Melek IT

Program Studi
Teknik Informatika
Universitas Wijaya Kusuma Surabaya

Jurnal Teknologi Infromasi Dan Komunikasi

Volume 1 No 2 Juli 2012

ISSN 2252-9128

Melek IT
Program Studi Teknik Informatika Universitas Wijaya Kusuma Surabaya
Jurnal Teknologi Informasi Dan Komunikasi
Volume 3 No 2 Juli 2014
ISSN : 2252-9128

Daftar Isi

(4)

ANALISIS KOMUNIKASI DATA PADA KOMPUTER IPv4 DAN IPv6 MENGGUNAKAN DUAL STACK TRANSITION MECHANISM (DSTM)

Adi Bagus Santoso[1], F.X. Wisnu Yudo Untoro[2]
Program Studi Teknik Informatika, Fakultas Teknik, Univesitas Wijaya Kusuma Surabaya
Jl. Dukuh Kupang XXV/ 54Surabaya 60225
Email: b_guez44@yahoo.com

ABSTRAK

Perkembangan teknologi jaringan komputer saat ini semakin berkembang pesat seiring semakin meningkatnya pengguna yang memanfaatkan jaringan komputer. Dalam suatu jaringan komputer terdapat sistem pengalamatan yang dinamai dengan *IP address*.

IP address yang umum digunakan saat ini adalah IPv4. Pertambahan *user* yang semakin banyak menyebabkan habisnya IPv4 yang tersedia di internet. Untuk mengatasi permasalahan tersebut dikembangkan jenis IPv6. Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, diperlukan suatu sistem *Dual Stack Transition Mechanism* (DSTM) untuk mengkoneksikan kedua IP tersebut dalam sebuah jaringan komputer.

Pembuatan aplikasi ini menggunakan *Dual Stack Transition Mechanism* (DSTM) yang nantinya akan diuji melalui parameter *throughput* dan *delay* guna mengetahui gambaran antara IPv4 dan IPv6 dimasa depan.

Kata Kunci : Dual Stack Transition Mechanism (DSTM), IPv4, IPv6

PENDAHULUAN

Latar Belakang

Perkembangan teknologi saat ini sudah sangat modern, terutama perkembangan teknologi jaringan komputer semakin pesat seiring dengan kebutuhan *user* yang memanfaatkan layanan ini. Pada sistem jaringan komputer terdapat IP (*Internet Protocol*). IP yang dikenal secara umum saat ini adalah IP versi 4 (IPv4) yang ketersediaanya semakin berkurang, sehingga dikembangkan IP versi 6 (IPv6).

IP versi 6 (IPv6) merupakan *protocol* internet baru yang dikembangkan pada tahun 1994 oleh *Internet Engineering Task Force* (IETF) untuk menggantikan IP versi 4 (IPv4) yang saat ini tengah mendekati ambang batas alokasi alamatnya. Ruang alamat IPv4 ini diperkiraakan akan habis pada tahun 2011. Tujuan utama dikembangkan IPv6 adalah untuk meningkatkan ruang alamat internet sehingga mampu mengakomodasi perkembangan internet yang semakin pesat.

Sejak pelepasan IPv4dirancang untuk mendukung 2^{32} IP *address*, Di lain pihak IPv6, yang mampu mendukung 2^{128} IP *address*dapat menampung 2^{96} kali jumlah alamat yang dapat disediakan oleh IPv4. Tetapi, itu bukanlah satusatunya keuntungan dari IPv6. IPv6 juga memiliki banyak kelebihan lain, *addressing* dan *routing* yang lebih baik, *header* yanglebih sederhana, dukungan untuk kualitas dan *classes of service* ditingkat paket, keamanan yang lebih baik dalam hal autentifikasi, integritas pesan dan privasi, berkurangnya jumlah administrasi karena auto konfigurasi yang lebih baik dan peningkatan dukungan terhadap mobilitas.

Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, oleh karena itu menjadi permasalahan bagaimana jaringan IPv4 mampu berinteraksi dengan jaringan IPv6. Salah satu cara yaitu dengan dilakukan migrasi IPv4 ke IPv6 atau bisa menggunakan mekanisme lain. Walaupun terlihat sederhana, migrasi ini tidak mudah dilakukan, karena bila hal ini dilakukan maka semua bagian dari jaringan, termasuk diantaranya *firewall, server*, dan *workstation* yang digunakan juga harus di-*up grade*. Dalam mekanisme transisi yang memungkinkan keduanya bisa berhubungan. Mekanisme itu misalnya dengan menggunakan *dual protocol stack*, translasi, *tunneling*, atau menggantinya menjadi IPv6 sepenuhnya.

Dalam tugas akhir ini penulis menggunakan teknik *DualStack Transition Mechanism (DSTM)* merupakan solusi yang ditujukan untuk jaringan untuk dilakukan teknik tersebut maka akan di analisa dari jaringan IPv4 dan IPv6 untuk

mengetahui perbandingan dari kedua jaringan tersebut dengan parameter *throughput dan delay*.

TINJAUAN PUSTAKA

*Internet Protocolversion*4 (IPv4)

IPv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-*bit,* dan secara reoritis dapat mengalamati hingga 232 *host* komputer di dunia. Alamat IPv4 umumnya diekspresikan dalam notasi desimal bertitik (*dotted-decimal notation*), yang dibagi kedalam empat buat oktet berukuran 8-*bit*sehingga nilainya berkisar antara 0 hingga 255.

