ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE141599

ANALISA UNJUK KERJA PADA METODE *TUNNELING MANUAL*, *6TO4*, DAN *ISATAP* PADA IPV4/IPV6

Wahyu Narendra Jati
NRP 2210100012

Dosen Pembimbing
Dr. Ir. Achmad Affandi, DEA
Ir. Djoko Suprajitno Rahardjo, MT

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015

Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 141599

Performance Analysis of Manual, 6to4, and ISATAP Tunneling Methode On IPv4/IPV6 Networks

Wahyu Narendra Jati
NRP 2210 100 012

Lecture Advisor
Dr. Ir. Achmad Affandi, DEA
Ir. Djoko Suprajitno Rahardjo, MT

ELECTRICAL ENGINEERING DEPARTEMENT
Industry Technology Faculty
Sepuluh Nopember Technology Institute
Surabaya 2014

ANALISA UNJUK KERJA PADA METODE TUNNELING MANUAL, 6TO4, DAN ISATAP PADA IPV4/IPV6

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Telekomunikasi Multimedia
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I,	Dosen Pembimbing II,
Dr. Ir. Achmad Affandi, DEA	**Ir. Djoko Suprajitno Rahardjo**
NIP. 196510141990021001	**NIP. 195506221987011001**

SURABAYA
JANUARI, 2015

ANALISA UNJUK KERJA PADA METODE TUNNELING MANUAL, 6TO4, DAN ISATAP PADA JARINGAN IPv4/IPv6

Wahyu Narendra Jati
2210 100 012

Dosen Pembimbing I : Dr. Ir. Achmad Affandi, DEA
Dosen Pembimbing II : Ir. Djoko Suprajitno Rahardjo, MT.

Abstrak :

Tunneling merupakan salah satu teknik yang digunakan pada saat transisi dari IPv4 ke IPv6. Metode *tunneling* yang digunakan merupakan hal yang perlu diperhatikan untuk membuat jaringan yang optimal.

Tujuan dari tugas akhir ini adalah untuk membandingkan unjuk kerja beberapa teknik *tunneling* dengan jaringan IPv6 tanpa *tunnel* sehingga dapat diketahui perubahan yang terjadi ketika menggunakan teknik-teknik tersebut. Hal ini dilakukan menggunakan testbed berupa tiga buah router dan dua buah end *host* dengan menyimulasikan jaringan *tunnel* IPv6.

Dari hasil pengambilan data diperoleh bahwa *tunnel* manual mengalami penurunan bandwidth terhadap jaringan native-IPv6 sebesar 52,963%, *tunnel* ISATAP mengalami penurunan sebesar 56,281%, dan *tunnel* 6to4 mengalami penurunan sebesar 56,429%. Penurunan bandwidth tersebut tercerminkan dalam parameter loss dan RTT. Parameter jitter dari ketiga *tunnel* mengalami peningkatan 294,023%, 394,988%, dan 419,678% dibandingkan jaringan native-IPv6, namun jitter dari ketiga jaringan tunnel tersebut masih dibawah 1ms.

Kata kunci : IPv6, *Manual Tunneling, 6to4, ISATAP, Bandwidth, Packet Loss, Jitter, RTT*

Halaman ini sengaja dikosongkan

PERFORMANCE ANALYSIS OF MANUAL, 6TO4, AND ISATAP TUNNELING METHOD ON IPV4/IPV6 NETWORKS

Wahyu Narendra Jati
2210 100 012

Lecture Advisor I : Dr. Ir. Achmad Affandi, DEA
Lecture Advisor II : Ir. Djoko Suprajitno Rahardjo, MT.

Abstract :

Tunneling is one of the techniques used at the time of transition from IPv4 to IPv6. The tunneling method used is to be consider in order to make an optimal network.

The purpose of this thesis is to compare the performance of some of the tunneling techniques with a native IPv6 network (without a tunnel) in order to see the changes that occur when using these techniques. This is done with using a testbed of three routers and two end hosts by simulating an IPv6 tunnel network.

From the data results, it is obtained that the manual tunnel's bandwidth compared to the native-IPv6 network decreases by 52,963%, the ISATAP tunnel decreases by 56,281%, and the 6to4 tunnel decreases by 56,429%. This drop in bandwidth is also reflected on the packet loss and RTT parameter. The jitter on all three tunnels increase by 294,023%, 394,988%, and 419,678% compared to the native-IPv6, however the jitters are still bellow 1ms.

Key Word : *IPv6, Manual Tunneling, 6to4, ISATAP, Bandwidth, Packet Loss, Jitter, RTT*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Alhamdullilah, puji syukur penulis panjatkan kehadirat Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan buku Tugas Akhir dengan judul :

"ANALISA UNJUK KERJA PADA METODE TUNNELING MANUAL, 6TO4, DAN ISATAP PADA IPV4/IPV6 "

Tugas akhir merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program studi Strata-1 pada Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa dalam penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, dan kerjasama dari berbagai pihak sehinggala kendala-kendala tersebut dapat diatasi. Untuk itu pada kesempatan ini penulis menyampaikan banyak terimakasih dan penghargaan setingi-tingginya kepada :

1. Kedua Orang tua
2. Kedua Dosen Pembimbing
3. Bapak Dr. Istas Pratomo, ST., MT.
4. Teman-teman yang telah membantu penyelesaian tugas akhir ini

Penulis menyadari bahwa pada penyusunan laporan tugas akhir ini masih terdapat kekurangan-kekurangan karena keterbatasan kemampuan yang penulis miliki, walaupun demikian penulis berharap tugas akhir ini dapat bermanfaat bagi yang membutuhkannya.

Surabaya, Januari 2015

Penulis

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

DAFTAR ISI

DAFTAR TABEL

BAB 1
PENDAHULUAN

1.1 Latar Belakang

IPv4 merupakan metode pengalamatan dalam jaringan dan internet yang dikembangkan pada awal 70-an. Perkembangan jaringan yang sangat pesat dalam teknologi jaringan menyebabkan kebutuhan akan alamat IP membesar dan pada akhirnya tidak dapat lagi dibendung oleh IPv4 sehingga dikembangkan protokol baru untuk mengingkatkan ruang internet yaitu IPv6. Berbeda dengan IPv4 yang terdiri dari 32 bit, IPv6 terdiri dari 128 bit sehingga secara teori dapat menampung 2^{96} kali jumlah alamat IPv4.

Pengimplementasian IPv6 tidak bisa serentak dan membutuhkan waktu yang lama sehingga terdapat suatu masa transisi dimana IPv4 dan IPv6 berjalan bersamaan. Pada masa transisi ini diperlukan teknik-teknik yang dapat diimplementasikan oleh IPv6 untuk dapat kompatibel dengan IPv4, teknik-teknik ini disebut dengan mekanisme transisi. Terdapat tiga macam mekanisme transisi yaitu *Dual Protocol Stack, Tunneling,* dan *Protocol Translation*.

Tugas akhir ini disusun untuk mengetahui kinerja dari beberapa metode transisi *tunneling* IPv4/IPv6 yaitu secara manual, 6to4, dan ISATAP.

1.2 Perumusan Masalah

Permasalahan yang dihadapi adalah IPv4 sudah tidak mampu menampung jumlah pengguna internet, sehingga harus beralih ke IPv6. Karena besarnya investasi pada IPv4 maka metode *tunneling* merupakan metode yang menjanjikan, sehingga pemilihan metode *tunneling* merupakan hal yang perlu diperhatikan sehingga diperoleh jaringan yang optimal. Dengan permasalahan tersebut dibuatlah tugas akhir ini.

1.3 Batasan Masalah

Agar penelitian tidak menyimpang dari permasalahan maka penulis membatasi masalah sebagai berikut :

1. Implementasi interkoneksi menggunakan *tunneling* manual, 6to4, dan ISATAP
2. Pengukuran kinerja interkoneksi dengan parameter *bandwidth, end-to-end delay, jitter,* dan *packet loss.*

1.4 Tujuan

Tujuan dari tugas akhir ini adalah untuk :

1. Mengevaluasi kinerja interkoneksi antara jaringan IPv4 dan jaringan IPv6 atau sebaliknya dengan metode *tunneling* manual, 6to4, dan ISATAP
2. Membandingkan hasil evaluasi kinerja interkoneksi tersebut dengan kinerja koneksi jaringan *native-IPv6*.

1.5 Metodologi

Metode Penelitian yang digunakan pada tugas akhir ini terbagi menjadi tujuh tahap sebagai berikut:

1. Studi literatur

Studi literatur dilakukan dengan mencari dan mempelajari beberapa paper dan jurnal. Pada tahap ini akan dipelajari cara kerja serta konfigurasi ketiga macam *tunneling* tersebut.

2. Analisa kebutuhan sistem

Pada tahap ini dilakukan analisa kebutuhan sistem. Tahap ini bertujuan untuk memperoleh *software* dan *hardware* serta spesifikasi yang diperlukan untuk mendukung pengujian, implementasi, dan pengambilan data dari sistem.

3. Rancangan sistem

Pada tahap ini ditentukan parameter apa saja yang akan diuji dan dilakukan simulasi untuk perangkat keras yang akan digunakan sebelum diimplementasikan agar perangkat keras dapat mendukung skenario kerja yang telah ditentukan.

4. Konfigurasi sistem

Pada tahap ini dilakukan konfigurasi terhadap perangkat keras yang digunakan sesuai dengan sistem yang telah dirancang sebelumnya.

5. Pengujian

Pada tahap ini dilakukan pengujian terhadap kinerja perangkat keras yang telah terintegrasi sesuai dengan parameter uji yang telah ditentukan untuk mengamil data.

6. Analisa data

Pada tahap ini dilakukan pengamatan dan analisa terhadap data yang telah diperoleh pada tahap pengujian.

7. Penarikan kesimpulan

Penarikan kesimpulan dilakukan berdasarkan analisa data yang telah dilakukan.

1.6 Sistematika Pembahasan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut.

BAB I PENDAHULUAN

Pada bab ini akan diuraikan mengenai latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan tentang tinjauan pustaka yang akan membahas tentang OSI model, protokol TCP/IP, internet protocol, teknik transisi IPv6, *tunnel* manual, *tunnel* otomatis, dan parameter unjuk kerja

BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan tentang pengimplementasian sistem berdasarkan teori pada Bab II serta melakukan pengujian unjuk kerja sistem.

BAB IV HASIL DAN ANALISA DATA

Pada bab ini akan ditampilkan hasil pengujian, kemudian dilakukan analisa dari data yang telah diperoleh sehingga dapat memudahkan melakukan penarikan kesimpulan.

BAB V PENUTUP

Pada bab ini berisi tentang kesimpulan, dan saran berdasarkan yang telah dilakukan dalam pengerjaan tugas akhir ini.

1.7 Relevansi

Hasil yang didapat dari tugas akhir ini diharapkan dapat memberi manfaat sebagai berikut :

1. Dapat memberikan kontribusi berupa informasi mengenai perbandingan unjuk kerja *tunnel* manual, ISATAP, dan 6to4 pada jaringan IPv4/IPv6 terhadap jaringan *native-IPv6*.
2. Menjadi referensi dalam pengimplementasian *tunnel* pada jaringan IPv4/IPv6.

Halaman ini sengaja dikosongkan

BAB II
TEORI DASAR

2.1 Jaringan

Jaringan komputer dapat diartikan sebagai sekumpulan komputer maupun perangkat lain (printer, scanner, hub, dll) yang saling terhubung satu sama lain melalui media perantara. Media perantara tersebut dapat berupa kabel maupun nirkabel (wireless). Jaringan komputer berfungsi sebagai salah satu bentuk komunikasi antar komputer.

2.1.1 Jenis-Jenis Jaringan

Jaringan terdefinisi menjadi 3 jenis yaitu jaringan berdasarkan fungsi, jaringan berdasarkan media transmisi, dan jaringan berdasarkan area.

2.1.1.2 Berdasarkan Media Transmisi

Jaringan berdasarkan media transmisi dibagi menjadi 2, yaitu :

- *Wired Network (kabel)*, menggunakan media kabel sebagai penghantarnya. Kabel yang biasa digunakan adalah kabel UTP, Coaxial, ataupun Fiber Optik.
- *Wireless Network (nirkabel)*, menggunakan media gelombang radio, Infra Red, atau bluetooth sebagai media penghantarnya. Salah satu penerapan *Wireless Network* adalah area Hotspot.

2.1.1.2 Berdasarkan Area

Jaringan komputer berdasarkan area dibagi menjadi 4, yaitu :

- LAN (Local Area Network), merupakan jaringan komputer yang hanya mencakup wilayah kecil dan diatur oleh administrator yang sama, seperti jaringan komputer kampus dan gedung, kantor
- MAN (Metropolitan Area Network), Jaringan yang secara fisik lebih besar lari LAN namun lebih kecil dari WAN. Suatu MAN umumnya dimiliki dan dioperasikan oleh sebuah organisasi.
- WAN (Wide Area Network), merupakan kumpulan dari jaringan-jaringan LAN yang dapat mencakup daerah geografis yang sangat luas. Internet merupakan jaringan WAN terbesar.

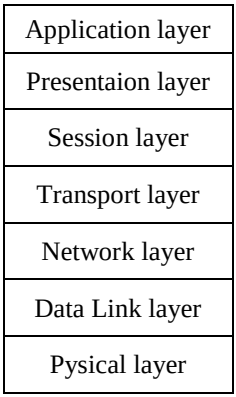
Sebelum munculnya model referensi jaringan, sistem jaringan komputer sangat tergantung kepada pemasok (*vendor*). Model referensi jaringan berupaya membentuk standar umum jaringan komputer untuk

menunjang interoperatibilitas antar pemasok yang berbeda. Model-model tersebut terdiri dari model OSI dan TCP/IP.

2.2 Model *Open System Interconnection* (*OSI model*)

Model OSI diluncurkan pada tahun 1984 dan digunakan sebagai model untuk semua protokol baru. Pada model OSI terdapat tujuh lapisan (*layer*) yang masing-masing mendefinisikan fungsi tertentu dalam jaringan komunikasi. Tiap lapisan berkomunikasi dengan lapisan yang sama pada perangkat yang berbeda dalam jaringan (komunikasi peer-to-peer). Beberapa perangkat hanya beropersai pada lapisan tertentu (contohnya router *layer* 3, dan ethernet switch *layer* 2). Perangkat-perangkat tersebut buta terhadap lapisan yang berada diatasnya.

Model OSI memiliki tujuh lapis, yaitu : *physical layer, data link layer, network layer, transport layer, session layer, presentation layer,* dan *application layer* dengan struktur sebagai berikut:



Gambar 2.1 Lapisan Model OSI **[6]**

2.2.1 *Physical Layer*

Lapisan yang pertama merupakan *physical layer*. Lapisan ini berfungsi untuk mendefinisikan media transmisi jaringan, mengkonversi unit data (termasuk *protocol header*) menjadi sinyal listrik atau optik untuk ditransmisikan melalui jaringan, dan bagaimana *Network Interface Card* (NIC) dapat berinteraksi dengan media kabel atau radio. Lapisan ini tidak menambahkan header kecuali pada sistem yang

bersifat *synchronous* contohnya pada *Synchronous Optical Network* (SONET). Unit data pada lapisan ini dinamakan bits.

2.2.2 *Data Link Layer*

Lapisan kedua merupakan *data link layer*. Unit data pada lapisan ini dinamakan *frame*. Lapisan *data link* memiliki dua *sublayer* yaitu *Logical Link Control* (LLC) dan *Media Access Control* (MAC).

2.2.2.1 LLC Sublayer

LLC merupakan sub lapisan teratas. LLC berfungsi melakukan multipleks terhadap protokol yang berjalan diatas *data link layer* dan dapat menyediakan *flow control* dan *automatic repeat request* (ARQ) [5].

2.2.2.2 MAC Sublayer

Sub lapisan MAC bertindak sebagai antarmuka antara LLC dengan *physical layer*. Menurut standar IEEE 802-2001 sesi 6.2.3, fungsi utama dari sub lapisan MAC adalah [5]:

- Menentukan batas frame
- Pengalamatan node tujuan
- Membawa informasi alamat node sumber
- Transfer data LLC *Protocol Data Unit* (PDU) yang transparan
- Membuat dan memeriksa urutan *frame*
- Mengatur akses ke media transmisi fisik.

2.2.3 *Network Layer*

Lapisan ketiga adalah *network layer*. Berfungsi untuk mendefinisikan alamat-alamat *Internet Protocol* (IP), membuat header untuk paket-paket data, dan kemudian melakukan *routing* dengan menggunakan *internetworking* melalui router atau switch layer-3. Unit data pada lapisan ini dinamakan paket.

2.2.4 *Transport Layer*

Transport Layer, lapisan keempat, berada dalam perangkat lunak *host* dan tidak berinteraksi dengan perangkat jaringan. Suatu *host* berkomunikasi dengan *host* tujuan dengan mengirimkan pesan protokol dan melakukan sesi negosiasi. Unit data lapisan ini bernama *segmen*. Transport layer berfungsi mengatur cara memecah rentetan data ke dalam paket-paket data serta memberikan nomor urut ke paket-paket

tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima. Selain itu, pada lapisan ini juga akan mengirimkan sebuah sinyal apabila paket telah diterima dengan sukses yang dikenal dengan istilah *acknowledgement* (ACK), dan *host* yang menerima request akan melakukan pengiriman ulang apabila paket data hilang selama proses transmisi. Contoh dari protokol pada *transport layer* adalah TCP dan UDP [1].

2.2.5 *Session Layer*

Lapisan kelima dinamakan *session layer*. lapisan ini bertanggung jawab untuk mempertahankan dialog dengan aplikasi *host* tujuan dalam sebuah *connection-oriented protocol* seperti pada TCP. Lapisan ini hanya berfungsi ketika menggunakan *connection-oriented protocol* [5].

2.2.6 *Presentation Layer*

Lapisan keenam merupakan *presentation layer*. Berfungsi untuk menerjemahkan data yang akan ditransmisikan oleh aplikasi ke dalam format yang dapat ditransmisikan melalui jaringan dan sebaliknya.

2.2.7 *Application Layer*

Lapisan ketujuh merupakan *application layer*. Berfungsi sebagai antarmuka antara pengguna dengan aplikasi yang membutuhkan akses ke jaringan (contohnya *email*), dan mengatur bagaimana aplikasi dapat mengakses jaringan. Protokol yang berada dalam lapisan ini antara lain HTTP, FTP, SMTP, dan NFS.

Dapat disimpulkan bahwa tiga lapisan pertamalah yang digunakan untuk komunikasi jaringan. Fungsi dari lapisan-lapisan ini dapat ditemukan dalam beberapa perangkat seperti router, switch, kabel UTP, dll. Lapisan keempat memberikan komunikasi yang handal, terlepas dari apa yang disediakan oleh lapisan dibawahnya. Misalnya ketika menggunakan pelayanan *connection-oriented*.

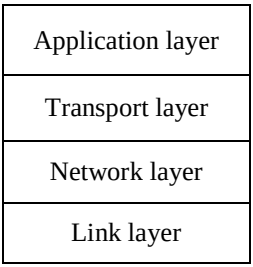
Sedangkan layer lima sampai tujuh berfungsi dalam menentukan bagaimana data akan ditampilkan dan mengatur dialog antar *host*.

2.3 Protokol TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) menyediakan konektivitas *end-to-end* dengan mespesifikasikan bagaimana data dipaketkan, diberikan alamat, ditransmisikan, dirutekan, dan diterima sampai pada tujuan. TCP/IP merupakan kumpulan dari

beberapa protokol, dimana tiap protokol memiliki fungsi tertentu dalam sebuah jaringan. Protokol-protokol tersebut menjalankan fungsinya sebagaimana didefinisikan dalam RFC1122 dan RFC 1123 [4].

Fungsi dari TCP/IP dipecah menjadi empat lapisan, yaitu: *application layer*, *transport layer*, *network layer*, dan *link layer*, dengan struktur sebagai berikut:



Gambar 2.2 Lapisan Model TCP/IP **[6]**

2.3.1 *Application Layer*

Application layer terdiri dari protokol komunikasi dan metode antarmuka yang digunakan dalam komunikasi *process-to-process* melalui jaringan komputer. Contoh dari protokol pada lapisan ini antara lain Telnet, SSH, FTP, dan TFTP. Lapisan aplikasi dengan transport dipisahkan oleh nomor *port* dan *socket* [4].

2.3.2 *Transport Layer*

Data kemudian dikirimkan ke *transport layer*, *transport layer* menyediakan jasa komunikasi dari suatu titik ke titik lainnya dengan fitur sebagai berikut:

- Komunikasi *connection-oriented* atau *connectionless*. *Connection-oriented* berarti terjadi "jabat tangan" antara pengirim dengan penerima sehingga menjamin data yang ditransfer sampai pada tujuan, sedangkan connectionless merupakan kebalikannya dimana tidak terjadi "jabat tangan" sehingga tidak menjamin data sampai pada tujuan.
- Pemberian urutan pengiriman, *network layer* tidak menjamin bahwa paket data yang dikirimkan akan sampai pada urutan yang sesuai. Hal ini dilakukan dengan pemberian nomor pada tiap segmen, sehingga transport layer pada penerima akan dapat

menyampaikan segmen tersebut ke *application layer* dengan urutan yang sesuai.

- *Reliability*, dengan menggunakan code pendeteksi error, seperti checksum, sebuah protokol transpor dapat memerika apakah suatu segmen rusak atau tidak dan memberitahu bahwa data telah sampai dengan mengirimkan pesan ACK atau NACK kepada pengirim.
- *Flow control*, kecepatan transmisi data antara dua node terkadang harus diatur untuk menghindari kecepatan transmisi yang melebihi daya tampung penerima.

