BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Se-Joon Yoon et al (2012) melakukan penelitian yang berjudul *Performance Comparison of 6to4, 6RD, and ISATAP Tunnelling Methods on Real Testbeds*. Penelitian ini dilakukan dengan menerapkan *IPv6* secara ekslusif dan metode *6to4*, *ISATAP*, dan *6RD*. Penelitian ini dilakukan dengan berbasiskan beberapa lingkungan*kernel* Linux yang nantinya akan menjadi parameter pembanding. Mereka melakukan penelitian guna mengukur efisiensi pengiriman data dari setiap metode yang diterapkan. Dari hasil penelitian yang dilakukan metode *ISATAP* menunjukkan kecenderungan tidak efisien dibanding metode lain.

Jun Bi, Jianping Wu, dan Xiaoxiang Leng (2007) melakukan penelitian tentang transisi *IPv4/IPv6*. Penelitian ini berjudul *IPv4/IPv6 Transition Technologies and Univer6 Architecture*. Penelitian ini meliputi pengamatan terhadap mekanisme dasar transisi dan beberapa isu-isu keamanan yang terjadi dalam proses transisi *IPv4/IPv6*. Menurut Jun Bi, Jianping Wu, dan Xiaoxiang Leng perpindahan dari *IPv4* ke *IPv6* tidak akan berjalan cepat dan mudah. Hal ini karena mayoritas infrastruktur internet saat ini menggunakan *IPv4* dan tingkat keamanan yang masih perlu dilakukan kajian lebih dalam terutama pada bagian *firewall* di *IPv6*.

Yao-Chung Chang et al (2004) melakukan penelitian terhadap mekanisme transisi *IPv4/IPv6* dan performa jaringan yang dihasilkan dari pengujian terhadap beberapa mekanisme transisi. Penelitian ini berjudul *Performance Investigation of IPv4/IPv6 Transition Mechanisms*. Yao-Chung Chang et al (2004) berargumen bahwa mekanisme *configured*

*tunneling*merupakan mekanisme yang lebih tepat untuk diterapkan. Hal ini karena mekanisme ini dapat dikontrol secara ketat sehingga menciptakan *QoS* jaringan, *multicast* dan *unicast* yang lebih baik.

Chiranjit Dutta dan Ranjeet Singh (2012) berpendapat mekanisme transisi *dual stack* memiliki *latency* yang paling kecil daripada mekanisme lainnya. Penelitian ini berjudul *Sustainable IPv4 to IPv6 Transition*. Penelitian ini membahas kinerja dari beberapa mekanisme transisi dan peralatan jaringan yang dipakai dalam implementasi mekanisme transisi.

2.2. IP (Internet Protokol)

Protokol Internet (*Internet Protocol*/IP) adalah protokol lapisan jaringan (*network layer* dalam *OSI Reference Model)* atau protokol lapisan *internetwork* (*internetwork layer* dalam *DARPA Reference Model*) yang digunakan oleh protokol TCP/IP untuk melakukan pengalamatan dan *routing* paket data antar *host-host* di jaringan komputer berbasis TCP/IP.Protokol IP merupakan salah satu protokol kunci di dalam kumpulan protokol TCP/IP.

Sebuah paket IP akan membawa data aktual yang dikirimkan melalui jaringan dari satu titik ke titik lainnya. Metode yang digunakan adalah *connectionless* yang berarti tidak diperlukannya membuat dan memelihara sebuah sesi koneksi. Selain itu, protokol ini juga tidak menjamin penyampaian data, tapi hal ini diserahkan kepada protokol pada lapisan yang lebih tinggi (lapisan transport dalam *OSI Reference Model* atau lapisan antar *host* dalam *DARPA Reference Model*), yakni protokol *Transmission Control Protocol* (TCP) (Dye, 2007: 55-56).

2.2.1. Layanan yang Ditawarkan oleh Protokol IP

IP menawarkan layanan sebagai protokol antar jaringan (*inter-network*), karena itulah IP juga sering disebut sebagai protokol yang bersifat *routable*. *Header IP* mengandung informasi yang dibutuhkan untuk menentukan rute paket, yang mencakup alamat IP sumber (*source IP address)* dan alamat IP tujuan (*destination IP address*). Anatomi alamat IP terbagi menjadi dua bagian, yakni alamat jaringan (*network address*) dan alamat *node* (*node address*/*host address*). Penyampaian paket antar jaringan (umumnya disebut sebagai proses *routing*) dimungkinkan karena adanya alamat jaringan tujuan dalam alamat IP. Selain itu, IP juga mengizinkan pembuatan sebuah jaringan yang cukup besar, yang disebut sebagai *IP internetwork*, yang terdiri atas dua atau lebih jaringan yang dihubungkan dengan menggunakan *router* berbasis IP (Dye, 2007: 143-144). IP mendukung banyak protokol klien, karena memang IP merupakan "kurir" pembawa data yang dikirimkan oleh protokol-protokol lapisan yang lebih tinggi dibandingkan dengannya.

Protokol IP dapat membawa beberapa protokol lapisan tinggi yang berbeda-beda, tapi setiap paket IP hanya dapat mengandung data dari satu buah protokol dari banyak protokol tersebut dalam satu waktu. Karena setiap paket dapat membawa satu buah paket dari beberapa paket data, maka harus ada cara yang digunakan untuk mengindikasikan protokol lapisan tinggi dari paket data yang dikirimkan sehingga dapat diteruskan kepada protokol lapisan tinggi yang sesuai pada sisi penerima. Mengingat klien dan *server* selalu menggunakan protokol yang sama untuk sebuah data yang saling dipertukarkan, maka setiap paket tidak harus mengindikasikan sumber dan tujuan yang terpisah. Contoh dari protokol-protokol lapisan yang lebih tinggi dibandingkan IP adalah *Internet Control Management Protocol* (ICMP), *Internet Group ManagementProtocol* (IGMP), *User Datagram Protocol* (UDP), dan *Transmission ControlProtocol* (TCP)(Dye, 2007: 232-233).

2.3. Internet Protocol Version 4 (IPv4)

Pada dasarnya komunikasi data merupakan proses pertukaran data dari satu komputer ke komputer yang lain. Proses komunikasi ini melibatkan sekumpulan protokol. Salah satu protokol yang terlibat dalam proses komunikasi data adalah *internet protocol* (IP). Pada *layer OSI*, protokol IP terletak pada *internet layer* (layer tiga). IP berfungsi untuk menyampaikan paket data ke alamat yang tepat. Saat ini, *IPv4* adalah protokol IP yang dipakai secara luas di jaringan internet untuk menghubungkan banyak jaringan komputer di seluruh dunia (Dye, 2007: 136).

2.3.1. Struktur Pengalamatan IPv4

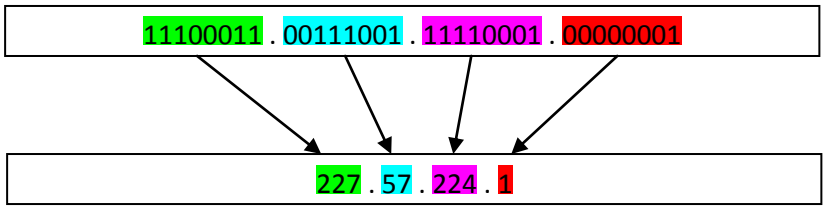
Alamat *IPv4* merupakan bilangan biner 32 bit yang dipisahkan oleh tanda pemisah berupa tanda titik (.) setiap 8 bitnya. Tiap 8 bit ini disebut sebagai oktet. Bentuk alamat *IPv4* adalah sebagai berikut :

xxxxxxxx.xxxxxxxx.xxxxxxxx.xxxxxxxx

Setiap simbol "x" dapat digantikan oleh angka 0 dan 1, misalnya sebagai berikut :

11100011.00111001.11110001.00000001

Untuk lebih mempermudah penulisan dan pembacaan, maka format *IPv4*biasa dituliskan menjadi 4 buah angka desimal yang juga dipisahkan oleh tanda titik. Setiap angka desimal yang juga dipisahkan oleh tanda titik. Setiap angka desimal tersebut merupakan nilai dari satu oktet alamat *IPv4*. Penulisan seperti ini disebut dengan *dotted-decimal notation* (notasi desimal bertitik). Gambar 2.1 berikut memperlihatkan bagaimana sebuah alamat *IPv4*dituliskan dalam notasi desimal bertitik (Dye, 2007: 173-175).



Gambar 2.1 Contoh penulisan notasi dotted-decimal.

Setiap *host* yang terhubung ke internet memilki alamat internet unik sepanjang 32 bit untuk dapat saling berkomunikasi. Setiap alamat yang ada terdiri dari sepasang *network ID&host ID*. *Network ID* mengidentifikasikan jaringan yang digunakan dan *host id* mengidentifikasikan *host* yang terhubung ke jaringan tersebut. Sistem pengalamatan *IPv4* dibagi menjadi beberapa kelas berdasarkan jumlah *network id* dan *host id* yang bisa digunakan:

- Kelas A (1.0.0.0 – 127.255.255.255)
- Kelas B (128.0.0.0 – 191.255.255.255)
- Kelas C (192.0.0.0 – 223.255.255.255)

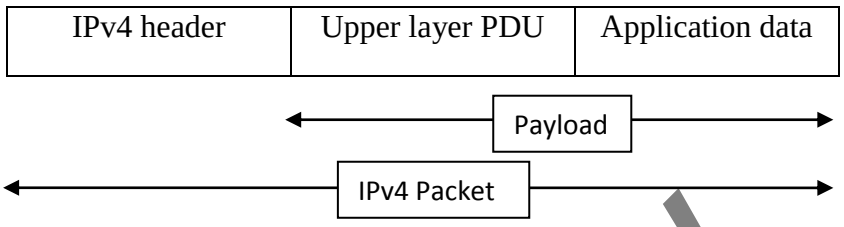
Selain ketiga kelas A, B, dan C yang sering dipakai, terdapat kelas yang lain.

- Kelas D (224.0.0.0–239.255.255.255) digunakan untuk alamat m*ulticast*.
- Kelas E (240.0.0.0–247.255.255.255) digunakan untuk eksperimen (Dye, 2007: 196-197).

2.3.2. Struktur Paket IPv4

Setiap data yang dikirim ke jaringan akan diubah dalam bentuk paket–paket yang kemudian diberi tambahan *header* yang disebut *datagram*. *Datagram* ini berisi tentang berbagai informasi yang digunakan oleh paket selama berada dalam jaringan. Dalam lapisan protokol OSI. Setiap data yang melewati tiap–tiap lapisan akan mengalami enkapsulasi, yaitu penambahan informasi berupa PDU.

PDU atau protokol data unit merupakan kombinasi informasi kontrol bagi suatu lapisan data dari satu lapisan ke lapisan berikutnya yang lebih tinggi atau rendah (*lower-layer/upper layer).Datagram* terdiri dari beberapa *field* yang memilki fungsi tersendiri dan memberikan informasi yang berbeda–beda. Gambar 2.2 berikut menunjukkan struktur *IPv4 header*.



Gambar 2.2 Struktur Paket IPv4

- *Version* (4 bit)

  Mengindikasikan versi IP yang digunakan.Oleh karena ini *IPv4*, maka nilainya diset ke "4".*Size field* ini adalah 4 bit.

- *IHL* (4 bit)

  Internet header length, merupakan panjang *header* internet dengan kelipatan 4 byte.*Header IPv4* memiliki ukuran minimum 20 byte, sedangkan ukuran maksimumnya adalah 60 byte.

- *Type of service* (8 bit)

  Mengindikasikan layanan (*service*) yang diharapkan oleh paket bersangkutan, berisi tentang informasi prioritas, waktu *delay*, keluaran dan karakteristik realibilitas.

- *Total length* (16 bit)

  Mengindikasikan panjang total paket *IPv4* (*IPv4 header* dan *IPv4 payload*).

- *Identification* (16 bit)

  Mengindikasikan nilai yang ditetapkan pengirim paket untuk membantu *reassembly fragmen datagram*.

- *Flags* (3 bit)

  Mengindikasikan *flag–flag* untuk proses fragmentasi. *Size field* ini 3 bit, tetapi hanya 2 bit yang ditetapkan untuk penggunaan ini.

  Bit 0 = reserved, harus diset 0.

  Bit 1 = 0 : bisa difragmentasi, 1 : tidak dapat difragmentasi.

  Bit 2 = 0 : fragmentasi terakhir, 1 : terdapat fragmentasi lagi.

- *Fragment offset* (13 bit)

  Mengindikasikan posisi *fragment relative* terhadap *payload IPv4 original.*

- Ti*me to Live* (8 bit)

  Mengindikasikan jumlah *link* maksimum dimana paket boleh berada di jaringan sebelum dibuang.*TTL* digunakan sebagai hitungan waktu dimana *router IPv4* menentapkan besar waktu yang dibutuhkan (dalam detik) untuk menghantarkan paket.

- *Protocol* (8 bit)

  Mengidentifikasikan *protocol* yang digunakan di layer yang lebih tinggi.Sebagai contoh, TCP menggunakan *Protocol 6*, UDP menggunakan *Protocol 17*, dan ICMP menggunakan *Protocol 1*.

- *Header Checksum* (16 bit)

  Memberikan kemampuan *checksum* (pengecekan *error*), tetapi dalam *header IPv4* saja.*Payload IPv4* tidak dimasukkan dalam kalkulasi *checksum* karena biasanya memiliki *checksum* tersendiri.

- *Source Address* (32 bit)

  Menyimpan I*P address*(*IPv4 address)host* pengirim.

- *Destination address*

  Menyimpan *IP address* (*IPv4 address*) *host* tujuan.

- *Option* (32 bit)

  Menyimpan satu atau lebih opsi *IPv4*. Jika opsi–opsi tersebut tidak genap menggunakan 32 bit maka akan ditambahkan sehingga *header* genap memiliki blok–blok 4 byte, yang dapat dindikasikan oleh *field IHL* (Dye, 2007: 143-144).

2.4. Kekurangan IPv4

InternetProtocol Version4(*IPv4*) yangdidefinisikan olehIETF(http://www.ietf.org) RFC791. RFC791diterbitkanpada tahun 1981. Desain awaldari*IPv4*tidak mengantisipasipertumbuhaninternetdan hal ini menciptakanbanyak masalah, yang terbukti*IPv4*perludiubah. Keterbatasanutama*IPv4*tercantum di bawah ini.

- Keterbatasan alamat IPv4

Sistem pengalamatan*IPv4*menggunakan32-bit *address space. Addressspace*32-bit ini lebih lanjut diklasifikasikandigunakanuntuk kelas A, B, danC. 32-bit *address space*memungkinkan untuk mengalamati 4294967296alamat *IPv4*, tetapipenggunaan *IPv4* selama ini hingga sekarang hampir menyentuh kemampuan *IPv4* untuk mengalamati.Banyakalamatyang dialokasikan untukbanyak perusahaanyangtidak digunakandan hal ini menciptakankelangkaanalamat *IPv4*.

Karenakelangkaanalamat *IPv4*,banyak organisasimenerapkanNAT(*Network Address Translation*) untuk memetakanbeberapaalamat privat ke alamatIP publik. Dengan

menggunakanNAT(*Network Address Translation*) kitadapat memetakanbanyakalamat*IPv4* privat internalke alamat*IPv4*publik, hal ini juga membantudalam melestarikanalamat *IPv4*. TapiNAT(*Network AddressTranslation*) juga memilikibanyak keterbatasan. NAT(*Network AddressTranslation*) tidak mendukungstandar keamananlapisan jaringandantidak mendukungpemetaansemua protokol *upper layer*.NATjuga dapat membuatmasalah jaringanketika duaorganisasiyangmenggunakanrentang alamat*IPv4 private* yang samaberkomunikasi. Lebih banyak *server,workstation* danperangkat yang terhubungke internetjugamenuntutkebutuhan untuk lebihalamat danstatistik saat inimembuktikan bahwapublikruang alamat *IPv4*akan segera habis.Kelangkaanalamat*IPv4*adalah keterbatasanutamasistem pengalamatan*IPv4*.

- Masalah keamanan terkait

Komunikasi *private*melalui mediapublik sepertiinternet membutuhkanlayanankriptografi yangmelindungidata yang dikirim dari kemungkinan untuk dilihatatau diubah oleh sesuatu yang tidak diizinkan dalam transit. Meskipunstandarsekarang ada untukmemberikan keamanan bagipaket *IPv4*, *Internet Protocol Secuurity* (dikenal sebagai*IPsec*), standar iniadalah opsional untuk*IPv4*, dansolusikeamanan tambahan, bukan terintegrasi.

- Masalah konfigurasi alamat

Menurut Joseph Davies pada bukunya yang berjudul *Understanding IPv6* (2012), kebanyakan implementasi*IPv4*saat iniharusdikonfigurasi secara manualatau menggunakan protokol konfigurasi alamat stateful seperti *DynamicHost Configuration Protocol(DHCP)*. Dengan banyaknya pengguna komputer dan peralatan yang menggunakan IP, maka dibutuhkan konfigurasi yang lebih sederhana dan konfigurasi otomatis yang tidak bergantung pada infrastruktur *DHCP*.

- Kebutuhan untuk dukungan yang lebih baik untuk pengiriman data diprioritaskan secara *real-time*

Quality of Service(QoS) tersedia dalam*IPv4*dan hal itu bergantungpada8bit dari bagian *Type of Service*(TOS) danidentifikasi*payloadIPv4*. Bagian *Type of Service*(TOS) IPv4 memiliki fungsi terbatasdan identifikasi*payload*(menggunakan TCPatau UDPport)tidak mungkin dilakukan ketika*payload* paket*IPv4*dienkripsi.

2.5. Internet Protocol Version 6 (IPv6)

Seperti yang telah yang disebutkan, bahwa *IPv6*adalah versi terbaru dari *internet protocol*. IP versi ini didesain sebagai pengganti *IPv4* yang sekarang digunakan. Pada dasarnya *IPv6* dikembangkan untuk mengantisipasi keterbatasan alamat IP yang terdapat pada *IPv4*. Dibandingkan dengan *IPv4* yang memiliki alokasi alamat 32 bit, pada *IPv6* berkembang menjadi 128 bit. *IPv6* memiliki beberapa kelebihan antara lain:

- Alokasi alamat yang besar.
- Penyederhanaan format *header*.
- Infrastruktur *hierarchical addressing* dan *routing* yang efisien.
- Konfigurasi *address stateless* dan *statefull*.
- *Built-in security*.
- Dukungan yang baik untuk *QoS*.
- Protokol baru untuk interaksi *neighboring node* (Vachon, 2008: 490-492).

2.5.1. Fitur Alamat IPv6

IPv6 memiliki beberapa fitur. Adapun fitur-fitur tersebut adalah:

- Format paket dan *header* yang baru

IPv6 menggunakan format paket yang baru. Format paket *IPv6* yang beda dengan format paket *IPv4* inimembantu untuk meminimalkan pengolahan *header* paket oleh *router*. Hal ini bisa terjadi karena bagian *nonessential* dan optional dipindahkan ke *extension headers*yang ditempatkansetelahheader*IPv6*. Sejakpaket *IPv4*dan*IPv6*paketberbeda secara signifikan, dua protokol ini tidak*interoperable*.

- *Space address* yang besar

IPv4 memiliki *space address* 32 bit (4-byte), tapi *IPv6* memiliki *space address* 128-bit (16-byte). *Space address IPv6* yang sangat besar mendukung total 2^{128} (3.4×10^{38}) alamat. *Space address* yang besar ini memberikan alokasi yang lebih baik, sistematis, alokasi hirarkis alamat dan agregasi rute efisien. Dengan jumlah besar alamat yang tersedia kita bisa menghilangkan teknik *address-conservation* seperti *NAT* (*Network Address Translation*).

- Konfigurasi*IPv6 statefull* dan *stateless*

IPv6 bisa dikonfigurasi secara *statefull*atau*stateless*. *Host* pada sebuah *link* dapat dikonfigurasi *IPv6* secara otomatis, ini disebut *link-local address* dan bisa juga dengan alamat yang berasal dari *prefix* yang di-*advertised* dari sebuah *router*. Ketika pertama kaliterhubung ke jaringan,*host*mengirimkanpermintaan sebuah *link-local router* secara *multicast* untuk permohonan konfigurasi sebuah parameter. *Router*yangtersedia dilinkmeresponpermintaandari *host*denganpaket *advertisement router* yangberisi konfigurasi *network layer*. *Host*dapat mengkonfigurasialamat*link-local* secara otomatis dan berkomunikasisatu sama lain tanpakonfigurasi manualbahkan ketikatidak ada *router* yangtersedia.Hostjuga mungkin memilikikonfigurasi*stateful*dengan*Dynamic HostConfigurationProtocol*versi 6(*DHCPv6*) atau konfigurasistatis, sebagai*IPv4*.

- *Multicast*

Ketiga jeniskomunikasi yang tersediadidalam*IPv4* adalah*unicast*, *multicast* dan*broadcast*. *Unicast*adalahsatu-ke-satu komunikasi, multicastadalah satu-ke-banyak komunikasi dan*broadcast* adalahsatu-ke-komunikasi semua. Transmisidari sebuah paketke semua *host*dilakukan dengan menggunakanalamat *broadcast*khusus dalam*IPv4*. Komunikasi*broadcast*tidak tersedia di*IPv6*dan maka dari itu *IPv6*tidak mendefinisikanalamat *broadcast.*Dalam *IPv6*, fungsi broadcast didapatkan dengan mengirimkan sebuah paketke *link-lokal*semua *node*grup *multicast*di alamatFF02::1.

- *Integrated Internet Protocol Security (IPSec)*

InternetProtocol Security(*IPSec*) adalah seperangkatstandar internetyang menggunakanlayanan keamanankriptografi untuk menyediakankerahasiaan, otentikasi, integritas data. Dukungan untuk*Internet ProtocolSecurity*(*IPSec*) dalam*IPv4*bersifat pilihan. *InternetProtocol Security*(*IPSec*) merupakan bagian yang terintegrasidari basisprotokoldi*IPv6*.Dukungan*InternetProtocol Security*(*IPSec*) adalahwajib di*IPv6*.

- *Neighbor Discovery Protocol*

Neighbor Discovery Protocol (NDP) adalah protokol yang tersedia pada *IPv6*. *Neighbor Discovery Protocol* (NDP) didasarkan pada pesan-pesan *Internet Control Message Protocol Version* 6 (ICMPv6) yang mengelola interaksi node pada *link* yang sama. Tidak ada *Address Resolution Protocol* (ARP) untuk *IPv6*dan aturan *Address Resolution Protocol* (ARP), ini diganti dengan *NeighborDiscovery Protocol* (NDP).

- *Extensibility*

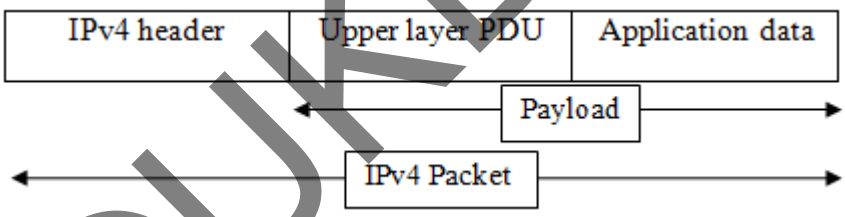
Fitur dari*IPv6*dapat diperpanjangdengan menambahkan*Extension header* setelah*header IPv6*. ukuran *Extension headerIPv6*dibatasihanyaoleh ukurandari paket *IPv6*, tidak seperti 40 bytepilihandari*IPv4.*

- *Jumbograms*

*Jumbograms*merupakan fitur opsionaldari*IPv6*. *Jumbograms*memungkinkan paketdenganmuatan2^32-1(4,294,967,295) byte dengan memanfaatkanpanjanglapangan32-bit.

2.5.2. Struktur Paket IPv6

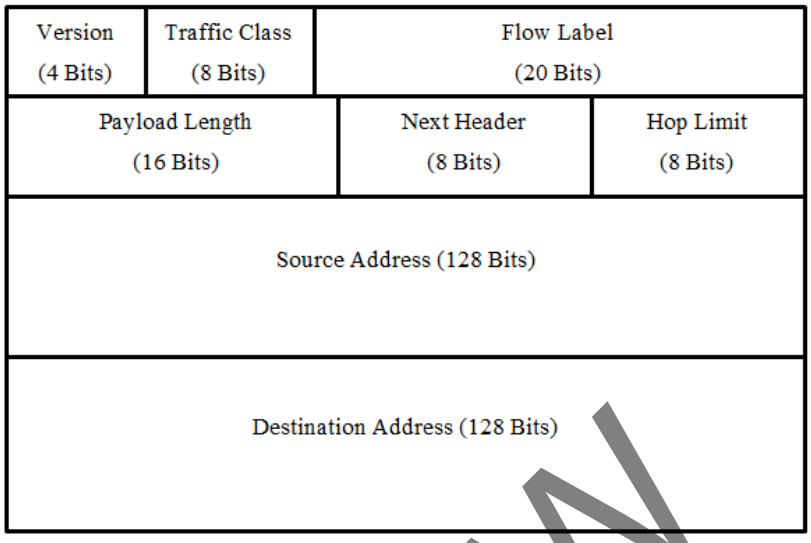
IPv6 memiliki *header* paket jauh lebih sederhana dibandingkan dengan *IPv4*. Hal ini karena *header IPv6* hanya berisi informasi penting untuk meneruskan *IP datagram*. *IPv4* memiliki panjang *header* tetap ukuran 40 byte. *Header IPv6* yang berukuran tetap membolehkan *router* untuk memproses *IPv6 datagram* paket lebih efisien. Berikut gambar 2.3 menunjukkan struktur paket *IPv6.*



Gambar 2.3 Struktur Paket IPv6

*IPv6*datagrampaket *header* bisa dibagimenjaditiga bagian. Ketiga bagian itu adalah*IPv6 datagram packet header*, *Extension Header* dan*Upper Layer Protocol Data*. *IPv6*datagrampaket juga memiliki*Extension Header*.Jika*Extension Header* ada dalam*IPv6*datagrampaket,*Headerfield*berikutnya dalam*headerIPv6*menunjukkan*Extension Header* pertama. Setiap *Extension Header*mengandung*Next Header*lain, ini untuk menunjukheaderekstensiberikutnya. *IPv6* datagram *packet extension header* terakhir menunjukkan upper layer *protocol header* (*Transmission Control Protocol*/TCP), *User*, atau *Internet Control Message Protocol* (*ICMPv6*)). Tidak

ada pilihan dalam *IPv6datagram*paket *header*yang ada dalam*IPv4header*.Berikut gambar 2.4 menunjukkan format *header IPv6.*



Gambar 2.4 Format Header IPv6

Berikut penjelasan elemen-elemen dari *headerIPv6*:

- ***Version*:** UkuranbagianVersiadalah 4 bit. bagianVersimenunjukkanversiIP dandiatur menjadi6.
- ***Traffic Class*:** Ukuran bagian *Traffic Class* adalah 8 bits. *Traffic Class field* sama dengan bagian *IPv4 Type of Service* (*ToS*). Bagian *Traffic Class*menunjukkankelas*IPv6*paketatauprioritas.
- ***Flow Label*:** Ukuranbagian*Flow Label*adalah 20bit. Bagian*Flow Label*memberikan dukungan tambahanuntuk pengirimandatagram secara *real-time*dan fitur *quality of service*.Tujuan daribagian *Flow Label*adalah untukmenunjukkan bahwa paketinitermasuk dalam urutanpaket tertentuantarasumber dan tujuandandapat digunakan untukpengirimanpaketprioritasuntuk seperti layanan suara.
- ***Payload Length*:** Ukuran dari bagian *Payload Length* adalah 16 bits. Bagian *Payload Length* menunjukkan panjang *IPv6 payload*, termasuk *extension headers* dan *upper layer protocol data.*

- ***Next Header*:** Ukuran dari *Next Header* adalah 8 bit. *Next Header* menunjukkan baik jenis perpanjangan pertama (jika ada *header extension* tersedia) atau protokol di lapisan atas seperti *TCP*, *UDP*, atau *ICMPv6*.
- ***Hop Limit*:** Ukuran dari bagian *hop limit* adalah 8 bit. Bagian *hop limit* menunjukkan jumlah maksimum *router* paket *IPv6* dapat melakukan perjalanan. Bagian *hop limit* ini mirip dengan bagian *time to live IPv4* (*TTL*).
- ***Source Address*:** Ukuran bagian *source address* adalah 128 bit. Bagian *Source address* menunjukkan sumber paket alamat *IPv6.*
- ***Destination Address*:**Ukuran bagian *destination address* adalah 128 bit. Bagian *destination address* menunjukkan tujuan paket alamat *IPv6*(Deering, 1998).

2.5.3. Struktur Pengalamatan IPv6

Arsitektur pengalamatan *IPv6* dijelaskan secara formal dalam dokumen RFC 3513. Perbedaan kontras di antara *IPv4* dan *IPv6* adalah panjang alamatnya. Alamat–alamat *IPv6* berukuran jauh lebih besar dibandingkan alamat *IPv4*. Tidak seperti pada *IPv4* yang memiliki panjang *address* 32 bit, panjang alamat *IPv6* adalah 128 bit. Ini berarti alamat *IPv6* memiliki ukuran 4 kali lebih panjang dibandingkan *IPv4.* Hal ini membuat pengalamatan *IPv6* dapat dikatakan lebih kompleks dari pengalamatan *IPv4* yang telah dikenal baik selama ini. Untuk *IPv6,* alamat sebesar 128bit dibagi ke dalam 8 blok 16bit. Masing–masing blok dikonversi ke 4digit heksadesimal dan dipisahkan oleh tanda titik dua ":". Hasil representasi ini dinamakan "*colon-hexadecimal".* Format penulisannya adalah sebagai berikut:

XXXX : XXXX : XXXX : XXXX : XXXX : XXXX : XXXX : XXXX

Setiap simbol X merupakan representasi dari 1-digit heksadesimal (4-bit). Sehingga pada prakteknya, *IPv6 address* ditulis dalam representasi heksadesimal, bukan desimal. Karena ditulis dalam bentuk heksadesimal, address

IPv6 tidak hanya terdiri atas angka–angka numeral (0-9), tetapi juga beberapa huruf *alphabet (*A-F). Huruf–huruf ini tidak bersifat *case-sensitif*, yang berarti huruf kapital diinterpretasikan sama dengan huruf kecil. Contoh penulisan *address IPv6*:

21DA : 00D3 : 0000 : 2F3B : 02AA : 00FF : FE28 : 9C5A

Oleh karena format alamat *IPv6* yang panjang, ada beberapa alternatif untuk menuliskannya menjadi lebih sederhana. Penyederhanaan *address–addressIPv6* dilakukan dengan mengkompresi nilai–nilai nol "0" jika terdapat pada sebuah alamat.

- Alamat *IPv6,* normal disederhanakan dengan menghilangkan nilai-nilai nol yang terletak di depan. Sehingga contoh alamat di atas dapat dituliskan menjadi:

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

- Blok–blok berangkaian dan bersebelahan yang memiliki nilai nol '0' dapat dikompresi ke "::", dikenal dengan istilah *double–colon.*

  Contoh:

  ❖ Salah : F02 : 30 : 0 : 0 : 0 : 0 : 0 : 5 → FF02:3::5

  ❖ Benar : F02 : 30 : 0 : 0 : 0 : 0 : 0 : 5 → FF02:30::5

  Setelah mengetahui representasi penulisan alamat *IPv6,* maka perlu diketahui juga tentang representasi penulisan *prefix*. Implementasi *IPv4* umumnya menggunakan representasi *dotted-decimal* untuk *network-prefix*-nya, yang lebih dikenal dengan nama *netmask*. Istilah *netmask* tidak digunakan dalam *IPv6,* melainkan menggunakan istilah *prefix-length*. Sebuah *prefix IPv6*ditulis mengikuti notasi *Classless Inter Domain Routing* (*CIDR*). Notasi *CIDR* untuk *prefix IPv6*adalah sebagai berikut:

  Contoh penulisan notasi *CIDR*:

  ❖ IP : 3FFE:FFFF:0100:F101:0210:A4FF:FEE3:9566

  ❖ Mask : FFFF:FFFF: FFFF:FFFF:0000:0000:0000:0000

❖ Network : 3FFE:FFFF:0100:F101:0000:0000:0000:0000

❖ CIDR : 3FFE:FFFF:0100:F101::/64

Dimana:

- *IPv6 address* adalah notasi alamat *IPv6* yang telah dijelaskan di atas.
- *Prefix-length* adalah nilai decimal yang menspesifikasikan berapa banyak bit yang berurutan dari sebelah kiri (awal bit) yang termasuk dalam *prefix* (Vachon, 2008: 493-495).

2.6. Format Prefix

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

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

3FFE:2900:D005:F28B::/64

Pada contoh di atas, 64 bit pertama dari alamat tersebut dianggap sebagai *prefix* alamat, sementara 64 bit sisanya dianggap sebagai *interface ID*(Deering, 1998).

2.7. Jenis–Jenis Alamat IPv6

Alamat *IPv6* memiliki beberapa jenis. Jenis-jenis dari alamat *IPv6* ini akan mempengaruhi *prefix* yang digunakan. *IPv6* mendukung beberapa jenis *format prefix*.

2.7.1. Alamat *Unicast*

Alamat *unicast* yang menyediakan komunikasi secara *point-to-point*, secara langsung antara dua *host* dalam sebuah jaringan. Alamat *IPv6 unicast* dapat diimplementasikan dalam berbagai jenis alamat, yakni:

- Alamat *unicast global*

  Alamat *unicast globalIPv6* mirip dengan alamat publik dalam alamat *IPv4*. Dikenal juga sebagai *Aggregatable Global Unicast Address*. Seperti halnya alamat publik *IPv4* yang dapat secara global dirujuk oleh *host-host* di internet dengan menggunakan proses *routing*, alamat ini juga mengimplementasikan hal serupa. Struktur alamat *IPv6unicast global* terbagi menjadi topologi tiga level (*Public*, *Site*, dan *Node*).
- Alamat *unicast site-local*

  Alamat *unicast site-local IPv6* mirip dengan alamat privat dalam *IPv4*. Ruang lingkup dari sebuah alamat terdapat pada *internetwork* dalam sebuah *site* milik sebuah organisasi. Penggunaan alamat *unicast global* dan *unicast site-local* dalam sebuah jaringan adalah mungkin dilakukan. *Prefix* yang digunakan oleh alamat ini adalah FEC0::/48.
- Alamat *unicast link-local*

  Alamat *unicast link-local* adalah alamat yang digunakan oleh *host-host* dalam *subnet* yang sama. Alamat ini mirip dengan konfigurasi APIPA (*Automatic Private Internet Protocol Addressing*) dalam sistem operasi*Microsoft Windows XP*. *Host-host* yang berada di dalam *subnet*yang sama akan menggunakan alamat-alamat ini secara otomatis agar dapat

berkomunikasi. Alamat ini juga memiliki fungsi resolusi alamat, yang disebut dengan *Neighbor Discovery*. *Prefix* alamat yang digunakan oleh jenis alamat ini adalah FE80::/64.

- Alamat *unicast* yang belum ditentukan (*unicast unspecified address*)

  Alamat *unicast* yang belum ditentukan adalah alamat yang belum ditentukan oleh seorang *administrator*jaringan atau tidak menemukan sebuah *DHCP Server* untuk meminta alamat. Alamat ini sama dengan alamat *IPv4* yang belum ditentukan, yakni 0.0.0.0. Nilai alamat ini dalam *IPv6* adalah 0:0:0:0:0:0:0:0 atau dapat disingkat menjadi dua titik dua (::).

- Alamat *unicast loopback*

  Alamat *unicast loopback* adalah sebuah alamat yang digunakan untuk mekanisme *Interprocess Communication* (*IPC*) dalam sebuah *host*. Dalam *IPv4*, alamat yang ditetapkan adalah 127.0.0.1, sementara dalam *IPv6* adalah 0:0:0:0:0:0:0:1, atau ::1.

- Alamat *unicast 6to4*

  Alamat *unicast 6to4* adalah alamat yang digunakan oleh dua *hostIPv4* dan *IPv6* dalam *internet IPv4* agar dapat saling berkomunikasi. Alamat ini sering digunakan sebagai pengganti alamat publik *IPv4*. Alamat ini aslinya menggunakan *prefix* alamat 2002::/16, dengan tambahan 32 bit dari alamat publik *IPv4* untuk membuat sebuah *prefix* dengan panjang 48-bit, dengan format 2002:WWXX:YYZZ::/48, dimana WWXX dan YYZZ adalah representasi dalam notasi *colon-decimal format* dari notasi *dotted-decimal* formatw.x.y.z dari alamat publik *IPv4*. Sebagai contoh alamat *IPv4* 157.60.91.123 diterjemahkan menjadi alamat *IPv6*2002:9D3C:5B7B::/48.

  Meskipun demikian, alamat ini sering ditulis dalam format *IPv6 Unicast global address*, yakni 2002:WWXX:YYZZ:SLA ID:Interface ID.

- Alamat *unicast ISATAP*

  Alamat *Unicast ISATAP* adalah sebuah alamat yang digunakan oleh dua *hostIPv4* dan *IPv6* dalam sebuah intranet*IPv4* agar dapat saling berkomunikasi. Alamat ini menggabungkan *prefix* alamat *unicast link-local*, alamat *unicast site-local* atau alamat *unicast global* (yang dapat berupa

*prefix*alamat *6to4*) yang berukuran 64-bit dengan 32-bit *ISATAP Identifier* (0000:5EFE), lalu diikuti dengan 32-bit alamat *IPv4* yang dimiliki oleh *interface* atau sebuah *host*. *Prefix* yang digunakan dalam alamat ini dinamakan dengan *subnet prefix*. Meski alamat *6to4* hanya dapat menangani alamat *IPv4* publik saja, alamat ISATAP dapat menangani alamat pribadi *IPv4* dan alamat publik *IPv4*.

2.7.2. Alamat Multicast

Alamat *multicast* yang menyediakan metode untuk mengirimkan sebuah paket data ke banyak *host* yang berada dalam grup yang sama. Alamat ini digunakan dalam komunikasi *one-to-many*. Alamat *multicast IPv6* sama seperti halnya alamat *multicast* pada *IPv4*. Paket-paket yang ditujukan ke sebuah alamat *multicast* akan disampaikan terhadap semua *interface* yang dikenali oleh alamat tersebut. *Prefix* alamat yang digunakan oleh alamat *multicast IPv6* adalah FF00::/8.

2.7.3. Alamat Anycast

Alamat *anycast* yang menyediakan metode penyampaian paket data kepada anggota terdekat dari sebuah grup. Alamat ini digunakan dalam komunikasi *one-to-one-of-many*. Alamat ini juga digunakan hanya sebagai alamat tujuan (*destination address*) dan diberikan hanya kepada *router*, bukan kepada *host-host* biasa.

Alamat *anycast* dalam *IPv6* mirip dengan alamat *anycast* dalam *IPv4*, tapi diimplementasikan dengan cara yang lebih efisien dibandingkan dengan *IPv4*. Umumnya, alamat *anycast* digunakan oleh *Internet Service Provider*(*ISP*) yang memiliki banyak klien. Meskipun alamat *anycast* menggunakan ruang alamat *unicast*, tapi fungsinya berbeda daripada alamat *unicast*.

IPv6 menggunakan alamat *anycast* untuk mengidentifikasikan beberapa *interface* yang berbeda. *IPv6* akan menyampaikan paket-paket yang dialamatkan ke sebuah alamat *anycast* ke *interface* terdekat yang dikenali oleh alamat tersebut. Hal ini sangat berbeda dengan alamat *multicast*, yang menyampaikan paket ke banyak penerima, karena alamat *anycast* akan menyampaikan paket kepada salah satu dari banyak penerima.

Jika dilihat dari cakupan alamatnya, alamat *unicast* dan *anycast* terbagi menjadi alamat-alamat berikut:

- *Link-Local*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam satu *subnet.*
- *Site-Local*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam sebuah intranet.
- *Global Address*, merupakan sebuah jenis alamat yang mengizinkan sebuah komputer agar dapat berkomunikasi dengan komputer lainnya dalam internet berbasis *IPv6.*

Sementara itu, cakupan alamat *multicast* dimasukkan ke dalam struktur alamat (Hinden, 1998).

2.8. Perbandingan Alamat IPv6 dan IPv4

Tabel 2.1 berikut menjelaskan perbandingan karakteristik antara alamat IP versi 4 dan alamat IP versi 6.

Tabel 2.1 perbandingan karakteristik antara alamat IP versi 4 dan alamat IP versi 6

Kriteria	Alamat IP versi 4	Alamat IP versi 6
Panjang alamat	32 bit	128 bit
Jumlah total *host* (teoritis)	2^{32}=±4 miliar *host*	2^{128}

Tabel 2.1 (Sambungan)

Menggunakan kelas alamat	Ya, kelas A, B, C, D, dan E. Belakangan tidak digunakan lagi, mengingat telah tidak relevan dengan perkembangan jaringan internet yang pesat.	Tidak
Alamat *multicast*	Kelas D, yaitu 224.0.0.0/4	Alamat *multicast IPv6*, yaitu FF00:/8
Alamat *broadcast*	Ada	Tidak ada
Alamat yang belum ditentukan	0.0.0.0	::
Alamat *loopback*	127.0.0.1	::1
Dukungan IPSec	Dukungan *IPSec* adalah pilihan.	Dukungna *IPSec* terintegrasi.
Fragmentasi	Fragmentasi dilakukan dengan *router* pengirim dan *forwarding*.	Fragmentasi dilakukan hanya oleh pengirim.
Identifikasi aliran paket.	Tidak ada identifikasi aliran paket.	Identifikasi aliran paket tersedia dalam header *IPv6* menggunakan bagian *Flow Label*
Checksum	Bagian *Checksum* tersedia pada *header*	Tidak ada bagian *checksum* pada *header*.

Tabel 2.1 (Sambungan)

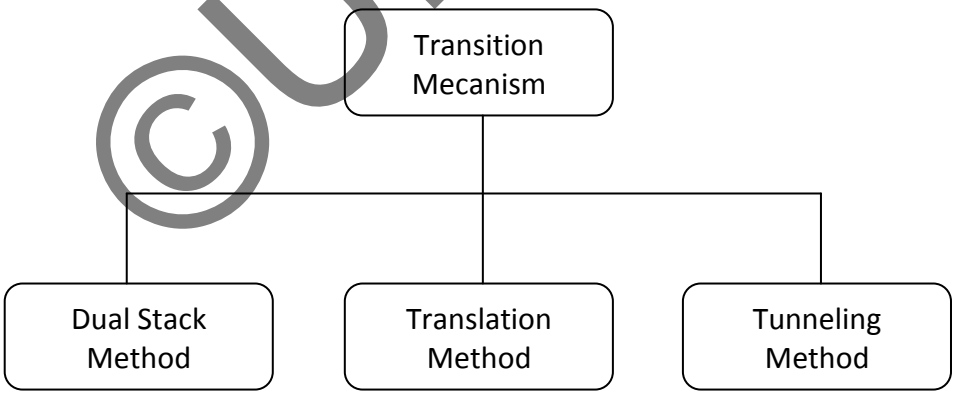
Bagian *Options*	Bagian *Options* tersedia *header*.	Tidak ada *optionfields*, tapi *Extensionheaders* tersedia
AddressResolutionProtocol (*ARP*)	*AddressResolutionProtocol* (*ARP*) tersedia untuk memetkan alamat *IPv4* ke *MACaddresses*.	*AddressResolutionProtocol* (*ARP*) diganti dengan *NeighborDiscoveryProtocol*.
Internet Group Management Protocol (*IGMP*)	*Internet Group Management Protocol* (*IGMP*) digunakan untuk mengatur keanggotaan multicast *group*.	*IGMP* diganti dengan pesan *Multicast Listener Discovery* (*MLD*).
Broadcast messages	*Broadcastmessages* tersedia	*Broadcast messages* tidak tersedia. Sebaliknya lingkup semua alamat *link-localmulticast* digunakan untuk *broadcast*.
Konfigurasi *address* IP.	Konfigurasi manual (*Static*) dari alamat IP atau *DHCP* (*Dynamicconfiguration*) diperlukan untuk mengkonfigurasi *address* IP.	Konfigurasi alamat otomatis tersedia.

2.9. Mekanisme Transisi IPv4/IPv6

Kunci sukses dalam transisi *IPv6* adalah kompatibilitas *IPv6* dengan *host* dan *router IPv4* yang sudah ada pada jaringan. Namun, pada dasarnya *IPv4* dan *IPv6* tidak kompatibel, sehingga memerlukan suatu mekanisme yang biasa disebut mekanisme transisi. Mekanisme transisi digunakan untuk mencapai dua hal berikut, yaitu:

1. Melewatkan paket *IPv6* melalui jaringan *IPv4* yang sudah ada atau sebaliknya.
2. Membuat agar terminal *IPv6* dapat berkomunikasi dengan terminal *IPv4.*

Teknik transisi dibutuhkan dalam setiap implementasi sistem baru yang berbeda dengan sistem yang telah ada, demikian pula untuk implementasi *IPv6.* Tujuan teknik transisi ini adalah agar infrastruktur yang bebrbasis *IPv4* dapat tetap berlangsung selama masa peralihan menuju penggunaan *IPv6.* Teknik transisi dilaksanakan dengan menyediakan interoperabilitas langsung antara *node–node* yang berbasis *IPv6* dengan *node–node* yang berbasis *IPv4.*Berikut gambar 2.5 menunjukkan pembagian teknik transisi.



Gambar 2.5 Mekanisme Transisi

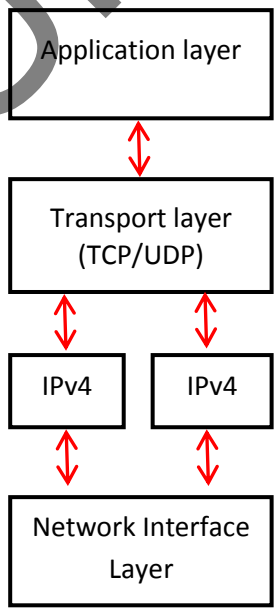
Gambar 2.5 di atas menunjukkan beberapa teknik transisi paling umum yang digunakan sesuai dengan *draft internet engineering task force* (*IETF*).

Teknik transisi *IPv4* menuju *IPv6* dibahas dalam *IETF IPng Transition Working Group* (*NGtrans*). Strategi migrasi *IPv6* terdiri dari tiga komponen utama, yaitu:

1. Teknik transisi *dual stack.*
2. Teknik transisi translasi.
3. Teknik transisi *tunneling* (Gilligan, 2000).

2.9.1. Teknik Transisi Dual Stack

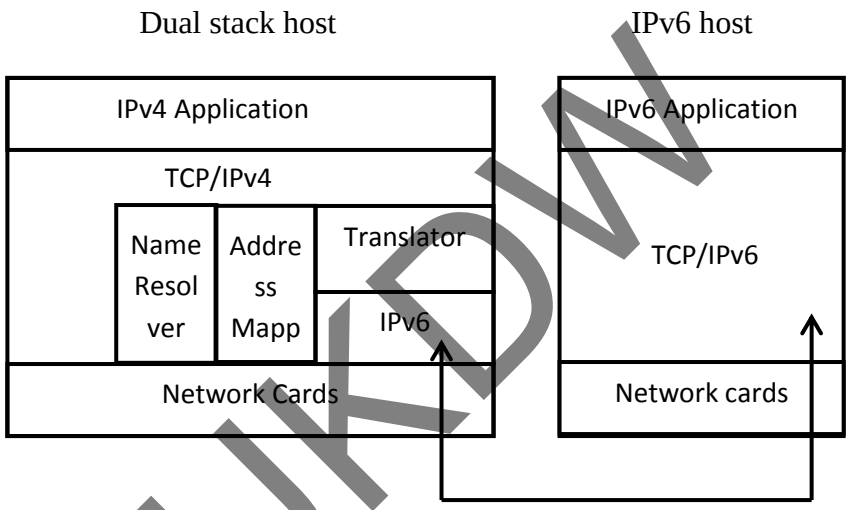
Teknik *dual stack* merupakan teknik dasar untuk teknik transisi yang lain. Teknik *dual stack* pada dasarnya menggunakan *dual stack protocol* pada *internet layerTCP/IP*, yaitu *IPv4* dan *IPv6.* Dengan teknik *dual stack,* sebuah perangkat mampu beroperasi melalui kedua protokol. Pada *end-system,dual stack*memungkinkan aplikasi–aplikasi *IPv4* dan *IPv6* beroperasi dalam *node* yang sama. Sedangkan teknik *dual stack*yang ada pada *router* memberi kemampuan *router* dalam menangani kedua jenis paket *IPv4* dan *IPv6*(Gilligan, 2000). Berikut gambar 2.6 menunjukkan mekanisme *dual stack.*



Gambar 2.6 Mekanisme Dual Stack

2.9.2. Teknik Transisi Translation

Teknik *translation* merupakan konversi langsung protokol–protokol, antara *IPv4* dan *IPv6*, yang memuat transformasi kedua *header* dan *payload protokol.* Transisi dapat terjadi pada beberapa *layer* dalam *stack protokol*, termasuk *layer network,transport* dan aplikasi. Translasi mengijinkan *IPv6-onlynode* untuk berkomunikasi *IPv4-only node.* Berikut gambar 2.7 menunjukkan mekanisme translasi.