IPv4 *Addressing*

Alamat IP yang dimiliki oleh sebuah *host*dapat dibagi dengan menggunakan *subnet mask* jaringan kedalam dua buah bagian, yakni: *1. Network Identifier*/NetID atau *network address* (alamat jaringan)yang digunakan khusus untuk mengidentifikasikan alamat jaringan di mana *host* berada. Dalam banyak kasus, sebuah alamat *network identifier* adalah sama dengan segmen jaringan fisik dengan batasan yang dibuat dan didefinisikan oleh *router* IP. Meskipun demikian, ada beberapa kasus di mana beberapa jaringan logis terdapat di dalam sebuah segmen jaringan fisik yang sama dengan menggunakan sebuah praktek yang disebut sebagai *multinetting*. Semua sistem di dalam sebuah jaringan fisik yang sama harus memiliki alamat *network identifier* yang sama. *Network identifier* juga harus bersifat unik dalam sebuah *internetwork*. Jika semua node di dalam jaringan logis yang sama tidak dikonfigurasikan dengan menggunakan *network identifier* yang sama, maka terjadilah masalah yang disebut dengan *routing error*. Alamat *network identifier* tidak boleh bernilai 0 atau 255. 2. *Host Identifier*/*Host* ID atau *Host address* (alamat *host*) yang digunakan khusus untuk mengidentifikasikan alamat *host* (dapat berupa workstation, *server* atau sistem lainnya yang berbasis teknologi TCP/IP) di dalam jaringan. Nilai *host identifier* tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam *network identifier*/segmen jaringan di mana ia berada.

Alamat IPv4 terbagi menjadi beberapa jenis, yakni sebagai berikut:

1. *Alamat Unicast*, merupakan alamat IPv4 yang ditentukan untuk sebuah antarmuka jaringan yang dihubungkan ke sebuah *internetwork*IP. Alamat *unicast*digunakan dalam komunikasi *point-to-point* atau *one-to-one*.
2. *Alamat Broadcast*, merupakan alamat IPv4 yang didesain agar diproses oleh setiap *node* IP dalam segmen jaringan yang sama. Alamat *broadcast* digunakan dalam komunikasi *oneto-everyone.*
3. *Alamat Multicast*, merupakan alamat IPv4 yang didesain agar diproses oleh satu atau beberapa node dalam segmen jaringan yang sama atau berbeda. Alamat *multicast* digunakan dalam komunikasi *one-to-many*.

Alamat IP versi 4 dibagi ke dalam beberapa kelas, dilihat dari oktet pertamanya, seperti terlihat pada tabel 2.1. Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama
(utamanya adalah *bit-bit*awal/*high-order bit*), tapi untuk lebih mudah mengingatnya, akan lebih cepatdiingat dengan menggunakan representasi desimal.

Kelas Alamat IP	Oktet Pertama (desimal)	Oktet Pertama (biner)	Digunakan Oleh
Kelas A	1 – 126	0xxx xxxx	Alamat *unicast* untuk jaringan skala besar
Kelas B	128 – 191	1xxx xxxx	Alamat *unicast* untuk jaringan skala menengah hingga skala besar
Kelas C	192 – 223	110x xxxx	Alamat *unicast* untuk jaringan skala kecil
Kelas D	224 – 239	1110 xxxx	Alamat *multicast* (bukan alamat *unicast*)
Kelas E	240 – 255	1111 xxxx	Diresevasikan, umumnya digunakan sebagai alamat percobaan (eksperimen); (bukan alamat unicast)

Tabel 2.1 Pembagian Kelas Dalam IPv4

1. Kelas A

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

2. Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua *bit*pertama di dalam octet pertama alamat IP kelas B selalu diset ke bilangan biner 10. 14 *bit*berikutnya (untuk melengkapi dua oktet pertama), akan

membuat sebuah *network identifier*. 16 *bit*sisanya (dua octet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki hingga 16,384 jaringan, dan 65,534 *host*untuk setiap jaringannya.

3. Kelas C

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

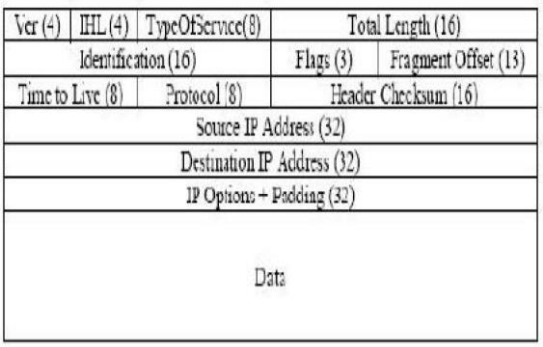
4. Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas.Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28 *bit* sisanya digunakan sebagai alamat yang dapatdigunakan untuk mengenali *host*.

5. Kelas E

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

Struktur *Header*Paket IPv4 paket-paket data dalam protokol IPv4 dikirimkan dalam bentuk datagram. Sebuah paket IPv4 terdiri atas *header* IP dan muatan IP (*payload*). *Header* IP menyediakan dukungan untuk memetakan jaringan (*routing*), identifikasi muatan IP, ukuran *header*IP dan paket IP, dukungan fragmentasi, dan juga IP *options*. Sedangkan *payload* IP berisi informasi yang dikirimkan. *Payload* IP memeliki ukuran bervariasi, berkisar dari 8 *byte* hingga 65515 *byte*. Sebelum dikirimkan di dalam saluran jaringan, paket IP akan dibungkus (*encapsulation*) dengan *header protocol* lapisan antarmuka jaringan dan*trailer*-nya, untuk membuat sebuah frame jaringan. Setiap paket terdiri dari beberapa field yang memiliki fungsi tersendiri dan memeliki informasi yang berbeda-beda. Pada gambar 2.10 dapat dilihat struktur dari paket IPv4.



