paket IPv6 didalam paket IPv4 untuk dikirim melalui jaringan IPv4.
- ⊗ Mengirimkan trafik data diantara 6to4 dan jaringan IPv6.

Alokasi alamat blok IPv6 pada *tunneling* 6to4 diawali dengan alamat 2002 (hex) dan diikuti oleh alamat IPv4 suatu *host* yang sudah diubah menjadi bilangan *hexadecimal*. Contohnya, jika terdapat suatu *host* yang memiliki alamat IPv4 192.1.2.3 maka untuk konfigurasi dari alamat 6to4 *prefix* adalah 2002:C001:0203::/48. Format dari alamat *tunneling* 6to4 dapat dilihat pada Gambar 2.6 dibawah ini :



Gambar 2.6 Format Alamat 6to4 [9]

## 2.3 Media Transmisi

### 2.3.1 Twister Pair Cable

*Twisted pair cable* atau kabel pasangan berpilin terdiri dari dua buah konduktor yang digabungkan dengan tujuan untuk mengurangi atau meniadakan interferensi elektromagnetik dari luar seperti radiasi elektromagnetik dari kabel *Unshielded twisted-pair* (UTP), dan *crosstalk* yang terjadi di antara kabel yang berdekatan.

Ada dua macam *Twisted Pair Cable*, yaitu kabel STP dan UTP. Kabel STP (*Shielded Twisted Pair*) merupakan salah satu jenis kabel yang digunakan dalam jaringan komputer. Kabel ini berisi dua pasang kabel (empat kabel) yang setiap pasang dipilin. Kabel STP lebih tahan terhadap gangguan yang disebabkan posisi kabel yang tertekuk. Pada kabel STP *attenuasi* akan meningkat pada

frekuensi tinggi sehingga menimbulkan *crosstalk* dan sinyal *noise*. Kabel UTP (*Unshielded Twisted Pair*) banyak digunakan dalam instalasi jaringan komputer.

### 2.3.1.1 *Unshield Twisted Pair (UTP) Cable*

Kabel UTP (*Unshielded Twisted Pair*) banyak digunakan dalam instalasi jaringan komputer. Kabel ini berisi empat pasang kabel yang tiap pasangannya dipilin (*twisted*). Kabel ini tidak dilengkapi dengan pelindung (*unshilded*). Kabel UTP mudah dipasang, ukurannya kecil, dan harganya lebih murah dibandingkan jenis media lainnya. Kabel UTP biasa menggunakan konektor *Registered Jack 45* (RJ 45), seperti gambar 2.7 dibawah ini:



Gambar 2.7 Konektor RJ-45 *Cable*

Kabel UTP sangat rentan dengan efek interferensi listrik yang berasal dari media di sekelilingnya. Dan jarak antara sinyal lebih pendek untuk UTP daripada kabel koaksial dan kabel serat optik. UTP dapat dikelompokkan sebagai berikut :

- ⊗ Kategori 1 – Digunakan untuk komunikasi telepon. Tidak cocok untuk transmisi data.
- ⊗ Kategori 2 – Mampu mengirimkan data pada kecepatan hingga 4 *megabyte* per detik (Mbps).
- ⊗ Kategori 3 – Digunakan untuk jaringan *10Base-T*. Dapat mengirimkan data pada kecepatan hingga 10 Mbps.
- ⊗ Kategori 4 – Digunakan untuk jaringan *Token Ring*. Dapat mengirimkan data pada kecepatan hingga 16 Mbps.
- ⊗ Kategori 5 – Dapat mengirimkan data pada kecepatan hingga 100 Mbps.

- ⊗ Kategori 5e – Digunakan dalam jaringan yang berjalan pada kecepatan sampai dengan 1000 Mbps.
- ⊗ Kategori 6 – Terdiri dari empat pasang dari 24 *American Wire Gauge* (AWG) kabel tembaga.

### 2.3.2 Wireless

*Local Area Network* (LAN) merupakan jaringan dari gabungan beberapa komputer yang tersambung melalui kabel UTP. Kemudian seiring dengan perkembangan teknologi serta kebutuhan untuk akses jaringan bergerak, muncullah *Wireless Local Area Network* (WLAN) dimana hubungan antar computer terhubung menggunakan teknologi *Radio Frequency* (RF). [6]

#### 2.3.2.1 Standarisasi Wireless

Berikut ini adalah 4 standarisasi WLAN yang digunakan saat ini:

1. **802.11a**

Merupakan standar untuk perangkat WLAN 802.11a yang memungkinkan untuk pengiriman data mencapai kecepatan 54 Mbps dan beroperasi pada frekuensi 5 GHz. Standar ini bisa menjangkau jarak maksimum 150 kaki (45,7 m).

2. **802.11b**

Standar 802.11b ini beroperasi direntang frekuensi 2,4 GHz dengan *data rate* maksimum 11 Mbps. Perangkat ini dapat menjangkau jarak maksimum 300 kaki (91 m).

3. **802.11g**

IEEE 802.11g secara teoritis memiliki kecepatan maksimum yang sama dengan 802.11a, yaitu 54 Mbps, namun beroperasi pada frekuensi 2,4 GHz, sama seperti 802.11b. Standar 802.11g memiliki jangkauan maksimum 300 kaki (91 m).

4. **802.11n**

802.11n adalah standar nirkabel baru yang memiliki *bandwidth* mencapai 540 Mbps dan beroperasi baik dalam frekuensi 2,4 GHz ataupun 5 GHz. Jarak maksimum jangkauan bisa mencapai 984 kaki (250 m).

Tabel 2.4 Spesifikasi WLAN

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	2,4 GHz	b
802.11a	54 Mb/s	5 GHz	a
802.11g	54 Mb/s	2,4 GHz	b,g
802.11n	100 Mb/s	2,4 GHz	b,g,n

#### 2.4 *File Transfer Protocol (FTP)*

Semakin berkembangnya pengguna internet di seluruh dunia menyebabkan semakin berkembang juga aplikasi – aplikasi baru. Salah satu aplikasi yang sering digunakan adalah FTP yang merupakan kepanjangan dari *File Transfer Protocol*. Aplikasi FTP pada internet diterapkan pada jenis – jenis *file server* seperti *rapidshare*, *megaupload*, *mediafire*, dan sebagainya. Domain – domain tersebut merupakan *File Server* yang bekerja dengan menggunakan *File Transfer Protocol (FTP)* untuk *uploading* dan *downloading* data. FTP digunakan dalam proses pengiriman data baik *uploading* maupun *downloading* melalui jaringan TCP/IP.

Tabel 2.5 *Assigned Port*

Port	Protocol	Use
21	FTP	File Transfer
23	Telnet	Remote Login
25	SMTP	Email
69	TFTP	Trivial file transfer protocol
79	Finger	Lookup Information about a user
80	HTTP	World Wide Web
110	POP-3	Remote email Acces
119	NNTP	USENET news

Protokol FTP merupakan sebuah protokol yang digunakan untuk melakukan pemindahan satu atau lebih file dari suatu *local host* menuju *remote host* atau *host* tujuan. Dapat dilihat pada Tabel 2.5, Protokol FTP ditempatkan pada *port* 21. [7] FTP memiliki kemampuan yang tidak terbatas pada pemindahan

file saja, namun juga sangat memungkinkan pengguna untuk dapat melakukan *remote* (pengendalian) secara jarak jauh. Kemampuan *transfer* data dari satu komputer ke komputer yang dengan sistem operasi yang berbeda merupakan kemampuan lain yang dimiliki oleh FTP. Sebagai contoh, sebuah *local host* yang menggunakan sistem operasi *windows* XP (sistem file NTFS) menghubungkan diri dengan sebuah *remote host* yang menggunakan sistem operasi *linux* Ubuntu (dengan file sistem *ex2fs*).

Dua hal yang penting dalam FTP adalah FTP *Server* dan FTP *Client* :

1. FTP *server* adalah suatu *server* yang menjalankan *software* yang berfungsi untuk memberikan layanan tukar menukar file dimana *server* tersebut selalu siap memberikan layanan FTP apabila mendapat permintaan (*request*) dari FTP *client*.
2. FTP *client* adalah komputer yang *request* koneksi ke FTP *server* untuk tujuan tukar menukar file. Setelah terhubung dengan FTP *server*, maka *client* dapat *men-download*, *meng-upload*, *me-rename*, *men-delete*, dll sesuai dengan *permission* yang diberikan oleh FTP *server*. [8]

Tujuan dari FTP *server* adalah sebagai berikut :

- ☒ Untuk tujuan *sharing* data
- ☒ Untuk menyediakan *indirect* atau *implicit remote computer*
- ☒ Untuk menyediakan tempat penyimpanan bagi user
- ☒ Untuk menyediakan *transfer* data yang *reliable* dan efisien

#### 2.4.1. *Anonymous*

Sistem FTP *anonymous* dibuat dengan tujuan agar setiap orang yang terkoneksi ke dalam dunia internet dapat saling berbagi file dengan orang lain yang belum memiliki *account* dalam *server*. Dengan sistem ini setiap orang dapat menggunakan sebuah *account* yang umum (*public account*) berupa *anonymous*.

Melihat kondisi diatas yang menggunakan *public account*, hak yang dimiliki seorang pengguna sangat terbatas kepada aturan-aturan yang dimiliki oleh pemilik *server* (*remote host*).

Keterbatasan yang dimiliki ketika pengguna menggunakan jenis FTP *anonymous* biasanya meliputi keterbatasan dalam proses akses *directory* dan file yang tersedia dalam *server* yang dituju. Selain itu, pengguna yang menggunakan sistem ini tidak dapat melakukan *uploading data* terhadap *server* yang dituju. Namun sebaliknya ia hanya memiliki kemampuan dalam *downloading*, baca file tertentu dan pindah *directory* yang diizinkan oleh pemilik *server*.

#### **2.4.2. User Legal (Authenticated User)**

Sistem FTP *User Legal* adalah sebuah cara lain yang digunakan oleh pengguna internet dalam mengakses sebuah *server* dengan menggunakan FTP. Untuk dapat mengakses *remote host*, cara *user legal (authenticated user)* menuntut kita untuk memiliki sebuah *account* khusus yang dimiliki secara pribadi.

Untuk dapat memiliki *account* khusus ini, seorang pengguna harus mendaftarkan diri terlebih dahulu kepada pemilik *remote host* tersebut. Terdapat banyak *server* yang memberikan fasilitas *account* FTP secara gratis, selain *server* lain yang mengharuskan pengguna untuk membeli sebuah *account* yang tentunya dengan fasilitas yang lebih banyak dibandingkan dengan sebuah *account* yang dimiliki secara *free*.

Dengan menggunakan *account* ini, seorang pengguna memiliki hak akses yang jauh berbeda dengan seorang pengguna jenis *anonymous*. Selain kemampuan yang dimiliki oleh pengguna *anonymous*, seperti *download* dan berpindah dari satu *directory* ke *directory* lain serta kemampuan baca file tertentu, *uploading*, membuat sebuah *directory*, menghapus file dan *directory*. Hak yang dimiliki adalah hak seorang pemilik bukan seorang pengunjung biasa.

## BAB 3

### PERANCANGAN TOPOLOGI JARINGAN

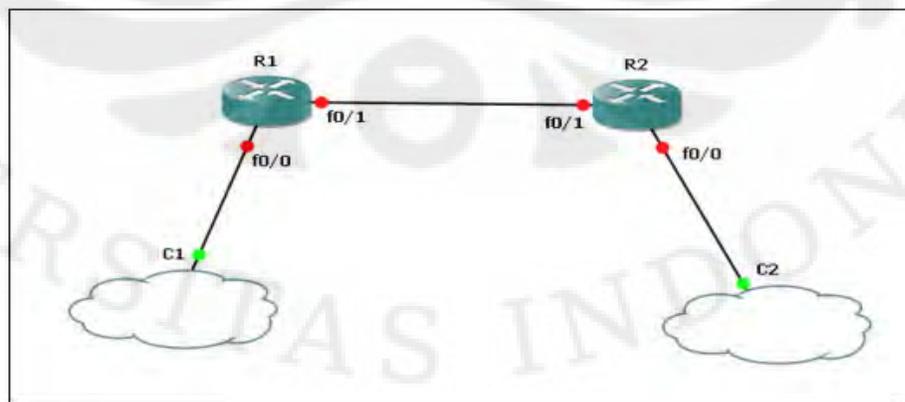
#### 3.1 Perancangan Topologi Jaringan

Pada perancangan topologi jaringan *test-bed* terdiri dari 3 tipe jaringan yang diklasifikasikan menurut teknik konfigurasi pengalamatannya, yaitu jaringan IPv4, jaringan IPv6, dan jaringan IPv6 *tunneling* 6to4.

Lalu dari 3 tipe jaringan ini akan dibagi lagi masing-masing menjadi 2 tipe jaringan menurut media yang digunakannya, yaitu jaringan dengan media *wireless* dan jaringan dengan media *wired*. Dari perbedaan media ini akan dilakukan pengamatan mengenai kualitas *File Transfer Protocol* (FTP) dari masing-masing tipe jaringan dan memperoleh informasi mengenai kelebihan dan kekurangan dari jaringan tersebut.

Sebagai perangkat pendukung utama dalam *test-bed* akan digunakan PC yang akan bertindak sebagai *router*. Alasan penggunaan PC-*router* adalah sebagai solusi dari permasalahan keterbatasan *router* yang tidak dimiliki karena keterbatasan biaya dan lain sebagainya. Dalam implementasinya, PC yang difungsikan sebagai *router* menggunakan aplikasi GNS3 untuk melakukan emulasi terhadap PC sehingga IOS dapat berjalan dalam sistem jaringan tersebut.

Tentu saja ini berbeda dengan simulasi karena aplikasi GNS3 ini merupakan aplikasi *emulator* yang akan mempraktekkan dan menjalankan PC secara *real* agar berperan seperti *router* yang sesungguhnya. Dengan IOS yang disesuaikan, maka PC-*router* tersebut dapat digunakan sebagai *router* seri 1800, 3700, 7200 dan lain sebagainya.



Gambar 3.1 Tampilan Topologi Umum Jaringan *Test-bed* Pada GNS3

Gambar 3.1 di atas adalah contoh *display* topologi umum dari jaringan *test-bed* pada GNS3. Pada gambar tersebut diperlihatkan tampilan aplikasi GNS3 pada PC yang menggunakan 2 *router* dan 2 *cloud*. Fungsi *cloud* disini hanya sebagai tampilan GNS3 yang menandakan C1 adalah hubungan langsung antara *router* R1 dengan *server* dan C2 adalah hubungan langsung antara *router* R2 dengan *client*. *Cloud* disini bukan menunjukkan jaringan secara luas akan tetapi hanya tampilan yang ada pada aplikasi GNS3 yang mengemulasikan PC sebagai *router*.