Ketika data dikirimkan dari sebuah aplikasi ke *transport layer*, terdapat kemungkinan data tersebut terlalu besar untuk dimuatkan menjadi satu unit data. TCP menyediakan fungsi yang disebut *fragmantation* dan *reassembly* untuk mengatasi masalah ini. Data tersebut dibagi menjadi beberapa unit data dengan ukuran yang sama untuk kemudian diteruskan ke network layer [4].

2.3.3 *Network Layer*

Network layer, disebut juga lapisan *internetwork* atau *internet*, merupakan jenis protokol *connectionless*. *Internet Protocol* (IP) merupakan protokol utama dalam lapisan ini. IP menyediakan fungsi penentuan jalur dalam transmisi data. Protokol-protokol lain yang terdapat pada lapisan ini adalah ICMP, IGMP, ARP, dan RARP. Fungsi dari network layer adalah:

- Pengalamatan *host*, tiap *host* dalam Internet harus memiliki alamat unik yang menentukan lokasi *host* tersebut. Alamat ini berupa alamat IP.
- Penerusan pesan, karena banyak jaringan yang terpartisi menjadi sub jaringan dan terhubung dengan jaringan lainnya, router dan gateway digunakan untuk meneruskan paket antar jaringan tersebut.

Pada *network layer*, unit data individual diatas mungkin masih memerlukan fragmentasi lebih lanjut. Fragmentasi ini tidak akan mempengaruhi apa yang telah dilakukan pada *transport layer* karena merupakan hubungan *peer-to-peer* sesama lapisan [4].

2.3.4 *Link Layer*

Lapisan terbawah dari TCP/IP adalah *link layer*, disebut juga lapisan *network interface* atau *data link*, merupakan antarmuka dengan peranti keras. TCP/IP tidak memberikan spesifikasi protokol lapisan

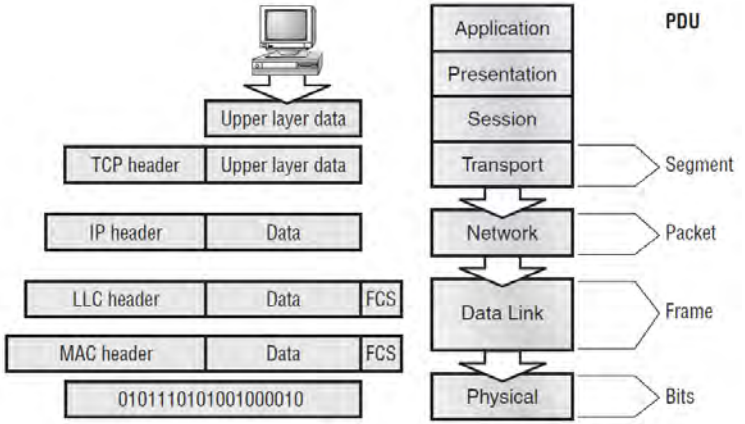
data link, melainkan patokan cara mengakses protokol tingkatan yang lebih tinggi melalui lapisan *link*.

2.4 Enkapsulasi

Ketika sebuah host mengirimkan data melalui jaringan ke perangkat lain, data akan melalui proses yang disebut enkapsulasi dan "dibungkus" dengan informasi protokol pada tiap lapisan model OSI. Setiap lapisan hanya berkomunikasi dengan lapisan rekannya pada perangkat penerima.

Untuk berkomunikasi dan bertukar informasi, setiap lapisan menggunakan *Protocol Data Unit* (PDU). PDU mengandung lampiran informasi kontrol untuk data pada tiap lapisan model.

Tiap PDU menempel ke data dengan cara mengenkapsulasikannya pada tiap lapisan dari model OSI, dan tiap PDU memiliki nama yang berbeda berdasarkan informasi yang disediakan dalam tiap header. Informasi PDU hanya dapat dibaca oleh lapisan rekannya pada perangkat penerima.



Gambar 2.3 Enkapsulasi Data

Gambar 2.3 menunjukkan PDU pada tiap lapisan. *Data-stream* dari *upper-layer data* diturunkan ke *Transport layer*. Kemudian data-stream tersebut dipecah menjadi bagian-bagian yang lebih kecil dan diberikan header *Transport layer* (sebuah PDU) untuk tiap data field, bagian ini dikenal dengan sebutan "*segment*". Tiap segmen diberikan urutan

sehingga nantinya dapat disatukan kembali pada sisi penerima sesuai dengan urutannya ketika ditransmisikan.

Tiap segmen kemudian diturunkan ke *Network layer* untuk pengalamatan jaringan dan routing. Pengalamatan logic (misalnya IP) digunakan agar tiap segmen dapat mencapai jaringan yang benar. Protokol pada *Network layer* menambahkan header kontrol pada segmen, PDU pada lapisan ini dikenal dengan "*packet*" atau "*datagram*".

Data Link layer melakukan enkapsulasi untuk tiap paket sehingga menjadi "*frame*". Header pada *frame* membawa informasi mengenai alamat *hardware* dari *source* dan *destination host*. Jika perangkat tujuan terletak pada jaringan yang lain maka *frame* tersebut dikirimkan ke router untuk dirutekan melalui internetwork. Sesampainya di jaringan tujuan, *frame* yang baru digunakan agar paket dapat sampai ke *host* tujuan.

Agar *frame* dapat diteruskan dalam jaringan, *frame* tersebut harus direpresentasikan dalam bentuk sinyal digital. *Physical layer* bertanggung jawab untuk melakukan encoding digit-digit *frame* menjadi sinyal digital yang kemudian dibaca oleh perangkat pada jaringan lokal. Perangkat penerima akan melakukan sinkronisasi dan decoding terhadap sinyal digital yang diterima. Pada tahap ini perangkat penerima akan membangun *frame*, melakukan *Cyclic Redundency Check* (CRC), dan kemudian membandingkan hasil yang diperoleh dengan jawaban pada bagian *Frame Check Sequence* (FCS) *frame*. Jika cocok maka paket ditarik dari *frame* dan sisa dari *frame* dibuang. Proses ini disebut de-encapsulasi. Paket diberikan ke *Network layer*, dimana alamatnya dicek. Jika alamatnya sesuai, maka segmen ditarik dari paket dan sisa dari paket dibuang. Segmen tersebut diproses pada *Transport layer*, yang membangun ulang *data stream* dan kemudian meneruskan *data-stream* tersebut ke aplikasi *upper-layer* [14].

2.5 *Internet Protocol* (IP)

IP terdapat pada lapisan ke-3 (*network layer*) dari model OSI. IP bersifat *connectionless* sehingga tidak menjamin keandalan transmisi, namun hal ini tidak menjadi masalah jika lapisan atasnya menyediakan jasa yang mendukung keandalah transmisi seperti deteksi dan koreksi error.

Sebuah *IP-host* harus mengenkapsulsi data ke dalam IP header yang kemudian diteruskan ke lapisan *data link*. Protokol yang digunakan

dalam lapisan *data link* kemudian mengenkapsulasi data dan IP header tersebut dengan data unitnya sendiri sehingga menjadi frame data. Frame tersebut kemudian disampaikan ke *physiscal layer* untuk diteruskan ke jaringan.

Data yang memiliki tujuan diluar jaringan lokal harus dikirimkan ke router atau perangkat *layer-3* lainnya. Router adalah perangkat yang bekerja pada lapisan ke-3 dan memilki fungsi untuk mengolah IP header. Jika data hendak dikirim ke jaringan lain maka router akan melepaskan data link header dan mengolah IP header dari data tersebut.

IP mempunyai dua fungsi utama, yaitu [6]:

- Pengalamatan secara logis, menyediakan alamat unik yang membedakan pengguna satu dengan lainnya dan jaringan tempat pengguna tersebut berada.
- Penentuan rute, menentukan jalur terbaik untuk jaringan tujuan tertentu.

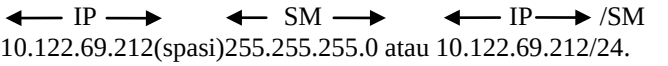
IPv4 memiliki keterbatasan pada jumlah alamat yang dapat didukung. Masalah ini merupakan latar belakang dikembangkannya jenis IP baru yang dinamakan *Internet Protocol next generation* (IPng) atau biasa disebut dengan IPv6. Saat ini terdapat dua macam protokol pengalamatan IP yaitu IPv4 dan IPv6, kedua versi ini bekerja secara terpisah sehingga paket dengan header IPv4 tidak dapat diteruskan melalui jaringan IPv6 secara langsung dan harus melalui proses tambahan.

2.5.1 *Internet Protocol Version* 4 (IPv4)

IPv4 atau biasa disebut dengan IP saja, mula-mula didefinisikan oleh RFC 760 dan telah direvisi beberapa kali. IPv4 merupakan versi pertama yang telah digunakan secara luas, IPv4 didefinisikan dalam RFC 791. IPv4 memberikan sebuah alamat IP khusus sebesar 32-bit kepada tiap pengguna internet. Untuk mempermudah penulisannya, 32-bit tersebut dibagi menjadi 4 segmen yang masing-masing terdiri dati 8-bit dan direpresentasikan dalam bentuk desimal, contohnya 10.122.69.212 [6].

Suatu alamat IPv4 terdiri dari dua bagian. Bagian pertama (*network-ID*) untuk mengidentifikasi alamat jaringan, diiktui oleh bagian kedua (*host-ID*) yang mengidentifikasi alamat tiap pengguna dalam jaringan tersebut. Dalam pelayanan internet, alamat jaringan diberikan oleh *Internet Service Provider* (ISP) yang berkoordinasi dengan *Internet Assigned Number Authority* (IANA) [6].

Panjangnya kedua bagian ini (*network-ID* dan *host-ID*) ditentukan oleh ukuran subnetmask yang digunakan. Subnetmask memiliki format yang sama dengan alamat IPv4 dan berfungsi menandakan berapa jumlah bit yang digunakan sebagai *network-ID*. Bit yang digunakan sebagai *network-ID* ditandai dengan nilai "1". Contoh, asumsikan sebuah komputer memiliki alamat IP 10.122.69.212 dengan niali subnetmask 11111111.11111111.11111111.00000000, berarti IP tersebut memiliki *network-ID* 10.122.69.0, subnetmask tersebut biasanya ditulis dalam format desimal sehingga menjadi 255.255.255.0 atau dapat juga dituliskan menggunakan tanda "/" diikuti oleh jumlah bit yang digunakan sebagai *network-ID*. Sehingga IP diatas beserta subnetmasknya dituliskan seperti Gambar 2.4.



Gambar 2.4 Format Pengalamatan IPv4

Keterangan :
IP : alamat IPv4
SM : ukuran subnetmask

2.5.1.1 Kelas-kelas Pengalamatan IPv4

Terdapat lima macam kelas pengalamatan pada IPv4. Struktur dari kelas-kelas ini dapat dilihat pada Gambar 2.5. Pengalamatan kelas A biasanya digunakan dalam jaringan "tertutup" pribadi. Yang dimaksud dengan jaringan yang tertutup adalah jaringan yang tidak terhubung dengan koneksi luar. Pengalamatan kelas B juga umumnya digunakan untuk jaringan yang bersifat tertutup. Kelas C merupakan jenis pengalamatan yang paling sering digunakan dan digunakan untuk komunikasi dengan jaringan luar. Kelas D digunakan untuk multicast addressing sedangkan kelas E direservasi untuk keperluan dimasa mendatang.

	Prefix	network ID	*host* ID
Kelas A	0	7-bit	24-bit
Kelas B	10	14-bit	16-bit
Kelas C	110	21-bit	8-bit
Kelas D	1110	Alamat Multicast	
Kelas E	1111		

Gambar 2.5 Kelas-kelas IPv4

Seperti yang ditunjukkan dalam Gambar 2.5, pengalamatan kelas A dapat menampung 126 alamat jaringan dan 16.777.124 pengguna. Kelas ini tidak baik digunakan pada internet yang besar karena jumlah jaringan melebihi jumlah pengguna, hal ini disebabkan karena sebagian besar node dalam internet tersebut terdiri dari router dan gateway.

Pengalamatan kelas B dapat menampung 16.384 jaringan dan 65.534 pengguna. Pengalamatan jenis ini dirancang untuk ukuran jaringan skala menengah dan tidak cocok untuk ukuran internet skala besar.

Pengalamatan kelas C dapat mendukung sebanyak 2.097.152 jaringan dan 254 pengguna. Pengalamatan kelas ini cocok digunakan untuk koneksi ke Internet karena mendukung jumlah alamat jaringan yang banyak.

Kelas D merupakan alamat yang digunakan untuk melakukan *multicast*. Digunakan untuk koneksi dengan grup yang terdiri dari beberapa pengguna individu dalam jaringan. Tiap grup diberikan alamat multicast dan dapat diakses dengan menggunakan pengalamatan kelas D. Contohnya protokol *routing* OSPF menggunakan alamat *multicast* 224.0.0.5 dan 224.0.0.6 untuk *multicast* paket HELLO dan update informasi *routing*.

2.5.1.2 Jenis-jenis Alamat IPv4

Terdapat dua jenis alamat pada IPv4 yaitu *public address* dan *private addess*. IP *Public* adalah alamat IP yang telah ditetapkan oleh InterNIC dan berisi beberapa buah alamat IP yang dijamin unik yang digunakan untuk lingkup internet, *host* yang menggunakan IP *public*

dapat mengakses seluruh user yang tergabung dalam Internet baik secara langsung maupun tidak langsung (melalui proxy/NAT). *IP Addressing* juga dikelompokkan berdasarkan negara, Indonesia sebagian besar dimulai dengan kepala 202 dan 203. Contoh dari IP Public adalah akses Speedy modem yang merupakan IP Public 125.126.0.1 [9].

Komputer yang tidak terhubung ke Internet seperti mesin-masin pabrik yang cukup berkomunikasi melalui TCP/IP tidak memerlukan alamat IP yang secara global unik. Jenis IP inilah yang disebut dengan alamat IP *private*. Namun jika dibutuhkan, maka IP ini dapat terhubung ke Internet dengan melakukan translasi alamat seperti *network address translation* (NAT). Alamat IP yang telah ditetapkan sebagai *private address* oleh IANA adalah:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
-

2.5.1.3 Struktur Paket

Sebuah paket IP terdiri bagian header dan *payload data*. Paket IP tidak memiliki checksum atau footer lainnya setelah bagian data. Umumnya paket IP akan dienkapsulasikan oleh *link layer* dengan CRC footer yang dapat mendeteksi sebagian besar error

Secara umum, *payload* adalah data yang dikirirmkan berdasarkan permintaan aplikasi. Ukuran sebuah *payload data* dapat bervariasi sampai pada panjang maksimum yang ditetapkan oleh protokol jaringan atau peralatan sepanjang rute. Beberapa jaringan dapat memecahkan paket besar menjadi paket-paket kecil bila diperlukan.

Header dari sebuah paket IPv4 terdiri dari 14 bagian dimana salah satunya bersifat opsional. Format header IPv4 ditunjukkan dalam Gambar 2.6.

Version	IHL	Type of Service	Total Length
Identification	Flags	Fragment Offset	
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options			Padding

Gambar 2.6 *Header* IPv4 [10]

Version, terdiri dari 4 bit dan berfungsi untuk mengindikasikan format dari internet header.

IHL (*Internet Header Length*) terdiri dari 4 bit dan merupakan panjang dari header internet dalam 32 bit words, sehingga menunjukkan kapan mulainya bagian data.

Type of Service terdiri dari 8 bit dan berfungsi menyediakan indikasi parameter abstrak dari *Quality of Service* yang diinginkan. Parameter ini akan digunakan sebagai panduan dari parameter pelayanan sebenarnya ketika mengirimkan datagram melalui suatu jaringan.

Beberapa jaringan menawarkan jasa *precedence*, dimana traffic yang memiliki nilai precedence yang lebih tinggi diperlakukan lebih penting (biasanya hal ini dilakukan dengan hanya melewatkan traffic dengan precedence diatas suatu nilai tertentu). Keterangan dari kedelapan bit ini dapat dilihat pada Tabel 2.1 berikut.

Tabel 2.1 Tabel *Precedence*

Bit ke-	Keterangan
0-2	Precedence
3	0 = Delay normal 1 = Delay rendah
4	0 = Throughput normal 1 = Throughput rendah
5	0 = Reliability normal 1 = Reliability rendah
6-7	Disimpan untuk penggunaan dimasa mendatang

Precedence :
111 = *Network Control*
110 = *Internetwork Control*
101 = CRITIC/ECP
100 = *Flash Override*
011 = *Flash*
010 = *Immediate*
001 = *Priority*
000 = *Routine*

Total Length terdiri dari 16 bit. Total length merupakan panjang dari datagram yang diukur dalam octet, termasuk internet header dan

data. Biasanya panjang datagram maksimal dibatasi sebesar 576 octet (namun masih bisa dipecah menjadi beberapa fragmen). Batasan nilai 576 oktet digunakan untuk memungkinkan pengiriman ukuran blok data yang cukup besar beserta dengan informasi header yang diperlukan.

Identification terdiri dari 16 bit dan merupakan suatu nilai yang diberikan oleh pengirim untuk membantu perakitan fragmen datagram.

Flags terdiri dari 3bit, terdapat beberapa macam *flag* antara lain:

- Bit 0 : cadangan, harus bernilai 0
- Bit 1 (DF) : 0 = boleh difragmentasi; 1 = tidak boleh difragmentasi
- Bit 2 (MF) : 0 = Fragmen terakhir; 1 = masih ada fragmen lebih banyak

Fragment Offset tersiri dari 13 bit dan berfungsi untuk menandakan dimana posisi fragmen tersebut berada dalam datagram.

Time to Live (TTL) terdiri dari 8 bit. Dalam proses transmisi, paket datagram melewati router dan gateway, setiap kali datagram melewati salah satu perangkat tersebut disebut sebagai satu kali hop.

TTL menandakan berapa banyak "hop" yang diperbolehkan pada datagram untuk tetap berada dalam jaringan. Jika TTL bernilai 0 maka datagram harus dibuang. TTL bertujuan untuk menghindari loop secara terus-menerus sehingga dapat menyebabkan kemacetan dalam jaringan.

Protocol terdiri dari 8 bit, bagian ini menandakan protokol yang digunakan pada bagian data. Terdapat beberapa nilai dari bagian ini, misalnya nilai 1 untuk ICMP, 6 untuk TCP, dan 17 untuk UDP.

Header Checksum terdiri dari 16 bit, checksum hanya untuk header. Bagian ini digunakan untuk melakukan pengecekan integritas terhadap header IP. Setiap router yang berada di dalam jalur transmisi antara sumber dan tujuan akan melakukan verifikasi terhadap bagian ini sebelum memproses paket. Jika verifikasi dianggap gagal, router akan mengabaikan datagram IP tersebut.

Karena setiap router yang berada di dalam jalur transmisi antara sumber dan tujuan akan mengurangi nilai TTP, maka header checksum pun akan berubah setiap kali datagram tersebut melewati suatu router.

Source Address, Berisi alamat IP pengirim datagram dan terdiri dari 32 bit. Sedangkan *Destination Address* Barisi alamat IP tujuan kemana datagram tersebut ingin disampaikan dan juga terdiri dari 32 bit.

Options, panjangnya bervariasi dan bersifat opsional, sehingga tidak harus disertakan dalam tiap datagram. Terdapat beberapa macam internet option antara lain :

- End of Option list
- No Operation
- Security
- Loose Source Routing
- Strict Source Routing
- Record Route
- Internet Timestamp

Padding panjangnya bervariasi. Padding pada internet header digunakan untuk memastikan panjang internet header merupakan kelipatan dari 32 bit. Padding bernilai nol [10].

2.5.2 *Internet Protocol Version* 6 (IPv6)

IPv6 merupakan versi IP yang dikembangkan oleh *Internet Engineering Task Force* (IETF) untuk mengatasi habisnya alamat pada IPv4. Meskipun IPv6 dan IPv4 secara umum memiliki fungsi yang sama, format pengalamatan, header, dan konfigurasi pada IPv6 berbeda dengan IPv4.

2.5.2.1 Representasi Alamat IPv6

Berbeda dengan IPv4 yang menggunakan 32 bit, IPv6 menggunakan 128-bit. Selain jumlah bitnya, dilakukan juga perubahan pada cara penulisan alamatnya. Untuk mempermudah penulisan alamat IP yang panjang tersebut, 128 bit alamat IPv6 dibagi menjadi 8 segmen yang masing-masing terdiri dari 16 bit biner, antar segmen dipisahkan dengan ":" dan dituliskan dalam bilangan hexadesimal. Contohnya adalah [4] :

2001:0d38:6abd:0000:0000:f585:002b

Untuk mengetahui batas *network-id* dan *host-id* maka setelah penulisan alamat IPv6 diikuti oleh tanda "/" dan besar *network-id* (subnetmask). Contoh sebagai berikut:

2001:0d38:6abd:0000:0000:f585:002b/**48**

Dimana 48 menunjukkan jumlah bit yang digunakan sebagai network-id yaitu :

2001:0d38:6abd

Sedangkan sisanya merupakan alamat *host*-id yaitu:

0000:0000:f585:002b

Penulisan alamat yang panjang tersebut dapat dipersingkat dengan mengikuti cara berikut:

- Apabila grup hexadecimal diawali oleh nol sebanyak satu atau lebih, maka nol tersebut boleh tidak ditulis. Sehingga alamat sebelumnya,

  2001:**0d38**:6abd:**0000**:**0000**:f585:**002b**

  menjadi

  2001:**d38**:6abd:**0**:**0**:f585:**2b**

- Apabila terdapat beberapa segmen yang secara berturut-turut hanya terdiri dari nol, maka segmen-segmen tersebut dapat digantikan dengan dua buah tanda titik dua (::). Sehingga alamat

  2001:d38:6abd:**0:0**:f585:2b

  dapat juga ditulis

  2001:d38:6abd**::**f585:2b/48

Menyingkat alamat IPv6 dengan menggunakan dua buah tanda titik dua (::) hanya boleh digunakan sekali untuk menghindari kedwiartian.