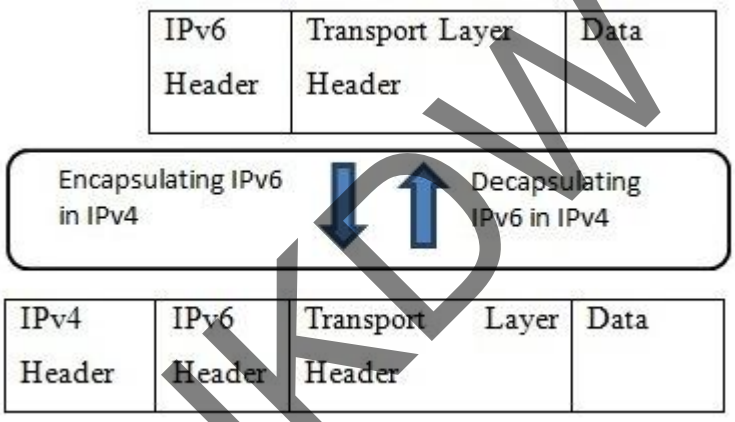


Gambar 2.7 Mekanisme Translasi

Translasi protokol seringkali menimbulkan hilangnya fitur, dimana tidak terdapat peletakan fitur yang jelas antara protokol–protokol yang ditranslasikan. Sebagai contoh translasi *header IPv6* menjadi *header IPv4* akan meyebabkan hilangnya *flow label IPv6.* Sehingga dalam interkoneksi *IPv4/IPv6* tidak direkomendasikan menggunakan *translation* dalam transisi *IPv6* (Hagino, 2001).

2.9.3. Teknik Transisi Tunneling

Teknik t*unneling* merupakan teknik yang paling banyak diaplikasikan dalam interkoneksi *IPv4/IPv6.* Teknik tunneling melakukan enkapsulasi paket *IPv6* dalam paket–paket *IPv4* sehingga paket tersebut dapat dikirimkan melalui *backbone IPv4,* serta mengijinkan *end-system* dan *router IPv6* yang terisolasi oleh jaringan *IPv4* untuk dapat berkomunikasi tanpa perlu men-*upgrade* infrastruktur *IPv4* yang berada diantaranya, sehingga tidak mempunyai efek samping. Berikut gambar 2.8 menunjukkan mekanisme tunneling.



Gambar 2.8 Mekanisme Tunneling

Dalam implementasinya, terdapat dua jenis mekanisme *tunneling* dalam melewatkan paket–paket *IPv6* di atas jaringan *IPv4,* yaitu:

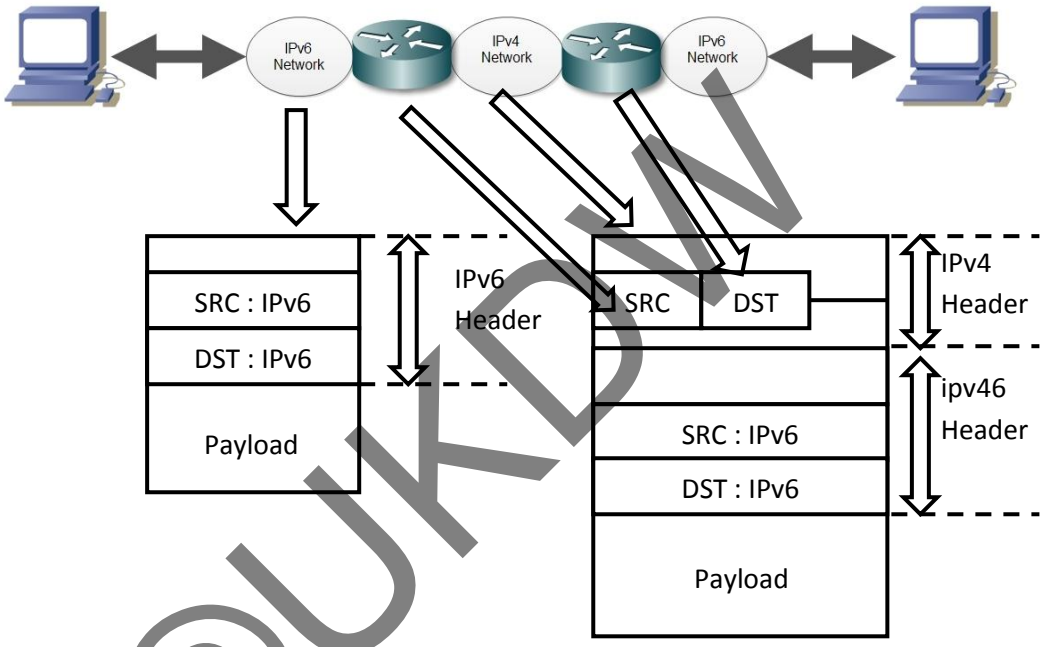
1. *Configured tunneling*.
2. *Automatic tunneling*.

Kedua tipe *tunnel* di atas memiliki perbedaan mendasar, yaitu dari bagaimana alamat *endpoint tunnel* ditentukan.

- *Configured tunneling*

Dalam *configurd tunneling,* alamat *endpoint tunnel* ditentukan dari informasi konfigurasi pada *node* yang melakukan enkapsulasi (enkapsulator). Dalam setiap *tunnel,* enkapsulator harus menyimpan alamat *endpoint*

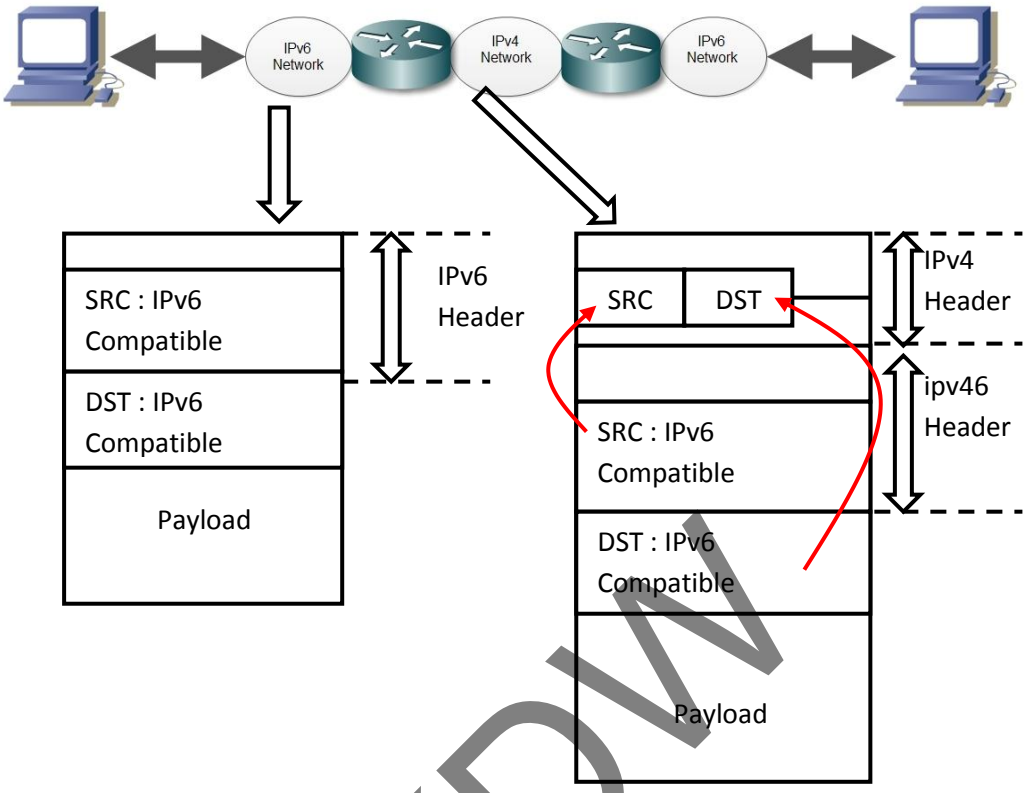
*tunnel.*Ketika paket *IPv6* akan dikirimkan melalui sebuah *tunnel,* alamat pengirim dan tujuan untuk enkapsulasi *header IPv4* harus diset terlebih dahulu. *Field protocol* diset bernilai 41. Ini menunjukkan bahwa tipe *payload* adalah paket *IPv6.* Kemudian *field source address* diset dengan alamat *IPv4* dari enkapsulator. *Destination address* diset dengan alamat *endpoint* dari *tunnel.* Gambar 2.9 berikut menunjukkan ilustrasi dari jaringan dengan menggunakan *configured tunneling.*



Gambar 2.9 Configured Tunneling

- *Automatic tunneling*

Dalam *automatic tunneling,* alamat *tunnel endpoint* ditentukan oleh alamat *IPv4-compatible node* tujuan dari paket *IPv6* yang dikirim. Dengan demikian, komunnnikasi antar *node IPv6/IPv4* bisa dilakukan tanpa perlu melakukan konfigurasi awal sebelum paket dikirim. Gambar 2.10 menunjukkan ilustrasi *automatic tunneling* pada sebuah jaringan. Dari gambar 2.10 bisa terlihat bahwa *destination address* dari paket *IPv6* dan paket *IPv4* adalah sama.



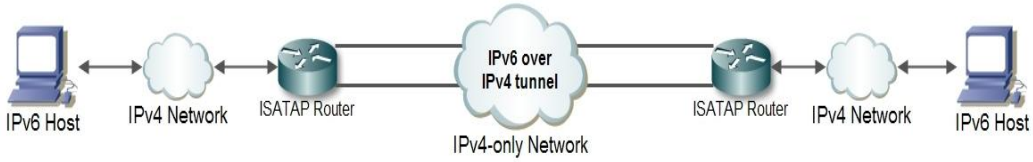
Gambar 2.10 Automatic Tunneling

Metode yang populer dan paling banyak digunakan dalam mekanisme *tunneling* adalah a*utomatic tunneling*, ini dikarenakan *host–host end point IPv6* memerlukan konfigurasi secara manual dan addr*ess–address ISATAP* menggunakan mekanisme autokonfigurasi *address* standar(Durand, 2001).

2.9.4. Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

ISATAP adalah sebuah teknologi penetapan *address* dan *automatic tunneling host-to-host, router-to-host,* dan *host-to-router* yang berguna untuk konektivitas *unicast*antar *IPv6* yang melewati jaringan internet *IPv4.* Mekanisme ISATAP dinyatakan dalam RFC 5214. Sesuai dengan fungsinya sebagai mekanisme *tunneling* otomatis, *host-host*ISATAP tidak memerlukan konfigurasi secara manual dan *address-address*ISATAP menggunakan mekanisme

autokonfigurasi *address* standar. Gambar 2.11 berikut menunjukkan contoh topologi ISATAP.



Gambar 2.11 Topologi ISATAP

Prinsip pada topologi ISATAP yaitu:

- Adanya jaringan atau *node IPv6-only*.
- Adanya jaringan atau *node IPv4-only* untuk memastikan adanya *tunneling IPv6* di dalam jaringan *IPv4* tersebut.
- Terdapat *host dual stack* yang mendukung *IPv4* dan *IPv6* untuk implementasi mekanisme ISATAP.

ISATAP dapat diimplementasikan untuk komunikasi antara *node-nodeIPv4/IPv6* dalam sebuah jaringan *IPv4.Address-address*ISATAP menggunakan *interface identifier local*::0:5EFE:w.x.y.z (dimana w.x.y.z adalah *addressunicast IPv4* yang memuat *address public* dan *address private*). *Interface identifier* ISATAP dapat dikombinasikan dengan *prefix-prefix*64-bit yang valid untuk *address-address unicast IPv6.* Ini termasuk *prefix address link-local*(FE80::/64) dan *prefix-prefix* global (termasuk *prefix-prefix6to4*). Seperti halnya *address-address IPv4-compatible. Address*ISATAP memuat *address IPv4* sumber atau tujuan dalam *header IPv4* saat trafik *IPv6ISATAP-address* di-*tunnel* melewati jaringan *IPv4.Link-local ISATAP address* memungkinkan dua buah *host* berkomunikasi melalui jaringan *IPv4* menggunakan *address ISATAP link-local*satu sama lainnya.

Sebagai strategi transisi, ISATAP memberikan keleluasaan untuk mengaktifkan dan menghubungkan konektivitas *IPv6* pada jaringan *IPv4* yang sudah ada ketika infrastruktur secara bertahap berpindah untuk berintegrasi ke

jaringan *IPv6*. Kelebihan mekanisme transisi *automatic tunnelling*ISATAP dalam implementasinya antara lain:

- Lebih mudah dalam implementasi

Dalam implementasinya tidak memerlukan banyak komputer, cukup menggunakan komputer yang sudah ada. Khusus untuk *gateway tunnel, operating system* perlu di-*upgrade* menjadi *operating system* yang *dual stack* yang mendukung *IPv6* dan *IPv4.*

- Lebih mudah dalam hal konfigurasi pada sistem operasi

Dalam konfigurasitidak diperlukan konfigurasi yang rumit, cukup dengan konfigurasi *interface tunnel* dan konfigurasi tabel *routing*-nya saja.

- Tidak memerlukan *server* yang melayani transisi

Dalam implementasi tidak memerlukan *server* khusus yang melayani mekanisme transisinya, enkapsulasi dan dekapsulasi dilakukan antar *gateway tunnel* secara *point to point.*

Karena perubahan dari *IPv4* ke *IPv6* dalam waktu yang singkat adalah hal yang mustahil, disebabkan ukuran jaringan *internet* yang besar dan jumlah pengguna *IPv4* yang sangat banyak. Perubahan atau migrasi dari *IPv4* ke *IPv6* ini perlu dilakukan secara bertahap, *node* demi *node,* dengan metode konfigurasi otomatis, agar tidak perlu lagidilakukan konfigurasi disetiap host secara manual. Dengan cara seperti ini, pengguna akan lebih cepat merasakan kelebihan dari *IPv6*, sementara di sisi lain terus mengembangkan *IPv6* (Bi, 2007).

2.10. Parameter Performansi Jaringan IPv6 Over IPv4

Parameter performansi jaringan atau yang disebut dengan *QoS* (*Quality of Service*) merupakan parameter yang digunakan dalam pengukuran dan menunjukkan seberapa baik dan buruknya performansi jaringan. *QoS* merupakan hal yang sangat sensitif apabila dikaitkan dengan proses pertukaran data dalam lalu lintas *internet*. Berikut ini beberapa parameter performansi jaringan terkait dengan pengukuran interkoneksi *IPv6 over IPv4.*

- *Round Trip Time (RTT)*

Round trip time atau yang disebut juga *Round Trip Delay* merupakan waktu yang diperlukan sebuah paket yang dikirim dari sumber menuju tujuan dan kembali lagi ke sumber. RTT memberikan informasi tentang waktu yang diperlukan sebuah paket *ICMP* untuk kembali diterima oleh pengirim setelah mendapat balasan dari penerima (pengiriman *ICMP* paket *echo-request* dan menunggu paket *ICMPecho-reply*) (Dye, 2007: 231).

- *Jitter*

Jitter merupakan variasi *end-to-end delay* yang terjadi akibat adanya selisih waktu atau interval antar kedatangan paket di penerima. Besarnya nilai *jitter* sangat dipengaruhi besarnya tumbukan antar paket *congestion* yang ada pada jaringan IP. Untuk mengatasi *jitter* maka paket data yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar. Maksimum *jitter* yang dapat diperbolehkan sebesar 50ms(Demichelis, 2002).

- *Throughput*

Throughput adalah jumlah bit yang diterima dengan sukses perdetik melalui sebuah sistem atau media komunikasi. *Throughput* diukur setelah transisi data (*host/client*) karena suatu sistem akan menambah *delay* yang disebabkan *processor limitations*, kongesti jaringan, *buffering inefficients, error* transmisi, *traffic loads* atau mungkin desain *hardware* yang tidak mencukupi. Aspek utama *throughput* yaitu berkisar pada ketersediaan *bandwidth* yang cukup untuk menjalankan aplikasi. Hal ini menentukan besarnya trafik yang dapat diperoleh suatu aplikasi saat melewati jaringan. Aspek penting lainnya adalah *error* (pada umumnya berhubungan dengan *link error rate*) dan *losses* (pada umumnya berhubungan dengan kapasitas *buffer*). Satuan yang biasa digunakan adalah *bit per s*econd (bps) atau paket per detik (packet/sec).

*Throughput*tergantung pada faktor–faktor berikut ini:

1. Karakteristik *link*: *bandwidth, error rate.*

2. Karakteristik *node*: kapasitas *buffer*, daya pemrosesan(Bradner, 1999)

- *Packet Loss*

Packet Loss adalah merupakan besar dari paket yang hilang dalam jaringan karena terjadi tabrakan atau *collision*. Dalam suatu jaringan *packet loss* akan selalu mempunyai nilai dengan satuan persen (%). Yang menjadi faktor timbulnya *packet loss* adalah kepadatan *traffic* dan *bandwidth*. Semakin besar *bandwidth*, maka akan memperkecil terjadinya tabrakan data antara *user* yang satu dan yang lainnya. Jika terjadi *packet loss* maka *protocol network* yang ada pada *router* akan meminta pengirim untuk mengirim ulang paket data yang hilang tersebut. Pada saat proses pengiriman ulang data yang hilang tersebut maka akan menyebabkan meningkatnya nilai *Jitter*. Detektor dari *packet loss* berada didalam router yang bernama *Carrier Sense Multiplexing And Collision Detection* (*CSMA-CD*). Standar *ITU* untuk *packet loss* adalah tidak boleh melebihi 10% dari jumlah paket data keseluruhan (Morton, 2012).

- *Response Time*

Response time adalah selisih waktu antara permintaan dengan respon terhadap permintaan. *Response Time* juga biasa diartikan waktu tanggap yang diberikan oleh antar muka/*interface* ketika *user* mengirim permintaaan ke komputer. Secara umum, pengguna menginginkan bahwa program aplikasinya dapat memberikan waktu tanggap yang sependek-pendeknya. Tetapi waktu tanggap yang baik memang tidak dapat ditentukan, karena ada beberapa aspek yang mempengaruhi, antara lain yakni ragam interaksi yang diinginkan dan kefasihan pengguna dalam menjalankan program aplikasi tersebut (Fenner, 1997).

BAB III
RANCANGAN PENELITIAN

3.1. Pendahuluan

Tujuan dari tugas akhir ini adalah melihat bagaimana proses transisi dari penggunaan *IPv4* ke *IPv6*, mengamati mekanisme dan mengukur kinerja dua teknik transisi *IPv4/IPv6.* Penulis akan membuat sistem jaringan dengan penggunaan *IPv4-only* dan lalu melakukan proses migrasi ke *IPv6* dengan menerapkan teknik transisi. Teknik transisi yang akan digunakan adalah *dual stack* dan ISATAP. Pengukuran kinerja akan dilakukan terhadap *end-to-end* di dalam proses pengiriman paket *IPv6.*

Di dalam bab ini juga berisi penjelasan yang berkaitan dengan perancangan konfigurasi sistem yang digunakan dalam tugas akhir dan langkah–langkah yang digunakan untuk mendapatkan parameter pengukuran kinerja yang ditentukan.

3.2. Penentuan Operating System Bagi Setiap PC

Operating system yang digunakan untuk PC yang bertindak sebagai *router* adalah *Windows Server 2003*dan untuk PC yang bertindak sebagai *client* adalah *Windows XP Professional SP2.* Pemilihan *operating system* berbasis *windows* dikarenakan *operating system* ini mempunyai banyak kelebihan didalam menagani sistem jaringan. Hal tersebut didasarkan pada:

a) Mudah untuk dikembangkan, dikelola dan digunakan.

Dengan antarmuka *Windows* yang telah familiar bagi banyak orang tentunya Windows Server 2003 akan lebih mudah digunakan. Dalam pengkonfigurasian akan sangat dibantu dengan adanya *wizard* yang akan

mempersingkat pengaturan. Hal ini akan membuat kegiatan manajemen *server* menjadi sederhana sehingga tidak diperlukan seseorang *administrator* pengalaman untuk melakukan beberapa pengaturan.

b) Banyak *software* berbasis *Windows.*

Sebagai sistem operasi yang memegang pasar yang besar di dunia, hal ini mengakibatkan banyak perusahaan-perusahaan pembuat *software* yang memilih membuat produk mereka berjalan di atas *windows operating system.*

c) Dukungan *driver* yang lebih banyak.

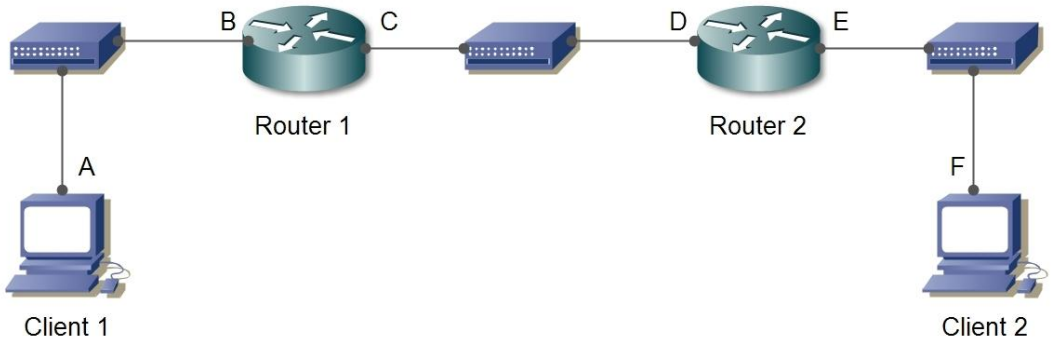
Dukungan *driver* yang lebih banyak akan sangat membantu ketika seorang *network administrator* menambahkan modul jaringan pada sebuah *server*.

d) Dukungan terhadap implementasi ISATAP dan *Dual Stack.*

ISATAP (*Intra-site Automatic Tunnel Addressing Protocol*) pada *Windows Server 2003* dan *Windows XP Professional* memiliki nama lain yaitu *Automatic Tunneling Pseudo. Windows Server 2003* dan *Windows XP* memiliki kemampuan menerapkan teknik ISATAP dan *Dual Stack* tanpa harus menambahkan *software-software* tertentu. Hal ini sangat berbeda dengan beberapa *distroLinux* yang mengharuskan pembaharuan kernel dan pemasangan beberapa *software* pendukung untuk mengimplementasikan teknik ISATAP. Protokol *IPv6* pada *Windows Server 2003* dan *Windows XP Professional* memberikan konfigurasi alamat *link-local* ISATAP pada *Interface Automatic Tunneling Pseudo* untuk setiap alamat *IPv4* yang diberikan pada *interface* ketika *IPv6* diaktifkan pada PC tersebut.

3.3. Topologi Awal Sebelum Dilakukan Teknik Transisi

Teknik transisi adalah teknik penerapan *IPv6* secara bertahap pada sistem yang telah terpasang *IPv4* sebelumnya. Dalam penelitian ini, penulis melakukan penerapan teknik transisi secara bertahap. Pertama-tama penulis membuat jaringan *native IPv4*. Dari topologi yang dihasilkan inilah akan dilakukan penerapan teknik transisi. Berikut gambar 3.1 menunjukkan topologi *native IPv6*.



Gambar 3.1. Topologi awal sebelum dilakukan teknik transisi

Topologi ini hanya menerapkan *IPv4* dalam setiap *interface*-nya. Topologi ini dibuat untung membandingkan dan menjadi acuan dari perubahan-perubahan yang terjadi ketika transisi dari *IPv4*ke *IPv6* mulai diterapkan. *Routing* yang dipakai dalam topologi ini adalah *dynamic routing*. Dalam penelitian ini penulis tidak memperhatikan *routing protocol*. Hal ini dikarenakan penulis memfokuskan pada implementasi teknik tansisi dan mengamati perubahan-perubahan yang terjadi ketika transisi dari *IPv4* ke *IPv6*.Berikut tabel 3.1 daftar alamat *IPv4* yang dipakai pada setiap *interface*.

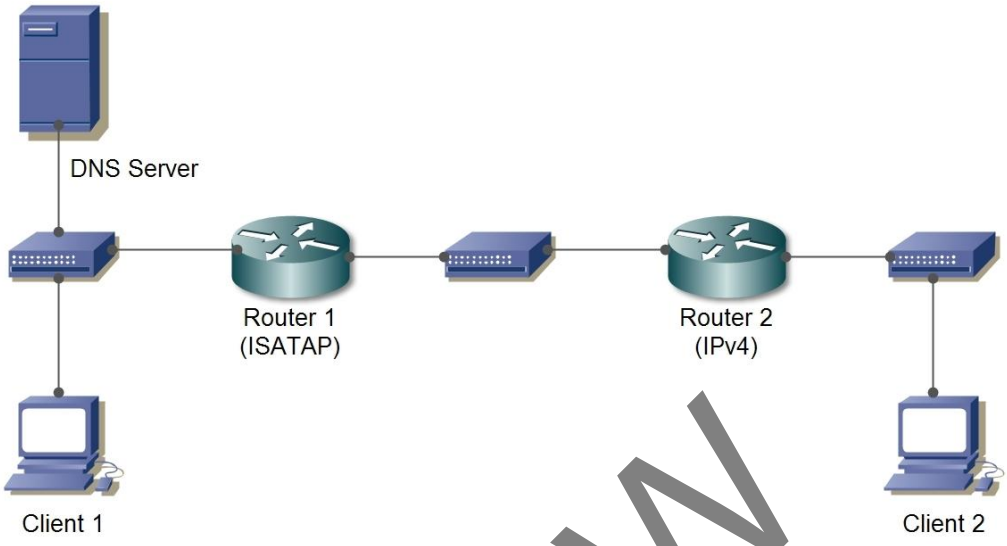
Tabel 3.1. Daftar alamat IPv4 yang dipakai pada setiap interface

Interface	Alamat IP
A	10.27.10.2/24
B	10.27.10.1/24
C	10.27.20.1/24
D	10.27.20.2/24
E	10.27.30.1/24
F	10.27.30.2/24

3.4. Desain Jaringan untuk Implementasi Teknik ISATAP pada LAN

Dari topologi *native IPv4* yang digambarkan pada gambar 3.1, penulis melakukan penerapan teknik transisi ISATAP pada topologi tersebut. Berikut

gambar 3.2 merupakan hasil topologi setelah diterpkannya teknik transisi ISATAP.



Gambar 3.2 Desain mekanisme implementasi ISATAP pada LAN

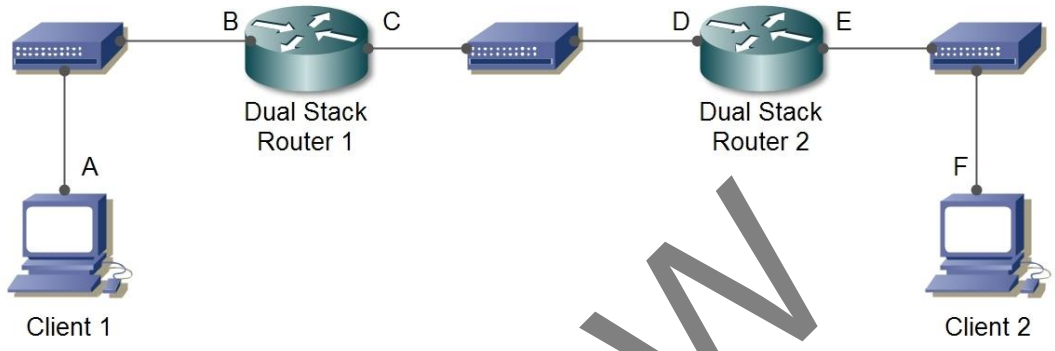
Untuk menguji jaringan ISATAP digunakan suatu desain arsitektur jaringan yang dapat menggambarkan hal-hal sebagai berikut:

- Adanya *client/node IPv6-only*, dalam tugas akhir ini *IPv6-only* diwakili oleh ISATAP *client 1*
- Adanya *router IPv4-only* untuk memastikan adanya *tunneling IPv6* di dalam jaringan *IPv4* tersebut. Pada tugas akhir ini *IPv4 only* diwakili oleh *router 2*
- Terdapat *host dual stack* yang mendukung *IPv4* dan *IPv6* untuk implementasi mekanisme ISATAP. Hal ini diwakili oleh ISATAP *router 1*

Pengalamatan *IPv6* pada mekanisme ISATAP ini akan dilakukan dengan pengalamatan otomatis. Pengalamatan global *IPv6* akan diberikan melalui setiap *router* yang terhubung dengan *client.* Pengalamatan globat *IPv6* akan dibatasi dengan beberapa *subnetprefix*.

3.5. Desain Jaringan untuk Implementasi Teknik Dual Stack pada LAN

Setelah melakukan penerapan teknik ISATAP, dari topologi *native IPv4* yang digambarkan pada gambar 3.1, penulis melakukan penerapan teknik transisi *dual stack* pada topologi tersebut. Berikut gambar 3.3 merupakan hasil topologi setelah diterpkannya teknik transisi *dual stack.*



Gambar 3.3Desain jaringan untuk implementasi teknik dual stack

Untuk menguji jaringan *dual stack*, digunakan suatu desain arsitektur jaringan yang dapat menggambarkan hal-hal sebagai berikut:

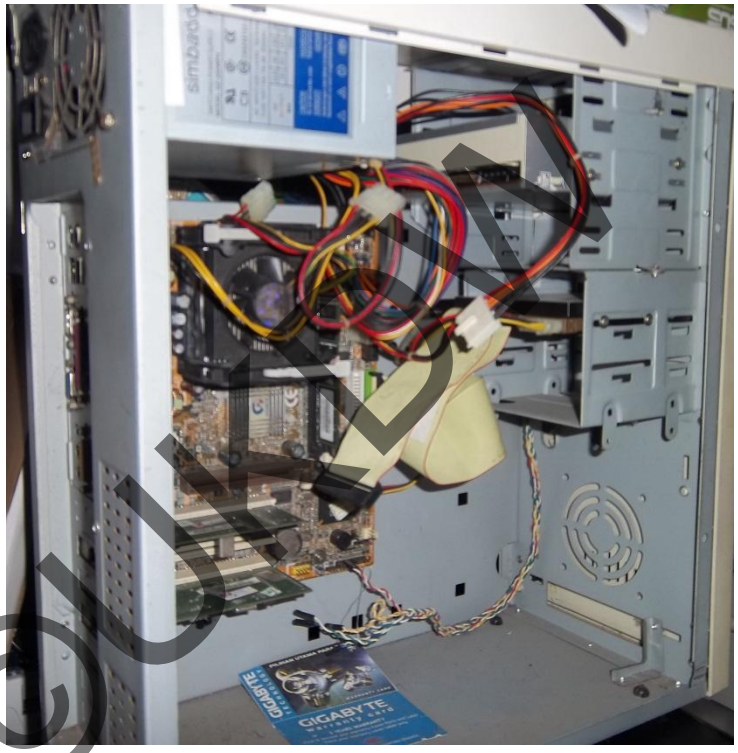
- Setiap *node* pada jaringan ini, baik *router* maupun klien, semuanya menerapkan *dual stack IP*.
- *Router Dual Stack 1* dan *Router Dual Stack 2* akan menjadi penghubung antara *subnet network.*
- Pada setiap *router* akan dikonfigurasi alamat global *IPv6.* Alamat Global *IPv6* ini lah yang akan memberikan konektivitas *IPv6* dalam topologi ini.

3.6. Peralatan yang Digunakan dalam Implementasi Teknik Transisi

Teknik transisi yang diimplementasikan pada penelitian ini menggunakan beberapa peralatan jaringan. Berikut perangkat jaringan yang digunakan dalam penerapan teknik transisi.

a. *Router 1*

Router 1 adalah *router* yang menjadi *gateway* dari *subnet* 10.27.10.0/24 dan menjadi penghubung antara subnet 10.27.10.0/24 dan *subnet* 10.27.20.0/24. *Router* 1 pada teknik transisi ISATAP digunakan sebagai *router* ISATAP dan menjadi *gateway* dari *tunnel* ISATAP. Pada teknik transisi *dual stack router* 1 berfungsi sebagai *router* yang melakukan *routing* untuk *stack IPv4* dan *IPv6*. Berikut gambar 3.4 menunjukkan PC yang digunakan sebagai *router* 1.



Gambar 3.4 PC yang digunakan sebagai router 1

Berikut tabel 3.2 menunjukkan spesifikasi dari PC yang digunakan sebagai *router* 1.

Tabel 3.2 Spesifikasi PC yang digunakan sebagai router 1

Fitur Perangkat	**Spesifikasi**
Motherboard	Gigabyte SIS-651
Processor	Intel(R) Pentium(R) 4 CPU 1,7 GHZ

Tabel 3.2 (Sambungan)

Fitur Perangkat	Spesifikasi
Memory	512 MB
Hardisk	40 GB
NIC 1	Realtek RTL8139/810x Family Fast
NIC 2	Realtek RTL8139/810x Family Fast
Operating system	Microsoft Windows Server 2003 Standard Edition Service Pack 2 v2825

b. Router 2

Router 2 adalah *router* yang menjadi *gateway* dari *subnet* 10.27.20.0/24 dan menjadi penghubung antara *subnet* 10.27.20.0/24 dan *subnet* 10.27.30.0/24. Router 2 pada teknik transisi ISATAP digunakan sebagai *router* yang menerapkan *IPv4*. Pada teknik transisi *dual stack router* 2 berfungsi sebagai *router* yang melakukan *routing* untuk *stack IPv4* dan *IPv6*. Berikut gambar 3.5 menunjukkan PC yang digunakan sebagai *router* 2.

Gambar 3.5 PC yang digunakan sebagai router 2

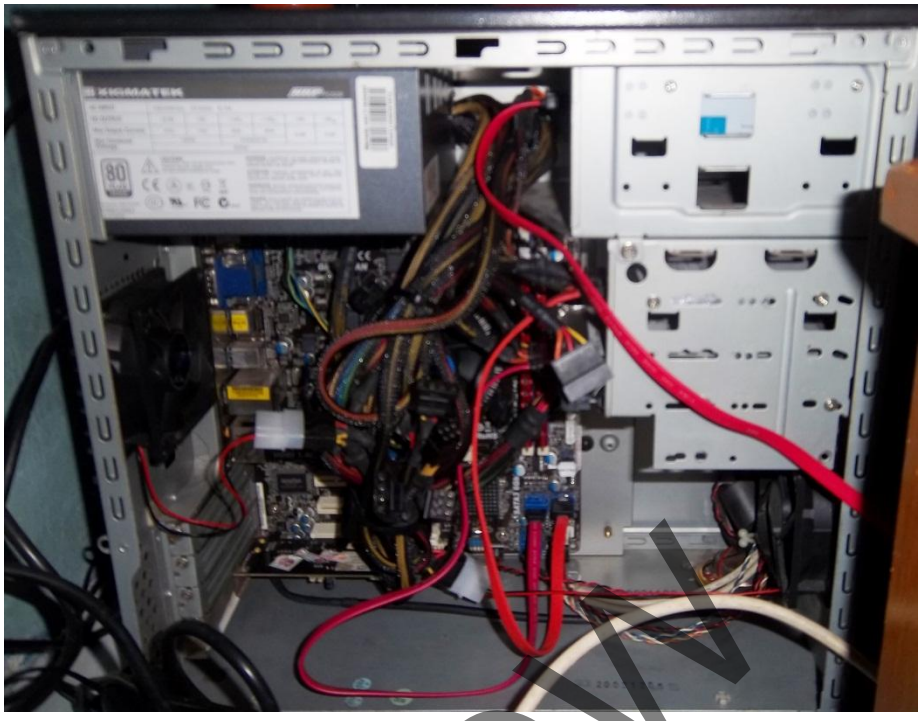
Berikut tabel 3.3 menunjukkan spesifikasi dari PC yang digunakan sebagai *router* 2.

Tabel 3.3 Spesifikasi PC yang digunakan sebagai router 2

Fitur Perangkat	**Spesifikasi**
Motherboard	MSI 7142
Processor	AMD Sempron 2800+ 1,6 GHZ
Memory	512 MB
Hardisk	40 GB
NIC 1	VIA Rhine II Fast Ethernet
NIC 2	Realtek RTL8139/810x Family Fast
Operating system	Microsoft Windows Server 2003 Standard Edition Service Pack 2 v2825

c. DNS Server

Pada penerapan teknik transisi, DNS *server* hanya digunakan pada teknik transisi ISATAP. Penggunaan DNS *server* diperuntukkan sebagai penentu *gateway tunnel* ISATAP. Pada teknik transisi *dual stack* tidak diperlukan penggunaan DNS *server*. Penggunaan DNS *server* pada teknik *dual stack* bisa dimanfaatkan untuk menerjemahkan nama komputer ke *IP address*, namun ini tidak berhubungan langsung dengan teknik transisi. Berikut gambar 3.6 menunjukkan PC yang digunakan sebagai DNS *server.*



Gambar 3.6 PC yang digunakan sebagai DNS server

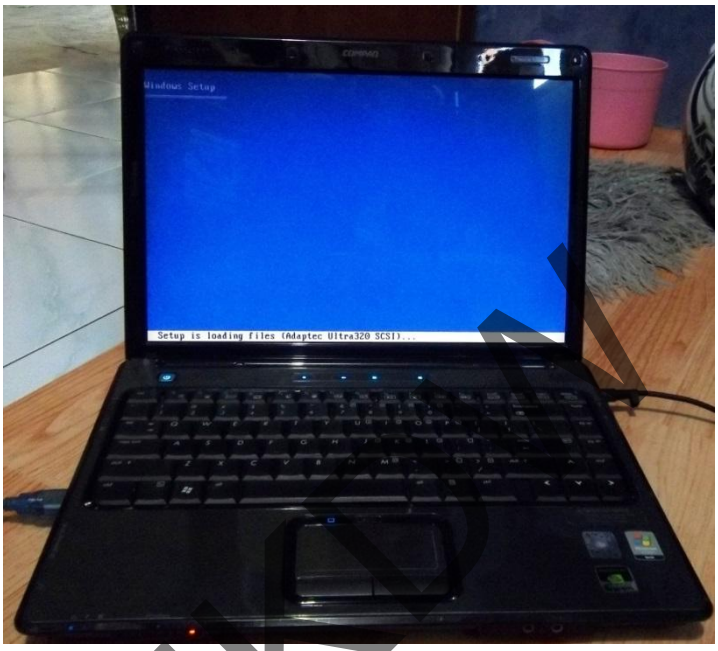
Berikut tabel 3.4 menunjukkan spesifikasi dari PC yang digunakan sebagai DNS *server*.

Tabel 3.4 Spesifikasi PC yang digunakan sebagai DNS server

Fitur Perangkat	**Spesifikasi**
Motherboard	ASRock H67M
Processor	Intel Core i3 2100 3,1 GHZ
Memory	2,73 GB
Hardisk	500 GB
NIC 1	Realtek PCIe GBE Family Controller
NIC 2	Realtek RTL8139/810x Family Fast
Operating system	Microsoft Windows Server 2003 Standard Edition Service Pack 2 v2825

d. Client 1

Client 1 adalah PC yang menjadi *client* dari *subnet* 10.27.10.0/24. *Client* 1 menjadi tujuan dari setiap tes yang dilakukan pada teknik transisi. Berikut gambar 3.7 menunjukkan PC yang digunakan sebagai *client* 1.



Gambar 3.7 PC yang digunakan sebagai client 1

Berikut tabel 3.5 menunjukkan spesifikasi dari PC yang digunakan sebagai *client* 1.

Tabel 3.5 Spesifikasi PC yang digunakan sebagai client 1

Fitur Perangkat	**Spesifikasi**
Motherboard	Wistron 30CD
Processor	Intel Mobile Core 2 Duo T7500 2,2 GHZ
Memory	2 GB
Hardisk	160 GB
NIC	Marvell Yukon 88E8039 PCI-E Fast Ethernet Controller
Operating System	Microsoft Windows XP Professional SP3

e. Client 2

Client 2 adalah PC yang menjadi *client* dari *subnet* 10.27.30.0/24. *Client* 2 menjadi asal dari setiap tes yang dilakukan pada teknik transisi. Berikut gambar 3.8 menunjukkan pc yang digunakan sebagai *client* 2.

Gambar 3.8 PC yang digunakan sebagai client 2

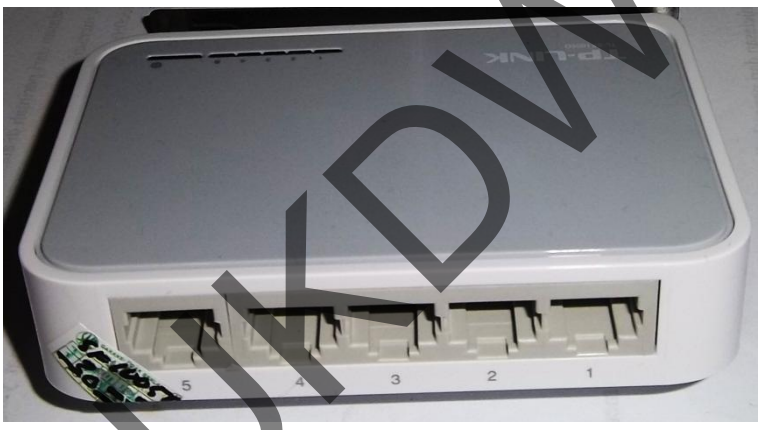
Berikut tabel 3.6 menunjukkan spesifikasi dari PC yang digunakan sebagai *client* 2.

Tabel 3.6 Spesifikasi PC yang digunakan sebagai client 2

Fitur Perangkat	**Spesifikasi**
Motherboard	Wistron 30B5
Processor	AMD Turion 64 X2 Mobile Tl-50 1,61 GHZ
Memory	1 GB
Hardisk	160 GB
NIC	NVDIA nForce Networking Controller
Operating System	Microsoft Windows XP Professional SP3

f. 5-Port 10/100Mbps Desktop Switch

5-Port 10/100Mbps *Desktop Switch* adalah sebuah *switch hub* yang digunakan untuk mengelompokkan setiap subnet pada sebuah segmen jaringan. Dalam implementasi teknik transisi, perangkat jaringan dibedakan menjadi 3 segmen. Segmen 1 untuk *subnet* 10.27.10.0/24, Segmen 2 untuk *subnet* 10.27.20.0/24 dan Segmen 3 untuk *subnet* 10.27.30.0/24. *Switch hub* ini digunakan sebanyak tiga buah pada setiap implementasi teknik transisi. Berikut gambar 3.9 menunjukkan *switch hub* yang digunakan pada implementasi teknik transisi dan tabel 3.7 yang menunjukkan spesifikasi *5-Port 10/100Mbps Desktop Switch*.



Gambar 3.9 5-Port 10/100Mbps Desktop Switch yang digunakan sebagai switch hub

Tabel 3.7 Spesifikasi 5-Port 10/100Mbps Desktop Switch yang digunakan sebagai switch hub

Fitur Perangkat	Spesifikasi
Interface	5 10/100Mbps RJ45 Ports AUTO Negotiation/AUTO MDI/MDIX
External Power Supply	100-240VAC, 50/60Hz
Dimensions (W X D X H)	4.1 x 2.8 x 0.9 in. (103.5 x 70 x 22 mm)
Fan Quantity	Fanless
Transfer Method	Store and Forward

Tabel 3.7 (Sambungan)

Fitur Perangkat	Spesifikasi
Advanced Functions	Green Technology, saving power up to 60% 802.3X Flow Control, Back Pressure Auto-Uplink Every Port
Certification	FCC, CE, RoHs
Package Contents	5-Port 10/100Mbps Desktop Switch Power Adapter User Guide
System Requirements	Microsoft® Windows® 98SE, NT, 2000, XP, Vista™ or Windows 7, MAC® OS, NetWare®, UNIX® or Linux.
Environment	Operating Temperature: 0℃~40℃ (32°F~104°F); Storage Temperature: -40℃~70℃ (-40°F~158°F); Operating Humidity: 10%~90% non-condensing; Storage Humidity: 5%~90% non-condensing

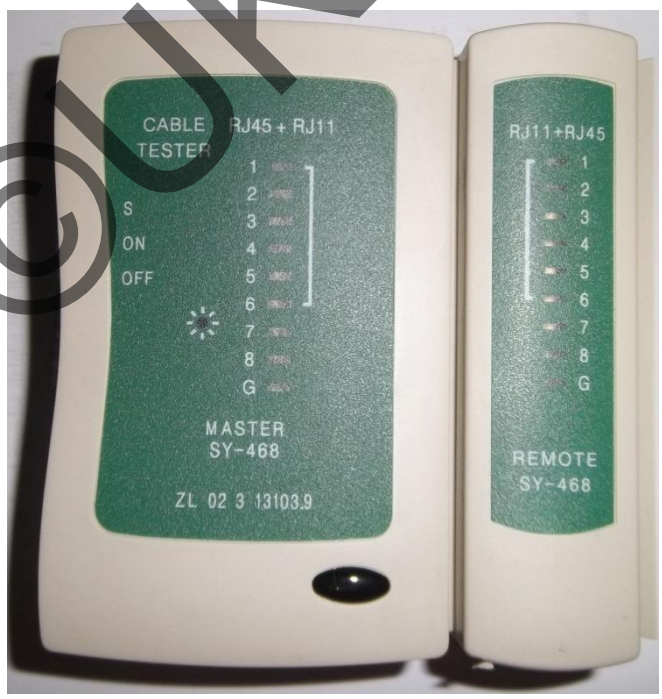
g. Kabel RJ45

Kabel RJ45 adalah kabel yang digunakan untuk menghubungkan antara setiap *node* dengan *switch hub*. Kabel RJ45 yang digunakan adalah kabel RJ45 dengan tipe *straight over*. Spesifikasi kabel RJ45 yang digunakan adalah RJ45 Cat5e. Berikut gambar 3.10 menunjukkan kabel RJ45 yang digunakan pada implementasi teknik transisi.

Gambar 3.10 Kabel RJ45 yang digunakan untuk menghubungkan setiap node

h. LAN Cable Tester

LAN cable tester adalah alat untuk mengecek apakah kabel RJ45 bekerja dengan baik. Alat ini digunakan untuk meminimalkan kemungkinan kabel RJ45 tidak berfungsi sehingga menimbulkan kegagalan konektivitas. Berikut gambar 3.11 yang menunjukkan *lan cable tester*.



Gambar 3.11 LAN cable tester

3.7. Perangkat Lunak yang Digunakan sebagai Pendukung Implementasi Teknik Transisi

Dalam penerapan teknik transisi *ISATAP* dan *dual stack*, penulis menggunakan *software*VMware® Workstation 9.0.1. *Software* ini digunakan untuk mensimulasikan teknik transisi secara virtual. Simulasi secara virtual ini dilakukan untuk mempermudah proses percobaan membangun sebuah sistem teknik transisi. *Software* ini dijalankan pada sebuah PC dengan *prossesor*Intel Core i3 2100 3,1 GHZ, *memory* 8GB, *hardisk* 500 GB dan *operatingsystem* Windows 7 Ultimate 64 bit. Hal ini bisa menghemat waktu penulis dan penggunaan listrik karena *software* ini dapat mensimulasikan 5 PC dan beberapa *switch* secara langsung. Kelima PC dan beberapa *switch*tersebut dapat dikoneksikan menjadi sebuah jaringan. Jadi ketika melakukan *trial and error* dalam membangun sebuah jaringan menjadi lebih mudah. Setelah mendapatkan konfigurasi yang tepat, penulis langsung dapat mengimplementasikan hasil dari percobaan pada simulasi tadi ke perangkat nyata. Dalam melakukan pengukuran kinerja jaringan teknik transisi, penulis melakukannya pada jaringan dengan perangkat nyata. Hal ini dilakukan untuk menjaga keakuratan data yang dihasilkan.

3.8. Konfigurasi pada Windows Server 2003 dan Windows XP dengan Menggunakan Perintah Netsh

Beberapa konfigurasi pada *Windows Server 2003* dan *Windows XP* menggunakan perintah *netsh.*Salah satu konfigurasi yang menggunakan perintah *netsh* adalah konfigurasi *IPv6.*Beberapa konfigurasi *IPv6* menggunakan perintah *netsh*adalah menambah, memodifikasi, atau menghapus alamat dan rute, menambahkan *tunneling 6 over 4* atau *6 in 4*, dan lain-lain.

Berikut cara mengkonfigurasi dan atribut-atribut interface:

a) Buka *start menu* pada halaman *desktop.*
b) Pilih *All programs* lalu pilih *folder accessories.*
c) Pada *folder accessories* pilih *command prompt.*
d) Pada *command prompt,* ketikan *netsh,* kemudian *ENTER.*
e) Ketik *interface ipv6,* kemudian *ENTER.*
f) Ketik *set interface* [interface=]String [[forwarding=]{enabled | disabled}] [[advertise=]{enabled | disabled}] [[mtu=]Integer] [[siteid=]Integer] [[metric=]Integer] [[firewall=]{enabled | disabled}] [[siteprefixlength=]Integer] [[store=]{active | persistent}].

Dimana :

[interface=] String

Konfigurasi ini untuk menentukan nama *interface*

[[forwarding=]{ enabled| disabled}]

Konfigurasi ini untuk menentukan apakah paket-paket yang diterima pada *interface* ini dapat di-*forward* ke *interface* yang lainnya. Konfigurasi secara *default* adalah *disabled.*

[[advertise=]{ enabled| disabled}]

Konfigurasi ini untuk menentukan apakah *router advertisements* akan dikirim pada *interface* ini. Konfigurasinya*default*-nya adalah *disabled.*

[[mtu=] Integer]

Konfigurasi ini untuk menentukan *Maximum Transmission Unit* (MTU) dari *interface* ini. Jika MTU tidak ditentukan, maka MTU *default* dari *link* akan digunakan.

[[siteid=] Integer]

Konfigurasi ini untuk menentukan *site scope* dari *zone identifier*. *Site Identifier* digunakan untuk membedakan *interface-interface* yang terkelompok ke dalam

wilayah administratif yang berbeda yang menggunakan pengalamatan *site-local.*

[[metric=] Integer]
Konfigurasi ini untuk menentukan *metric* dari *interface,* yang telah ditambahkan ke rute *metric* untuk semua *rute* yang melewati *interface.*

[[firewall=]{ enabled| disabled}]
Konfigurasi ini menentukan apakah *firewall*akan diroperasikan atau tidak.

[[siteprefixlength=] Integer]
Konfigurasi ini untuk menentukan panjang *default* dari *prefix* global untuk semua tempat.

