



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI APLIKASI FTP (*FILE TRANSFER
PROTOCOL*) PADA JARINGAN MOBILE IPV6 BERDASARKAN DELAY,
THROUGHPUT DAN TRANSFER TIME**

SKRIPSI

WINDA ACTARINA

0606078531

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
DESEMBER 2010**



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI APLIKASI FTP (*FILE TRANSFER
PROTOCOL*) PADA JARINGAN MOBILE IPV6 BERDASARKAN DELAY,
THROUGHPUT DAN TRANSFER TIME**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

WINDA ACTARINA

0606078531

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
DEPOK
DESEMBER 2010**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Winda Actarina

NPM : 0606078531

Tanda Tangan :

Tanggal : 15 Desember 2010

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Winda Actarina

NPM : 0606078531

Program Studi : Teknik Komputer

Judul Skripsi : Analisa Performansi Aplikasi FTP (*File Transfer Protocol*)
Pada Jaringan *Mobile IPv6* Berdasarkan *Delay, Throughput*
dan *Transfer Time*

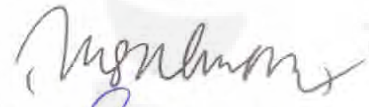
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

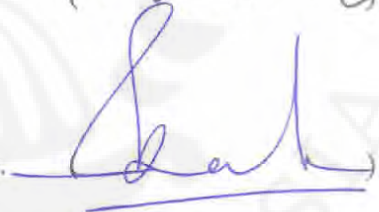
Pembimbing : Ir. Endang Sriningsih, MT, Si



Penguji : Muhammad Salman ST, MIT



Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng.



Ditetapkan di : Depok

Tanggal : 04 Januari 2011

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat wajib untuk mencapai gelar Sarjana Teknik Jurusan Teknik Komputer pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa adanya dorongan, bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya dalam penyelesaian skripsi ini. Oleh karena itu, saya mengucapkan banyak terima kasih kepada :

- (1) Ir. Endang Sriningsih, MT, selaku Dosen Pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) Orang Tua dan Keluarga saya yang telah memberikan bantuan dukungan material dan moral ; dan
- (3) Sahabat – sahabat saya yang telah mendukung dan banyak membantu saya selama menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu dan saya mengucapkan maaf yang sebesar – besarnya apabila ada kata – kata atau tindakan yang kurang berkenan selama penyelesaian skripsi ini. Semoga skripsi ini membawa manfaat bagi pembaca dan bagi pengembangan ilmu. Penulis menyadari bahwa dalam skripsi ini masih terdapat kekurangan dan masih jauh dari kesempurnaan, maka saran dan kritik yang bersifat membangun akan sangat dibutuhkan.

Depok, 2010

Penulis

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

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

Nama : Winda Actarina
NPM : 0606078531
Program Studi : Teknik Komputer
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

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

ANALISA PERFORMANSI APLIKASI FTP (*FILE TRANSFER PROTOCOL*) PADA JARINGAN MOBILE IPV6 BERDASARKAN DELAY, THROUGHPUT DAN TRANSFER TIME

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

Demikian pernyataan ini saya buat dengan yang sebenarnya.

Dibuat di : Depok
Pada tanggal : 15 Desember 2010
Yang menyatakan

(Winda Actarina)

ABSTRAK

Nama : Winda Actarina
Program Studi : Teknik Komputer
Judul : Analisa Performansi Aplikasi FTP (*File Transfer Protocol*) Pada Jaringan *Mobile IPv6* Berdasarkan *Delay*, *Throughput* dan *Transfer Time*

Mobile IP merupakan teknologi dalam suatu infrastuktur jaringan IP yang memperbolehkan satu atau beberapa host dapat berpindah jaringan dari jaringan satu ke jaringan yang lainnya tanpa terputusnya proses komunikasi yang dilakukan host tersebut.

Dunia telekomunikasi semakin berkembang pesat terutama pada Internet Protocol (IP) dengan adanya protocol internet IPv6 yang memperbarui IPv4. Adanya IPv6 ini diharapkan dapat mengungguli performansi dari IPv4 terutama pada *mobile IP* yaitu dengan dikembangkannya *mobile IPv6*. Tujuan dari penulisan skripsi ini adalah untuk menganalisa performansi dari jaringan *mobile IPv6* dengan aplikasi yang diterapkan adalah aplikasi FTP (*File Transfer Protocol*).

Topologi jaringan *mobile IPv6* yang dibuat terdiri dari *home agent*, *correspondent node* sebagai *server* FTP, *foreign router* dan *home router* sebagai *intermediate* dan *mobile node* sebagai *client*. Pengambilan data dilakukan dengan cara men-download file dengan ukuran yang berbeda – beda dari *server* ke *client*. Parameter uji coba yang akan dibandingkan adalah *delay*, *throughput* dan *transfer time*.

Kesimpulan yang didapat bahwa pada skenario 1 memiliki nilai *throughput*, *transfer time* dan *delay* paling baik dibandingkan dengan skenario 2. Berdasarkan parameter *throughput*, pada skenario 2 mengalami penurunan *throughput* sebesar 21 % untuk file pdf, 21.46 % untuk file doc dan 25.83 % untuk file jpg. Berdasarkan parameter *transfer time*, pada skenario 2 mengalami kenaikan *transfer time* sebesar 65.3 % untuk file jenis pdf, 39.58 % untuk jenis file doc dan 13.61 % untuk jenis file jpg. Berdasarkan parameter *delay*, pada skenario 2 mengalami kenaikan *delay* sebesar 37.93 % untuk file jenis pdf, 16.44 % untuk jenis file doc dan 2.69 % untuk jenis file jpg.