2.5.2.2 Jenis-jenis Alamat IPv6

Berdasarkan metode pengalamatan dan routingnya dalam jaringan, alamat IPv6 secara umum dibedakan menjadi 3 yaitu: pengalamatan *unicast*, *anycast*, dan *multicast*.

Alamat *unicast* merupakan alamat yang menunjuk pada satu antarmuka, digunakan untuk pengiriman data dari satu titik ke titik lain. Alamat *unicast* disusun oleh *subnet prefiks* dan *interface identifier* seperti pada Gambar 2.7.

Subnet Prefix (n bit)	Interface ID (128-n bit)

Gambar 2.7 Format Alamat *Unicast*

Terdapat beberapa macam alamat *unicast* yaitu : *link-local unicast*, *site-local unicast*, dan *global unicast*. Alamat *link-local* digunakan untuk komunikasi dengan pengguna pada jaringan yang sama sehingga router tidak meneruskan paket yang alamat asal atau tujuannya mengandung *prefix link-local* yaitu fe80::/10. Alamat *site-local* digunakan untuk komunikas dalam sebuah situs sehingga router tidak meneruskan paket yang memiliki alamat tujuan jika *mengandung prefix site-local yaitu fec0::/10*. Alamat global digunakan untuk komunikasi *unicast* pada jaringan manapun, prefix yang digunakan diawali dengan 001.

Alamat anycast merupakan alamat yang menunjuk pada beberapa antarmuka. Paket yang dikirimkan ke alamat *anycast* akan diteruskan ke antarmuka yang terdaftar dalam *anycast* tersebut dan memilki jarak terdekat dari router tergantung protokol routing yang digunakan.

Alamat *anycast* hanya dapat diberikan kepada router, dan paket tidak boleh berasal dari alamat *anycast*. *Anycast* memilki format yang sama dengan *unicast* sehingga tidak dapat dibedakan berdasarkan format alamat. Alamat *anycast* didefinisikan secara administratif.

Alamat *multicast* merupakan alamat yang menunjuk pada beberapa antarmuka. *Prefix* yang digunakan untuk *multicast* adalah ff::/8. Jika sebuah paket dikirimkan ke alamat *multicast* maka salinannya akan diteruskan ke semua antarmuka dalam grup tersebut. Format dari alamat multicast dapat dilihat pada Gambar 2.8.

Prefix ff (8 bit)	Flag (4 bit)	Scope (4 bit)	Group ID (122 bit)

Gambar 2.8 Format Alamat *Multicast*

Pada IPv6 tidak terdapat alamat *broadcast*. Pengalamatan *multicast* yang telah didefiniskan adalah sebagai berikut :

- *Multicast* ke semua antarmuka *host* (FF01::1)
- *Multicast* ke semua node dalam jaringan lokal (FF01::2)
- *Multicast* ke semua router dalam link lokal (FF02::2)
- *Multicast* ke semua router dalam situs lokal (FF05::2)

2.5.2.3 Struktur Paket

Paket IPv6 terdiri dari dua bagian yaitu : Paket Header dan Paket Payload. Ukuran paket header terdiri dari 40 oktet (320 bit) dengan format seperti pada Gambar 2.9.

Version	Traffic Class	Flow Label
Payload Length	Next Header	Hop Limit
Source Address		
Destination Address		

Gambar 2.9 *Header* IPv6 [11]

Version, bagian dari header ini terdiri dari 4 bit. *Version* menunjukkan versi protokol internet yang digunakan, untuk header IPv6 bagian ini selalu bernilai 6 (0110). Lokasi kolom ini sama untuk header IPv6 dan IPv4 sehingga memudahkan sebuah node untuk membedakan apakah merupakan paket IPv4 atau IPv6.

Traffic Class terdiri dari 8 bit dan digunakan untuk mengidentifikasi dan membedakan kelas atau prioritas suatu paket IPv6.

Flow Label, terdiri dari 20 bit dan digunakan untuk memberi label paket-paket tertentu yang membutuhkan penanganan khusus, misalnya pada jasa *non-default QoS* atau real-time.

Payload Length terdiri dari 16 bit. Berisi informasi mengenai panjang dari *payload* IPv6 atau sisa paket setelah header (dalam oktet). Nilai maksimum dari kolom ini adalah 65.535, jika kolom ini berisi nol, berarti paket berisi *payload* yang lebih besar dari 64Kbyte dan panjang *payload* yang sebenarnya ada di *Jumbo Payoad hop-by-hop option*.

Next Header terdiri dari 8 bit, berfungsi untuk mengidentifikasi tipe header selanjutnya.

Hop Limit terdiri dari 8 bit dan berfungsi menspesifikasikan jumlah hop maksimum yang dapat dilalui sebelum paket dibuang. Nilai ini diatur oleh pengirim dan akan dikurangi 1 setiap kali melewati sebuah node. Paket akan dibuang jika nilai Hop Limit mencapai nilai nol.

Source Address dan *Destination Address* terdiri dari 128 bit dan berisi informasi mengenai alamat IPv6 dari pengirim/asal dan tujuan akhir paket.

Extension Header. Dalam IPv6, informasi optional lapisan internet dikodekan dalam *extenstion header* yang terpisah dan diletakkan antara header IPv6 basic (dasar) dan header protokol lapisan yang lebih tinggi. Sebuah paket IPv6 mungkin memiliki nol, satu, atau banyak extension header, dan ditunjukkan oleh bagian next header dari header sebelumnya seperti yang dapat dilihat pada Gambar 2.10.

IPv6 header Next Header = Routing	Routing header Next Header = TCP	TCP header + data

Gambar 2.10 Contoh *Extention Header*

Extension header tidak diperiksa atau diproses oleh node manapun sepanjang jalur pengiriman paket, sampai paket diterima oleh node yang ditunjukkan oleh bagian *Destination Addess* dari *IPv6 header* [11].

2.6 Teknik Transisi IPv6

Teknik transisi secara umum dapat dibagi menjadi tiga kategori yaitu: *dual stack, protocol translation*, dan *tunneling*.

2.6.1 *Dual Stack*

Pada teknik *dual stack*, kedua protokol IPv4 dan IPv6 diimplementasikan pada sebuah node. Teknik dual stack merupakan pondasi untuk teknik transisi lainnya. Teknik ini digunakan dalam teknologi *protocol translation* dan *tunneling*.

2.6.2 *Protocol Translation*

Pada teknik *protocol translation*, format pesan dan informasi diterjemahkan antar protokol IP yang berbeda. Keuntungan dari teknik *protocol tranlation* adalah kedua aplikasi yang menggunakan protokol IP yang berbeda dapat saling berkomunikasi. Kerugian utamanya adalah teknik ini merusak karakteristik *end-to-end* dari internet dan tidak mendukung aplikasi jaringan tertentu.

Ketika pengguna internet yang menggunakan IPv6 ingin berkomunikasi dengan pengguna yang menggunakan IPv4, maka teknik *protocol translation* dapat menjadi pilihan.

2.6.3 *Tunneling*

Teknik *tunneling* digunakan untuk menghubungkan *host*/jaringan yang sama-sama menggunakan IPv6 melalui jaringan dimana terdapat router yang tidak mendukung IPv6. Teknik ini menyediakan link virtual melalui jaringan fisik dimana paket IPv6 dienkapsulasi dengan header IPv4 kemudian langsung dikirim ke jaringan IPv4. Enkapsulasi dilakukan oleh pengirim dan pada penerima dilakukan dekapsulasi.

Secara umum teknik ini dibagi menjadi dua yaitu teknik *tunneling* manual dan otomatis.

Kelebihan dari *tunneling* adalah teknik ini tidak mempengaruhi lapisan atasnya, namun tidak dapat digunakan pada node yang menggunakan protokol IP yang berbeda. Tugas akhir ini berfokus pada metode *tunneling* [3].

2.7 *Tunnel* Manual

Paket IPv6 dienkapsulasi didalam paket IPv4. Dibutuhkan router yang mendukung *dual-stack* pada kedua ujung *tunnel*. Header asli paket IPv6 tidak diubah. Paket dirutekan melalui jaringan IPv4 berdasarkan alamat tujuan dari *tunnel*. Keputusan jalur mana yang diambil bukan berdasarkan alamat tujuan IPv6 paket.

Cara *tunneling* seperti ini akan menyebabkan kesulitan dalam pengaturan jaringan yang besar karena banyaknya interkoneksi domain IPv6 melalui *tunnel* manual [3].

2.8 *Tunnel* Otomatis

32 bit paling belakang dari alamat IPv6 digantikan dengan alamat IPv4 32 bit dan 96 bit paling depan adalah 0:0:0:0:0:0. Sehingga diperoleh alamat IPv6 0:0:0:0:0:0:A.B.C.D, dimana A.B.C.D merupakan alamat IPv4. Meskipun teknik ini memungkinkan kita membuat *tunnel* secara otomatis, kelebihan IPv6 berupa banyaknya alamat tidak dapat direalisasikan karena tiap *host* harus memiliki alamat IPv4.

2.8.1 *Tunnel* 6to4

Diatur dalam RFC 3056 – *Connection of IPv6 Domains via IPv4 Clouds*. Jaringan IPv4 diperlakukan sebagai unicast point-to-point lapisan link. Tujuan dari penggunaan *tunnel* 6to4 adalah untuk menghubungkan domain IPv6 dan bukan untuk *host* individual. Meskipun paket IPv6 dienkapsulasi dalam paket IPv4, berbeda dengan metode manual, alamat tujuan *tunnel* tidak didefinisikan.

Internet Assigned Number Authority (IANA) telah memberikan awalan unik untuk routing *tunnel* 6to4. 13 bit unik Top Level Aggregator (TLA) adalah 0x0002, atau sering juga dituliskan 2002::/16. 32 bit berikutnya adalah alamat unik IPv4 dari *interface tunnel* 6to4. Ketika sebuah router 6to4 menerima paket IPv6 yang bukan bagian dari domainnya dengan FP/TLA berformat 2002, maka router tersebut akan mengenkapsulasi paket kedalam paket IPv4 dan mengubah protokol

fieldnya menjadi 41. Kemudian router akan menggunakan 32 bit IPv4 sebelumnya sebagai alamat tujuan dan mengirimkan paket sesuai dengan alokasi routing pada router tersebut. [3]

Jumlah bit	3	13	32	16	64
Medan	FP 001	TLA 0x0002	v4 addr	SLA ID	Int ID

Gambar 2.11 Pengalamatan *Tunnel* 6to4 [3]

2.8.2 *Tunnel Intra-Site Automatic Tunnel Addressing Protocol* (ISATAP)

ISATAP menghubungkan node *dual-stack* melalui jaringan IPv4. Jaringan tersebut diperlakukan sebagai sebuah lapisan *link* untuk IPv6 dan mendukung *tunneling* secara otomatis yang serupa dengan model *Non-Broadcast Multiple Access* (NBMA).

Teknik ini sangat mirip dengan 6to4 dengan perbedaan utama berupa letak alamat IPv4. IANA mengidentifikasi ISATAP dengan memberikan 16 bit unik hexadesimal pada bit ke 16 sampai 31 yaitu 5EFE dan 32 bit alamat IPv4 diletakkan di paling belakang [3].

Jumlah bit	3	13	8	24	16	8	8	8	8	32
Medan	FP	RES	TLA ID	NLA ID	SLA ID	0x00	0x00	0x5E	0xFE	v4 addr

Gambar 2.12 Pengalamatan *Tunnel* ISATAP [3]

2.9 Parameter Unjuk Kerja

Ada beberapa ukuran kinerja yang dijadikan sebagai parameter baik-buruknya peforma sistem, yaitu *bandwidth*, *packet loss*, *round-trip delay* (RTT), dan *jitter*. Parameter tersebut didefinisikan sebagai berikut:

2.9.1 *Bandwidth*

Bandwidth adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah jaringan. *Bandwidth* atau kapasitas saluran informasi merupakan kemampuan maksimum dari suatu alat untuk menyalurkan informasi per satuan waktu.

2.9.2 *Packet Loss*

Packet loss terjadi ketika satu atau lebih paket data gagal mencapai tujuannya. Packet loss dapat terjadi karena beberapa hal, misalnya dalam proses transmisi ada kemungkinan terjadi perubahan digit pada bit-bit yang ditransmisikan, misalnya bit "1" menjadi "0" akibat noise menyebabkan paket tersebut korup dan akhirnya di-discard, contoh lainnya apabila terjadi kemacetan jaringan, misalnya ketika sebuah router menerima paket data dengan laju konstan yang lebih besar dari kemampuannya memproses dan meneruskan, maka paket-paket yang tidak dapat ditampung akan di-discard.

2.9.3 *Roud-Trip Time* (RTT)

RTT merupakan penjumlahan waktu yang diperlukan sebuah paket untuk terkirim dan waktu yang dibutuhkan sampai balasan bahwa sinyal tersebut sudah sampai diterima oleh sumber. RTT disebut juga waktu ping.

2.9.4 *Jitter*

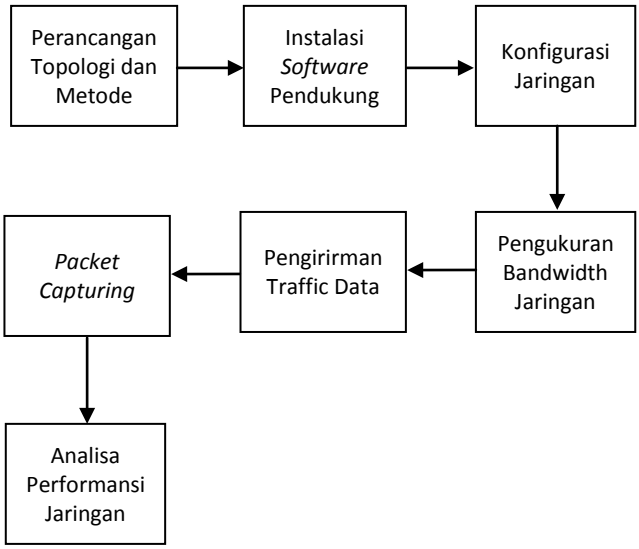
Seperti telah dijelaskan sebelumnya, ketika suatu file hendak dikirimkan dari satu host ke host lain maka file tersebut diubah menjadi data-stream dan dipecah menjadi beberapa segmen kemudian diberikan header Network layer dan menjadi paket.

Jitter terjadi ketika paket-paket data mengalami delay yang berbeda, hal ini menjadi masalah apabila aplikasi pada penerima bersifat time-sensitive (misalnya audio atau video).

BAB III
PERANCANGAN DAN IMPLEMENTASI

Tujuan dari tugas akhir ini adalah untuk mengetahui unjuk kerja dan pengaruh dari pengimplementasian *tunnel* manual, ISATAP, dan 6to4 pada jaringan IPv6 *over* IPv4. Hal ini dilakukan dengan mengirimkan paket IPv6 antar *end node* (*end-to-end*). Dari hasil yang diperoleh akan dibandingkan dengan jaringan *native-IPv6*, yaitu jaringan yang alamat IP-nya hanya terdiri dari IPv6.

Di dalam bab ini juga dibahas tentang hal-hal yang berkaitan dengan perancangan konfigrasi sistem yang digunakan dalam tugas akhir ini serta langkah-langkah yang dilakukan untuk memperoleh parameter-parameter yang ditentukan. Alur penelitian Tugas akhir ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Alur Penelitian

3.1 Perancangan Uji Unjuk Kerja Sistem

Untuk dapat menguji unjuk kerja suatu jaringan *tunnel* perlu dibuat suatu desain untuk mengimplementasikan ketiga metode *tunneling* tersebut. Pada bagian ini dijelaskan tentang perancangan, desain, dan implementasi ketiga jaringan *tunnel* tersebut. Pengujian dilakukan pada jaringan *Local Area Network* (LAN).

Dibawah ini merupakan langkah-langkah yang digunakan untuk perancangan dan implementasi dari masing-masing metode *tunneling* :

1. Desain arsitektur jaringan menggunakan program GNS3
2. Instalasi perangkat lunak pendukung
3. Konfigurasi *host* dan router
4. Melakukan pengujian terhadap parameter-parameter yang telah ditentukan

Sedangkan metode pengujian dan pengambilan data dilakukan dengan beberapa langkah sebagai berikut:

1. Implementasi jaringan *tunnel*
2. Pengukuran besar *bandwidth* kanal, dilakukan dengan mengirimkan paket TCP menggunakan program iperf dengan ukuran *window* 64Kbyte, 128Kbyte, 256Kbyte, dan 512 Kbyte. Hal ini dilakukan selama 100 detik dan sebanyak 10 kali kemudian dirata-ratakan
3. Besar *bandwidth* pengiriman paket untuk pengujian *jitter* dan *packet loss* ditentukan berdasarkan hasil *bandwidth* maksimum pada langkah 2, yaitu 25Mbps
4. Pengukuran besar *jitter* dan *packet loss*, dilakukan dengan mengirimkan paket UDP menggunakan iperf dengan ukuran *bandwidth* pengiriman sebesar 10Mbit/s, 15Mbit/s, 20Mbit/s, dan 25Mbit/s. Hal ini dilakukan selama 100 detik dan sebanyak 10 kali
5. Pengukuran besar RTT dengan mengirikan paket ICMPv6 menggunakan ping dengan ukuran paket sebesar 10Kbyte, 15Kbyte, 20Kbyte, 25Kbyte. Hal ini dilakukan selama 100 detik dan sebanyak 10 kali
6. Pengolahan data dan Penarikan kesimpulan

3.2 Kebutuhan Pendukung Infrastruktur

Kebutuhan pendukung infrastruktur pada perancangan dan implementasi sistem terbagi menjadi dua macam, yaitu *software* dan *hardware*, dimana keduanya saling mendukung satu sama lain.

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam implementasi sistem sinkronisasi di tugas akhir ini adalah sebagai berikut:

1. Komputer Server, komputer server akan bertindak sebagai server yang menerima pengiriman paket data UDP dan TCP dengan ukuran *window* tertentu serta mengirimkan paket ping. Dalam tugas akhir ini komputer server direpresentasikan sebagai "*Host 1*". Spesifikasi dari komputer server yang digunakan adalah:
    - Model : Toshiba Satellite L645
    - Processor : Intel(R) Core(TM) i3
    - RAM : 4 GB (2,93 GB usable)
    - System type : 32-bit Operating System
    - Operating System : Windows 7 Home Premium
2. Komputer *Client,* komputer *client* akan melakukan pengiriman paket UDP dengan besar *bandwidth* tertentu dan TCP dengan ukuran *window* tertentu ke komputer server. Komputer *client* direpresentasikan sebagai "*Host 2*".
    - Model : Lenovo 3359A86
    - Processor : AMD E2-2000 APU
    - RAM : 2 GB
    - System type : 32-bit Operating System
    - Operating System : Windows Ultimate 7
3. Router, router adalah perangkat keras jaringan yang berfungsi untuk meneruskan paket data antara jaringan komputer melalui proses yang dikenal dengan *routing*. Pada tugas akhir ini digunakan tiga buah router dengan spesifikasi yang sama, yaitu:
    - Model : Cisco 2801
    - DRAM : 128 MB
    - Compact Flash : 64 MB
    - NVRAM : 191 KB
    - Port LAN : two 10/100 Mbps
    - Consol Port : up to 115,2 kbps
    - Auxiliary Port : up tp 115,2 kbps
    - Low-speed serial (sync/async) interfaces : 2

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak digunakan untuk membuat dan mengelola struktur data dan mengakses data. Spesifikasi perangkat lunak yang digunakan adalah:

3.2.2.1 Graphic Network Simulator

GNS3 (*Graphic Network Simulator*) adalah *software open source* yang mensimulasikan jaringan yang kompleks semirip mungkin dengan cara kerja jaringan nyata. Semua ini tanpa didedikasikan perangkat keras jaringan seperti router dan switch. GNS3 memungkinkan simulasi router sesuai dengan versi IOSnya.

3.2.2.2 Cisco Internetwork Operating Sistem

Cisco IOS (*Internetwork Operating System*), Cisco IOS adalah perangkat lunak yang digunakan pada kebanyakan *Cisco Systems router* dan *network switch* Cisco saat ini. IOS adalah paket fungsi *routing, switching, internetworking* dan telekomunikasi yang terintegrasi ke dalam sistem operasi *multitasking*. Untuk dapat mengimplementasikan testbed maka versi IOS yang terinstall pada router harus dapat menjalankan layanan yang diinginkan dan memenuhi spesifikasi router.

3.2.2.3 Putty

Putty, merupakan terminal emulator gratis dan *open source*, sebuah serial konsol dan aplikasi transfer file jaringan. Putty mendukung beberapa protokol jaringan, termasuk SCP, SSH, Telnet, dan rlogin. Merupakan program yang siap dijalankan sehingga tidak perlu melakukan instalasi. Dapat diunduh dari www.putty.org.

3.2.2.4 Iperf

Iperf, merupakan alat pengujian jaringan yang dapat digunakan untuk membuat laju data *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP) dan mengukur *bandwidth* jaringan. Iperf ditulis dalam bahasa C dan dikembangkan oleh *Distributed Apllications Support Team* (DAST). Iperf memiliki fungsi klien dan server, dan dapat mengukur *bandwidth* antara kedua ujung, baik secara *unidirectional* atau *bidirectional*. Iperf dapat menguji kapasitas UDP dan TCP dan memungkinkan pengguna untuk menentukan ukuran datagram dan memberikan hasil ukuran *bandwidth* datagram dan *packet*

loss. Merupakan program yang siap dijalankan sehingga tidak perlu melakukan instalasi. Dapat diunduh dari www.iperf.fr.