Gambar 2.2 Struktur Paket IPv4

***Internet Protocol version*6 (IPv6)**

Sistem pengalamatan IPv6 disebut juga dengan IPng (*Internet Protocol, nextgeneration*) karena merupakan *generasiter*baru pengganti IPv4 sebagai standar IP. IPv6 menggunakan sistem pengalamatan 128 *bits*, 4 kali lebih besar daripada IPv4 yang artinya mampu menghasilkan alokasi alamat sebesar 2(128-32) kali lebih besar daripada IPv4. Sistem pengalamatan ini dipetakan secara heksa (16 *bits*) untuk mempermudah pembacaannya. Setiap 16 *bits* tersebut ditampilkan dalam bentuk *section* secara heksadesimal 4 digit dengan dipisahkan olehtanda titik dua. Walaupun ditampilkan secara heksadesimal, IPv6 dirasa terlalu rumit untuk diingat karena panjangnya mencapai 32 digit angka. Selain itu, IPv6 seringkali terdiri dari banyak angka nol sehingga dianggap kurang efisien. Pada kasus tersebut, IPv6 memiliki kelonggaran untuk memperpendek alamatnya dengan ketentuan sebagai berikut:

1. Angka nol yang mengawali setiap *section* dapat dihilangkan.
2. *Section* minimal memiliki satu digit angka.
3. *Section* yang berurutan dan hanya terdiri dari angka nol dapat diganti dengan tanda titik dua yang ditulis rangkap. Ketentuan ini hanya berlaku satu kali penulisan. Sebagai contoh penulisannya, dapat dilihat pada Gambar ini :

Gambar 2.3 Contoh penulisan alamat IPv6

Pada dasarnya, IPv6 terdiri dari 2 bagian utama yaitu prefiks yang menunjukkan tipe

pengalamatan dan sisanya mengikuti *system* yang digunakan prefiks tersebut. *Provider based unicast address* merupakan tipe prefix yang umum digunakan sebagai pengalamatan *unicast* pada *host* yang spesifik. Pengalamatan *unicast* memungkinkan suatu *host* berkomunikasi dengan satu *host*yang lain. *Provider-based unicast address* menggunakan prefiks 3 *bits* berupa "010" dengan diikuti sistem pengalamatannyasebagai berikut (Forouzan 2003):

1. *Registry identifier*, 5 *bits* penunjuk agensi pusat IPv6 yang telah mengalokasikan alamatnya. Sebagai contoh, untuk kawasan Asia-Pasifik dengan agensi pusat APNIC menggunakan kode 10100.
2. *Provider identifier*, menunjukkan ISP (*Internet Service Provider*) yang digunakan. Umumnya menggunakan 16 *bits*.
3. *Subscriber identifier*, menunjukkan kode berlangganan terhadap ISP tertentu. Umumnya menggunakan 24 *bits.*
4. *Subnet identifier*, menunjukkan *subnet* (sub jaringan) spesifik yang berada dibawah manajemen pengguna. Umumnya menggunakan 32 *bits*.
5. *Node identifier*, menunjukkan alamat spesifik suatu *host* di bawah *subnet* tertentu. Umumnya menggunakan 48 *bits*.

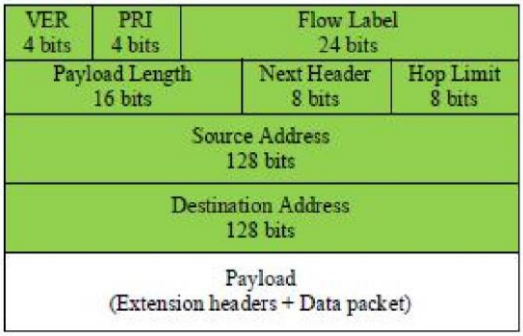
Format Prefix

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

Prefiks adalah sebuah bagian dari alamat IP, di mana *bit-bit* memiliki nilai-nilai yang tetap atau *bit-bit* tersebut merupakan bagian dari sebuah rute atau subnet identifier. Prefiks dalam IPv6 direpesentasikan dengan cara yang sama seperti halnya prefiks alamat IPv4, yaitu [alamat] /[angka panjang prefiks]. Panjang prefiks menentukan jumlah *bit* terbesar paling kiri yang membuat prefiks *subnet*. Sebagai contoh, prefiks sebuah alamat IPv6 dapat direpresentasikan yaitu : 3ffe:2900:d005:f28b::/64 Pada contoh tersebut, 64 *bit* pertama dari alamat tersebut dianggap sebagai prefiks alamat, sementara 64 *bit* sisanya dianggap sebagai interface ID.

Struktur *Header* IPv6

Datagram IPv6 terbagi menjadi dua bagian utama yaitu *header* dan *payload*. *Header* IPv6 memiliki ukuran yang tetap yakni 40 *bytes*. Akan tetapi, terdapat *header* tambahan(*extension*) untuk meningkatkan fungsionalitasnya di bagian *payload*. Dengan demikian, *payload* berisikan data paket beserta *header* tambahan tersebut.



Gambar 2.4 Struktur *header* IPv6.

Perbedaan IPv4 Dan IPv6

Perbedaan antara IPv4 dan IPv6 sebagai berikut.

IPv4

1. Kapasitas alamat 32 *bit*
2. IPv4 tidak mempunyai kemampuan auto renumbering
3. Bisa melakukan NAT (*Network Address* Translation).
4. Tidak semua produk dari IPv4 bisa mengimplementasikan IPSEC.
5. Alamat IPv4 terbagi menjadi 2 jenis yaitu *unicastaddress*, dan *multicastaddress*.
6. IP *header* IPv4 terdiri dari *version*. IHL, *Type* of *service*, Total *Length*, *Identification*, Flags, Fragment Offset, *Time* to Live, *Protocol*, *Header* checksum, Source *address*, Destination *address*, Option, Padding.
7. Alamat sumber (source *address*) dan alamat tujuan (*destinationaddress*) sebesar 32 *bit*.
8. Fragmetasi dikerjakan oleh both *router* dan *host* pengirim.
9. Checksum termasuk *Header* pada IPv4
10. Options termasuk *header* pada IPv4.
11. Menggunakan *hostaddress* (A) resource record didalam Domain Name *System* (DNS) untuk memetakan nama *host* pada pengalamatan IPv4.