Kata Kunci :

IPv6, *Mobile IP*, *File Transfer Protocol*, *Throughput*, *delay*, dan *transfer time*

ABSTRACT

Name : Winda Actarina
Study Program: Computer Engineering
Title : Performance Analysis of FTP (File Transfer Protocol)
Application in Mobile IPv6 Network Based on Delay,
Throughput and Transfer Time

Mobile IP (or IP mobility) is an Internet Engineering Task Force (IETF) standard communications protocol that is designed to allow mobile device users to move from one network to another while maintaining a permanent IP address or maintaining existing connections.

The rapid progress of telecommunication world particularly in Internet Protocol (IP) has created IPv6 as renovation of IPv4. The existence of IPv6 is expected to surpass IPv4 performance primarily in mobile IP. The aim of this thesis was to analyse the performance of Mobil Ipv6 Network with FTP (*File Transfer Protocol*) as the applied Application.

The topology of mobile IPv6 network consists of home agent and correspondent node as server FTP; foreign router and home router as intermediate; and mobile node as client. The data removal was conducted by file downloaded with different sizes from server to client. The parameter of test and trial would be compared to delay, throughput and transfer time.

The research result concluded that scenario 1 had the best score of throughput, transfer time and delay compared to scenario 2. Based on the throughput parameter, there was a throughput decrease in scenario 2 consisting of 21% for pdf file, 21.46 % for doc file, as well as 25.83 % for jpg file. Based on transfer time parameter, there was a transfer time increase in scenario 2 consisting of 65.3 % for pdf file, 39.58 % for doc file, as well as 13.61 % for jpg file. Based on delay parameter, there was a delay increase in scenario 2 consisting of 37.93 % for pdf file, 16.44 % for doc file, as well as 2.69 % for jpg file.

Key Words:

IPv6, Mobile IP, File Transfer Protocol, Throughput, delay, and transfer time

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN ORISINILITAS	ii
LEMBAR PENGESAHAN.....	iii
KATA PENGANTAR	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL	xii
DAFTAR SINGKATAN	xiii
1. PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan	2
1.3 Batasan Masalah	2
1.4 Metode Penelitian	3
1.5 Sistematika Penulisan	3
2. IPv6 DAN MOBILE IPv6.....	5
2.1 Definisi dan Latar Belakang IPv6.....	5
2.2 Spesifikasi IPv6	5
2.2.1 Struktur IPv6.....	6
2.2.2 Pengalamatan IPv6.....	8
2.2.2.1 Format Alamat IPv6.....	8
2.2.2.2 Penyederhanaan Bentuk Alamat	9
2.2.2.3 Format Prefiks.....	9
2.2.2.4 Jenis – Jenis Alamat IPv6.....	10
2.2.2.5 Perbedaan IPv6 dengan IPv4	11
2.2.3 Fitur – fitur IPv6	12
2.3 Mobile IPv6.....	13

2.3.1 Mobile IP	13
2.3.2 Mobile IPv6.....	13
2.3.2.1 Mekanisme <i>Mobile IPv6</i>	14
2.3.2.2 Perbedaan Mobile IPv6 dan Mobile IPv4	16
2.3.2.3 Proses <i>Handover</i> pada Mobile IPv6	17
2.4 File Transfer Protocol (FTP).....	18
2.4.1 <i>Anonymous</i>	20
2.4.2 <i>Authentication User</i>	21
3. PEMBANGUNAN JARINGAN DAN METODE PENGAMBILAN	
DATA	22
3.1 Topologi Jaringan	22
3.2 Perangkat Lunak yang Digunakan.....	25
3.3 Konfigurasi Jaringan.....	26
3.4 Aplikasi Uji	28
3.4.1 Aplikasi VSFTP	28
3.4.2 Tahap Instalasi dan Konfigurasi VSFTP.....	28
3.4.3 <i>Testing VSFTP</i>	30
3.5 Metode Pengambilan Data	32
3.5.1 Skenario Pengujian	34
4. ANALISA DATA DAN PERFORMANSI LAYANAN MOBILE IPV6	
DENGAN FTP	36
4.1 Analisa Performa Jaringan Pada FTP	36
4.2 Analisa Pada <i>Home Link</i>	38
4.2.1 Analisa <i>Throughput</i> (Skenario 1)	38
4.2.2 Analisa <i>Transfer Time</i> (Skenario 1)	42
4.2.3 Analisa <i>Delay</i> (Skenario 1)	43
4.3 Analisa Pada <i>Foreign Link</i>	44
4.3.1 Analisa <i>Throughput</i> (Skenario 2)	44
4.3.2 Analisa <i>Transfer Time</i> (Skenario 2)	46
4.3.3 Analisa <i>Delay</i> (Skenario 2)	47
4.4 Analisa Perbandingan Skenario 1 dan Skenario 2	48
4.4.1 Analisa Perbandingan <i>Throughput</i>	48
4.4.2 Analisa Perbandingan <i>Transfer Time</i>	51