3.2.2.5 Ping

Packet Internet Gopher (ping), program ini sudah termasuk dalam sistem operasi windows dan IOS router cisco. Ping dirancang untuk menguji koneksi jaringan secara sederhana dengan mengirimkan paket ICMPv4 atau ICMPv6 *echo-request* dan menunggu balasan paket *echo-reply*.

3.2.2.6 Trivial File Transfer Protocol Server

Trivial File Transfer Protocol (TFTP) server, merupakan protokol transfer file sederhana yang digunakan untuk transfer konfigurasi atau boot file antara dua mesin dalam lingkungan lokal. Dalam tugas akhir ini digunakan SolarWinds TFTP Server.

3.2.2.7 Wireshark

Wireshark, wireshark adalah program *paket analyzer* gratis dan *open-source*. Wireshark digunakan untuk pemecahan masalah jaringan, analisis, pengembangan perangkat lunak dan protokol komunikasi, dan pendidikan.

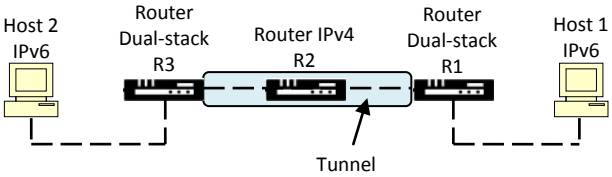
3.3 Arsitektur Jaringan

Jaringan *tunnel* merupakan cara yang mudah dan cepat untuk meneruskan paket IPv6 melalui jaringan IPv4. Dimana hanya router-router yang terletak di ujung jaringan *tunnel* saja yang diharuskan mendukung *dual- stack*.

Elemen-elemen yang diperlukan untuk menyimulasikan jaringan *tunnel* adalah :

- Node *IPv6-only*, pada tugas akhir ini digunakan dua buah *host* yang dikonfigurasi dengan IPv6.
- Router/jaringan *IPv4-only*, pada tugas akhir ini jaringan *IPv4-only* diwakilkan dengan sebuah router yang hanya dikonfigurasi dengan alamat IPv4.
- Router yang mendukung *dual-stack*, pada tugas akhir ini digunakan dua buah router. kedua router dikonfigurasi dengan alamat IPv6 pada *interface* yang terhubung dengan *host* dan IPv4 pada *interface* yang terhubung dengan router IPv4.

Sehingga arsitektur paling sederhana dari sebuah jaringan yang menggunakan *tunnel* dapat dilihat pada Gambar 3.2



Gambar 3.2 Topologi Jaringan *Tunnel*

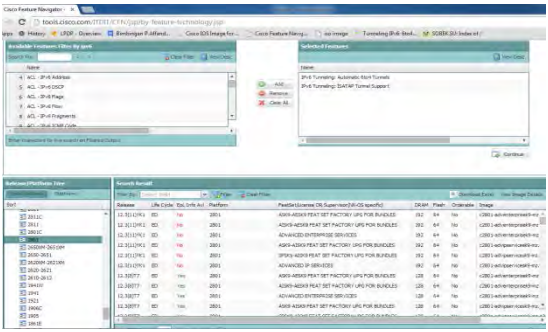
3.4 Instalasi IOS router

Sebelum melakukan konfigurasi pada router harus diperiksa apakah versi IOS pada router tersebut mendukung keperluan IPv6 dan *tunneling* yang ingin diuji. Versi IOS yang sedang digunakan dan yang ada pada flash dapat dilihat dengan perintah *show version* dan *show flash*. Dengan menggunakan perintah *show version* diperoleh ios router R1, R2, dan R3 masing-masing sebagai berikut :

- c2801-ipbasek9-mz.124-9.T7
- c2801-ipbasek9-mz.124-21
- c2801-ipbasek9-mz.124-18b

Ketiga versi ios ini tidak mendukung pengalamatan IPv6 maupun *tunnel* sehingga harus diganti. Penggantian IOS dilakukan dengan cara sebagai berikut :

Melalui situs *Cisco Feature Navigation*, cari versi IOS yang mendukung fitur yang diinginkan, seperti pada Gambar 3.3.



Gambar 3.3 *Cisco Feature Navigation*

Dari hasil pencarian akan muncul beberapa versi IOS yang mendukung. Selanjutnya periksa berapa besar memori DRAM dan Flash yang terinstall pada router, Gambar 3.4 memperlihatkan hasil perintah *show version*:

```
Cisco 2801 (revision 7.0) with 114688K/16384Kbytes of memory.
Processor board ID FHK1234F0WR
2 FastEthernet interfaces
2 Low-speed serial(sync/async) interfaces
DRAM configuration is 64 bits wide with parity disabled.
191K bytes of NVRAM.
62720K bytes of ATA CompactFlash (Read/Write)
```

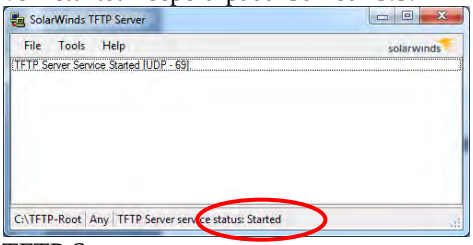
Gambar 3.4 Tampilan *show version*

Dari hasil "show version" tersebut terlihat bahwa router memiliki memori DRAM dan Flash sebesar (114688K+16384K) 131073Kbyte dan 62720Kbyte.

Dari hasil *Cisco Navigation Tool* dan besar memori yang tersedia pada router, diputuskan untuk menggunakan IOS versi c2801-adventerprisek9-mz.124-25a.bin untuk router *dual-stack* karena mendukung pengalamatan IPv6 dan konfigurasi *tunnel* yang diperlukan dan sesuai dengan memori yang tersedia, sedangkan untuk router IPv4 digunakan IOS versi c2801-advsecurityk9-mz.124-24.T6.bin, alasan penggunaan versi IOS ini akan dijelaskan pada BAB IV. IOS versi ini kemudian diunduh dari situs http://sobek.su/Cisco/IOS/ (situs ini bukanlah situs resmi milik cisco).

Dengan menggunakan server TFTP (SolarWinds TFTP Server), backup IOS router yang sedang berjalan kemudian copy IOS versi c2801-adventerprisek9-mz.124-25a.bin ke flash router. Hal tersebut dilakukan dengan cara sebagai berikut:

1. Nyalakan TFTP Server pada komputer, sehingga terlihat status TFTP server "*started*" seperti pada Gambar 3.5.



Gambar 3.5 TFTP Server

2. Susun topologi router dan server TFTP sehingga router dan laptop yang menjadi server dalam satu jaringan, contohnya dapat dilihat pada Gambar 3.6 :

Gambar 3.6 Contoh Topologi untuk *Update* IOS

3. Untuk melakukan backup terhadap IOS yang saat ini sedang digunakan dilakukan dengan perintah sebagai berikut:

```
Router#copy flash: c2801-ipbasek9-mz.124-9.T7.bin TFTP
Address or name of remote host []? 10.0.0.2
Destination filename [c2801-ipbasek9-mz.124-9.T7.bin]?
```

4. Menghapus IOS yang lama dapat dilakukan dengan menggunakan perintah:

```
Router#delete flash: c2801-ipbasek9-mz.124-9.T7.bin
```

5. Dengan menghapus IOS dari flash tidak akan mempengaruhi kinerja router karena IOS tersebut sudah di load ke RAM saat proses booting. Kemudian untuk mengganti IOS ke versi yang baru dilakukan dengan cara sebagai berikut:

```
Router#copy TFTP flash:
Address or name of remote host []? 10.0.0.2
Source filename []?c2801-adventerprisek9-mz.124-25a.bin
Destination filename[c2801-adventerprisek9-mz.124-25a.bin]?
Accessing   tftp://10.0.0.2/   c2801-adventerprisek9-mz.124-
25a.bin...
```

6. Selanjutnya atur agar router menggunakan IOS yang baru ketika booting dan lakukan booting ulang, hal tersebut dilakukan dengan perintah berikut:

```
Router(conf)#boot system flash flash:c2801-adventerprisek9-
mz.124-25a.bin
Router(conf)#exit
Router#reload

System configuration has been modified. Save? [yes/no]: y
Overwrite the previous NVRAM configuration?[confirm]
```

7. Lakukan hal yang sama untuk kedua router sisanya.

3.5 Konfigurasi *Host* dan Router

Konfigurasi pada setiap node berbeda-beda, hal ini disebabkan karena jenis perangkat dan konfigurasi masing-masing interfacenya. Pada subbab ini akan membahas mengenai pemberian alamat IP, routing, serta konfigurasi yang digunakan.

3.5.1 Pengalamatan IPv4

Traffic IPv6 dienkapsulasikan menjadi paket IPv4 untuk di teruskan melalui jaringan sehingga koneksi IPv4 antar router harus dipastikan berjalan dengan baik. Pengalamatan IPv4 yang digunakan dalam tugas akhir ini dapat dilihat pada Tabel 3.1, sedangkan teknik *routing* yang digunakan adalah *static routing*.

Tabel 3.1 Tabel Pengalamatan IPv4

Perangkat	Interface	Alamat IPv4
Router R1	f0/1	10.0.0.1/30
Router R2	f0/1	10.0.0.2/30
Router R2	f0/0	10.0.0.5/30
Router R3	f0/0	10.0.0.6/30

Pengaturan alamat IP dilakukan dengan menggunakan perintah-perintah:

```
Router#configure terminal
Router(config)#interface <nama interface>
Router(config-if)#no shutdown
Router(config-if)#ip address <alamat IP><subnet mask>
```

Keterangan:

- Configure terminal : untuk memasuki *global configuration mode*
- Interface : memilih antarmuka yang ingin dikonfigurasi
- No shutdown : untuk menyalakan antarmuka yang sedang dikonfigurasi, secara default antarmuka dalam keadaan tidak aktif
- Ip address : mengatur alamat IP antarmuka yang sedang dikonfigurasi

Sehingga perintah untuk melakukan konfigurasi pengalamatan IPv4 pada Router R1 adalah sebagai berikut:

```
R1#configure terminal
R1(config)#interface f0/1
R1(config-if)#no shut
R1(config-if)#ip address 10.0.0.1 255.255.255.252
```

Hal yang serupa diterapkan pada router R2 dan R3 dengan alamat IP, subnet mask, dan antarmuka sesuai dengan Tabel 3.1. Setelah semua alamat pada Tabel 3.1 sudah dialokasikan maka perlu dilakukan *routing* agar semua router dapat berkomunikasi. Teknik routing yang digunakan adalah *static routing* untuk menghindari *HELLO packet* dan paket-paket *neighbor adjacency* yang terjadi pada *dynamic routing*. Perintah umum untuk melakukan s*tatic routing* adalah sebagai berikut :

```
Router#configure terminal
Router(config)#ip route <network tujuan> <subnet mask
tujuan> <next hop>
```

Keterangan:

- Ip route : untuk membuat *static route*
- Network tujuan : alamat *network* tujuan
- Subnet mask : subnet mask dari alamat tujuan
- Next hop: alamat IP hop berikutnya

Sehingga perintah untuk melakukan konfigurasi routing IPv4 pada Router R1 adalah sebagai berikut :

```
Router#configure terminal
Router(config)#ip route 10.0.0.4 255.255.255.252 10.0.0.2
```

Sedangkan untuk Router R3 adalah sebagai berikut:

```
Router#configure terminal
Router(config)#ip route 10.0.0.0 255.255.255.252 10.0.0.5
```

Tidak perlu dilakukan *routing* pada Router R2 karena kedua *interface* Router R1 dan R3 yang diberikan alamat IPv4 bersifat *directly connected*.

3.5.2 Pengalamatan IPv6

Pengalamatan IPv6 yang digunakan dalam tugas akhir ini dapat dilihat pada Tabel 3.2.

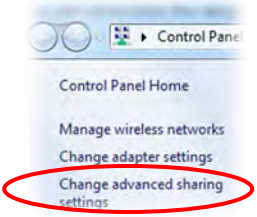
Tabel 3.2 Tabel Pengalamtan IPv6

Perangkat	Interface	Alamat IPv6
Host 1	Ethernet	fd87::1/127
Router R1	F0/0	fd87::/127
Router R3	F0/1	fd87::2/127
Host 2	Ethernet	fd87::3/127

3.5.2.1 Konfigurasi IPv6 pada Host

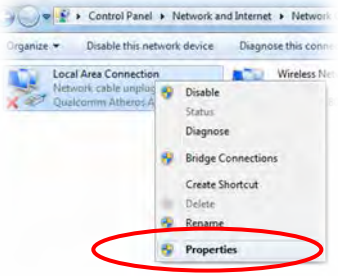
Kedua *host* merupakan *node IPv6-only* dan semua adaptor jaringan selain adaptor *Local Area Network* dinon-aktifkan untuk menghindari paket-paket yang tidak diinginkan. Untuk mengatur hal-hal tersebut pada kedua *host* dilakukan langkah-langkah sebagai berikut:

1. Klik gambar *network icon* kemudian pilih *Open Network and Sharing Center*, akan terbuka jendela baru yang memperlihatkan informasi dasar jaringan *host* tersebut
2. Pilih "*Change adapter settings*" yang terletak pada bagian kiri jendela seperti pada Gambar 3.7 sehingga terlihat adaptor-adaptor yang terinstall pada *host*



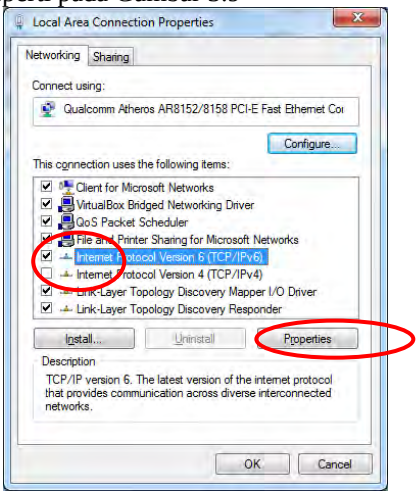
Gambar 3.7 Mengubah *Setting Adapter*

3. *Disable* semua adapator kecuali adaptor *Local Area Connection/Ethernet*, Klik kanan pada adaptor *Local Area Connection/Ethernet* dan pilih *Properties* seperti terlihat pada Gambar 3.8 , akan terbuka jendela baru yang memperlihatkan macam-macam properti adapator tersebut



Gambar 3.8 *Adapter Properties*

4. Hilangkan centang pada *Internet Protocol Version 4* (TCP/IPv4), Centang *Internet Protocol Version 6* (TCP/IPv6), dan klik *Properties* seperti pada Gambar 3.9



Gambar 3.9 Pemilihan Versi IP

5. Pilih "*Use the following IPv6 addess:*" dan masukkan setting IP yang diinginkan, sehingga untuk *host 1* seperti pada Gambar 3.10

Gambar 3.10 Pemberian Alamat IPv6

6. Hal yang serupa dilakukan pada Host 2 dengan *IPv6 address*dan *Subnet prefix length* sesuai dengan Tabel 3.2.

3.5.2.2 Konfigurasi IPv6 pada Router

Untuk memberikan alamat IPv6 pada router mirip dengan pada IPv4, bedanya adalah perintah "*ip address*" diubah menjadi "*ipv6 address*" dan cara menuliskan subnet mask menggunakan "/<*subnet mask*>". Sehingga perintah untuk melakukan konfigurasi pengalamatan IPv6 pada Router R1 adalah sebagai berikut:

```
R1#configure terminal
R1(config)#interface f0/0
R1(config-if)#no shut
R1(config-if)#ipv6 address fd87::/127
```

Perintah "*ipv6 address*" sama seperti "*ip address*" namun untuk pemberian alamat IPv6, sedangkan konfigurasi untuk router R3 adalah :

```
R1#configure terminal
R1(config)#interface f0/1
R1(config-if)#no shut
R1(config-if)#ipv6 address fd87::2/127
```

3.6 Konfigurasi *Tunnel* IPv6

Setelah melakukan konfigurasi IPv6 *host* dan router dilakukan uji coba ping dari *host* satu ke *host* satunya dan hasilnya gagal, hal ini disebabkan karena paket ICMP yang dikirim menggunakan alamat IPv6 sebagai *source* dan *destination IP addressnya* sehingga router R2 tidak bisa meneruskan paket tersebut, Hal yang sama berlaku untuk jenis

paket lainnya. Untuk mengatasi hal ini dilakukan konfigurasi *tunnel* antara router R1 dengan R3.

Pada subbab ini akan dibahas mengenai pengalamatan serta konfigurasi yang digunakan untuk masing-masing *tunnel*. Pengalamatan yang digunakan pada *interface tunnel* dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tabel Pengalamatan I*nterface Tunnel*

Perangkat	Mode Tunnel	Tunnel Source	Tunnel Destination	Alamat IPv6
Router R1	Manual	F0/1	10.0.0.6	2001::1/64
Router R2	Manual	F0/0	10.0.0.1	2001::2/64
Router R1	ISATAP	F0/1	-	2001::5efe:a00:1/64
Router R2	ISATAP	F0/0	-	2001::5efe:a00:6/64
Router R1	6to4	F0/1	-	2002:a00:1::1/64
Router R2	6to4	F0/0	-	2002:a00:6::1/64

Perintah-perintah yang digunakan untuk mengkonfigurasi *tunnel* adalah sebagai berikut:

```
Router#configure terminal
Router(config)#interface tunnel <nomor>
Router(config-if)#ipv6 enable
Router(config-if)#ipv6 address <alamat ip/subnet mask>
Router(config-if)#tunnel mode <mode tunnel> <jenis
tunnel>
Router(config-if)#tunnel source <antarmuka source>
Router(config-if)#tunnel destination <alamat tujuan>
```

Keterangan:

- Interface *tunnel* : mespesifikasikan nomor dan antarmuka *tunnel* serta memasuki mode konfigurasi antarmuka
- IPv6 enable : enable IPv6 pada antarmuka yang sedang dikonfigurasi
- Tunnel mode : metode enkapsulasi *tunnel*, dalam tugas akhir ini digunakan "ipv6ip" yang berarti *IPv6 over IP encapsulation.*
- Tunnel source : menunjukkan antarmuka sumber (dapat berupa jenis antarmukanya atau alamat IP dari antarmuka tersebut) antarmuka yang dispesifikasikan sebagai *tunnel source* harus terkonfigurasi dengan alamat IPv4
- Tunnel destination : menunjukkan alamat IPv4 antarmuka tujuan

Setelah semua antarmuka diberikan alamat IPv6 maka *host 1* belum bisa melakukan ping ke *host 2* dan sebaliknya, hal ini karena belum dilakukan *routing* sehingga kedua jaringan tersebut belum saling "kenal".

Perintah-perintah diatas juga diterapkan pada router R3 dengan menggunakan alamat IP serta *tunnel source* dan *destination* yang sesuai dengan Tabel 3.3. Protokol *routing* untuk *tunnel* menggunakan *routing statik*, perintah yang digunakan untuk mengatur *routing statik* adalah sebagai berikut:

```
Router(config)#ipv6 unicast-routing
Router(config)#ipv6 route <alamat tujuan>/<subnet mask>
<next hop>
```

Keterangan :

- IPv6 unicast-routing : *enable* penerusan datagram IPv6 *unicast*
- Ipv6 route : sama seperti perintah "*ip route*" namun digunakan untuk melakukan *static routing* pada IPv6

3.6.1 Konfigurasi *Tunnel* Manual

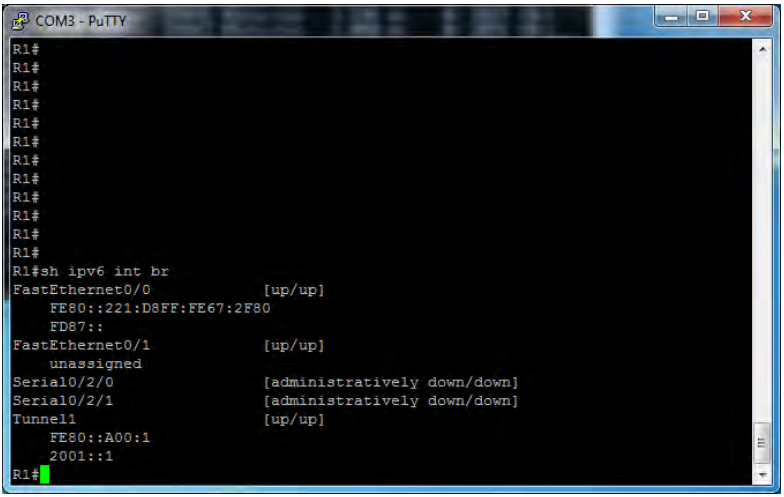
Berdasarkan penjelasan diatas maka perintah yang digunakan untuk melakukan konfigurasi *tunnel* manual adalah:

```
R1#configure terminal
R1(config)#interface tunnel 1
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2001::1/64
R1(config-if)#tunnel mode ipv6ip
R1(config-if)#tunnel source f0/1
R1(config-if)#tunnel destination 10.0.0.6
```

Untuk memeriksa apakah alamat IPv6 yang telah diberikan kepada antarmuka *tunnel* dan router benar dapat dilakukan dengan memasukkan perintah

```
Router(config)#show ipv6 interface brief
```

Sebagai contoh, Gambar 3.11 menunjukkan hasil dari perintah diatas yang diterapkan pada router R1.