IPv6

1. Kapasitas perluasan alamat 128 *bit*.
2. IPv6 mempunyai kemampuan autorenumbering (penomoran kembali alamat IP secara otomatis).
3. Tidak bisa melakukan NAT.

4. Setiap produk yang menggunakan IPv6 bisa mengimplementasikan IPSEC sehingga tidak perlu di upgrade. IPSEC adalah fitur yang dimiliki oleh IPv6 namun oleh beberapa *developer* diaplikasikan ke dalam IPv4. (Lihat IPSec).
5. Alamat IPv4 terbagi menjadi 3 yaitu *unicastaddress*, *multicastaddress*, dan anycast *address*.
6. IP *header* IPv6 terdiri dari *Version*, *Traffic* class, Flow Label, Payload *length*, *Next Header*, *Hop Limit*, *Source address*, *Destinationaddress*.
7. Alamat sumber (source *address*) dan alamat tujuan (*destination address*) sebesar 128 *bit*.
8. Fragmentasi tidak dikerjakan oleh *router*, hanya oleh *host* pengirim.
9. Checksum tidak termasuk *header* pada IPv6.
10. Semua optional data diusulkan untuk ekstensi *header* IPv6.
11. Menggunakan *hostaddress* (AAAA) resource record didalam Domain Name *system* (DNS) untuk memetakan nama *host* pada pengalamatan IPv6. AAAA adalah tipe *record* baru untuk menyimpan sebuah alamat IPv6 dengan nilai tipe 28.

Mekanisme Transisi

Mekanisme transisi secara umum didefinisikan sebagai sekumpulan teknik yang berupaya agar node IPv6 dapat saling berkomunikasi dengan node IPv4 yang sudah ada sebelumnya. Mekanisme ini terbagi menjadi empat kategori berdasarkan teknik yang digunakan, yaitu mekanisme *hybrid* (*dual* IPv4/IPv6), *aplication-layer gateways*, penerjemahan protokol, dan *tunneling*. Masingmasing kategori tersebut memiliki cara kerja dan tujuan yang berbeda-beda. *Tunneling* sangat dihandalkan sebagai mekanisme transisi pada saat IPv6 mulai dikembangkan.

Teknik yang digunakan yakni menghubungkan IPv4 dan IPv6 dengan cara enkapsulasi-dekapsulasi paket. Secara umum *tunneling* berupa IPv6-*over*-IPv4 yaitu membungkus paket IPv6 ke dalam paket IPv4 untuk kemudian dibuka kembali. Mekanisme ini sangat sesuai dalam kondisi jaringan yang didominasi IPv4 dan keberadaan node IPv6 yang menyebar tidak beraturan untuk saling berkomunikasi. Akan tetapi mekanisme ini kurang sesuai jika suatu jaringan didominasi IPv6. DSTM sebagai salah satu mekanisme *tunneling* terbaru menggunakan sistem yang berkebalikan yaitu IPv4-*over*-IPv6. jalur khusus paket IPv4. Paket IPv4 yang akan dikirim oleh DSTM *client*akan dibungkus dalam paket IPv6

***DualStack Transition Mechanism* (DSTM)**

DualStack Transition Mechanism (DSTM) merupakan salah satu mekanisme transisi *tunneling* (IPv4-*over*-IPv6) dengan membungkus paket IPv4 ke dalam bentuk paket IPv6 di sisi *host* IPv6 untuk kemudian dibuka kembali di batas akhir IPv6 ke IPv4dan dikirim menuju *host* dalam jaringan IPv4 (Bound 2002). Begitu pula sebaliknya untuk arah yang berlawanan. DSTM *client* (IPv6) dapat berkomunikasi dengan *host* IPv4 dengan cara meminta alamat IPv4 terlebih dahulu ke *server* DSTM. *Server* DSTM memberikan IPv4 secara dinamis kepada DSTM *client*yang kemudian dipetakan (*address mapping*) dengan alamat IPv6-nya dalam *cache* (penyimpanan sementara). Setelah mendapatkan IPv4 tersebut, DSTM *client*akan membangun *Dynamic Tunnel Interface* (DTI) sebagai menuju DSTM *Tunnel End Point* (DSTM TEP) untuk dibuka kembali dan dikirim sesuai tujuannya di jaringan IPv4. DSTM juga memungkinkan komunikasi sebaliknya antara *host* dalam jaringan IPv4 dengan jaringan IPv6. Dengan membaca *address mapping* dalam *cache* yang telah disediakan secara temporal sebelumnya, komunikasi data dapat segera berjalan. Namun jika *address mapping* tidak ada atau sudah hilang, maka perlu berhubungan dengan *Domain Name System* (DNS) untuk mengarahkan alamat sebenarnya dalam jaringan IPv6.

Delay

Delay merupakan lamanya waktu yang dibutuhkan oleh data informasi untuk sampai ke tempat tujuan data informasi tersebut dikirim. *Delay* pada suatu jaringan akan menentukan langkah apa yang akan kita ambil ketika kita memanjemen suatu jaringan. Ketika *delay* besar, dapat diketahui jaringan tersebut sedang sibuk atau kemungkian yang lain adalah kapasitas jaringan tersebut yang kecil sehingga bisa melekukan tindakan pencegahan agar tidak terjadi *overload*. Misalkan dengan memindahkan sebagian aliran data ke jalur lain atau memperbesar kapasitas jaringan kita, berikut tabel parameter kategori *delay*

Nilai Delay	Kategori
< 150 ms	Sangat Bagus
150 ms – 300 ms	Bagus
300 ms – 450 ms	Sedang
> 450 ms	Jelek

Tabel 2.2 Parameter Kategori *Delay*

Throughput

Throughput adalah ukuran dari *transferbit* di media selama jangka waktu tertentu. Karena sejumlah faktor, *throughput* biasanya tidak sesuai dengan bandwidth yang ditentukan dalam implementasi lapisan fisik seperti Ethernet. Banyak faktor yang mempengaruhi *throughput*. Diantara faktor-faktor tersebut jumlah lalu lintas, jenis lalu lintas, dan jumlah perangkat jaringan ditemui pada jaringan yang diukur. Dalam topologi multi-access seperti Ethernet, node bersaing untuk akses media dan penggunaannya. Oleh karena itu, *throughput* masing node terdegradasi penggunaan media meningkat.