4.4.3 Analisa Perbandingan <i>Delay</i>	53
4.5 Analisa Pada <i>Handover</i>	56
4.5.1 Analisa <i>Throughput</i> (Skenario 3)	57
4.5.2 Analisa <i>Transfer Time</i> (Skenario 3).....	58
4.5.3 Analisa <i>Delay</i> (Skenario 3)	59
5. KESIMPULAN	61
DAFTAR REFERENSI	62
DAFTAR LAMPIRAN.....	63
LAMPIRAN 1. Konfigurasi <i>Mobile Node</i>	63
LAMPIRAN 2. Konfigurasi <i>Home Agent</i>	64
LAMPIRAN 3. Konfigurasi <i>Home Router</i>	65
LAMPIRAN 4. Konfigurasi <i>Foreign Router</i>	66
LAMPIRAN 5. Konfigurasi <i>Coresspondent Node</i>	68
LAMPIRAN 6. Konfigurasi VSFTPD.....	69

DAFTAR GAMBAR

Gambar 2.1 Struktur packet IPv6	6
Gambar 2.2 Format Header IPv6	6
Gambar 2.3 Perbandingan packet IPv6 dan packet IPv4	7
Gambar 2.4 Mekanisme <i>bidirectional tunneling</i>	15
Gambar 2.5 Mekanisme <i>Route Optimization</i>	16
Gambar 2.6 Proses kerja FTP	20
Gambar 3. 1 Jaringan Mobile IPv6.....	22
Gambar 3. 2 Konfigurasi Jaringan Mobile IPv6	26
Gambar 3. 3 Autentikasi FTP.....	31
Gambar 3. 4 Tampilan Direktori File FTP.....	32
Gambar 3. 5 Tampilan <i>save file</i> saat men- <i>download</i>	32
Gambar 3. 6 Skenario 1.....	34
Gambar 3. 7 Skenario 2.....	34
Gambar 3. 8 Skenario 3.....	35
Gambar 4. 1 Tampilan FTP client saat terkoneksi dengan server.....	37
Gambar 4. 2 <i>Throughput</i> pada <i>Summary</i> Wireshark (Skenario 1)	39
Gambar 4. 3 <i>Chapture</i> Wireshark ketika Terjadi <i>Bad Checksum</i>	40
Gambar 4. 4 <i>capture</i> wireshark ketika mengalami <i>TCP retransmission</i>	41
Gambar 4. 5 Detail <i>TCP Bad Checksum</i>	41
Gambar 4. 6 Grafik <i>Throughput</i> Pada <i>Home Link</i>	41
Gambar 4. 7 <i>Transfer Time</i> pada <i>Summary</i> Wireshark (Skenario 1)	42
Gambar 4. 8 <i>Throughput</i> pada <i>Summary</i> Wireshark (Skenario 2).....	45
Gambar 4. 9 Grafik <i>Throughput</i> Pada <i>Foreign Link</i>	46
Gambar 4. 10 <i>Transfer Time</i> pada <i>Summary</i> Wireshark (Skenario 2)	46
Gambar 4. 11 Perbandingan <i>Throughput</i> Skenario 1 dan 2 pada File Pdf	49
Gambar 4. 12 Perbandingan <i>Throughput</i> Skenario 1 dan 2 pada File Doc.....	50
Gambar 4. 13 Perbandingan <i>Throughput</i> Skenario 1 dan 2 pada File Jpg.....	50
Gambar 4. 14 Perbandingan <i>Transfer Time</i> Skenario 1 dan 2 pada File Pdf.....	52
Gambar 4. 15 Perbandingan <i>Transfer Time</i> Skenario 1 dan 2 pada File Doc	52