Gambar 3.11 Alamat IPv6 R1 pada *Tunnel* Manual

Dari Gambar 3.11 terlihat bahwa alamat IPv6 yang diberikan pada router R1 sudah sesuai dan kondisi semua antarmuka siap untuk menerima dan mengirimkan paket data, hal yang sama dilakukan pada router R3.

Seusai dengan penjelasan sebelumnya, perintah yang digunakan untuk melakukan konfigurasi *routing* untuk jaringan *tunnel* manual adalah sebagai berikut:

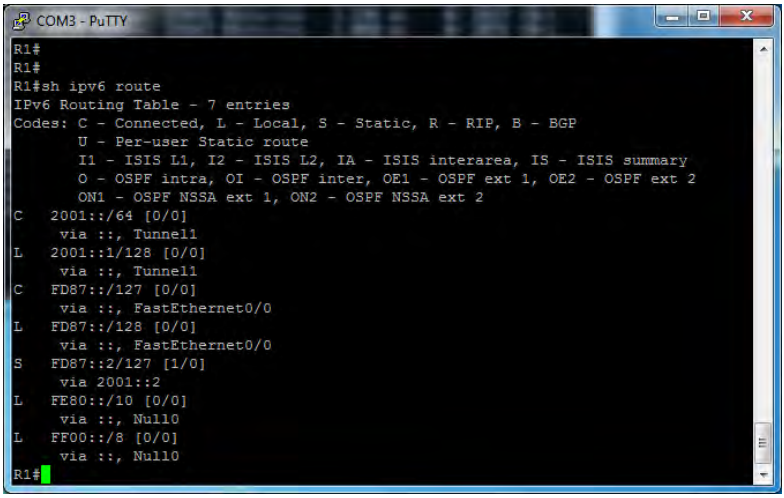
```
R1(config-if)#ipv6 unicast-routing
R1(config)#ipv6 route fd87::2/127 2001::2
R3(config-if)#ipv6 unicast-routing
R3(config)#ipv6 route fd87::/127 2001::1
```

Untuk memeriksa apakah tabel routing pada sebuah router dapat dilakukan dengan menggunakan perintah

```
Router#show ipv6 route
```

Routing table menunjukkan alamat atau antarmuka mana yang akan digunakan router untuk meneruskan paket ke alamat IP tujuan tertentu dan sering digunakan untuk melakukan *troubleshooting* apabila

terdapat koneksi yang putus. Gambar 3.12 menunjukkan routing tabel pada router R1



Gambar 3.12 Tabel *Routing* R1 pada *Tunnel* Manual

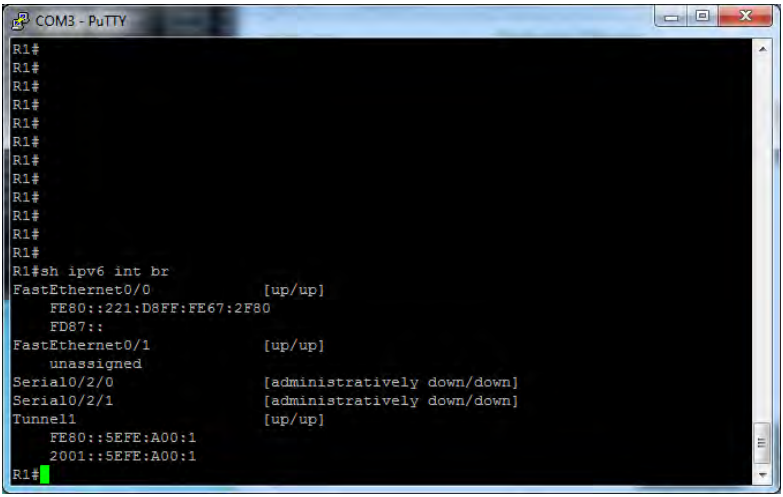
Dari Gambar 3.12 terlihat bahwa routing statis untuk menuju ke jaringan IPv6 *host2* (FD87::2) melewati tunnel manual (2001::2) sudah terdaftar dalam *routing table* router R1, hal yang sama dilakukan pada router R3.

3.6.2 Konfigurasi *Tunnel* ISATAP

Meneruskan dari subsub bab 3.4.1 dan 3.4.2, konfigurasi *tunnel* ISATAP dilakukan dengan menggunakan perintah berikut :

```
R1#configure terminal
R1(config-if)#interface tunnel 1
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2001::/64 eui-64
R1(config-if)#tunnel mode ipv6ip isatap
R1(config-if)#tunnel source f0/1
```

Sebagai contoh, Gambar 3.13 menunjukkan hasil "*show ipv6 interface brief*" pada router R1



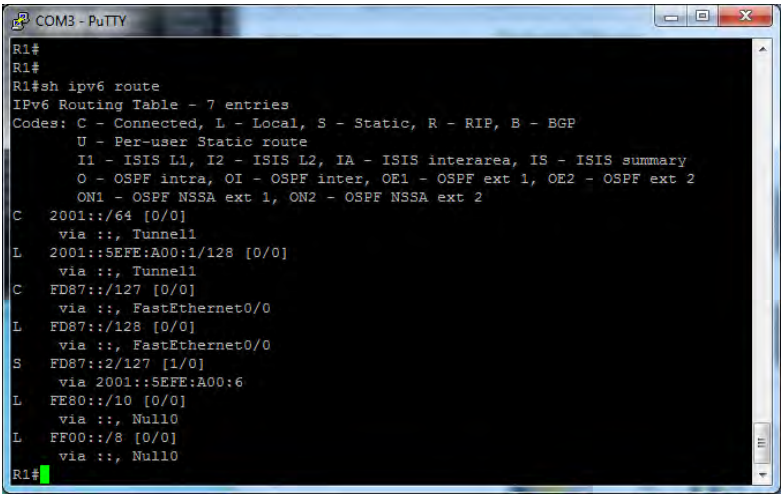
Gambar 3.13 Alamat IPv6 R1 pada *Tunnel* ISATAP

Dari Gambar 3.13 terlihat bahwa alamat IPv6 yang diberikan pada router R1 sudah sesuai dan kondisi semua antarmuka siap untuk menerima dan mengirimkan paket data, hal yang sama dilakukan pada router R3.

Sedangkan konfigurasi untuk routing jaringan *tunnel* ISATAP adalah sebagai berikut:

```
R1(config-if)#ipv6 unicast-routing
R1(config)#ipv6 route fd87::2/127 2001::5efe:a00:6
R3(config-if)#ipv6 unicast-routing
R3(config)#ipv6 route fd87::/127 2001::5efe:a00:1
```

Gambar 3.14 menunjukkan *routing tabel* router R1 pada jaringan *tunnel* ISATAP



Gambar 3.14 Tabel *Routing* R1 pada *Tunnel* ISATAP

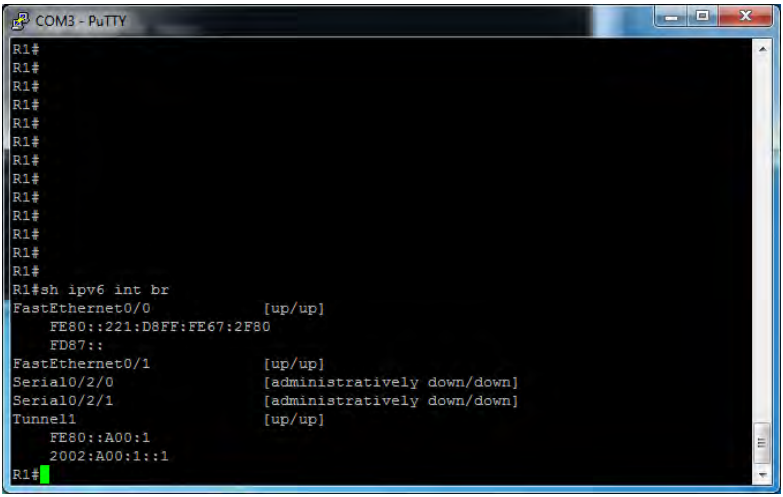
Dari Gambar 3.14 terlihat bahwa routing statis untuk menuju ke jaringan IPv6 *host2* (FD87::2) melewati tunnel ISATAP (2001::5EFE:A00:6) sudah terdaftar dalam *routing table* router R1, hal yang sama dilakukan pada router R3.

3.6.3 Konfigurasi *Tunnel* 6to4

Dan konfigurasi *tunnel* 6to4 adalah:

```
R1#configure terminal
R1(config-if)#interface tunnel 1
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2002:a00:1::1/64
R1(config-if)#tunnel mode ipv6ip 6to4
R1(config-if)#tunnel source f0/1
```

Sebagai contoh, Gambar 3.15 menunjukkan hasil "*show ipv6 interface brief*" pada router R1



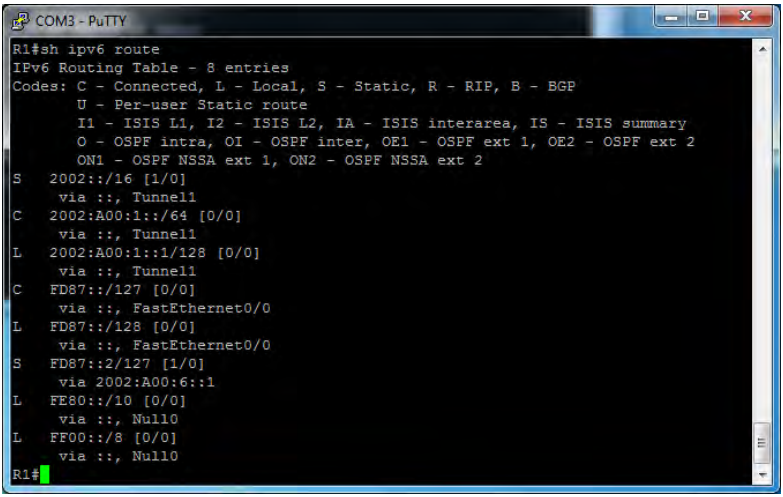
Gambar 3.15 Alamat IPv6 R1 pada *Tunnel* 6to4

Dari Gambar 3.15 terlihat bahwa alamat IPv6 yang diberikan pada router R1 sudah sesuai dan kondisi semua antarmuka siap untuk menerima dan mengirimkan paket data, hal yang sama dilakukan pada router R3.

Dan konfigurasi untuk routing jaringan *tunnel* 6to4 adalah sebagai berikut:

```
R1(config-if)#ipv6 unicast-routing
R1(config)#ipv6 route 2002::/16 tunnel 1
R1(config)#ipv6 route fd87::2/127 2002:a00:6::1
R3(config-if)#ipv6 unicast-routing
R3(config)#ipv6 route 2002::/16 tunnel 1
R3(config)#ipv6 route fd87::/127 2002:a00:1::1
```

Gambar 3.16 menunjukkan *routing tabel* router R1 pada jaringan *tunnel* ISATAP

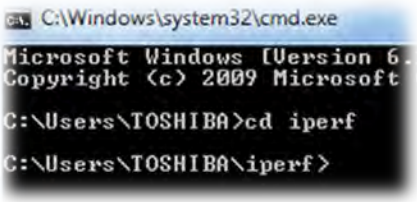


Gambar 3.16 Tabel *Routing* R1 pada *Tunnel* 6to4

Dari Gambar 3.16 terlihat bahwa routing statis untuk menuju ke jaringan IPv6 *host2* (FD87::2) melewati tunnel 6to4 (2002:A00:6::1) sudah terdaftar dalam *routing table* router R1, hal yang sama dilakukan pada router R3.

3.7 Desain dan Implementasi Pengukuran

Pengambilan data dilakukan dengan cara kedua *host* mengakses iperf dengan cara membuka *command prompt* kemudian mengubah direktori menjadi *folder* disimpannya program iperf, sebagaimana diperlihatkan pada Gambar 3.17.



Gambar 3.17 Perubahan Direktori Iperf

3.7.1 Pengujian *Bandwidth*

Host client mengirimkan paket TCP menggunakan iperf dengan *window size* sebesar 64Kbyte, 128Kbyte, 256Kbyte, dan 512Kbyte pada *port* yang telah ditentukan selama 100 detik sebanyak 10 kali. Perintah yang digunakan pada *host client* adalah sebagai berikut:

```
iperf -c fd87::1 -i 1 -V –p 5001 -f k -w 64k  -t 100
```

Host server akan mendengarkan (*listen*) paket TCP pada *port* yang telah ditentukan. Perintah yang digunakan pada *host server* adalah sebagai berikut:

```
iperf -s -i 1 –V –p 5001 -f k
```

keterangan:

- -s : mode server
- -c : mode client
- -i : interval antar laporan
- -V : menggunakan protokol IPv6
- -p : port yang digunakan
- -f : format laporan
- -w : ukuran window
- -t : lama pengiriman dilakukan (dalam satuan detik)

Hal yang sama dilakukan untuk pengukuran data dengan ukuran *window* seperti telah dijelaskan sebelumnya.

3.7.2 Pengujian *Jitter* dan *Packet Loss*

Host client mengirimkan paket UDP dengan laju data sebesar 10Mbit/s, 15Mbit/s, 20Mbit/s, dan 25Mbit/s menggunakan iperf pada *port* yang telah ditentukan selama 100 detik sebanyak 10 kali. Perintah yang digunakan adalah sebagai berikut:

```
iperf -c fd87::1 -u -i 1 -p 5001 -V -f k -b 10M -t 100
```

Host server mendengarkan (*listen*) paket UDP pada port yang telah ditentukan. Perintah yang digunakan adalah sebagai berikut:

```
iperf -s -u -i 1 -p 5001 -V -f k
```

keterangan :

- -s : mode server
- -c : mode client
- -u : menggunakan udp
- -i : interval antar laporan
- -p : port yang digunakan untuk mengirim
- -V : menggunakan protokol IPv6
- -f : format laporan
- -b : ukuran bandwidth
- -t : lama pengiriman dilakukan

Hal yang sama dilakukan untuk pengukuran data dengan laju data seperti telah dijelaskan sebelumnya.

3.7.3 Pengujian *Round-Trip Time*

Pengujian RTT dilakukan dengan cara mengirimkan paket *echo-request* ICMPv6 dari *host* client dengan ukuran paket 10kbyte, 15kbyte, 20kbyte, dan 25kbyte menggunakan ping sebanyak 100 paket dan diulangi sebanyak 10 kali. Perintah yang digunakan adalah sebagai berikut:

```
Ping fd87::7 –n 100 –l 10000
```

Keterangan:

- -n : jumlah paket yang dikirimkan
- -l : besar paket yang dikirimkan

3.8 Desain Jaringan Native IPv6 Sebagai Pembanding

Pada tugas akhir ini akan dibuat testbed jaringan *native-IPv6* sebagai parameter *control*/pembanding. Jaringan *native-IPv6* adalah sebuah jaringan yang hanya menggunakan pengalamatan IPv6.

3.8.1 Alamat IPv6 Jaringan *Native-IPv6*

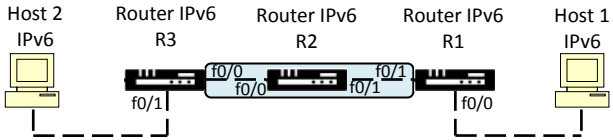
Pengalamatan alamat IPv6 yang digunakan untuk jaringan ini seperti pada Tabel 3.4.

Tabel 3.4 Tabel Pengalamatan Jaringan *Native-IPv6*

Perangkat	Interface	Alamat IPv6
Host server	Ethernet	fd87::1/127
Router R1	f0/0	Fd87::/127
Router R1	F0/1	Fd8::2/127
Router R2	F0/1	Fd87::3/127
Router R2	F0/0	Fd87::4/127
Router R3	F0/0	Fd87::5/127
Router R3	F0/1	Fd87::6/127
Host client	Ethernet	Fd87::7/127

3.8.2 Topologi Konfigurasi Jaringan *Native-IPv6*

Jaringan *native-IPv6* terdiri dari tiga buah router dan dua buah *host* dengan spesifikasi yang sama dengan yang digunakan pada jaringan *tunnel* seperti yang diperlihatkan Gambar 3.18



Gambar 3.18 Topologi Jaringan *Native-IPv6*

Dengan pengalamatan sesuai dengan Tabel 3.4 serta menggunakan routing statik. Pada jaringan ini dilakukan pengujian *bandwidth, jitter, packet loss*, dan RTT dengan cara yang sama dengan yang dilakukan pada jaringan *tunnel*.

BAB IV
ANALISA DATA

4.1 Uraian Umum

Pada bab ini ditampilkan data yang telah diperoleh dari pengujian ketiga sistem *tunnel* manual, ISATAP, dan 6to4 dan dilakukan analisa performansi serta cara kerja ketiga metode *tunneling* tersebut berupa pegujian interkoneksi, sistem pengalamatan, dan proses enkapsulasi paket. Pengujian interkoneksi dilakukan dengan menggunakan tracert dari *host1* ke *host2* untunk memastikan kedua *host* sudah terhubung melalui *tunnel*, sedangkan sistem pengalamatan dan proses enkapsulasi dilakukan dengan cara menangkap paket-paket yang melalui router R2.

Analisa performansi dilakukan dengan cara membandingkan parameter-parameter performansi pada ketiga jaringan *tunnel* dengan hasil yang diperoleh pada jaringan *native-IPv6*. Parameter-parameter analisa performansi ini meliputi *bandwidth*, RTT, *jitter*, dan *packet loss*. Pengukuran *bandwidth* dilakukan dengan mengirmkan paket TCP dari client ke server dengan menggunakan ukuran *window* 32, 64, 128, 256, dan 512 Kbyte. Pengukuran jitter dan packet loss dilakukan dengan pengiriman paket UDP dari client ke server, dimana ukuran paket disesuaikan dengan besar bandwidth yang telah diperoleh. Sedangkan pengukuran RTT dilakukan dengan mengirimkan paket *echo-request* ICMPv6. Data-data tersebut digunakan sebagai dasar untuk menarik kesimpulan dan ditampilkan dalam bentuk grafik dan tabel.

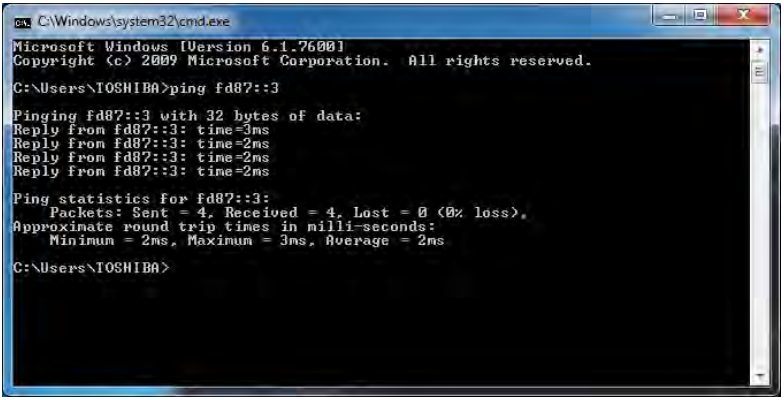
4.1.1 Analisa Sistem Pengalamatan *Tunnel* Manual

Tunnel manual dianggap setara dengan link permanen antara dua domain IPv6 melalui *backbone* IPv4. Sebuah alamat IPv6 dikonfigurasi secara manual pada antarmuka *tunnel*, dan alamat IPv4 dikonfigurasi secara manual sebagai *tunnel source* dan *tunnel destination*. Host atau perangkat di setiap ujung *tunnel* harus mendukung kedua protokol IPv4 dan IPv6. Untuk memeriksa alamat IPv6 yang telah diberikan kepada router dapat dilakukan dengan menggunakan perintah

Pemberian alamat IPv6 pada antarmuka *tunnel* manual tidak perlu mengikuti format alamat yang khusus selama kedua ujung *tunnel* berada dalam alamat *network-id* yang sama.

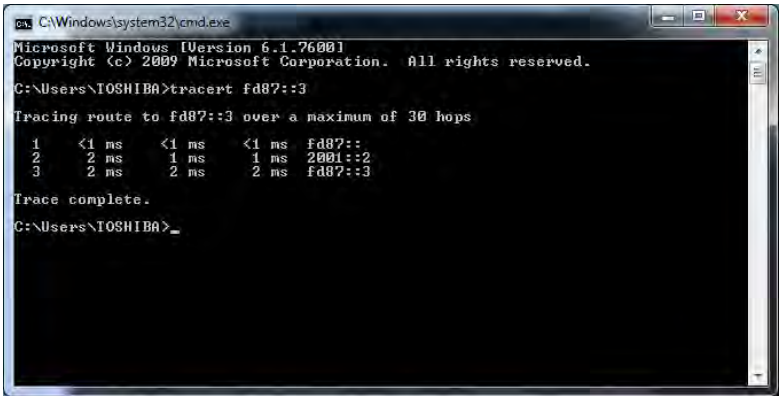
4.1.2 Analisa Pengujian Interkoneksi Sistem *Tunnel* Manual

Sistem jaringan yang telah dikonfigurasi harus diperiksa apakah node yang satu sudah terhubung dengan yang lainnya. Pertama dilakukan pengujian interkoneksi jaringan IPv4 dengan cara melakukan ping dari antarmuka FastEthernet 0/1 router R1 ke antarmuka FastEthernet 0/0 router R3. Jika ping berhasil maka dilanjutkan dengan menguji interkoneksi jaringan IPv6 dengan cara melakukan ping dari *host1* ke *host2*.