[[store=]{ active| persistent}]
Jika pada konfigurasi ini *active* yang dipilih, maka perubahan-perubahan yang telah dilakukan akan hilang pada saat komputer di-*restart*. Jika *persistent* yang dipilih, maka perubahannya adalah permanen. Secara *default*konfigurasinya adalah *Persistent.*

Berikut cara mengkonfigurasi *IPv6* dengan alamat manual.
1. Buka *command prompt.*
2. Pada *command prompt,* ketik *netsh,* kemudian *ENTER.*
3. Ketik *interface ipv6,* kemudian *ENTER.*
4. Ketik *add address* [interface=]String [address=]alamatIPv6.

3.9. Konfigurasi yang Diterapkan pada Topologi Pembanding (IPv4)

Topologi yang menerapkan *IPv4* dibuat sebagai acuan untuk melihat bagaimana perubahan-perubahan yang terjadi ketika menerapkan teknik transisi.

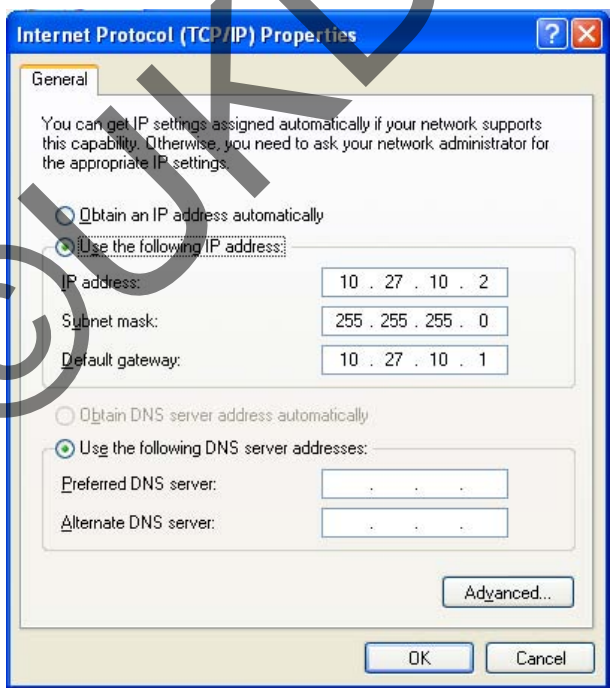
Sisi yang akan diamati perubahannya adalah sisi perangkat yang ditambahkan dan sisi konfigurasi yang digunakan.

3.9.1. Konfigurasi Client 1

Client 1 adalah sebuah *node* yang dirancang untuk menjadi tujuan akhir dari beberapa tes performa jaringan. Adapun konfigurasi yang diterapkan pada *client* 1 adalah sebagai berikut:

a) Melakukan instalasi *operating system Windows XP Professional SP2* di PC *client* 1.

b) Memberikan alamat IP 10.27.10.2, *subnet mask* 255.255.255.0 dan *gateway* 10.27.10.1 pada *interface* yang berhubungan dengan *Router* 1. Berikut gambar 3.12 menunjukkan konfigurasi TCP/IP yang digunakan.

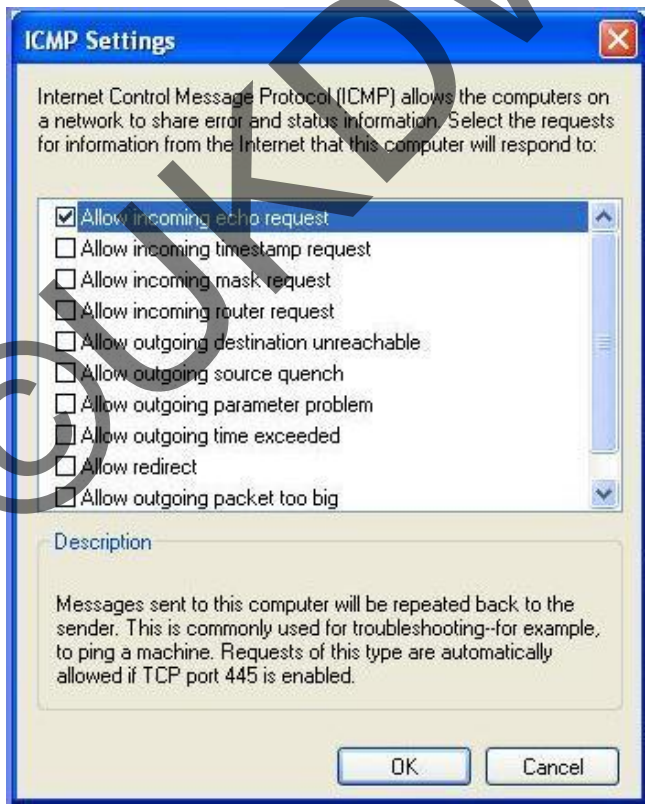


Gambar 3.12 Konfigurasi TCP/IP Client 1

c) Mengkonfigurasi *firewall* agar *client* 1dapat menerima *echo request.*

Secara *default, firewall* pada *WindowsXP Professional* tidak mengizinkan adanya *echo request.* Dengan tidak diizinkannya *echo request* pada konfigurasi *firewall*, maka *node* lain tidak akan bisa melakukan *ping* ke PC *client* 1. Untuk membolehkan *echo request* pada konfigurasi firewall, konfigurasinya adalah sebagai berkut:

1. Klik *start* di halaman *desktop*, lalu pilih menu *Control Panel*, kemudian pilih submenu *Security Center*.
2. Pilih *Windows Firewall*, pada jendela *Windows Firewall* pilih *tab Advanced.*
3. Klik *Settings for ICMP*, kemudian *check Allow incoming echo request*, lalu klik *OK* untuk mengakhiri konfigurasi. Berikut gambar 3.13 menunjukkan setelan yang diterapkan pada *Windows Firewall.*



Gambar 3.13 Setelan ICMPpada Windows Firewall

Selain menggunakan cara di atas, dapat juga dengan cara menonaktifkan fungsi *firewall* pada*Security Center*, namun hal ini sangat tidak disarankan untuk jaringan komputer yang berhubungan dengan *internet* karena alasan keamanan.

3.9.2. Konfigurasi Client 2

Client 2 merupakan sebuah komputer yang digunakan sebagai tempat asal dilakukannya beberapa tes terhadap jaringan. Adapun konfigurasinya adalah:

a) Melakukan instalasi *operating system Windows XP Professional SP2* di PC *client 2*.
b) Memberikan alamat IP 10.27.30.2, *subnet mask* 255.255.255.0 dan *gateway* 10.27.30.1 pada konfigurasi TCP/IP.Cara mengkonfigurasi TCP/IP pada *client 2* sama dengan cara mengkonfigurasi TCP/IP pada *client*1.
c) Mengkonfigurasi *firewall* agar *client*2dapat menerima *echo request.* Cara konfigurasi *firewall* pada *client 2* sama dengan konfigurasi *firewall* pada *client 1*.

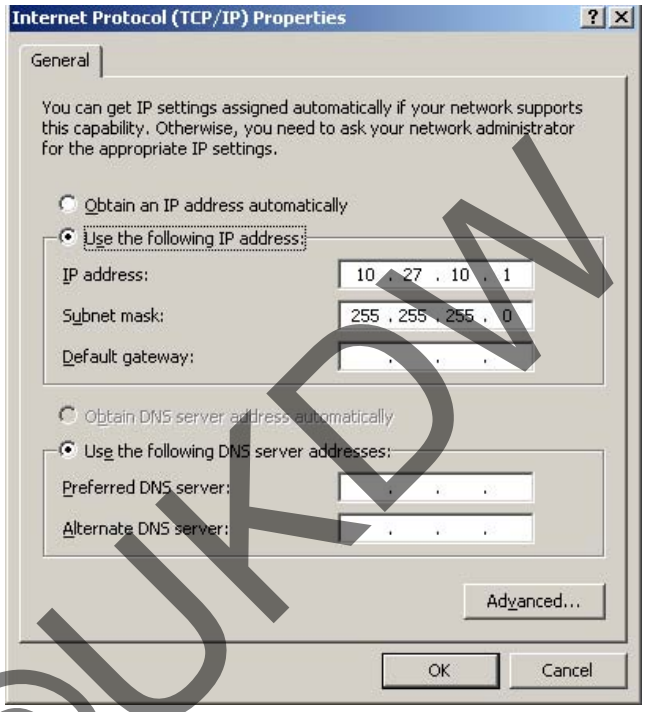
3.9.3. Konfigurasi Router 1

Router 1 adalah komputer yang digunakan sebagai *router* yang menghubungkan *subnet* 10.27.10.0/24 ke *subnet* 10.27.20.0/24. Berikut konfigurasi yang diterapkan pada *router* 1:

a) Melakukan instalasi *Windows Server 2003 Standard Edition SP1 RC2* pada PC yang digunakan untuk *router* 1.
b) Setelah instalasi selesai, *log on* sebagai *Administrator*. Ini dilakukan agar penulis memiliki hak akses penuh terhadap *server* tersebut.
c) Mengubah nama *interface* yang terhubung dengan *subnet* 10.27.10.0/24 menjadi “Koneksi Subnet 1”.

d) Mengubah nama *interface* yang terhubung dengan *subnet* 10.27.20.0/24 menjadi “Koneksi Subnet 2”.

e) Memberikan alamat 10.27.10.1 dan *subnet mask* 255.255.255.0 pada interface Koneksi Subnet 1.*Router* 1 bertindak sebagai *gateway* dari *subnet* 10.27.10.0/0. Berikut gambar 3.14 menunjukkankonfigurasi yang diterapkan pada *interface*tersebut.



Gambar 3.14 Konfigurasi TCP/IP interface Koneksi Subnet 1 pada router 1

f) Memberikan alamat *IP* 10.27.20.1, *subnet mask* 255.255.255.0 dan *gateway* 10.27.20.2pada konfigurasi TCP/IP untuk *interface*Koneksi Subnet 2. Cara mengkonfigurasi TCP/IP pada *interface*Koneksi Subnet 2 sama dengan cara mengkonfigurasi TCP/IP pada *interface* Koneksi Subnet 1.

g) Konfigurasi untuk membolehkan *routing IPv4* antara *subnet* 10.27.10.0/24 dan 10.27.20.0/24. Konfigurasi tersebut dilakukan melalui *registry editor*. Konfigurasinya sebagai berikut:

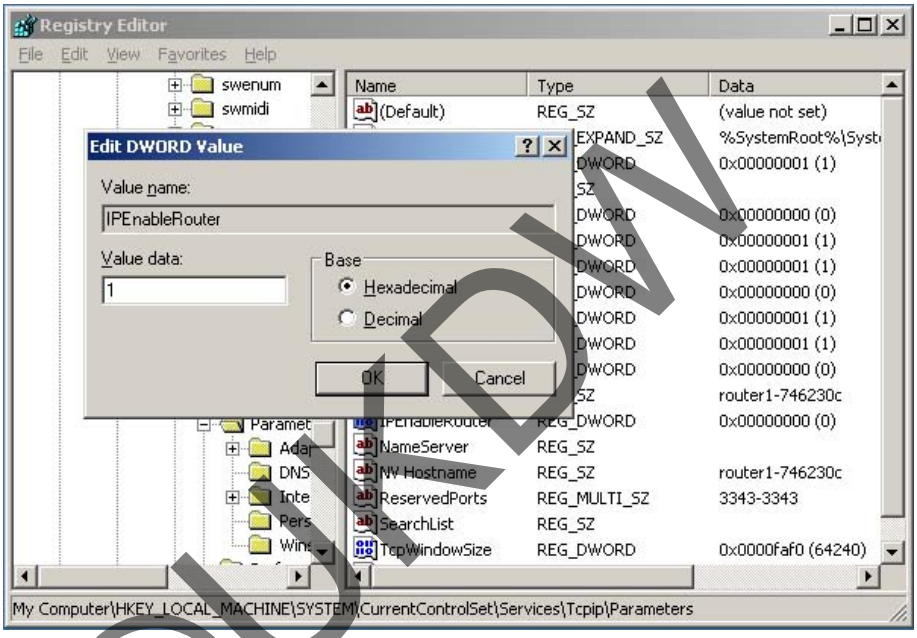
1. Klik *start* pada halaman *desktop*, lalu klik menu *run.*
2. Pada *window run* ketikkan *regedit* pada *text field* di *window run*, lalu *enter*.

3. Kemudian masuk ke bagian:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\. Kemudian klik kanan pada **IPEnableRouter**, lalu pilih kemudian pilih *Modify*.

4. *Set value* data menjadi 1, lalu klik *OK* untuk mengakhiri.

Berikut gambar 3.15 menunjukkan konfigurasi yang dilakukan pada *Registry Editor*.



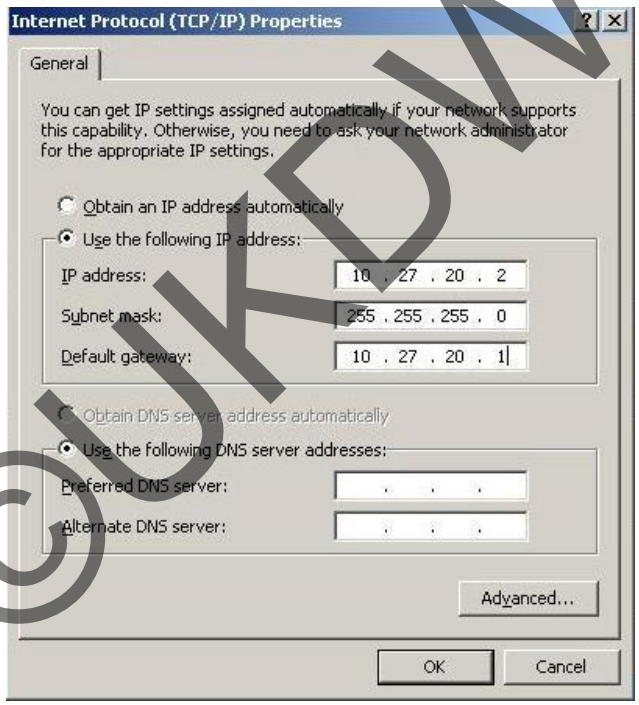
Gambar 3.15 Pengubahan value data pada IP Enable Router

3.9.4. Konfigurasi Router 2

Router 2 adalah komputer yang digunakan sebagai *router* yang menghubungkan *subnet* 10.27.20.0/24 ke *subnet* 10.27.30.0/24. Berikut konfigurasi yang diterapkan pada *router* 2:

a. Melakukan instalasi *Windows Server 2003 Standard Edition SP1 RC2* pada PC yang digunakan untuk *router* 2.

b. Setelah instalasi selesai, *log on* sebagai *Administrator*. Ini dilakukan agar penulis memiliki hak akses penuh terhadap *server* tersebut.
c. Mengubah nama *interface* yang terhubung dengan *subnet* 10.27.20.0/24 menjadi “Koneksi Subnet 2”.
d. Mengubah nama *interface* yang terhubung dengan *subnet* 10.27.30.0/24 menjadi “Koneksi Subnet 3”.
e. Memberikan alamat 10.27.20.2, *subnet mask* 255.255.255.0 dan gateway 10.27.20.1 pada konfigurasi TCP/IP untuk *interface*Koneksi Subnet 2. Berikut gambar 3.16 menunjukkan konfigurasi yang diterapkan pada *interface*Koneksi Subnet 2.



Gambar 3.16 Konfigurasi TCP/IP interface Koneksi Subnet 2 pada router 2

f. Memberikan alamat 10.27.30.1 dan *subnet mask* 255.255.255.0 pada konfigurasi TCP/IP untuk *interface*Koneksi Subnet 2.*Router* 2 bertindak sebagai *gateway* dari *subnet* 10.27.30.0/0.Cara mengkonfigurasi TCP/IP pada *interface*Koneksi Subnet 3 sama dengan cara mengkonfigurasi TCP/IP pada *interface* Koneksi Subnet 2.

g. Konfigurasi untuk membolehkan *routing IPv4* antara *subnet* 10.27.20.0/24 dan 10.27.30.0/24. Konfigurasi tersebut dilakukan melalui *registry editor*. Konfigurasinya sebagai berikut:

1. Klik *start* pada halaman *desktop,* lalu klik menu *run.*
2. Pada *window run* ketikkan *regedit* pada *text field* di *window run,* lalu *enter*.
3. Kemudian masuk ke bagian:

   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\. Kemudian klik kanan pada **IPEnableRouter**, lalu pilih kemudian pilih *Modify*.
4. *Set value* data menjadi 1, lalu klik *OK* untuk mengakhiri.

   Cara mengkonfigurasi *registry editor* pada *router 2* sama dengan konfigurasi *registry editor* pada *router* 1.

3.10. Konfigurasi yang Diterapkan pada Topologi ISATAP

Konfigurasi teknik transisi ISATAP akan diterapkan langsung pada topologi *IPv4*. Hal ini dilakukan untuk melihat adanya perubahan-perubahan pada topologi *IPv4* yang terjadi dalam implementasi teknik ISATAP. Berikut konfigurasi yang digunakan untuk implementasi ISATAP.

3.10.1. Konfigurasi DNS Server

DNS adalah sebuah komputer yang menggunakan *Windows Server 2003 Standard Edition* sebagai *operating system*-nya. *Server* ini memberikan layanan *DNS server* untuk domain **ipv6isatap.skripsi.com**. Untuk mengkonfigurasi DNS untuk layanan ini, lakukan langkah-langkah sebagai berikut:

3.10.1.1. Instalasi OS dan Konfigurasi IP

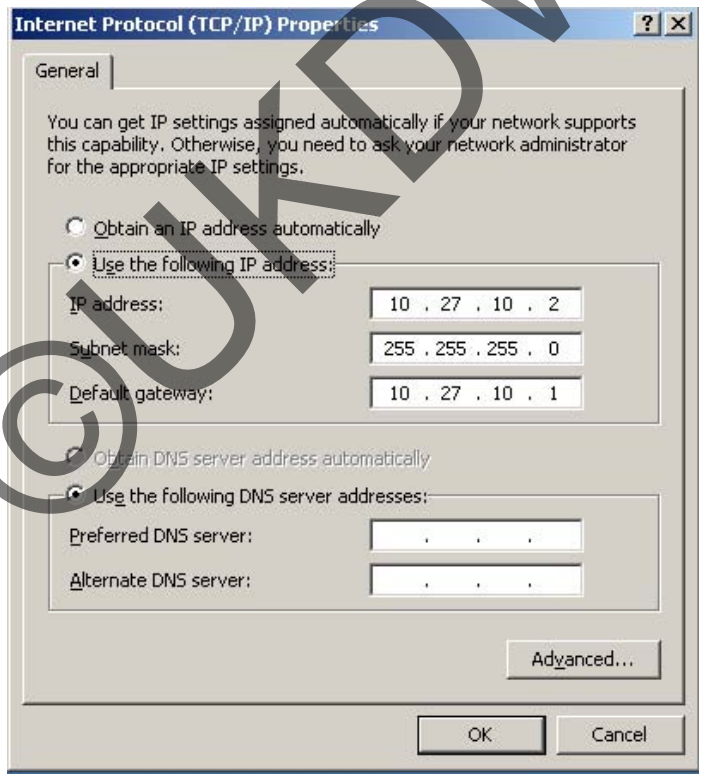
Berikut proses awal dari konfigurasi PC yang digunakan sebagai DNS *server*:

a) Pasang *Windows Server 2003 Standard Edition SP1 RC2* pada sebuah komputer yang telah disiapkan untuk menjadi DNS *server.*

b) Setelah instalasi selesai, *log on* sebagai Administrator. Ini dilakukan agar penulis memiliki hak akses penuh terhadap *server* tersebut.

c) Memberikan alamat 10.27.10.2, *subnet mask* 255.255.255.0 dan *gateway* 10.27.10.1 pada konfigurasi *TCP/IP*.

Berikut gambar 3.17 menunjukkan konfigurasi yang dipasang pada *TCP/IP* pada DNS *server*.



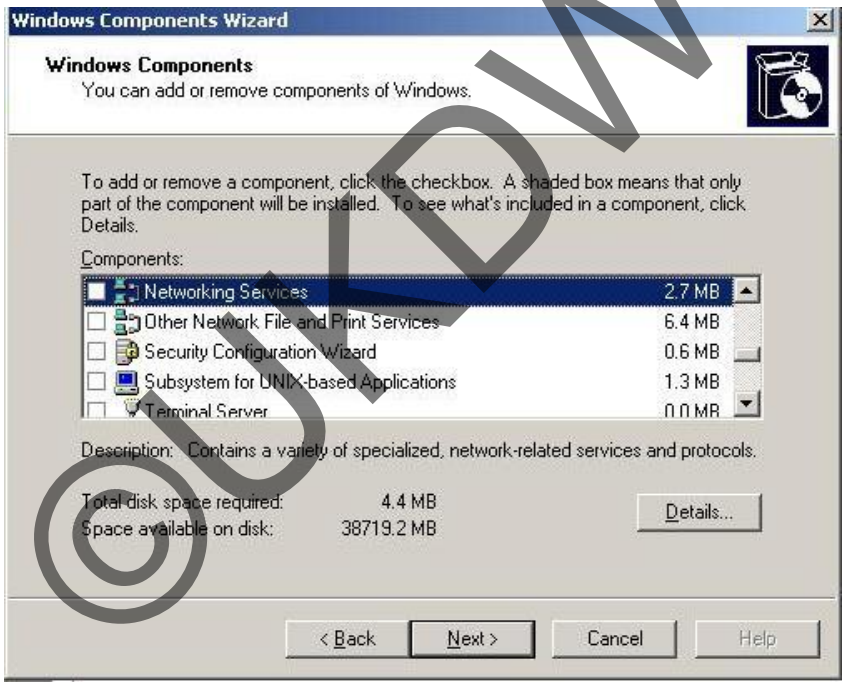
Gambar 3.17 Konfigrurasi TCP/IP DNS

3.10.1.2. Konfigurasi Layanan DNS Server

Pada bagian ini akan dilakukan konfigurasi yang ditujukan untuk memberikan layanan DNS *server.* Adapun konfigurasinya sebagai berikut:

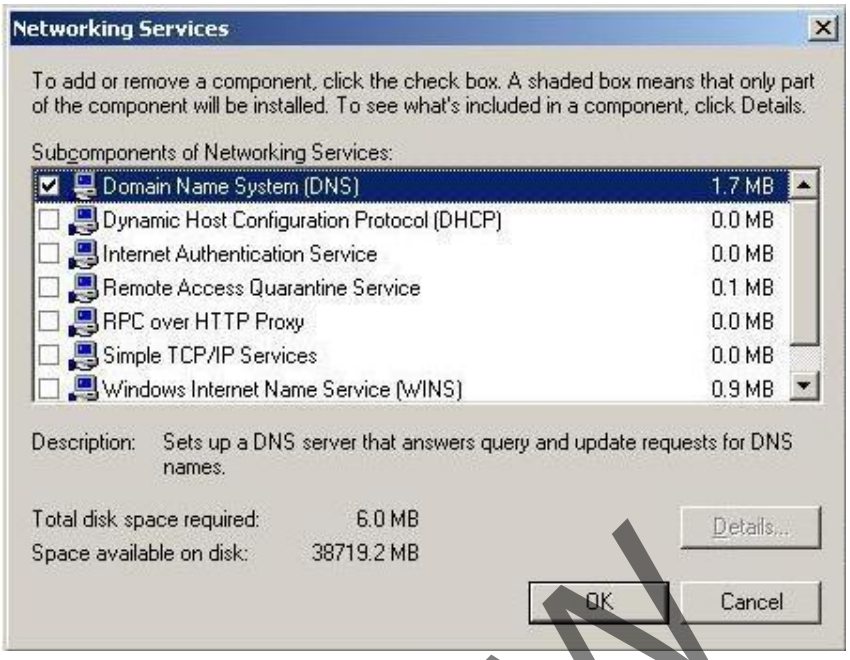
a) Buka *Windows Component Wizard.* Untuk mengakses fasilitas ini, klik *Start,* kemudian pilih menu *Control Panel,* kemudian pilih *Add or Remove Programs,* dan pilih menu Add*/Remove Windows Components*.

b) Pada pilihan menu *Components,* pilih bagian *Networking Services,* namun jangan di-*check,* kemudian klik *Details.* Berikut gambar 3.18 menunjukkan setelan yang diterapkan pada saat pemasangan *Windows Component.*



Gambar 3.18 Windows Component yang dipilih untuk dipasang

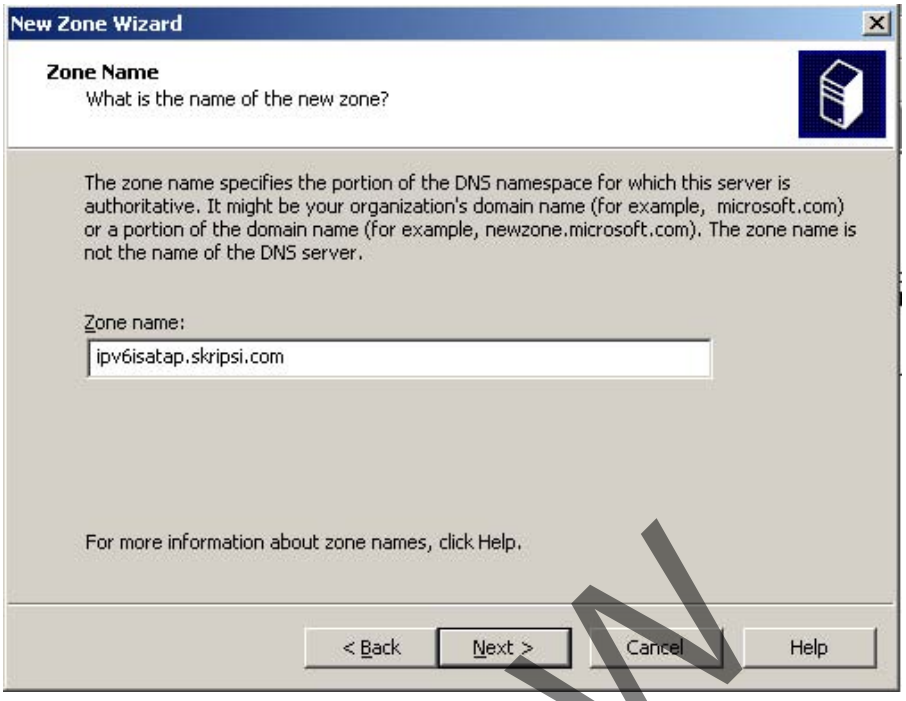
Pada *Subcomponents of Networking Services,* pilih *Domain Name System* (DNS), klik *OK,* kemudian *Next.* Berikut gambar 3.19 menunjukkan komponen yang terdapat pada bagian *Networking Services.*



Gambar 3.19 Bagian dari Networking Service yang dipilih

3.10.1.3. Menentukan Forward Lookup Zone dari DNS Server

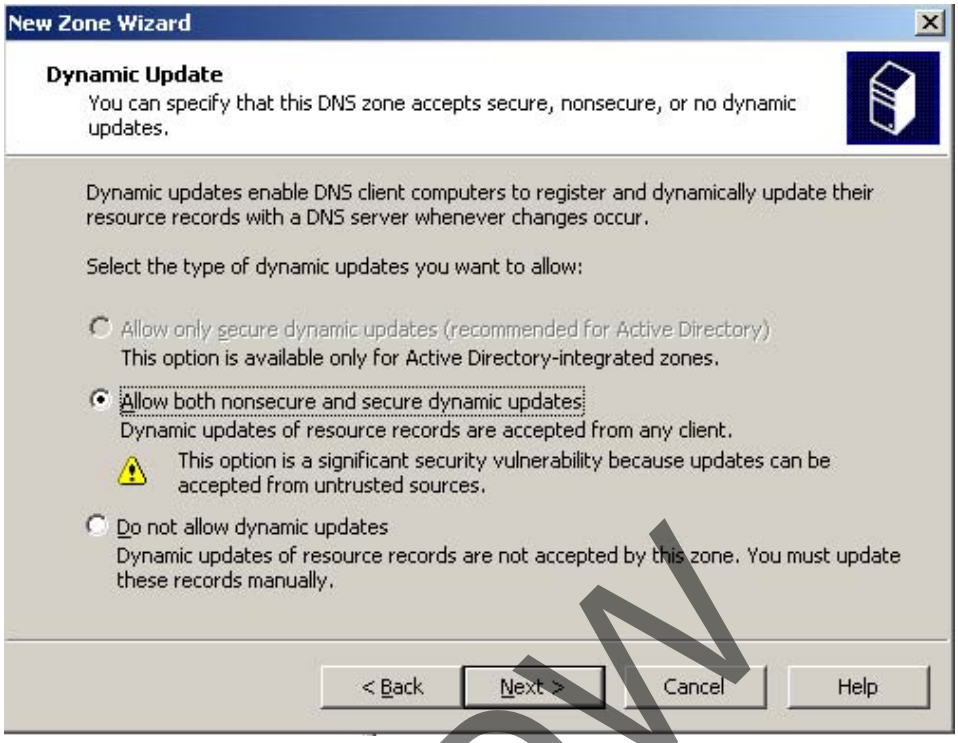
a) Buka menu DNS yang telah dipasang sebelumnya. Untuk mengakses menu ini, klik *Start*, kemudian pilih *Administrative Tools*, lalu pilih menu DNS.
b) Pada *tree console*, klik kanan pada DNS *server* yaitu DNS14, lalu pilih *New Zone.*
c) Setelah itu akan muncul menu *wizard.* Pada jendela *Welcome to the New Zone Wizard*, klik *next* untuk melanjutkan.
d) Pada jendela *Zone Type*, pilih *Primary zone,* lalu lanjutkan dengan klik *next.*
e) Pada jendela *Forward or Reverse Lookup Zone*, pilih *Forward Lookup Zone,* lalu klik *next* untuk melanjutkan.
f) Pada jendela *Zone Name*, ketik *zone name* yang akan diberikan pada DNS. *Zone Name* yang digunakan adalah bebas sesuai dengan yang diingingkan. Dalam penelitian ini, penulis menggunakan *zone name* **ipv6isatap.skripsi.com.**Berikut gambar 3.20 tampilan saat memberikan DNS *name.*



Gambar 3.20 Pemberian Zone Name pada konfigurasi DNS server

g) Pada jendela *Zone File*, secara *default* nama *zone* yang baru akan menjadi **ipv6isatap.skripsi.com.dns.** Lalu klik *next* untuk menyetujui dan melanjutkan.

h) Pada jendela *Dynamic Update*, pilih *Allow both nonsecure and dynamicupdates*. Kemudian klik *next* untuk melanjutkan. Berikut gambar 3.21 tampilan dari jendela *Dynamic Update* saat dilakukan setelan.



Gambar 3.21 Setelan yang diterapkan pada Dynamic Update

Pada akhir *wizard* akan muncul jendela *Completing the New Zone Wizard,* klik *finish* untuk mengakhiri *wizard.*

i) Mengaktifan *IPv6*pada DNS Server.Konfigurasinya dengan cara mengetikkan **netsh interface ipv6 install** pada *command prompt.DNS server* akan mendapatkan *link-local IPv6* secara otomatis.

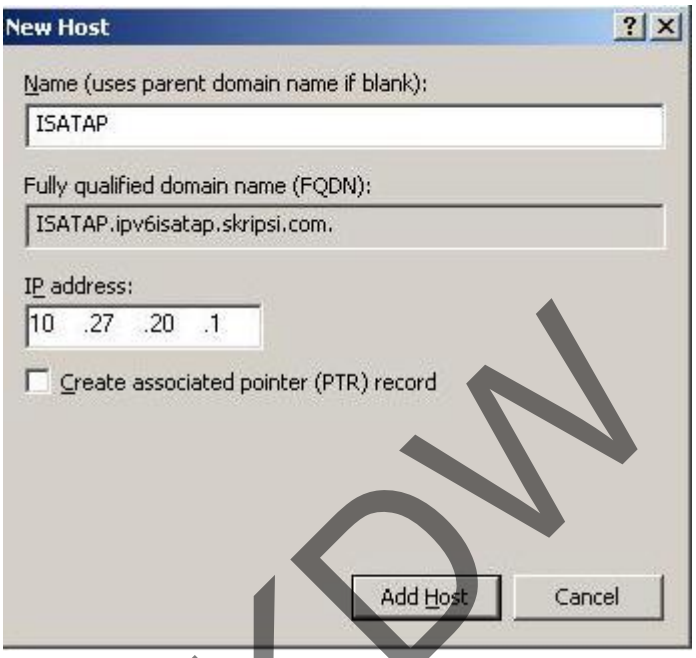
j) Penambahan *record resources*pada layanan DNS. Untuk penambahan *record resources*, klik *start menu* pada halaman *desktop* dan pilih *Administrative Tools* lalu pilih DNS.

k) Pada *console tree,* klik kanan pada **ipv6isatap.skripsi.com.** Lalu pada folder *Forward Lookup Zones* pilih *New Host* (A).

l) Pada *window New Host*, isikan ISATAP pada kolom *name.* Lalu *Fully qualified domain name* (FQDM) akan terisi secara *default* mengikuti *name host* yang diisikan.

m) Pada *IP address box,* isikan alamat *IP interface* pada *router* 1 yang berhubungan dengan *subnet* 10.27.20.0/24. Berikut tampilan setelan pada

penambahan *New host* pada DNS. Lalu klik *Add host* untuk mengakhiri konfigurasi. Berikut gambar 3.22 menunjukkan tampilan konfigurasi penambahan *record resources*pada DNS *server*.



Gambar 3.22 Penambahan New host pada zone ipv6isatap.skripsi.com

3.10.2. Konfigurasi Client 1 dan Client 2

Konfigurasi yang diterapkan pada *Client 1* dan *Client 2*adalah pengaktifan *IPv6*, penambahan akhiran DNS dan khusus *client* 1 *IPv4* yang terpasang pada *interface*-nya akan dihilangkan. Berikut konfigurasi yang digunakan pada *client* 1 dan *client* 2.

a) Mengaktifkan *IPv6* pada *client*

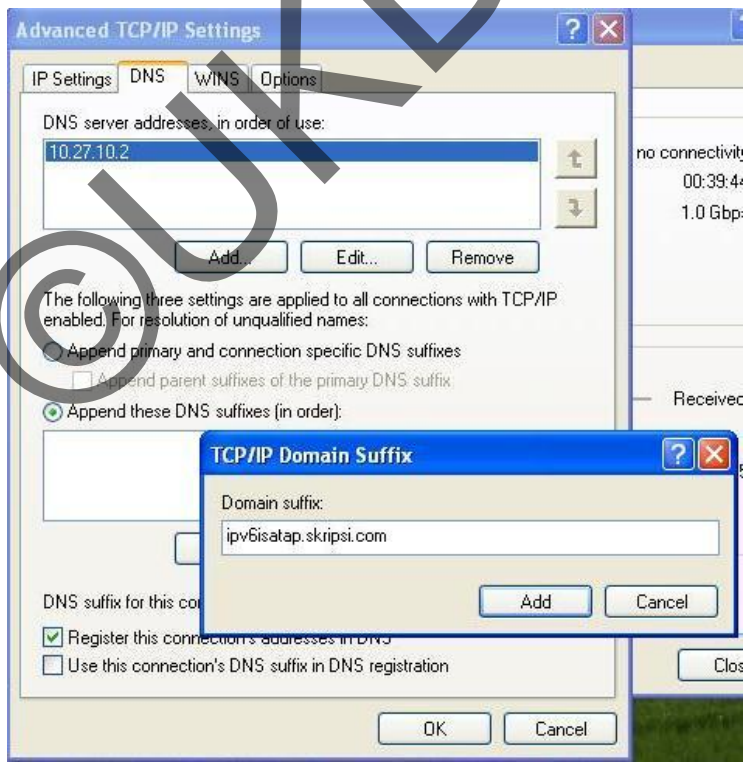
Cara pengaktifan *IPv6* pada *Windows XP* adalah ketikkan **netsh interface ipv6 install** pada *command prompt*, lalu *enter*.

Dengan melakukan konfigurasi ini, *client* 1 dan *client* 2 akan mendapatkan *link-local IPv6 address* secara otomatis. Alamat *link-local IPv6*memungkinkan

client 1 dan *client* 2 berhubungan dengan *node* lain yang masih berada pada satu *subnet*.

b) Menambahkan akhiran DNS (*Append the DNS suffix*).

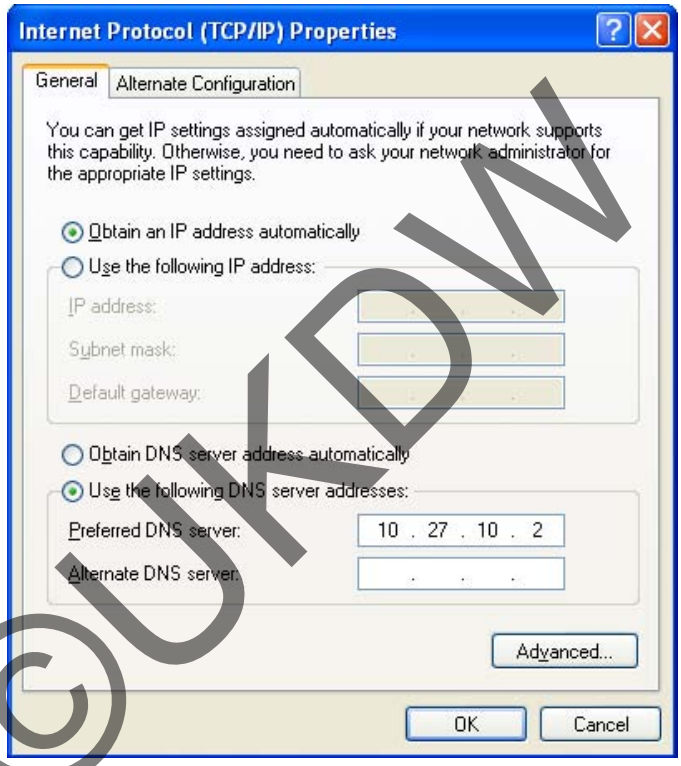
Untuk menambahkan akhiran DNS bisa dilakukan dengan cara berikut. Klik *start*, lalu pilih *Control Panel*, kemudian pilih *Networkand Internet Connections*, lalu pilih *Network Connections.* Pada menu *network connections* tadi, klik kanan pada *interface* yang ingin dikonfigurasi, lalu pilih *Properties.* Pada jendela *General tab*, pilih *Internet Protocol* (*TCP/IP*), kemudian klik *Properties.* Pada jendela *Internet Protocol* (*TCP/IP*) *Properties*, pilih *Advanced*, lalu klik tab DNS. Pada tab DNS tersebut, klik *radiobutton Append these DNS suffixes* (*in order*), lalu klik *add* untuk menambahkan akhiran DNS. Pada Domain *suffix*, ketikan **ipv6isatap.skripsi.com**, lalu klik *add* untuk mengakhiri konfigurasi. Berikut gambar 3.23 menunjukkan setelan yang digunakan pada saat penambahan DNS *suffix*.



Gambar 3.23 Penambahan DNS Suffix pada client 1

c) Menghilangkan konfigurasi *IPv4* pada *client* 1

Dihapusnya konfigurasi *IPv4* pada *client* 1 bertujuan agar *client* 1 benar-benar hanya berkomunikasi dengan *IPv6.* Alamat yang dihilangkan adalah alamat *IPv4* yang diberikan pada *interface*, sedangkan pada DNS tidak akan dihapus. Hal ini dikarenakan *client* 1 perlu berhubungan dengan DNS *server* guna mendapatkan informasi *gateway* dari *router* ISATAP. Berikut gambar 3.24 menunjukkan konfigurasi yang diterapkan pada TCP/IP *client* 1.



Gambar 3.24 Konfigurasi yang diterapkan pada TCP/IP client 1

3.10.3. Konfigurasi Router 1 dan 2

Konfigurasi pada *router* ISATAP bertujuan untuk memberitahukan keberadaanya (*advertise*) dan mendistribusikan *prefix* alamat, membolehkan alamat global ISATAP untuk dikonfigurasi, meneruskan paket-paket *IPv6* antara *host* ISATAP yang berada pada *intranet IPv4* dan *host IPv6* yang berada di luar

jaringan.Semua konfigurasi yang diterapkan melalui program *command prompt*. Berikut konfigurasi yang diterapkan pada kedua *router* ISATAP.

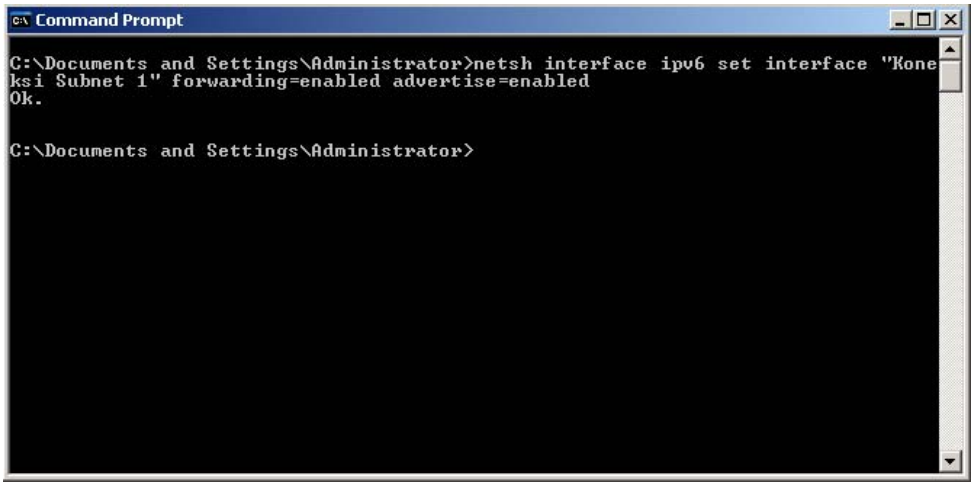
3.10.3.1. Konfigurasi pada Router 1 (Router ISATAP)

a) Mengaktifkan protokol *IPv6* pada *router* 1.

Pengaktifan protokol *IPv6* pada *router* 1 sama dengan pada *DNS server*, *client* 1 dan 2. Konfigurasinya dengan cara mengetikkan **netsh interface ipv6 install** pada *command prompt.* Dengan mengaktifkan perintah ini, maka *router* 1 akan mendapat alamat *link-local IPv6* secara otomatis pada setiap *interface* yang ada di *router* tersebut. Alamat *link-local* ini akan digunakan pada konfigurasi *routing static* antara *router* 1 dan *router* 2.

b) Mengkonfigurasi agar *router* dapat meneruskan paket *IPv6.*

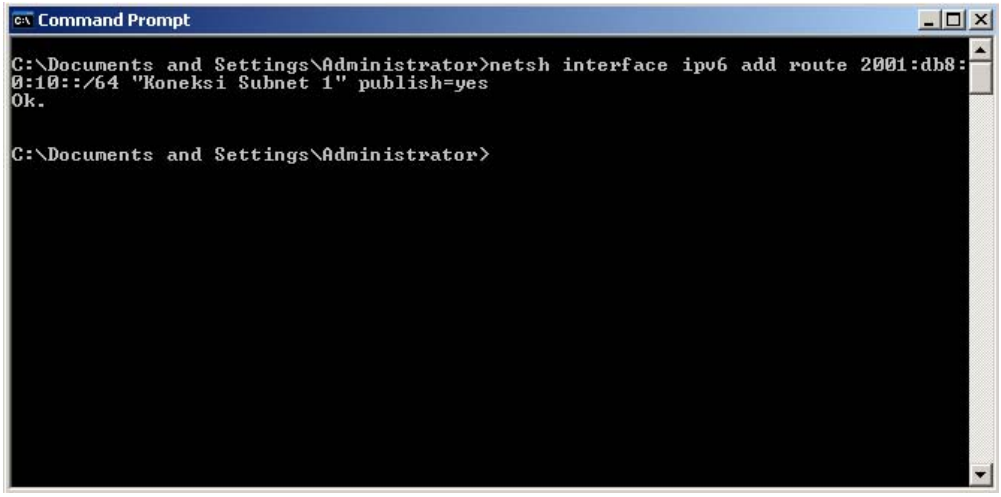
Caranya dengan mengetikkan **netsh interface ipv6 set interface** "*interface yang akan digunakan untuk meneruskan paket IPv6*" **forwarding=enabled advertise=enabled.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 1, maka konfigurasinya adalah **netsh interface ipv6 set interface "Koneksi Subnet 1" forwarding=enabled advertise=enabled.** Konfigurasi ini hanya agar *router* meneruskan paket *IPv6*ke *interface* Koneksi Subnet 1namun, tidak memberikan alamat pada *node* yang terhubung. Pada *interface* Koneksi Subnet 2 tidak dilakukan konfigurasi seperti pada *interface* Koneksi Subnet 1. Hal ini dikarenakan agar antara jaringan *IPv6* dan jaringan *IPv4* benar-benar terpisah. Ini juga akan menjadi penanda apakah konektivitas antara *IPv6* dan *IPv4* sudah terbuat dengan baik. Berikut gambar 3.25 tampilan konfigurasi untuk meneruskan paket *IPv6* melalui *router* 1.



Gambar 3.25 Konfigurasi untuk meneruskan paket IPv6 pada teknik ISATAP

c) Mengkonfigurasi agar router dapat memberikan alamat global *IPv6* pada *subnet* yang terhubung pada *interface* yang diinginkan.

Caranya dengan mengetikkan **netsh interface ipv6 add route** "*prefix yang akan digunakan*""*interface yang akan digunakan untuk menyebarkan alamat global IPv6*" **publish=yes.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 1. Untuk *interface* Koneksi Subnet 1 akan digunakan *subnet prefix* 2001:DB8:0:10::/64, maka konfigurasinya adalah **netsh interface ipv6 add route 2001:DB8:0:10::/64 "Koneksi Subnet 1" publish=yes.** Konfigurasi ini akan memberikan alamat global *IPv6* dengan *prefix*2001:DB8:0:10::/64 pada setiap *node* yang terhubung dengan *interface* Koneksi Subnet 1. Konfigurasi ini juga memberikan alamat global *IPv6* pada setiap *interface* yang ada di *router* 1. Berikut gambar 3.26 tampilan konfigurasi pemberian alamat global.



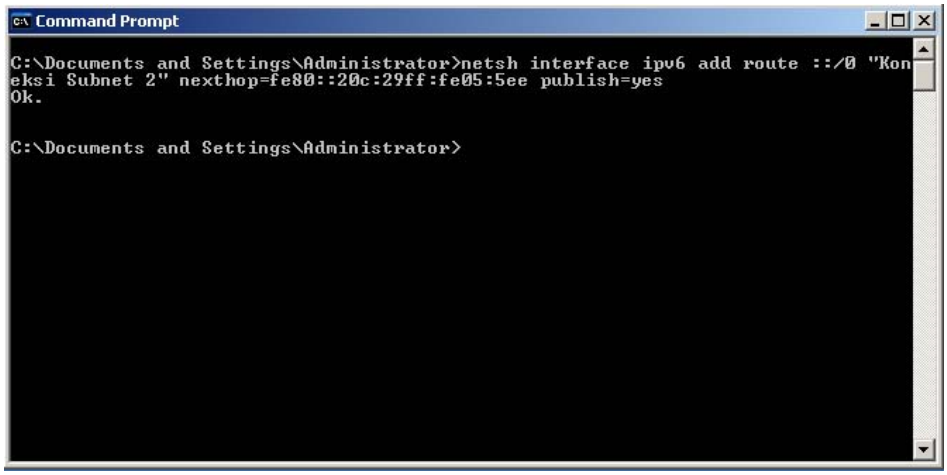
Gambar 3.26 Pemberian alamat global pada interface Koneksi Subnet 1

d) Mengkonfigurasi *routing static* pada *router* agar *IPv6* pada *router* 1 dapat terhubung ke *router* 2.

Cara mengkonfigurasinya adalah dengan mengetikkan**netsh interface ipv6 add route ::/0** "*interface yang menghubungkan router 1 dan router 2*"**nexthop=***"alamat link-local pada interface di router 2 yang berhubungan dengan router 1"* **publish=yes.** Pada topologi ini, alamat *link-local* pada *interface* di *router* 2 yang berhubungan dengan*router* 1 adalah FE80::20C:29FF:FE05:5EE, maka konfigurasinya adalah

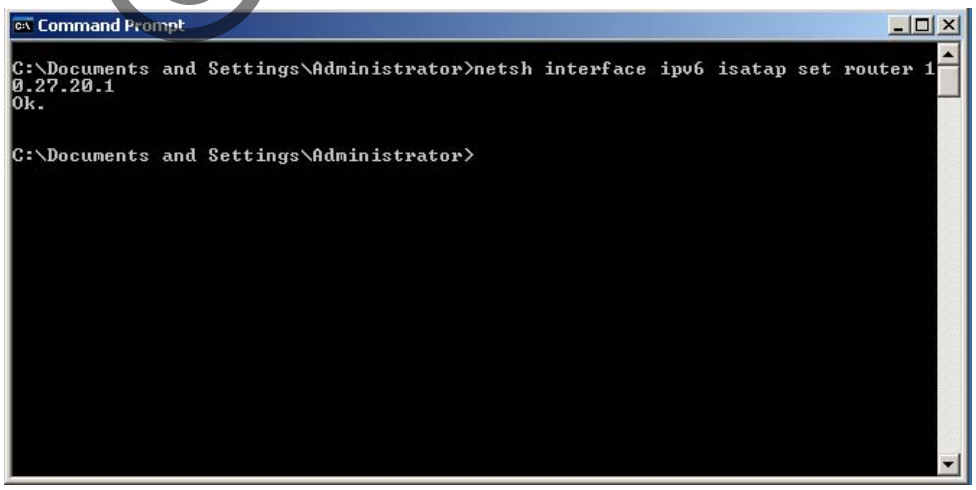
netsh interface ipv6 add route ::/0 "Koneksi Subnet 2" nexthop=FE80::20C:29FF:FE05:5EE publish=yes.

Berikut gambar 3.27 menunjukkan konfigurasi penambahan *routing static* pada *router* 1.