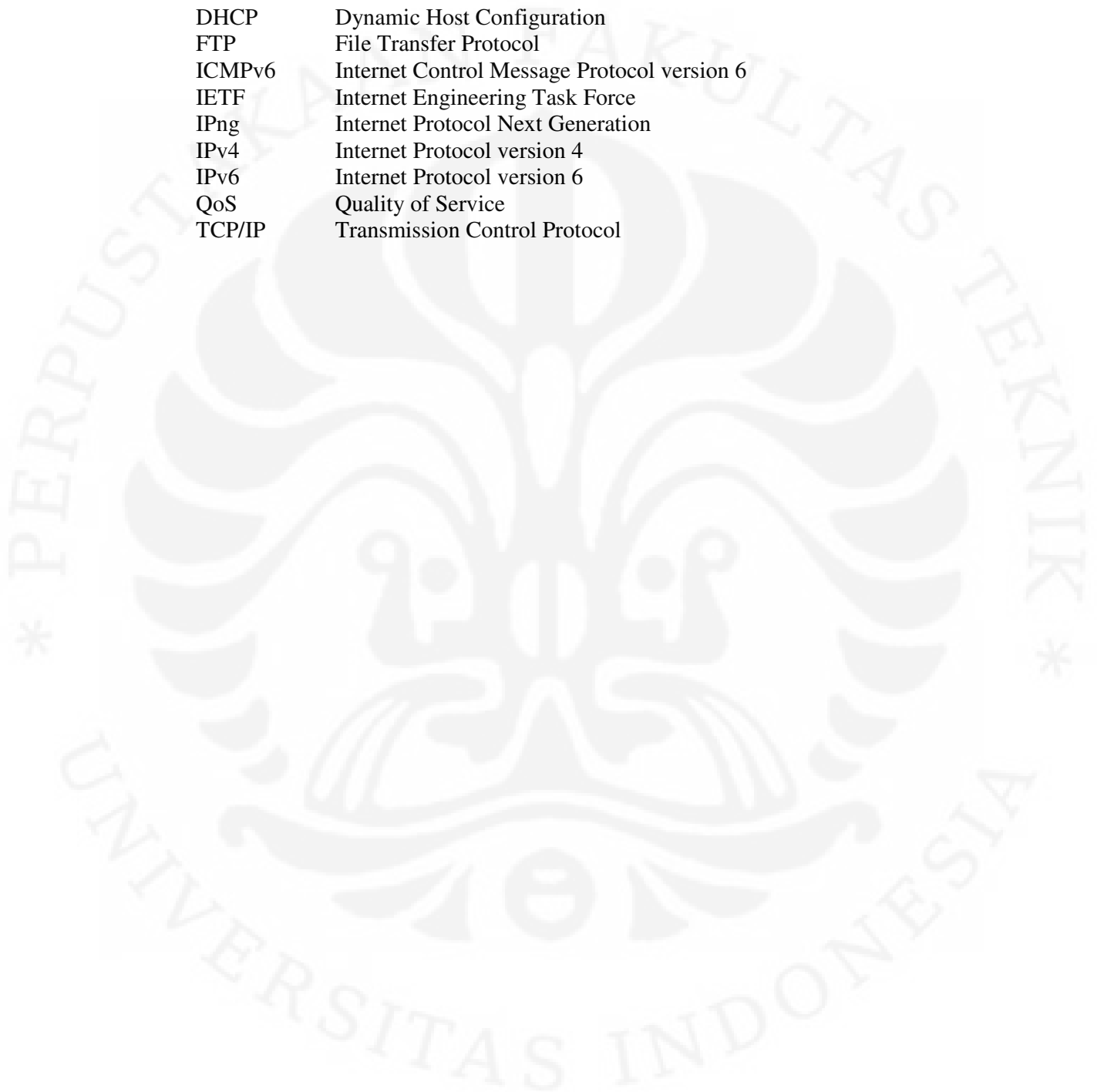
Gambar 4. 16 Perbandingan <i>Transfer Time</i> Skenario 1 dan 2 pada File Jpg.....	53
Gambar 4. 17 Perbandingan <i>Delay</i> Skenario 1 dan 2 pada File Pdf	54
Gambar 4. 18 Perbandingan <i>Delay</i> Skenario 1 dan 2 pada File Doc	55
Gambar 4. 19 Perbandingan <i>Delay</i> Skenario 1 dan 2 pada File Jpg	55
Gambar 4. 20 <i>Capture</i> Wireshark pada saat <i>Handover</i>	57
Gambar 4. 21 <i>Throughput</i> pada <i>Summary</i> Wireshark (Skenario 3)	57
Gambar 4. 22 <i>Transfer Time</i> pada <i>Summary</i> Wireshark (Skenario 3)	58
Gambar 4. 23 <i>Delay</i> pada <i>Summary</i> Wireshark (Skenario 3)	59

DAFTAR TABEL

Tabel 2.1 Perbedaan IPv6 dan IPv4.....	11
Tabel 3. 1 Spesifikasi Mobile node	23
Tabel 3. 2 Spesifikasi Correspondent Node.....	23
Tabel 3. 3 Spesifikasi Home Agent	24
Tabel 3. 4 Spesifikasi Akses Point 1.....	25
Tabel 3. 5 Spesifikasi Akses Point 2.....	25
Tabel 3. 6 Spesifikasi Switch	25
Tabel 4. 1 File Uji Coba.....	37
Tabel 4. 2 Data Nilai <i>Throughput</i>	39
Tabel 4. 3 Data Nilai <i>Transfer time</i>	42
Tabel 4. 4 Data Perhitungan <i>Delay</i>	43
Tabel 4. 5 Data Nilai <i>Throughput</i>	45
Tabel 4. 6 Data Nilai <i>Transfer Time</i>	47
Tabel 4. 7 Data Perhitungan <i>Delay</i>	47
Tabel 4. 8 Data Nilai <i>Throughput</i> Keseluruhan	48
Tabel 4. 9 Data Nilai <i>Transfer Time</i> Keseluruhan.....	51
Tabel 4. 10 Data Nilai <i>Delay</i> Keseluruhan	53
Tabel 4. 11 Data Nilai <i>Throughput</i>	58
Tabel 4. 12 Data Nilai <i>Transfer Time</i>	58
Tabel 4. 13 Data Nilai <i>Delay</i>	59

DAFTAR SINGKATAN

DHCP	Dynamic Host Configuration
FTP	File Transfer Protocol
ICMPv6	Internet Control Message Protocol version 6
IETF	Internet Engineering Task Force
IPng	Internet Protocol Next Generation
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
QoS	Quality of Service
TCP/IP	Transmission Control Protocol



BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

IP atau *Internet Protocol* berbasis TCP/IP merupakan Internet Protokol yang sangat penting dalam jaringan Komputer. TCP/IP ini melakukan pengalamatan dan *routing* paket data antar *host* di suatu jaringan. IP yang dikembangkan saat ini adalah IPv6 (*Internet Protocol Version 6*) merupakan protokol generasi baru yang didesain oleh *Internet Engineering Task Force* (IETF). IPv6 biasa disebut dengan *Internet Protocol Next Generation* (IPng). IPv6 inilah yang dikembangkan dan dapat menggantikan IPv4 yaitu internet protokol yang muncul sebelum IPv6 yang saat ini sukses digunakan dalam jaringan komputer. IPv6 memiliki kemampuan yang lebih unggul dari IPv4 dalam hal pengalamatannya. Banyaknya aplikasi saat ini yang diaplikasikan menggunakan internet seperti FTP (*File Transfer Protocol*), E-mail, Remote Access, Video Streaming, Video Conference, dan masih banyak lagi aplikasi dan multimedia yang menggunakan *internet* serta banyaknya penggunaan pengalamatan *internet*, maka kapasitas dan kemampuan jaringan berbasis IP harus dapat mengikuti dan mengimbangi aplikasi yang ada untuk menyediakan layanan dan fungsi yang diperlukan dalam jangka panjang.