Gambar 4.1 Ping Antar *Host* pada Jaringan *Tunnel* Manual

Dari Gambar 4.1 terlihat bahwa *host1* menerima balasan dari *host2*, hal ini menandakan bahwa kedua *host* sudah dapat saling berkomunikasi. Setelah diketahui bahwa kedua *host* sudah saling terkoneksi maka langkah selanjutnya adalah menguji apakah kedua *host* tersebut terkoneksi melalui jaringan *tunnel* atau jaringan yang lain. Dengan mengggunakan tracert dapat diketahui jalur mana yang digunakan *host1* untuk mencapai *host2* atau sebaliknya. Gambar 4.2 memperlihatkan hasil tracert dari *host*1 ke *host*2.



Gambar 4.2 Tracert Jaringan *Tunnel* Manual

Dari Gambar 4.2 terlihat bahwa rute yang dilalui paket untuk mencapai *host2* adalah melalui gateway (fd87::) kemudian *tunnel* manual (2001::2), dan akhirnya sampai ke *host*2 (fd87::3). Dari hasil ping dan tracert terlihat bahwa kedua *host* telah menjalin komunikasi melalui *tunnel* manual.

4.1.3 Analisa Sistem Pengalamatan *Tunnel* ISATAP

ISATAP menggunakan alamat *unicast* yang terdiri dari 64-bit IPv6 *prefix* dan *interface identifier* 64-bit. *Interface identifier* diberikan dalam format EUI-64 dimana 32 bit pertama berisi nilai 000:5EFE untuk menunjukkan bahwa alamat tersebut adalah alamat ISATAP IPv6.

Perintah *tunnel source* yang digunakan dalam mengkonfigurasi *tunnel* ISATAP menunjuk ke sebuah antarmuka dengan alamat IPv4 yang telah terkonfigurasi pada router. Dalam tugas akhir ini *prefix* alamat IPv6 ISATAP dibuat berbeda agar mudah membedakan alamat *tunnel* dengan *native*. Alamat IPv6 antarmuka *tunnel* harus dikonfigurasi dengan alamat EUI-64 karena 32 bit terakhir di *interface identifier* diperoleh dengan menggunakan alamat *tunnel source* IPv4.

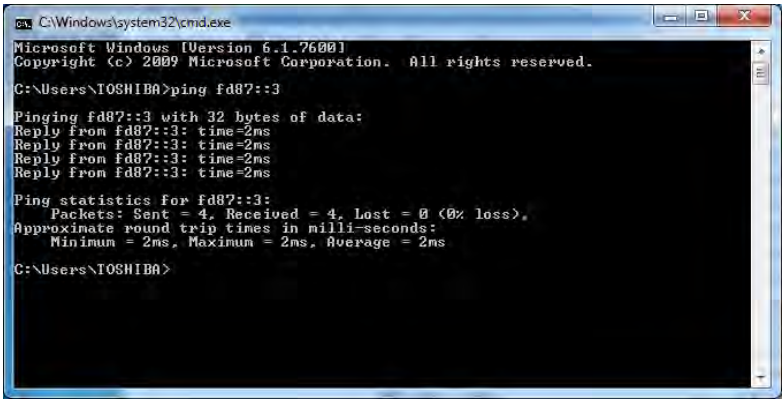
Sesuai dengan format pengalamatan *tunnel* ISATAP, alamat IPv6 yang diperoleh dari konfigurasi antarmuka *tunnel* di router R3 adalah 2001::5efe:a00:6 karena menggunakan FastEthernet 0/0 sebagai *tunnel source*-nya, hal ini terlihat dari 32 bit terakhir alamat IPv6 *tunnel* (0a00:0006) yang merupakan alamat IPv4 FastEthernet 0/0 (10.0.0.6)

yang dituliskan dalam format hexadecimal. Hal yang sama berlaku pada antarmuka *tunnel* router R1.

4.1.4 Analisa Pengujian Interkoneksi Sistem *Tunnel* ISATAP

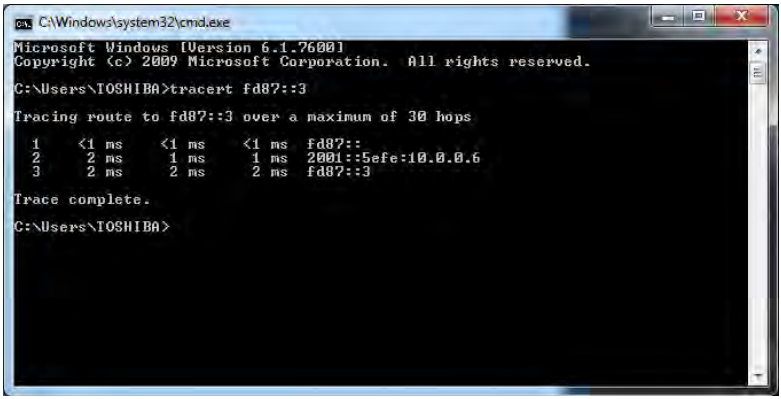
Metode pengujian interkoneksi dari jaringan *tunnel* ISATAP sama dengan pengujian pada jaringan *tunnel* manual.

Seperti halnya yang dilakukan sebelumnya, setelah dipastikan terjalinnya interkoneksi jaringan IPv4 dilakukan ping dari *host1* ke *host2* untuk memastikan interkoneksi IPv6. Gambar 4.3 menunjukkan hasil ping dari *host1* ke *host2*.



Gambar 4.3 Ping Antar *Host* pada Jaringan *Tunnel* ISATAP

Dari Gambar 4.3 terlihat bahwa *host1* menerima balasan dari *host2*, hal ini menandakan bahwa kedua *host* sudah dapat saling berkomunikasi. Selanjutnya dilakukan tracert untuk melihat rute yang ditempuh *host1* untuk menghubungi *host2*. Gambar 4.4 menunjukkan hasil dari tracert *host1* ke *host 2*



Gambar 4.4 Tracert Jaringan *Tunnel* ISATAP

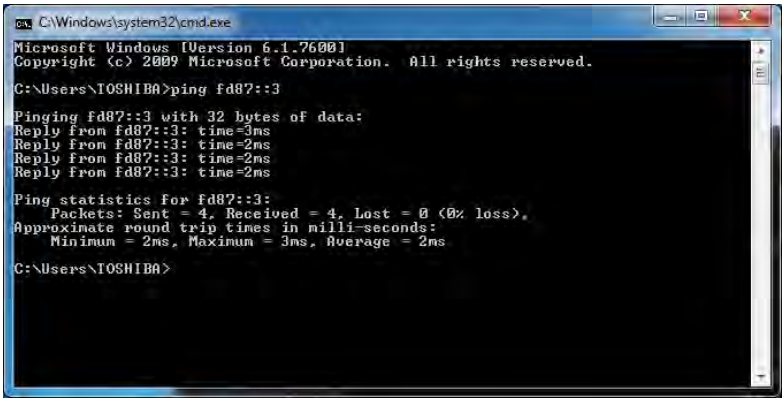
Dari Gambar 4.4 terlihat bahwa rute yang dilalui paket untuk mencapai *host2* adalah melalui gateway (fd87::) kemudian *tunnel* ISATAP (2001::5efe:a00:6), dan akhirnya sampai ke *host*2 (fd87::3). Hal ini menunjukkan terjalinnya koneksi antar kedua *host* melalui *tunnel* ISATAP.

4.1.5 Analisa Sistem Pengalamatan *Tunnel* 6to4

Sebuah *tunnel* 6to4 dapat dikonfigurasi pada *boarder router* di jaringan IPv6 yang terisolasi, sehingga menciptakan sebuah *tunnel* ke *boarder router* yang berada dalam jaringan IPv6 lain berbasis per-paket melalui infrastruktur IPv4. Tujuan *tunnel* ditentukan oleh alamat IPv4 dari *boarder router* yang diperoleh dari alamat IPv6 yang dimulai dengan *prefix* 2002::/16, dimana formatnya adalah 2002:*border-router-IPv4-address*::/48. Setelah alamat IPv4 *border router* adalah 16 bit yang dapat digunakan untuk nomor jaringan situs. *Boarder router* di kedua ujung *tunnel* 6to4 harus mendukung kedua protokol IPv4 dan IPv6.

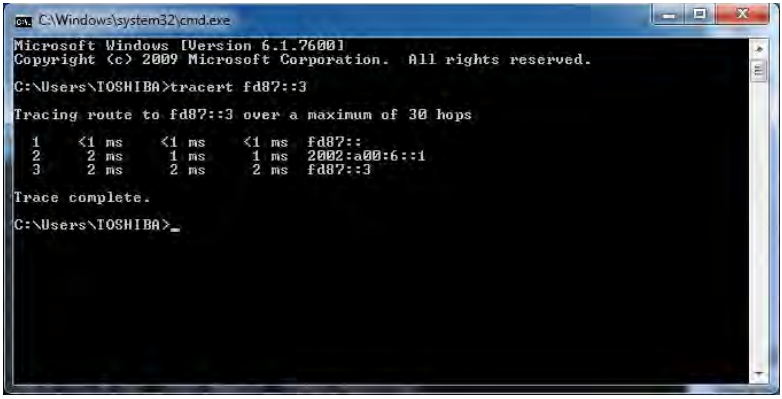
4.1.6 Analisa Pengujian Interkoneksi Sistem *Tunnel* 6to4

Metode pengujian interkoneksi dari jaringan *tunnel* 6to4 sama dengan pengujian pada jaringan *tunnel* manual. Seperti halnya yang dilakukan sebelumnya, setelah dipastikan terjalinnya interkoneksi jaringan IPv4 dilakukan ping dari *host1* ke *host2* untuk memastikan interkoneksi IPv6. Gambar 4.5 menunjukkan hasil ping dari *host1* ke *host2*.



Gambar 4.5 Ping Antar *Host* pada Jaringan *Tunnel* 6to4

Dari Gambar 4.5 terlihat bahwa *host1* menerima balasan dari *host2*, hal ini menandakan bahwa kedua *host* sudah dapat saling berkomunikasi. Selanjutnya dilakukan tracert untuk melihat rute yang ditempuh *host1* untuk menghubungi *host2*. Gambar 4.6 menunjukkan hasil dari tracert *host1* ke *host 2*.



Gambar 4.6 Tracert Jaringan *Tunnel* 6to4

Dari Gambar 4.6 terlihat bahwa rute yang dilalui paket untuk mencapai *host*2 adalah melalui gateway (fd87::) kemudian *tunnel* 6to4

(2002:a00:6::1), dan akhirnya sampai ke *host2* (fd87::3). Hal ini menunjukkan terjalinnya koneksi antar kedua *host* melalui *tunnel* 6to4.

4.2 Analisa Enkapsulasi Paket

Metode *tunneling* dalam melewatkan paket melalui jaringan IPv4 menggunakan cara enkapsulasi paket dimana paket IPv6 yang hendak dikirimkan dienkapsulasi sebagai payload dengan header IPv4. Pada tugas akhir ini pengamatan proses enkapsulasi IPv6 adalah dengan cara menangkap paket-paket yang melewati router R2, hal ini dilakukan dengan cara mengkonfigurasi *monitor capture buffer* dan *monitor capture point* pada router R2 dan kemudian menyalin hasil *capture* tersebut ke TFTP server, hasil *capture* inilah yang akan dianalisa.

Perintah yang digunakan adalah sebagai berikut :

```
R2#monitor capture buffer MYBUFFER size 512 max-size 1024
circular
R2#monitor capture point ip cef FA0_1 f0/1 both
R2#monitor capture point associate FA0_1 MYBUFFER
R2#monitor capture point start
```

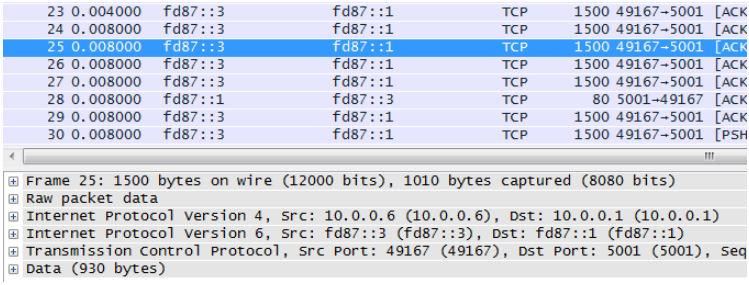
Apabila data telah selesai dikirimkan proses penangkapan paket dihentikan dan hasil yang telah tersimpan pada *buffer* dikirimkan ke TFTP server dengan menggunakan perintah berikut

```
R2#monitor capture point stop
R2#monitor capture buffer MYBUFFER export tftp://10.0.0.1/
mycapture.pcap
```

Hal ini dilakukan secara terpisah dengan proses pengambilan data karena akan menambah pekerjaan router yang berdampak pada menurunnya performansi router R2.

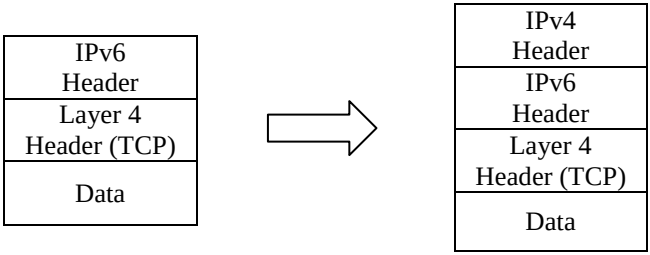
Fitur ini didukung oleh cisco IOS versi 12.4 (20)T atau yang lebih baru, karena terbatasnya DRAM pada router maka digunakan IOS versi c2801-advipservicesk9-mz.124-20.T.bin yang memerlukan DRAM lebih kecil dibandingkan dengan versi adventerprise dan ipservice.

Hasil penangkapan pada buffer akan tersimpan dengan nama “mycapture” dan dalam format pcap. Untuk mempermudah analisa dapat dibuka dengan menggunakan *software* pihak ketiga seperti wireshark. Contoh hasil penangkapan paket dapat dilihat pada Gambar 4.7.



Gambar 4.7 Hasil Penangkapan Paket

Dari Gambar 4.7 terlihat bahwa bahwa paket IPv6 telah dienkapsulasi dalam header IPv4, lebih jelasnya diilustrasikan dalam Gambar 4.8.



Gambar 4.8 Proses Enkapsulasi

Setelah paket dienkapsulasi, barulah paket IPv6 tersebut dapat dilewatkan ke jaringan IPv4. Ketika paket telah sampai pada ujung *tunnel* tujuan maka dilakukan dekapsulasi dimana header IPv4 yang diberikan saat enkapsulasi dihilangkan sebelum paket tersebut diteruskan ke jaringan IPv6.

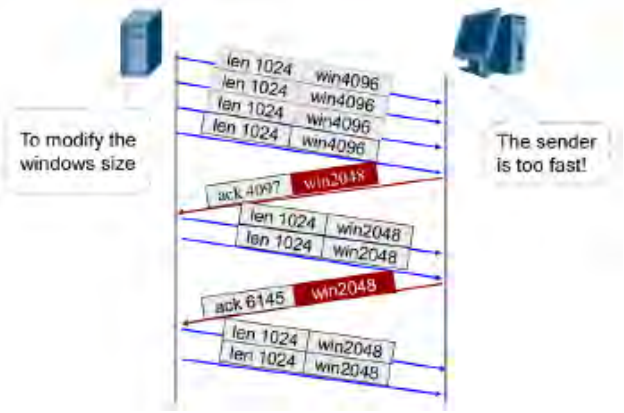
4.3 Analisa Pengukuran Performansi pada *Local Area Network*

Untuk mengetahui unjuk kerja jaringan dilakukan pengukuran beberapa parameter sebagai acuan baik tidaknya kinerja jaringan. Pada tugas akhir ini performansi jaringan didasarkan pada *bandwidth*, *jitter*, *packet loss*, dan RTT. Hasil yang diperoleh akan dibandingakan dengan parameter-parameter yang diperoleh pada jaringan *native-IPv6*.

4.3.1 Pengukuran *Bandwidth*

Bandwidth adalah besaran yang menunjukkan seberapa banyak data yang dapat dilewatkan dalam koneksi melalui sebuah jaringan. *Bandwidth* atau kapasitas saluran informasi merupakan kemampuan maksimum dari suatu alat untuk menyalurkan informasi per satuan waktu.

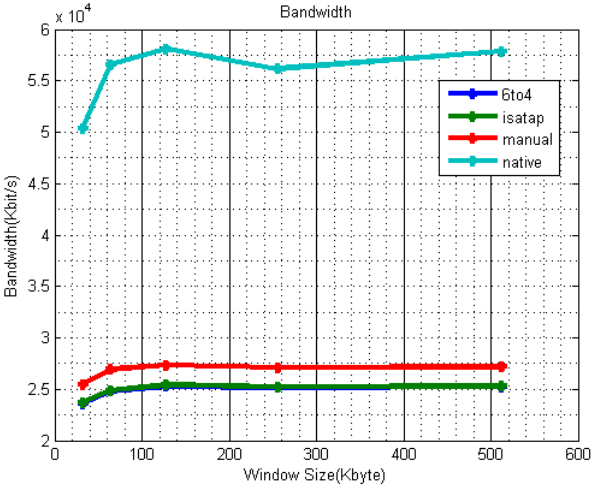
Teknik TCP *Sliding Window* mampu mengatur *data flow* antara dua *host* dengan cara mengubah ukuran *window* secara dinamis. Semua TCP/IP *host* mendukung transmisi *data full-duplex*. Seperti pada Gambar 4.9 menunjukkan contoh bagaimana *Sliding Window* memperoleh *flow control*.



Gambar 4.9 *Sliding Window* [2]

Server mengirimkan empat buah segmen 1024byte ke *client* dan ukuran *window* pada pengirim adalah 4096byte. Penerima akan mengirimkan paket ACK4097 untuk memberitahu bahwa paket telah diterima dan memodifikasi ukuran *window* menjadi 2048byte. Hal ini berarti penerima hanya memiliki ukuran *buffer* 2048byte sehingga pengirim mengubah kecepatan pengiriman dan mengirimkan segmen 2048byte.

Dari hasil pengambilan data diperoleh rata-rata *bandwidth* IPv6 pada jaringan *tunnel* manual, ISATAP, 6to4, dan *native-IPv6*. Data tersebut dapat dilihat pada Gambar 4.10 dan Tabel 4.1



Gambar 4.10 Hubungan *Bandwidth* Terhadap *Window*

Tabel 4.1 Tabel *Bandwidth*

Window	6to4 (kbit/s)	ISATAP (kbit/s)	Manual (kbit/s)	Native (kbit/s)
32Kbyte	23567,8	23685,1	25460,9	50406
64Kbyte	24808,6	24871,8	26925,5	56506,3
128Kbyte	25305	25391,9	27318,5	58079,9
256Kbyte	25190,5	25235,4	27117,3	56151,7
512Kbyte	25231	25292,9	27134,7	57891

Nilai yang diperoleh pada jaringan tunnel ISATAP dan 6to4 sangat dekat sehingga nilai dari kedua tunnel tersebut terlihat saling berhimpit pada Gambar 4.10.

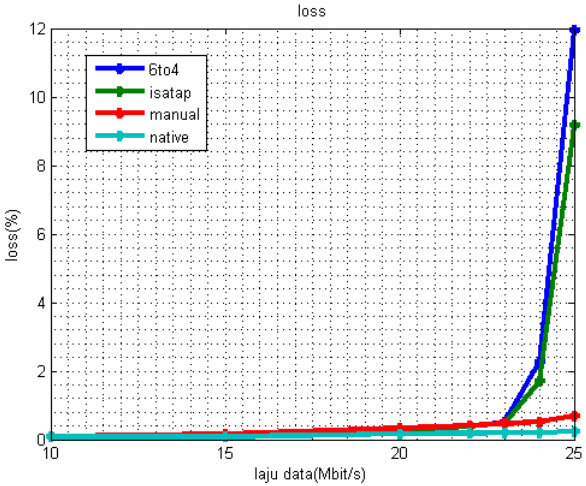
Dapat terlihat bahwa *bandwidth* ketiga jaringan *tunnel* lebih kecil dibandingkan dengan jaringan *native-IPv6*. *Bandwidth* maksimum yang diperoleh pada jaringan *native-IPv6* adalah 58079,9kbit/s sedangkan untuk jaringan *tunnel* manual, ISATAP, dan 6to4 masing-masing adalah 27318,5kbit/s, 25391,9kbit/s, 25305kbit/s. Proses enkapsulasi dan dekapsulasi pada ketiga jaringan *tunnel* menyebabkan *bandwidth* yang diperoleh menjadi lebih rendah dibandingkan jaringan *native-IPv6* yang

tidak perlu melakukan proses enkapsulasi dan dekapsulasi paket IPv6 ke dalam header IPv4 saat melewati router *dual-stack*. Hal ini berarti jaringan *tunnel* akan lebih lambat dalam transmisi data dibandingkan dengan jaringan *native-IPv6*.

4.3.2 Pengukuran *Packet Loss*

Pada transmisi yang bersifat reliable, packet loss dapat menyebabkan peningkatan latency akibat penambahan waktu yang dibutuhkan untuk melakukan transmisi ulang. Hal ini menyebabkan perbedaan latency antara paket yang di-drop dan paket yang tidak sehingga terjadi jitter. Pada transmisi yang bersifat tidak reliable, packet loss menyebabkan error pada data yang diterima sehingga dapat menyebabkan data yang diterima tidak dapat dibaca oleh perangkat atau tidak dapat ditampilkan dengan benar.

Dari hasil pengambilan data diperoleh rata-rata *packet loss* IPv6 pada jaringan *tunnel* manual, ISATAP, 6to4, dan *native-IPv6*. Data tersebut dapat dilihat pada Gambar 4.11 dan Tabel 4.2.