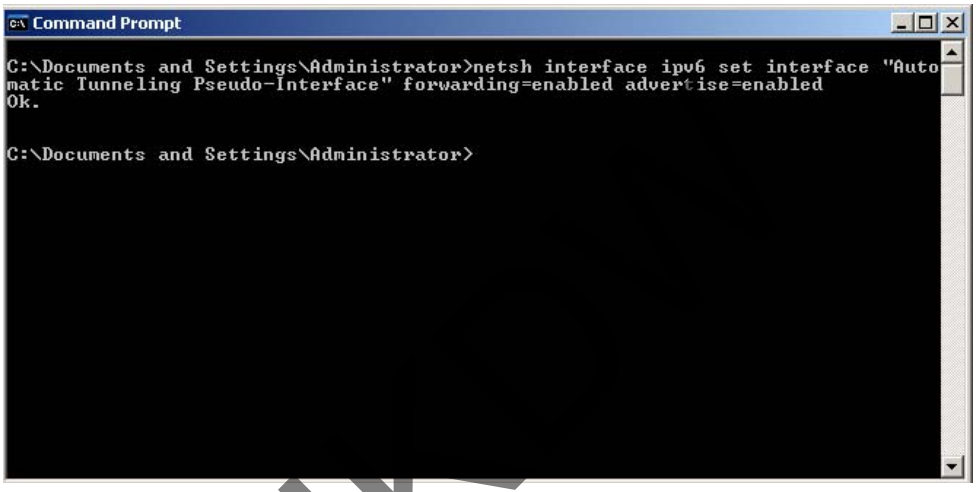
Gambar 3.27 Pemberian routing static pada router ISATAP

e) Mengkonfigurasi agar fungsi *Automatic Tunneling Pseudo* menjadi aktif. Cara mengkonfigurasinya sebagai berikut: **netsh interface ipv6 isatap set router***"alamat IPv4 pada interface yang berhubungan dengan subnet yang tidak diberikan konfigurasi untuk melanjutkan paket IPv6".*Dalam hal ini *interface* yang tidak diberikan konfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 2, dan alamat pada Koneksi Subnet 2 adalah 10.27.20.1. Jadi konfigurasinya adalah **netsh interface ipv6 isatap set router 10.27.20.1.** Konfigurasi ini akan mengaktifkan teknik ISATAP pada *router* tersebut dan alamat 10.27.20.1 sebagai pintu masuk dari proses *tunneling.* Berikut gambar 3.28 menunjukkan konfigurasi pengaktifan fungsi ISATAP.



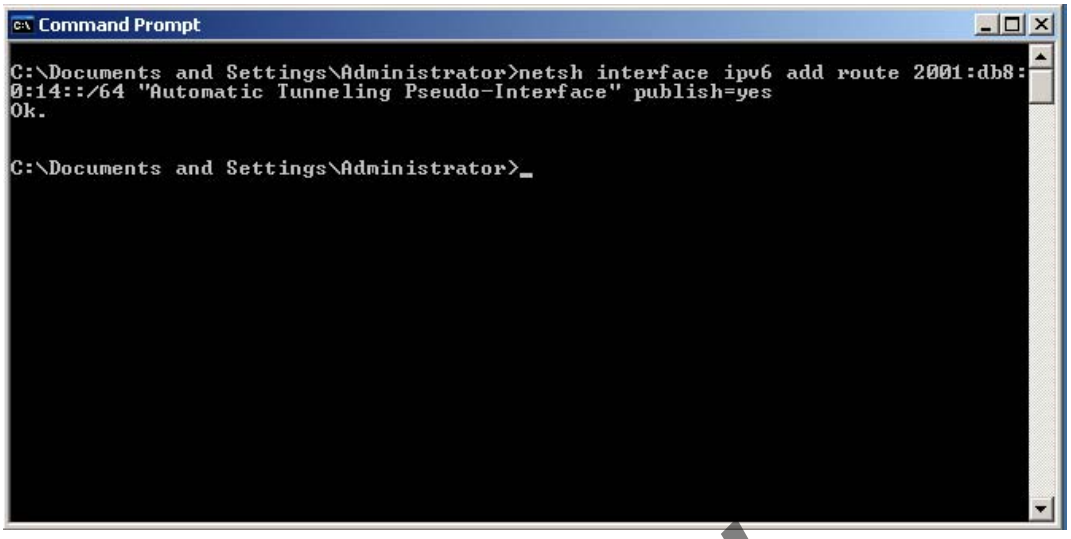
Gambar 3.28 Konfigurasi pengaktifan fungsi ISATAP pada Router 1

f) Mengkonfigurasi agar fungsi *forwarding* dan *advertising* pada*interface Automatic Tunneling Pseudo*. Konfigurasinya adalah sebagai berikut: **netsh interface ipv6 set interface "Automatic Tunneling Pseudo-Interface" forwarding=enabled advertise=enabled.** Konfigurasi ini berfungsi untuk meneruskan paket *IPv4* melalui tunneling ke *network IPv6.* Berikut gambar 3.29 menunjukkan konfigurasi untuk mengaktifkan *interface Automatic Tunneling Pseudo*.



Gambar 3.29 Konfigurasi pengaktifan interface Automatic Tunneling Pseudo

g) Mengkonfigurasi untuk menambahkan *route* untuk *subnetprefix* dari jaringan *IPv4* ke *interface Automatic Tunneling Pesudo*. Adapun cara konfigurasinya sebagai berikut: **netsh interface ipv6 add route** *"subnet prefix yang digunakan"***Automatic Tunneling Pseudo-Interface publish=yes.** *Subnet prefix* yang digunakan adalah 2001:DB8:0:14::/64, maka konfigurasinya menjadi **netsh interface ipv6 add route 2001:DB8:0:14::/64 "Automatic Tunneling Pseudo-Interface" publish=yes.** Konfigurasi ini menambahkan *route* dengan *subnet*2001:DB8:0:14::/64 yang melalui *interface AutomaticTunnelingPseudo-Interface.* Berikut gambar 3.30 menunjukkan konfigurasi penambahan *subnet prefix*.



Gambar 3.30 Konfigurasi penambahan route untuk pemberian subnet prefix

3.10.3.2. Konfigurasi Router 2

Pada *router* 2, semua *interface* tidak dilakukan konfigurasi untuk dapat melanjutkan paket *IPv6.* Hal ini dikarenakan untuk memisahkan jaringan *IPv6* dan jaringan *IPv4*. Pada topologi ini, *router* 2 berada pada lingkup jaringan *IPv4*. Pengaktifan protokol *IPv6* pada *router* 2 sama dengan pada *DNS server*, *client* 1 dan 2 dan *router* 1. Konfigurasinya dengan cara mengetikkan **netsh interface ipv6 install** pada *command prompt.* Dengan mengaktifkan konfigurasi ini, maka *router*2 akan mendapat alamat *link-local IPv6* secara otomatis pada setiap *interface* yang ada di *router* tersebut. Alamat *link-local* ini akan digunakan pada konfigurasi *routing static* antara *router* 2 dan *router* 1.

3.11. Konfigurasi yang Diterapkan pada Topologi Dual Stack

Seperti halnya penerapan teknik ISATAP yang melanjutkan dari penerapan topologi *IPv4,* teknik *dual stack* juga akan dilakukan dengan cara melanjutkan konfigurasi pada topologi *IPv4.* Hal ini dilakukan untuk melihat

perubahan-perubahan yang terjadi ketika menerapkan teknik transisi *dual stack.*Berikut konfigurasi yang digunakan untuk implementasi *dual stack.*

3.11.1. Konfigurasi Client 1 dan Client 2

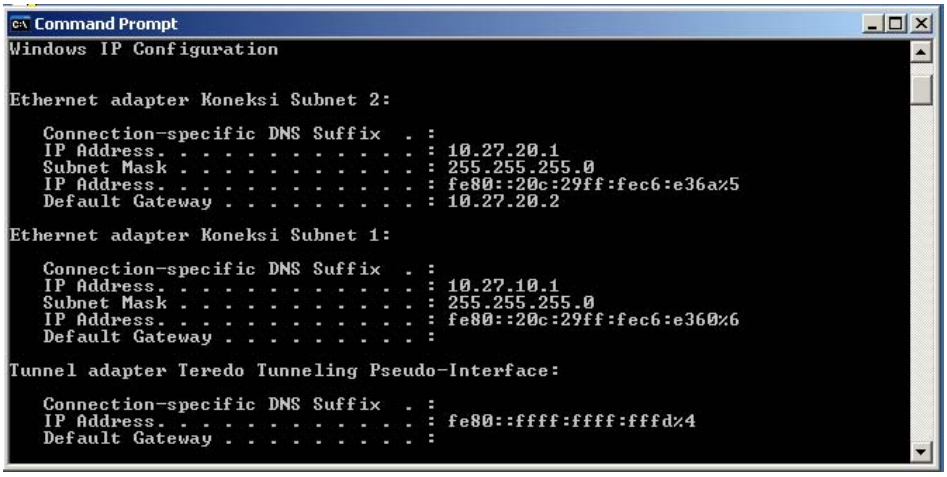
Konfigurasi yang diterapkan pada *Client 1* dan *Client 2*adalah pengaktifan *IPv6.* Adapun konfigurasi yang digunakan untuk mengaktifkan *IPv6* pada *client* 1 dan *client*adalah dengan menjalankan perintah**netsh interface ipv6 install** pada *command prompt,* lalu *enter.*

Dengan melakukan konfigurasi ini, *client* 1 dan *client* 2 akan mendapatkan *link-local IPv6 address* secara otomatis. Alamat *link-local IPv6*memungkinkan *client* 1 dan *client* 2 berhubungan dengan *node* lain yang masih berada pada satu *subnet.*

3.11.2. Konfigurasi pada Router 1

a) Mengaktifkan protokol *IPv6* pada *router* 1.

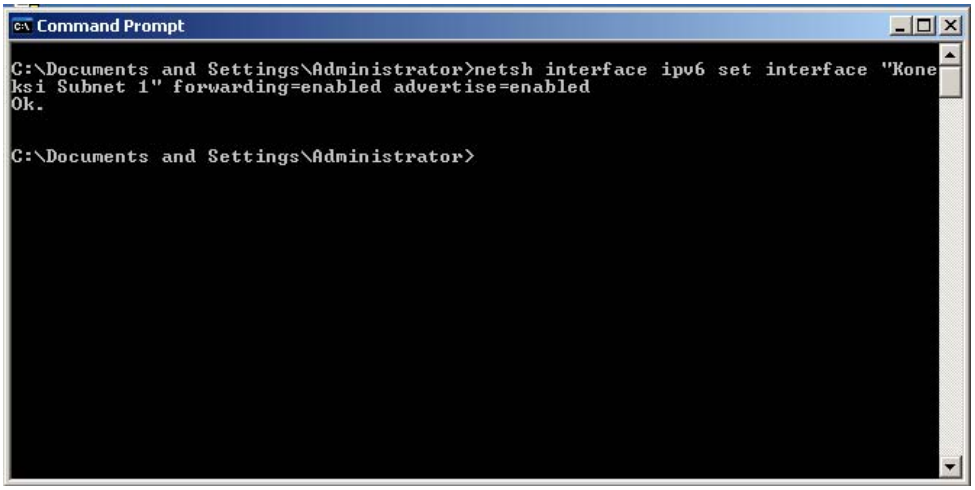
Konfigurasinya dengan cara mengetikkan **netsh interface ipv6 install** pada *command prompt.* Dengan mengaktifkan konfigurasi ini, maka *router* 1 akan mendapat alamat *link-local IPv6* secara otomatis pada setiap *interface* yang ada di *router* tersebut. Alamat *link-local* ini akan digunakan pada konfigurasi *routingstatic* antara *router* 1 dan *router* 2. Berikut gambar 3.31hasil dari konfigurasi IP di *router* 1.



Gambar 3.31 Konfigurasi IP di Router 1

b) Mengkonfigurasi agar *router* dapat meneruskan paket *IPv6.*

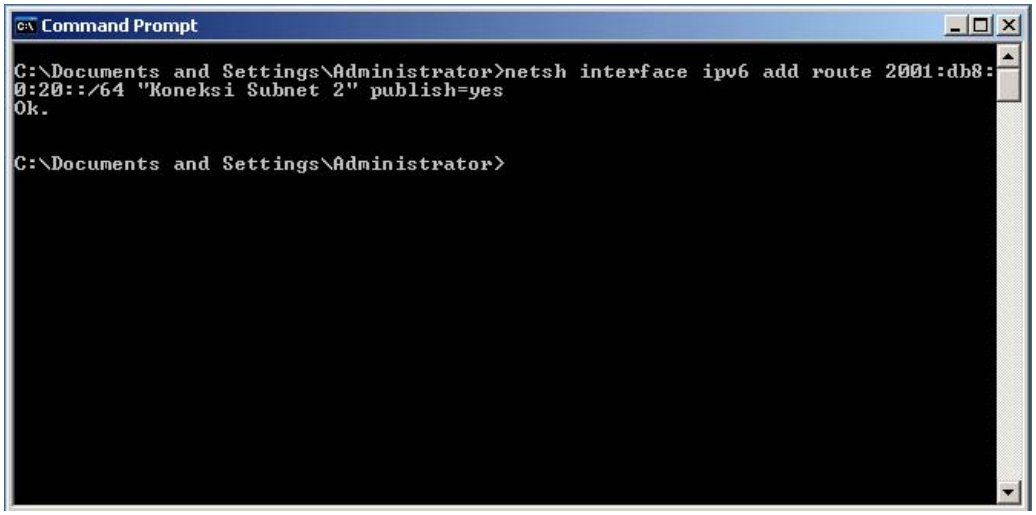
Caranya dengan mengetikkan **netsh interface ipv6 set interface** *"interface yang akan digunakan untuk meneruskan paket IPv6"* **forwarding=enabled advertise=enabled.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 1 dan Koneksi Subnet 2, maka konfigurasinya untuk *interface* Koneksi Subnet 1 adalah **netsh interface ipv6 set interface "Koneksi Subnet 1" forwarding=enabled advertise=enabled** dan untuk *interface* Koneksi Subnet 2 adalah **netsh interface ipv6 set interface "Koneksi Subnet 2" forwarding=enabled advertise=enabled.** Konfigurasi ini hanya agar *router*meneruskan paket *IPv6* ke *interface*Koneksi Subnet 1 dan Koneksi Subnet2namun tidak memberikan alamat pada *node* yang terhubung. Berikut gambar 3.32 menunjukkan konfigurasi untuk meneruskan paket *IPv6* melalui *interface* Koneksi Subnet 1.



Gambar 3.32 Konfigurasi untuk meneruskan paket IPv6 pada interface Koneksi Subnet 1 pada teknik dual stack

c) Mengkonfigurasi agar *router* dapat memberikan alamat global *IPv6* pada *subnet* yang terhubung pada *interface* yang diinginkan.

Caranya dengan mengetikkan **netsh interface ipv6 add route** *"prefix yang akan digunakan""interface yang akan digunakan untuk menyebarkan alamat global IPv6"* **publish=yes.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 1 dan Koneksi Subnet 2. Untuk *interface* Koneksi Subnet 1 akan digunakan *subnet prefix*2001:DB8:0:10::/64, maka konfigurasinya adalah **netsh interface ipv6 add route 2001:DB8:0:10::/64 "Koneksi Subnet 1" publish=yes.** Untuk *interface* Koneksi Subnet 2 akan digunakan *subnet prefix*2001:DB8:0:20::/64, maka konfigurasinya adalah **netsh interface ipv6 add route 2001:DB8:0:20::/64 "Koneksi Subnet 2" publish=yes.** Konfigurasi ini akan memberikan alamat global *IPv6* dengan *prefix*2001:DB8:0:10::/64 pada setiap *node* yang terhubung dengan *interface* Koneksi Subnet 1 dan *prefix*2001:DB8:0:20::/64 pada setiap *node* yang terhubung dengan *interface* Koneksi Subnet 2. Konfigurasi ini juga memberikan alamat global *IPv6* pada setiap *interface* yang ada di *router* 1. Berikut gambar 3.33 tampilan konfigurasi pemberian alamat global pada *interface* Koneksi Subnet 2.



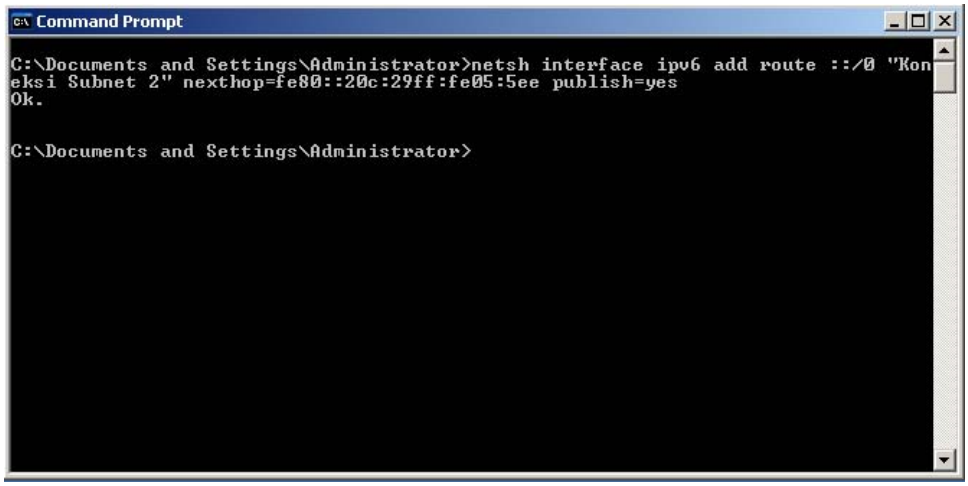
Gambar 3.33 Pemberian alamat global pada interface Koneksi Subnet 2

d) Mengkonfigurasi *routing static* pada *router* agar *IPv6* pada *Router* 1 dan *Router* 2 dapat terhubung.

Cara mengkonfigurasinya adalah dengan mengetikkan **netsh interface ipv6 add route ::/0** *"interface yang menghubungkan router 1 dan router 2"***nexthop=***"alamat link-local pada interface di router 2 yang berhubungan dengan router 1"* **publish=yes**.

Pada topologi ini, alamat *link-local* pada *interface* di *router* 2 yang berhubungan dengan *router* 1 adalah FE80::20C:29FF:FE05:5EE, maka konfigurasinya adalah **netsh interface ipv6 add route ::/0 "Koneksi Subnet 2" nexthop=FE80::20C:29FF:FE05:5EE publish=yes.**

Berikut gambar 3.34 menunjukkan konfigurasi penambahan *routing static* pada *router* 1.

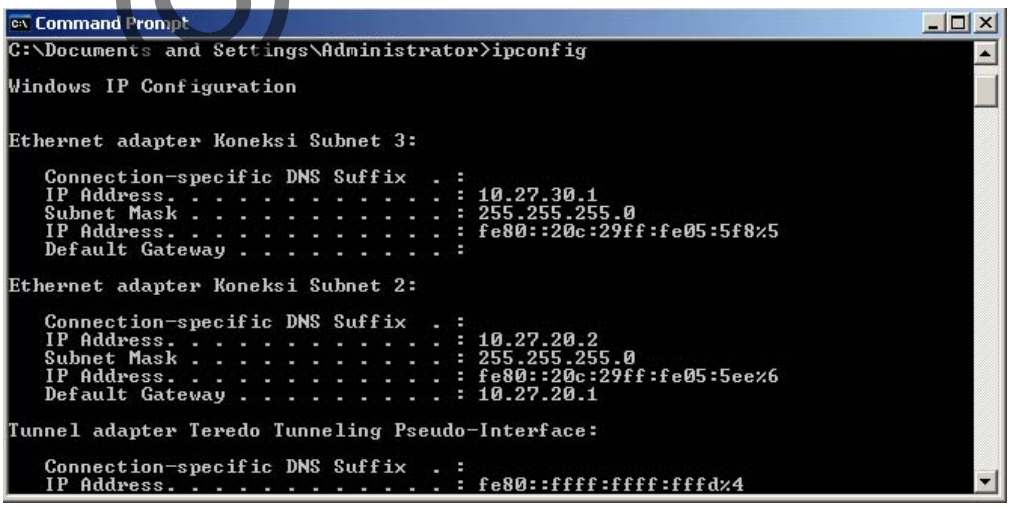


Gambar 3.34 Pemberian routing static pada Router 1 pada teknik dual stack

3.11.3. Konfigurasi pada Router 2

a) Mengaktifkan protokol *IPv6* pada *router* 1.

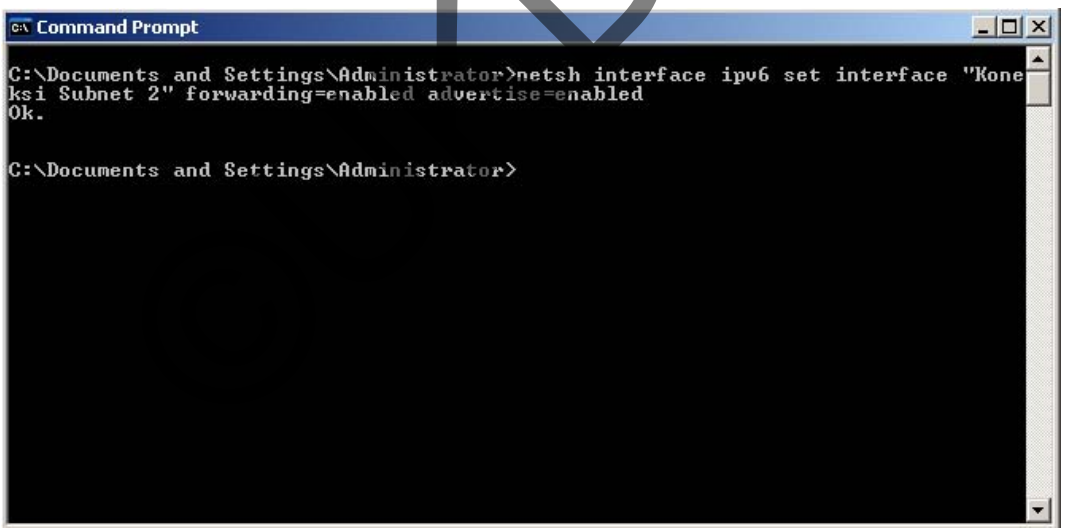
Konfigurasinya dengan cara mengetikkan **netsh interface ipv6 install** pada *command prompt.* Dengan mengaktifkan konfigurasi ini, maka *router* 2 akan mendapat alamat *link-local IPv6* secara otomatis pada setiap *interface* yang ada di *router* tersebut. Alamat *link-local* ini akan digunakan pada konfigurasi *routing static* antara *router* 2 dan *router* 1. Berikut gambar 3.35hasil dari konfigurasi IP di *router* 2.



Gambar 3.35 Konfigurasi IP di router 2 pada teknik transisi dual stack

b) Mengkonfigurasi agar *router* dapat meneruskan paket *IPv6.*

Caranya dengan mengetikkan **netsh interface ipv6 set interface** *"interface yang akan digunakan untuk meneruskan paket IPv6"***forwarding=enabled advertise=enabled.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 2 dan Koneksi Subnet 3, maka konfigurasinya untuk *interface* Koneksi Subnet 2 adalah **netsh interface ipv6 set interface "Koneksi Subnet 2" forwarding=enabled advertise=enabled** dan untuk *interface* Koneksi Subnet 3 adalah **netsh interface ipv6 set interface "Koneksi Subnet 3" forwarding=enabled advertise=enabled.** Konfigurasi ini hanya agar *router* meneruskan paket *IPv6* ke *interface*Koneksi Subnet 2 dan Koneksi Subnet 3 namun tidak memberikan alamat pada *node* yang terhubung. Berikut gambar 3.36 menunjukkan konfigurasi untuk meneruskan paket *IPv6* melalui melalui *interface* Koneksi Subnet 2.



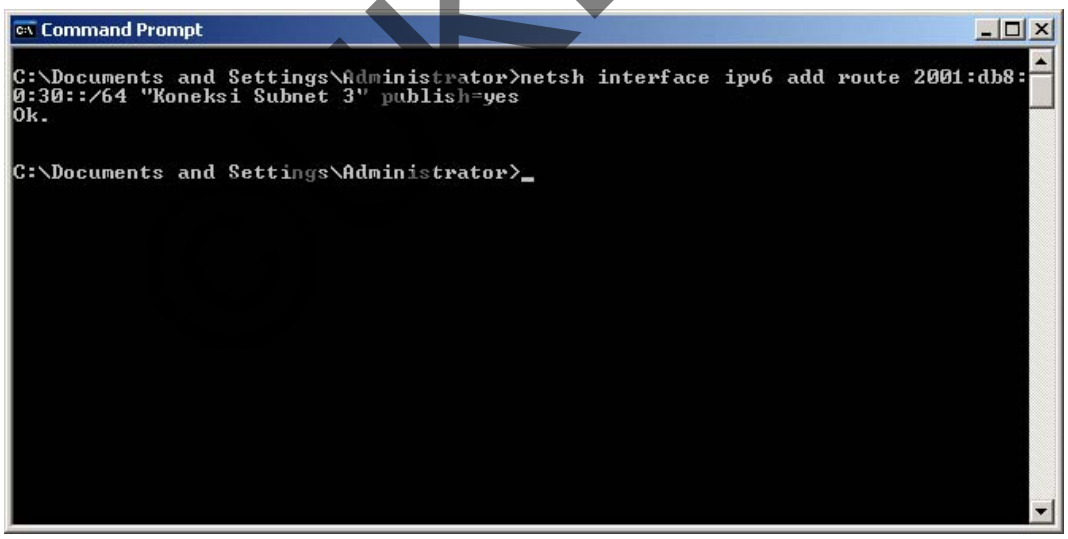
Gambar 3.36 Konfigurasi untuk meneruskan paket IPv6 pada interface Koneksi Subnet 2

c) Mengkonfigurasi agar *router* dapat memberikan alamat global *IPv6* pada *subnet* yang terhubung pada *interface* yang diinginkan.

Caranya dengan mengetikkan **netsh interface ipv6 add route** "*prefix yang akan digunakan" "interface yang akan digunakan untuk meneruskan*

paket IPv6" **publish=yes.** Pada *router* ini, *interface* yang akan dikonfigurasi untuk melanjutkan paket *IPv6* adalah *interface* Koneksi Subnet 2 dan Koneksi Subnet 3. Untuk *interface* Koneksi Subnet 2 akan digunakan *subnetprefix*2001:DB8:0:20::/64, maka konfigurasinya adalah **netsh interface ipv6 add route 2001:DB8:0:20::/64 "Koneksi Subnet 2" publish=yes.** Untuk *interface* Koneksi Subnet 3 akan digunakan *subnetprefix*2001:DB8:0:30::/64, maka konfigurasinya adalah **netsh interface ipv6 add route 2001:DB8:0:30::/64 "Koneksi Subnet 3" publish=yes.**

Konfigurasi ini akan memberikan alamat *global IPv6* dengan *prefix* 2001:DB8:0:20::/64 pada setiap *node* yang terhubung dengan *interface* Koneksi Subnet 2 dan *prefix*2001:DB8:0:30::/64 pada setiap *node* yang terhubung dengan *interface* Koneksi Subnet 3. Konfigurasi ini juga memberikan alamat global *IPv6* pada setiap *interface* yang ada di *router* 2. Berikut gambar 3.37 tampilan konfigurasi pemberian alamat global *IPv6* pada *router* 2.

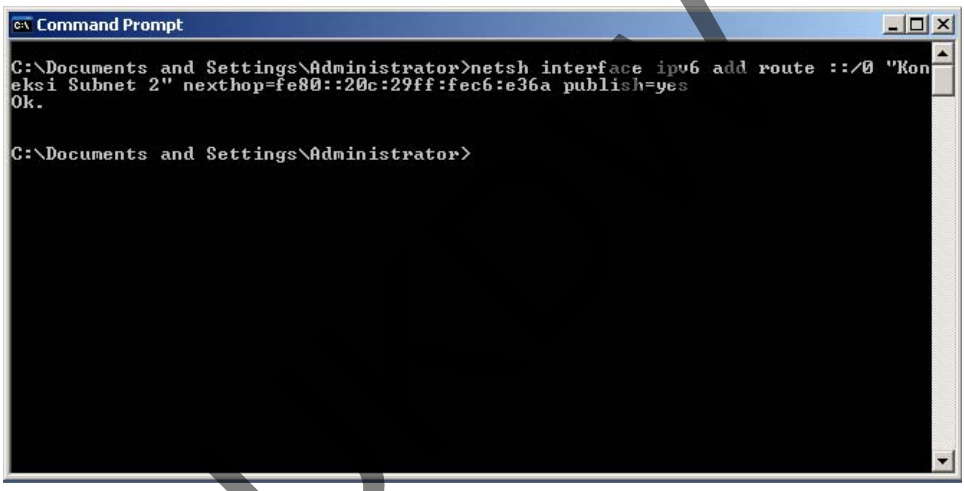


Gambar 3.37 Pemberian alamat global pada interface Koneksi Subnet 3

e) Mengkonfigurasi *routing static*pada *router* agar*IPv6*pada Router 2 dan *Router 1* dapat terhubung.

Cara mengkonfigurasinya adalah dengan mengetikkan **netsh interface ipv6 add route ::/0** "*interface yang menghubungkan router 1 dan router 2*" **nexthop=***alamat link-local pada interface di router 1yang berhubungan dengan router 2* **publish=yes.**

Pada topologi ini, alamat *link-local* pada *interface* di *router* 1 yang berhubungan dengan *router* 2 adalahFE80::20C:29FF:FEC6:E36A**,** maka konfigurasinya adalah **netsh interface ipv6 add route ::/0 "Koneksi Subnet 2" nexthop=FE80::20C:29FF:FEC6:E36A publish=yes.**Berikut gambar 3.38 tampilan konfigurasi penambahan *routing static* pada *router* 2.



Gambar 3.38 Pemberian routing static pada Router 2

3.12. Proses Pengukuran Parameter Jaringan

Kinerja dari teknik transisi ISATAP dan *dual stack* akan diukur dengan melihat tiga parameter. Parameter tersebut adalah *throughput, jitter* dan *round trip time* (RTT). Pengukuran ini akan menggunakan tiga *tools* jaringan, yaitu *iperf* untuk mengukur *throughput* dan *jitter,* dan *ping* dan *wireshark* untuk mengukur *round trip time.*

3.12.1. Iperf

Iperf adalah salah satu *tool* untuk mengukur *troughput bandwidth* dalam sebuah *link network.* Agar bisa dilakukan pengukuran, *Iperf* harus dipasang pada *point to point,* baik disisi *server* maupun *client. Server* di sini bukan *server* pada istilah jaringan pada umumnya. *Server* di sini dimaksudkan pada *node* dimana akan dilakukan pemantauan. Peran sebagai *server* ditandai dengan IP yang hendak dijadikan sebagai tujuan. *Iperf* sendiri bisa digunakan untuk mengukur *performancelink* dari sisi TCP maupun UDP. Berikut tabel 3.8 menunjukkan parameter yang biasa digunakan dalam pengujian *iperf.*

Tabel 3.8 Parameter yang digunakan dalam iperf

Parameter	Fungsi
-b	Data format
-r	Bi-directional bandwidth
-d	Simultaneous bi-directional bandwidth
-w	TCP Window size
-p	Port
-t	Timing
-i	Interval
-u	UDP tests
-b	Pengaturan bandwidth
-m	Maximum Segment Size display
-M	Pengaturan maximum Segment Size
-P	Parallel tests
-f	Format display
-h	Help

Berikut contoh perintah yang dimasukkan ketika melakukan pengujian.

a. Pada sisi *client*

iperf –c **2001:db8:0:10:20c:29ff:fe89:c211** *–V–u –p 5123 –P 1 –b 12.5M –i 1 –t 10 –l 128 –T 1*

Penjelasan dari penggunaan perintah di atas adalah sebagai berikut.

- -c, menunjukkan kalau node ini bertindak sebagai *client*
- 2001:db8:0:10:20c:29ff:fe89:c211 adalah alamat yang dituju dalam pengiriman paket pada tes ini.
- -V, menunjukkan bahwa tes ini akan ditujukan pada sebuah IPv6
- -u, menunjukkan bahwa tes ini adalah *UDP test*.
- -p 5123, menunjukkan *port* yang digunakan pada tes kali ini adalah *port* 5123
- -P 1, menunjukkan bahwa tes ini menggunakan *pararell test* dengan nilai 1
- -b 12.5M, menunjukkan batasan *bandwith* yang dipakai adalah 12.5 MBps
- -1 1, menunjukkan interval dari data yang ditangkap adalah setiap 1 detik
- -t 10, menunjukkan tes akan dilakukan selama 10 detik
- -l 128, menunjukkan besar paket yang akan dikirim adalah 128 Byte
- -T 1, menunjukkan *time to live* pada tes ini adalah 1 detik

b. Pada sisi *server*

iperf –s –u –p 5123 –P 0 –i 1 –l 128 –V

Penjelasan dari penggunaan perintah di atas adalah sebagai berikut.

- -s, menunjukkan kalau node ini bertindak sebagai *server*
- -u, menunjukkan bahwa tes ini adalah *UDP test*.
- -p 5123, menunjukkan *port* yang digunakan pada tes kali ini adalah *port* 5123
- -P 1, menunjukkan bahwa tes ini menggunakan *pararell test* dengan nilai 0
- -1 1, menunjukkan interval dari data yang ditangkap adalah setiap 1 detik
- -l 128, menunjukkan besar paket yang akan dikirim adalah 128 Byte
- -V, menunjukkan bahwa tes ini akan ditujukan pada sebuah IPv6

3.12.2. Ping

Ping (*Packet Internet Gopher*) adalah sebuah program utilitas yang dapat digunakan untuk memeriksa konektivitas jaringan berbasis teknologi *Transmission Control Protocol/Internet Protocol* (*TCP/IP*). Dengan menggunakan utilitas ini, dapat diuji apakah sebuah komputer terhubung dengan komputer lainnya. Hal ini dilakukan dengan mengirim sebuah paket kepada alamat IP yang hendak diujicoba konektivitasnya dan menunggu respon darinya.

Utilitas *ping* akan menunjukkan hasil yang positif jika dua buah komputer saling terhubung di dalam sebuah jaringan. Hasil berupa statistik keadaan koneksi kemudian ditampilkan di bagian akhir. Kualitas koneksi dapat dilihat dari besarnya waktu pergi-pulang (*round trip*) dan besarnya jumlah paket yang hilang (*packet loss*). Semakin kecil kedua angka tersebut, semakin bagus kualitas koneksinya. Berikut tabel 3.9 menunjukkan parameter yang biasa digunakan saat melakukan ping tes.

Tabel 3.9 Parameter yang digunakan dalam tes ping

Parameter	Fungsi
-n	Menentukan jumlah permintaan *echo* untuk mengirim. Standarnya adalah 4permintaan.
-w	Untuk mengatur *time-out*(dalam milidetik). Standarnya adalah 1.000(1-second time-out).
-l	Untuk mengatur ukuran paket *ping*. Ukuran standarnya adalah 32 byte.
-f	Untuk mengatur tidak melakukan fragmen bit pada paket *ping*.Secara *default*, paket *ping* mengizinkan fragmentasi.

Berikut contoh perintah yang digunakan untuk melakukan tes *ping.*

ping **2001:db8:0:10:20c:29ff:fe89:c211** *–l 128*

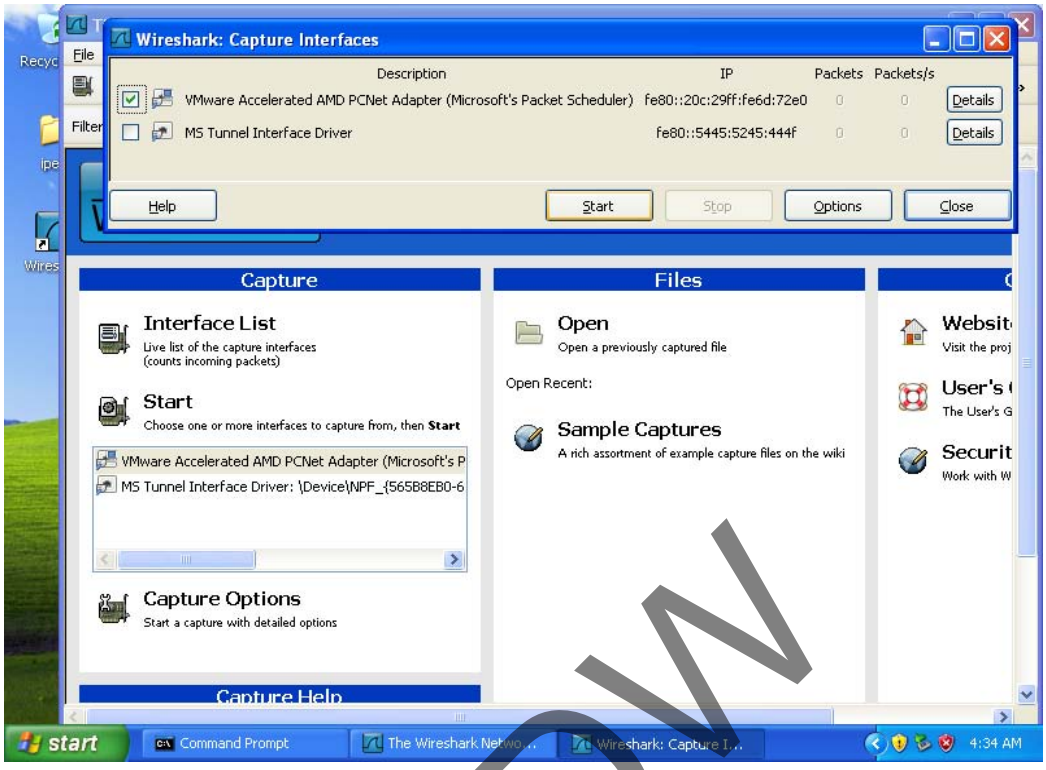
Penjelasan dari penggunaan perintah di atas adalah sebagai berikut.

- 2001:db8:0:10:20c:29ff:fe89:c211 adalah *IP* yang menjadi alamat tujuan dari tes ping
- -l 128, untuk mengatur besarnya paket yang dikirim sebesar 128 Byte

3.12.3. Wireshark

Wireshark merupakan salah satu dari sekian banyak *tool NetworkAnalyzer* yang banyak digunakan oleh *Network administrator* untuk menganalisa kinerja jaringannya terrmasuk protokol di dalamnya. *Wireshark* menggunakan tampilan grafis dalam menyajikan hasil dari analisa. *Wireshark* mampu menangkap paket-paket data atau informasi yang lalu lintas dalam jaringan. *Wireshark* dapat membaca data secara langsung dari *Ethernet, Token-Ring, FDDI, serial (PPP dan SLIP), 802.11 Wireless LAN*, dan koneksi *ATM*. *Tool wireshark* dapat menganalisa transmisi paket data dalam jaringan, proses koneksi dan transmisi data antar komputer.

Penggunaan *tool* ini cukup mudah. Berikut gambar 3.39 contoh penggunaan *tool* ini.



Gambar 3.39 Contoh penggunaan wireshark

Wireshark akan menampilkan semua *interface* yang terdapat pada sebuah PC. *Interface* tersebut bebas untuk dipilih salah satu atau keseluruhan untuk dimonitoring. Dalam contoh gambar 3.38 di atas, *interface* yang dipilih adalah *interface VMware Accelarated AMD PCnet Adapter* (*Microsoft's Packet Scheduler*).

BAB IV
ANALISA DATA

4.1. Uraian Umum

Pada bab ini akan dilakukan pengamatan cara kerja teknik transisi *IPv4*ke *IPv6,* yaitu teknik ISATAP (*Intra-Site Automatic Tunnel Addressing Protocol*) dan *Dual Stack*. Pengamatan tersebut meliputi hasil dari konfigurasi, mekanisme yang terjadi pada setiap protokol dan interkoneksi setiap *node* dengan menggunakan kedua teknik transisi tersebut. Penulis juga mengambil data-data dari hasil pengukuran performansi jaringan. pengukuran ini menggunakan beberapa parameter jaringan. Parameter tersebut adalah RTT (*Round Trip Time*), *Jitter* dan *Throughput*. pengukuran dilakukan dengan menggunakan beberapa *tool* jaringan.

Untuk mengukur RTT, penulis menggukan *tools ping*, untuk mengukur *jitter* dan *throughput* menggunakan *iperf*, dan untuk mengetahui mekanisme yang terjadi pada setiap protokol menggunakan *wireshark*. Mekanisme pengujian dilakukan dengan mengirimkan ICMP *payload* dengan variasi beban ICMP *payload*.Data-data yang dihasilkan akan ditampilkan dalam bentuk tabel dan grafik. Data yang dihasilkan dari pengujian ini hanya digunakan untuk mengukur kinerja, tapi tidak bisa sebagai ukuran baik tidaknya sebuah protokol.

4.2. Pengamatan Sistem Pengalamatan

Pada setiap teknik transisi *IPv4*ke *IPv6* memiliki cara pengalamatan yang berbeda. Pengalamatan yang dihasilkan pun akan menjadi identitas sebuah teknik transisi. Pada bagian ini akan dijelaskan bagaimana pengalamatan yang dilakukan pada setiap teknik transisi sehingga menghasilkan alamat tertentu. Pada bagian ini

juga akan dijelaskan bagaimana proses yang terjadi pada setiap protokol tersebut ketika antara *node* melakukan interkoneksi.

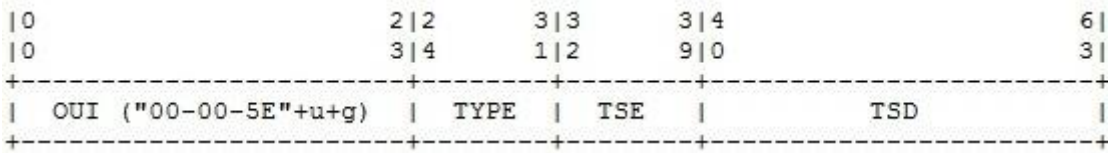
4.2.1. Pengalamatan Teknik Transisi ISATAP

Alamat ISATAP digunakan untuk merepresentasikan sebuah *host* yang menerapkan mekanisme teknik transisi *automatic tunneling*ISATAP. Alamat ISATAP merupakan alamat *unicast globalIPv6* yang menyisipkan alamat *IPv4.* Alamat*IPv6* bekerja dengan mengidentifikasi *interface-interface,* bukan *node.* Maka alamat *unicast* digunakan untuk mengidentifikasi sebuah *interface* tunggal.

Setiap *host* dan *router* agar dapat teridentifikasi menggunakan pengalamatan ISATAP, maka setiap *node* ISATAP menggunakan ISATAP*Interface Identifiers* (*Interface*ID). *Interface Identifier* digunakan untuk proses identifikasi sebuah *interface* pada sebuah *link*. Untuk alamat ISATAP sendiri menggunakan *interface Identifier* ::0:5efe:w.x.y.z. dimana w.x.y.z adalah alamat *IPv4* yang terdapat pada sebuah *interface* yang akan ditambahkan *interface Identifier*ISATAP. Sebuah *interface interface ID* untuk sebuah *ethernet interface* berdasarkan pada format modifikasi *EUI-64*. Panjang sebuah *interface ID* adalah 64-bit. ISATAP*interface IDs* ditentukan dari tiga hal berikut:

a) 24-bit IANA oui 00-00-5e.
b) 8-bit hexadecimal dengan nilai 0cfe.
c) 32-bit alamat IPv4.

Susunan bit dari sebuah ISATAP*interface ID* ditunjukkan pada gambar 4.1 sebagai berikut:



Gambar 4.1Susunan bit ISATAP interface IDs

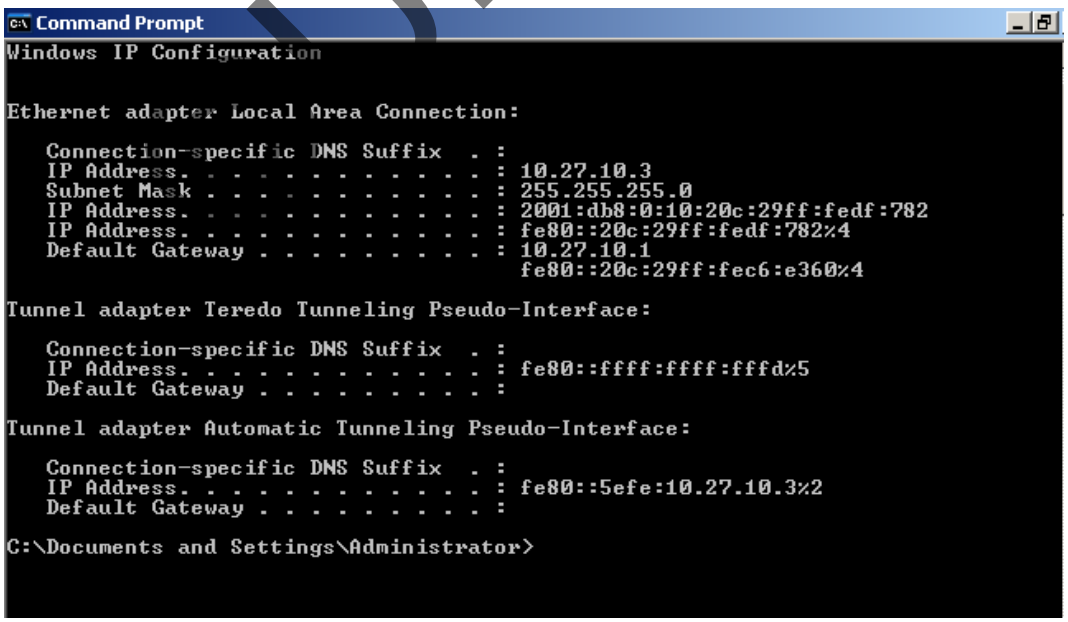
Adapun keterangan dari gambar 4.1 adalah sebagai berikut.

- OUI : IANA OUI: 00-00-5E dengan bit-bit 'u' dan 'g' (3 octet)
- TYPE : Jenis field; menentukan interpretasi dari (TSE, TSD) (1 octet)
- TSE : Type-Specific Extension (1 octet)
- TSD : Type-Specific Data (3 octets)

Pada gambar 4.1, hanya menunjukkan susunan dari *ISATAP interface ID*. Jika format *interface ID* digabungkan dengan prefix 64-bit, maka format alamat ISATAP-nya menjadi seperti gambar 4.2 berikut:

Gambar 4.2 Susunan alamat unicast global ISATAP

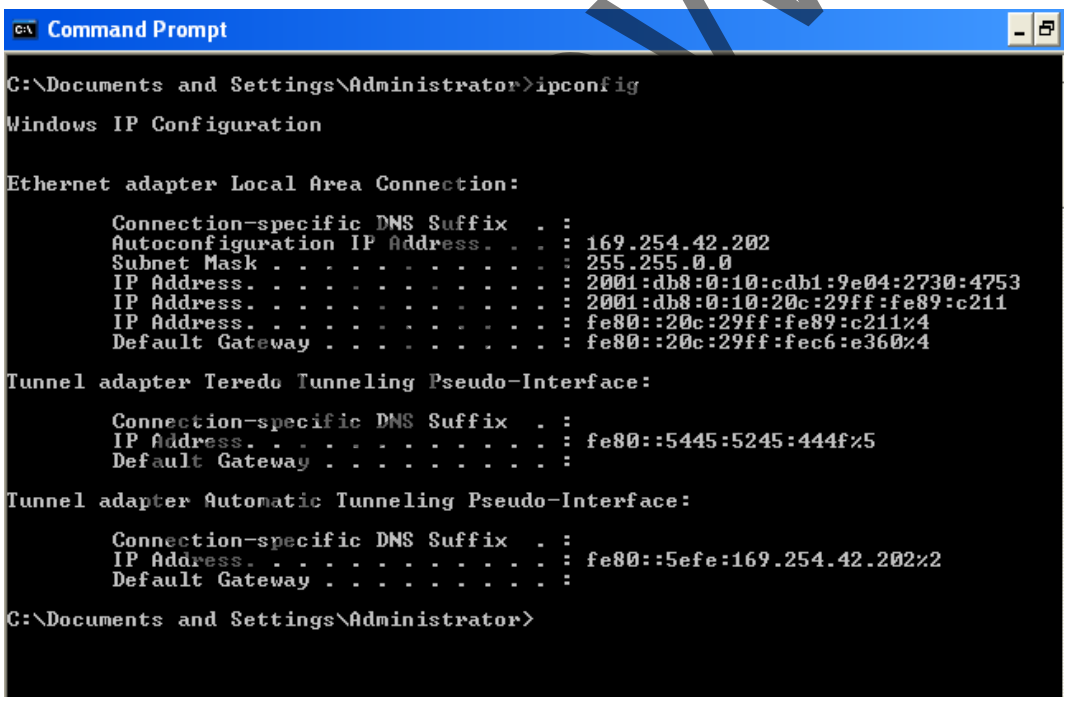
Pada susunan alamat *unicast global*ISATAP bisa dilihat bagaimana susunan dari *prefix* alamat dan *ISATAP interface ID*.Berikut gambar-gambar yang menunjukkan konfigurasi alamat IP pada setiap *node* pada teknik transisi ISATAP. Gambar 4.3 menunjukkan konfigurasi *alamat* ISATAP pada DNS 1.



Gambar 4.3 Tampilankonfigurasi alamat ISATAP pada DNS 1

Pada DNS 1 bisa kita dilihat tidak adanya alamat ISATAPyang dipasang pada *Tunnel adapter Automatic Tunneling Pseudo-Interface.* Pada *interfce* itu hanya terpasang alamat *IPv6link-local.* Alamat *link-local*tersebut tidak dapat berhubungan dengan subnet luar. DNS 1 berada pada *site* yang ditujukan menjadi pengguna alamat *IPv6*. DNS menerapkan dua IP secara bersamaan.