Berikut ini, spesifikasi perangkat keras yang diperlukan untuk jaringan *test-bed* :

1. *Server (laptop Dell)*
  - ☒ Sistem operasi *Windows XP SP3*
  - ☒ Intel Centrino CPU 2.00 GHz
  - ☒ 1 GByte RAM
  - ☒ Ethernet 10/100 Mbps
2. *Client (laptop HP Compaq 510)*
  - ☒ Sistem operasi *Windows XP SP2*
  - ☒ Intel Core 2 Duo CPU 2.00 GHz
  - ☒ 2 GByte RAM
  - ☒ Ethernet 10/100 Mbps
3. *Router (PC)*
  - ☒ Sistem operasi *Windows XP SP2*
  - ☒ Pentium 4 CPU 2.00 GHz
  - ☒ 1 GByte RAM
  - ☒ IOS C3725 versi 12.4.(15)T5
  - ☒ Ethernet 10/100 Mbps
4. LAN Card
  - ☒ Supports 10/100 Mbps
  - ☒ *Integrated Fast Ethernet MAC*
5. *Wireless router (BLR33N)*
  - ☒ Supports 802.11b/g/n *standars*
  - ☒ Kecepatan hingga 300 Mbps
  - ☒ Maximum 100 m indoor dan 300 m outdoor

## 6. 2 Buah kabel Ethernet

Selain perangkat-perangkat keras penunjang diperlukan juga aplikasi perangkat lunak yang diperlukan untuk mendukung proses penelitian antara lain :

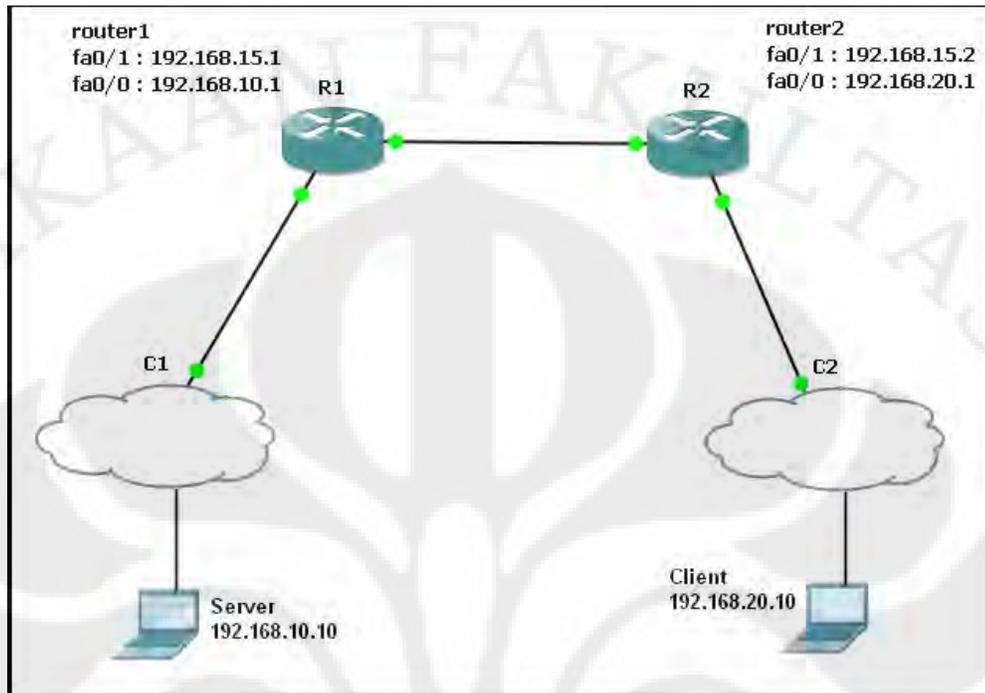
1. Xlight FTP Server merupakan aplikasi penunjang yang digunakan untuk proses *transfer file* pada *File Transfer Protocol (FTP)*. Pada aplikasi ini Xlight FTP Server akan bertindak sebagai *server*.
2. GNS3, merupakan aplikasi *emulator* yang membuat suatu PC dapat bertindak menjadi sebuah *router*. Aplikasi ini khusus ditempatkan pada *laptop* yang akan digunakan sebagai *router*.
3. Wireshark, merupakan aplikasi yang digunakan untuk meng-*capture* dan menganalisa trafik yang terjadi pada suatu *interface*. Aplikasi ditempatkan pada sisi *client* untuk mengetahui keberhasilan dari kinerja jaringan yang dibuat.

Skenario proses *file transfer protocol (ftp)* yang diterapkan pada tiap topologi jaringan terbagi menjadi 2 media, yaitu media *wireless* dan media *wired*. Data yang akan diperoleh merupakan analisa *test-bed file transfer protocol (ftp)* antara jaringan *wireless* dan jaringan *wired*. Sebagai penunjang proses *file transfer protocol (ftp)* maka digunakan aplikasi Xlight FTP Server yang diterapkan pada sisi *server*. Sedangkan pada sisi *client* menggunakan perintah *command prompt* untuk proses pengambilan data dari *server* / proses *downloading*. Kemudian digunakan juga aplikasi wireshark untuk melihat trafik dari paket – paket yang ada pada *ethernet* tersebut.

### 3.2 Konfigurasi Jaringan Wired

Pada bagian ini akan dijelaskan mengenai perancangan topologi jaringan yang menggunakan *wired* untuk *test-bed*. Topologi jaringan terdiri dari jaringan IPv4 *wired*, jaringan IPv6 *wired*, dan jaringan IPv6 *tunneling 6to4 wired*.

### 3.2.1 Konfigurasi IPv4 Wired

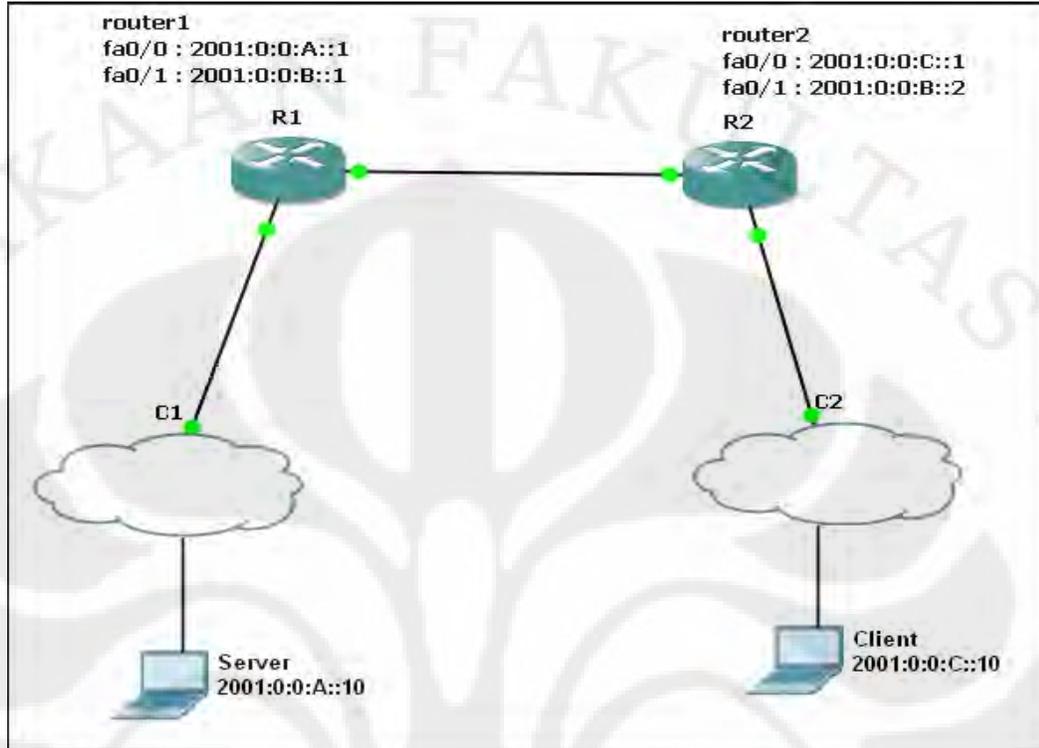


Gambar 3.2 Topologi Jaringan *Test-bed* IPv4 Wired

Pada Gambar 3.2 diatas merupakan gambaran topologi jaringan *test-bed* IPv4. Pada *node* C1 merupakan *ethernet1* pada PC-router yang nantinya akan dihubungkan ke *ethernet laptop-server*. Sementara pada *node* C2 merupakan *ethernet2* pada PC-router yang nantinya akan dihubungkan ke *ethernet laptop-client*. Pengertian *cloud* C1 dan C2 di sini adalah tampilan dari aplikasi GNS3 yang mendefinisikan bahwa *router* R1 secara langsung terhubung dengan *server* dan *router* R2 secara langsung terhubung dengan *client*.

Topologi jaringan IPv4 *wired* ini menggunakan *protocol routing* RIP versi 2, sehingga dapat memberikan fitur *classless routing*. Pada jaringan IPv4 dengan media *wired* ini *ethernet client* langsung terhubung dengan *ethernet PC-router* melalui kabel. Dengan tipe jaringan seperti ini langsung dapat terlihat kendala yang dihadapi adalah keterbatasan *mobilitas client* karena ruang geraknya terbatas oleh panjang kabel *ethernet* yang digunakan. Konfigurasi lengkap untuk jaringan IPv4 murni ada pada lampiran 1.

### 3.2.2 Konfigurasi IPv6 Wired

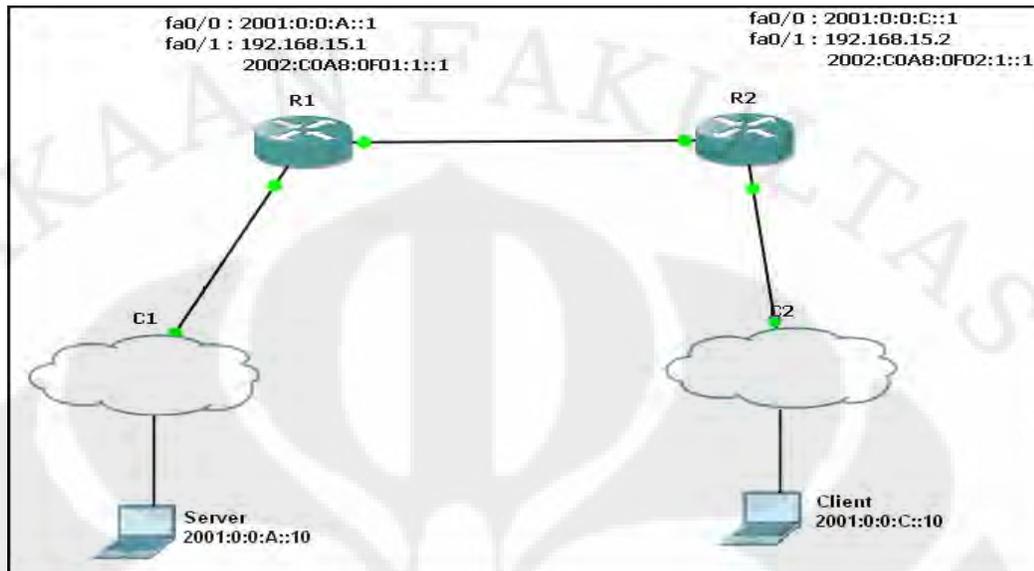


Gambar 3.3 Topologi Jaringan *Test-bed* IPv6 Wired

Pada Gambar 3.3 diatas merupakan gambaran topologi jaringan *test-bed* IPv6. Pada *node* C1 merupakan *ethernet1* pada PC-router yang nantinya akan dihubungkan ke *ethernet laptop-server*. Sementara pada *node* C2 merupakan *ethernet2* pada PC-router yang nantinya akan dihubungkan ke *ethernet laptop-client*. Pengertian *cloud* C1 dan C2 di sini adalah tampilan dari aplikasi GNS3 yang mendefinisikan bahwa *router* R1 secara langsung terhubung dengan *server* dan *router* R2 secara langsung terhubung dengan *client*.

Pada jaringan ini tipe pengalamatan yang digunakan adalah alamat IPv6. Topologi jaringan IPv6 *wired* ini menggunakan *protocol routing* RIPng, sehingga mampu melakukan *routing* dengan tipe pengalamatan IPv6. Konfigurasi lengkap untuk jaringan IPv6 murni ada pada lampiran 2.

### 3.2.3 Konfigurasi IPv6 *tunneling 6to4 wired*



Gambar 3.4 Topologi Jaringan *Test-bed IPv6 Tunneling 6to4 Wired*

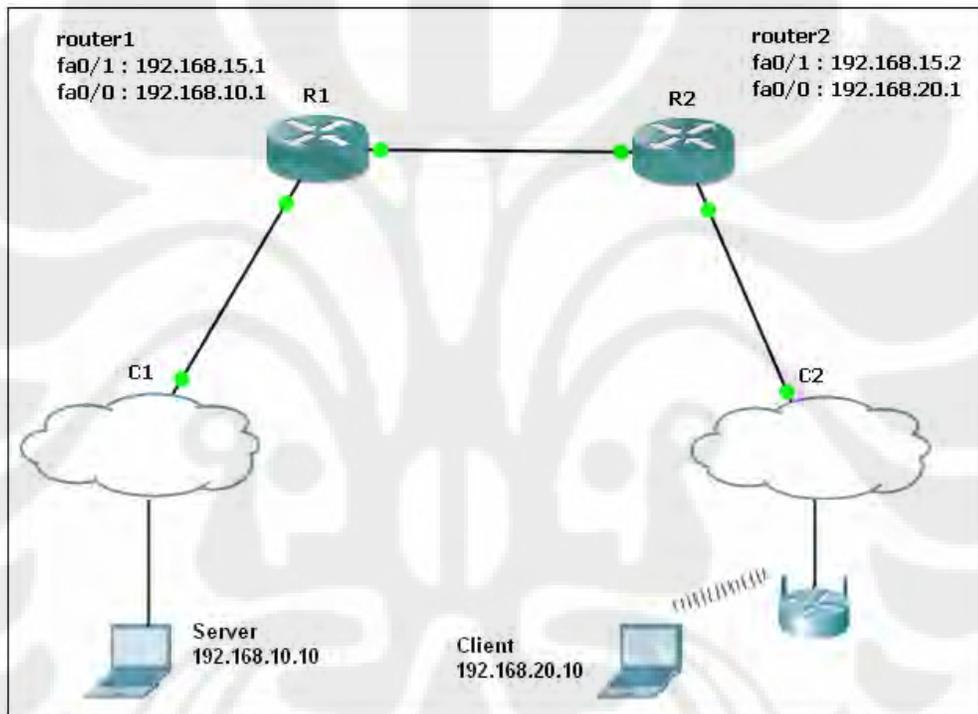
Pada Gambar 3.4 diatas merupakan gambaran topologi jaringan *test-bed IPv6 tunneling 6to4 wired*. Pada *node C1* merupakan *ethernet1* pada *PC-router* yang nantinya akan dihubungkan ke *ethernet laptop-server*. Sementara pada *node C2* merupakan *ethernet2* pada *PC-router* yang nantinya akan dihubungkan ke *ethernet laptop-client*. Fungsi *cloud* di sini sama seperti jaringan sebelumnya, yaitu hanya tampilan dari aplikasi GNS3 saja.

Pada jaringan ini terdapat perbedaan konfigurasi *router* maupun pengaturan alamat IP pada sisi *client* dan *server*. *Client* dan *server* akan diberikan alamat IPv6, sedangkan *Router R1* dan *R2* akan dikonfigurasi dengan teknik *tunneling 6to4*. Pada *R1 port fa 0/1* diberi alamat IPv4 yaitu 192.168.15.1, kemudian akan dikonversikan menjadi *hexadecimal* sesuai format alamat 6to4 menjadi 2002:C0A8:0F01:1::1. Demikian juga dengan *R2* yaitu dengan format alamat 6to4 2002:C0A8:0F02:1::1. Pada bagian *routing protocol* di tiap-tiap *router* akan diimplementasikan 2 tipe *routing protocol* yaitu *RIP* versi 2 dan *RIPng*. Alasan digunakannya 2 tipe *routing* tersebut adalah karena pada jaringan ini harus mampu melakukan *routing* terhadap tipe pengalaman IPv4 dan tipe pengalaman IPv6.

### 3.3 Konfigurasi Jaringan *Wireless*

Pada bagian ini akan dijelaskan mengenai perancangan topologi jaringan *wireless* untuk *test-bed*. Perancangan topologi dengan media *wireless* di sini pada dasarnya hampir sama dengan media *wired*, dimana menggunakan *access point* hanya pada sisi *client*. Topologi jaringan terdiri dari jaringan IPv4 *wireless*, jaringan IPv6 *wireless*, dan jaringan IPv6 *tunneling 6to4 wireless*.