Dalam inter*network* atau jaringan dengan beberapa segmen, *throughput* tidak bisa lebih cepat dari link paling lambat path dari sumber ke tujuan. Bahkan jika semua atau sebagian besar segmen memiliki bandwidth yang tinggi, itu hanya akan mengambil satu segmen dalam jalur dengan *throughput* yang rendah untuk menciptakan hambatan ke *throughput* seluruh jaringan

FileTransferProtocol

Protokol pengiriman berkas atau biasa di sebut *file transfer protocol* adalah sebuah protocol internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pengiriman berkas (*file*). komputer antarmesinmesin dalam sebuah antar jaringan. FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan (*download*) dan penggugahan (*upload*) berkasberkas komputer antara klien FTP dan *server* FTP. Sebuah Klien FTP merupakan aplikasi yang dapat mengeluarkan perintah-perintah FTP ke sebuah *server* FTP, sementara *server* FTP adalah sebuah Windows *Service* atau daemon yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah klien FTP. Perintahperintah FTP dapat digunakan untuk mengubah direktori, mengubah modus pengiriman antara biner dan ASCII, menggugah berkas komputer ke *server* FTP, serta mengunduh berkas dari *server* FTP.

Sebuah *server* FTP diakses dengan menggunakan Universal Resource Identifer (URI) dengan menggunakan format ftp://namaserver Klien FTP dapat menghubungi *server* FTPdengan membuka URI tersebut. FTP menggunakan protokol Transmission Control *Protocol* (TCP) untuk komunikasi data antara klien dan*server*, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum pengiriman data dimulai. Sebelum membuat koneksi, port TCP nomor 21 di sisi *server* akan "mendengarkan" percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai port pengatur (*control port*) untuk membuat sebuah koneksi antara klien dan*server*, untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada *server*dan juga mengembalikan respons *server* ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka *server* akan mulai membuka port
TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mengirim data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan penggugahan.

FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan password yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan password-nya untuk mengakses, men-*download*, dan meng-*upload* berkas-berkas yang ia kehendaki. Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode anonymous login, yakni dengan menggunakan nama pengguna anonymous dan password yang diisi dengan menggunakan alamat e-mail atau kosong.

ANALISA DAN PERANCANGAN

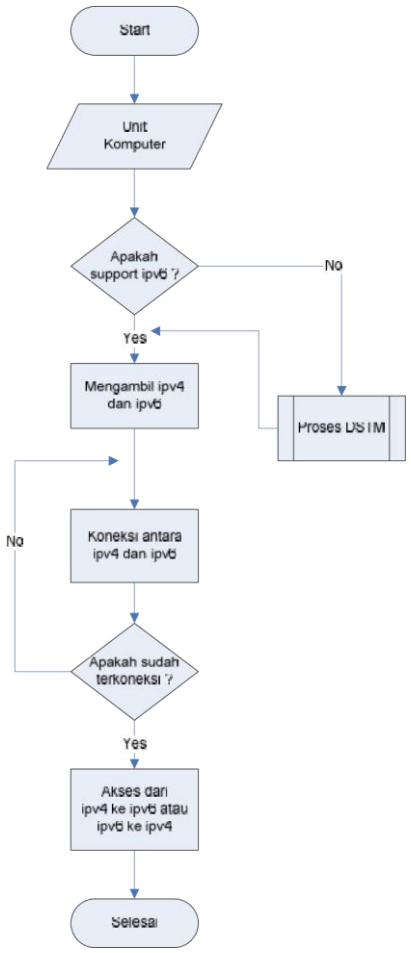
Perancangan sistem dibuat untuk memenuhi kebutuhan pengguna mengenai gambaran yang jelas bagaimana suatu sistem dibentuk. Perancangan sistem ini merupakan tahapan persiapan sistem sebelum diimplementasikan.

Perencanaan konsep sistem

Pada konsep ini merupakan terdapat sistem jaringan menggunakan IPv4 dan ada juga *system* jaringan menggunakan IPv6 karena ada yang berbeda maka bagaimana agar kedua sistem ini bisa berkomunikasi data antara satu sistem dengan sistem lain. Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, oleh karena itu menjadi permasalahan bagaimana jaringan IPv4 mampu berkomunikasi data dengan jaringan IPv6. Dalam implementasinya maka dibutuhkan aplikasi tambahan yaitu dibutuhkan suatu mekanisme tersendiri.