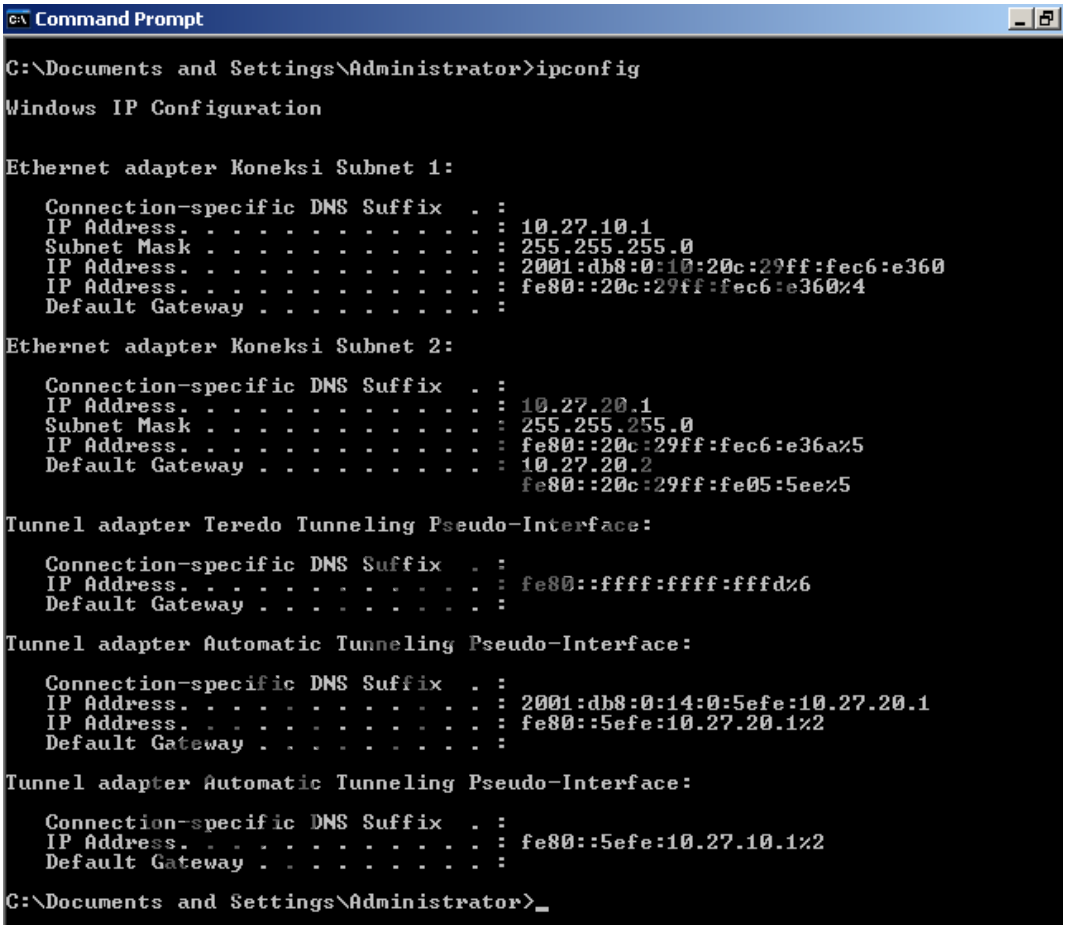
Dua IP tersebut adalah*IPv4* dan*IPv6*. *IPv4* tetap digunakan karena *client* 2 berada di *site* yang ditujukan menggunakan *IPv4*. Alamat IPv4 pada DNS ini akan digunakan pada konfigurasi alamat DNS pada setiap *client.*Konfigurasi alamat IP pada DNS hampir sama dengan konfigurasi pada *client* 1, hanya saja pada *client* 1 tidak menggunakan alamat *IPv4* lagi. Berikut gambar 4.4 merupakan tampilan konfigurasi teknik ISATAP pada *client* 1.



Gambar 4.4 Tampilan konfigurasi alamat ISATAP pada client 1

Pada gambar 4.4 bisa dilihat alamat *IPv4*yang terpasang adalah 169.254.42.202. alamat itu bukanlah sebuah alamat *IPv4* yang diapkai. Alamat tersebut didapat dari fungsi *DHCP* dari *WINDOWS XP*. *SubnetIPv4* yang dipakai untuk koneksi *site* 1 adalah 10.27.10.0/24.

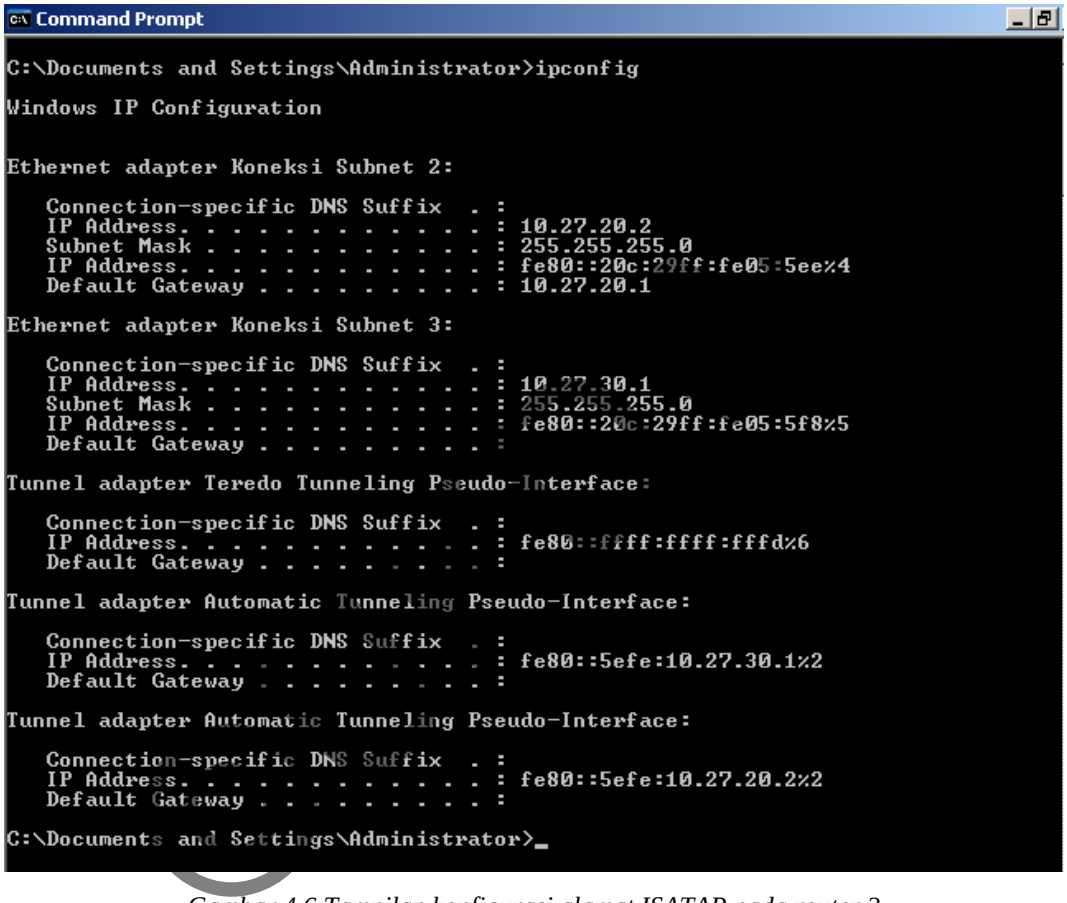
Pada *node* ini alamat*IPv6*yang dipakai adalah 2001:db8:0:10:20c:29ff:fe89:c211. Alamat *IPv6* inilah yang akan berhubungan dengan *tunnel* pada ISATAP.Pada ISATAP, *tunnel* dikonfigurasi pada sebuah *router*. Untuk topologi yang dipakai pada penelitian ini, *router* 1 menjadi *router*ISATAP. Berikut gambar 4.5 merupakan tampilan konfigurasi dari *router* 1.



Gambar 4.5 Tampilan konfigurasi alamat ISATAP pada router 1

Pada gambar 4.5 adalah tampilan konfigurasi alamat ISATAP pada*router* 1. Pada *interface Automatic Tunneling Pseudo-Interface* alamat yang dihasilkan adalah 2001:db8:0:10:0:5efe:10.27.20.1. Dari alamat tersebut dapat dilihat per bagian. Bagian 2001:db8:0:10 merupakan *prefix* yang diberikan untuk menjadi awalan dari alamat ISATAP dan 0:0:5efe merupakan tambahan untuk menjadikan alamat ISATAP menjadi sebuah alamat *unicast global.* Lalu pada akhiran dari

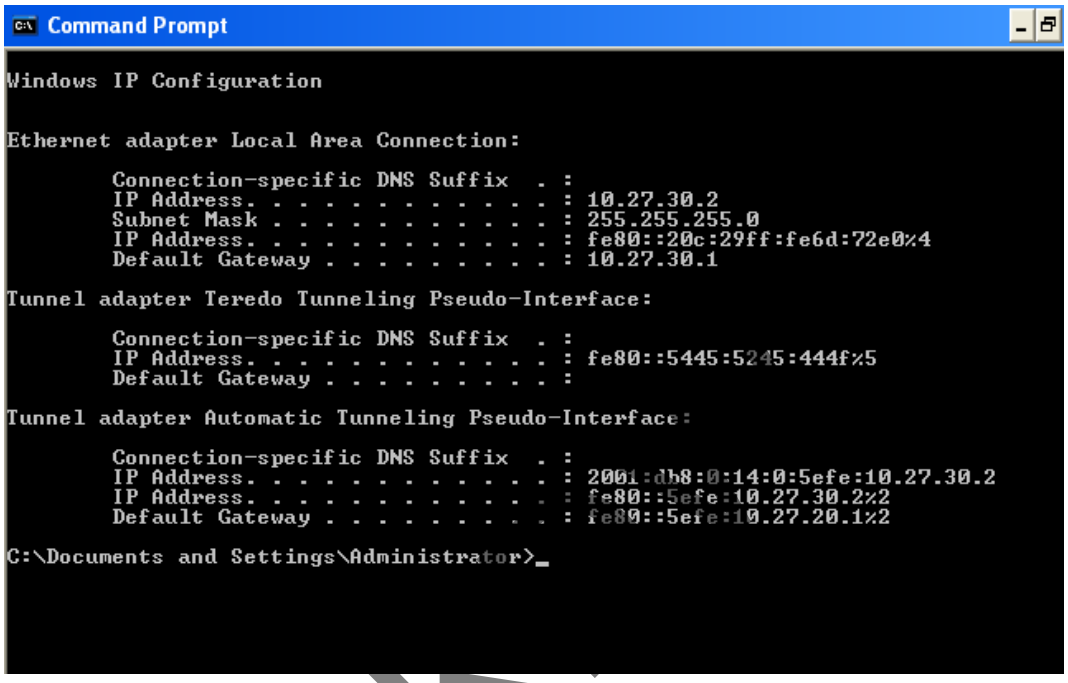
alamat ISATAP ini ditambahkan alamat *IPv4* yang menjadi alamat dari sebuah *interface*, pada contoh ini alamat *IPv4*-nya adalah 10.27.20.1.*Router* 1 akan berhubungan dengan *router* 2. *Router* 2 adalah *router* yang berada pada *site* yang ditujukan untuk penggunaan *IPv4.*Berikut gambar 4.6 tampilan konfigurasi dari *router* 2.



Gambar 4.6 Tampilan konfigurasi alamat ISATAP pada router 2

Pada gambar 4.6 bisa kita lihat*IPv6* yang ada pada konfigurasi adalah alamat *IPv6 link-local,* termasuk pada *Tunnel adapter Automatic Tunneling Pseudo-Interface.* Alamat*IPv6link-local* hanya bisa digunakan untuk berkomunikasi dalam satu subnet.Hal ini menandakan *node* ini menerapkan *IPv4.* Alamat 10.27.20.2 pada *interface* Koneksi Subnet 2 digunakan untuk berhubungan dengan *router* 1 dan Alamat 10.27.30.1 pada *interface* Koneksi Subnet 3 digunakan untuk menjadi *gateway* pada setiap *node* di

*subnet*10.27.30.0/24.*Router* 2 berhubungan dengan *client* 2. *Client* 2 sama dengan *router* 2. *Client* 2 hanya menerapkan*IPv4*dalam berkomonikasi dalam jaringan. Berikut gambar 4.7 tampilan konfigurasi dari *client* 2.

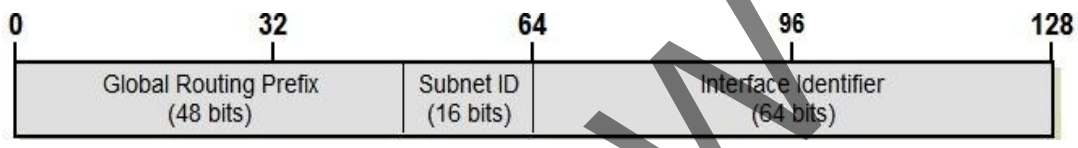


Gambar 4.7 Tampilan konfigurasi alamat ISATAP pada client 2

Pada gambar 4.7 adalah tampilan konfigurasi alamat ISATAP pada *client* 2. Pada *interface Automatic Tunelling Pseudo-Interface* alamat yang dihasilkan adalah 2001:db8:0:10:0:5efe:10.27.30.2. Jika alamat tersebut dipisahkan menurut bagiannya maka 2001:db8:0:10 merupakan *prefix* yang diberikan untuk menjadi awalan dari alamat ISATAP dan 0:0:5efe merupakan tambahan*ISATAP interface id* untuk menjadikan alamat ISATAP menjadi sebuah alamat *unicast global.* Lalu pada akhiran dari alamat ISATAP ini ditambahkan alamat *IPv4*yang menjadi alamat dari sebuah *interface,* pada contoh ini alamat*IPv4*-nya adalah 10.27.30.2. alamat ISATAP ini yang akan digunakan jika *client* satu akan melakukan konektivitas ke *client* 2.

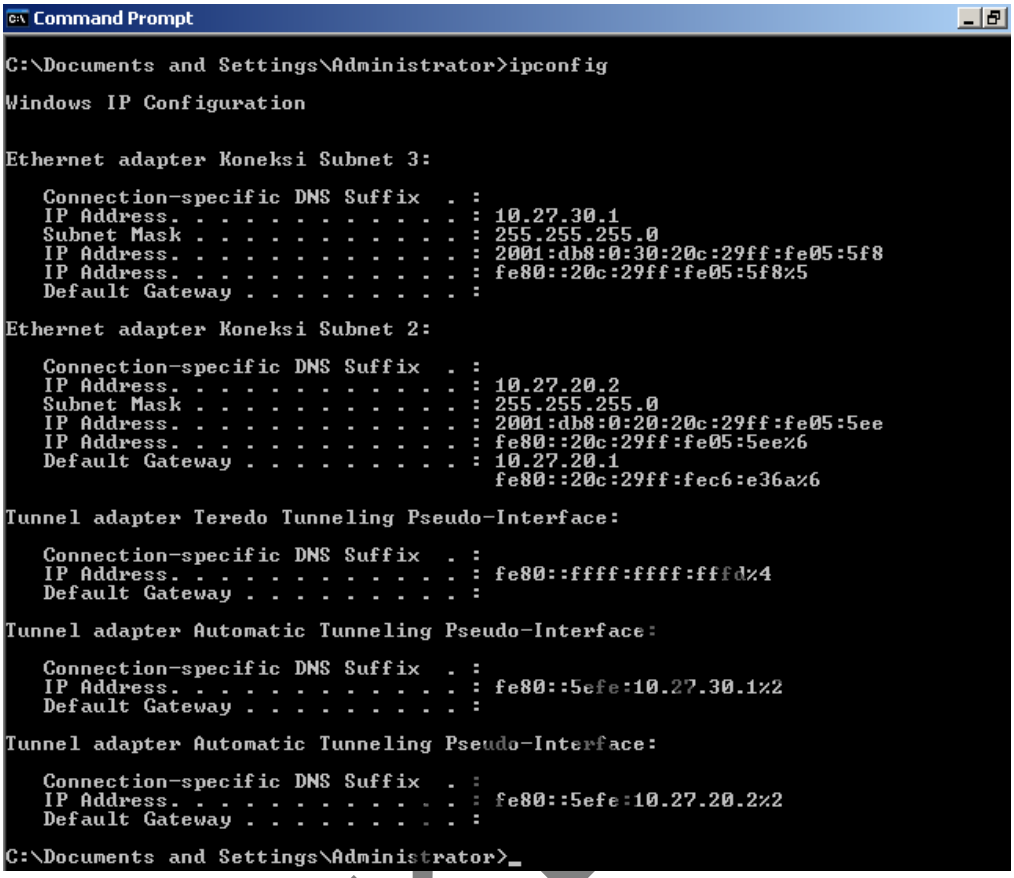
4.2.2. Pengalamatan Teknik Dual Stack

Pada percobaan teknik transisi *dual stack*, penulis menggunakan alamat *unicast IPv6*.Alamat *unicast global IPv6* miripdengan alamat publik pada*IPv4*.Sama seperti alamat publik *IPv4* yang dapat secara global dirujuk oleh *host-host* di *internet* dengan menggunakan proses *routing*, alamat ini juga mengimplementasikan demikian. Struktur alamat *IPv6unicast global* terbagi menjadi topologi tiga level (*Public*, *Site*, dan *Node*). Berikut gambar 4.8susunan alamat *IPv6 unicast global* pada *router* 2.



Gambar 4.8 Struktur alamat IPv6unicast

Berikut gambar 4.9 tampilan konfigurasi alamat *unicast IPv6*yang digunakan pada teknik transisi *dual stack.*



Gambar 4.9 Konfigurasi alamatunicast IPv6 pada router 2

Pada interface Koneksi Subnet 2 pada *router* dua memiliki alamat *IPv6 unicast* 2001:db8:0:20:20c:29ff:fe05:5ee. Dengan merujuk pada struktur alamat *IPv6 unicast*, alamat 2001:db8:0:20:20c:29ff:fe05:5ee dapat dipisahkan menurut strukturnya. 2001:db8:0 adalah *Global Routing Prefix*, 0002 atau disederhanakan menjadi 2 adalah *Subnet ID* dan 20c:29ff:fe05:5ee adalah *Interface Identifier*.Alamat *unicast IPv6* dikonfigurasi dengan cara memberikan *subnet prefix*. Pada*subnet* 10.27.20.0/24 diberikan *subnet prefix*2001:db8:0:20::/64.

4.3. Pengujian Interkoneksi dan Pengamatan Proses yang Terjadi pada Setiap Protokol Teknik Transisi

Sebuah sistem jaringan komputer akan memberikan interkoneksi pada setiap *node* yang diinginkan untuk tersambung. Konektivitas dari setiap *node*adalah tanda dari sebuah jaringan telah tercipta dan memastikan antar *node* ini telah terhubung. Untuk memastikan sebuak konektivitas jaringan bisa menggunakan beberapa cara, salah satunya menggunakan utilitas *ping. Ping* adalah sebuah program utilitas yang dapat digunakan untuk memeriksa konektivitas jaringan berbasis teknologi *Transmission Control Protocol/Internet Protocol* (*TCP*/*IP*). Dengan menggunakan utiliti ini, anda dapat menguji apakah sebuah komputer terhubung dengan komputer lainnya atau *internet*. Hal ini dilakukan dengan mengirim sebuah paket kepada alamat *IP* yang hendak diujicoba konektivitasnya dan menunggu respon darinya.

Selain pengujian konektivitas, pengujian *traceroute* juga akan dilakukan pada sistem jaringan ini. Pengujian *traceroute* dilakukan untuk bisa melihat *hop* apa saja yang dilalui ketika melakukan konektivitas antar *node.* Pengujian ini akan dilakukan dengan menggunakan utilitas *tracert* pada *Windows. Tracert.exe* adalah sebuah *rute-tracing* utilitas yang dapat digunakan untuk menentukan jalur jaringan jalan ke tujuan. Perintah *tracert* menunjukkan serangkaian *router IP* yang digunakan untuk mengirim paket dari *node* sumber ke *node* tujuan dan menunjukkan berapa lama waktu yang dibutuhkan untuk setiap *hop*.

Dalam teknik transisi juga akan dilihat proses yang terjadi pada protokol teknik protokol tersebut ketika melakukan konektivitas. Untuk melihat apa yang terjadi pada setiap protokol teknik transisi, penulis menggunakan *software wireshark* untuk menangkap paket-paket yang dilalui dari sebuah *interface. Wireshark* adalah aplikasi yang bisa digunakan pada *Windows* maupun *Linux. Wireshark* mampu menangkap paket-paket data atau informasi yang melewati infrastruktur jaringan. Semua jenis paket informasi dalam berbagai format protokol akan ditangkap dan dianalisa.

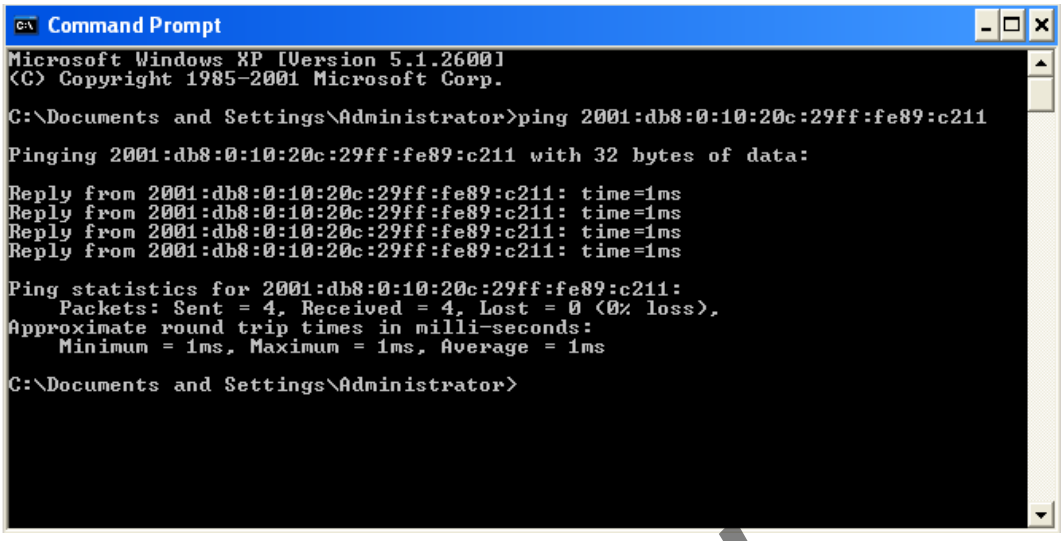
Manfaat*Wireshark* yang bisa didapatkan dari penggunaan *wireshark* antara lain: dapat melakukan *troubleshooting* jaringan dengan menganalisa paket yang di-*capture*, dan mengamankan jaringan dengan melakukan filter paket berdasarkan protokol yang mudah dimasuki oleh pihak *unauthorized.*

4.3.1. Pengujian pada Implementasi Teknik Transisi ISATAP

Ada tiga pengujian yang dilakukan pada teknik transisi ISATAP. Pengujian itu ada lah pengujian interkoneksi, pengujian jejak dan penangkapan paket-paket yang melalui*interface* yang digunakan.

4.3.1.1. Pengujian Interkoneksi pada Implementasi Teknik Transisi ISATAP

Pada implementasi teknik transisi ISATAP, pengujian konektivitas akan dilakukan dengan melakukan *ping*. *Ping* dilakukan dari *client* 2 dan ditujukan ke *client* 1. *Client* 2 dipilih sebagai sumber karena *client* 2 berbasiskan *IPv4* dan *client* 1 berbasiskan *IPv4*. Pada dasarnya *client* 2 juga memiliki *IPv6 link-local*, hanya saja pada *router* 1 dan *router* 2 dikonfigurasi untuk tidak meneruskan paket *IPv6* untuk Subnet 2 dan Subnet 3. Berikut gambar 4.10 hasil dari uji konektivitas teknik transisi ISATAP.

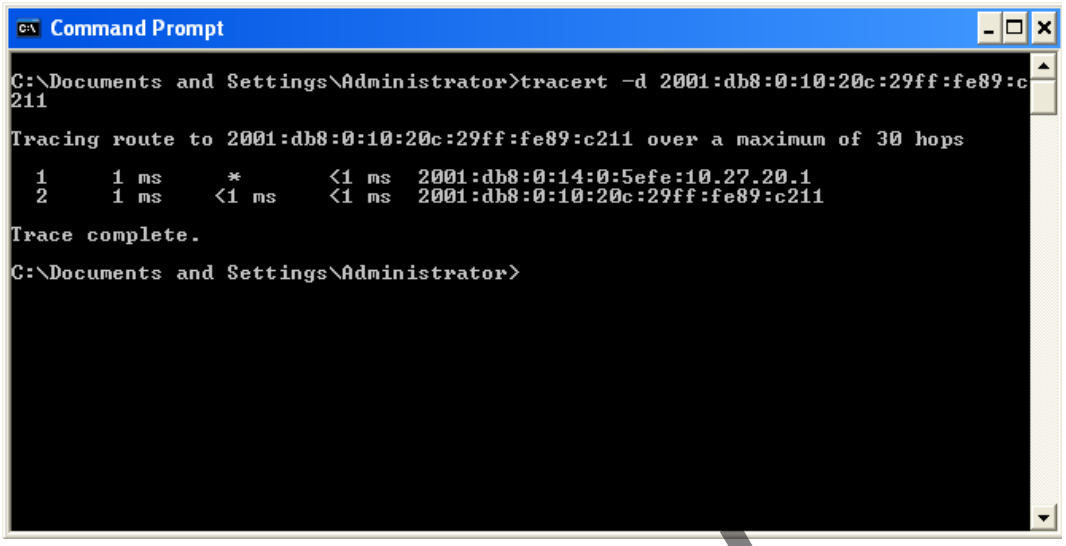


Gambar 4.10Hasil ping test dari client 2 ke client 1 pada teknik transisi ISATAP

Pada gambar 4.10 di atas bisa dilihat semua *request* yang dilakukan melalui utiliti *ping* mendapatkan *reply* dari *client* 1 yang beralamat *IPv6* 2001:db8:0:10:20c:29ff:fe89:c211. Hal ini menunjukkan bahwa konektivitas antar *node* tertentu pada teknik transisi ISATAP telah terjadi. Selain menunjukkan adanya konektivitas, hal ini juga menunjukkan telah terbentuknya *tunnel* yang menghubungkan antara*IPv4*dan *IPv6.*

4.3.1.2. Pengujian Traceroute pada Implementasi Teknik Transisi ISATAP

Pada implementasi teknik transisi ISATAP, pengujian *traceroute* akan dilakukan dengan melakukan *tracert. Tracert* dilakukan dari *client* 2 dan ditujukan ke *client* 1. *Client* 2 yang berbasiskan *IPv4*akan melakukan tes jejak terhadap *client* 1 yang berbasiskan *IPv6.* Berikut gambar 4.11 menunjukkan hasil tes *tracert* yang dilakukan *client* 2.



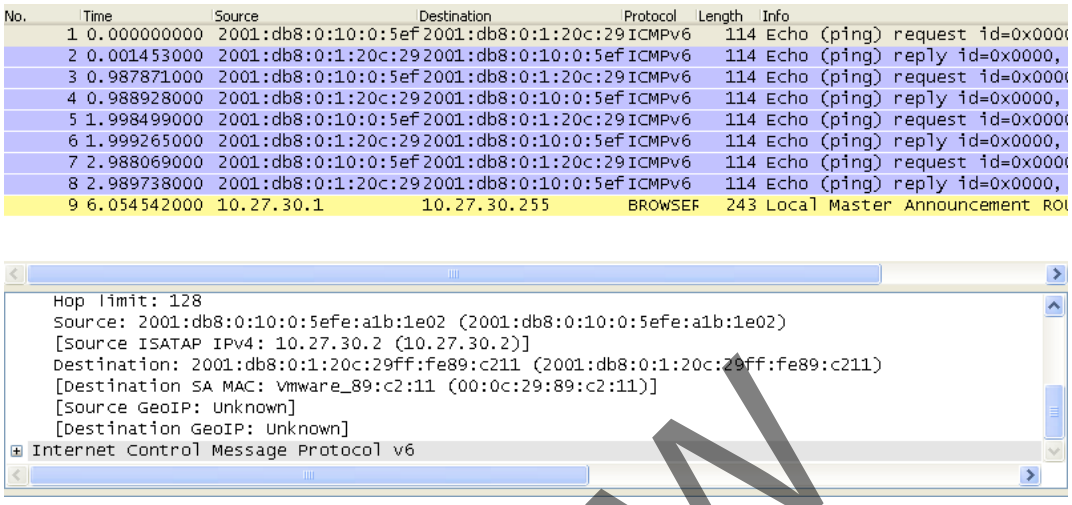
Gambar 4.11 Hasil test tracert dari client 2 ke client 1pada teknik transisi ISATAP

Pada gambar 4.11 di atas bisa dilihat proses tes jejak telah berhasil dilakukan. Pada hasil di atas, jejak yang didapatkan adalah 2 *hop*. Secara infrastruktur perangkat jaringan, sesungguhnya paket tersebut melalui 3 hop. *Hop* tersebut adalah *router* 2, *router* 1, lalu *client* 1. Hasil 2 hop yang didapat dari tes *tracert* dikarenakan paket yang dikirimkan dari client 2 dimasukkan ke dalam *tunnel ISATAP* yang telah dikonfigurasi lalu diteruskan ke *client* 1. Jadi secara protokol teknik transisi ISATAP, konektivitas yang terjadi adalah *client* 2 mengirim paket yang dimasukkan ke *tunnel* ISATAP, lalu *tunnel*ISATAP ini meneruskan paket ke *client* 1. *Tunnel* ISATAP yang terbentuk ini ditandai dengan alamat 2001:db8:0:10:0:5efe:10.27.20.1.

4.3.1.3. Penangkapan Paket pada Proses Konektivitas Implementasi Teknik Transisi ISATAP

Pada implementasi teknik transisi ISATAP, penangkapan paket akan menggunakan *software wireshark. Wireshark* akan dipasang pada *clien*t 2. Paket yang akan ditangkap adalah paket yang dikirmkan melalui test *ping. Wireshark* akan diset untuk meng-*capture* setiap paket yang melalui *interface* yang

berhubungan dengan *router* 2. Berikut gambar 4.12 menunjukkan hasil tangkapan yang berhasil dilakukan oleh *software wireshark.*



Gambar 4.12Hasil tangkapan yang dilakukan wireshark pada teknik transisi ISATAP

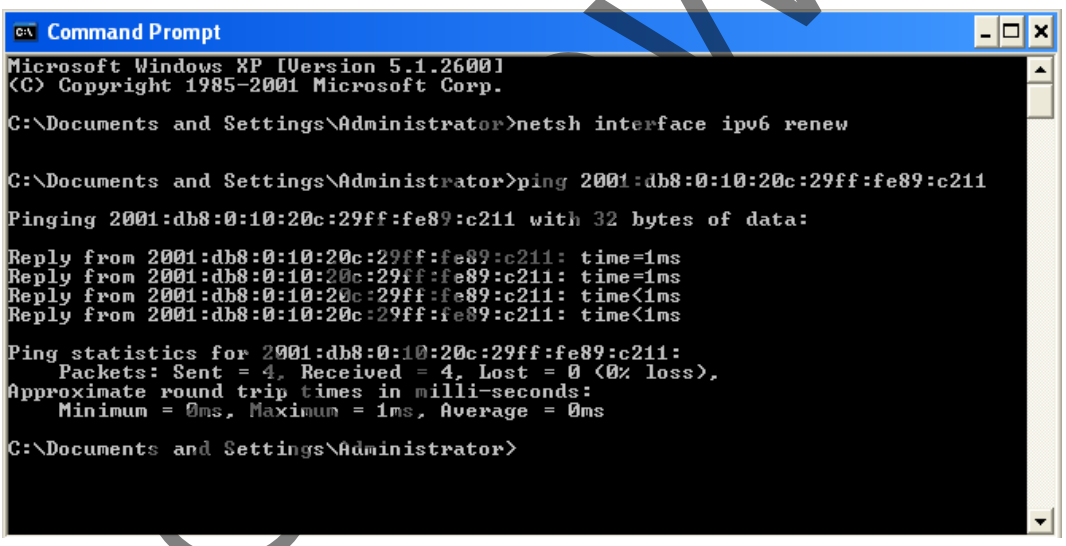
Pada gambar 4.12 di atas bisa dilihat proses yang terjadi pada teknik transisi ISATAP. *IPv4*yang digunakan pada *client* 2 dianggap sebagai ISATAP*IPv4. Client* 2 berhubungan dengan *client* 1 yang memiliki *IPv6*2001:db8:0:10:20c:29ff:fe89:c211 melalui alamat ISATAP2001:db8:0:14:0:5efe:a1b:1eo2. Alamat 2001:db8:0:14:0:5efe:a1b:1eo2 bisa teridentifikasi sebagai alamat ISATAP karena alamat ini menggunakan *prefix*2001:db8:0:10. *Prefix*2001:db8:0:10adalah *prefix* yang diberikan pada konfigurasi pada *router* 1 yang digunakan untuk memberikan alamat pada *interface*ISATAP.

4.3.2. Pengujian pada Implementasi Teknik Transisi Dual Stack

Pengujian teknik transisi *dual stack* sama dengan pengujian teknik transisi ISATAP. Ada tiga pengujian yang dilakukan pada teknik transisi *dual stack.* Pengujian itu adalah pengujian interkoneksi, pengujian jejak dan penangkapan paket-paket yang melalui *interface* yang digunakan.

4.3.2.1. Pengujian Interkoneksi Pada Implementasi Teknik Transisi Dual Stack

Pada implementasi teknik transisi *dual stack,* pengujian konektivitas akan dilakukan dengan melakukan *ping. Ping* dilakukan dari *client* 2 dan ditujukan ke *client* 1. Dalam implementasi teknik transisi *dual stack,* setiap *node* yang menjadi bagian dari topoogi *dualstack* akan mendapatkan dua *stack IP, IPv4*dan *IPv6*. Dalam pengujian ini penulis memfokuskan pada *stack IPv6.* Hal ini dikarenakan tujuan dari penerapan teknik ini adalah transisi dari *IPv4*ke *IPv6.* Berikut gambar 4.13 menunjukkan hasil dari uji konektivitas teknik transisi *dual stack.*

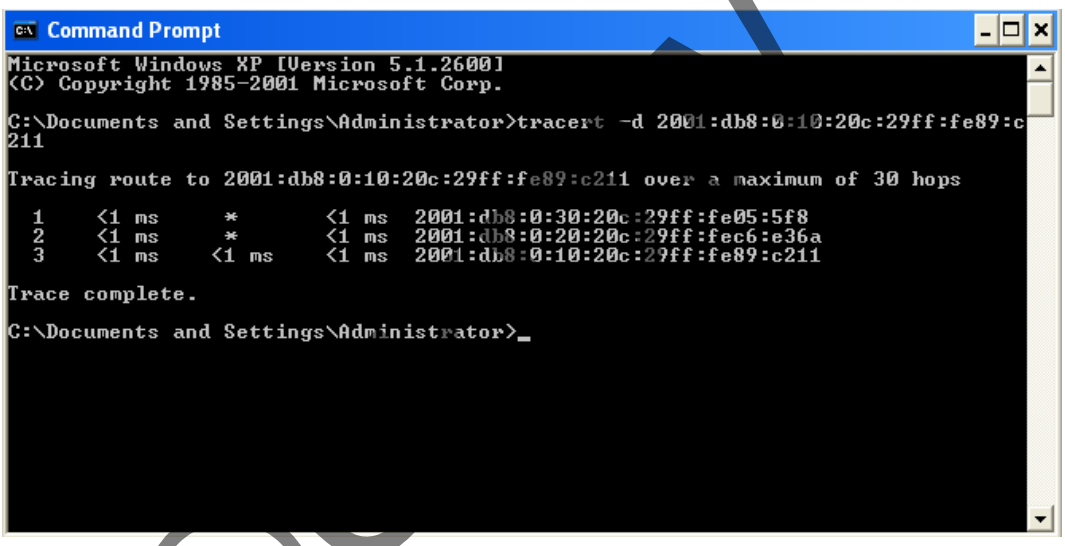


Gambar 4.13 Tampilan hasil ping test dari client 2 ke client 1 pada teknik transisi dual stack

Pada gambar 4.13 di atas bisa dilihat semua *request* yang dilakukan melalui utiliti *ping* mendapatkan *reply* dari *client* 1 yang beralamat*IPv6*2001:db8:0:10:20c:29ff:fe89:c211. Hal ini menunjukkan bahwa konektivitas antar *node* tertentu pada teknik transisi *dual stack* telah terjadi.Selain menunjukkan adanya konektivitas, hal ini juga menunjukkan telah terbentuknya *routing global IPv6* yang menghubungkan antara node dalam teknik transisi *dual stack* ini.

4.3.2.2. Pengujian Traceroute pada Implementasi Teknik Transisi Dual Stack

Pada implementasi teknik transisi *dual stack,* pengujian *traceroute* akan dilakukan dengan melakukan *tracert. Tracert* dilakukan dari *client* 2 dan ditujukan ke *client* 1. *Client* 2 berada pada Subnet 3 dan *client* 1 berada pada Subnet 1. Berikut gambar 4.14 menunjukkanhasil tes *tracert* yang dilakukan *client* 2.



Gambar 4.14 Hasil test tracert dari client 2 ke client 1pada teknik transisi dual stack.

Pada gambar 4.14 di atas bisa dilihat proses tes jejak telah berhasil dilakukan. Jejak yang didapatkan adalah 3 hop. 3 hop tersebut mewakili tiga *node* yang dilalui oleh paket yang dikirimkan dari *client* 2. 3 hop yang dilalui adalah:

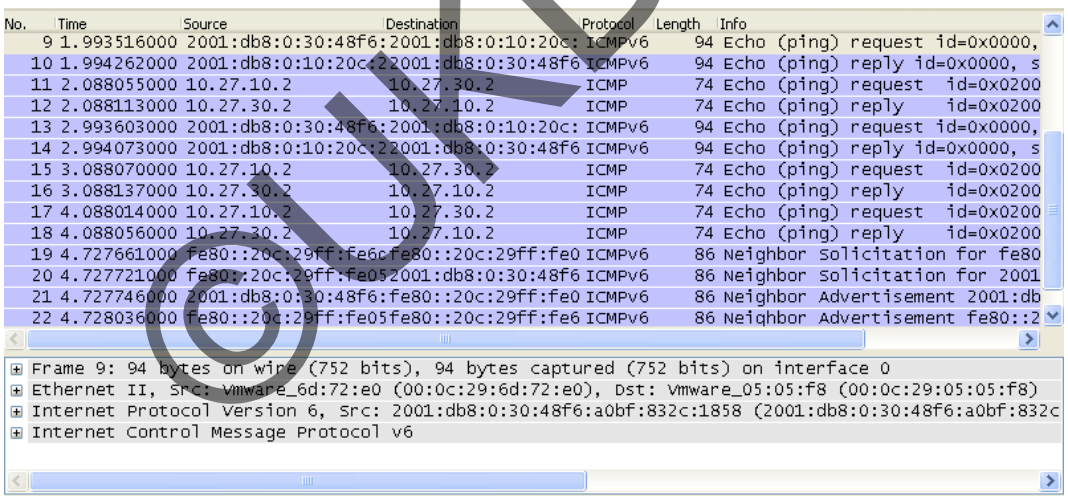
a. 2001:db8:0:30:20c:29ff:fe05:5f8,
b. 2001:db8:0:20:20c:29ff:fec6:e36adan,
c. 2001:db8:0:10:20c:29ff:fe89:c211.

Alamat 2001:db8:0:30:20c:29ff:fe05:5f8 adalah alamat*IPv6* pada *router* 2. Alamat 2001:db8:0:20:20c:29ff:fec6:e36a adalah alamat*globalIPv6* pada *router* 1. Alamat 2001:db8:0:20:20c:29ff:fec6:e36a adalah alamat *global IPv6* pada

client 1 yang mana menjadi tujuan dari tes jejak. Selain menunjukkan jejak yang dilalui, hal ini juga menunjukkan telah terbentuknya *routing globalIPv6* yang menghubungkan antara *node* dalam teknik transisi *dual stack* ini.

4.3.2.3. Penangkapan Paket pada Proses Konektivitas Implementasi Teknik Transisi Dual Stack

Pada implementasi teknik transisi *dual stack*, penangkapan paket akan menggunakan *software wireshark. Wireshark* akan dipasang pada *client* 2. Paket yang akan ditangkap adalah paket yang dikirmkan melalui test *ping*. *Wireshark* akan diset untuk meng-*capture* setiap paket yang melalui *interface* yang berhubungan dengan *router* 2. Berikut gambar 4.15 menunjukkan hasil tangkapan yang berhasil dilakukan oleh s*oftware wireshark.*



Gambar 4.15 Hasil tangkapan yang dilakukan wireshark pada teknik transisi dual stack

Pada gambar 4.15 di atas bisa dilihat proses yang terjadi pada teknik transisi *dual stack.*Pada teknik transisi *dual stack*, kedua *stack IPv4* dan*IPv6*dinyalakan pada setiap *interface.* Pada hasil tangkapan di atas bisa dilihat ada 2 jenis IP yang tertangkap.*IPv4* yang tertangkap adalah 10.27.10.3 dan 10.27.30.2laluIPv6yang tertangkap adalah 2001:db8:0:10:20c:29ff:fe89:c211 dan

2001:db8:0:30:810d:d332:239b:341c. Dua IP ini bekerja masing-masing dan tidak ada konektivitas antara kedua jenis IP ini.

4.4. Pengukuran Kinerja Teknik Transisi

Untuk mengetahui kinerja dari teknik transisi yang diterapkan, penulis melakukan pengukuran terhadap teknik transisi ISATAP dan *dual stack.* Penulis melakukan pengukuran terhadap tiga parameter jaringan. Pengukuran kinerja teknik transisi didasarkanpada *throughput, round trip time*(RTT) dan *jitter*. Analisa kinerja teknik transisi akan dilakukan dengan melihat angka yang dihasilkan dari tiap-tiap pengujian pada teknik transisi. Angka yang didapat hanya merepresentasikan dari kinerja sebuah teknik transisi. Angka-angka ini tidak mewakili sebuah efisiensi dari sebuah teknik transisi.

4.4.1. Pengukuran pada Teknik Transisi ISATAP

Pada teknik transisi ISATAP akan dilakukan pengukuran terhadap tiga parameter jaringan. Parameter itu adalah *throughput, jitter* dan *round trip time*(RTT). Pengujian akan dilakukan dengan mengirimkan paket dengan ukuran tertentu. Dalam pengiriman paket ini akan didapatkan data dari alat pengukur yang digunakan. Alat pengukur yang digunakan dalam pengujian ini ada 3, yaitu *iperf, ping*dan *wireshark.*

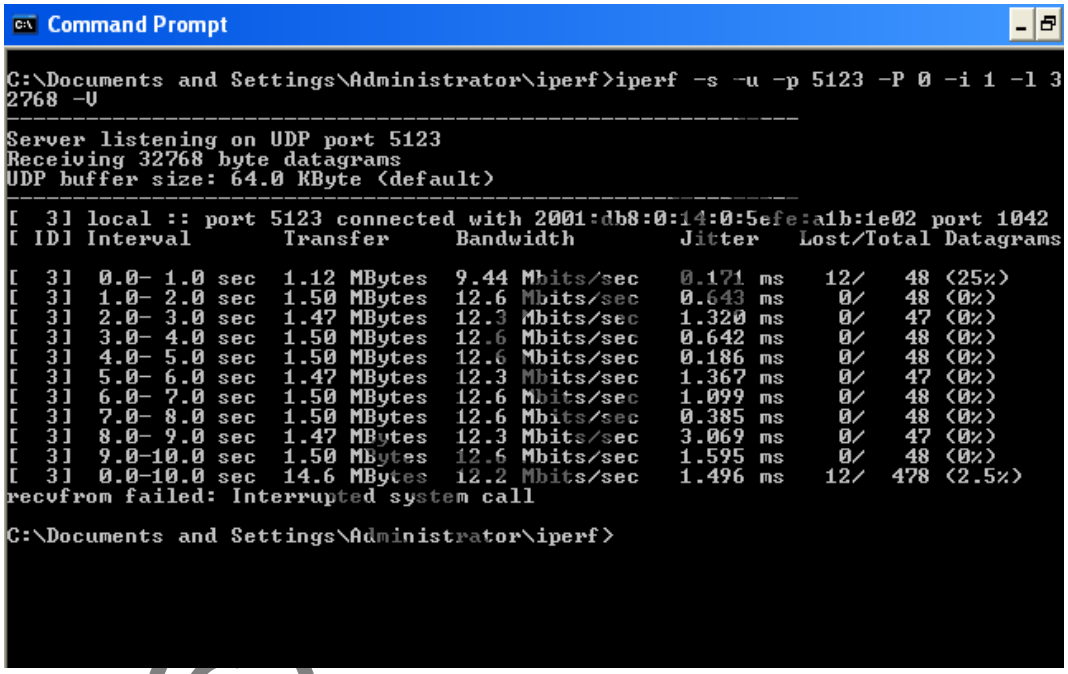
4.4.1.1. Pengukuran Throughput, Jitter dan Packet Loss pada Teknik Transisi ISATAP

Througput, jitter dan *packet loss*akan diukur menggunakan *iperf.* Perintah *iperf* akan dijalankan pada dua buah node yang dipilih menjadi sumber

dan tujuan dari tes ini. *Node* yang dipilih dalam tes ini adalah *client* 1 dan *client* 2. *Client* 1 bertindak sebagai *server* atau tujuan dan *client* 2 bertindak sebagai *client* atau sumber. Berikut beberapa gambar dari hasil tes *througput* pada teknik transisi ISATAP.

a. Hasil listening tes throughput, jitter dan packet loss pada client 1

Berikut gambar 4.16 menunjukkan tampilan konfigurasi dan hasil tes*iperf* yang dilakukan pada *client* 1.

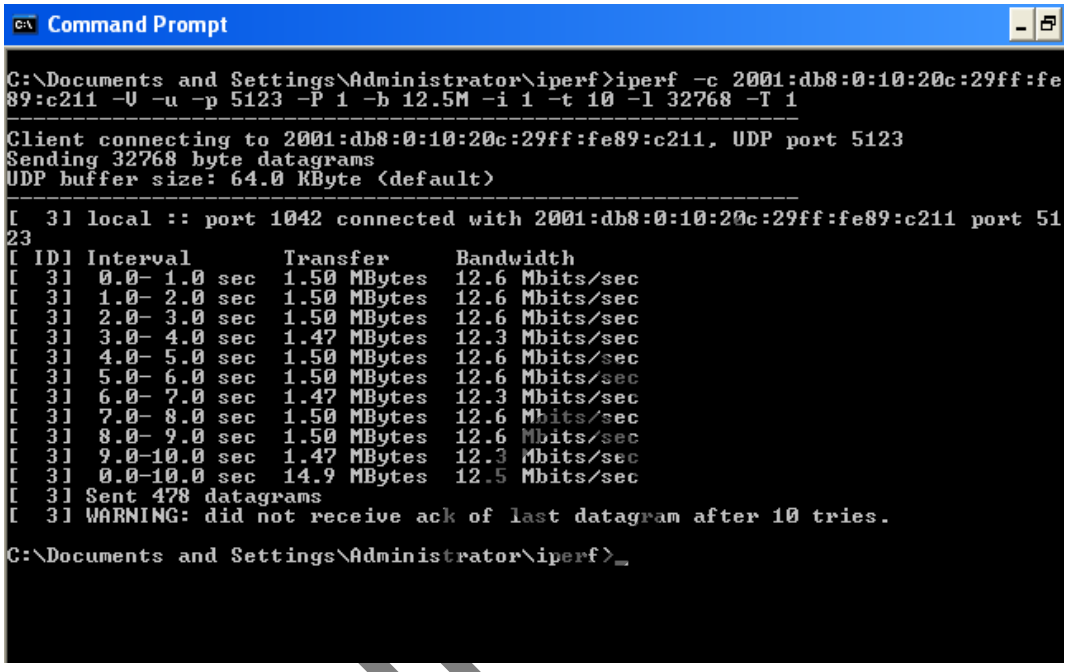


Gambar 4.16 Hasil throughput, jitter packet loss dari client 1 pada teknik ISATAP

Pada gambar 4.16 di atas, bisa dilihat perintah yang digunakan adalah iperf –s –u –p 5123 –P 0 –i 1 –l 32768 –V. Dari perintah yang dijalankan, bahwa PC ini bertindak sebagai *server* dan melakukan *listening* tes UDP melalui *port* 5123 dan paket yang dikirim sebesar 32768 Byte. Dari hasil tes tersebut dapat dilihat bahwa proses ini mengirimkan total 14,6 MBytes dan menghasilkan*throughput bandwith* 12,2 Mbps, *jitter* 1,496 ms dan 12*packet loss* (2,5%).

b. Hasil tes throughput pada client 2

Berikut gambar 4.17 menunjukkan tampilan konfigurasi dan hasil tes*iperf*yang dilakukan pada *client* 2.



Gambar 4.17Hasil throughput di client 2 pada teknik ISATAP

Pada gambar 4.17 bisa dilihat perintah yang dijalankan adalah: iperf –c 2001:db8:0:10:20c:29ff:fe89:c211 –u –p 5123 –P 1 –b 12.5M –i 1 –t 10 –l 32768 –T 1. Dari perintah yang dijalankan, bisa dilihat bahwa PC ini melakukan tes *UDP* ke alamat IP 2001:db8:0:10:20c:29ff:fe89:c211melalui *port* 5123 dengan batasan *bandwith* 12,5 Mbps dan paket yang dikirim sebesar 32768 Byte. Tes ini berlangsung selama 10 detik dan dicatat hasilnya setiap satu detik.Berikut tabel 4.1 hasil dari semua tes pengukuran *througput*,*jitter* dan *packet loss* pada teknik transisi ISATAP.