#### 3.3.1 Konfigurasi IPv4 *Wireless*



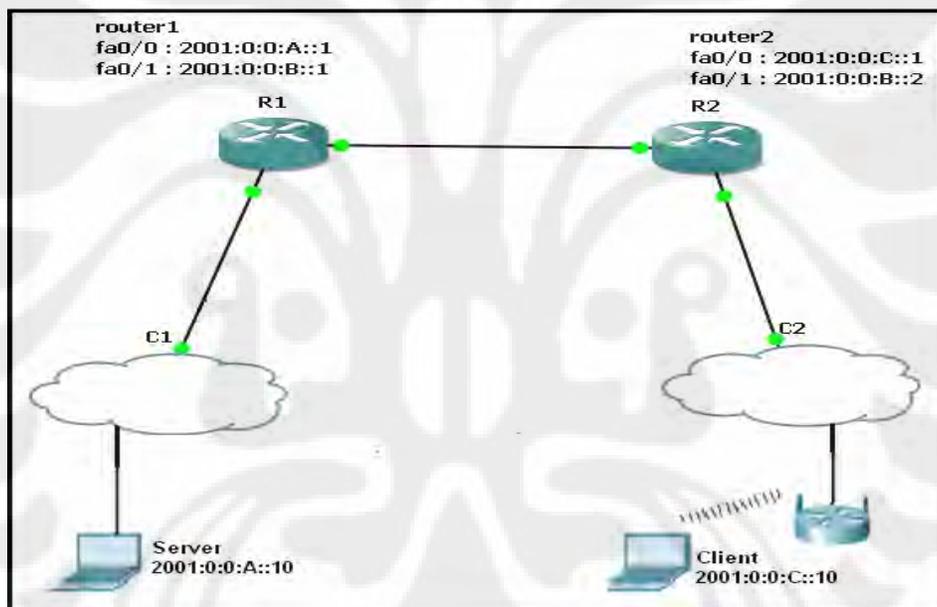
Gambar 3.5 Topologi Jaringan *Test-bed* IPv4 *Wireless*

Pada Gambar 3.5 diatas merupakan gambaran topologi jaringan *test-bed* IPv4. Dua *router* di atas adalah *PC-Router* yang menggunakan aplikasi GNS3. Pada *PC-Router* tersebut terdapat 2 *port Ethernet* dengan menggunakan tambahan LAN Card. Pengertian *cloud* C1 dan C2 di sini adalah tampilan dari aplikasi GNS3 yang mendefinisikan bahwa *router* R1 secara langsung terhubung dengan *server* dan *router* R2 secara langsung terhubung dengan *client*.

Dapat dilihat *node C1* terdapat *fast-ethernet 0/0* yang merupakan *port* dari *router R1* yang terhubung ke *port ethernet laptop-server*. Sementara pada *node C2* terdapat *fast-ethernet 0/0* yang merupakan *port router R2* yang terhubung ke *port Access Point*. Dari *Access Point* ini, kemudian akan memancarkan kearah *client*.

Topologi jaringan IPv4 *wireless* ini menggunakan *protocol routing* RIP versi 2, sehingga dapat memberikan fitur *classless routing*. Untuk konfigurasi jaringan IPv4 murni menggunakan *wireless* sama seperti konfigurasi jaringan IPv4 murni menggunakan *wired*. Konfigurasi lengkapnya ada pada lampiran 1.

### 3.3.2 Konfigurasi IPv6 Wireless

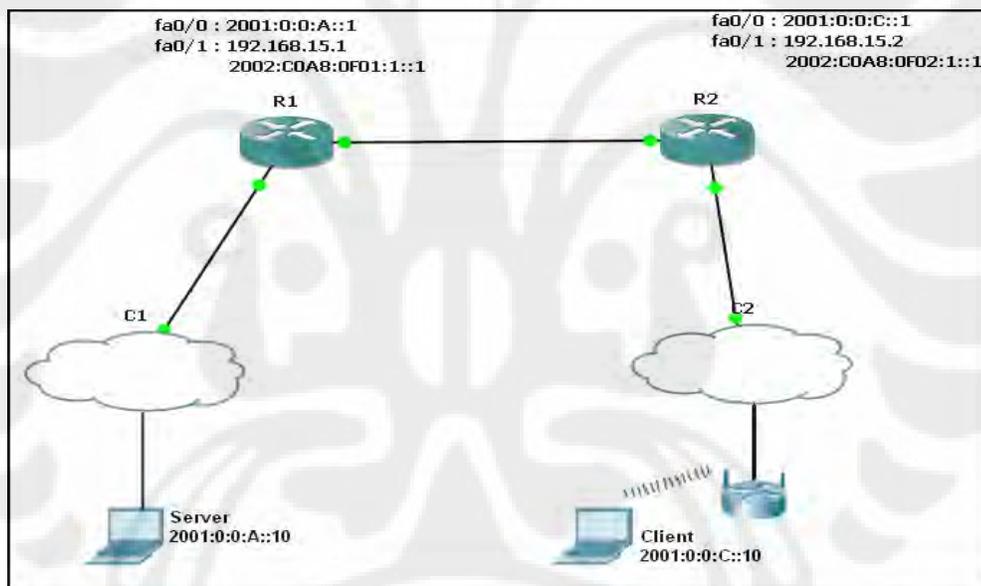


Gambar 3.6 Topologi Jaringan *Test-bed* IPv6 *Wireless*

Pada Gambar 3.6 diatas merupakan gambaran topologi jaringan *test-bed* IPv6. Pada *node C1* merupakan *port ethernet1* pada *PC-router* yang nantinya akan dihubungkan ke *port ethernet laptop-server*. Sementara pada *node C2* merupakan *port ethernet2* pada *PC-router* yang nantinya akan dihubungkan ke *port ethernet AP (Access Point)*. Sama seperti jaringan sebelumnya, tampilan *cloud* di sini hanya sebagai tampilan dari aplikasi GNS3 saja, bukan tampilan jaringan secara luas sebagaimana arti dari tampilan *cloud* itu sendiri.

Pada jaringan ini tipe pengalamatan yang digunakan adalah alamat IPv6. Topologi jaringan IPv6 *wireless* ini menggunakan *protocol routing* RIPng, sehingga mampu melakukan *routing* dengan tipe pengalamatan IPv6. Untuk konfigurasi jaringan IPv6 murni menggunakan *wireless* sama seperti konfigurasi jaringan IPv6 murni menggunakan *wired*. Alasan penggunaan tipe routing yang sama adalah agar analisa yang dilakukan seimbang, karena proses yang diamati adalah menitikberatkan pada perbandingan kualitas antara media *wired* dan *wireless*. Untuk konfigurasi jaringan IPv6 murni menggunakan *wireless* sama seperti konfigurasi jaringan IPv6 murni menggunakan *wired*. Konfigurasi lengkapnya ada pada lampiran 2.

### 3.3.3 Konfigurasi IPv6 Tunneling 6to4 Wireless



Gambar 3.7 Topologi Jaringan *Test-bed* IPv6 Tunneling 6to4 Wireless

Pada Gambar 3.7 diatas merupakan gambaran topologi jaringan *test-bed* IPv6 tunneling 6to4. Pada node C1 merupakan port *ethernet1* pada PC-router yang nantinya akan dihubungkan ke port *ethernet laptop-server*. Sementara pada node C2 merupakan port *ethernet2* pada PC-router yang nantinya akan dihubungkan ke port *ethernet AP (Access Point)*.

Pada jaringan ini terdapat perbedaan konfigurasi *router* maupun pengaturan alamat IP pada sisi *client* dan *server* dibanding dengan jaringan IPv4

dan IPv6 murni. *Client* dan *server* akan diberikan alamat IPv6, sedangkan *router* R1 dan R2 akan dikonfigurasi dengan teknik *tunneling* 6to4. Konsep *tunneling* adalah melewati IPv6 melalui jaringan IPv4. Akan tetapi alamat dari IPv4 diubah terlebih dahulu sesuai format alamat dari *tunneling* 6to4. Proses *enkapsulasi* ini berlangsung pada waktu melewati IPv6 melalui jaringan IPv4. Dan setelah tiba di sisi seberang akan *didekapsulasi* agar tersisa alamat IPv6 saja. Pada R1 *port* fa 0/1 diberi alamat IPv4 yaitu 192.168.15.1, kemudian akan dikonversikan menjadi *hexadecimal* sesuai format alamat 6to4 menjadi 2002:C0A8:0F01:1::1. Demikian juga dengan R2 yaitu dengan format alamat 6to4 2002:C0A8:0F02:1::1. Untuk konfigurasi jaringan *tunneling* 6to4 menggunakan *wireless* sama seperti konfigurasi jaringan *tunneling* 6to4 menggunakan *wired*. Konfigurasi lengkapnya ada pada lampiran 3.

### 3.4 Metode Pengambilan Data

Pengambilan data dilakukan dengan cara melakukan *download* file dari *server* ke *client* pada konfigurasi jaringan yang berbeda – beda, yaitu IPv4 murni, IPv6 murni dan *tunneling* 6to4. Untuk proses pengambilan data ini akan dilakukan penangkapan paket – paket yang lewat pada sisi *client* dengan menggunakan aplikasi *wireshark*.

Terdapat tiga parameter yang akan diambil dalam pengambilan data, yaitu *transfer time*, *throughput*, dan *delay*. Parameter tersebut dianggap dapat mewakili unjuk kerja dari FTP dalam melakukan proses *download* data. Jika dilihat dari ketiga parameter tersebut memiliki keterkaitan satu sama lain.

File yang di-*download* dari FTP server dibedakan dalam bermacam – macam ukuran, yaitu 5 Mbytes, 10 Mbytes, 15 Mbytes dan 20 Mbytes. Perbedaan ukuran file ini bertujuan untuk mengamati hubungan antara ukuran file dengan parameter – parameter uji.

Pengambilan data pada masing - masing parameter uji coba dilakukan sebanyak 10 kali untuk setiap kapasitas file pada masing – masing konfigurasi jaringan. Pada setiap konfigurasi jaringan dilakukan pengambilan data sebanyak 120 kali. Pengambilan data dilakukan dengan kondisi jaringan *test-bed* semirip mungkin dengan jaringan pada umumnya. Hal ini penting untuk dilakukan agar

dapat diperoleh kesimpulan yang tepat. Untuk itu pada waktu pengambilan data dilakukan *ping* dengan beban 64 Kbytes dari *client* ke *server*.

Pada skripsi ini dilakukan pengambilan data melalui 2 media, yaitu menggunakan media *wired* dan *wireless*. Pada media tersebut dilakukan percobaan pada masing-masing topologi, yaitu pada jaringan IPv4 murni, IPv6 murni dan *tunneling* 6to4. Pada jaringan yang menggunakan *wireless* digunakan *access point* sebagai penghubung pada sisi *client*. Pada proses pengambilan data menggunakan media *wireless*, jarak antara *client* dan *access point*  $\pm 10$  m. Sehingga total pengambilan data yang dilakukan adalah 720 data.

Penangkapan-penangkapan data melalui aplikasi Wireshark pada sisi *client*. Pada proses penangkapan (*capture*) data tersebut akan menampilkan banyak paket yang diterima, maka dari itu dilakukan proses *filtering* terlebih dahulu. Dengan melakukan beberapa proses *filtering*, sehingga yang akan terlihat hanya paket-paket *file transfer protocol* (FTP) saja. Kemudian dapat dilihat pada *summary*, dan mendapat parameter-parameter yang diinginkan, yaitu *transfer time*, *throughput* dan *delay*.

## BAB 4

### ANALISA DATA

#### 4.1. Analisa Konfigurasi Jaringan

Pengambilan data dilakukan dengan menggunakan jaringan sederhana pada setiap konfigurasi jaringan yang diujikan. Secara keseluruhan untuk topologi yang digunakan adalah sama, yaitu menggunakan dua buah *laptop* dan sebuah PC yang digunakan sebagai PC-router. *Laptop* 1 difungsikan sebagai *client*, sedangkan untuk *laptop* 2 difungsikan sebagai *server*. Masing-masing *laptop* tersebut terhubung melalui PC yang difungsikan sebagai *router* dengan aplikasi emulator GNS3.

Pada penelitian ini dilakukan 2 percobaan melalui media *wired* dan *wireless*. Untuk media *wired*, yaitu menghubungkan antara perangkat keras baik itu *laptop* dengan PC dan sebaliknya digunakan kabel UTP yang mempunyai tipe *cross-over*. Sedangkan untuk media *wireless*, pada sisi *client* ditempatkan sebuah *access point* sebagai media *wireless* yang akan menghubungkan *laptop-client*.

Konfigurasi yang digunakan dalam penelitian ini ada 3 macam, yaitu konfigurasi jaringan IPv4 murni, IPv6 murni, IPv6 dengan menggunakan *tunneling* 6to4.

##### 4.1.1. Konfigurasi IPv4 Murni

Pada konfigurasi jaringan IPv4 murni seluruh perangkat baik itu PC-Router maupun *laptop* diberikan alamat IPv4. Dalam konfigurasi ini jenis *routing* yang digunakan adalah *static routing*, dikarenakan semua konfigurasi menggunakan alamat IPv4 maka proses transmisi data yang terjadi pada saat melalui *router* hanya *routing* dan *forwarding* paket seperti jaringan pada umumnya. Untuk konfigurasi lengkapnya dapat dilihat pada lampiran 1.

Hasil *traceroute* pada Gambar 4.1 dilakukan dari *server* ke *client*. Pada hasil *traceroute* tersebut terlihat bahwa seluruh *hop* yang dilewati merupakan alamat IPv4.

```

C:\Documents and Settings\Administrator>tracert 192.168.10.10
Tracing route to 192.168.10.10 over a maximum of 30 hops
  1  33 ms  13 ms  5 ms  192.168.20.1
  2  26 ms  43 ms  24 ms  192.168.15.1
  3  42 ms  22 ms  18 ms  192.168.10.10
Trace complete.

```

Gambar 4.1 Tampilan *Traceroute* Pada Jaringan IPv4 Murni

#### 4.1.2. Konfigurasi IPv6 Murni

Pada konfigurasi jaringan IPv6 murni juga memiliki topologi yang sama dengan topologi IPv4 murni sehingga tidak ada perbedaan jumlah *hop*. Perbedaannya terletak pada konfigurasi alamat IP yang diberikan adalah menggunakan alamat IPv6 murni. Dalam konfigurasi ini fungsi *router* juga tidak berbeda dengan IPv4 murni yaitu hanya *routing* dan *forwarding* paket seperti jaringan pada umumnya. Untuk konfigurasi router jaringan IPv6 dapat dilihat pada lampiran 2.

Jenis *routing* yang digunakan pada konfigurasi ini yaitu *static routing*. Hasil *traceroute* pada Gambar 4.2 merupakan hasil *traceroute* dari *server* ke *client*. Dari hasil *traceroute* tersebut dapat diketahui *hop* yang dilewati oleh data. Hasil *traceroute* menunjukkan bahwa seluruh hop yang dilewati merupakan alamat IPv6.

```

C:\Documents and Settings\Priyo Sadono>tracert 2001:0:0:c::10
Tracing route to 2001:0:0:c::10 over a maximum of 30 hops
  1   7 ms   3 ms   4 ms  2001:0:0:a::1
  2  13 ms   9 ms  12 ms  2001:0:0:b::2
  3  16 ms  14 ms  11 ms  2001:0:0:c::10
Trace complete.

```

Gambar 4.2 Tampilan *Traceroute* Pada Jaringan IPv6 Murni

#### 4.1.3. Konfigurasi *Tunneling 6to4*

Pada konfigurasi jaringan *tunneling 6to4* juga memiliki topologi sama dengan jaringan IPv4 murni dan IPv6 murni sehingga tidak ada perbedaan jumlah *hop*. Pada konfigurasi ini menggunakan mekanisme *tunneling 6to4* sehingga pada sisi *router* perlu ditambahkan *interface tunnel* yang di-*setting* menjadi *mode 6to4*.