Pengalamatan yang digunakan dalam IPv4 adalah pengalamatan 32 bit dengan jumlah pengalamatan sebesar 2^{32} atau 4.294.967.296 ($4,294 \times 10^9$) alamat. Jumlah pengalamatan ini akan semakin sedikit dengan banyaknya pengguna yang ada, sehingga akan sulit berkembangnya penyedia layanan internet (ISP). Dengan bertambahnya aplikasi dan pengguna yang menggunakan pengalamatan internet maka terdapat solusi dengan dikembangkannya IPv6 yang menyediakan pengalamatan dalam jumlah yang lebih besar yaitu berbasis 128 bit dengan kemampuan pengalamatan 2^{128} alamat atau sekitar 40.282.366.920.938.463.463.374.607.431.768.211.456 ($3,402 \times 10^{38}$) alamat.

Mobile IP merupakan Internet Protokol yang memiliki kemampuan bergerak secara mobile. *Mobile IP* merupakan teknologi yang memungkinkan

node (*host*) dapat berpindah dari suatu subnet ke subnet lain tanpa memutuskan koneksi ke jaringan tersebut. *Mobile IP* diterapkan pada Internet Protokol IPv4 dan IPv6.

FTP (*File Transfer Protocol*) merupakan salah satu aplikasi yang dapat diterapkan dalam jaringan IPv6. Jenis – jenis FTP yang dapat digunakan dalam jaringan IPv6 yang menggunakan sistem operasi linux ubuntu 8.04 adalah proFTP, TFTP, VSFTPD (*Very Secure Transfer Protocol Daemon*) dan masih banyak lagi. Pada skripsi ini yang digunakan adalah VSFTPD. VSFTPD digunakan selain mensupport IPv6, juga karena kemudahan konfigurasi dan penggunaannya serta memiliki keamanan yang cukup baik.

1.2 TUJUAN

Penulisan skripsi ini bertujuan untuk mendesain atau merancang suatu jaringan berbasis *mobile* IPv6 dan menganalisa performansi dengan aplikasi FTP (*File Transfer Protocol*) yaitu VSFTPD (*Very Secure File Transfer Protocol Demon*) yang diterapkan pada jaringan *mobile* IPv6 murni. Parameter – parameter uji FTP pada kinerja *mobile* IPv6 yang akan dianalisa antara lain adalah *delay*, *throughput* dan *transfer time* dengan menggunakan jenis file yang berbeda. Berdasarkan parameter – parameter tersebut, suatu *mobile node* dapat dibandingkan dan dianalisa performansinya serta menganalisa pengaruh perbedaan jenis file pada performansi jaringan *mobile* IPv6.

1.3 BATASAN MASALAH

Batasan masalah dalam penulisan skripsi ini adalah menganalisa performansi atau kemampuan dari FTP (*File Transfer Protocol*) yang diterapkan pada jaringan *mobile* IPv6 murni. Dalam analisa performa FTP dibatasi pada parameter *delay*, *throughput* dan *transfer time*. Pengambilan data akan dibatasi pada FTP *download*. Sistem operasi yang digunakan untuk membangun jaringan *mobile* IPv6 yaitu Ubuntu 8.04. Konfigurasi jaringan *mobile* IPv6 dilakukan di Ubuntu dan sedangkan aplikasi FTP yaitu VSFTPD dikonfigurasi di bagian yang difungsikan sebagai *server* yaitu *Correspondent node* tempat dimana file yang akan *download* oleh *client* atau *mobile node* berada. Untuk pengambilan

datanya, file yang digunakan dibedakan menjadi tiga jenis file dengan ukuran yang berbeda yaitu KB. Berbeda dengan pengambilan data pada saat handover, file yang digunakan adalah file jenis rar dan ukuran yang lebih besar yaitu MB.

1.4 METODE PENELITIAN

Metode yang digunakan dalam penelitian untuk penulisan skripsi ini adalah merancang atau mendesain jaringan *mobile IPv6* skala kecil. Dalam pengujian yang digunakan adalah 4 buah PC yang diterapkan masing – masing sebagai *home agent*, *correspondent node* sebagai *server*, *foreign router*, *home router* dan satu buah laptop sebagai *mobile node* yang digunakan sebagai *client*. Jaringan tersebut akan dikonfigurasi agar dapat digunakan dalam menganalisa performansi dari jaringan *mobile IPv6* yang di aplikasikan untuk FTP. Parameter – parameter uji coba yang digunakan untuk dianalisa antara lain adalah *delay*, *throughput* dan *transfer time*. Proses pengambilan data dilakukan dengan cara sisi *client* melakukan *download* file dari *server* dengan kapasitas file yang berbeda – beda, kemudian dilakukan analisa performansi FTP terhadap ketiga parameter uji coba tersebut dengan menggunakan wireshark.