Tabel 4.1 Throghput, jitter dan packet loss dari pengukuran pada teknik transisi ISATAP

Besar paket yang dikirmkan	**Througput (Mbits/sec)**	**Jitter (milliseconds)**	**Packet Loss (packet)**
128 Byte	8,35	0,115	264 (0,32%)
256 Byte	11,9	0,059	71 (0,12%)
512 Byte	12,5	0,061	0 (0%)
1024 Byte	12,4	0	0 (0%)
2048 Byte	12,3	0,319	0 (0%)
4096 Byte	12,4	0,164	0 (0%)
8192 Byte	12,4	0,123	0 (0%)
16384 Byte	12,4	1,341	0 (0%)
32768 Byte	12,2	1,496	12 (2,5%)
65500 Byte	12,5	6,187	0 (0%)
Rata-rata	**11,935**	**0,9865**	**0,294%**

Dari tabel 4.1 di atas diketahui bahwa *througput* terbesar terjadi pada besar paket yang kirimkan sebesar 512 Byte dan 65500 Byte. Untuk konektivitas yang menghasilkan *packet loss* terkecil didapatkan pada tes yang menggunakan besar paket kiriman sebesar 512 Byte,1024 Byte, 2048 Byte, 4096 Byte,8192 Byte, 16384 Byte dan 65500 Byte.

Selain untuk mendapatkan *throughput, iperf*juga digunakan untuk mendapatkan *jitter*. Dari tabel 4.1 di atas, perolehan *jitter* terbaik pada saat pengujian dengan besar paket 1024 Byte. Jika di rata-rata, hasil *jitter* menunjukkan angka 0,9865*milliseconds*. Adanya *jitter* dalam suatu jaringan menunjukkan adanya suatu kekurangan dalam sebuah infrastruktur perangkat dan atau sebuah protokol jaringan. Walaupun memiliki rata-rata *jitter* sebesar 0,9865*milliseconds*, nilai ini masih masuk kategori *good* dalam penerapan *VoIP* (Buetler, 2009: 17-19).

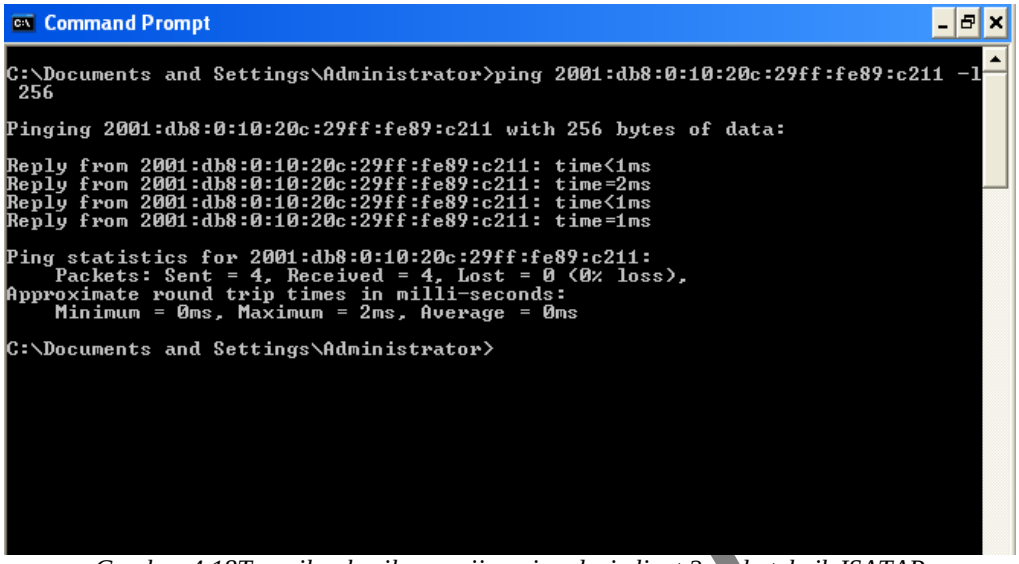
Untuk *packet loss* yang dihasilkan dari semua percobaan ini, semua mendapatkan hasil dibawah 2,5%. *Packet loss* terbesar terdapat pada pengujian dengan besar paket 32768 Byte. Jika di rata-rata, hasil *packet loss* menunjukkan angka 0,294%. Seperti halnya *jitter*,adanya *packet loss* dalam suatu jaringan menunjukkan adanya suatu kekurangan dalam sebuah infrastruktur perangkat dan atau sebuah protokol jaringan. Walaupun memiliki rata-rata *packet loss* sebesar 0,294%, nilai ini masih masuk kategori *good*dalam penerapan *VoIP*(Buetler, 2009: 17-19).

4.4.1.2. Pengukuran Round Trip Time (RTT) dan Response Time pada Teknik Transisi ISATAP

Round Trip Time(RTT) akan diukur menggunakan *tool ping* dan *response time* akan diukur menggunakan *wireshark. Tool ping* dan *wireshark* akan dijalankan pada sebuah *node* yang dipilih menjadi sumber dari tes ini. *Node* yang dipilih dalam tes ini adalah *client* 2. *Client* 2 bertindak sebagai sumber. Berikut beberapa gambar dari hasil tes RTT dan *response time* pada teknik transisi ISATAP.

a. Hasil pengukuran RTT pada client 2

Berikut gambar 4.18 tampilan konfigurasi dan hasil tes *ping* yang dilakukan pada *client* 2.

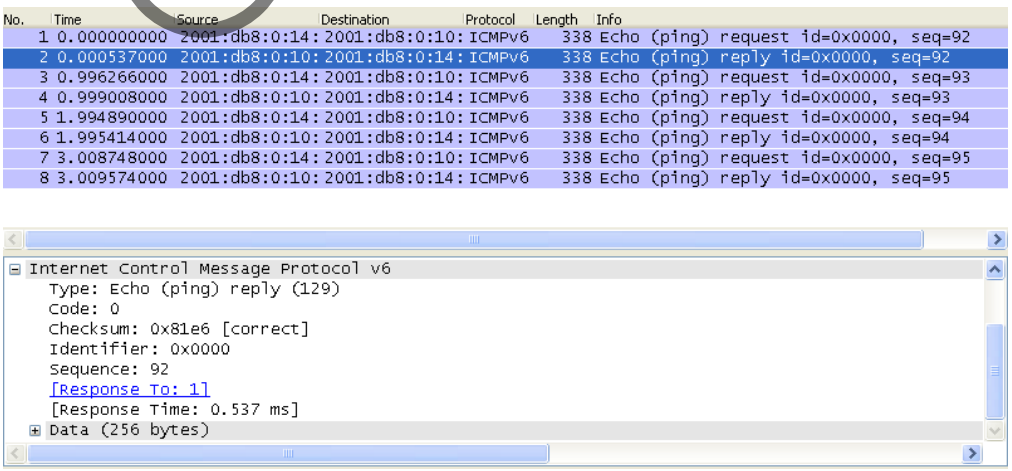


Gambar 4.18Tampilan hasil pengujian ping dari client 2 pada teknik ISATAP

Pada gambar 4.18 diatas bisa dilihat perintah yang dijalankan adalah ping2001:db8:0:10:20c:29ff:fe89:c211–l 256. Dari perintah yang dijalankan, bisa dilihat bahwa PC ini melakukan tes *ping* ke *client*1 yang beralamat IP 2001:db8:0:10:20c:29ff:fe89:c211 dengan besar paket yang dikirimkan sebesar 256 Byte. Tes ini mendapatkan RTT sebesar 0*milliseconds*.

b. Hasil pengukuran response time pada client 2

Berikut gambar 4.19 tampilan hasil tangkapan paket yang melalui teknik transisi ISATAP.



Gambar 4.19 Hasil penagkapan paket menggunakan wireshark pada teknik ISATAP

Dari gambar 4.19di atas, diketahui asal tes ping adalah 10.27.30.2, yang mana alamat IP tersebut merupakan alamat *IPv4 client* 2 dan tujuan dari tes ini adalah 2001:db8:0:10:20c:29ff:fe89:c211, yang mana alamat *IPv6* tersebut adalah alamat *client*1. Pada tampilan di atas dapat dilihat bahwa tes *ping* dari *client*2 ke *client*1 dengan besar paket yang dikirim sebesar 256 Byte menghasilkan *response time* 0,537*millisecond*.Berikut tabel 4.2 hasil dari pengukuran RTT dan *response time* pada teknik transisi ISATAP.

Tabel 4.2 RTT dan response time dari pengukuran pada teknik transisi ISATAP

Besar paket yang dikirmkan	**RTT (milliseconds)**	**Response time (milliseconds)**
128 Byte	0	0,81925
256 Byte	0	1,15725
512 Byte	1	1,62975
1024 Byte	0	1,16225
2048 Byte	1	1,5165
4096 Byte	1	1,0395
8192 Byte	1	1,19625
16384 Byte	4	1,809
32768 Byte	7	3,24025
65500 Byte	14	6,7735

Pada tabel 4.2 di atas bisa dilihat untuk RTT dan *response time* mengalami peningkatan seiring dengan besar paket yang dikirmkan. Hal ini dikarenakan semakin besar paket yang dikirmkan akan semakin besar pula waktu hantaran dari sebuah paket tersebut. Dari hasil yang didapatkan, RTT dan *response time* pada tes dengan besar paket 128 Byte dan1024 Byte menghasilkan waktu sekitar 0*millisecond*. Ini menggambarkan hasil yang baik untuk sebuah jaringan intranet.

4.4.2. Pengukuran pada Teknik Transisi Dual Stack

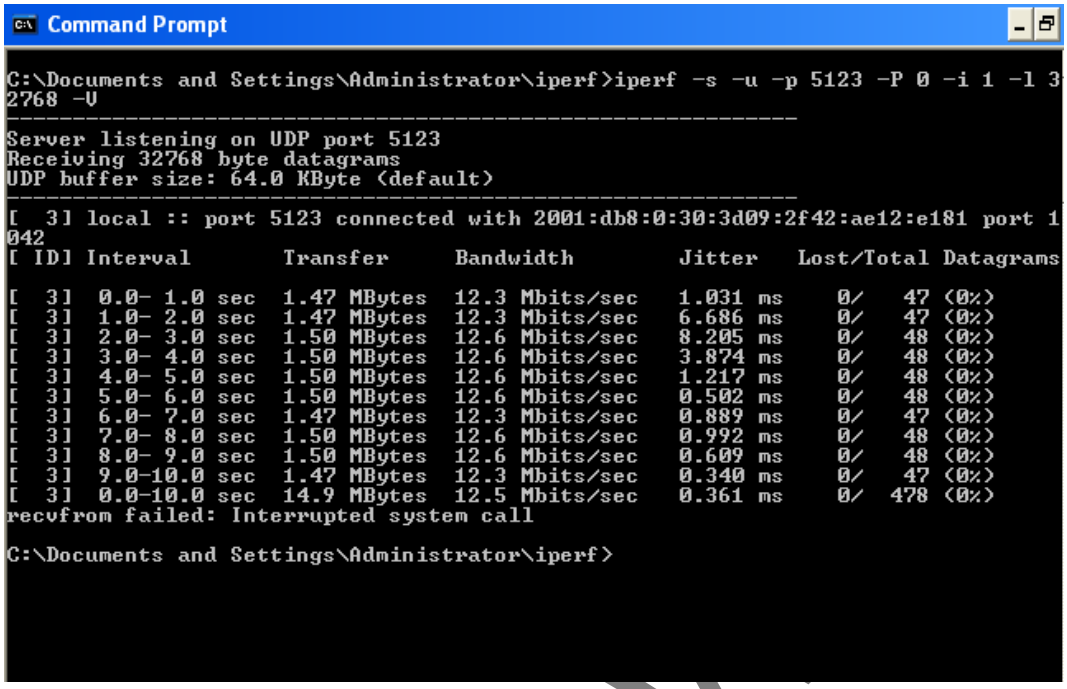
Pengujian pada teknik transisi *dual stack* sama dengan pengujian pada *dual stack.* Pengujian dilakukan pengukuran terhadap tiga parameter jaringan. Parameter itu adalah *throughput, jitter* dan *round trip time* (RTT). Pengujian akan dilakukan dengan mengirimkan paket dengan ukuran tertentu. Dalam pengiriman paket ini akan didapatkan data dari alat pengukur yang digunakan. Alat pengukur yang digunakan dalam pengujian ini ada 3, yaitu *iperf, ping* dan *wireshark.*

4.4.2.1. Pengukuran Throughput, Jitter dan Packet Loss pada Teknik Transisi Dual Stack

Sama seperti pengujian pada teknik transisi ISATAP, *througput, jitter* dan *packet loss* akan diukur menggunakan *iperf*. Perintah *iperf*akan dijalankan pada dua buah *node* yang dipilih menjadi sumber dan tujuan dari tes ini. *Node* yang dipilih dalam tes ini adalah *client* 1 dan *client* 2. *Client* 1 bertindak sebagai *server* atau tujuan dan *client* 2 bertindak sebagai *client* atau sumber. Berikut beberapa gambar dari hasil tes *througput* pada teknik *dual stack.*

a. Hasil listening tes throughput, jitter dan packet loss pada client 1

Berikut gambar 4.20 tampilan konfigurasi dan hasil tes*iperf*yang dilakukan pada *client* 1.

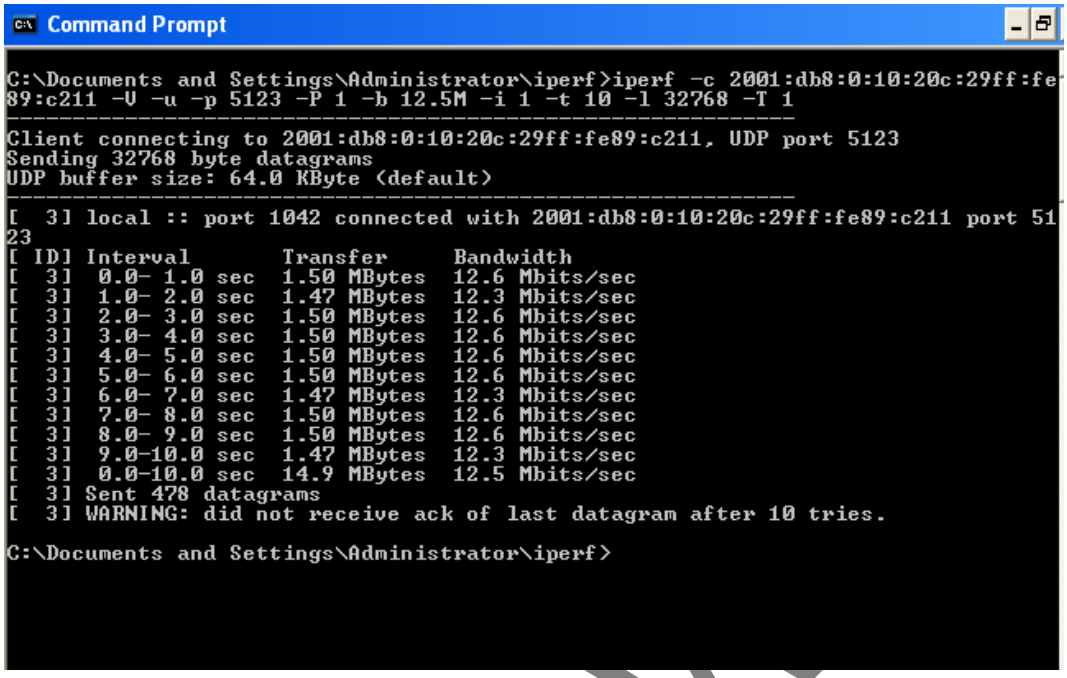


Gambar 4.20Hasil throughput, jitter packet loss dari client 1 pada teknikdual stack

Pada gambar 4.20di atas, bisa dilihat perintah yang digunakan adalah iperf –s –u –p 5123 –P 0 –i 1 –l 32768 –V. Dari perintah yang dijalankan, bahwa PC ini bertindak sebagai *server* dan melakukan *listening* tes UDP melalui *port* 5123 dan paket yang dikirim sebesar 32768 Byte. Dari hasil tes tersebut dapat dilihat bahwa proses ini mengirimkan total 14,9 MBytes dan menghasilkan*throughput bandwith*12,5 Mbps,*jitter* 0,361 ms dan 0*packet loss* (0%).

b. Hasil tes throughput pada client 2

Berikut gambar 4.21 tampilan konfigurasi dan hasil tes *iperf* yang dilakukan pada *client* 2.



Gambar 4.21Hasil pengujian ping dari client 2 pada teknik dual stack

Pada gambar 4.21 diatas bisa dilihat perintah yang dijalankan adalah: iperf –c 2001:db8:0:10:20c:29ff:fe89:c211 –u –p 5123 –P 1 –b 12.5M –i 1 –t 10 –l 32768 –T 1.

Dari proses *listening* yang berlangsung selama 10 detik dan dicatat per 1 detiknya, bisa dilihat bahwa PC ini melakukan tes UDP ke alamat IP 2001:db8:0:10:20c:29ff:fe89:c211melalui *port* 5123 dengan batasan *bandwith* 12,5 MBps dan paket yang dikirim sebesar 128 Byte.Berikut tabel 4.3 hasil dari semua tes pengukuran *througput, jitter* dan *packet loss* pada teknik transisi *dual stack.*

Tabel 4.3 Throghput, jitter dan packet loss dari pengukuran pada teknik transisi dual stack

Besar paket yang dikirmkan	**Througput (Mbits/sec)**	**Jitter (milliseconds)**	**Packet Loss (packet)**
128 Byte	9,51	0,059	368 (0,39%)
256 Byte	10,7	0,059	0 (0%)
512 Byte	12,0	1,051	0 (0%)
1024 Byte	10,7	0	0 (0%)

Tabel 4.3 (Sambungan)

Besar paket yang dikirmkan	Througput (Mbits/sec)	Jitter (milliseconds)	Packet Loss (packet)
2048 Byte	11,3	0,001	0 (0%)
4096 Byte	11,5	0,103	0 (0%)
8192 Byte	11,9	0,254	0 (0%)
16384 Byte	12,2	0,077	939 (0,75%)
32768 Byte	12,5	0,361	0 (0%)
65500 Byte	12,5	2,609	0 (0%)
Rata-rata	**11,481**	**0,4574**	**0,114%**

Dari tabel 4.3 di atas diketahui bahwa *througput* terbesar terjadi pada besar paket yang kirimkan sebesar 32768 Byte dan 65500 Byte. Untuk perolehan *packet loss* dari pengujian *iperf*, pada tes dengan besar paket yang dikirim sebesar 16384 Byte mendapatkan 0,75%, sedangkan lainnya memperoleh *packet loss* sebesar 0%. Selain untuk mendapatkan *throughput, iperf*juga digunakan untuk mendapatkan *jitter*. Dari tabel 4.3 di atas, perolehan *jitter* terbaik pada saat pengujian dengan besar paket 1024 Byte. Jika dirata-rata, hasil *jitter* menunjukkan angka 0,4574 milliseconds. Adanya *jitter* dalam suatu jaringan menunjukkan adanya suatu kekurangan dalam sebuah infrastruktur perangkat dan atau sebuah protokol jaringan. Walaupun memiliki rata-rata jitter sebesar 0,4574*milliseconds*, nilai ini masih masuk kategori *good* dalam penerapan VoIP(Buetler, 2009: 17-19).

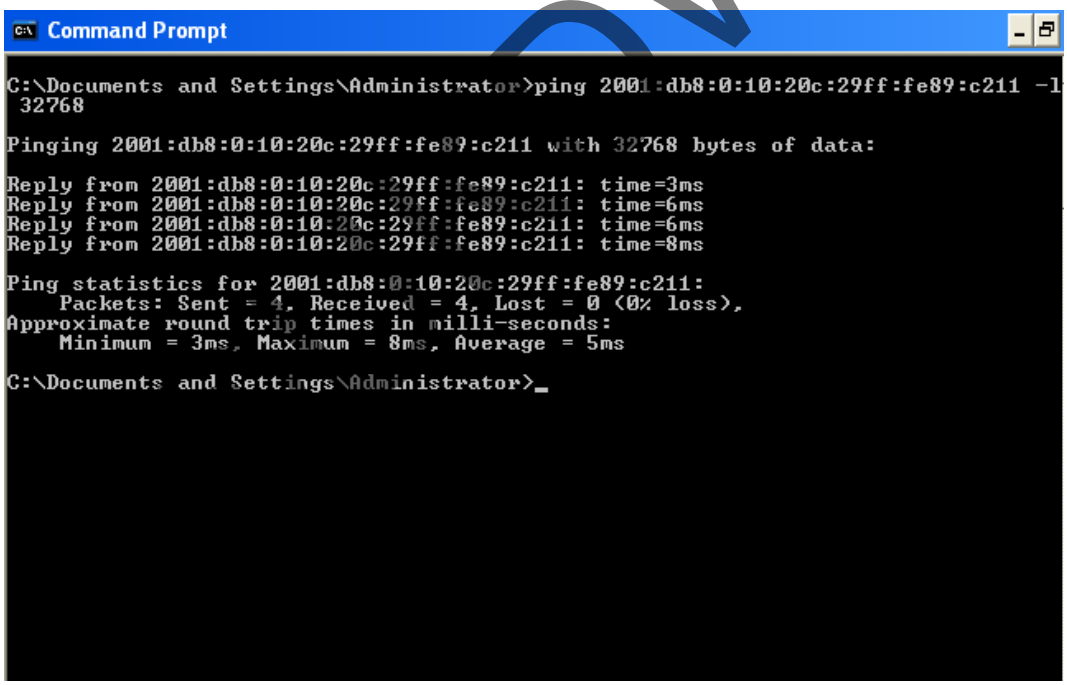
Untuk *packet loss* yang dihasilkan dari semua percobaan ini, semua mendapatkan hasil dibawah 5%. *Packet loss* terbesar terdapat pada pengujian dengan besar paket 16384 Byte. Jika dirata-rata, hasil *packet loss* menunjukkan angka 0,114% Sama seperti *jitter,packet loss* diharapkan dapat ditekan seminimal mungkin bahkan diharapkan tidak ada. Hal ini bertujuan untuk menciptakansuatu konektivitas jaringan yang baik. Walaupun memiliki rata-rata *packet loss* sebesar 0,114%, nilai ini masih masuk kategori *good* dalam penerapan VoIP(Buetler, 2009: 17-19).

4.4.2.2. Pengukuran Round Trip Time (RTT) dan Response Time pada Teknik Transisi Dual Stack

Round Trip Time(RTT) *tool ping* dan *response time* akan diukur menggunakan *wireshark.Tool ping* dan *wireshark* akan dijalankan pada sebuah *node* yang dipilih menjadi sumber dari tes ini. *Node* yang dipilih dalam tes ini adalah *client* 2. *Client* 2 bertindak sebagai sumber. Berikut beberapa gambar dari hasil tes RTT dan *response time* pada teknik transisi *dual stack.*

a. Hasil pengukuran RTT pada client 2

Berikut gambar 4.22 tampilan pengukuran RTTyang dilakukan di *client*2 pada teknik transisi *dual stack.*



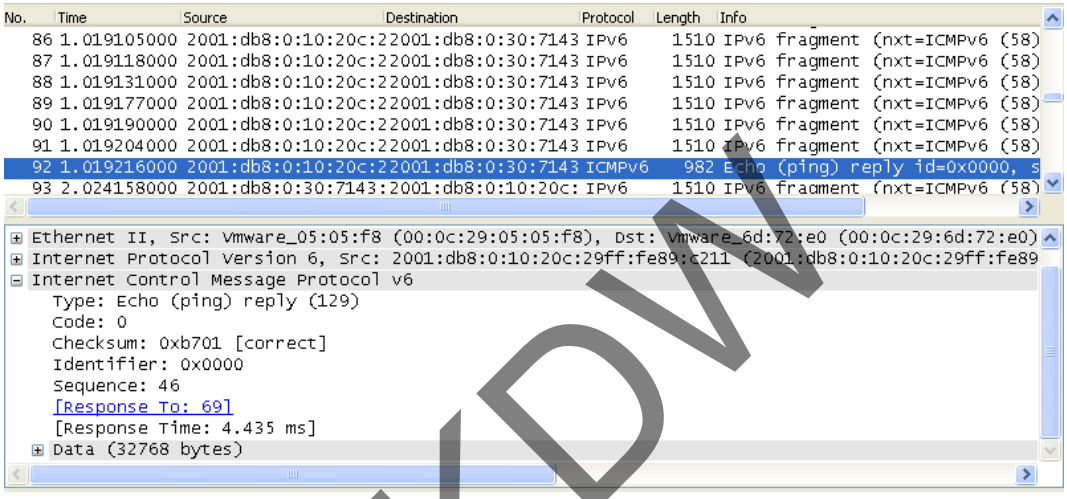
Gambar 4.22Hasil pengujian RTT dari client 2 pada teknik dual stack

Pada gambar 4.22 diatas bisa dilihat perintah yang dijalankan adalah *ping*2001:db8:0:10:20c:29ff:fe89:c211 –l 32768. Dari perintah yang dijalankan, bisa dilihat bahwa PC ini melakukan tes *ping* ke alamat

IP2001:db8:0:10:20c:29ff:fe89:c211 dengan besar paket yang dikirimkan sebesar 32768 Byte. Tes ini mendapatkan RTT sebesar 5*milliseconds*.

b. Hasil pengukuran response time pada *client* 2

Berikut gambar 4.23 tampilan hasil tangkapan paket yang melalui teknik transisi *dual stack.*



Gambar 4.23Hasil penagkapan paket menggunakan wireshark pada teknik dual stack

Dari gambar 4.23 di atas, diketahui asal tes *ping* dari 2001:db8:0:30:20c:29ff:fe6d:72e0 yang mana alamat IP tersebut merupakan alamat *unicast globalIPv6 client* 2 dan tujuan dari tes ini adalah 2001:db8:0:10:20c:29ff:fe89:c211, yang mana alamat *IPv6* tersebut adalah alamat *client*1. Pada tampilan di atas dapat dilihat bahwa tes *ping* dari *client* 2 ke *client1* dengan besar paket yang dikirim sebesar 32768 Byte menghasilkan *response time* 4,435 *milliseconds*.Berikut tabel 4.4 hasil dari pengukuran RTT dan *response time* pada teknik transisi *dual stack.*.

Tabel 4.4 RTT dan response time dari pengukuran pada teknik transisi dual stack

Besar paket yang dikirmkan	RTT (milliseconds)	Response time (milliseconds)
128 Byte	0	0,95075
256 Byte	1	2,4025
512 Byte	0	0,82575
1024 Byte	0	1,17425
2048 Byte	1	1,1995
4096 Byte	0	1,27675
8192 Byte	1	1,73925
16384 Byte	3	3,316
32768 Byte	5	4,64925
65500 Byte	9	7,00075

Pada tabel 4.4 di atas bisa dilihat untuk RTT dan *response time* mengalami peningkatan seiring dengan besar paket yang dikirmkan. Sama semakin teknik ISATAP, semakin besar paket yang dikirmkan akan semakin besar pula waktu hantaran dari sebuah paket tersebut. Dari hasil yang didapatkan, RTT dan *response time* pada tes dengan besar paket 128 Byte hingga 8192 Byte menghasilkan waktu sekitar 1 millisecond. Ini menggambarkan hasil yang baik untuk sebuah jaringan intranet.

4.5. Pengamatan Tingkat Kerumitan dalam Penerapan Teknik Transisi ISATAP dan Dual Stack

Dalam penerapannya, teknik *dual stack* cenderung lebih mudah untuk digunakan. Dari sisi infrastruktur perangkat jaringan, dalam penerapannya teknik transisi *dual stack*tidak memerlukan DNS *server*. Pada penerapan teknik transisi ISATAP dibutuhkan adanya sebuah DNS*server*. PadaISATAPDNS*server*

berfungsi untuk menetukan*gateway* dari ISATAP. Teknik transisi ISATAP memang memiliki ketergantungan terhadap DNS *server*, namun teknik ISATAP memiliki kelebihan dalam interoperasi pada *IPv4* dan *IPv6*, sedangkan pada teknik transisi *dual* stack *tidak*.

4.6. Pengamatan Tingkat Keamanan pada Teknik Transisi ISATAP dan Dual Stack

Teknik transisi ISATAP dan *dual stack* memiliki karakteristik tersendiri. Karakteristik pada ISATAP adanya *automatic tunnel* dan pada *dual stack*tidak memiliki *tunnel. Tunnel* pada ISATAP membuat setiap *node* tidak dapat berhubungan secara bebas. Hanya *node* yang dikonfigurasi untuk ikut serta dalam *tunnel* saja yang bisa melakukan konektivitas. Misalnya pada sistem dalam penelitian ini, *client* 1 tidak dapat melakukan konektivitas dengan *router* 2. Sedangkan pada *dual stack,* setiap *node* dapat berhubungan secara langsung. Pada *dual stack* juga menggunakan *dual stack* secara bersamaan. *Stack* itu adalah *stack IPv4* dan *stack IPv6* .

Dengan aktifnya kedua *stack* ini secara bersamaan, maka jalur yang ada dalam sistem ini menjadi 2 jalur. Adanya dua jalur dalam sebuah dalam sebuah sistem dapat memberi keuntungan jika sebuah jalur mengalami kegagalan koneksi, maka jalur lainnya dapat digunakan. Namun,semakin banyak jalur untuk melakukan akses, maka semakin banyak celah dari jaringan yang memungkinkan adanya *unathorized acces.*

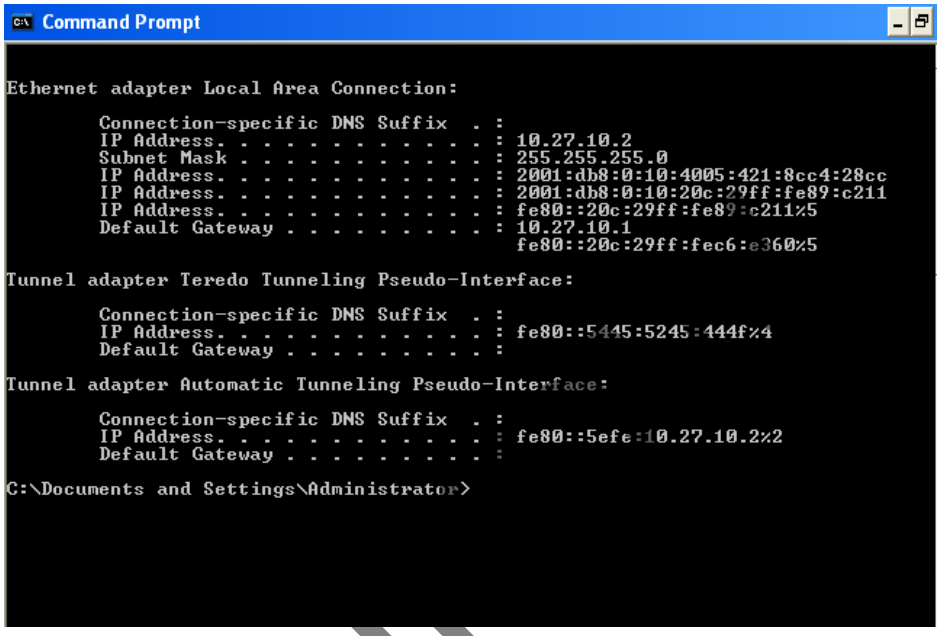
ISATAP biasanya membangun *PRL*-nya dengan melakukan konsultasi pada DNS, maka pada model OSI, ISATAP merupakan *lower-layer protocol* yang tergantung pada lapisan yang lebih tinggi. Sebuah bentuk lingkaran pada mekanisme teknik transisi ISATAP dihindari dengan mengandalkan DNS *serverIPv4* yang tidak tergantung pada *routingIPv6* yang dibangun, namun, ini adalah pelanggaran terhadap prinsip-prinsip desain jaringan, dan terasa rapuh untuk beberapa spesialis jaringan.

ISATAP membawa risiko keamanan yang sama seperti *6over4*, yaitu*link* maya IPv4 harus dipisahkan dengan hati-hati pada bagian jaringan, sehingga *hostIPv4*eksternal tidak bisa berpura-pura menjadi bagian dari *link* ISATAP. Itu biasanya dilakukan dengan memastikan bahwa *proto*-41 (*6in4*) tidak dapat melewati *firewall*.

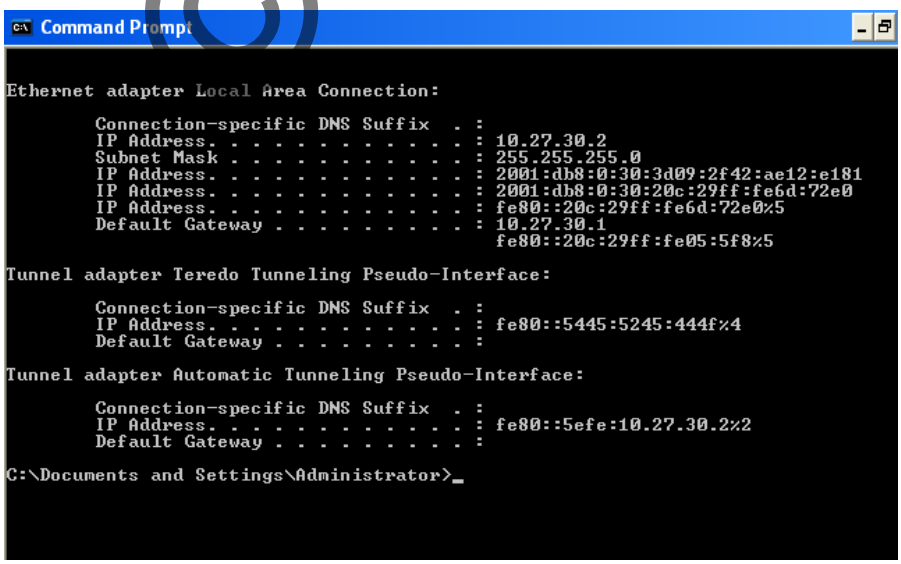
Lampiran A

Konfigurasi Alamat IP Perangkat Jaringan pada Teknik Transisi Dual Stack

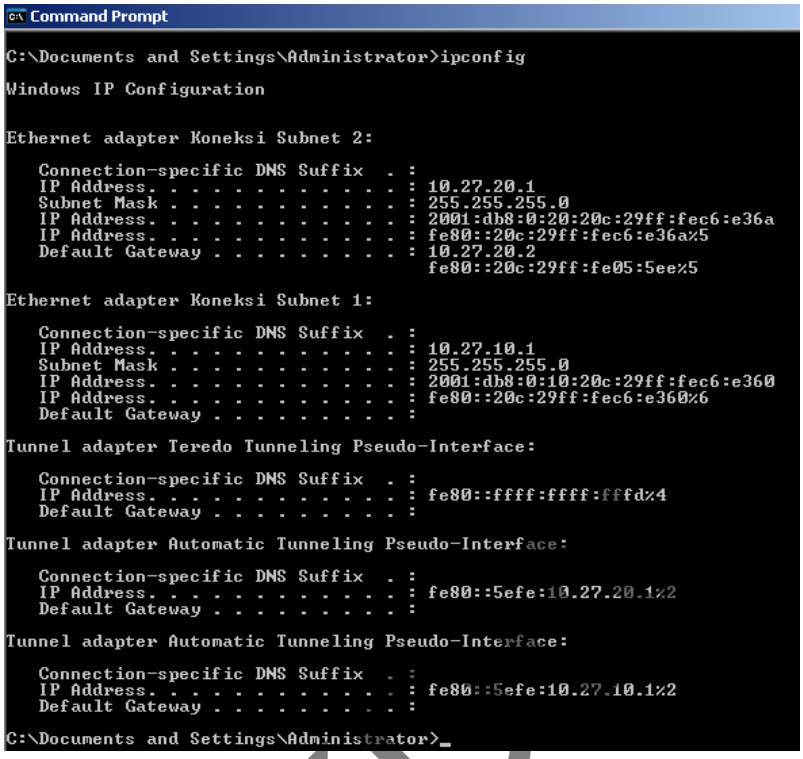
A.1. Konfigurasi Alamat IP pada Client 1 Dual Stack



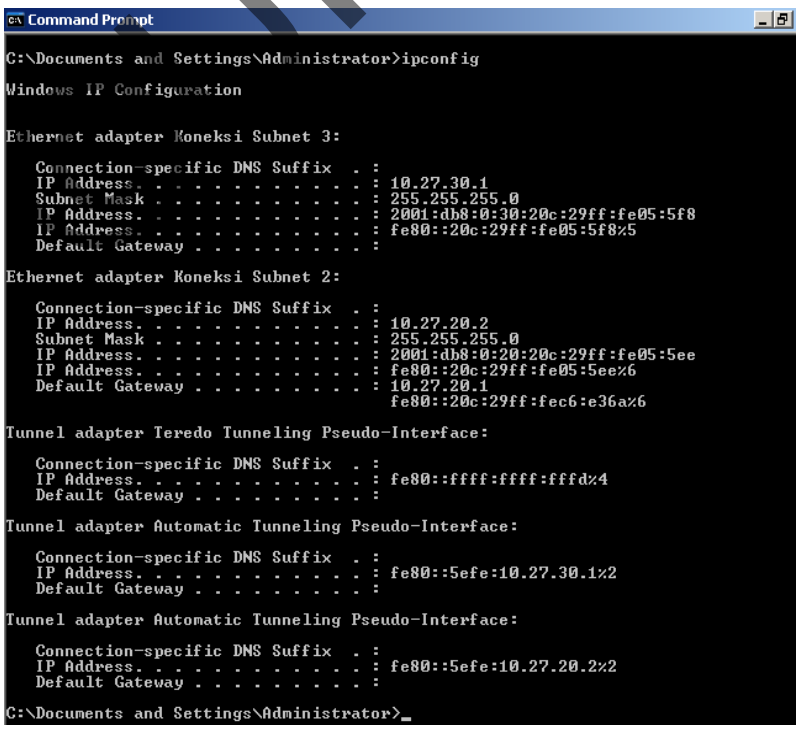
A.2. Konfigurasi Alamat IP pada Client 2 Dual Stack



A.3. Konfigurasi Alamat IP pada Router 1 Dual Stack



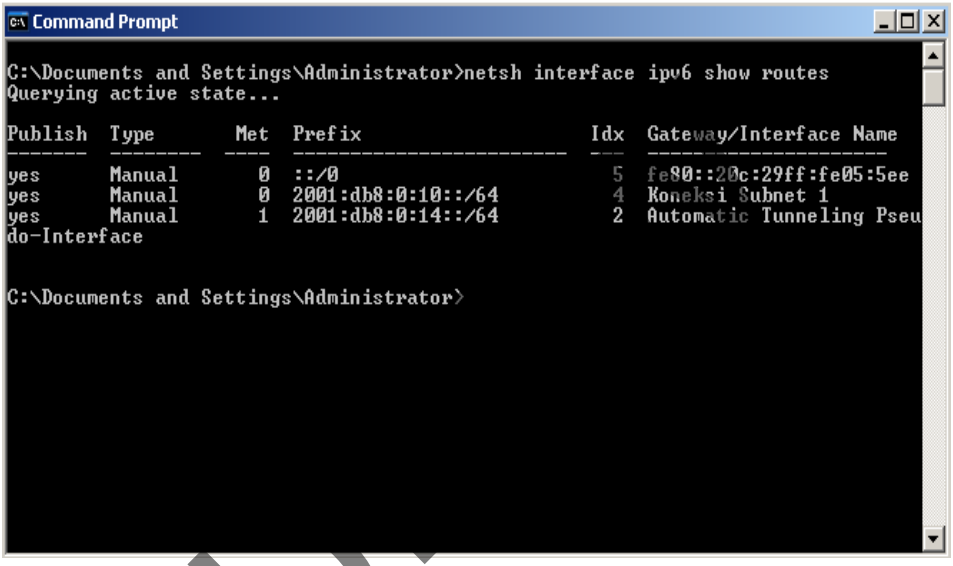
A.4. Konfigurasi Alamat IP pada Router 2 Dual Stack



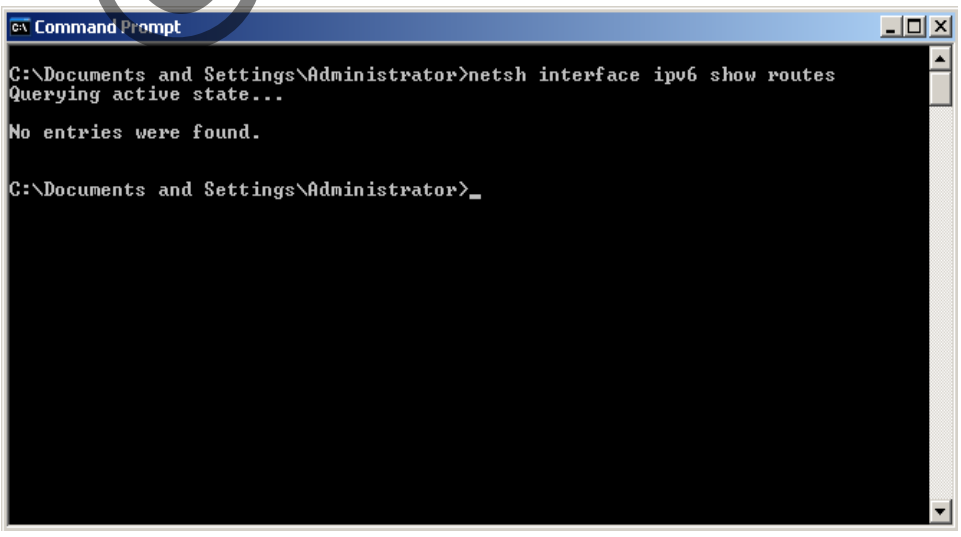
LAMPIRAN B

Tabel Routing pada Setiap Router untuk Teknik Transisi ISATAP dan Dual Stack

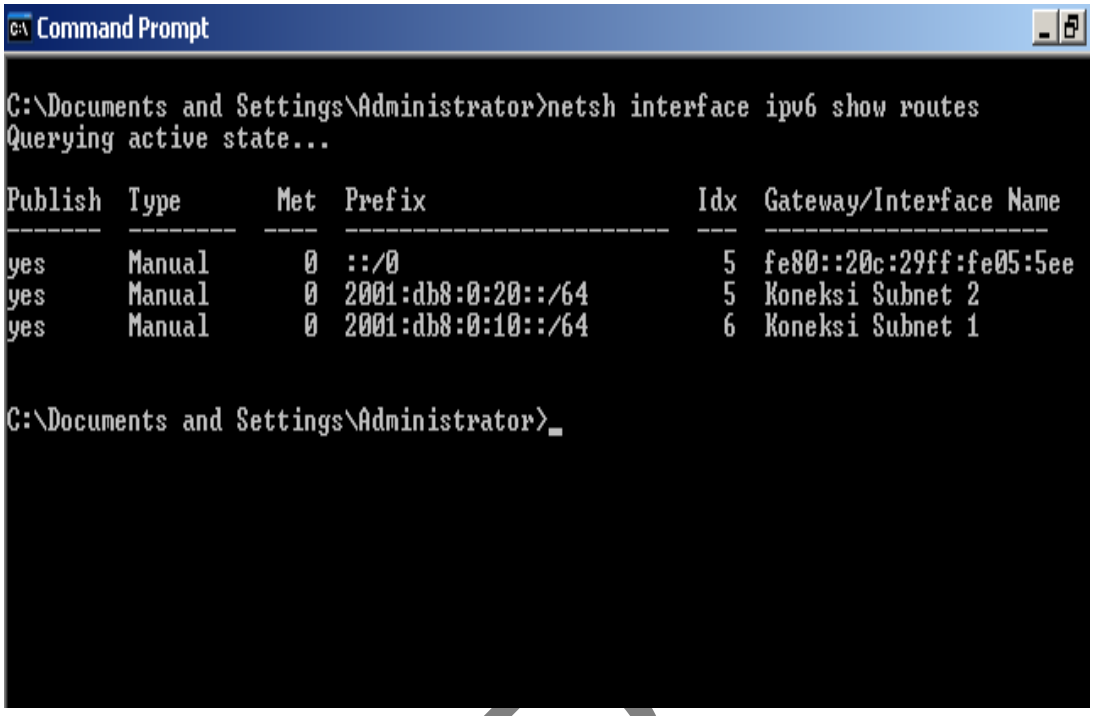
B.1. Tabel Routing pada Router ISATAP pada Teknik Transisi ISATAP



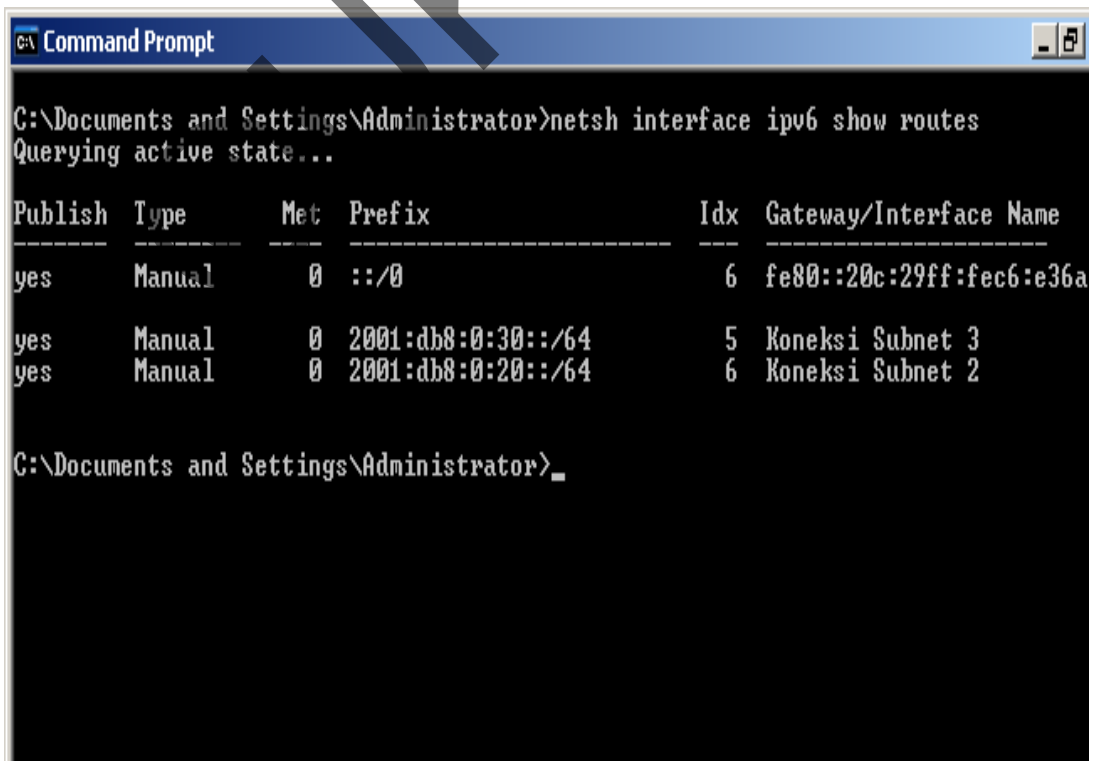
B.2. Tabel Routing pada Router 2 pada Teknik Transisi ISATAP



B.3. Tabel Routing pada Router 1 pada Teknik Transisi Dual Stack



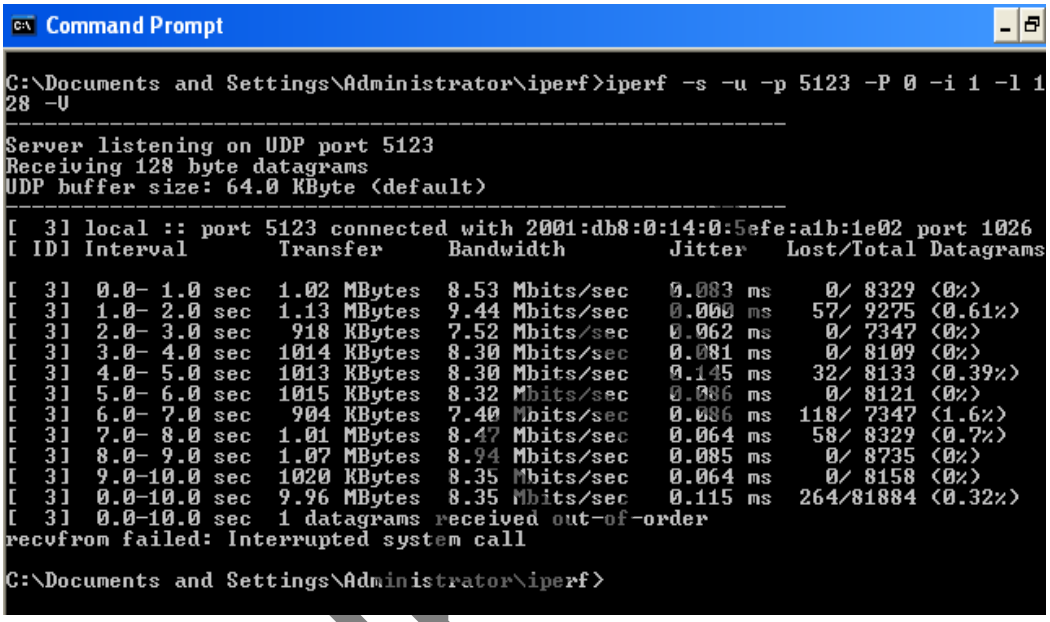
B.4. Tabel Routing pada Router 2 pada Teknik Transisi Dual Stack