Peran *router* pada konfigurasi ini selain *routing*, juga berfungsi untuk proses *enkapsulasi* dan *dekapsulasi* paket. Konsep *tunneling* adalah melewati IPv6 melalui jaringan IPv4. Akan tetapi alamat dari IPv4 diubah terlebih dahulu sesuai format alamat dari *tunneling* 6to4. Proses *enkapsulasi* ini berlangsung pada waktu melewati IPv6 melalui jaringan IPv4. Dan setelah tiba di sisi seberang akan *didekapsulasi* agar tersisa alamat IPv6 saja sesuai dengan awal. Untuk konfigurasi lengkap jaringan *tunneling* 6to4 dapat dilihat pada lampiran 3.

Hasil *tracert* pada Gambar 4.3 dilakukan dari sisi *server* ke *client*. Dapat dilihat pada hasil *tracert* bahwa pada *hop* kedua terdapat alamat IPv6 2002:COA8:F01:1::1. Alamat IPv6 pada *hop* kedua tersebut menunjukkan alamat *identifikasi* dari *tunneling* 6to4 sehingga bisa dipastikan bahwa data yang dikirim dari *server* ke *client* melalui mekanisme *tunneling* 6to4 terlebih dahulu.

```

C:\WINDOWS\system32\cmd.exe
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 24ms, Maximum = 51ms, Average = 39ms

C:\Documents and Settings\Administrator>tracert 2001:0:0:a::10

Tracing route to 2001:0:0:a::10 over a maximum of 30 hops
  0  5 ms    8 ms    11 ms  2001:0:0:c::1
  1  38 ms   17 ms   16 ms  2002:c0a8:f01:1::1
  2  52 ms   23 ms   48 ms  2001:0:0:a::10

Trace complete.

```

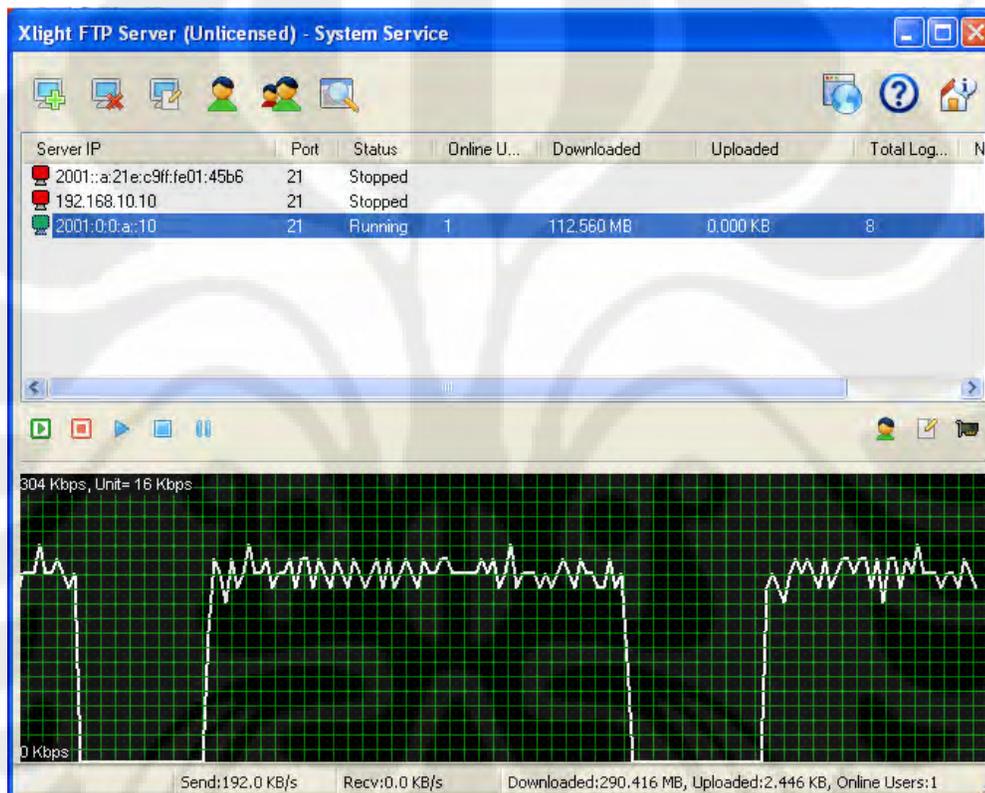
Gambar 4.3 Tampilan *Tracert* Pada Jaringan *Tunneling* 6to4

#### 4.2. Analisa Performa Jaringan Pada FTP

*File Transfer Protocol* (FTP) merupakan suatu aplikasi yang digunakan untuk memindahkan suatu file data dari suatu *server* ke *client* atau sebaliknya. FTP merupakan suatu jenis protokol yang bekerja dengan memanfaatkan protokol TCP/IP yang pada umumnya menggunakan *port* 21.

FTP dibedakan jadi 2, yaitu FTP *Server* dan FTP *Client*. FTP *server* merupakan tempat untuk menyimpan file- file yang akan didownload oleh *client*, sedangkan FTP *client* berfungsi untuk melakukan *download* file dari *server*. *Laptop* yang berfungsi sebagai *client* akan melakukan permintaan koneksi FTP yang ditujukan pada FTP *server*. Proses *requesting* ini diawali dengan *three way handshaking* yang merupakan ciri khas dari protokol TCP/IP.

Pada skripsi ini digunakan aplikasi Xlight FTP Server sebagai *server* FTP. Pada masing-masing topologi yang dibuat, yaitu jaringan IPv4 murni, jaringan IPv6 murni dan jaringan *tunneling* 6to4 akan diimplementasikan penggunaan aplikasi tersebut. Secara otomatis Xlight dapat mendeteksi *address* yang ada pada masing-masing jaringan tersebut. Gambar 4.4 di bawah ini adalah gambaran aplikasi Xlight FTP Server :



Gambar 4.4 Aplikasi Xlight FTP Server

Setelah *server* dan *client* terkoneksi, maka selanjutnya akan diminta *username* dan *password* oleh FTP *server* sebagai *user* untuk memakai aplikasi FTP tersebut. Dalam pengujian ini digunakan modus *anonymous* sehingga tidak perlu mendaftarkan *account* terlebih dahulu pada *server*. Proses *autentification* pada *username* dan *password* yang dimasukkan *client* akan dilakukan oleh *server*. Sehingga cukup dengan nama *account anonymous*, *user* sudah dapat masuk ke dalam aplikasi FTP tersebut tanpa *password*. Setelah dinyatakan tersambung oleh *server* maka *client* sudah mendapatkan layanan FTP dengan modus *anonymous*.

Gambaran proses koneksi FTP *client* dan *server* dapat dilihat pada Gambar 4.5 berikut ini :