Mekanisme tersebut ada dengan cara DSTM (*DualStack Transition Mechanism*) merupakan suatu mekanisme penggambilan IPv4 untuk diubah

menjadi IPv6 yaitu dengan cara *enkapsulasi* atau pembukusan IPv4 yang akan di ambil untuk dijadikan IPv6 setalah itu akan dilakukan *dekapsulasi* yaitu pelepasan IP apabila sudah menjadi IPv6 dan akan dilakukan koneksi antar kedua sistem yang berbeda tersebut yaitu IPv4 koneksi dengan IPv6. Apabila sudah bisa terkoneksi kedua sistem maka akan dilakukan uji coba dengan parameter *throughput* dan *delay* dengan parameter tersebut bisa mengetahui hasil perbandingan dari kedua sistem yang dicapai dan perbandingan dari kedua masing-masing sistem tersebut untuk apakah baik dan buruk dari kedua sistem. Untuk lebih jelasnya bisa melihat desain *flowchart* sistem berikut:



Gambar 3.1 *flowchart* sistem

Pada gambar 3.1 cara kerja sebagai berikut, terdapat komputer IPv4 yang akan melakukan koneksi dengan IPv6 dilanjutkan dengan melakukan apakah support IPv6 komputer tersebut, apabila belum maka akan dilakukan proses DSTM (*DualStack Transition Mechanism)* tapi apabila sudah support IPv6 langsung melakukan pengambilan IPv4 untuk dilakukan koneksi antara IPv4 ke IPv6 atau IPv6 ke IPv4.

Gambar 3.2 *flowchart*DSTM

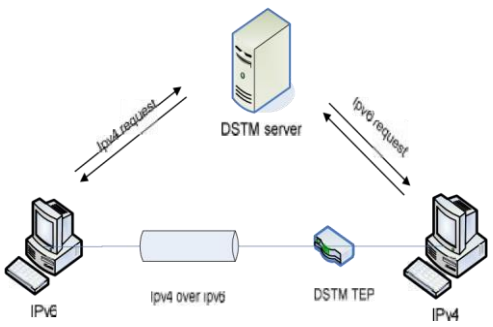
Pada gambar 3.2 cara kerja sebagai berikut, pertama melalui enkapsulasi IPv4 kedalam IPv6 jadi pengambilan suatu IP untuk diterjemahkan kedalam IPv6,apabila sudah di enkapsulasi maka diambil IP tersebut untuk dilakukan dekapsulasi IP yang sudah diterima untuk disampaikan dan selesai.

Perencanaan kebutuhan sistem

Berisi perancangan (desain) dari perangkat keras maupun lunak yang akan digunakan dalam melakukan simulasi sistem, penentuan perangkat lunak dan topologi yang akan digunakan, sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.

Desain Jaringan

Desain jaringan komputer nirkabel yang akan dibuat ini merupakan desain yang ingin dicapai dalam pengerjaan Tugas Akhir berikut merupakan design jaringan yang telah dibuat :



Gambar 3.3 Desain Jaringan

Pada gambar 3.3 dapat di gambarkan sebagai berikut IPv6 dapat berkomunikasi dengan *host*IPv4

dengan cara meminta alamat IPv4 terlebih dahulu ke *server*DSTM (*DualStack Transition Mechanism)*. *Server* DSTM (*DualStack Transition Mechanism)* memberikan IPv4 secara dinamis kepada IPv6yang kemudiandipetakan (*address mapping*) dengan alamat IPv6-nya dalam *cache* (penyimpanan sementara). Setelah mendapatkan IPv4tersebut, IPv6akan mengirimkan jalur khusus paket IPv4. Paket IPv4 yang akan dikirim oleh IPv6akan dibungkus dalam paket IPv6 menuju DSTM *Tunnel EndPoint* (DSTM TEP) untuk dibuka kembali dan dikirim sesuai tujuannya di jaringan IPv4.

***Hardware* yang digunakan**

Mulai dari tahap analisa sampai dengan tahap implementasi dalam ini menggunakan perangkat *laptop* dengan spesifikasi sebagai berikut:
***Hardware*:**

1. Laptop AMD Turion *Dual* Core 2.0 GHz
2. Harddisk 320 GB
3. RAM 2 GB
4. Keyboard

***Software* yang digunakan**

Adapun untuk kebutuhan *software* mulai tahap analisa sampai implementasi ini menggunakan beberapa *software* berikut:

Software :

1. Sistem Operasi windows
2. Aplikasi *DualStack Transition Mechanism (DSTM)*
3. *Wireshark*
4. *Xlight FTP server*

Perencanaan pengujian komunikasi data
Perencanaan

Pada perencanaan dibutuhkan 4 uji coba komunikasi data yang berbeda-beda yaitu sebagai berikut :

1. Pengujian komunikasi data IPv4 ke IPv4.

Pada sistem ini dilakukan hanya dengan koneksi antara IPv4 dan IPv4, komputer 1 disetting IPv4 dan komputer 2 disetting IPv4 setelah kedua sistem ini terkoneksi maka digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

2. Pengujian komunikasi data IPv6 ke IPv6.

Pada sistem ini dilakukan hanya dengan koneksi antara IPv6 dan IPv6, komputer 1 disetting IPv6 dan komputer 2 disetting IPv6 setelah kedua sistem ini terkoneksi maka digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

3. Pengujian komunikasi data IPv4 ke IPv6.

Pada sistem ini dilakukan dengan koneksi antara IPv4 dan IPv6, komputer 1 disetting IPv4 tanpa menghilangkan IPv6 yang dia punya dan komputer 2 disetting IPv6 tanpa menghilangkan IPv4 yang dia punya maka setelah itu digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

4. Pengujian komunikasi data IPv6 ke IPv4.

Pada sistem ini dilakukan dengan koneksi antara IPv4 dan IPv6, komputer 1 disetting IPv4 tanpa menghilangkan IPv6 yang dia punya dan komputer 2 disetting IPv6 tanpa menghilangkan IPv4 yang dia punya maka setelah itu digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

Pengujiaan

Pengujiaan dalam hal ini menggunakan cara men*download file* dari komputer 1 ke komputer 2 pada konfigurasi yang berbeda-beda. Agar mendapatkan hasil dari trafik yang didapat maka diperlakukan parameter, parameter tersebut adalah ukuran paket *file* yang akan di *download* masing-masing berukuran 5, 10, 20, 30,dan 40 *Megabyte* dalam keragaam pengukuran paket tersebut bisa dilihat nantinya pada table trafik yang didapat dan bisa dilihat perbedaan dari tiap ukuran *Megabyte* yang sudah ditentukan dengan *transfer* data dan *download* data.