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan skripsi ini disusun atas 5 bab dengan pembagian sebagai berikut :

Bab 1 Pendahuluan

Bab ini berisi tentang latar belakang, tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

Bab 2 IPv6 dan *Mobile IPv6*

Bab ini berisi tentang penjelasan IPv6 dan *mobile IPv6* sebagai dasar teori dan referensi penunjang dalam menyusun skripsi ini.

Bab 3 Pembangunan Jaringan dan Metode Pengambilan Data

Bab ini berisi tentang penjelasan perancangan topologi jaringan dan konfigurasi *mobile IPv6*. Pada bab ini juga berisi mengenai metode pengambilan data.

Bab 4 Analisa Data dan Kualitas Layanan *Mobile IPv6* dengan FTP

Bab ini berisi tentang pengujian aplikasi pada jaringan dan analisa hasil data yang di ambil.

Bab 5 Kesimpulan

Bab ini berisi tentang kesimpulan yang didapat dari hasil analisa yang dilakukan pada hasil data yang didapat.



BAB 2

IPv6 DAN MOBILE IPv6

2.1 DEFINISI DAN LATAR BELAKANG IPv6

Semakin berkembangnya teknologi didunia terutama pada internet protokol atau IP dan banyaknya pengguna internet yang melebihi kapasitas pengalamatan IPv4 yang semakin tidak dapat memenuhi kebutuhan dari alamat IP, maka dibuatlah alamat IP yang dapat memenuhi kebutuhan yaitu IPv6.

IPv6 merupakan singkatan dari *Internet Protocol versi 6* atau biasa disebut dengan IPng atau singkatan dari *Internet Protocol Next Generation*. IPv6 ini merupakan internet protokol versi baru yang dipelajari dan dikembangkan untuk dapat mengikuti perkembangan teknologi internet untuk kedepannya agar lebih baik dari versi yang sebelumnya yaitu IPv4. IPv6 ini dapat memperbaiki kekurangan dari IPv4 dan diharapkan dapat memenuhi kebutuhan alamat IP yang semakin banyak dalam jangka waktu yang lama.

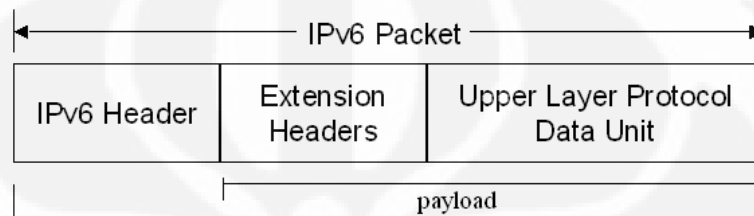
Alamat protokol IPv6 ini memiliki keunggulan lebih banyak dari protokol IPv4. Panjang pengalamatan IPv6 adalah 28 bit dengan jumlah alamat 2^{128} atau 40.282.366.920.938.463.463.374.607.431.768.211.456 ($3,402 \times 10^{38}$) alamat dan sedangkan IPv4 panjang alamatnya 32 bit dengan jumlah alamat 2^{32} alamat atau sekitar 4.294.967.296 ($4,294 \times 10^9$) alamat. Dengan kapasitas pengalamatan IPv6 yang melebihi IPv4, tidak diragukan lagi bila IPv6 dapat mengatasi permasalahan banyaknya pengguna internet di seluruh dunia.

2.2 SPESIFIKASI IPv6

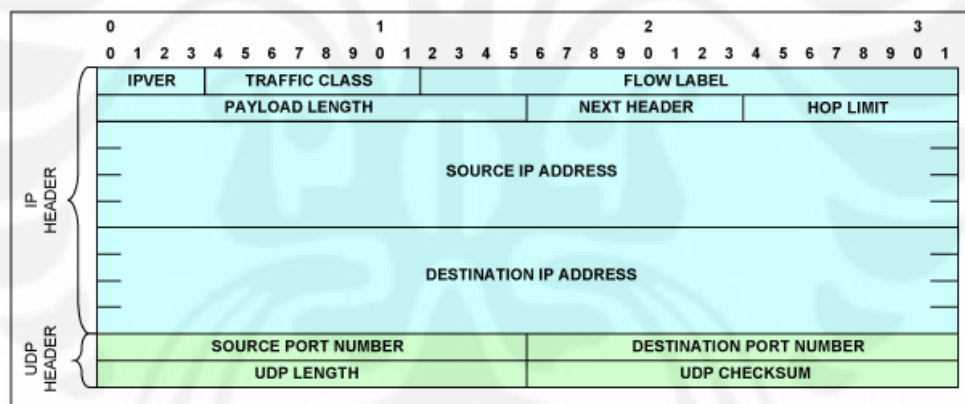
IPv6 memiliki spesifikasi yang berbeda dari IPv4 mulai dari *header*, struktur, pengalamatan, dan kemampuan atau kelebihan dari masing – masing protokol. IPv6 memiliki panjang protokol 128 bit dengan penulisan alamat menggunakan bilangan hexadecimal dan memiliki header yang panjangnya 40 bytes.

2.2.1 Struktur IPv6

Struktur pada IPv6 terdiri dari IPv6 *header*, *extension headers* dan *upper layer protocol data unit*. Struktur IPv6 dapat diperlihatkan pada Gambar 2.1. IPv6 *header* memiliki kapasitas sebesar 40 bytes. Pada bagian *header* masih terdapat bagian *ver* atau *version*, *traffic class*, *flow label*, *payload length*, *next header* dan *hop limit*. Bagian – bagian IPv6 *header* dapat ditunjukkan pada Gambar 2.2.