LAMPIRAN C

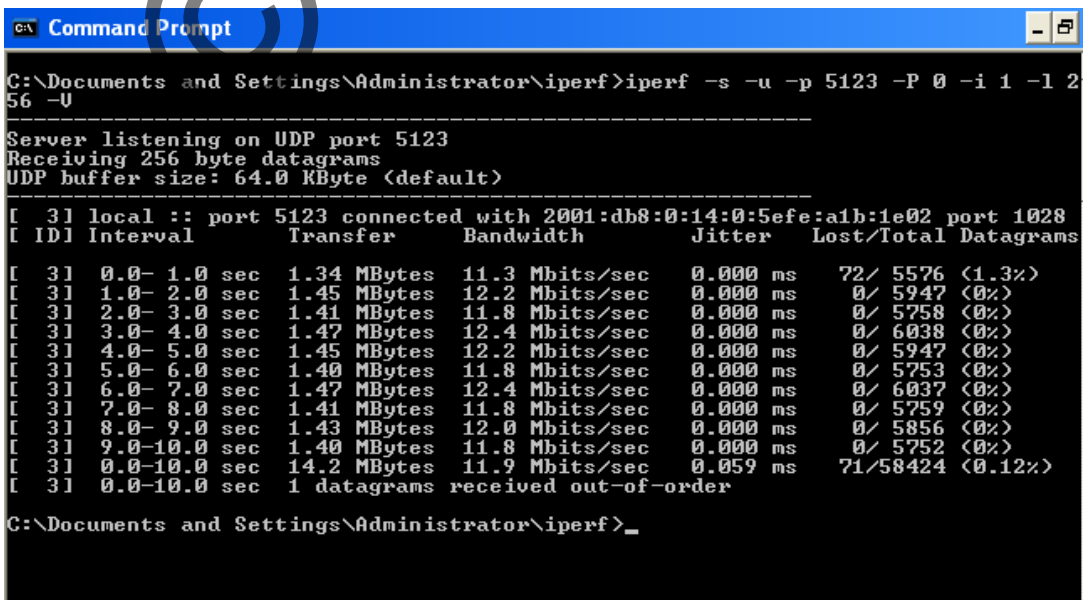
Hasil Tes Iperf pada Teknik Transisi ISATAP

C.1. Tes Iperf dengan Beban Paket 128 Byte



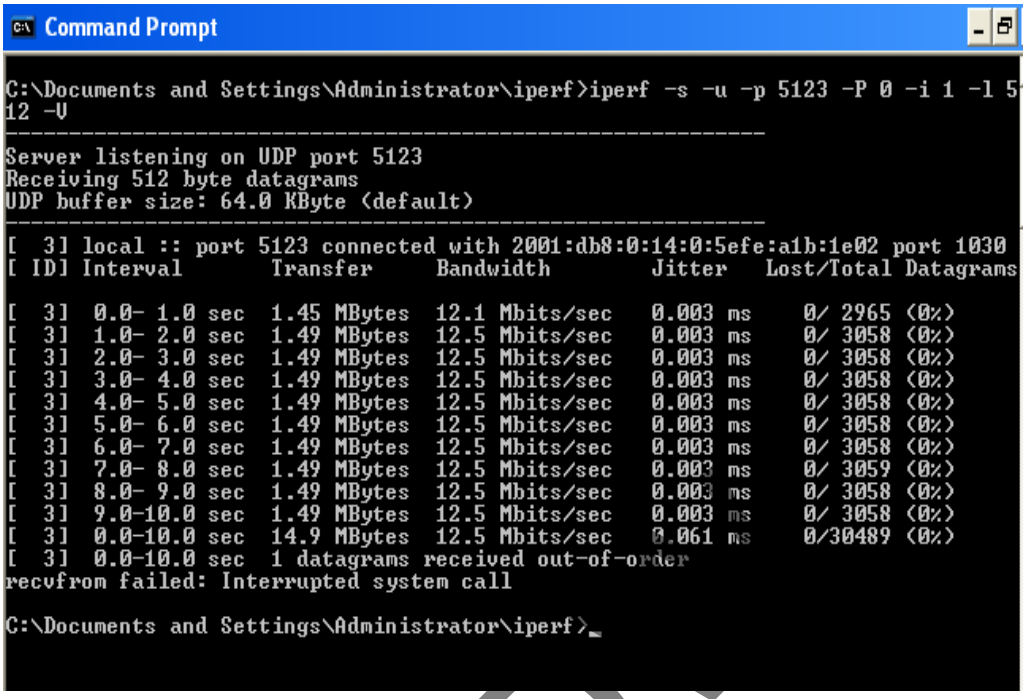
```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 1
28 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 128 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:14:0:5efe:a1b:1e02 port 1026
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.02 MBytes  8.53 Mbits/sec   0.083 ms    0/ 8329 (0%)
[  3]  1.0- 2.0 sec  1.13 MBytes  9.44 Mbits/sec   0.000 ms   57/ 9275 (0.61%)
[  3]  2.0- 3.0 sec   918 KBytes  7.52 Mbits/sec   0.062 ms    0/ 7347 (0%)
[  3]  3.0- 4.0 sec  1014 KBytes  8.30 Mbits/sec   0.081 ms    0/ 8109 (0%)
[  3]  4.0- 5.0 sec  1013 KBytes  8.30 Mbits/sec   0.145 ms   32/ 8133 (0.39%)
[  3]  5.0- 6.0 sec  1015 KBytes  8.32 Mbits/sec   0.086 ms    0/ 8121 (0%)
[  3]  6.0- 7.0 sec   904 KBytes  7.40 Mbits/sec   0.086 ms  118/ 7347 (1.6%)
[  3]  7.0- 8.0 sec  1.01 MBytes  8.47 Mbits/sec   0.064 ms   58/ 8329 (0.7%)
[  3]  8.0- 9.0 sec  1.07 MBytes  8.94 Mbits/sec   0.085 ms    0/ 8735 (0%)
[  3]  9.0-10.0 sec  1020 KBytes  8.35 Mbits/sec   0.064 ms    0/ 8158 (0%)
[  3]  0.0-10.0 sec  9.96 MBytes  8.35 Mbits/sec   0.115 ms  264/81884 (0.32%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

C.2. Tes Iperf dengan Beban Paket 256 Byte

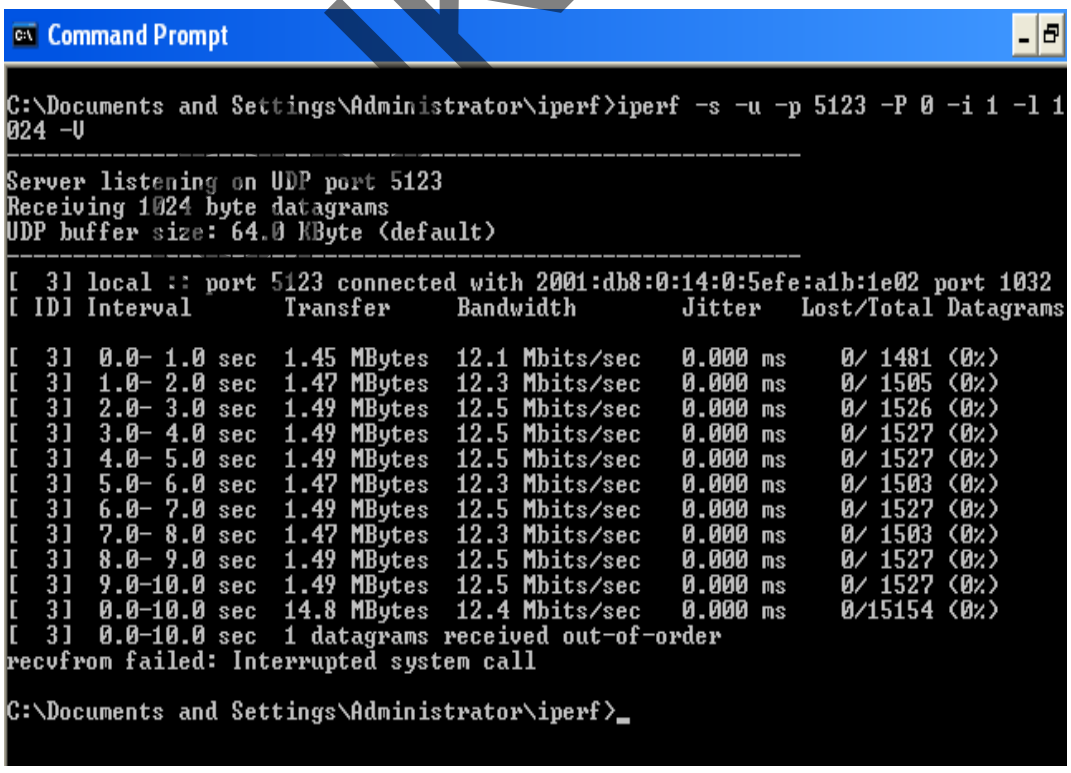


```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 2
56 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 256 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:14:0:5efe:a1b:1e02 port 1028
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.34 MBytes  11.3 Mbits/sec   0.000 ms   72/ 5576 (1.3%)
[  3]  1.0- 2.0 sec  1.45 MBytes  12.2 Mbits/sec   0.000 ms    0/ 5947 (0%)
[  3]  2.0- 3.0 sec  1.41 MBytes  11.8 Mbits/sec   0.000 ms    0/ 5758 (0%)
[  3]  3.0- 4.0 sec  1.47 MBytes  12.4 Mbits/sec   0.000 ms    0/ 6038 (0%)
[  3]  4.0- 5.0 sec  1.45 MBytes  12.2 Mbits/sec   0.000 ms    0/ 5947 (0%)
[  3]  5.0- 6.0 sec  1.40 MBytes  11.8 Mbits/sec   0.000 ms    0/ 5753 (0%)
[  3]  6.0- 7.0 sec  1.47 MBytes  12.4 Mbits/sec   0.000 ms    0/ 6037 (0%)
[  3]  7.0- 8.0 sec  1.41 MBytes  11.8 Mbits/sec   0.000 ms    0/ 5759 (0%)
[  3]  8.0- 9.0 sec  1.43 MBytes  12.0 Mbits/sec   0.000 ms    0/ 5856 (0%)
[  3]  9.0-10.0 sec  1.40 MBytes  11.8 Mbits/sec   0.000 ms    0/ 5752 (0%)
[  3]  0.0-10.0 sec  14.2 MBytes  11.9 Mbits/sec   0.059 ms   71/58424 (0.12%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order

C:\Documents and Settings\Administrator\iperf>_
```

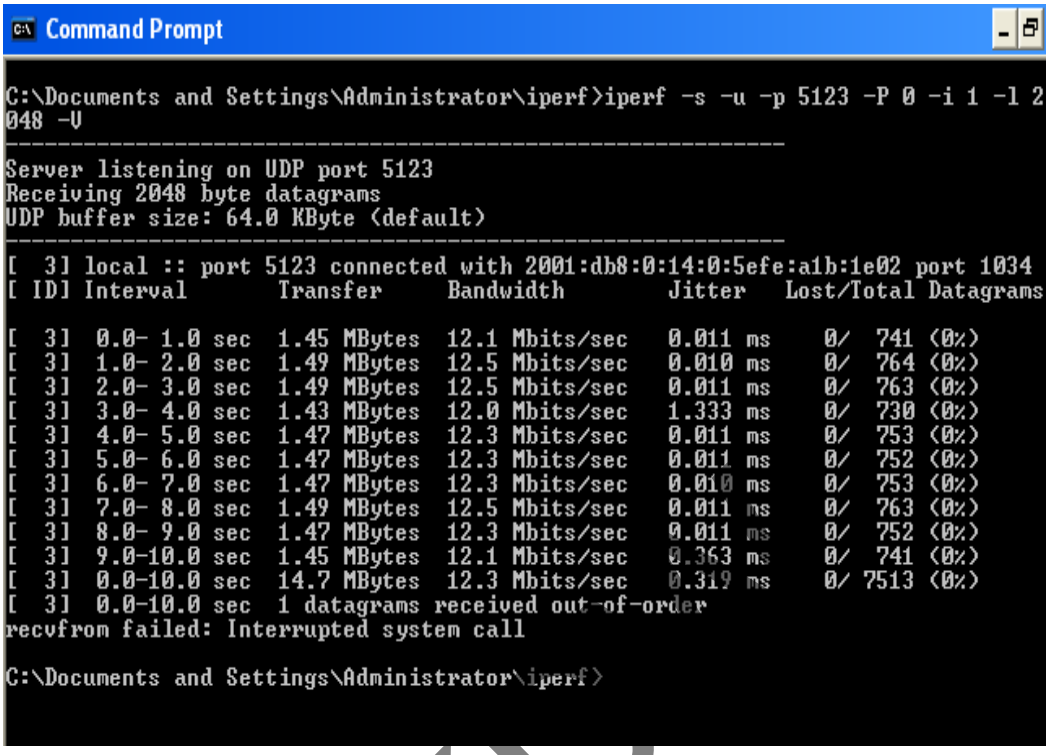
C.3. Tes Iperf dengan Beban Paket 512 Byte



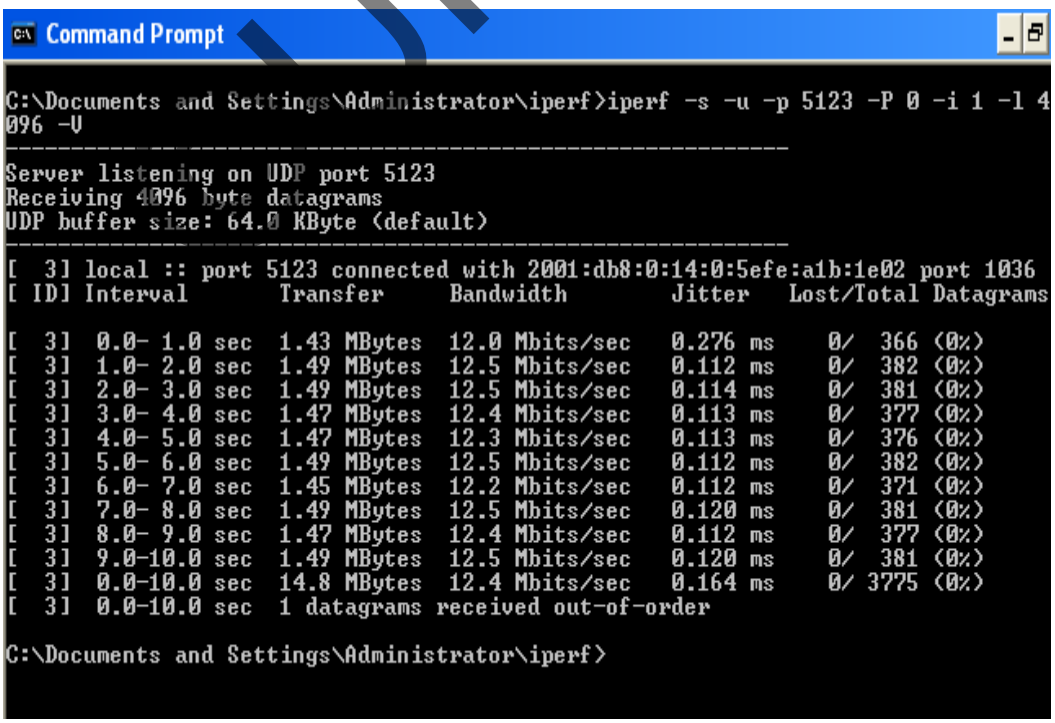
C.4. Tes Iperf dengan Beban Paket 1024 Byte



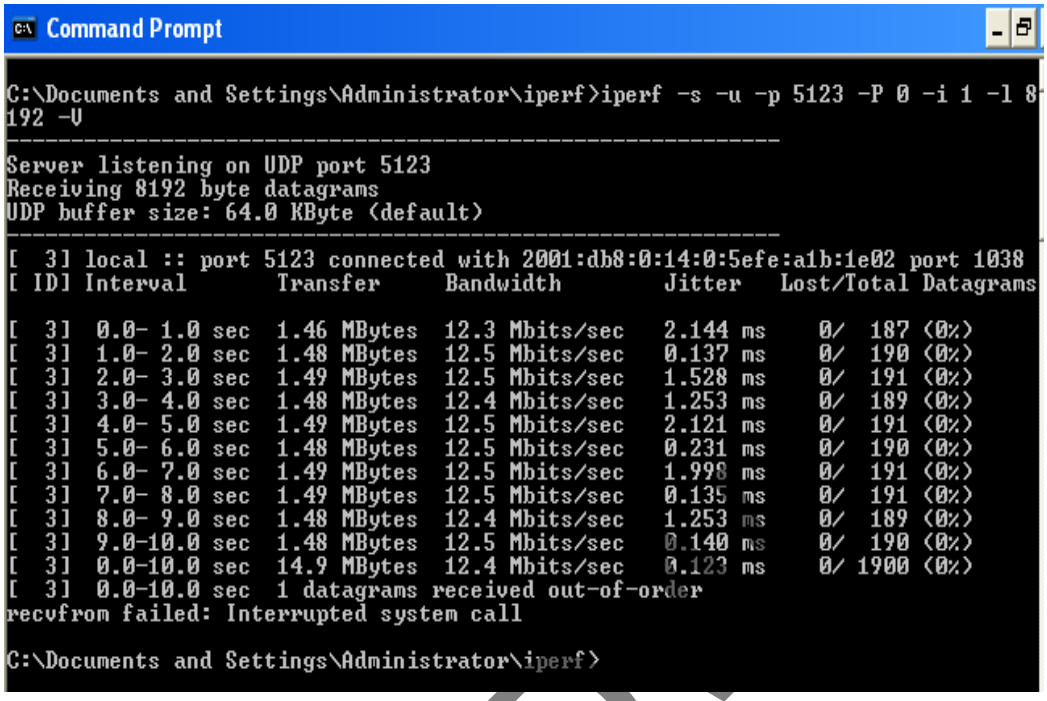
C.5. Tes Iperf dengan Beban Paket 2048 Byte



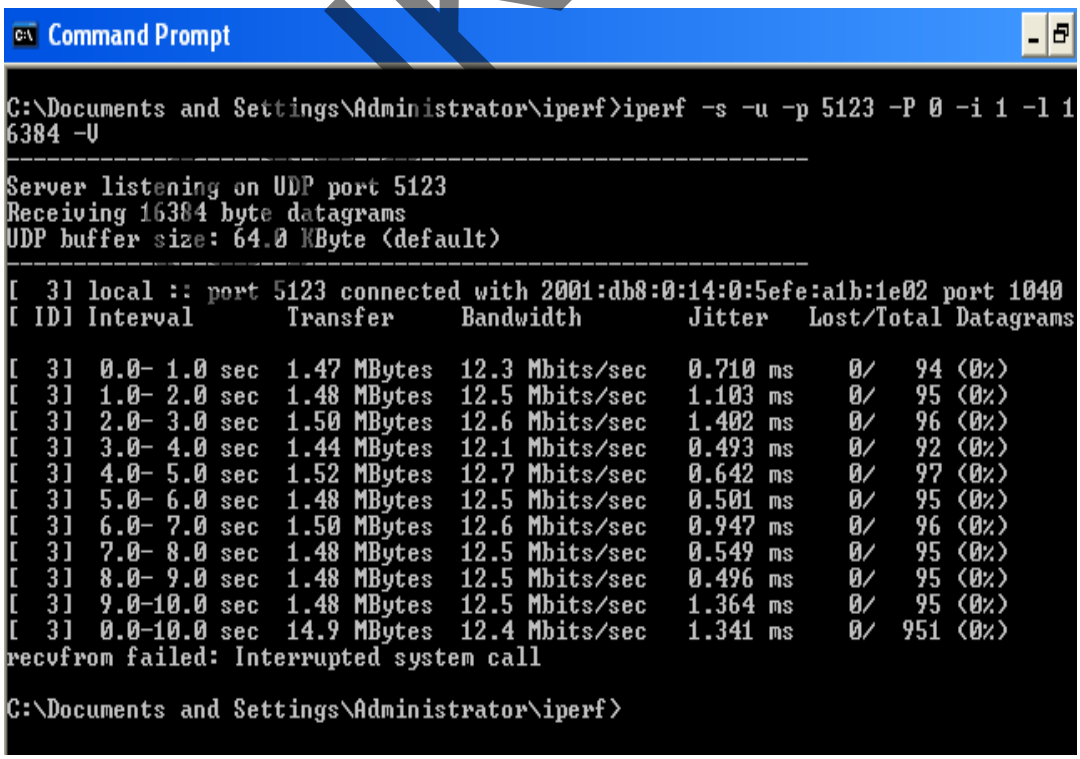
C.6. Tes Iperf dengan Beban Paket 4096 Byte



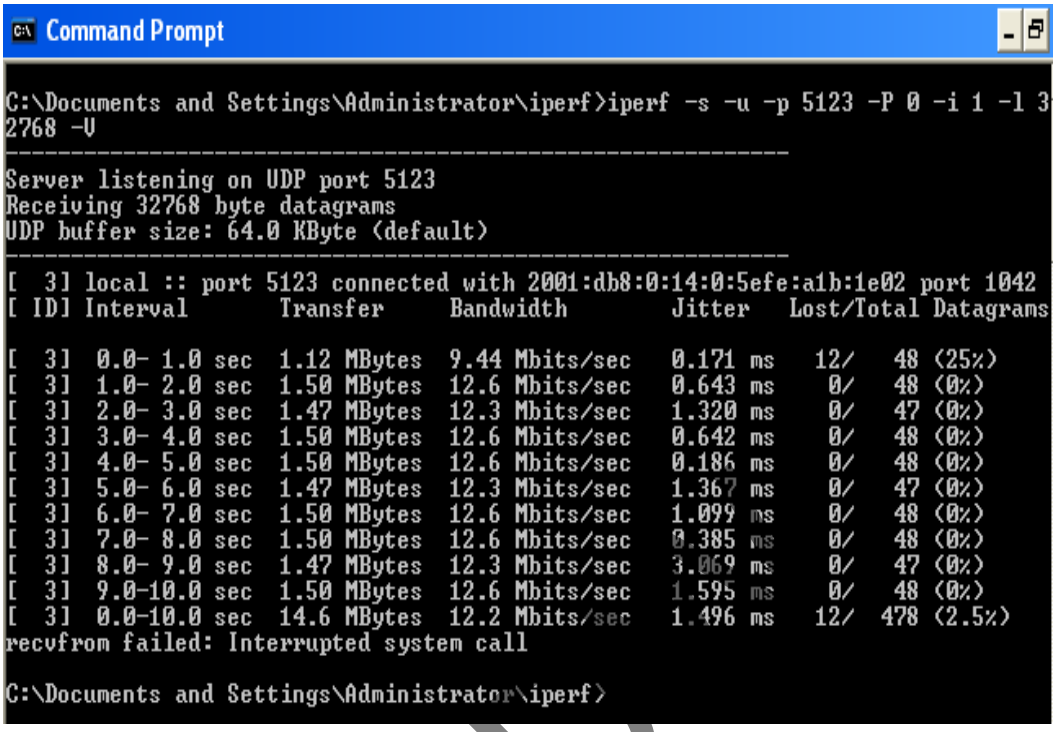
C.7. Tes Iperf dengan Beban Paket 8192 Byte



C.8Tes Iperf dengan Beban Paket 16382 Byte

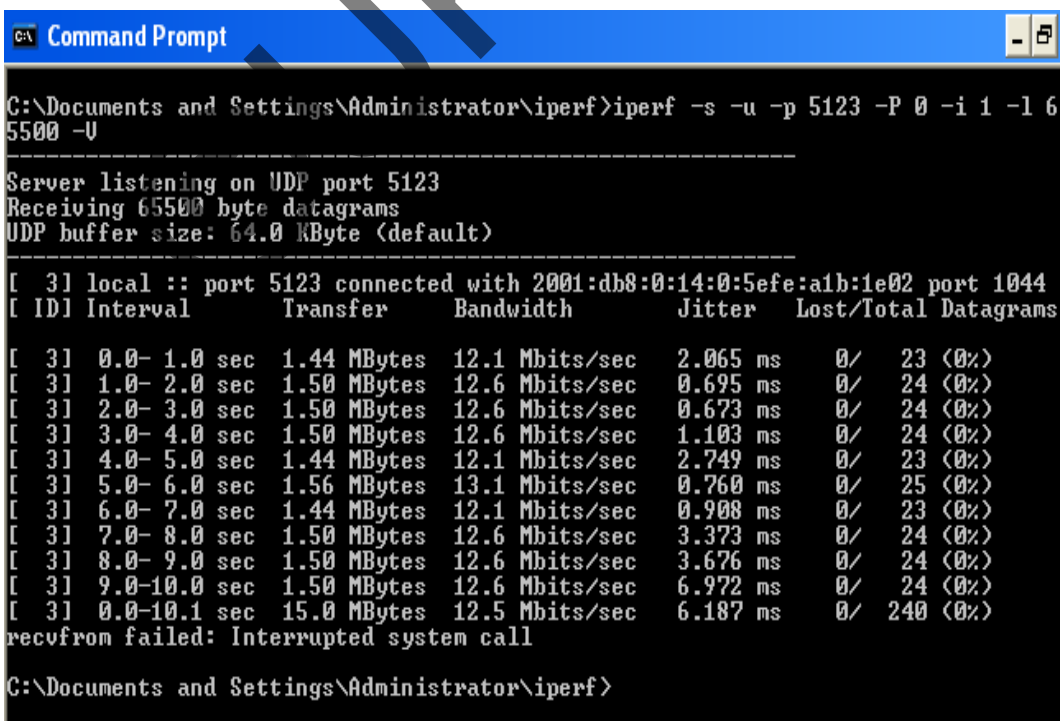


C.9. Tes Iperf dengan Beban Paket 32768 Byte



```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 3
2768 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 32768 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:14:0:5efe:a1b:1e02 port 1042
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams

[  3]  0.0- 1.0 sec  1.12 MBytes  9.44 Mbits/sec   0.171 ms   12/   48 (25%)
[  3]  1.0- 2.0 sec  1.50 MBytes  12.6 Mbits/sec   0.643 ms    0/   48 (0%)
[  3]  2.0- 3.0 sec  1.47 MBytes  12.3 Mbits/sec   1.320 ms    0/   47 (0%)
[  3]  3.0- 4.0 sec  1.50 MBytes  12.6 Mbits/sec   0.642 ms    0/   48 (0%)
[  3]  4.0- 5.0 sec  1.50 MBytes  12.6 Mbits/sec   0.186 ms    0/   48 (0%)
[  3]  5.0- 6.0 sec  1.47 MBytes  12.3 Mbits/sec   1.367 ms    0/   47 (0%)
[  3]  6.0- 7.0 sec  1.50 MBytes  12.6 Mbits/sec   1.099 ms    0/   48 (0%)
[  3]  7.0- 8.0 sec  1.50 MBytes  12.6 Mbits/sec   0.385 ms    0/   48 (0%)
[  3]  8.0- 9.0 sec  1.47 MBytes  12.3 Mbits/sec   3.069 ms    0/   47 (0%)
[  3]  9.0-10.0 sec  1.50 MBytes  12.6 Mbits/sec   1.595 ms    0/   48 (0%)
[  3]  0.0-10.0 sec  14.6 MBytes  12.2 Mbits/sec   1.496 ms   12/  478 (2.5%)
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

C.10. Tes Iperf dengan Beban Paket 65500 Byte

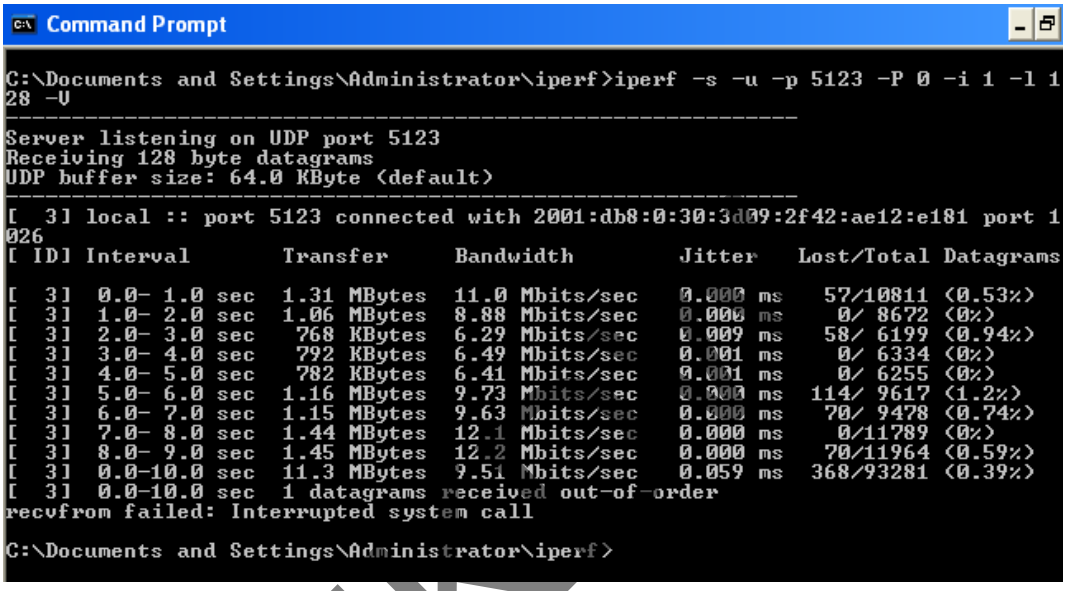


```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 6
5500 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 65500 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:14:0:5efe:a1b:1e02 port 1044
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams

[  3]  0.0- 1.0 sec  1.44 MBytes  12.1 Mbits/sec   2.065 ms    0/   23 (0%)
[  3]  1.0- 2.0 sec  1.50 MBytes  12.6 Mbits/sec   0.695 ms    0/   24 (0%)
[  3]  2.0- 3.0 sec  1.50 MBytes  12.6 Mbits/sec   0.673 ms    0/   24 (0%)
[  3]  3.0- 4.0 sec  1.50 MBytes  12.6 Mbits/sec   1.103 ms    0/   24 (0%)
[  3]  4.0- 5.0 sec  1.44 MBytes  12.1 Mbits/sec   2.749 ms    0/   23 (0%)
[  3]  5.0- 6.0 sec  1.56 MBytes  13.1 Mbits/sec   0.760 ms    0/   25 (0%)
[  3]  6.0- 7.0 sec  1.44 MBytes  12.1 Mbits/sec   0.908 ms    0/   23 (0%)
[  3]  7.0- 8.0 sec  1.50 MBytes  12.6 Mbits/sec   3.373 ms    0/   24 (0%)
[  3]  8.0- 9.0 sec  1.50 MBytes  12.6 Mbits/sec   3.676 ms    0/   24 (0%)
[  3]  9.0-10.0 sec  1.50 MBytes  12.6 Mbits/sec   6.972 ms    0/   24 (0%)
[  3]  0.0-10.1 sec  15.0 MBytes  12.5 Mbits/sec   6.187 ms    0/  240 (0%)
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

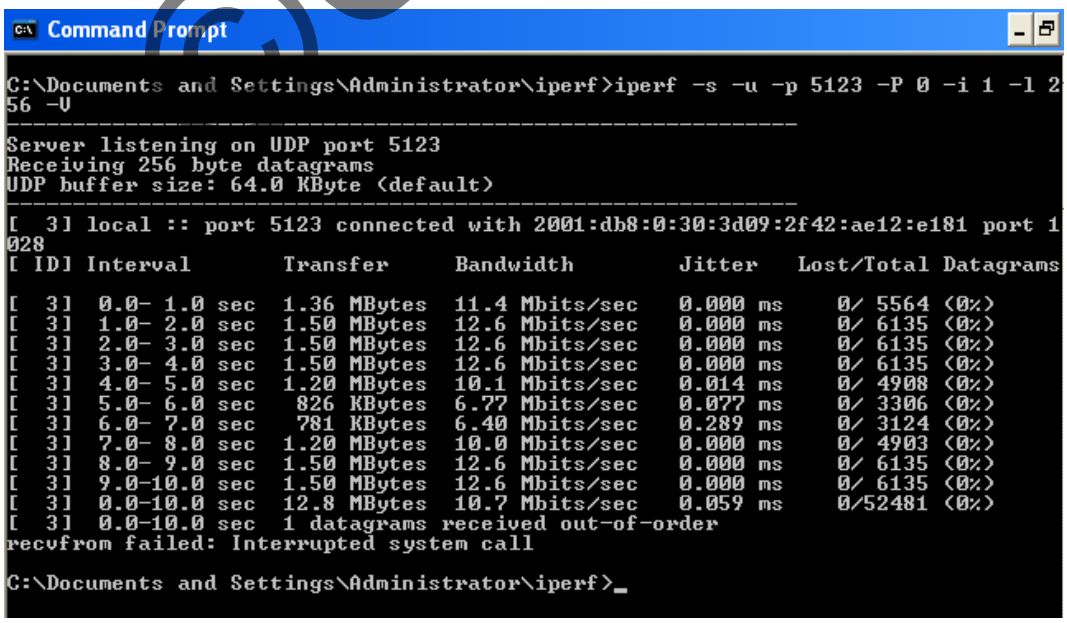
LAMPIRAN D

Hasil Tes Iperfpada Teknik Transisi Dual Stack

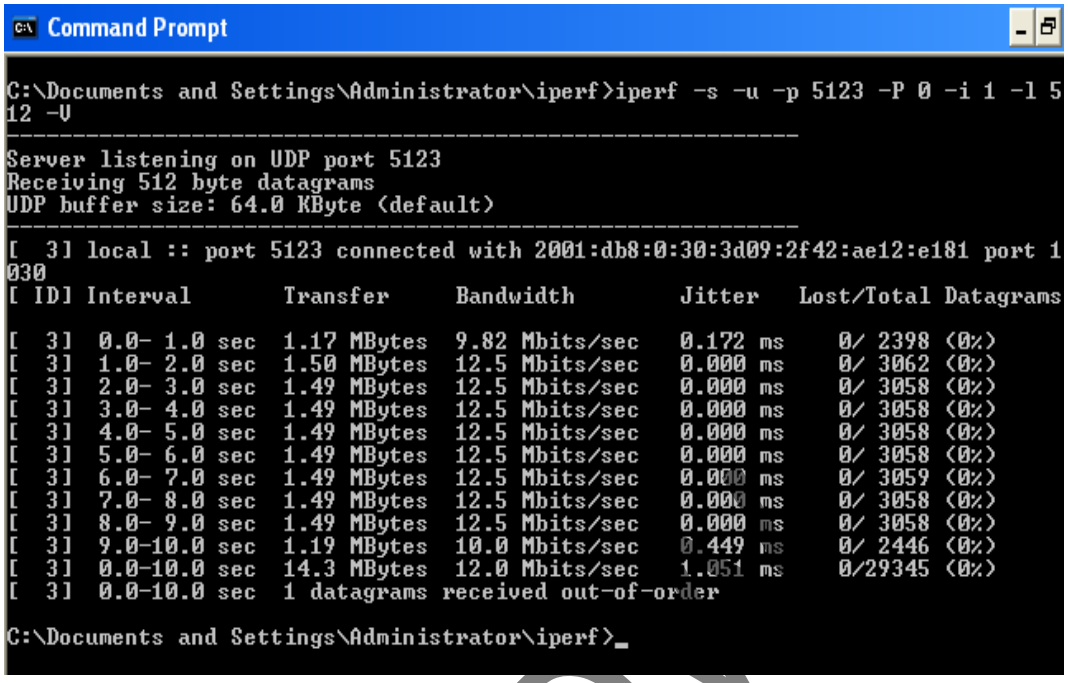
D.1. Tes Iperf dengan Beban Paket 128 Byte



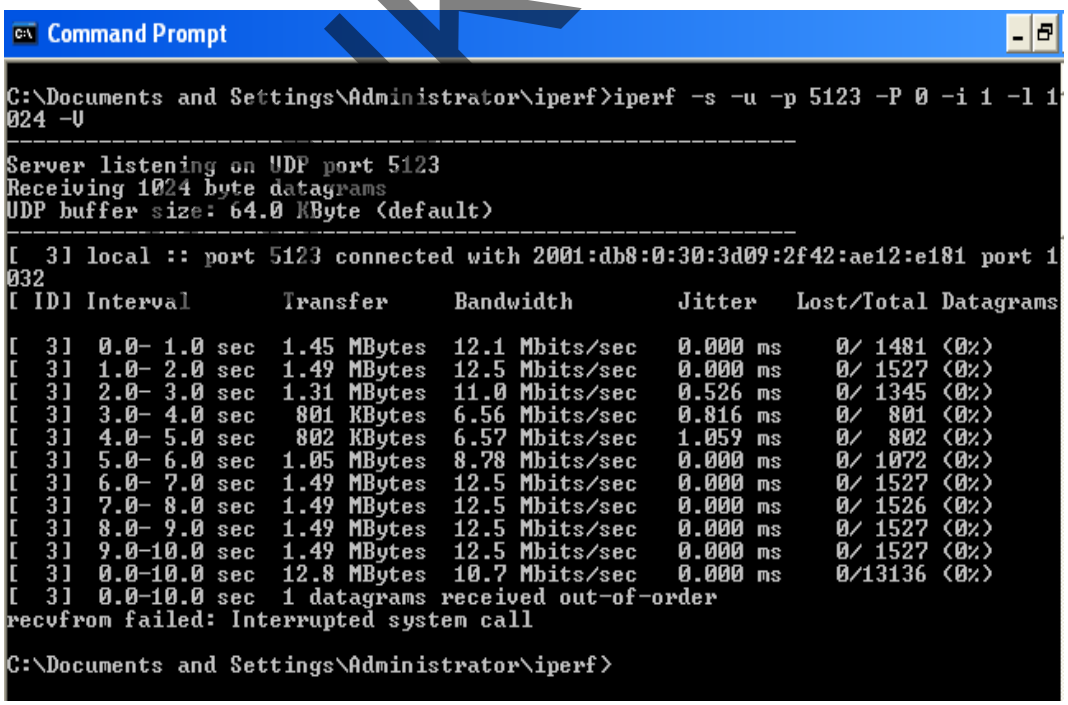
D.2. Tes Iperf dengan Beban Paket 256 Byte



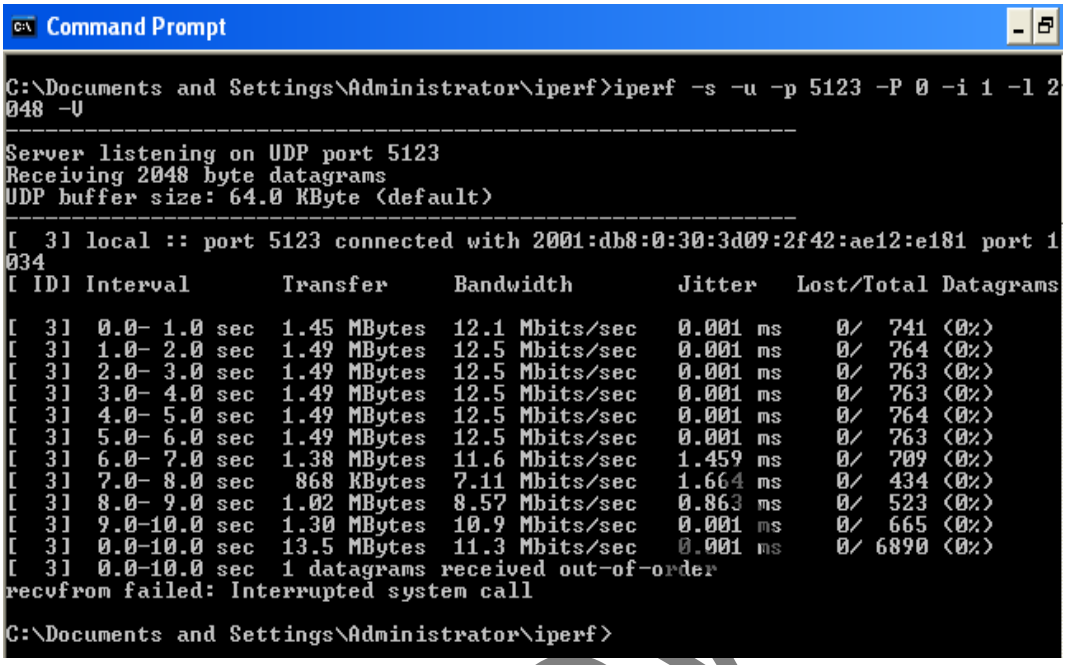
D.3. Tes Iperf dengan Beban Paket 512 Byte



D.4. Tes Iperf dengan Beban Paket 1024 Byte

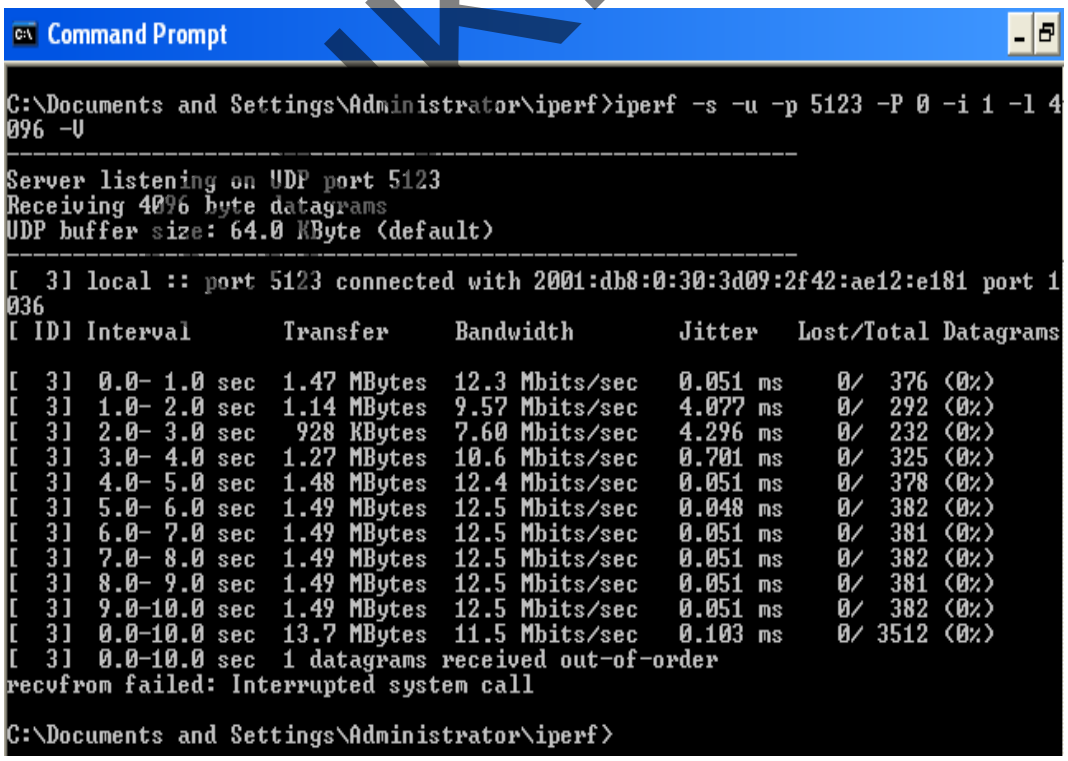


D.5. Tes Iperf dengan Beban Paket 2048 Byte



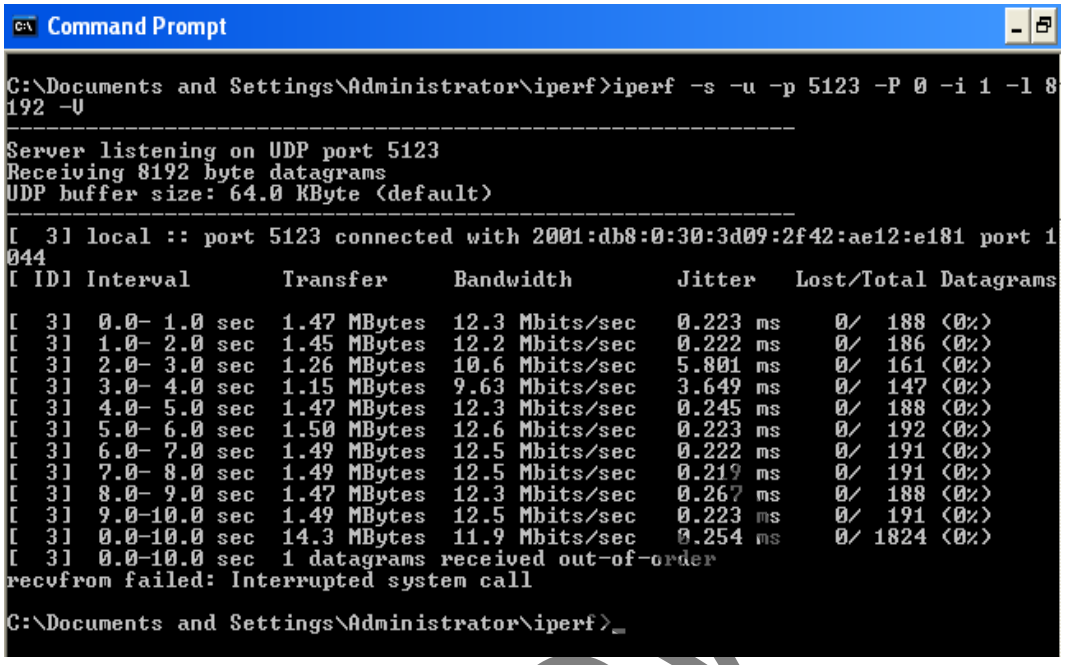
```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 2
048 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 2048 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:30:3d09:2f42:ae12:e181 port 1
034
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.45 MBytes  12.1 Mbits/sec   0.001 ms    0/  741 (0%)
[  3]  1.0- 2.0 sec  1.49 MBytes  12.5 Mbits/sec   0.001 ms    0/  764 (0%)
[  3]  2.0- 3.0 sec  1.49 MBytes  12.5 Mbits/sec   0.001 ms    0/  763 (0%)
[  3]  3.0- 4.0 sec  1.49 MBytes  12.5 Mbits/sec   0.001 ms    0/  763 (0%)
[  3]  4.0- 5.0 sec  1.49 MBytes  12.5 Mbits/sec   0.001 ms    0/  764 (0%)
[  3]  5.0- 6.0 sec  1.49 MBytes  12.5 Mbits/sec   0.001 ms    0/  763 (0%)
[  3]  6.0- 7.0 sec  1.38 MBytes  11.6 Mbits/sec   1.459 ms    0/  709 (0%)
[  3]  7.0- 8.0 sec   868 KBytes  7.11 Mbits/sec   1.664 ms    0/  434 (0%)
[  3]  8.0- 9.0 sec  1.02 MBytes  8.57 Mbits/sec   0.863 ms    0/  523 (0%)
[  3]  9.0-10.0 sec  1.30 MBytes  10.9 Mbits/sec   0.001 ms    0/  665 (0%)
[  3]  0.0-10.0 sec  13.5 MBytes  11.3 Mbits/sec   0.001 ms    0/ 6890 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