Gambar 4.11 Gambar Hubungan *Packet Loss* Terhadap Laju Data

Tabel 4.2 Tabel *Packet Loss*

Laju Data Sender	6to4 (%)	ISATAP (%)	Manual (%)	Native (%)
10Mbit/s	0,0953	0,087	0,0997	0,0751
15Mbit/s	0,1242	0,1134	0,1464	0,0845
20Mbit/s	0,2774	0,2725	0,3181	0,1640
22Mbit/s	0,3816	0,3770	0,4201	0,1962
23Mbit/s	0,5022	0,4739	0,4711	0,2009
24Mbit/s	2,2799	1,7041	0,5199	0,2015
25Mbit/s	11,9455	9,1619	0,6963	0,2383

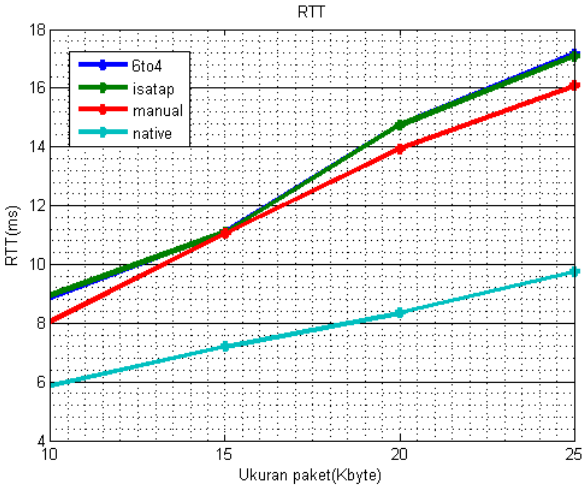
Nilai yang diperoleh pada ketiga jaringan tunnel sampai laju data 23Mbit/s sangat dekat sehingga nilai dari ketiga tunnel tersebut terlihat saling berhimpit pada Gambar 4.11.

Terlihat bahwa *packet loss* dari ketiga jaringan *tunnel* tersebut lebih besar dibandingkan dengan jaringan *native-IPv6*. *Packet loss* pada jaringan *native-IPv6* dan *tunnel* manual hanya mengalamai peningkatan sedikit saja dari percobaan yang satu ke percobaan berikutnya. Sedangkan jaringan ISATAP dan 6to4 mulai mengalami lonjakan packet loss ketika diberikan laju data 25Mbit/s, hal ini disebabkan kedua jaringan ini memiliki *bandwidth* maksimum sekitar 25Mbit/s sehingga apabila dibebankan dengan laju data melebihi *bandwidth* tersebut akan menyebabkan kongesti yang berakibat pada peningkatan *packet-loss* seiring dengan lama pengiriman paket data. Hal ini berarti jaringan tunnel ISATAP dan 6to4 tidak dapat melakukan transmisi dengan laju data 25Mbit/s atau lebih dengan baik.

4.3.3 Pengukuran *Round-Trip Time*

RTT (*Round-Trip Time*) merupakan penjumlahan waktu yang diperlukan sebuah paket untuk terkirim dan waktu yang dibutuhkan sampai balasan bahwa sinyal tersebut sudah sampai diterima oleh sumber. RTT disebut juga waktu ping.

Dari hasil pengambilan data diperoleh rata-rata RTT IPv6 pada jaringan *tunnel* manual, ISATAP, 6to4, dan native-IPv6. Data tersebut dapat dilihat pada Gambar 4.12 dan Tabel 4.3



Gambar 4.12 Gambar Hubungan RTT Terhadap Ukuran Paket

Tabel 4.3 Tabel RTT

Ukuran paket	6to4 (ms)	ISATAP (ms)	Manual (ms)	Native (ms)
10KByte	8,87	8,96	8,06	5,88
15KByte	11,1	11,09	11,06	7,2
20KByte	14,74	14,74	13,93	8,35
25KByte	17,15	17,08	16,08	9,75

Nilai yang diperoleh pada jaringan tunnel ISATAP dan 6to4 sangat dekat sehingga nilai dari kedua tunnel tersebut terlihat saling berhimpit pada Gambar 4.12.

Terlihat bahwa RTT terbesar dialami oleh jaringan *tunnel* 6to4, diikuti oleh jaringan *tunnel* ISATAP, manual, dan *native-IPv6*. Seperti telah dibahas sebelumnya jaringan *tunnel* melakukan proses tambahan berupa enkapsulasi dan dekapsulasi yang tidak dilakukan pada jaringan *native-IPv6* sehingga ketiga jaringan *tunnel* tersebut memiliki waktu RTT yang lebih besar dari jaringan *natiave-IPv6*.

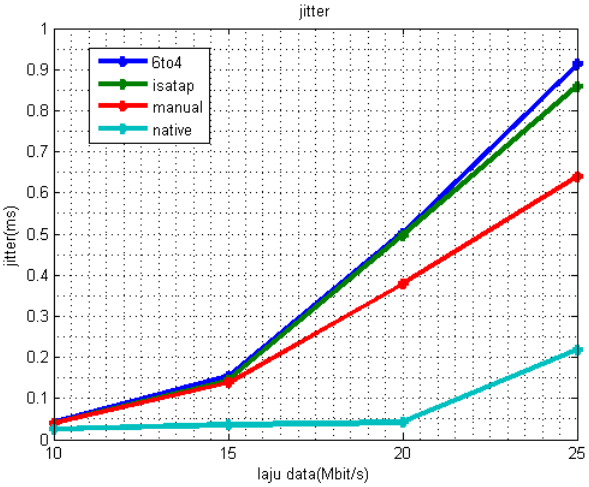
Jaringan *tunnel* manual memiliki waktu RTT yang paling kecil karena bersifat seolah-olah *interface source* dan *destination* dari kedua

router *dual-stack* secara fisik terhubung langsung. Jaringan *tunnel* ISATAP memiliki waktu RTT yang lebih singkat dibanding dengan 6to4 karena format pengalamatannya menyebabkan *interface tunnel* kedua router *dual-stack* berada dalam *network-id* yang sama sehingga tidak memerlukan proses *routing* untuk mencapai router *dual-stack* tujuan terhubung.

4.3.4 Pengukuran *Jitter*

Jitter merupakan variasi interval waktu pengiriman dan penerimaan paket yang terjadi pada jaringan. Jitter dapat menyebabkan tampilan pada layar monitor berkedip atau terlihat tersedak-sedak serta dapat menyebabkan efek sinyal audio yang tidak diinginkan sehingga mempengaruhi kualitas layanan.

Dari hasil pengambilan data diperoleh rata-rata *jitter* IPv6 pada jaringan *tunnel* manual, ISATAP, 6to4, dan native-IPv6. Data tersebut dapat dilihat pada Gambar 4.13 dan Tabel 4.4



Gambar 4.13 Gambar Hubungan *Jitter* Terhadap Laju Data

Tabel 4.4 Tabel *Jitter*

Laju Data Sender	6to4 (ms)	ISATAP (ms)	Manual (ms)	Native (ms)
10Mbit/s	0,0404	0,0402	0,0391	0,0259
15Mbit/s	0,1535	0,1438	0,1381	0,0354
20Mbit/s	0,5017	0,4845	0,4845	0,0421
25Mbit/s	0,9128	0,8591	0,6395	0,2175

Nilai yang diperoleh pada jaringan tunnel ISATAP dan 6to4 sangat dekat sehingga nilai dari kedua tunnel tersebut terlihat saling berhimpit pada Gambar 4.13.

Dari Gambar 4.13 terlihat bahwa ketiga metode *tunnel* mengalami *jitter* yang lebih besar dibandingkan dengan *jitter* pada jaringan *native-IPv6*, namun *jitter* yang dialami oleh semua jaringan ini masih tergolong kecil karena beban traffic cenderung konstan dan tidak ada perubahan pada rute yang digunakan.

4.4 Proses CPU Router

Untuk mengetahui besarnya utilisasi router dapat dilakukan dengan menggunakan perintah "*show cpu process sort*". Perintah ini akan menampilkan informasi tentang proses yang sedang aktif dan statistik utilisasi CPU router.

Pengukuran utilisasi dilakukan dengan pemberian beban UDP menggunakan program iperf dengan ukuran laju data 10, 15, 20, 25 Mbit/s. Kemudian menjalankan perintah *show proccess cpu sort* setelah program iperf telah berjalan melewati 5 detik sehingga muncul tampilan seperti pada Gambar 4.14.

```
CPU utilization for five seconds: 41%/17%; one minute: 9%; five minutes: 10%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
  57      452672    414372       1092 17.61%  3.06%  3.89%   0 IP Input
  27      272692    365254        746  6.04%  1.63%  2.05%   0 IPv6 Input
   5        2184       289       7557  0.49%  0.09%  0.06%   0 Check heaps
 109         304     27220         11  0.08%  0.04%  0.04%   0 RBSCP Background
 139         128     10418         12  0.08%  0.01%  0.00%   0 MLD
   4           4         1       4000  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   6          16        10       1600  0.00%  0.00%  0.00%   0 Pool Manager
   2          60       547        109  0.00%  0.01%  0.00%   0 Load Meter
   7           0         2          0  0.00%  0.00%  0.00%   0 Timers
  10           0         1          0  0.00%  0.00%  0.00%   0 Crash writer
  11           4        65         61  0.00%  0.00%  0.00%   0 ARP Input
```

Gambar 4.14 Proses CPU Jaringan *Tunnel*

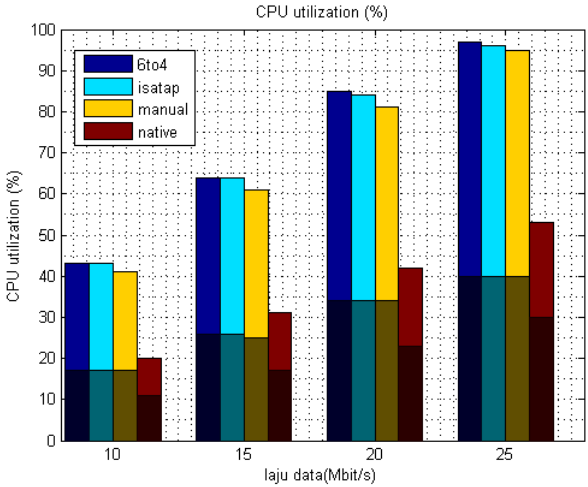
Gambar 4.14 menunjukkan process CPU router ketika diberikan beban paket UDP dengan laju data 10M. Angka pertama, 41%, merupakan utilisasi CPU total untuk semua proses sistem yang aktif selama 5 detik terakhir. Sedangkan angka yang kedua, 17% menunjukkan persentase waktu pada level *interrupt* selama 5 detik terakhir. Interupsi adalah sinyal yang dipancarkan oleh perangkat keras atau perangkat lunak ke prosesor untuk menunjukkan suatu peristiwa yang membutuhkan perhatian segera. Suatu interupsi memberi tahukan prosesor ketika terjadi kondisi dengan prioritas tinggi yang memerlukan interupsi terhadap proses yang sedang dieksekusi oleh *processor*.

```
CPU utilization for five seconds: 20%/11%; one minute: 3%; five minutes: 0%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
  27        1880      5954        315  8.92%  1.47%  0.31%   0 IPv6 Input
 139           0       851          0  0.08%  0.01%  0.00%   0 MLD
   3         144       230        626  0.08%  0.01%  0.01%   0 Exec
 109           0      2758          0  0.08%  0.04%  0.01%   0 RBSCP Background
   2           0        57          0  0.00%  0.01%  0.00%   0 Load Meter
   4           4         1       4000  0.00%  0.00%  0.00%   0 EDDRI_MAIN
   5         196        28       7000  0.00%  0.06%  0.03%   0 Check heaps
   8           0         1          0  0.00%  0.00%  0.00%   0 OIR Handler
   6           0         1          0  0.00%  0.00%  0.00%   0 Pool Manager
  10           0         1          0  0.00%  0.00%  0.00%   0 Crash writer
  11           0        16          0  0.00%  0.00%  0.00%   0 ARP Input
```

Gambar 4.15 Proses CPU Jaringan *Native-IPv6*

Dari Gambar 4.14 terlihat bahwa process yang memerlukan utilisasi CPU terbesar adalah "*IP Input*" dan "*IPv6 Input*". *IP input* menunjukkan *process-switched* pada paket IPv4, sedangkan *IPv6 input* berarti *process-switched* pada IPv6. Sedangkan Gambar 4.15 menunjukkan process yang terjadi pada router pada jaringan *native-IPv6*. Dari Gambar 4.15 terlihat bahwa process yang terjadi sebagian besar hanyalah "*IPv6 Input*", hal ini dikarenakan tidak terdapat process enkapsulasi seperti yang terjadi pada router *dual-stack* pada jaringan *tunnel*. Hal ini menunjukkan bahwa enkapsulasi/dekapsulasi header IPv4 merupakan process yang bersifat *resource intensive* sehingga membebankan router dan menurunkan unjuk kerjanya.

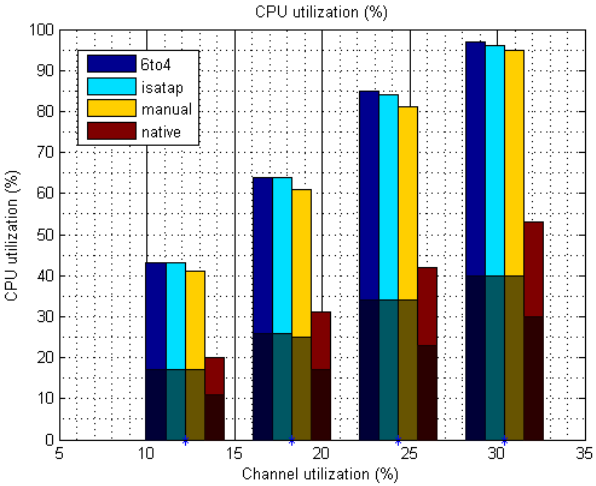
Ketika router diberikan laju data 10M, 15M, 20M, dan 25M maka kinerja routerpun akan menjadi semakin berat, seperti yang dapat dilihat pada Gambar 4.16 dimana warna yang cerah menunjukkan persentasi utilisasi CPU total sedangkan warna yang gelap manunjukkan persentasi *interrupt*.



Gambar 4.16 Utilisasi CPU Jaringan *Tunnel*

Dari Gambar 4.16 terlihat bahwa ketika diberikan laju data sebesar 25M, utilisasi router pada jaringan *tunnel* sudah mendekati 100% sehingga dapat disimpulkan pada jaringan *tunnel*, router tidak dapat memproses paket dengan baik apabila dibebankan dengan laju data melebihi 25M.

Dilakukan pengukuran terhadap utilisasi kanal, hal ini dilakukan dengan mengukur *bandwidth* pada semua kabel ethernet yang digunakan dan diambil nilai terkecil yang diperoleh sebagai *bandwidth* kanal. Diperoleh *bandwidth* kanal sebesar 82129Kbit/s sehingga diperoleh hubungan antara utilisasi kanal dengan utilisasi CPU seperti pada Gambar 4.17 dimana utilisasi CPU pada jaringan tunnel sudah mendekati 100% ketika utilisasi kanal masih mencapai 30,4%



Gambar 4.17 Hubungan Utilisasi Kanal dengan Utilisasi CPU

DAFTAR PUSTAKA

[1] B. Hill, "*Cisco - The Complete Reference*". Brandon A. Nordin, California, 2002.

[2] L. Huawei Technologies Co., "*Huawei Networking Technology and Device*". Huawei Technologies Co., Ltd, Jakarta, 2012.

[3] S. Brown, B. Browne, and N. Chen, "*Configuring IPv6 for Cisco IOS*". Syngress, Rockland, 2002.

[4] S. Hagen, "*IPv6 Essentials*", 3rd ed. O'REILLY, Sebastopol, 2014.

[5] T. Lammle, "*CCNA Routing and Switching Study Guide*". Sybex, Indianapolis, 2013.

[6] T. Russell, "*Telecommunications Protocol*". McGraw-Hill, New York , 1997.

[7] C. Jiann-Liang, "Performance Investigation of IPv4/IPv6 Transition Mechanisms," IEEE, vol. 2, pp. 545-550, Pebruari, 2004.

[8] Y. Wu and X. Zhou, "Research on the IPv6 Performance Analysis Based on Dual-Protocol Stack and Tunnel Transition," ICCSE, pp. 1091-1093, Agustus, 2011.

[9] Telkomspeedy, "Teknologi Internet". <URL: http://opensource.telkomspeedy.com/wiki/index.php/Teknologi_Internet>, Maret, 2011.

[10] RFC 791 - DARPA Internet Program Protocol Spesification.

[11] RFC 2460 - Internet Protocol, Version 6 (IPv6) Specification.

[12] RFC 5214 - Intra-Site Automatic Tunnel Addressing Protocol (ISATAP).

[13] RFC 6343 - Advisory Guidelines for 6to4 Development.

[14] Behrouze A. Forouzan, “Data Communication and Networking”. Mc.Graw Hill, New York, 2007.

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Wahyu Narendra Jati dilahirkan di Kota Tanggerang pada tanggal 14 September 1992 namun dibesarkan di Kota Jakarta, merupakan anak kedua dari tiga bersaudara. Penulis menempuh pendidikan formal di SDN Cipinang 01 Pagi pada tahun 1998, kemudian penullis melanjutkan pendidikan di SMP Labschool Rawamangun pada tahun 2004, dan dilanjutkan ke SMAN 68 pada tahun 2007 sampai tamat pada tahun 2010. Setelah tamat dari SMA, penulis meneruskan pendidikan di Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan mengambil Bidang Studi Telekomunikasi Multimedia. Selama masa perkuliahan penulis aktif dalam kegiatan lab berupa menjadi koordinator asisten praktikum laboratorium komunikasi data, asisten praktikum laboratorium dasar sistem telekomunikasi, dan asisten praktikum laboratorium pengolahan sinyal komunikasi.

BAB V
KESIMPULAN DAN SARAN

Setelah rangkaian penelitian yang telah dilakukan dianalisa maka akan dapat ditarik kesimpulan. Pembahasan dari bab-bab sebelumnya dan kendala-kendala yang terjadi selama pengerjaan tugas akhir ini akan menjadi bahan pertimbangan atau referensi dalam melakukan penelitian pengembangan dari penelitian ini atau penelitian setopik.

5.1 Kesimpulan

Berdasarkan hasil penelitian untuk mengimplementasikan jaringan *tunnel* manual, isatap, dan 6to4 pada perangkat router cisco seri c2801 dan dari perbandingan hasil pengukuran unjuk kerja ketiga sistem tersebut terhadap jaringan *native*-IPv6 dapat disimpulkan :

1. Tidak semua IOS router cisco mendukung fitur IPv6 dan IPv6 *tunnel* sehingga terlebih dahulu perlu diperiksa versi IOS apa saja yang sedang berada di flash router dan besar DRAM router.
2. Format pengalamatan border router *tunnel* 6to4 adalah 2002:border-router-IPv4-address::/48 sehingga dapat digunakan untuk menghubungkan jaringan-jaringan IPv6 pada situs (*network-id*) yang berbeda melewati jaringan IPv4.
3. Format pengalamatan border router *tunnel* isatap adalah ::5EFE:border-router-IPv4-address sehingga dapat digunakan untuk menghubungkan jaringan-jaringan IPv6 pada situs (*network-id*) yang sama melewati jaringan IPv4.
4. *Tunnel* manual tidak menggunakan format pengalamatan tertentu namun suatu *tunnel-id* hanya dapat memiliki satu *tunnel destination.*
5. *Bandwidth* ketiga jaringan tunnel leih kecil dari jaringan *native-IPv6*. Jaringan *tunnel* manual, isatap, dan 6to4 menurun sebesar 52,963%, 56,281%, dan 56,429% dibandingkan jaringan *native-IPv6*.
6. *Packet loss* yang terjadi pada ketiga jaringan *tunnel* lebih besar dari jaringan *native-IPv6*. Packet loss pada jaringan tunnel manual, ISATAP, dan 6to4 meningkat sebesar 0,458%, 8,9236%, dan 11,7072% dibandingkan jaringan *native-IPv6*.

7. RTT dari ketiga jaringan tunnel lebih besar dari jaringan native-IPv6. RTT pada jaringan tunnel manual, ISATAP, dan 6to4 meningkat sebesar 164,923%, 175,179%, dan 175,897% dibandingkan jaringan *native-IPv6*.
8. Jitter dari Ketiga jaringan tunnel lebih besar dari jaringan native-IPv6. Jitter pada jaringan tunnel manual, ISATAP, dan 6to4 meningkat sebesar 294,023%, 394,988%, dan 419,678% dibandingkan jaringan native-IPv6, namun jitter dari ketiga jaringan tunnel tersebut masih dibawah 1ms.
9. Proses enkapsulasi dan dekapsulasi bersifat *resource intensive* sehingga membebankan kinerja router yang berdampak pada bertambahnya packet loss, RTT, jitter dan menurunnya bandwidth jaringan *tunnel*.
10. Diantara ketiga jenis *tunnel* yang diuji, *tunnel* manual memiliki unjuk kerja terbaik.

5.2 Saran

Untuk mendapatkan unjuk kerja jaringan yang lebih baik sebaiknya menggunakan versi IOS terbaru, pada tugas akhir ini hal tersebut tidak dapat dilakukan karena keterbatasan DRAM sehingga untuk penelitian selanjutnya disarankan menggunakan ukuran DRAM yang lebih besar dan versi IOS terbaru.

Tugas akhir ini meneliti unjuk kerja jaringan *tunnel* manual, ISATAP, dan 6to4 secara umum, sehingga belum membahas mengenai pengaruh pengimplementasian *tunnel* pada hal-hal yang lebih spesifik seperti VPN, VOIP, dll.