HASIL DAN PEMBAHASAN

Pengujian Sistem

Pengujian sistem ini dilakukan sesuai dengan skenario yang sudah dibuat pada bab 3, dimana nantinya di bab ujicoba ini mempunyai beberapa bentuk dari pengujian antara lain Tujuan Pengujian Skenario dan Tahapan Pengujian.

Pengujian skenario komunikasi data IPv4 ke IPv4

Pada pengujian komunikasi data IPv4 ini ada 2 komputer yang akan saling berkomunikasi data, yang 2 komputer disetting dengan IPv4. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian

ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan*server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*

Pengujian skenario komunikasi data IPv6 ke IPv6

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, yang 2 komputer disetting dengan IPv6. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan*server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*

Pengujian skenario komunikasi data IPv4 ke IPv6

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, komputer disetting dengan IPv4 dan IPv6 mempunyai dua ip dalam 1 komputer. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *downloadfile* dan apabila request IP yang dituju mendapatkan balasan request Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *downloadfile* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*. Apabila komputer di setting IPv6 untuk komputer satu dan IPv4 untuk komputer dua maka tidak bisa melakukan koneksi di karena kedua ip saling berkaitan satu sama lain,maka dari itu harus ada 2 ip yaitu IPv4 dan IPv6 di setiap komputer.

Pengujian skenario komunikasi data IPv6 ke IPv4

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, komputer disetting dengan IPv4 dan IPv6 mempunyai dua IP dalam 1 komputer. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *downloadfile* dengan size yang berbeda-beda

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilain tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *downloadfile* dan apabila request IP yang dituju mendapatkan balasan request Dan*server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *downloadfile* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*. Apabila komputer di setting IPv6 untuk komputer satu dan IPv4 untuk komputer

dua maka tidak bisa melakukan koneksi di karena kedua ip saling berkaitan satu sama lain,maka dari itu harus ada 2 ip yaitu IPv4 dan IPv6 di setiap komputer.

Analisa Pengujian

Pada tahap ini dilakukan analisa pengujian yang didasarkan pada penilaian pengujian yang ada diskenario-skenario di atas yang terdiri dari 4 skenario, dengan parameter *throughput* dan *delay* berikut merupakan hasil dari keseluruhan uji coba pada skenario tersebut :

No	Skenario	*throughput*		*delay*
1	IPv4 ke IPv4	5Mb	58.256	36.683
		10Mb	23.847	89.737
		20Mb	17.788	145.770
		30Mb	15.236	199.917
		40Mb	13.271	263.659
2	ke IPv6 IPv6	5Mb	38.908	43.359
		10Mb	22.394	95.560
		20Mb	17.519	200.073
		30Mb	14.417	266.075
		40Mb	13.516	325.765

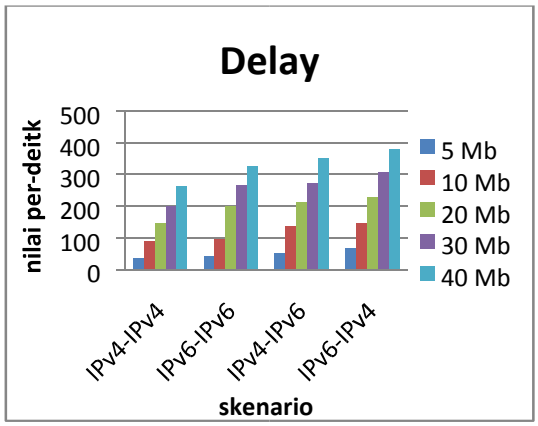
Table 5.1 hasil keseluruhan skenario 1 dan 2

No	Skenario	*throughput*		*delay*
3	IPv4 ke IPv6	5Mb	41.237	51.823
		10Mb	20.502	137.596
		20Mb	13.144	214.622
		30Mb	14.945	272.541
		40Mb	14.357	352.308
4	IPv6 ke IPv4	5Mb	31.445	67.897
		10Mb	18.596	145.297
		20Mb	11.231	230.876
		30Mb	9.897	307.782
		40Mb	9.171	381.540

Table 5.2 hasil keseluruhan skenario 3 dan 4

Analisa Pengujian *delay*

Dari hasil uji coba maka akan dilakukan analisa di data atau hasil yang di peroleh dari *delay* dengan ukuran berbeda-beda,berikut analisa dari *delay* :



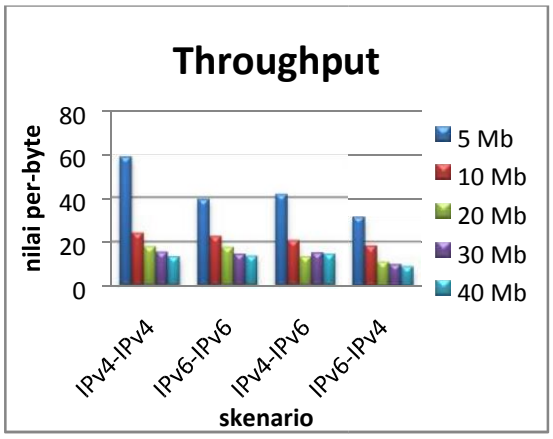
Gambar 5.33 Diagram hasil *delay*

Pada gambar 5.33 merupakan hasil dari *delay* yang sudah di uji coba di scenario-skenario sebelumnya dalam bentuk diagram. Dapat dilihat pada diagram bahwa ada perbedaan-perbedaan dalam nilai yang diperoleh di setiap masingmasing pengujian. Pada pengujian IPv6 ke IPv4 mendapatkan lebih besar nilai dari pada pengujian yang lain dan IPv4 ke IPv4 mendapatkan nilai yang lebih kecil dari pada pengujian yang lain.