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>ftp 2001:0:0:a::10
Connected to 2001:0:0:a::10.
220 Xlight FTP Server 3.5 ready...
User (2001:0:0:a::10:(none)): anonymous
331 Anonymous login OK, send your e-mail as password
Password:
230 Login OK
ftp> bin
200 Type set to I.
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for NLIST (73 bytes).
10M.rar
15M.rar
160M.rar
20M.rar
320M.rar
40M.rar
5M.rar
80M.rar
226 Transfer complete (0.453 KB/s).
ftp: 73 bytes received in 0.06Seconds 1.16Kbytes/sec.
ftp> get 5m.rar
200 PORT command successful
150 Opening BINARY mode data connection for 5m.rar (5696602 bytes).
226 Transfer complete (325.390 KB/s).
ftp: 5696602 bytes received in 17.39Seconds 327.58Kbytes/sec.
ftp> bye
221 Good-Bye

```

Gambar 4.5 Tampilan Koneksi FTP *Server* Dengan *Client*

Proses *download* file dari *server* dilakukan oleh *client* dengan ukuran file yang berbeda - beda. Ukuran file dibedakan menjadi empat macam, yaitu 5 MB, 10 MB, 15 MB, dan 20 MB. Semua tipe file dibuat sama dalam bentuk ekstensi .rar untuk memudahkan *client* melakukan *download* dari *server* dan menghindari pengaruh perbedaan tipe file pada performa FTP. Masing-masing file tersebut akan *download* pada tiap jaringan yang ada, baik menggunakan media *wired* ataupun *wireless*. File – file yang diujikan dapat dilihat pada Tabel 4.1 di bawah ini :

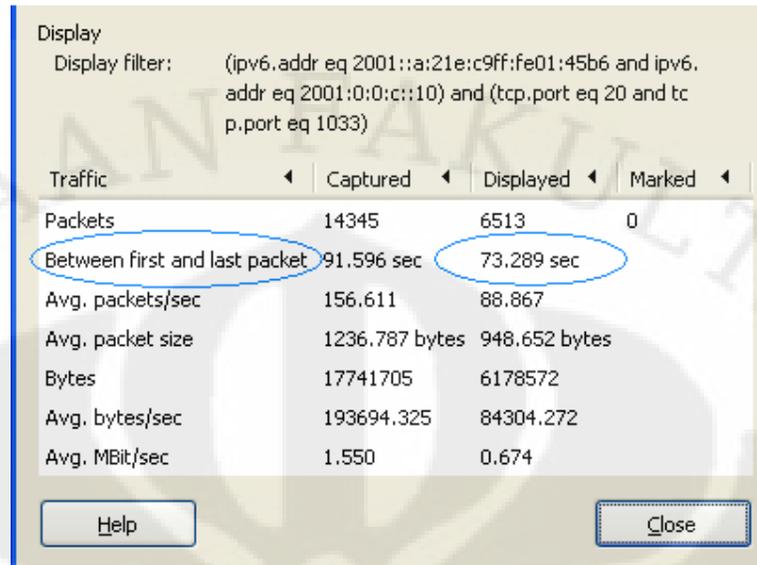
Tabel 4.1 Nama File dan Kapasitasnya

Nama File	Kapasitas (bytes)
FILE5M	5696602
FILE10M	10690520
FILE15M	15765158
FILE20M	20998049

Terdapat tiga parameter yang diambil dalam pengambilan data, yaitu *transfer time*, *throughput*, dan *delay*. Parameter tersebut dianggap dapat mewakili unjuk kerja dari FTP dalam melakukan proses *download* data. Jika dilihat dari ketiga parameter tersebut memiliki keterkaitan satu sama lain. Untuk analisa data tentang ketiga parameter tersebut dijelaskan pada bagian berikut ini :

#### 4.2.1. Analisa Transfer Time

*Transfer Time* adalah jumlah waktu yang dibutuhkan untuk mengirimkan seluruh paket dari *server* ke *client* yang dinyatakan dalam *second*. Pengambilan parameter *transfer time* dilakukan dengan cara *download* file dari *server* ke *client*. Kemudian disaat yang bersamaan pada sisi *client* melakukan *capture* data atau penangkapan paket - paket yang masuk melalui *interface ethernet* dengan aplikasi Wireshark. Berhubung paket – paket yang masuk pada *ethernet* tersebut bukan hanya paket FTP, maka dari itu dilakukan *filtering* terlebih dahulu pada hasil *capture* Wireshark sehingga hanya muncul paket - paket FTP saja. Hal ini dimaksudkan agar yang ditampilkan hanya bagian – bagian yang diinginkan saja. Pengambilan nilai *transfer time* pada *summary* Wireshark dapat dilihat pada Gambar 4.6 di bawah ini :



Gambar 4.6 Pengambilan Nilai *Transfer Time*

Dapat di lihat pada gambar di atas data yang digunakan hanya data paket yang ada pada *displayed* bukan pada *captured*. Data *transfer time* diambil dari perbedaan rentang waktu antara paket pertama sampai dengan paket terakhir. Dari pengujian data yang dilakukan sebanyak 10 kali pada masing – masing file dan masing – masing konfigurasi didapat data rata – rata dari nilai *transfer time* seperti pada Tabel 4.2.

Tabel 4.2 Data Percobaan Nilai Rata – rata *Transfer Time*

File	<i>Transfer time(sec)</i>					
	<i>wired</i>			<i>wireless</i>		
	IPv4	IPv6	6to4	IPv4	IPv6	6to4
5M(5696602)	24.1658	28.4429	47.6721	19.4187	25.5801	35.6444
10M(10690520)	45.0141	50.9262	88.2698	35.6246	48.9125	63.1707
15M(15765158)	66.7171	73.5221	133.1448	52.1724	74.7138	93.2114
20M(20998049)	87.6171	97.3969	174.9251	69.5856	94.2059	134.6146

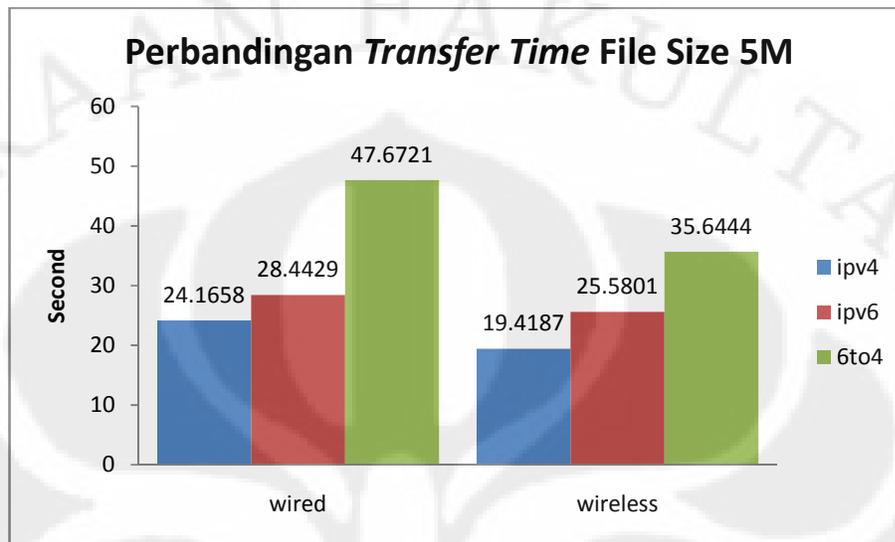
Berdasarkan hasil rata - rata pada Tabel 4.2, didapatkan bahwa terdapat pengaruh kapasitas data terhadap nilai *transfer time*. Semakin besar kapasitas data maka semakin besar pula nilai dari *transfer time*.

Untuk jaringan yang menggunakan *wired* secara persentase rata – rata, konfigurasi jaringan IPv4 murni memiliki nilai *transfer time* lebih kecil 9.26% dari konfigurasi jaringan IPv6 murni. Hal ini disebabkan karena pengaruh dari perbedaan *header* antara IPv4 dan IPv6, dimana *header* IPv6 lebih besar 2 kali dibanding IPv4. Panjang *header* pada IPv6 40 bytes, sedangkan panjang *header* pada IPv4 20 bytes. Sehingga dapat mempengaruhi dari waktu kirim dari keseluruhan paket tersebut. Dan jika dibandingkan dengan konfigurasi yang menggunakan jaringan *tunneling* 6to4, maka nilai *transfer time* jaringan IPv4 murni lebih kecil 49.89% dari konfigurasi jaringan *tunneling* 6to4. Hal ini dikarenakan adanya proses enkapsulasi dan dekapsulasi paket pada mekanisme *tunneling*. Selain *routing*, proses enkapsulasi dan dekapsulasi juga dapat menambah waktu *transfer* suatu file. Sedangkan pada konfigurasi IPv6 murni juga memiliki nilai jaringan *transfer time* lebih kecil 44.78% dari konfigurasi *tunneling* 6to4.

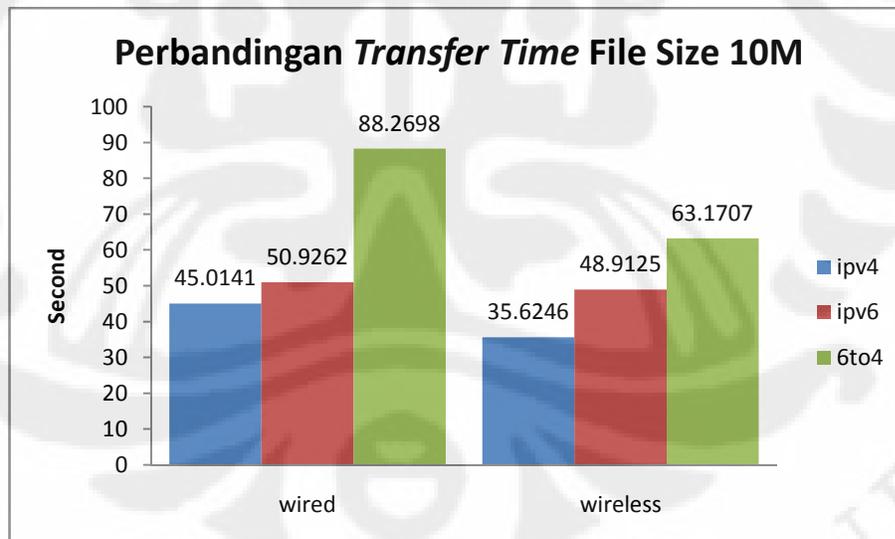
Untuk jaringan yang menggunakan *wireless* secara persentase rata – rata, konfigurasi jaringan IPv4 murni memiliki nilai *transfer time* lebih kecil 30.17% dari konfigurasi jaringan IPv6 murni. Hampir sama dengan jaringan menggunakan *wired*, hal ini disebabkan karena pengaruh *header* antara IPv4 dengan IPv6. Dan jika dibandingkan dengan konfigurasi yang menggunakan jaringan *tunneling* 6to4, maka nilai *transfer time* jaringan IPv4 murni lebih kecil 44.03% dari konfigurasi jaringan *tunneling* 6to4. Sedangkan pada konfigurasi IPv6 murni juga memiliki nilai jaringan *transfer time* lebih kecil 19.84% dari konfigurasi *tunneling* 6to4. Sama seperti perbandingan IPv4 dan IPv6 dengan mekanisme *tunneling*, hal ini dikarenakan adanya proses enkapsulasi dan dekapsulasi paket pada mekanisme *tunneling*. Selain *routing*, proses enkapsulasi dan dekapsulasi juga dapat menambah waktu *transfer* suatu file. Diagram perbandingan nilai *transfer time* dari empat beban file pada media *wired* dan *wireless* dapat dilihat pada Gambar 4.7 – 4.10.

Jika dibandingkan antar jenis konfigurasi, maka dapat disimpulkan konfigurasi IPv4 murni memiliki nilai *transfer time* lebih kecil dibandingkan dengan konfigurasi lain untuk media *wired* ataupun *wireless*. Hal ini

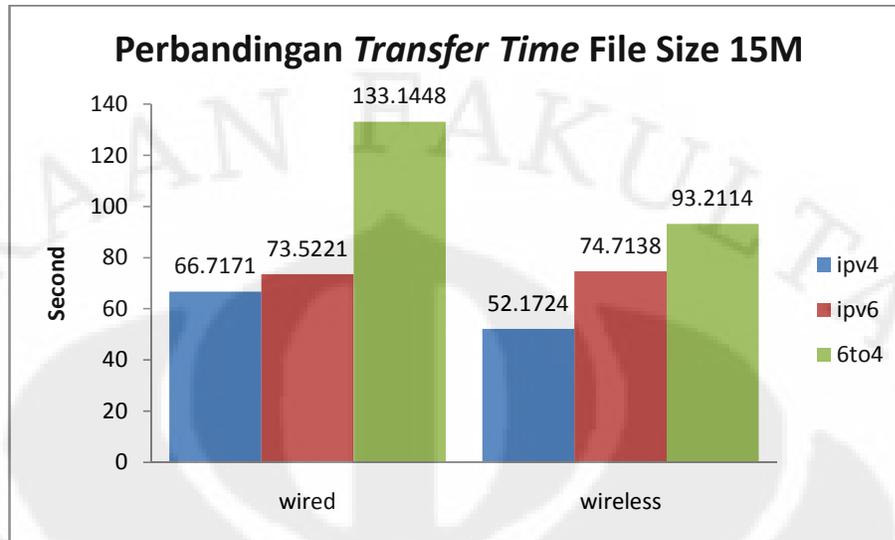
memperlihatkan jaringan IPv4 murni lebih baik untuk nilai *transfer time* dibandingkan pada jaringan IPv6 dan *tunneling* 6to4.



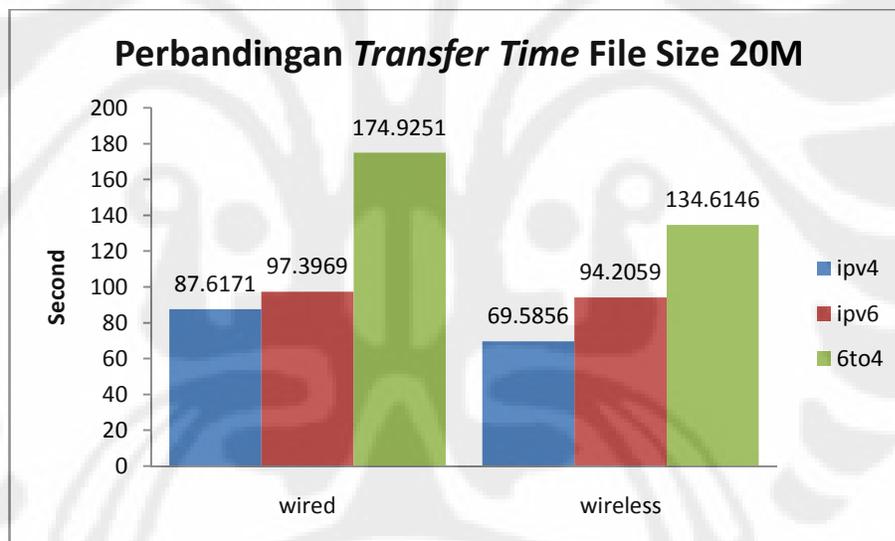
Gambar 4.7 Diagram Perbandingan *Transfer Time* Untuk File Size 5M



Gambar 4.8 Diagram Perbandingan *Transfer Time* Untuk File Size 10M



Gambar 4.9 Diagram Perbandingan *Transfer Time* Untuk File Size 15M



Gambar 4.10 Diagram Perbandingan *Transfer Time* Untuk File Size 20M

Dari diagram di atas dapat dilihat bahwa ada perbedaan pada media *wired* dan *wireless*. Jika membandingkan konfigurasi jaringan yang menggunakan media *wired* dan *wireless*, ternyata nilai *transfer time* paling kecil terjadi pada konfigurasi jaringan yang menggunakan *wireless*.

Secara persentase rata – rata, konfigurasi jaringan IPv4 yang menggunakan *wireless* memiliki nilai *transfer time* lebih kecil 19.64% dari konfigurasi jaringan IPv4 yang menggunakan *wired*. Demikian juga dengan

jaringan IPv6 *wireless* nilai *transfer timenya* lebih kecil 10.07% dibandingkan dengan IPv6 *wired*. Sedangkan untuk jaringan *tunneling* 6to4 lebih kecil 25.23% pada *wireless* daripada *wired*. Hal ini menunjukkan bahwa jaringan yang menggunakan media *wireless* lebih baik dalam *transfer time* dibandingkan dengan jaringan yang menggunakan media *wired*.

Hal yang memungkinkan perbandingan antara jaringan yang menggunakan *wired* dan *wireless* ini adalah pengaruh jarak dan kemampuan perangkat *wireless* tersebut. Jarak antara *client* dan *access point* dapat mempengaruhi karena semakin jauh jarak *client* dari *access point* membuat kecepatan koneksinya lebih lambat dibanding dengan *client* yang dekat dengan *access point*.

Pada pengujian ini jarak antara *access point* dengan *client* yang diuji cobakan adalah  $\pm 10$  m. Pengaruh dari *access point* BLR33N yang digunakan juga besar, karena *access point* dengan standarisasi WLAN 802.11n ini mempunyai kecepatan sampai 300 Mbps pada maksimum 100 m indoor dan 300 m outdoor. Sehingga tidak diragukan jika ternyata kecepatan performansi dari jaringan yang menggunakan *wireless* lebih baik dibandingkan jaringan yang menggunakan media *wired* pada parameter *transfer time* ini.

#### 4.2.2. Analisa *Throughput*

*Throughput* merupakan kecepatan transfer data rata – rata dari suksesnya paket yang dikirim per detik, pada umumnya menggunakan satuan *bit per second* (bps). Pengambilan parameter *transfer time* dilakukan dengan cara *download* file dari *server* ke *client*. Kemudian disaat yang bersamaan pada sisi *client* melakukan *capture* data atau penangkapan paket - paket yang masuk melalui *interface ethernet* dengan aplikasi Wireshark. Berhubung paket – paket yang masuk pada *ethernet* tersebut bukan hanya paket FTP, maka dari itu dilakukan *filtering* terlebih dahulu pada hasil *capture* Wireshark sehingga hanya muncul paket - paket FTP saja. Hal ini dimaksudkan agar yang ditampilkan hanya bagian – bagian yang diinginkan saja. Dapat dilihat pada Gambar 4.11 di bawah ini :

Traffic	Captured	Displayed	Marked
Packets	14345	6513	0
Between first and last packet	91.596 sec	73.289 sec	
Avg. packets/sec	156.611	88.867	
Avg. packet size	1236.787 bytes	948.652 bytes	
Bytes	17741705	6178572	
Avg. bytes/sec	193694.325	84304.272	
Avg. MBit/sec	1.550	0.674	

Gambar 4.11 Pengambilan Data *Throughput*

Dapat dilihat pada gambar di atas data yang digunakan hanya data paket yang ada pada *displayed* bukan pada *captured*. Data rata – rata yang tertangkap di Wireshark masih dalam satuan *bytes per second* sehingga perlu diubah menjadi satuan *bit per second* dengan mengalikan delapan dari data yang diambil dari Wireshark. Data keseluruhan hasil pengujian didapatkan nilai rata – rata parameter *throughput* yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Nilai Rata – rata Percobaan *Throughput*

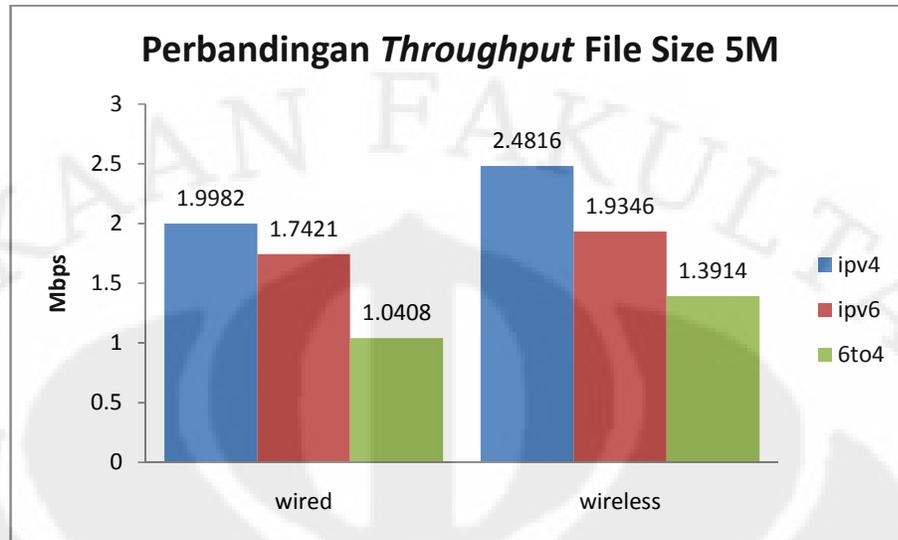
file	<i>throughput</i> (Mbps)					
	<i>wire</i>			<i>wireless</i>		
	IPv4	IPv6	6to4	IPv4	IPv6	6to4
5M(5696602)	1.9982	1.7421	1.0408	2.4816	1.9346	1.3914
10M(10690520)	2.0138	1.8258	1.0521	2.5371	1.8989	1.4731
15M(15765158)	2.0025	1.8613	1.0266	2.5541	1.8349	1.4712
20M(20998049)	2.0309	1.8708	1.0433	2.5522	1.9367	1.3571

Dari Tabel 4.3, dapat dilihat pada waktu ukuran file semakin besar nilai *throughput* tidak mengalami perubahan yang signifikan dan cenderung stabil sehingga perbedaan kapasitas ukuran file tidak mempengaruhi pada besarnya nilai *throughput*.

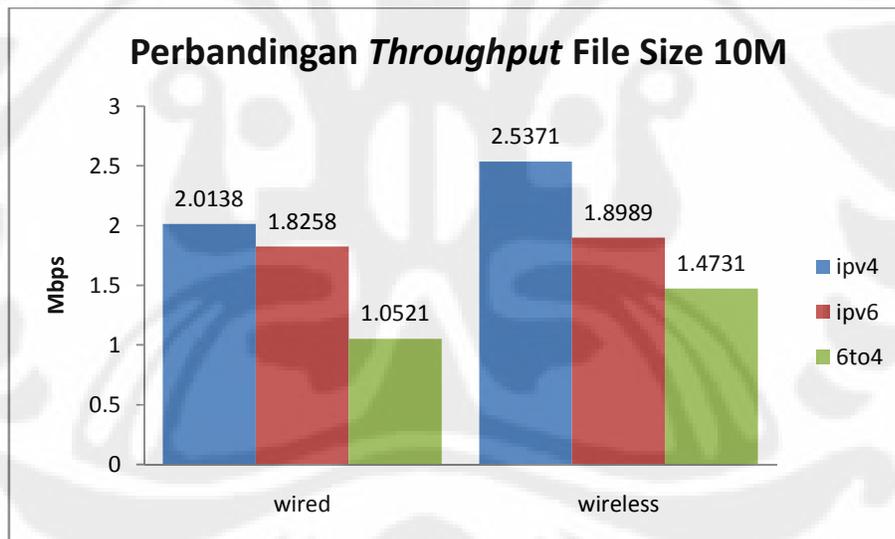
Untuk perbandingan jaringan yang menggunakan *wired* secara persentase rata – rata, nilai *throughput* IPv6 murni lebih kecil 12.82% dari nilai *throughput* IPv4 murni. Hal ini disebabkan karena pengaruh dari perbedaan *header* antara IPv4 dan IPv6, dimana *header* IPv6 lebih besar 2 kali dibanding IPv4. Panjang *header* pada IPv6 40 bytes, sedangkan panjang *header* pada IPv4 20 bytes. Sehingga dapat mempengaruhi kecepatan transfer data rata – rata suatu file. Jika dibandingkan dengan mekanisme *tunneling*, maka nilai *throughput* untuk konfigurasi *tunneling* 6to4 lebih kecil 47.93% dari IPv4 murni. Konfigurasi IPv6 murni juga memiliki nilai *throughput* lebih besar 40.26% dari konfigurasi *tunneling* 6to4. Hal ini dikarenakan adanya proses enkapsulasi dan dekapsulasi paket pada mekanisme *tunneling*. Selain *routing*, proses enkapsulasi dan dekapsulasi juga dapat mengurangi kecepatan rata – rata kirim keseluruhan paket.