LAMPIRAN A
PROPOSAL TUGAS AKHIR

Jurusan Teknik Elektro
Fakultas Teknologi Industri – ITS

TE091399 TUGAS AKHIR – 4 SKS

27 FEB 2014

Nama Mahasiswa : Wahyu Narendra Jati
Nomor Pokok : 2210 100 012
Bidang Studi : Teknik Sistem Telekomunikasi dan Multimedia
Tugas Diberikan : Semester Genap Th. 2013/2014
Judul Tugas Akhir : **Analisa Unjuk Kerja pada Metode *Tunneling Manual, 6to4, dan ISATAP* pada IPv4/IPv6**
(Performace Analysis of Manual, 6to4, and ISATAP Tunneling Methode On IPv4/IPv6 Networks)

Uraian Tugas Akhir :

Internet Protocol (IP) merupakan sistem pengalamatan yang digunakan untuk mengidentifikasi tiap perangkat dalam jaringan internet. Sejak dikeluarkan pada tahun 1980an IPv4 telah menjalankan fungsinya dengan baik, namun saat ini mengalami masalah dimana jumlah alamat yang belum teralokasikan pada IPv4 sudah habis. Untuk menghadapi masalah ini dalam IETF (*Internet Engineering Task Force*) diperkenalkan IPv6 yang menggunakan pengalamatan 128-bit daripada 32-bit seperti pada IPv4 [2].

Investasi pada IPv4 saat ini sudah sangat besar karena sebagian besar jaringan internet berbasis IPv4 sehingga cara transit internet dari IPv4 ke IPv6 menjadi topik yang penting. Tugas akhir ini membahas dan membandingkan unjuk kerja dari mekanisme transisi IPv6 menggunakan tunneling serta membandingkannya dengan IPv4 yang ada saat ini.

Terdapat berbagai macam mekanisme tunneling IPv6, dalam tugas akhir ini mekanisme yang dibahas adalah:

1. Manual
2. 6to4
3. ISATAP

Kata Kunci : *IPv6, Manual Tunneling, 6to4, ISATAP*

Dosen Pembimbing I,

Dr. Ir. Achmad Affandi, DEA
NIP. 196510141990021001

Dosen Pembimbing II,

Ir. Djoko Suprajitno Rahardjo, MT
NIP. 195506221987011001

Mengetahui,
Jurusan Teknik Elektro FTI – ITS
Ketua,

Dr. Tri Arief Sardjono, ST., MT.
NIP. 197002121995121001

Menyetujui,
Bidang Studi Telekomunikasi Multimedia
Koordinator,

27/2/2014

Dr. Ir. Endroyono, DEA
NIP. 196504041991021001

USULAN TUGAS AKHIR

A. JUDUL TUGAS AKHIR

"Analisa Unjuk Kerja pada Metode *Tunneling Manual, 6to4, dan ISATAP* pada IPv4/IPv6"

B. RUANG LINGKUP

1. *IPv6*
2. *Manual Tunneling*
3. *6to4*
4. *ISATAP*

C. LATAR BELAKANG MASALAH

IPv4 merupakan metode pengalamatan dalam jaringan dan internet yang dikembangkan pada awal 70-an. Perkembangan yang sangat pesat dalam teknologi jaringan menyebabkan kebutuhan akan alamat IP membesar dan pada akhirnya tidak dapat lagi dibendung oleh IPv4 sehingga dikembangkan protokol baru untuk mengingkatkan ruang internet yaitu IPv6. Berbeda dengan IPv4 yang terdiri dari 32 bit, IPv6 terdiri dari 128 bit sehingga secara teori dapat menampung 2^{32} kali jumlah alamat IPv4.

Pengimplementasian IPv6 tidak bisa serentak dan membutuhkan banyak waktu sehingga terdapat suatu masa transisi dimana IPv4 dan IPv6 berjalan bersamaan. Pada masa transisi ini diperlukan teknik-teknik yang dapat diimplementasikan oleh IPv6 untuk dapat kompatibel dengan IPv4, teknik-teknik ini disebut dengan mekanisme transisi. Terdapat tiga macam mekanisme transisi yaitu *Dual Protocol Stack, Tunneling, dan Protocol Translation*.

Mekanisme *Dual Protocol Stack* masih memerlukan IPv4 stack dan infrastruktur *routing* IPv4 dan IPv6 menyebabkan usaha yang besar dalam implementasi, konfigurasi, dan administrasi. Sehingga hanya

merupakan solusi jangka pendek dan tidak cocok untuk jaringan skala besar. Sedangkan metode *Protocol Translation* memiliki kelemahan berupa tidak mendukung beberapa aplikasi jaringan dan terdapat masalah keamanan seperti tidak dapat melakukan enkripsi *end-to-end* IPSEC sehingga mudah terkena serangan DoS pada *translation gateway*. Sehingga umumnya digunakan metode transisi *tunneling* [1].

Tugas akhir ini disusun untuk mengetahui kinerja dari beberapa metode transisi *tunneling* IPv4/IPv6 yaitu secara manual, 6to4, dan ISATAP.

D. PERUMUSAN MASALAH

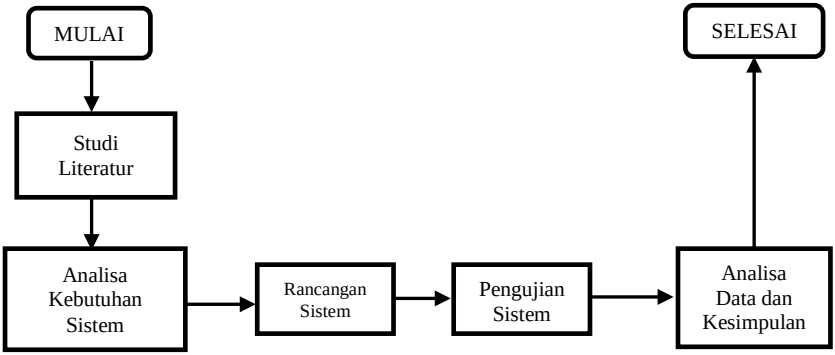
Permasalahan yang dihadapi adalah IPv4 sudah tidak mampu menampung jumlah pengguna internet, sehingga harus beralih ke IPv6. Karena besarnya investasi pada IPv4 maka metode tunneling merupakan metode yang menjanjikan, sehingga pemilihan metode tunneling merupakan hal yang perlu diperhatikan sehingga diperoleh jaringan yang optimal. Dengan permasalahan tersebut dibuatlah tugas akhir ini.

E. TUJUAN

1. Mengevaluasi kinerja interkoneksi antara jaringan IPv4 dan jaringan IPv6 atau sebaliknya dengan metode *tunneling* manual, 6to4, dan ISATAP
2. Membandingkan hasil evaluasi kinerja interkoneksi tersebut dengan kinerja koneksi IPv4-IPv4

F. METODOLOGI

Penelitian dilakukan dengan lima tahap yaitu: studi literatur, analisa kebutuhan sistem, rancangan sistem, pengujian sistem, dan analisa kinerja



Studi literatur dilakukan dengan mencari dan mempelajari beberapa paper dan jurnal. Pada tahap ini akan dipelajari cara kerja serta konfigurasi ketiga macam tunneling tersebut.

Pada tahap selanjutnya, dilakukan analisa kebutuhan sistem. Tahap ini bertujuan untuk memperoleh software atau hardware apa saja yang diperlukan untuk mendukung pengujian serta pengambilan data dari sistem. Untuk melakukan pengujian disusun sistem uji yang terdiri dari dua buah PC dan satu buah router. Perangkat lunak yang digunakan adalah:

- Iperf, digunakan untuk membangkitkan traffic baik TCP maupun UDP dan iperf juga memiliki kemampuan untuk mengukur bandwidth, delay jitter, dan datagram loss.
- Ping/Ping6, merupakan utilisasi yang dapat digunakan untuk menguji keterhubungan dua komputer dalam satu jaringan. Hal ini dilakukan dengan mengirim sebuah paket ICMP berupa Echo Request kepada alamat IP dan menunggu Echo Replay darinya.
- Iostat, merupakan aplikasi yang memberikan laporan statistik tentang penggunaan CPU dan juga penggunaan harddisk berdasarkan masing-masing partisi
- Dig, merupakan aplikasi yang melakukan permintaan alamat IP kepada server DNS dengan mengirimkan nama host dan menampilkan jawaban yang diberikan server DNS

- Ethereal, merupakan aplikasi penangkap informasi tentang paket-paket yang berlalu-lalang dalam jaringan
- Gawk, merupakan perangkat lunak yang dapat melakukan parsing terhadap suatu file. Fungsi utamanya adalah mencari isi dari sebuah file secara baris-per-baris yang berisikan pola tertentu dan kemudian melakukan seleksi atau pemformatan ulang terhadap file tersebut
- Gnuplot, merupakan perangkat lunak pembuat grafik berbasis command-line.

Kemudian tahap selanjutnya adalah rancangan sistem, terdapat dua jenis sistem pengujian, sistem pertama merupakan koneksi antar jaringan IPv4 dan sistem kedua merupakan koneksi jaringan IPv4 dengan IPv6. Pada sistem pertama, jaringan terdiri dari dua PC dan satu router dimana kedua PC dikonfigurasi menggunakan pengalamatan IPv4.

sedangkan pada sistem kedua salah satu PC merupakan dikonfigurasi dengan pengalamatan IPv4 dan PC satunya dikonfigurasi dengan pengalamatan IPv6, router digunakan untuk tunneling antar IP.

Setelah sistem selesai dikonfigurasi, maka dilakukan pengujian untuk mengetahui kinerja dari sistem dengan parameter *bandwidth, end-to-end delay, jitter, dan packet loss* [3]. Kemudian data yang diperoleh dianalisa untuk ditarik kesimpulan.

G. BATASAN MASALAH

Hal-hal yang dilakukan pada penelitian ini adalah sebagai berikut:

1. Implementasi interkoneksi menggunakan tunneling manual, 6to4, dan ISATAP
2. Pengukuran kinerja interkoneksi dengan parameter *bandwidth, end-to-end delay, jitter, dan packet loss.*

H. JADWAL PELAKSANAAN

KEGIATAN	MINGGU													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a. Studi Literatur														
b. Analisa Kebutuhan Sistem														
c. Rancangan Sistem														
d. Pengujian Sistem														
e. Analisa I Kesimpulan														
f. Penyusunan Laporan														

I. DAFTAR PUSTAKA

[1] Xiaoxiang Leng, Miao Zhang, Jun Bi, "*Study on High Performance IPv4/IPv6 Transition and Access Service*", Network Research Center, Beijing, China, 2007

[2] Sam Brown, Brian Browne, Neal Chen, Paul J. Fong, Robbie Harrell, Eric Knipp, Bart Saylors, Rob Webber, Edgar Parenti, "*Configuring IPv6 for Cisco IOS*", Syngress, Rockland, 2002

[3] E. Brent Kelly, "*Quality of Service In Internet Protocol (IP) Networks*", Brookline, 2002

LAMPIRAN B
DATA PERCOBAAN

Bandwidth Jaringan *Native-IPv6*

Data ke-	32 Kbyte	64 Kbyte	128 Kbyte	256 Kbyte	512 Kbyte
1	50425	56523	57961	55792	57589
2	50248	56454	58234	56045	57851
3	50195	56567	58171	56200	57619
4	50528	56568	58066	56076	57704
5	50412	56440	58265	56296	58027
6	50425	56484	57961	56244	57956
7	50509	56578	58140	56192	58144
8	50530	56483	57940	56065	57933
9	50415	56442	58035	56242	58206
10	50373	56524	58026	56365	57881

Bandwidth Jaringan *Tunnel* Manual

Data ke-	32 Kbyte	64 Kbyte	128 Kbyte	256 Kbyte	512 Kbyte
1	25864	27029	27339	27045	27077
2	25744	27115	27343	27188	27140
3	24483	26691	27345	27027	27192
4	25445	26910	27387	27161	27194
5	25845	26914	27350	27052	27115
6	25843	27035	27341	27157	27182
7	25001	26920	27268	27167	27098
8	25923	26851	27224	27146	27090
9	25856	26757	27188	27167	27113
10	24605	27033	27400	27063	27146

Bandwidth Jaringan *Tunnel* ISATAP

Data ke-	32 Kbyte	64 Kbyte	128 Kbyte	256 Kbyte	512 Kbyte
1	23942	24875	25393	25209	25283
2	23698	24860	25387	25213	25253

Data ke-	32 Kbyte	64 Kbyte	128 Kbyte	256 Kbyte	512 Kbyte
3	24005	24854	25345	25303	25231
4	23935	24846	25471	25217	25323
5	23003	24982	25351	25238	25200
6	23949	24878	25473	25206	25311
7	24016	24839	25391	25205	25304
8	23467	24860	25366	25265	25409
9	23474	24874	25366	25229	25248
10	23362	24850	25376	25269	25367

Bandwidth Jaringan *Tunnel* 6to4

Data ke-	32 Kbyte	64 Kbyte	128 Kbyte	256 Kbyte	512 Kbyte
1	23087	24763	25359	25201	25103
2	23230	24860	25254	25097	25218
3	23865	24867	25272	25236	25273
4	23834	24749	25258	25223	25224
5	22821	24760	25257	25230	25258
6	23555	24752	25255	25232	25196
7	23921	24861	25264	25114	25294
8	23516	24879	25376	25230	25227
9	23977	24756	25362	25234	25296
10	23872	24839	25393	25108	25221

Packet Loss Jaringan *Native-IPv6*

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
1	0,0737	0,0851	0,1767	0,2086	0,2100	0,2005	0,2459
2	0,0737	0,0831	0,1469	0,1925	0,1905	0,2327	0,2075
3	0,0707	0,084	0,1781	0,1867	0,2061	0,1798	0,2360
4	0,0707	0,0766	0,1925	0,1832	0,1933	0,2249	0,2449
5	0,074	0,0821	0,1701	0,1763	0,2043	0,1696	0,2372
6	0,0715	0,0968	0,1697	0,1844	0,1890	0,2287	0,2348
7	0,0929	0,0812	0,1580	0,1948	0,1948	0,1838	0,2564
8	0,0818	0,078	0,1567	0,1832	0,2080	0,2199	0,2384
9	0,0667	0,0945	0,1484	0,2086	0,2081	0,1746	0,2207

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
10	0,075	0,0836	0,1428	0,1763	0,2043	0,2003	0,2614

Packet Loss Jaringan *Tunnel* Manual

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
1	0,0969	0,1268	0,3602	0,4185	0,4686	0,5467	0,6932
2	0,0979	0,1525	0,3243	0,4061	0,4262	0,5408	0,7701
3	0,0989	0,1578	0,3280	0,4071	0,4623	0,5285	0,6935
4	0,1002	0,1542	0,3186	0,3959	0,4656	0,4936	0,7273
5	0,1000	0,1435	0,3007	0,4729	0,5324	0,4897	0,6917
6	0,0969	0,1578	0,3160	0,4031	0,4292	0,5378	0,6779
7	0,0979	0,1498	0,3131	0,4101	0,4593	0,5315	0,6927
8	0,1111	0,1309	0,2954	0,4155	0,4716	0,5437	0,6557
9	0,0979	0,1384	0,3326	0,4759	0,5294	0,4927	0,6632
10	0,0989	0,1520	0,2916	0,3946	0,4652	0,4936	0,6980

Packet Loss Jaringan *Tunnel* ISATAP

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
1	0,1010	0,0649	0,2574	0,3866	0,4230	2,0998	9,0596
2	0,0949	0,1325	0,2780	0,3777	0,4686	1,1984	9,1727
3	0,0949	0,1246	0,2483	0,3381	0,5071	1,6231	9,3657
4	0,0949	0,1149	0,2907	0,4054	0,5687	2,2158	8,9152
5	0,0960	0,1229	0,2683	0,4121	0,4020	1,3834	8,9035
6	0,0949	0,1112	0,2855	0,3752	0,4205	2,0878	9,3909
7	0,0949	0,1074	0,2768	0,3406	0,5046	1,2104	0,9524
8	0,0949	0,1234	0,2545	0,4146	0,4661	1,6111	0,8888
9	0,0949	0,1191	0,2868	0,4079	0,5712	2,2278	0,9195
10	0,0949	0,1131	0,2791	0,3866	0,4020	1,3834	0,9171

Packet Loss Jaringan *Tunnel* 6to4

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
1	0,0960	0,1377	0,2960	0,3959	0,5254	2,6604	12,915
2	0,0960	0,1255	0,2933	0,4032	0,5434	2,0809	11,991
3	0,0960	0,1169	0,2957	0,3915	0,4584	2,3053	11,971

Data ke-	10 Mbit/s	15 Mbit/s	20 Mbit/s	22 Mbit/s	23 Mbit/s	24 Mbit/s	25 Mbit/s
4	0,0960	0,1331	0,2488	0,3298	0,5292	1,9884	11,537
5	0,0949	0,1096	0,2289	0,3872	0,4543	2,3561	12,646
6	0,0949	0,1306	0,3170	0,3934	0,4426	2,5117	11,493
7	0,0949	0,1229	0,2983	0,3935	0,6512	1,7581	11,345
8	0,0929	0,1329	0,3034	0,4017	0,4211	2,2123	12,157
9	0,0949	0,1117	0,2487	0,3273	0,4498	1,9124	11,553
10	0,0960	0,1207	0,2441	0,3872	0,5427	2,0776	11,842

RTT Jaringan *Native-IPv6*

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	6,04	7,50	8,40	9,59
2	6,03	6,81	8,18	9,85
3	5,80	7,15	8,52	9,53
4	5,58	7,14	8,20	9,88
5	5,96	7,61	8,30	9,87
6	6,03	7,38	8,39	9,77
7	6,17	7,31	8,30	9,80
8	5,73	7,30	8,38	9,94
9	5,69	6,99	8,41	9,69
10	6,00	7,15	8,56	9,86

RTT Jaringan *Tunnel* Manual

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	8,15	11,10	13,72	16,07
2	7,97	11,17	14,03	16,03
3	8,05	11,02	13,82	15,85
4	8,05	11,13	13,96	16,10
5	7,93	11,08	13,93	15,89
6	8,12	10,94	13,90	16,01
7	8,11	10,87	13,86	15,99
8	8,09	11,02	13,91	16,22
9	8,05	10,89	14,01	16,37
10	8,01	11,13	13,98	16,09

RTT Jaringan *Tunnel* ISATAP

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	8,96	10,95	14,83	17,04
2	8,88	11,02	14,82	16,94
3	8,98	11,07	14,80	17,14
4	8,97	11,02	14,71	17,24
5	8,99	11,12	14,67	17,17
6	9,06	10,92	14,76	16,90
7	9,04	11,09	14,61	17,15
8	8,96	10,92	14,69	17,14
9	8,87	11,04	14,78	16,90
10	8,98	11,00	14,73	17,29

RTT Jaringan *Tunnel* 6to4

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	8,82	11,18	14,91	17,25
2	8,84	11,12	14,64	17,22
3	8,93	11,07	14,76	17,05
4	8,85	11,14	14,71	16,98
5	8,98	11,14	14,80	16,95
6	8,93	11,06	14,64	17,00
7	8,86	11,06	14,82	17,20
8	8,87	11,10	14,82	17,15
9	8,79	11,07	14,80	16,94
10	8,85	11,08	14,61	17,20

Jitter Jaringan *Native-IPv6*

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	0,0197	0,0135	0,0936	0,0149
2	0,0015	0,0102	0,0419	0,5395
3	0,0207	0,0211	0,0255	0,0106
4	0,0002	0,2109	0,0484	0,0075
5	0,0465	0,0115	0,0399	0,0388
6	0,0394	0,0146	0,0402	0,5565
7	0,0394	0,0113	0,0231	0,2164
8	0,0473	0,0099	0,0463	0,0041

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
9	0,0407	0,0261	0,0122	0,3528
10	0,0028	0,0243	0,0497	0,4344

Jitter Jaringan *Tunnel* Manual

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	0,0491	0,0224	0,2911	0,7547
2	0,0402	0,0227	0,3181	0,7936
3	0,0414	0,0220	0,1633	0,5356
4	0,0406	0,0222	0,1640	0,7690
5	0,0395	0,6194	0,3250	0,4441
6	0,0504	0,0217	0,2531	0,4505
7	0,0460	0,5992	0,0887	0,7225
8	0,0215	0,0211	1,1894	0,6700
9	0,0410	0,0094	0,6119	0,4637
10	0,0209	0,0210	0,378288	0,7914

Jitter Jaringan *Tunnel* ISATAP

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	0,0209	0,1080	0,7087	0,9881
2	0,0472	0,6029	0,1178	0,8228
3	0,0279	0,0240	0,5716	0,9650
4	0,0495	0,3430	0,7282	0,6311
5	0,0287	0,0222	0,4504	0,8228
6	0,0462	0,0163	0,3536	0,9857
7	0,0400	0,1448	0,2591	0,7449
8	0,0467	0,1135	0,8867	0,9857
9	0,0548	0,0336	0,6907	0,6569
10	0,0400	0,143829	0,1966	0,9881

Jitter Jaringan *Tunnel* 6to4

Data ke-	10 Kbyte	15 Kbyte	20 Kbyte	25 Kbyte
1	0,0214	0,0204	0,7458	1,0255
2	0,0234	0,0395	0,7129	1,0386
3	0,0390	0,2607	0,3614	0,8907
4	0,0361	0,0207	0,4708	0,8292
5	0,0344	0,8687	0,7505	1,0245
6	0,0448	0,2113	0,1105	0,8917
7	0,0540	0,0312	0,3096	0,8476
8	0,0513	0,0268	0,6606	0,8228
9	0,0591	0,0349	0,4976	0,9278
10	0,0403	0,0206	0,3972	0,8292

Halaman ini sengaja dikosongkan