Pada *delay* ini ditunjukkan nilai yang besar pada uji coba IPv6 ke IPv4 dikarena adanya mekanisme tambahan DSTM yang mempengaruhi *transfer* data yang dikirim atau diterima sehingga nilai *delay* lebih besar dari yang tidak ada mekanisme tambahan DSTM dalam *transfer* data yang diterima. Dan untuk uji coba yang IPv4 ke IPv6 juga ada mekanisme tambahan DSTM juga mempengaruhi *transfer* data yang dikirim atau diterima sehingga mendapatkan nilai *delay* lebih besar dari yang tidak ada mekanisme tambahan DSTM dalam *transfer* data yang diterima. Antara uji coba IPv4 ke IPv6 dan IPv6 ke IPv4 mengalami sedikit peningkatan nilai, kedua uji coba ini yang menggunakan mekanisme tambahan DSTM lebih bagus IPv4 ke IPv6 karena lebih kecil nilai yang didapat. Dan untuk yang tidak dipengaruhi mekanisme tambahan DSTM seperti IPv4 ke IPv4 dan IPv6 ke IPv6 lebih bagus IPv4 ke IPv4 karena nilai yang diperoleh lebih kecil dan dari kesimpulan diantara kesemua uji coba yang sudah dilakukan nilai yang paling bagus didapat yaitu IPv4 ke IPv4 dan yang paling kurang bagus yaitu IPv6 ke IPv4.

Analisa Pengujian *throughput*

Dari hasil uji coba maka akan dilakukan analisa di data atau hasil yang di peroleh dari *throughput* dengan ukuran berbeda-beda,berikut analisa dari *throughput* :



Gambar 5.34 Diagram hasil *Throughput*

Pada gambar 5.34 merupakan hasil dari *throughput* yang sudah di uji coba di scenarioskenario sebelumnya dalam bentuk diagram. Dapat dilihat pada diagram bahwa ada perbedaanperbedaan dalam nilai *Throughput* yang diperoleh di setiap masing-masing pengujiannya. Untuk nilai *Throughput* dengan nilai ukuran *file* 5Mb mendapatkan nilai yang sangat tinggi disetiap uji coba yang dilakukan merupakan nilai yang didapat sangat cepat dalam hal mensampaikan data dan merupakan nilai yang sangat bagus. Dan ukuran *file* yang selanjutnya seperti ukuran 10Mb, 20Mb, 30Mb, 40Mb mengalami banyak penurunan disetiap ukuran *file* yang berbedabeda,itu semua dikarena ukuran *file* yang semakin besar dapat di simpulkan semakain besar ukuran *file* untuk mensampaikan data maka nilai yang didapat juga akan mengalami penurunan di setiap hasilnya.

Pada uji coba IPv6 ke IPv4 memiliki nilai yang sangat kecil di antara semua uji coba yang dilakukan dan itu merupakan nilai yang tidak bagus dan untuk IPv4 ke IPv4 memiliki nilai yang cukup besar diantara uji coba yang lain dan itu merupakan nilai yang sangat bagus dalam mendapatkan nilai *throughput*.

Tidak hanya ukuran *file* yang mempengaruhi turunnya nilai *throughput* dalam IPv6 ke IPv4 dan IPv4 ke IPv6 mendapatkan nilai kecil juga adanya pengaruh mekanisme tambahan DSTM seperti halnya uji coba pada *transferfile* sebelumnya juga karena adanya mekanisme tambahan DSTM. Untuk yang tidak menggunakan mekanisme tambahan DSTM mendapatkan nilai yang cukup bagus karena tidak ada mekanisme tambahan DSTM tersebut, untuk IPv4 ke IPv4 mendapat nilai yang sangat bagus dan tinggi diantara uji coba yang lainnya, IPv6 ke IPv6 mengalami penurunan sedikit dengan uji coba IPv4 ke IPv4 dan itu semua uji coba tanpa menggunakan mekanisme tambahan DSTM.

PENUTUP

Kesimpulan

Dari pengujian yang dilakukan pada skenario-skenario yang telah di uji coba dapat disimpulkan bahwa sekenario yang menggunakan IPv4 ke IPv4 merupakan yang sangat bagus komunikasi datanya antara komputer 1 dengan komputer 2 mengalami *throughput* dan *delay* yang sangat kecil dan itu merupakan hasil yang sangat baik dan untuk sekenario yang menggunakan IPv6 ke IPv4 mendapatkan nilai yang sangat besar diantara nilai hasil uji coba yang lain dan itu merupakan hasil yang sangat tidak bagus.

Saran

Berdasarkan apa yang sudah dilakukan pada Tugas Akhir ini maka didapatkan beberapa Saran yang mungkin bisa digunakan untuk pengembangan, yaitu :

1. Karena uji coba menggunakan parameter uji coba *throughput* dan *delay*. bisa menggunakan parameter uji coba yang lain..
2. Diharapkan juga bisa diimplementasikan di *operation system* lain dan metode lain.

DAFTAR PUSTAKA

[1] Bound J. 2004.*Dual Stack Transition Mechanism*. IETF draft-bound-dstm-exp-01.txt.

[2] Houston G. 2005. *IPv4 AddresReport*. Http://www.potaroo.net/tools/ipv4/index.html

[3] Jogiyanto. 1995.*Analisa dan Perancangan Sistem Informasi*. Andy Offset.Yogyakarta

[4] Postel J. 1981. *Internet Protocol*. Request of Comment 791. IETF

[5] Ruiz PM. 2002.*Dual Stack Transition Mechanism*.www..ipv6es.com/02/docs/pedro_ruiz_2.pdf