Untuk perbandingan jaringan yang menggunakan *wireless* secara persentase rata – rata, nilai *throughput* IPv6 murni lebih kecil 22.04% dari nilai *throughput* IPv4 murni. Hampir sama dengan jaringan menggunakan *wired*, hal ini disebabkan karena pengaruh *header* antara IPv4 dengan IPv6. Jika dibandingkan dengan mekanisme *tunneling*, maka nilai *throughput* untuk konfigurasi *tunneling* 6to4 lebih kecil 43.93% dari IPv4 murni. Konfigurasi IPv6 murni juga memiliki nilai *throughput* lebih besar 28.08% dari konfigurasi *tunneling* 6to4. Sama seperti perbandingan IPv4 dan IPv6 dengan mekanisme *tunneling*, hal ini dikarenakan adanya proses enkapsulasi dan dekapsulasi paket pada mekanisme *tunneling*. Selain *routing*, proses enkapsulasi dan dekapsulasi juga dapat mengurangi kecepatan rata – rata kirim keseluruhan paket. Diagram perbandingan nilai *throughput* dari empat macam beban pada media *wired* dan *wireless* ada pada Gambar 4.12 - Gambar 4.15.

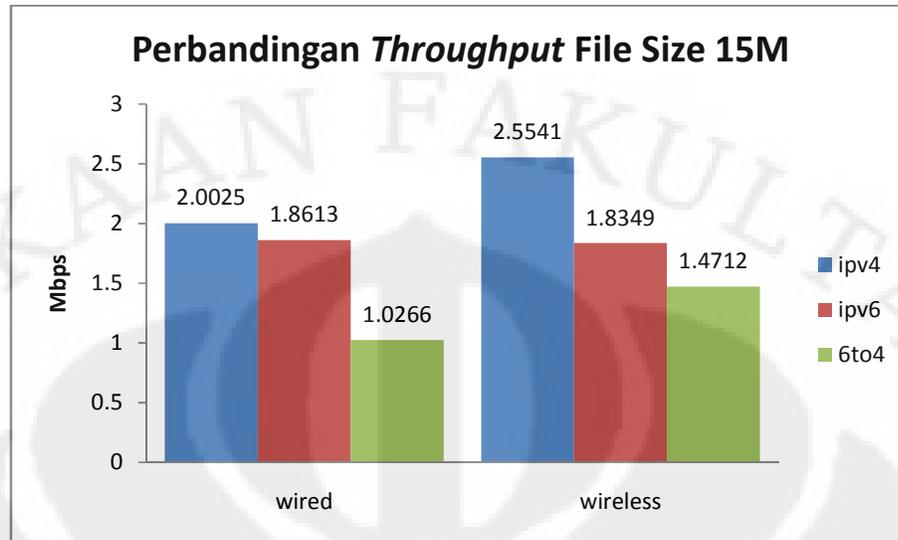
Jika dibandingkan antar jenis konfigurasi, maka dapat disimpulkan konfigurasi IPv4 murni memiliki nilai *throughput* lebih besar dibandingkan dengan konfigurasi lain untuk media *wired* ataupun *wireless*. Hal ini memperlihatkan jaringan IPv4 murni lebih baik untuk nilai *throughput* dibandingkan pada jaringan IPv6 dan *tunneling* 6to4.



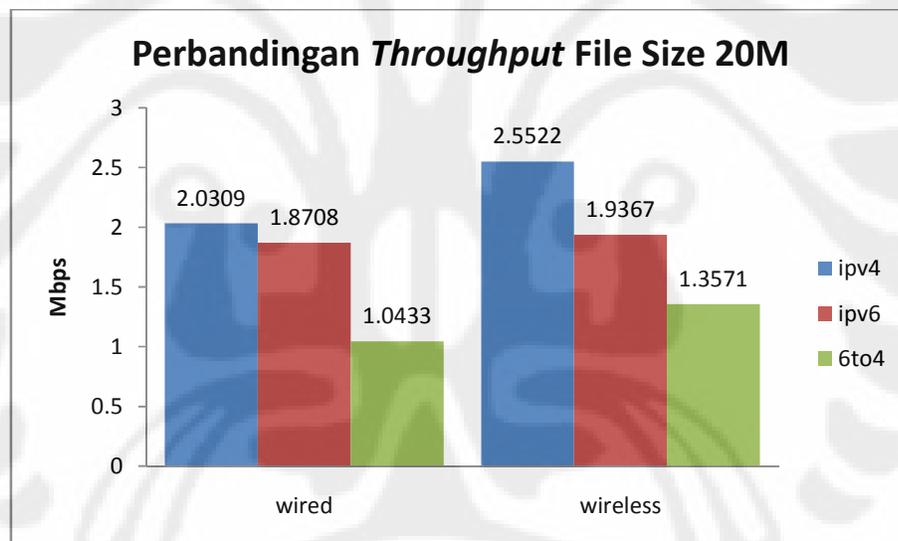
Gambar 4.12 Diagram Perbandingan *Throughput* Untuk File Size 5M



Gambar 4.13 Diagram Perbandingan *Throughput* Untuk File Size 10M



Gambar 4.14 Diagram Perbandingan *Throughput* Untuk File Size 15M



Gambar 4.15 Diagram Perbandingan *Throughput* Untuk File Size 20M

Dari diagram di atas dapat dilihat bahwa ada perbedaan pada media *wired* dan *wireless*. Jika membandingkan konfigurasi jaringan yang menggunakan media *wired* dan *wireless*, ternyata nilai *throughput* paling besar terjadi pada konfigurasi jaringan yang menggunakan *wireless*.

Secara persentase rata – rata, konfigurasi jaringan IPv4 yang menggunakan *wireless* memiliki nilai *throughput* lebih besar 20.63% dari konfigurasi jaringan IPv4 yang menggunakan *wired*. Demikian juga dengan

jaringan IPv6 *wireless* nilai *throughput* nya lebih besar 3.85% dibandingkan dengan IPv6 *wired*. Sedangkan untuk jaringan *tunneling* 6to4 lebih besar 28.58% pada *wireless* daripada *wired*. Hal ini menunjukkan bahwa jaringan yang menggunakan media *wireless* lebih baik dalam *throughput* dibandingkan dengan jaringan yang menggunakan media *wired*.

Seperti halnya perbedaan media *wired* dan *wireless* pada *transfer time*, pada *throughput* juga mengalami perbedaan. Dengan alasan yang hampir sama yaitu perbedaan jarak dan kemampuan perangkat *wireless*. Jarak antara *client* dan *access point* dapat mempengaruhi, semakin jauh jarak *client* dari *access point* membuat kecepatan koneksinya lebih lambat dibanding dengan *client* yang dekat dengan *access point*.

Pada pengujian ini jarak antara *access point* dengan *client* yang diuji cobakan adalah  $\pm 10$  m. Pengaruh dari *access point* BLR33N yang digunakan juga besar, karena *access point* dengan standarisasi WLAN 802.11n ini mempunyai kecepatan sampai 300 Mbps pada maksimum 100 m indoor dan 300 m outdoor. Sehingga tidak diragukan jika ternyata kecepatan performansi dari jaringan yang menggunakan *wireless* lebih baik dibandingkan jaringan yang menggunakan media *wired* pada parameter *throughput* ini.

#### 4.2.3. Analisa Delay

*Delay* adalah waktu tunda dari waktu yang seharusnya dari suksesnya seluruh paket yang diterima. Parameter *delay* dihitung dengan cara membagi nilai *transfer time* dengan jumlah bit data. Ditunjukkan pada rumus dibawah ini :

$$\text{delay (sec)} = \frac{\text{transfer time (sec)}}{\text{jumlah bit}}$$

Hasil pengambilan data secara keseluruhan untuk parameter *delay* dapat dilihat pada bagian lampiran sedangkan untuk hasil rata – rata dari perhitungan *delay* dapat dilihat pada Tabel 4.4 dibawah ini :

Tabel 4.4 Nilai Rata – rata Percobaan *Delay*

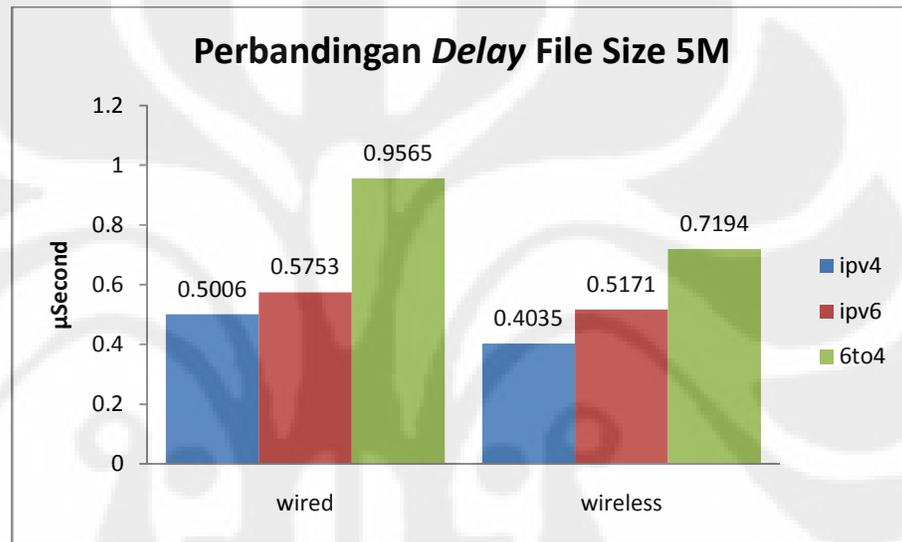
file	delay( $\mu$ sec)					
	wired			wireless		
	ipv4	ipv6	6to4	ipv4	ipv6	6to4
5M(5696602)	0.5006	0.5753	0.9565	0.4035	0.5171	0.7194
10M(10690520)	0.4969	0.5489	0.952	0.3943	0.5259	0.6795
15M(15765158)	0.4994	0.5375	0.9736	0.3908	0.5457	0.6798
20M(20998049)	0.4924	0.5346	0.9556	0.3922	0.5175	0.7369

Dari hasil perhitungan rata – rata delay pada Tabel 4.4, dapat dilihat pada waktu ukuran file semakin besar nilai *delay* tidak mengalami perubahan yang signifikan dan cenderung stabil sehingga perbedaan kapasitas ukuran file tidak mempengaruhi pada besarnya nilai *delay*.

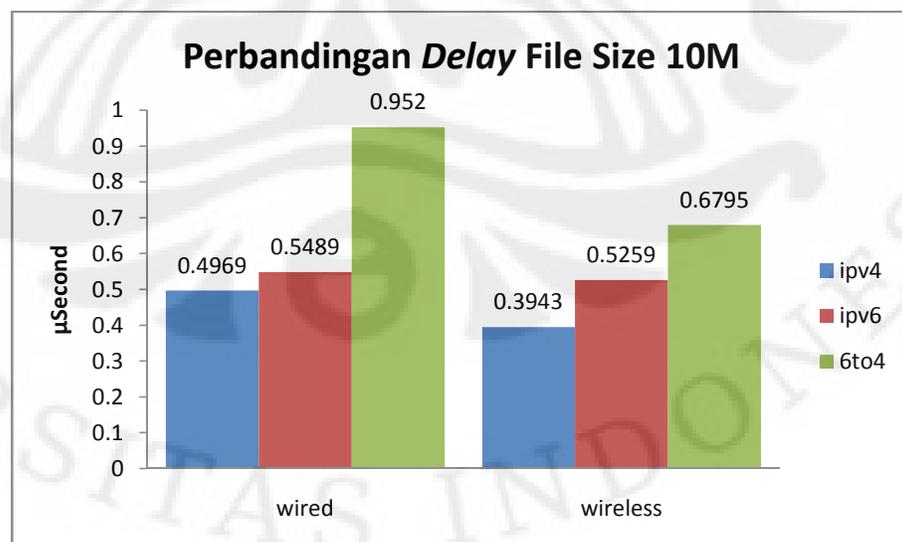
Untuk perbandingan jaringan yang menggunakan *wired* secara persentase rata – rata, konfigurasi IPv4 murni memiliki nilai *delay* lebih kecil 9.47% dari konfigurasi IPv6 murni. Ini disebabkan karena besar kapasitas *header* IPv6 dua kali lebih besar daripada IPv4. Panjang *header* pada IPv6 40 bytes, sedangkan panjang *header* pada IPv4 20 bytes. Sehingga dapat mempengaruhi waktu *delay* suatu file. Jika dibandingkan dengan konfigurasi *tunneling*, secara persentase rata – rata konfigurasi IPv4 murni memiliki *delay* lebih kecil 47.80% dari konfigurasi 6to4. Untuk *delay* dari konfigurasi IPv6 murni lebih kecil 42.34% dari konfigurasi yang menggunakan *tunneling* 6to4. Hal ini dikarenakan adanya proses enkapsulasi dan dekapsulasi paket pada mekanisme *tunneling*. Selain *routing*, proses enkapsulasi dan dekapsulasi juga dapat menambah waktu *delay* suatu file.

Untuk perbandingan jaringan yang menggunakan *wireless* secara persentase rata – rata, konfigurasi IPv4 murni memiliki nilai *delay* lebih kecil 25.02% dari konfigurasi IPv6 murni. Ini disebabkan karena besar kapasitas *header* IPv6 dua kali lebih besar daripada IPv4. Panjang *header* pada IPv6 40 bytes, sedangkan panjang *header* pada IPv4 20 bytes. Jika dibandingkan dengan konfigurasi *tunneling*, secara persentase rata – rata konfigurasi IPv4 murni memiliki *delay* lebih kecil 41.97% dari konfigurasi 6to4. Untuk *delay* dari konfigurasi IPv6 murni lebih kecil 22.61% dari konfigurasi yang menggunakan *tunneling* 6to4.

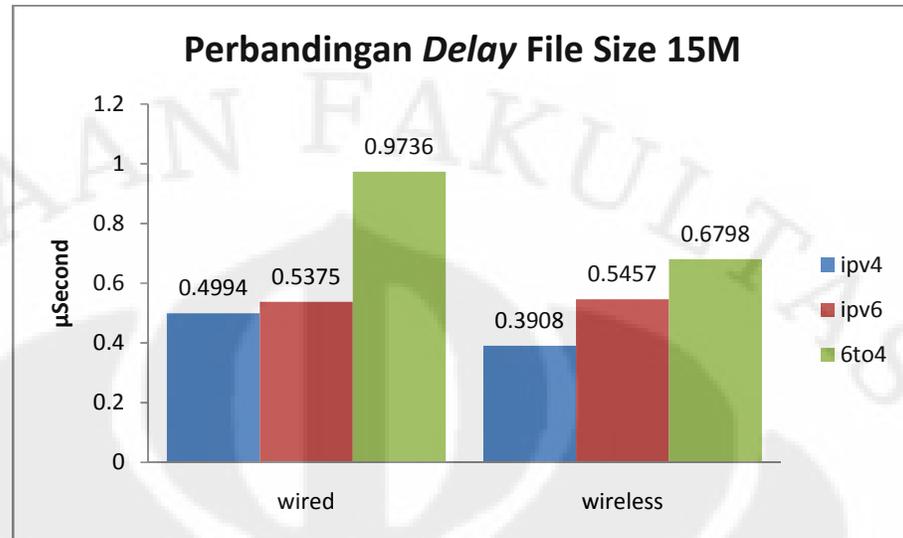
Ini menunjukkan bahwa selain proses *routing*, proses *tunneling* juga dapat menambah *delay* jaringan. Proses *tunneling* merupakan proses enkapsulasi dan dekapsulasi paket – paket yang masuk. Untuk diagram perbandingan nilai *delay* pada keempat beban file pada media *wired* dan *wireless* dapat dilihat pada Gambar 4.16 – 4.19. Maka dapat disimpulkan konfigurasi IPv4 murni memiliki nilai *delay* lebih kecil dibandingkan dengan konfigurasi lain untuk media *wired* ataupun *wireless*. Hal ini memperlihatkan jaringan IPv4 murni lebih baik untuk nilai *delay* dibandingkan pada jaringan IPv6 dan *tunneling* 6to4.



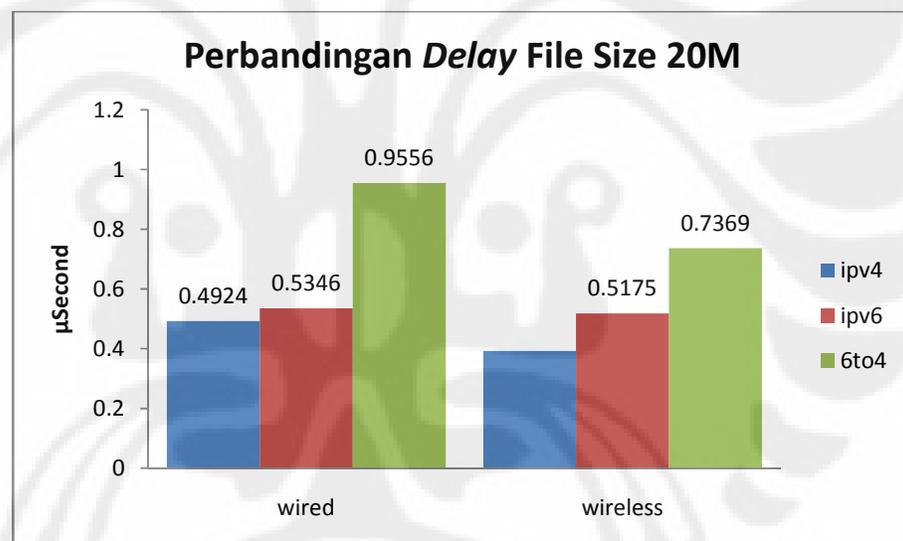
Gambar 4.16 Diagram Perbandingan *Delay* Untuk File Size 5M



Gambar 4.17 Diagram Perbandingan *Delay* Untuk File Size 10M



Gambar 4.18 Diagram Perbandingan Delay Untuk File Size 15M



Gambar 4.19 Diagram Perbandingan Delay Untuk File Size 20M

Dari diagram di atas dapat dilihat bahwa ada perbedaan pada media *wired* dan *wireless*. Jika membandingkan konfigurasi jaringan yang menggunakan media *wired* dan *wireless*, ternyata nilai *delay* paling kecil terjadi pada konfigurasi jaringan yang menggunakan *wireless*.

Secara persentase rata – rata, konfigurasi jaringan IPv4 yang menggunakan *wireless* memiliki nilai *delay* lebih kecil 20.64% dari konfigurasi jaringan IPv4 yang menggunakan *wired*. Demikian juga dengan jaringan IPv6

*wireless* nilai *delay*nya lebih kecil 4.19% dibandingkan dengan IPv6 *wired*. Sedangkan untuk jaringan *tunneling* 6to4 lebih kecil 28.62% pada *wireless* daripada *wired*. Hal ini menunjukkan bahwa jaringan yang menggunakan media *wireless* lebih baik dalam *delay* dibandingkan dengan jaringan yang menggunakan media *wired*.

Seperti halnya perbedaan media *wired* dan *wireless* pada *transfer time* dan *throughput* pada *delay* juga mengalami perbedaan. Dengan alasan yang hampir sama yaitu perbedaan jarak dan kemampuan perangkat *wireless*. Jarak antara *client* dan *access point* dapat mempengaruhi, semakin jauh jarak *client* dari *access point* membuat kecepatan koneksinya lebih lambat dibanding dengan *client* yang dekat dengan *access point*.

Pada pengujian ini jarak antara *access point* dengan *client* yang diuji cobakan adalah  $\pm 10$  m. Pengaruh dari *access point* BLR33N yang digunakan juga besar, karena *access point* dengan standarisasi WLAN 802.11n ini mempunyai kecepatan sampai 300 Mbps pada maksimum 100 m indoor dan 300 m outdoor. Sehingga tidak diragukan jika ternyata kecepatan performansi dari jaringan yang menggunakan *wireless* lebih baik dibandingkan jaringan yang menggunakan media *wired* pada parameter *delay* ini.

## BAB 5

### KESIMPULAN

1. Berdasarkan parameter *transfer time* pada *wireless*, IPv4 murni memiliki nilai *transfer time* lebih kecil 30.17% dari IPv6 murni; IPv4 murni lebih kecil 44.03% dari *tunneling* 6to4; sedangkan IPv6 murni lebih kecil 19.84% dari konfigurasi *tunneling* 6to4.
2. Berdasarkan parameter *transfer time* keseluruhan, IPv4 *wireless* memiliki nilai *transfer time* lebih kecil 19.64% dari IPv4 *wired*; IPv6 *wireless* lebih kecil 10.07% dibandingkan IPv6 *wired*; sedangkan *tunneling* 6to4 lebih kecil 25.23% pada *wireless* daripada *wired*.
3. Berdasarkan parameter *throughput* pada *wireless*, IPv6 murni memiliki nilai *throughput* lebih kecil 22.04% dari IPv4 murni; *tunneling* 6to4 lebih kecil 43.93% dari IPv4 murni; sedangkan *tunneling* 6to4 lebih kecil 28.08% dari IPv6 murni.
4. Berdasarkan parameter *throughput* keseluruhan, IPv4 *wireless* memiliki nilai *throughput* lebih besar 20.63% dari IPv4 *wired*; IPv6 *wireless* lebih besar 3.85% dibandingkan IPv6 *wired*; sedangkan *tunneling* 6to4 lebih besar 28.58% pada *wireless* daripada *wired*.
5. Berdasarkan parameter *delay* pada *wireless*, IPv4 murni memiliki nilai *delay* lebih kecil 25.02% dari IPv6 murni; IPv4 murni lebih kecil 41.97% dari *tunneling* 6to4; sedangkan IPv6 murni lebih kecil 22.614% dari *tunneling* 6to4.
6. Berdasarkan parameter *delay* keseluruhan, IPv4 *wireless* memiliki nilai *delay* lebih kecil 20.64% dari IPv4 *wired*; IPv6 *wireless* juga lebih kecil 4.19% dibandingkan IPv6 *wired*; sedangkan *tunneling* 6to4 lebih kecil 28.62% pada *wireless* daripada *wired*.
7. Perbedaan *header* IPv4 dan IPv6 serta proses enkapsulasi dan dekapsulasi *tunneling* mempengaruhi naik turunnya performansi jaringan IPv4, IPv6 dan *tunneling* 6to4 untuk parameter *transfer time*, *throughput* dan *delay*.
8. Perbedaan performansi jaringan antara media *wired* dan *wireless*, dipengaruhi kemampuan perangkat *access point* dengan kecepatan 300 Mbps.

## DAFTAR REFERENSI

- [1]. Introduction to IP Version 6, Microsoft Corporation's, February 2008.
- [2]. CISCO Networking Academy, "Network Fundamentals", CCNA Exploration 4.0 (CISCO SYSTEMS, Inc., 2007)
- [3]. Somad, Wahidi."Interkoneksi IPv6 dan IPv4 dengan Mekanisme Automatic Tunneling", IlmuKomputer.com, 2003
- [4]. Wikipedia,"Tunneling Protocol", Diakses Maret 2010 dari Wikipedia.  
[http://en.wikipedia.org/wiki/Tunneling\\_protocol](http://en.wikipedia.org/wiki/Tunneling_protocol)
- [5]. Wikipedia,"6to4", Diakses April 2010 dari Wikipedia.  
<<http://en.wikipedia.org/wiki/6to4>>
- [6]. Dwi Hantono, Gunadi. "WiFi (Wireless LAN) Jaringan Komputer Tanpa Kabel". Informatika, 2009.
- [7]. Wiryana, I Made. "File Transfer Protocol", March 2007.
- [8]. Sinaga, Andreas. "Aplikasi File Transfer Protokol Berbasis TCP/IP".
- [9]. Tremblay, Jean Francois. Lind, Mikael. "IPv6 Tunneling Techniques", Hexago Inc., September,2006.
- [10]. <http://www.ipv6.org>

## LAMPIRAN 1 : Konfigurasi Router Pada Jaringan IPv4 Murni