D.6. Tes Iperf dengan Beban Paket 4096 Byte

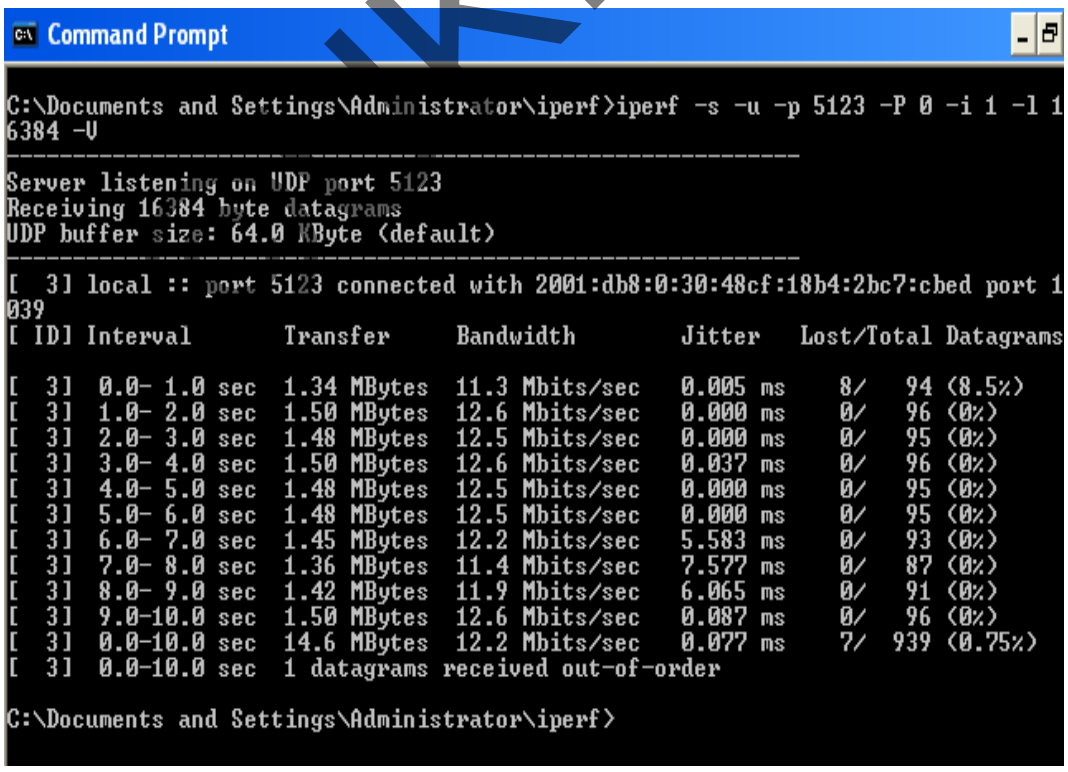


```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 4
096 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 4096 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:30:3d09:2f42:ae12:e181 port 1
036
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0- 1.0 sec  1.47 MBytes  12.3 Mbits/sec   0.051 ms    0/  376 (0%)
[  3]  1.0- 2.0 sec  1.14 MBytes  9.57 Mbits/sec   4.077 ms    0/  292 (0%)
[  3]  2.0- 3.0 sec   928 KBytes  7.60 Mbits/sec   4.296 ms    0/  232 (0%)
[  3]  3.0- 4.0 sec  1.27 MBytes  10.6 Mbits/sec   0.701 ms    0/  325 (0%)
[  3]  4.0- 5.0 sec  1.48 MBytes  12.4 Mbits/sec   0.051 ms    0/  378 (0%)
[  3]  5.0- 6.0 sec  1.49 MBytes  12.5 Mbits/sec   0.048 ms    0/  382 (0%)
[  3]  6.0- 7.0 sec  1.49 MBytes  12.5 Mbits/sec   0.051 ms    0/  381 (0%)
[  3]  7.0- 8.0 sec  1.49 MBytes  12.5 Mbits/sec   0.051 ms    0/  382 (0%)
[  3]  8.0- 9.0 sec  1.49 MBytes  12.5 Mbits/sec   0.051 ms    0/  381 (0%)
[  3]  9.0-10.0 sec  1.49 MBytes  12.5 Mbits/sec   0.051 ms    0/  382 (0%)
[  3]  0.0-10.0 sec  13.7 MBytes  11.5 Mbits/sec   0.103 ms    0/ 3512 (0%)
[  3]  0.0-10.0 sec  1 datagrams received out-of-order
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

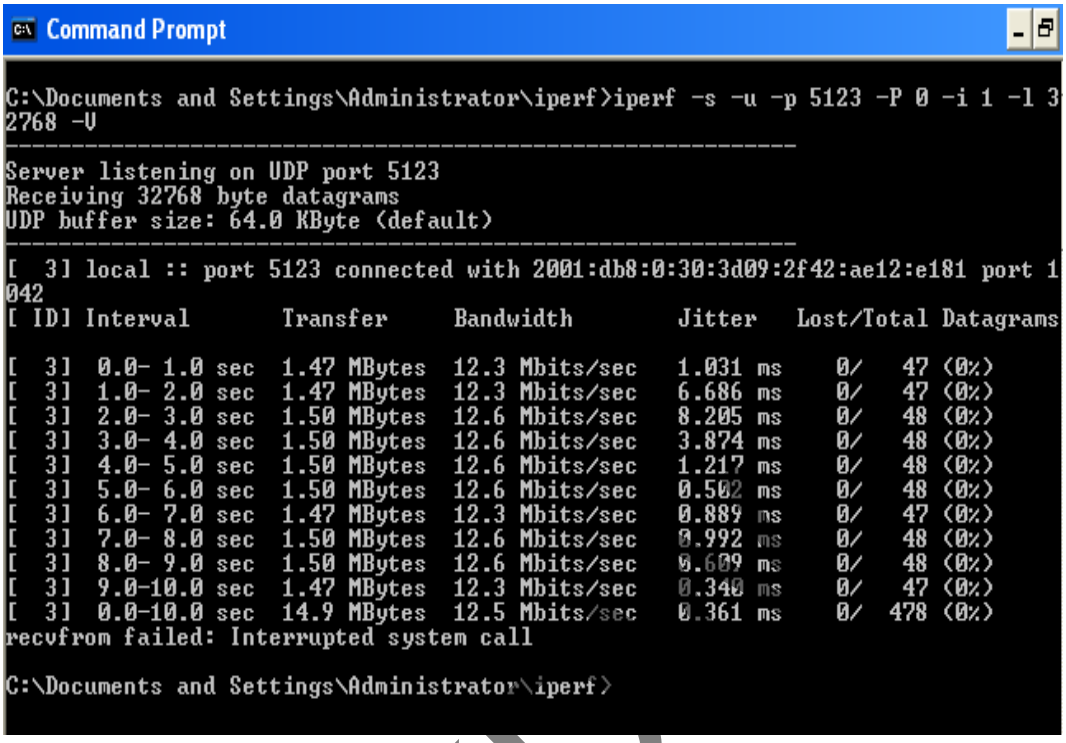
D.7. Tes Iperf dengan Beban Paket 8192 Byte



D.8. Tes Iperf dengan Beban Paket 16384 Byte

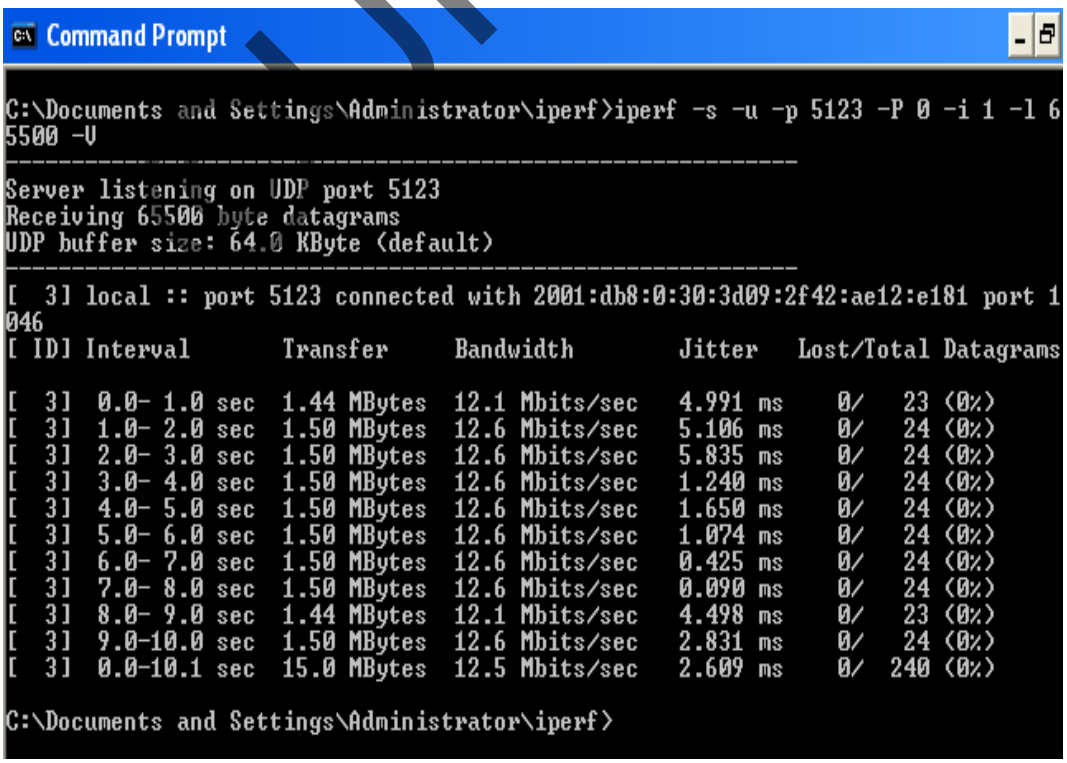


D.9. Tes Iperf dengan Beban Paket 32768 Byte



```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 3
2768 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 32768 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:30:3d09:2f42:ae12:e181 port 1
042
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams

[  3]  0.0- 1.0 sec  1.47 MBytes  12.3 Mbits/sec   1.031 ms    0/   47 (0%)
[  3]  1.0- 2.0 sec  1.47 MBytes  12.3 Mbits/sec   6.686 ms    0/   47 (0%)
[  3]  2.0- 3.0 sec  1.50 MBytes  12.6 Mbits/sec   8.205 ms    0/   48 (0%)
[  3]  3.0- 4.0 sec  1.50 MBytes  12.6 Mbits/sec   3.874 ms    0/   48 (0%)
[  3]  4.0- 5.0 sec  1.50 MBytes  12.6 Mbits/sec   1.217 ms    0/   48 (0%)
[  3]  5.0- 6.0 sec  1.50 MBytes  12.6 Mbits/sec   0.502 ms    0/   48 (0%)
[  3]  6.0- 7.0 sec  1.47 MBytes  12.3 Mbits/sec   0.889 ms    0/   47 (0%)
[  3]  7.0- 8.0 sec  1.50 MBytes  12.6 Mbits/sec   0.992 ms    0/   48 (0%)
[  3]  8.0- 9.0 sec  1.50 MBytes  12.6 Mbits/sec   0.609 ms    0/   48 (0%)
[  3]  9.0-10.0 sec  1.47 MBytes  12.3 Mbits/sec   0.340 ms    0/   47 (0%)
[  3]  0.0-10.0 sec  14.9 MBytes  12.5 Mbits/sec   0.361 ms    0/  478 (0%)
recvfrom failed: Interrupted system call

C:\Documents and Settings\Administrator\iperf>
```

D.10. Tes Iperf dengan Beban Paket 65500 Byte

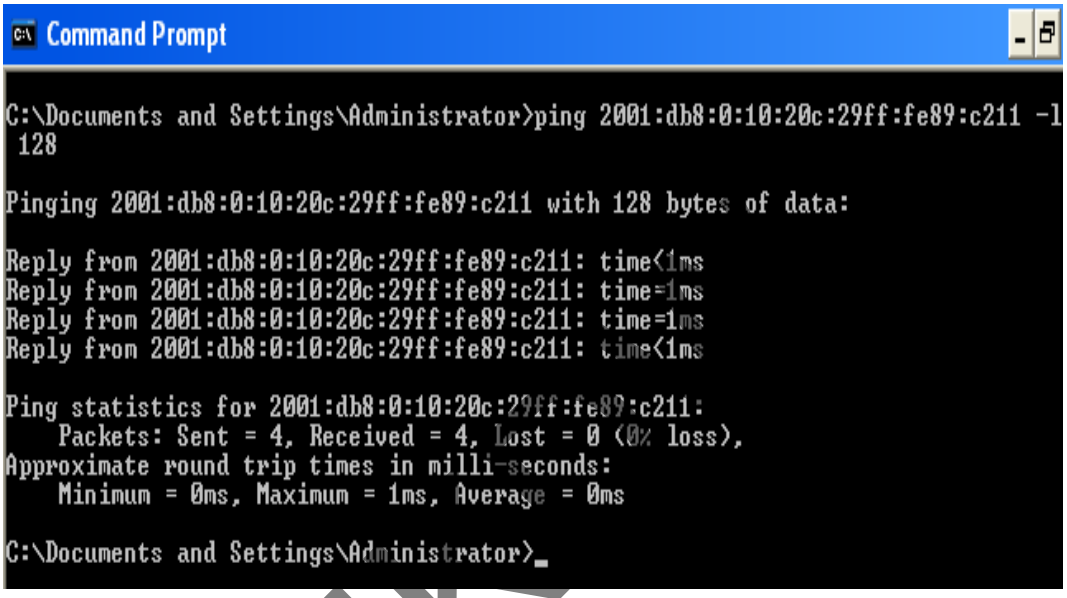


```
Command Prompt

C:\Documents and Settings\Administrator\iperf>iperf -s -u -p 5123 -P 0 -i 1 -l 6
5500 -V
------------------------------------------------------------
Server listening on UDP port 5123
Receiving 65500 byte datagrams
UDP buffer size: 64.0 KByte (default)
------------------------------------------------------------
[  3] local :: port 5123 connected with 2001:db8:0:30:3d09:2f42:ae12:e181 port 1
046
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams

[  3]  0.0- 1.0 sec  1.44 MBytes  12.1 Mbits/sec   4.991 ms    0/   23 (0%)
[  3]  1.0- 2.0 sec  1.50 MBytes  12.6 Mbits/sec   5.106 ms    0/   24 (0%)
[  3]  2.0- 3.0 sec  1.50 MBytes  12.6 Mbits/sec   5.835 ms    0/   24 (0%)
[  3]  3.0- 4.0 sec  1.50 MBytes  12.6 Mbits/sec   1.240 ms    0/   24 (0%)
[  3]  4.0- 5.0 sec  1.50 MBytes  12.6 Mbits/sec   1.650 ms    0/   24 (0%)
[  3]  5.0- 6.0 sec  1.50 MBytes  12.6 Mbits/sec   1.074 ms    0/   24 (0%)
[  3]  6.0- 7.0 sec  1.50 MBytes  12.6 Mbits/sec   0.425 ms    0/   24 (0%)
[  3]  7.0- 8.0 sec  1.50 MBytes  12.6 Mbits/sec   0.090 ms    0/   24 (0%)
[  3]  8.0- 9.0 sec  1.44 MBytes  12.1 Mbits/sec   4.498 ms    0/   23 (0%)
[  3]  9.0-10.0 sec  1.50 MBytes  12.6 Mbits/sec   2.831 ms    0/   24 (0%)
[  3]  0.0-10.1 sec  15.0 MBytes  12.5 Mbits/sec   2.609 ms    0/  240 (0%)

C:\Documents and Settings\Administrator\iperf>
```

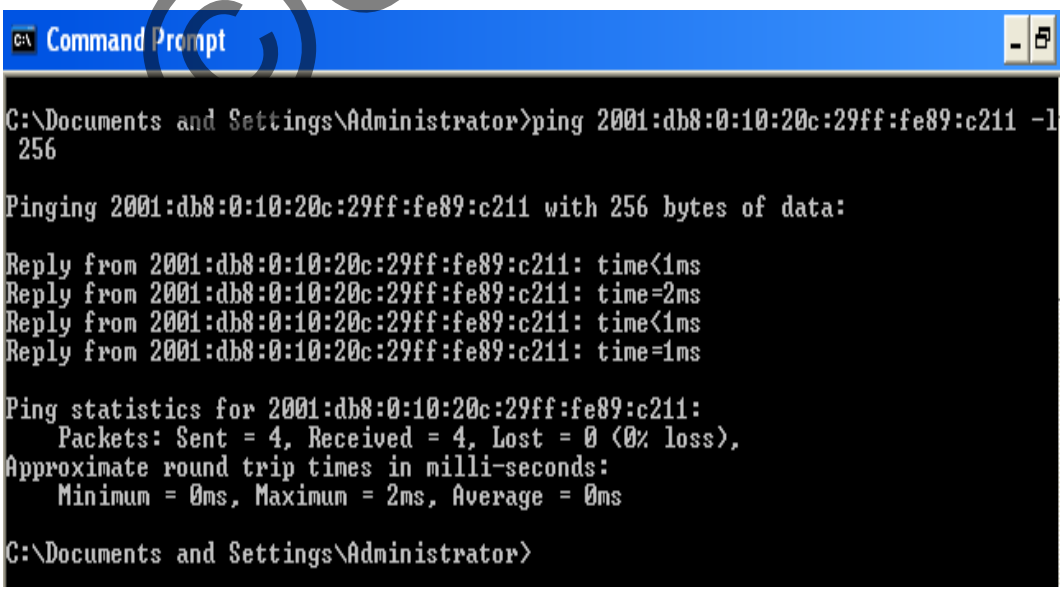
LAMPIRAN E

Hasil Tes RTT pada Teknik Transisi ISATAP

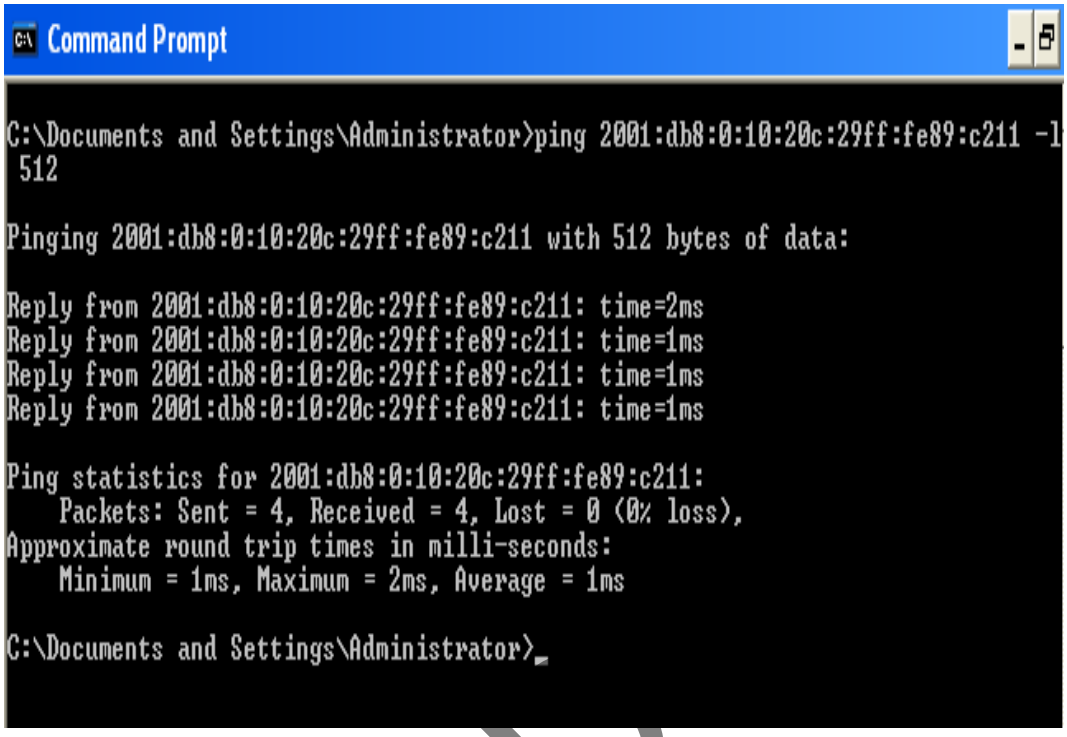
E.1. Tes RTT dengan Beban Paket 128 Byte



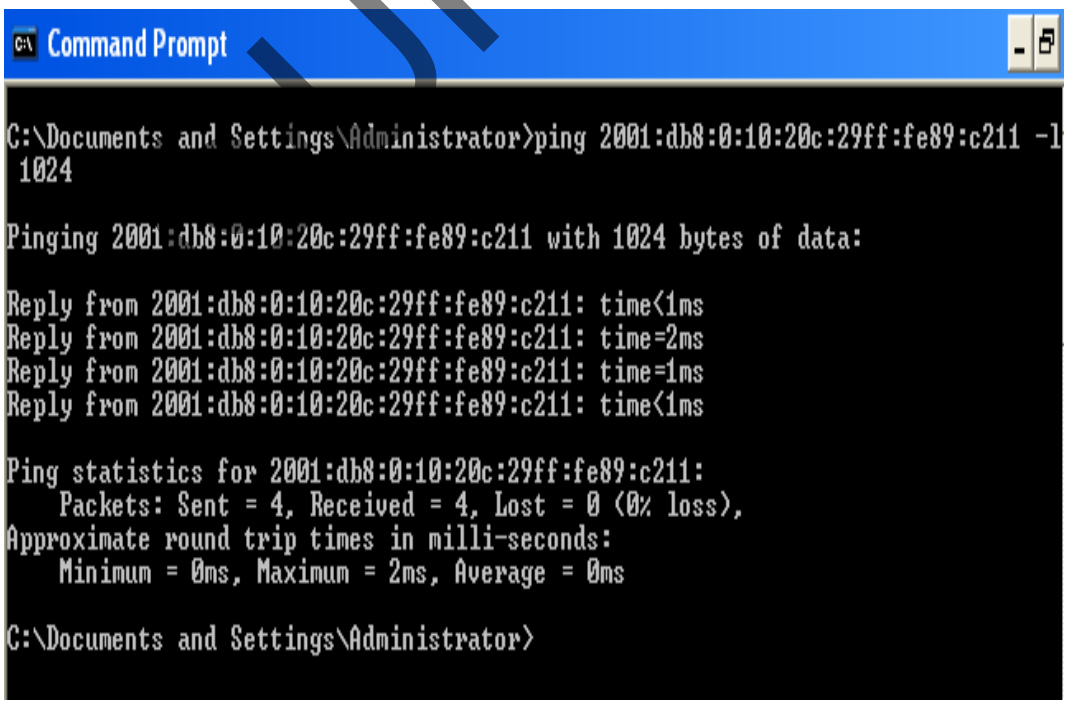
E.2. Tes RTT dengan Beban Paket 256 Byte



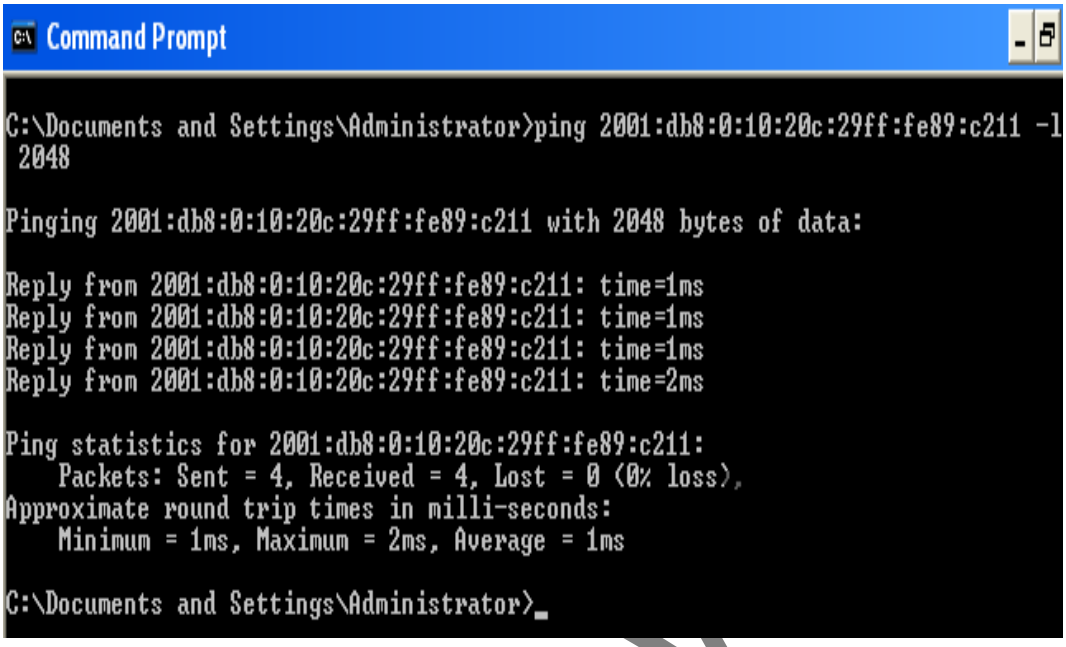
E.3. Tes RTT dengan Beban Paket 512 Byte



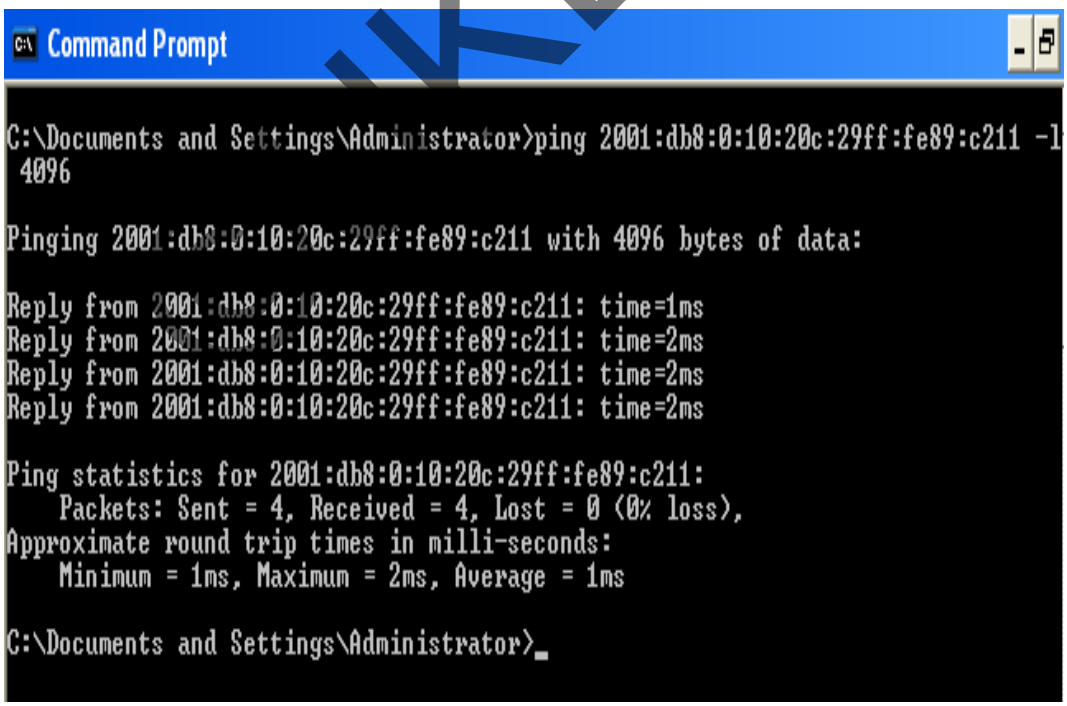
E.4. Tes RTT dengan Beban Paket 1024 Byte



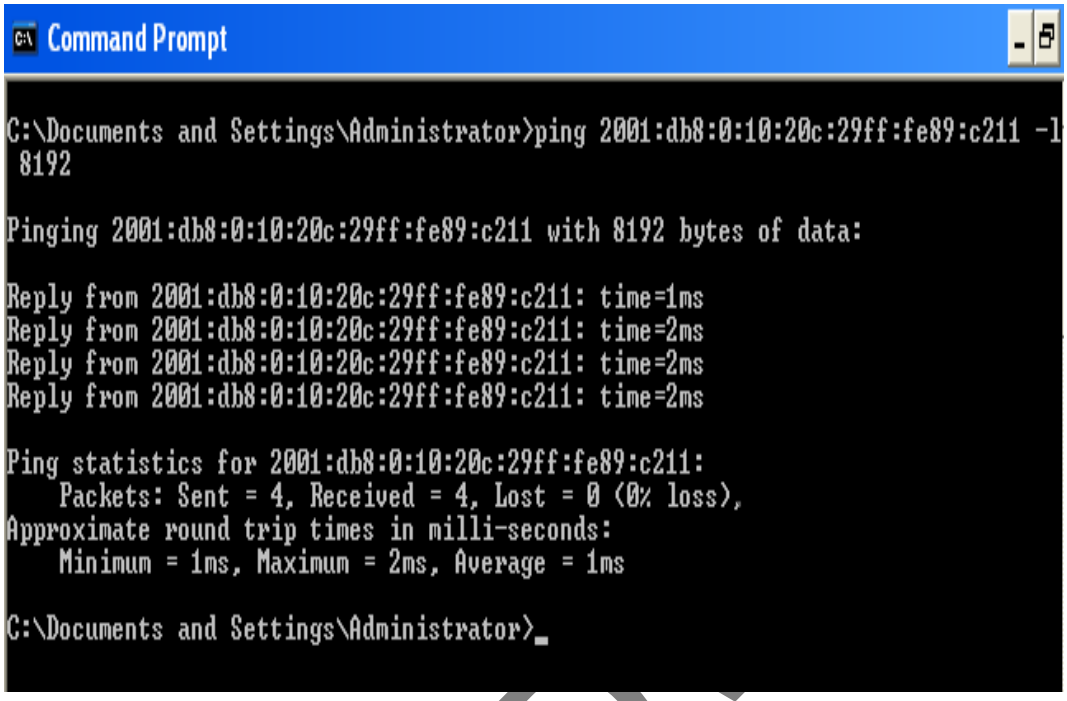
E.5. Tes RTT dengan Beban Paket 2048 Byte



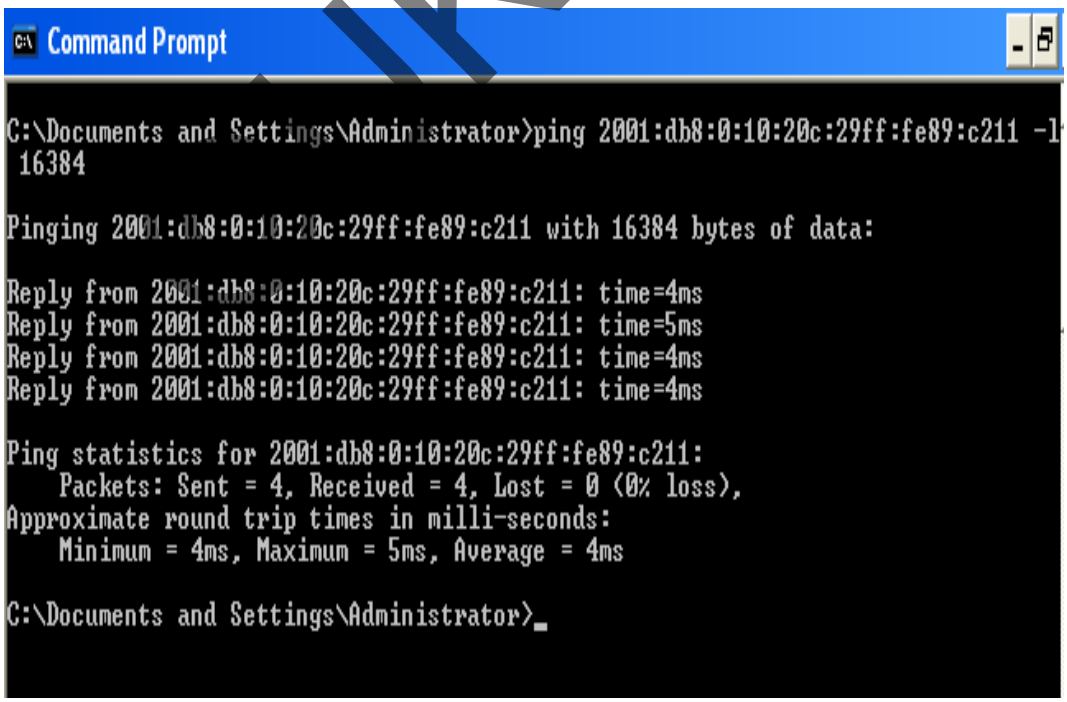
E.6. Tes RTT dengan Beban Paket 4096 Byte



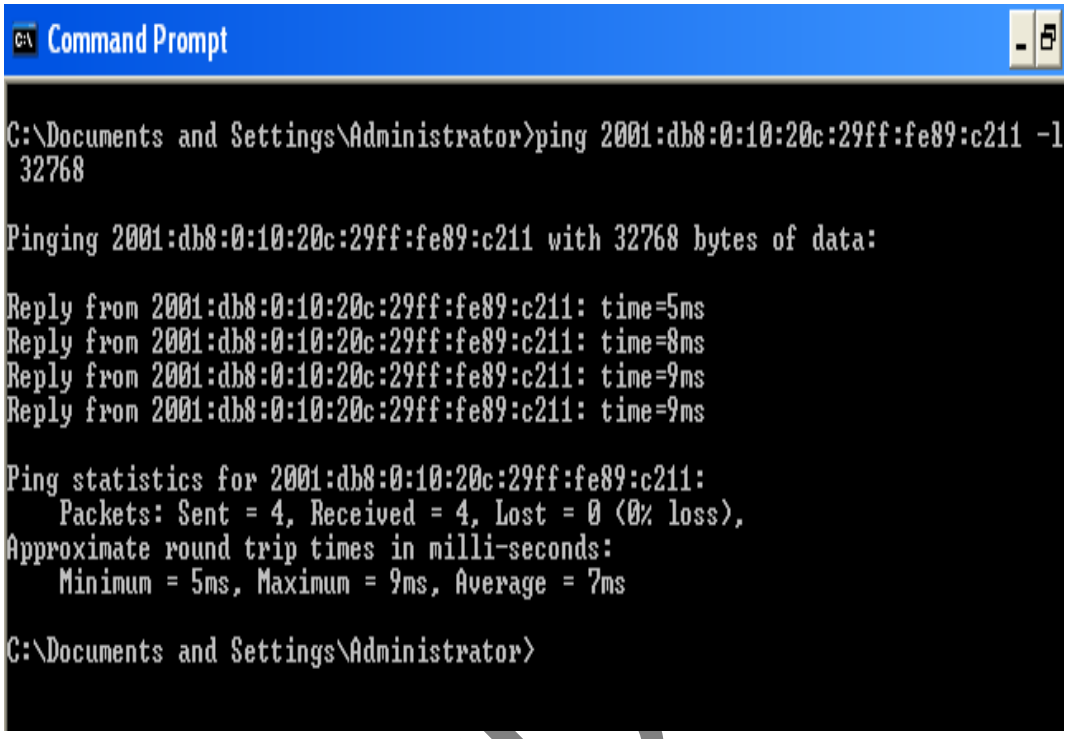
E.7. Tes RTT dengan Beban Paket 8192 Byte



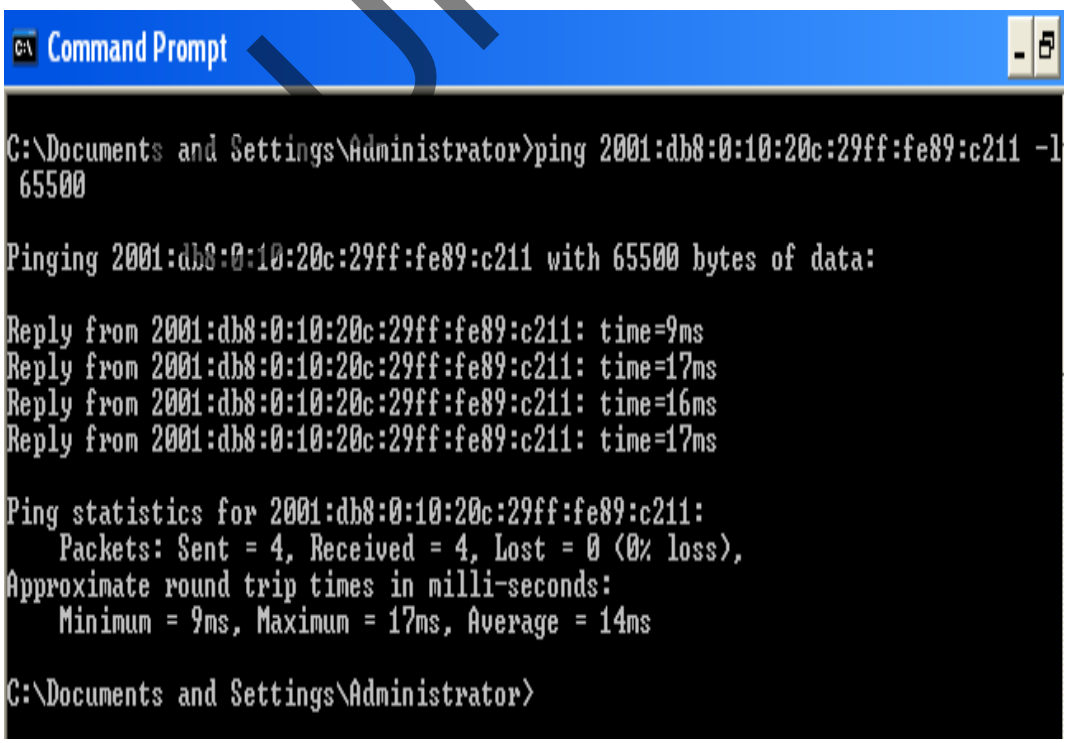
E.8. Tes RTT dengan Beban Paket 16384 Byte



E.9. Tes RTT dengan Beban Paket 32768 Byte



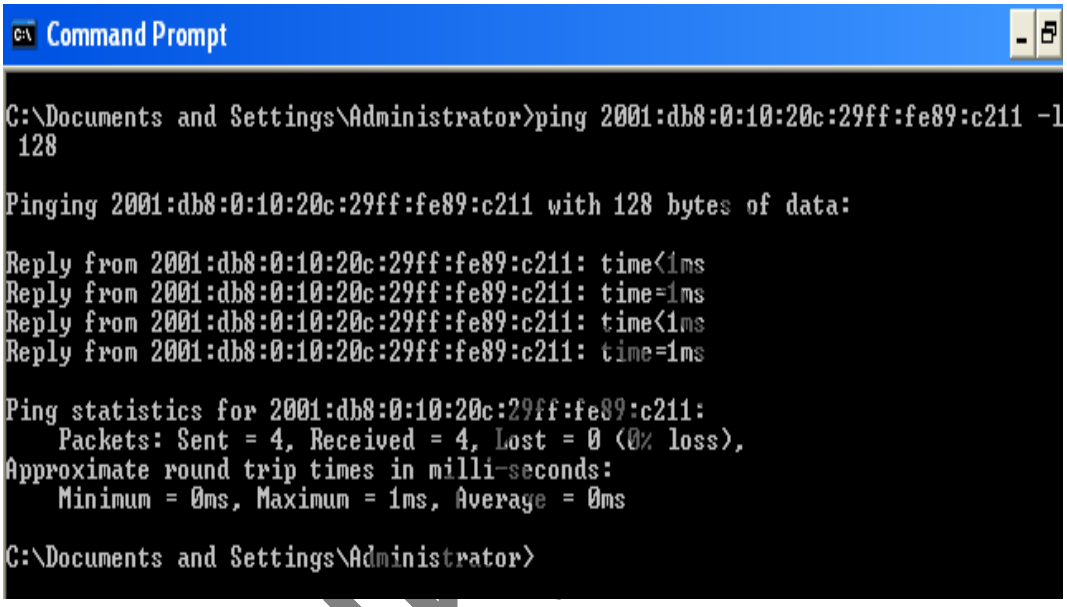
E.10. Tes RTT dengan Beban Paket 65500 Byte



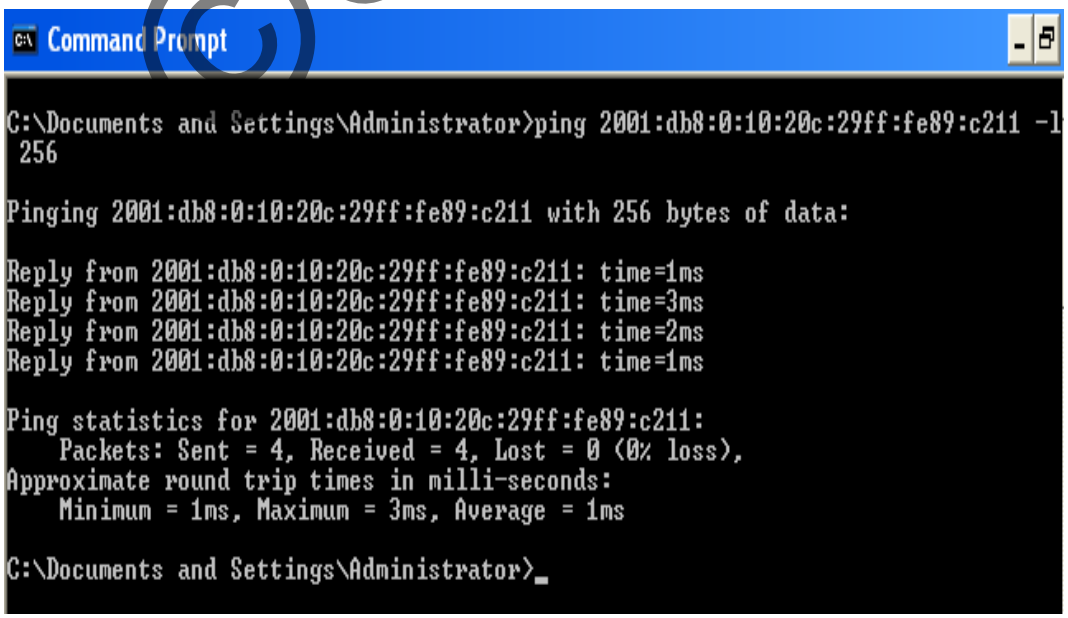
LAMPIRAN F

Hasil Tes RTT pada Teknik Transisi Dual Stack

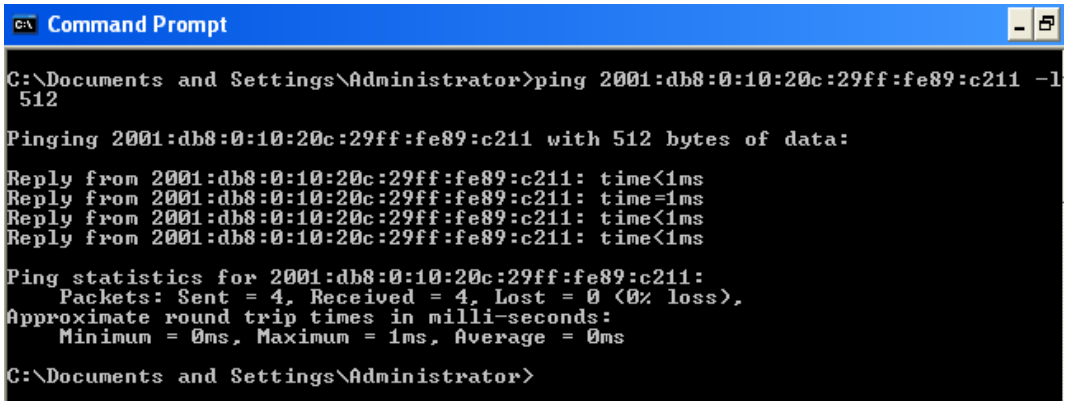
F.1. Tes RTT dengan Beban Paket 128 Byte



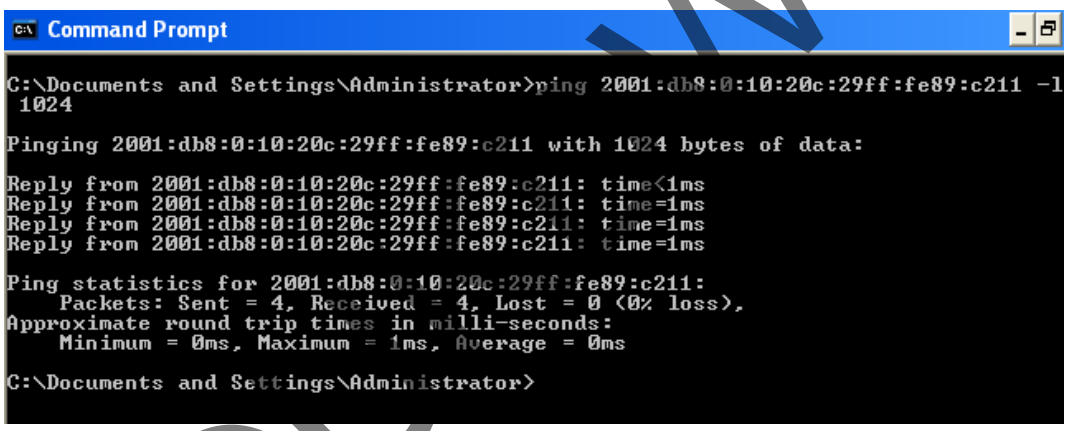
F.2. Tes RTT dengan Beban Paket 256 Byte



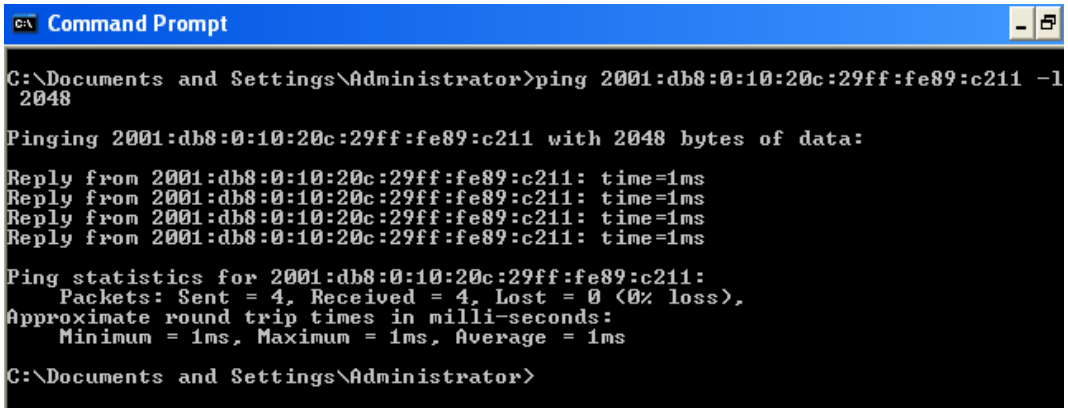
F.3. Tes RTT dengan Beban Paket 512 Byte



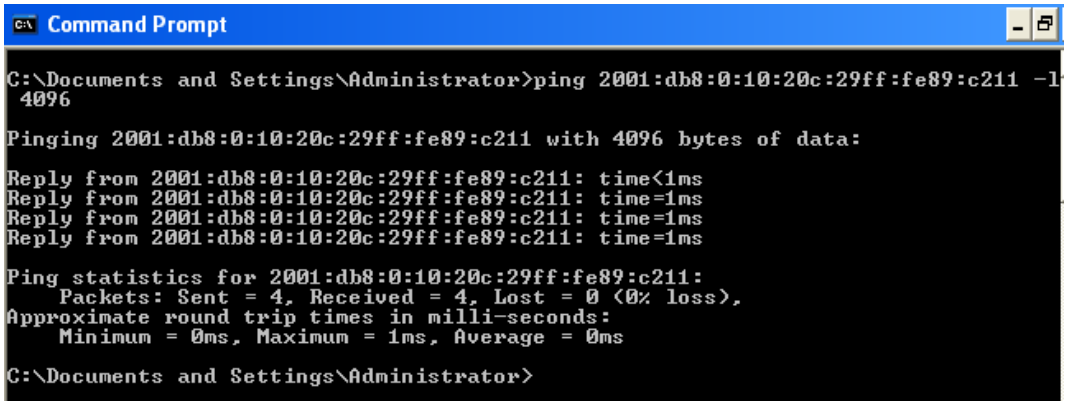
F.4. Tes RTT dengan Beban Paket 1024 Byte



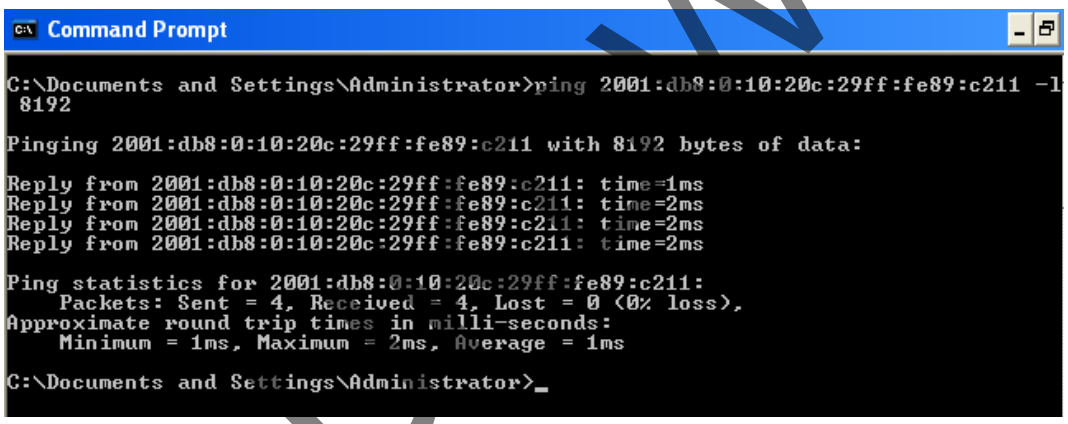
F.5. Tes RTT dengan Beban Paket 2048 Byte



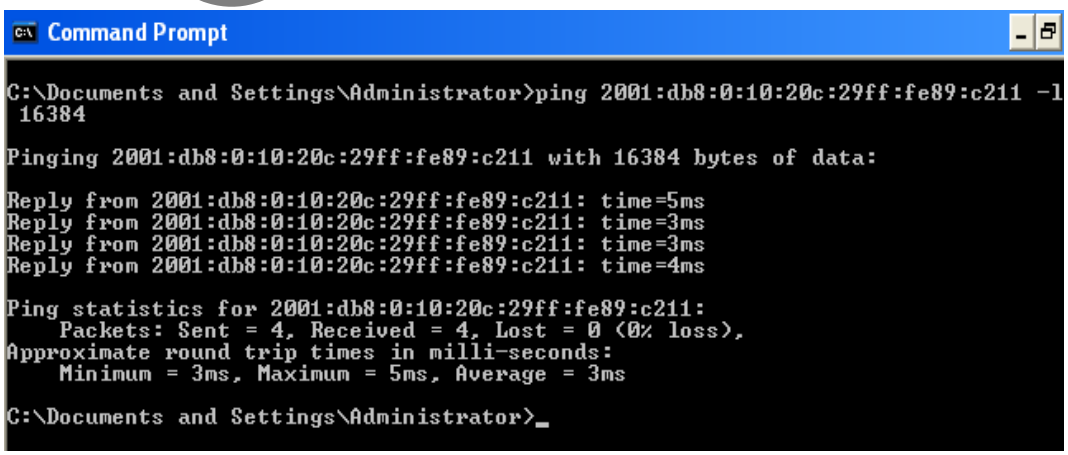
F.6. Tes RTT dengan Beban Paket 4096 Byte



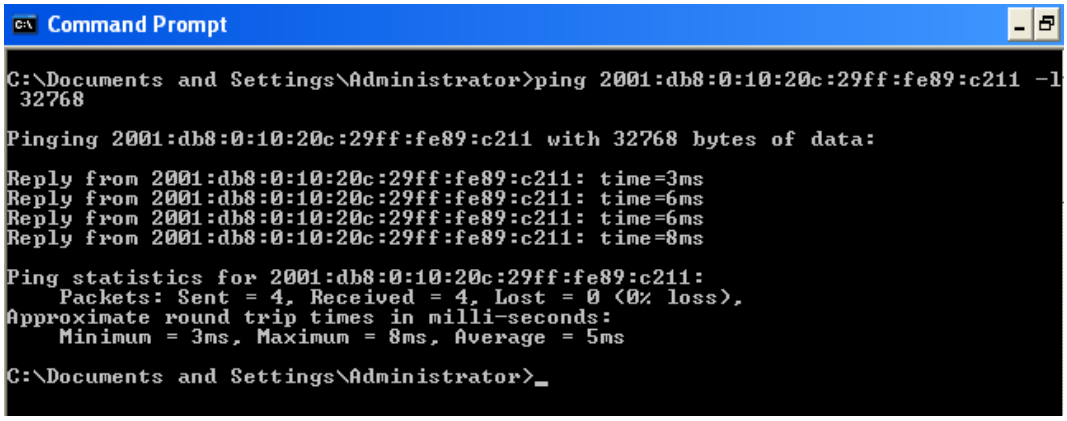
F.7. Tes RTT dengan Beban Paket 8192 Byte



F.8. Tes RTT dengan Beban Paket 16384 Byte



F.9. Tes RTT dengan Beban Paket 32768 Byte



F.10. Tes RTT dengan Beban Paket 65500 Byte