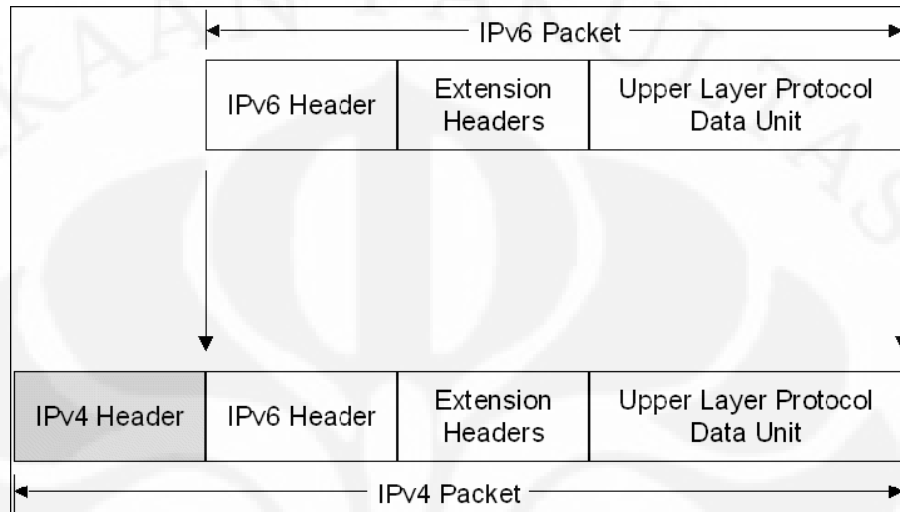
Gambar 2.1 Struktur packet IPv6



Gambar 2.2 Format Header IPv6 [1]

Pada Gambar 2.1 *Header* IPv6 memiliki ukuran tetap sebesar 40 bytes. Haeder ini merupakan penyederhanaan dari header IPv4 dengan menghilangkan bagian yang tidak diperlukan atau jarang digunakan agar mengurangi beban kerja router dan menambahkan bagian yang memberikan dukungan yang lebih baik. Sebagian besar dari penambahan tersebut adalah *traffic real time* seperti Gambar 2.3 dibawah. Sedangkan pada bagian *payload* yang terdiri dari *extension headers* dan *upper layer protocol data unit* terdapat

header tambahan yaitu *Optional Extension Headers* dan *ICMPv6 message*. Pada Gambar 2.2 source IP address dan destination IP address memiliki kapasitas sebesar 128 bit.



Gambar 2.3 Perbandingan packet IPv6 dan packet IPv4 [2].

Berikut ini adalah penjelasan untuk masing – masing bagian dari paket IPv6 :

Extension headers merupakan Header dan extension header pada IPv6 ini menggantikan header dan option pada IPv4.

Upper Layer Protocol Data Unit atau Protokol Data Unit (PDU) dari layer yang lebih tinggi (upper layer) memiliki header protokol layer yang lebih tinggi dan payload yang terkandung di dalamnya misalnya saja TCP, UDP atau ICMPv6.

Sedangkan pada Gambar 2.2 menggambarkan field – field pada header IPv6. Penjelasan adalah sebagai berikut :

IPVER atau IP version : merupakan 4 bit field yang menunjukkan versi suatu internet protokol IP, yaitu 6.

Traffic Class : merupakan 4 bit field yang menunjukkan prioritas pada paket. Field ini memungkinkan pengirim paket mengidentifikasi prioritas yang diinginkan untuk paket yang dikirimkan, relatif terhadap paket-paket lain dari pengirim yang sama.

Flow Label : merupakan 24 bit *field* yang digunakan oleh pengirim untuk memberi label pada paket-paket yang membutuhkan penanganan khusus dari router IPv6, seperti quality of service yang bukan default, misalnya service-service yang bersifat real-time.

Payload Length : merupakan 16 bit field yang menunjukkan panjang payload atau sisa paket yang mengikuti header IPv6

Next Header : merupakan 8 bit field yang menunjukkan *extension header* pada paket IPv6.

Hop Limit : merupakan 8 bit field yang menunjukkan jumlah link maksimum yang akan dilewati paket sebelum dibuang. Paket akan dibuang bila Hop Limit berharga nol.

Source Address : merupakan 128 bit field yang menunjukkan alamat pengirim dari paket IPv6 yang telah dikirimkan.

Destination Address : merupakan 128 bit field yang menunjukkan alamat penerima paket.

Optional Extension Headers : Merupakan header tambahan yang berfungsi sebagai informasi tambahan.

2.2.2 Pengalamatan IPv6

2.2.2.1 Format Alamat IPv6

Dalam IPv6, alamat 128-bit akan dibagi ke dalam 8 blok berukuran 16-bit, yang dapat dikonversikan ke dalam bilangan heksadesimal yang berukuran 4-digit antara 0x0000 sampai dengan 0xffff. Setiap blok bilangan heksadesimal tersebut akan dipisahkan dengan tanda titik dua (:). Sehingga format notasi yang digunakan oleh IPv6 juga sering disebut dengan *colon-hexadecimal format*, berbeda dengan IPv4 yang menggunakan *dotted-decimal format*.

Berikut ini adalah contoh alamat IPv6 dalam bentuk bilangan biner :

```
00100001110110100000000011010011000000000000000001011110011101
1000000101010101000000000111111111111100010100010011100010110
10
```

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

001000011101101010000000011010011100000000000000010010111100111
 0111000000101010101010000000011111111111111110001010001001110001
 011010.