```
1. ROUTER_1#sh run
Building configuration...
Current configuration : 936 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
ip cef
!
!
Ip auth-proxy max-nodata-conns 3
Ip admission max-nodata-conns 3
!
Multilink bundle-name authenticated
!
!
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.15.1 255.255.255.252
 duplex auto
 speed auto
!
router rip
 version 2
 passive-interface FastEthernet0/0
 network 192.168.10.0
 network 192.168.15.0
 no auto-summary
!
!
dial-peer cor custom
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
!
!
End
```

(LANJUTAN)

```

2. ROUTER_2#sh run
Building configuration...
Current configuration : 936 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
!
memory-size iomem 5
no aaa new-model
ip cef
!
!
Ip auth-proxy max-nodata-conns 3
Ip admission max-nodata-conns 3
!
Multilink bundle-name authenticated
!
!
interface FastEthernet0/0
 ip address 192.168.20.1 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.15.2 255.255.255.252
 duplex auto
 speed auto
!
!
router rip
 version 2
 passive-interface FastEthernet0/0
 network 192.168.20.0
 network 192.168.15.0
 no auto-summary
!
!
dial-peer cor custom
!
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
!
!
End

```

## LAMPIRAN 2 : Konfigurasi Router Pada Jaringan IPv6 Murni

```

1. ROUTER_1#sh run
Building configuration...
Current configuration : 954 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
ip cef
!
!
Ip auth-proxy max-nodata-conns 3
Ip admission max-nodata-conns 3
ipv6 unicast-routing
!
!
Multilink bundle-name authenticated
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
ipv6 address 2001:0:0:A::1/64
ipv6 enable
ipv6 rip process1 enable
!
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
ipv6 address 2001:0:0:B::1/64
ipv6 enable
ipv6 rip process1 enable
!
!
dial-peer cor custom
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
!
!
End

```

(LANJUTAN)

```

2. ROUTER_2#sh run
Building configuration...
Current configuration : 954 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
ip subnet-zero
ip cef
!
!
Ip auth-proxy max-nodata-conns 3
Ip admission max-nodata-conns 3
ipv6 unicast-routing
!
!
Multilink bundle-name authenticated
!
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
ipv6 address 2001:0:0:C::1/64
ipv6 enable
ipv6 rip process1 enable
!
!
interface FastEthernet0/1
no ip address
duplex auto
speed auto
ipv6 address 2001:0:0:B::2/64
ipv6 enable
ipv6 rip process1 enable
!
!
dial-peer cor custom
!
line con 0
password cisco
login
line aux 0
line vty 0 4
password cisco
login
!
!
End

```

### LAMPIRAN 3 : Konfigurasi Router Pada Jaringan *Tunneling 6to4*

```

1. ROUTER_1#sh run
Building configuration...

Current configuration : 1221 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
memory-size iomem 5
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ipv6 unicast-routing
!
!
interface Tunnel0
no ip address
no ip redirects
ipv6 unnumber FastEthernet0/1
ipv6 enable
tunnel source FastEthernet0/1
tunnel mode ipv6ip 6to4
!
interface FastEthernet0/0
no ip address
duplex auto
speed auto
ipv6 address 2001:0:0:A::1/64
ipv6 enable
ipv6 rip process1 enable
!
interface FastEthernet0/1
ip address 192.168.15.1 255.255.255.252
duplex auto
speed auto
!
router rip
version 2
network 192.168.15.0

```

(LANJUTAN)

```

no ip http server
ip classless
!
!
ipv6 route 2002::/16 Tunnel0
ipv6 route ::/0 2002:COA8:F01:1::1
ipv6 router rip process1
!
!
dial-peer cor custom
!
!
line con 0
  password cisco
  login
line aux 0
line vty 0 4
  password cisco
  login
!
!
End

```

2. ROUTER\_2#sh run  
Building configuration...

```

Current configuration : 1221 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
ip subnet-zero
!
!
no ip domain lookup
!
ip cef
ipv6 unicast-routing
!
!
interface Tunnel0
  no ip address
  no ip redirects
  ipv6 unnumbered FastEthernet0/1
  ipv6 enable

```

(LANJUTAN)

```
tunnel source FastEthernet0/1
 tunnel mode ipv6ip 6to4
 !
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address 2001:0:0:B::1/64
 ipv6 enable
 ipv6 rip process1 enable
 !
interface FastEthernet0/1
 ip address 192.168.15.2 255.255.255.252
 duplex auto
 speed auto
 !
router rip
 version 2
 network 192.168.15.0
 !
no ip http server
 ip classless
 !
 !
ipv6 route 2002::/16 Tunnel0
ipv6 route ::/0 2002:COA8:F01:1::1
ipv6 router rip process1
 !
 !
dial-peer cor custom
 !
 !
line con 0
 password cisco
 login
line aux 0
line vty 0 4
 password cisco
 login
 !
 !
End
```

### LAMPIRAN 4 : Hasil Capture Wireshark Konfigurasi IPv4 Murni

The screenshot shows the Wireshark interface with a capture filter applied: `(ip.addr eq 192.168.10.10 and ip.addr eq 192.168.20.10) and (tcp.port eq 2)`. The packet list pane displays a series of FTP data packets. Packet 14575 is highlighted, showing a TCP window update and an ACK for an FTP data packet.

No.	Time	Source	Destination	Protocol	Info
14556	49.641120	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14557	49.641161	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14558	49.641196	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14559	49.642232	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14560	49.642240	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14561	49.642253	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 892 bytes
14562	49.642264	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14563	49.642271	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14564	49.642278	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14565	49.642286	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14566	49.642293	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14567	49.642300	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 892 bytes
14568	49.642307	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14569	49.642314	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14570	49.642321	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14571	49.642328	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14572	49.642335	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14573	49.642342	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 892 bytes
14574	49.642843	192.168.20.10	192.168.10.10	TCP	d-cinema-rrp > ftp-data [ACK] Seq=1 Ack=9314305 win=45504 Len=0
14575	49.644869	192.168.20.10	192.168.10.10	TCP	[TCP Window Update] d-cinema-rrp > ftp-data [ACK] Seq=1 Ack=9314305 win=64240 Len=0
14576	49.683740	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14577	49.691359	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14578	49.692305	192.168.20.10	192.168.10.10	TCP	d-cinema-rrp > ftp-data [ACK] Seq=1 Ack=9317225 win=64240 Len=0
14579	49.700115	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14580	49.708140	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14581	49.708916	192.168.20.10	192.168.10.10	TCP	d-cinema-rrp > ftp-data [ACK] Seq=1 Ack=9320145 win=64240 Len=0
14582	49.716251	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes
14583	49.721161	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 892 bytes
14584	49.721709	192.168.20.10	192.168.10.10	TCP	d-cinema-rrp > ftp-data [ACK] Seq=1 Ack=9322497 win=64240 Len=0
14585	49.725341	192.168.10.10	192.168.20.10	FTP-DATA	FTP Data: 1460 bytes

Packet 14575 details:

- Frame 1791 (1514 bytes on wire, 1514 bytes captured)
- Ethernet II, Src: c2:01:12:ac:00:00 (c2:01:12:ac:00:00), Dst: vmware\_be:f1:a5 (00:0c:29:be:f1:a5)
- Internet Protocol, Src: 192.168.10.10 (192.168.10.10), Dst: 192.168.20.10 (192.168.20.10)
- Transmission Control Protocol, Src Port: ftp-data (20), Dst Port: d-cinema-rrp (1173), Seq: 553245, Ack: 1, Len: 1460
- FTP Data

Hex dump:

```

0000 00 0c 29 be f1 a5 c2 01 12 ac 00 00 08 00 45 00  ..)...E.
0010 05 dc 68 18 40 00 7e 06 ef 9e c0 a8 0a 0a c0 a8  ..h.~.....
0020 14 0a 00 14 04 95 4e 60 a9 5d 0f db b0 01 50 10  .....N'.....P.
0030 80 00 e7 eb 00 00 c2 eb 73 e2 c9 e4 c1 a8 a7 13  .....S.....
0040 06 77 67 d3 2a 1d ac df 82 d4 73 96 0f d3 ad 51  .wg.*...S...Q
0050 77 0b 0d 67 2f 24 88 e1 4b 01 50 f5 23 65 48 71  'h b n qf
  
```

### LAMPIRAN 5 : Hasil Capture Wireshark Konfigurasi IPv6 Murni

The screenshot shows a Wireshark capture of network traffic. The filter is set to `(ipv6.addr eq 2001:0:0:a::10 and ipv6.addr eq 2001:0:0:c::10) and (tcp.port > 20)`. The packet list shows a series of FTP data packets and control messages between source `2001:0:0:a::10` and destination `2001:0:0:c::10`. Packet 756 is highlighted, showing a TCP ACK for sequence 11521.

No.	Time	Source	Destination	Protocol	Info
739	7.447253	2001:0:0:a::10	2001:0:0:c::10	TCP	ftp-data > nimreg [SYN] Seq=0 win=16384 Len=0 MSS=1440
740	7.449389	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [SYN, ACK] Seq=0 Ack=1 win=17280 Len=0 MSS=1440
741	7.458960	2001:0:0:a::10	2001:0:0:c::10	TCP	ftp-data > nimreg [ACK] Seq=1 Ack=1 win=17280 Len=0
742	7.493773	2001:0:0:a::10	2001:0:0:c::10	TCP	[TCP window update] ftp-data > nimreg [ACK] Seq=1 Ack=1 win=32768 Len=0
744	7.509342	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
745	7.513175	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
746	7.513291	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=2881 win=17280 Len=0
747	7.526281	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
748	7.528432	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=4321 win=17280 Len=0
749	7.535642	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
750	7.538953	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
751	7.538992	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=7201 win=17280 Len=0
752	7.549360	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
753	7.550206	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=8641 win=17280 Len=0
754	7.560762	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
755	7.566849	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
756	7.566896	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=11521 win=17280 Len=0
757	7.571703	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
758	7.571761	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=12961 win=17280 Len=0
759	7.575935	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
760	7.580475	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
761	7.580519	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=15841 win=17280 Len=0
762	7.587323	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
763	7.587379	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=17281 win=17280 Len=0
764	7.594581	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
765	7.596248	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
766	7.596287	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=20161 win=17280 Len=0
767	7.601922	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes
768	7.602892	2001:0:0:c::10	2001:0:0:a::10	TCP	nimreg > ftp-data [ACK] Seq=1 Ack=21601 win=17280 Len=0
769	7.606452	2001:0:0:a::10	2001:0:0:c::10	FTP-DATA	FTP Data: 1440 bytes

Frame 756 (74 bytes on wire, 74 bytes captured)  
 Ethernet II, Src: vmware\_bef1:a5 (00:0c:29:be:f1:a5), Dst: c2:0a:11:cc:00:00 (c2:0a:11:cc:00:00)  
 Internet Protocol Version 6  
 Transmission Control Protocol, Src Port: nimreg (1059), Dst Port: ftp-data (20), Seq: 1, Ack: 11521, Len: 0

```

0000 c2 0a 11 cc 00 00 0c 29 be f1 a5 86 dd 60 00  ....).
0010 00 00 00 14 06 40 20 01 00 00 00 00 0c 00 00  ....@
0020 00 00 00 00 00 10 20 01 00 00 00 00 0a 00 00  ....
0030 00 00 00 00 00 10 04 23 00 14 cb 17 14 d4 d2 9d  ....#
0040 f5 5e 50 10 43 80 7f fd 00 00  ....AP.C...
  
```

File: D:\91lan9\dani acer\GILANG\_SKRIPSI\masi... Packets: 17991 Displayed: 12192 Marked: 0 Profile: Default

## LAMPIRAN 6: Hasil Capture Wireshark Konfigurasi Tunneling 6to4

The screenshot displays a Wireshark capture of network traffic. The filter is set to `(ipv6.addr eq 2001::a:21e:c9ff:fe01:45b6 and ipv6.addr eq 2001:0:0:c::10)`. The packet list shows a series of FTP data packets (1420 bytes) and TCP window updates. The selected packet (No. 879) is expanded to show the following details:

- Frame 879 (1494 bytes on wire, 1494 bytes captured)
- Ethernet II, Src: c2:0d:14:10:00:00 (c2:0d:14:10:00:00), Dst: vmware\_bef1:a5 (00:0c:29:be:f1:a5)
- Internet Protocol Version 6
- Transmission Control Protocol, Src Port: ftp-data (20), Dst Port: netinfo-local (1033), Seq: 1, Ack: 1, Len: 1420
- FTP Data

The raw packet bytes are shown in hexadecimal and ASCII format at the bottom of the packet details pane.

**LAMPIRAN 7 : Data Hasil Uji Coba Konfigurasi IPv4 Murni Dengan Media Wired**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	23.619	24.307	25.078	23.679	24.063	24.085	24.385	24.216	24.022	24.204	<b>24.1658</b>
10M(10690520)	45.537	44.431	44.629	43.877	45.651	48.326	44.289	44.221	44.462	44.717	<b>45.0141</b>
15M(15765158)	66.448	66.556	66.597	67.575	67.458	67.291	67.145	66.689	65.517	65.895	<b>66.7171</b>
20M(20998049)	87.384	86.336	87.799	87.821	86.201	88.865	88.486	87.174	87.886	88.219	<b>87.6171</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	2.0437	1.9861	1.9251	2.0386	2.0061	2.0042	1.9801	1.9938	2.0095	1.9943	<b>1.9982</b>
10M(10690520)	1.9884	2.0389	2.0301	2.0645	1.9845	1.8741	2.0452	2.0485	2.0377	2.0258	<b>2.0138</b>
15M(15765158)	2.0103	2.0071	2.0059	1.9771	1.9802	1.9852	1.9895	2.0032	2.0388	2.0273	<b>2.0025</b>
20M(20998049)	2.0362	2.0609	2.0265	2.0261	2.0641	2.0022	2.0108	2.0411	2.0245	2.0168	<b>2.0309</b>
File (bytes)	Delay (μsecond)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.4893	0.5035	0.5194	0.4905	0.4984	0.4989	0.5051	0.5016	0.4976	0.5014	<b>0.5006</b>
10M(10690520)	0.5029	0.4905	0.4926	0.4843	0.5039	0.5335	0.4889	0.4881	0.4907	0.4936	<b>0.4969</b>
15M(15765158)	0.4974	0.4982	0.4985	0.5058	0.5049	0.5037	0.5026	0.4991	0.4905	0.4932	<b>0.4994</b>
20M(20998049)	0.4911	0.4852	0.4934	0.4935	0.4844	0.4994	0.4973	0.4899	0.4939	0.4958	<b>0.4924</b>

**LAMPIRAN 8 : Data Hasil Uji Coba Konfigurasi IPv6 Murni Dengan Media Wired**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	27.496	31.821	30.529	27.911	28.016	27.381	27.562	27.928	28.274	27.511	<b>28.4429</b>
10M(10690520)	48.902	55.635	49.556	48.691	49.563	55.983	50.561	49.527	49.373	51.471	<b>50.9262</b>
15M(15765158)	73.291	77.446	72.462	74.336	73.966	71.871	72.326	73.527	72.801	73.195	<b>73.5221</b>
20M(20998049)	96.781	97.322	96.791	97.863	96.901	97.195	95.565	97.619	99.671	98.261	<b>97.3969</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	1.7967	1.5531	1.6188	1.7711	1.7637	1.8045	1.7933	1.7758	1.7479	1.7956	<b>1.7421</b>
10M(10690520)	1.8956	1.6672	1.8714	1.9045	1.8713	1.6588	1.8368	1.8722	1.8778	1.8022	<b>1.8258</b>
15M(15765158)	1.8656	1.7663	1.8867	1.8402	1.8485	1.9032	1.8911	1.8618	1.8812	1.8688	<b>1.8613</b>
20M(20998049)	1.8818	1.8722	1.8814	1.8611	1.8802	1.8742	1.9055	1.8662	1.8311	1.8544	<b>1.8708</b>
File (bytes)	Delay (µsecond)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.5563	0.6438	0.6177	0.5647	0.5669	0.5541	0.5576	0.5631	0.5722	0.5567	<b>0.5753</b>
10M(10690520)	0.5273	0.5999	0.5343	0.5251	0.5343	0.6027	0.5443	0.5341	0.5324	0.5551	<b>0.5489</b>
15M(15765158)	0.5359	0.5663	0.5298	0.5436	0.5408	0.5254	0.5288	0.5372	0.5317	0.5352	<b>0.5375</b>
20M(20998049)	0.5313	0.5343	0.5314	0.5373	0.5319	0.5336	0.5246	0.5359	0.5463	0.5394	<b>0.5346</b>

**LAMPIRAN 9 : Data Hasil Uji Coba Konfigurasi Tunneling 6to4 Dengan Media Wired**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	47.721	45.477	49.928	48.501	50.798	51.964	49.221	43.447	45.174	44.479	<b>47.6721</b>
10M(10690520)	85.659	87.691	85.604	95.619	88.767	87.152	91.125	90.019	89.275	81.787	<b>88.2698</b>
15M(15765158)	132.575	141.704	133.955	128.181	129.128	133.461	129.674	134.784	128.642	139.344	<b>133.1448</b>
20M(20998049)	182.522	167.592	160.433	170.947	178.484	171.811	182.469	170.908	184.991	179.093	<b>174.9251</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	1.0354	1.0902	0.9933	1.0188	0.9732	0.9511	1.0042	1.1368	1.0944	1.1108	<b>1.0408</b>
10M(10690520)	1.0842	1.0568	1.0827	0.9702	1.0442	1.0642	1.0168	1.0302	1.0383	1.1328	<b>1.0521</b>
15M(15765158)	1.0162	0.9648	1.0202	1.0684	1.0583	1.0242	1.0548	1.0144	1.0642	0.9808	<b>1.0266</b>
20M(20998049)	0.9989	1.0862	1.1348	1.0648	1.0204	1.0612	0.9984	1.0664	0.9846	1.0168	<b>1.0433</b>
File (bytes)	Delay (µsecond)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.8883	0.9178	1.0071	0.9816	1.0283	1.0515	0.9963	0.8792	0.9143	0.9001	<b>0.9565</b>
10M(10690520)	0.9227	0.9459	0.9235	1.0313	0.9574	0.9402	0.9829	0.9709	0.9631	0.8823	<b>0.952</b>
15M(15765158)	0.9701	1.0336	0.9799	0.9365	0.9449	0.9763	0.9481	0.9861	0.9403	1.0197	<b>0.9736</b>
20M(20998049)	1.0014	0.9026	0.8811	0.9388	0.9803	0.9428	1.0021	0.9381	1.0151	0.9535	<b>0.9556</b>

**LAMPIRAN 10 : Data Hasil Uji Coba Konfigurasi IPv4 Murni Dengan Media Wireless**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	20.012	20.274	19.431	18.109	19.312	20.054	19.756	18.243	19.661	19.335	<b>19.4187</b>
10M(10690520)	35.591	33.821	36.045	36.117	34.554	35.919	36.371	35.802	36.746	35.28	<b>35.6246</b>
15M(15765158)	52.302	52.499	54.067	51.641	52.883	51.459	54.349	52.208	50.075	50.241	<b>52.1724</b>
20M(20998049)	71.054	68.939	72.954	66.584	72.847	65.728	69.751	70.956	67.615	69.428	<b>69.5856</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	2.4042	2.3728	2.4763	2.6564	2.4912	2.3987	2.4363	2.6368	2.4469	2.4963	<b>2.4816</b>
10M(10690520)	2.5412	2.6688	2.5049	2.5001	2.6127	2.5148	2.4884	2.5229	2.4567	2.5603	<b>2.5371</b>
15M(15765158)	2.5463	2.5368	2.4629	2.5781	2.5181	2.5882	2.4509	2.5501	2.6588	2.6505	<b>2.5541</b>
20M(20998049)	2.4961	2.5749	2.4329	2.6644	2.4362	2.6989	2.5425	2.4987	2.6227	2.5551	<b>2.5522</b>
File (bytes)	Delay (μsecond)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.4159	0.4214	0.4039	0.3764	0.4014	0.4168	0.4105	0.3791	0.4086	0.4006	<b>0.4035</b>
10M(10690520)	0.3937	0.3746	0.3992	0.4005	0.3827	0.3976	0.4021	0.3946	0.4071	0.3907	<b>0.3942</b>
15M(15765158)	0.3927	0.3941	0.4059	0.3878	0.3871	0.3864	0.4081	0.3921	0.3761	0.3772	<b>0.3901</b>
20M(20998049)	0.4005	0.3884	0.4111	0.3753	0.4104	0.3705	0.3932	0.4001	0.3813	0.3914	<b>0.3922</b>

**LAMPIRAN 11 : Data Hasil Uji Coba Konfigurasi IPv6 Murni Dengan Media Wireless**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	25.616	24.991	25.911	26.433	24.604	25.402	25.924	26.012	25.264	25.643	<b>25.5801</b>
10M(10690520)	48.391	48.292	49.425	50.148	49.522	47.786	47.789	51.101	48.513	48.158	<b>48.9125</b>
15M(15765158)	74.737	74.145	73.198	69.941	72.847	78.932	77.866	78.189	73.862	73.421	<b>74.7138</b>
20M(20998049)	96.208	95.466	91.878	96.257	97.163	90.582	91.313	93.372	96.992	92.828	<b>94.2059</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	1.9302	1.9804	1.9081	1.8722	2.0102	1.9461	1.9088	1.9013	1.9585	1.9303	<b>1.9346</b>
10M(10690520)	1.9202	1.9214	1.8783	1.8529	1.8769	1.9421	1.9412	1.8149	1.9129	1.9282	<b>1.8989</b>
15M(15765158)	1.8329	1.8464	1.8701	1.9582	1.8803	1.7343	1.7589	1.7523	1.8528	1.8629	<b>1.8349</b>
20M(20998049)	1.8948	1.9102	1.9843	1.8948	1.8763	2.0129	1.9962	1.9523	1.8808	1.9642	<b>1.9367</b>
File (bytes)	Delay ( $\mu$ second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.5181	0.5051	0.5241	0.5341	0.4975	0.5138	0.5238	0.5261	0.5104	0.5181	<b>0.5171</b>
10M(10690520)	0.5207	0.5205	0.5235	0.5396	0.5327	0.5151	0.5151	0.5509	0.5226	0.5187	<b>0.5259</b>
15M(15765158)	0.5456	0.5416	0.5348	0.5108	0.5321	0.5767	0.5685	0.5709	0.5398	0.5366	<b>0.5457</b>
20M(20998049)	0.5277	0.5236	0.5039	0.5275	0.5331	0.4967	0.5009	0.5124	0.5396	0.5092	<b>0.5175</b>

**LAMPIRAN 12 : Data Hasil Uji Coba Konfigurasi Tunneling 6to4 Dengan Media Wireless**

File (bytes)	Transfer Time (second)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	35.462	33.935	37.726	34.922	36.229	35.141	34.626	35.752	37.412	35.239	<b>35.6444</b>
10M(10690520)	63.632	61.072	64.837	63.426	62.635	64.426	61.257	62.434	63.839	64.149	<b>63.1707</b>
15M(15765158)	91.502	95.044	92.484	93.152	91.894	94.139	94.856	92.894	93.762	92.387	<b>93.2114</b>
20M(20998049)	133.747	135.543	133.883	134.142	134.984	135.114	134.436	135.342	134.778	134.177	<b>134.6146</b>
File (bytes)	Throughput (Mbps)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	1.3962	1.4588	1.3122	1.4183	1.3664	1.4129	1.4309	1.3882	1.3247	1.4049	<b>1.3914</b>
10M(10690520)	1.4702	1.5228	1.4327	1.4658	1.4838	1.4435	1.5179	1.4884	1.4561	1.4492	<b>1.4731</b>
15M(15765158)	1.5001	1.4412	1.4819	1.4708	1.4932	1.4572	1.4452	1.4749	1.4639	1.4836	<b>1.4712</b>
20M(20998049)	1.3648	1.3482	1.3638	1.3624	1.3536	1.3516	1.3591	1.3496	1.3551	1.3625	<b>1.3571</b>
File (bytes)	Delay (µsecond)										Rata-rata
	1	2	3	4	5	6	7	8	9	10	
5M(5696602)	0.7163	0.6854	0.7619	0.7053	0.7318	0.7077	0.6988	0.7206	0.7548	0.7118	<b>0.7194</b>
10M(10690520)	0.6843	0.6567	0.6978	0.6822	0.6739	0.6927	0.6587	0.6718	0.6867	0.6899	<b>0.6795</b>
15M(15765158)	0.6667	0.6937	0.6748	0.6799	0.6697	0.6862	0.6919	0.6779	0.6831	0.6741	<b>0.6798</b>
20M(20998049)	0.7325	0.7417	0.7332	0.7339	0.7387	0.7398	0.7357	0.7409	0.7379	0.7344	<b>0.7369</b>