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

21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A. [10]

2.2.2.2 Penyederhanaan Bentuk Alamat

Alamat 21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A ini juga dapat disederhanakan lagi dengan membuang angka 0 pada awal setiap blok yang berukuran 16-bit di atas, dengan menyisakan satu digit terakhir. Dengan membuang angka 0, alamat di atas dapat disederhanakan menjadi sebagai berikut : 21DA:D3:0:2F3B:2AA:FF:FE28:9C5A. Konversi pengalamatan pada IPv6 juga dapat menyederhanakan pengalamatan dengan cara membuang lebih banyak karakter 0 pada sebuah alamat yang memiliki karakter 0 yang sangat banyak. Misalnya alamat 2001:0db8:3c4d:005a:0000:0000:0000:1 dapat disederhanakan sebagai berikut 2001:db8:3c4d5a:0:0:0:1 atau 2001:db8:3c4d:5a::1.

2.2.2.3 Format Prefiks

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*. Dalam IPv4, sebuah alamat dalam notasi *dotted-decimal* format dapat ditunjukkan dengan menggunakan angka prefiks yang digunakan pada subnet mask. IPv6 juga memiliki angka prefiks, tapi tidak digunakan untuk subnet mask, karena memang IPv6 tidak mendukung subnet mask. Prefiks dalam IPv6 direpresentasikan dengan cara yang sama seperti halnya prefiks alamat IPv4, yaitu [alamat]/[angka panjang prefiks]. Panjang prefiks menentukan jumlah bit terbesar paling kiri yang membuat prefiks subnet.

Sebagai contoh, prefiks sebuah alamat IPv6 adalah 3FFE:2900:D005:F28B::/64. Pada alamat tersebut /64 merupakan alamat prefiks.

2.2.2.4 Jenis – Jenis Alamat IPv6

Alamat IPv6 ini dapat diklasifikasikan menjadi 3, yaitu :

1. Alamat *Unicast*

Alamat unicast ini digunakan pada komunikasi yang bersifat *single interface* atau disebut juga *one to one communication* atau *point to point* antara dua *host* dalam sebuah jaringan. Pada *unicast address*, ditetapkan *address* yang bersifat global seperti *address* untuk provider, *address* geografis, *link local address*, dan *site local address*.

Link local address merupakan alamat yang digunakan pada satu link saja. *Link* merupakan jaringan lokal yang saling terhubung pada satu level. *Link local address* digunakan pada pemberian alamat IP secara otomatis.

Site local address sama seperti *private address*. Alamat dipakai hanya didalam *site*. Alamat ini dapat diberikan bebas, yang penting alamat yang digunakan harus unik.

2. Alamat *Anycast*

Alamat Anycast ini mengidentifikasi komunikasi *multiple interface*. Alamat ini digunakan pada komunikasi *one to one of many communication*. Alamat anycast menyampaikan paket data hanya kepada alamat terdekat (memiliki route yang dekat atau jalur terbaik) sesuai dengan pengkonfigurasinya.

3. Alamat *Multicast*

Alamat *multicast* ini digunakan pada komunikasi *one to many communication*. Alamat *multicast* ini akan mengirimkan sebuah paket data ke banyak *host* yang berada pada jaringan yang sama. Akan tetapi alamat ini hanya akan muncul sebagai alamat tujuan bukan alamat awal atau asalnya.

2.2.2.5 Perbedaan IPv6 dengan IPv4

Perbedaan antara IPv6 dan IPv4 adalah pada Tabel 2.1:

Tabel 2.1 Perbedaan IPv6 dan IPv4

IPv6	IPv4
Kapasitas alamat 128 bit	Kapasitas alamat 32 bit.
Memiliki kemampuan autonumbering	IPv4 tidak memiliki kemampuan autonumbering (penomoran kembali alamat IP secara otomatis ketika mengalami gangguan).
Tidak bisa melakukan NAT karena pada IPv6 pengalamatannya lebih dari IPv4 dan memiliki host yang sangat banyak maka NAT tidak efisien	Bisa melakukan NAT (<i>Network Address Translation</i>). Pada IPv4, NAT akan mengirimkan paket keseluruhan <i>host</i> . Karena pengalamatan IPv4 terbatas, maka <i>host</i> nya juga terbatas, sehingga kegunaan NAT pada IPv4 tidak dipermasalahkan.
Alamat IP terbagi menjadi 3 jenis, yaitu <i>unicast address</i> , <i>multicast address</i> dan <i>anycast address</i>	Alamat IP terbagi menjadi 3 jenis, yaitu <i>unicast address</i> , <i>broadcast</i> dan <i>multicast address</i> .
IP header pada IPv6 terdiri dari version, traffic class, flow lable, payload,length, next header, hop limit, source address, destination address.	IP header IPv4 terdiri dari version, IHL, Type of service, Total Length, Identification, Flags, Fragment Offset, Time to Live, Protokol, Header checksum, Source <i>address</i> , Destination <i>address</i> , Option, Padding.
source address (alamat sumber) dan destination (alamat tujuan) sebesar 128 bit.	Alamat sumber (<i>source address</i>) dan alamat tujuan (<i>destination address</i>) sebesar 32 bit.

Kelebihan – kelebihan yang dimiliki IPv6 dibandingkan IPv4 adalah sebagai berikut :

1. Kemampuan pengalamatan yang lebih banyak dari IPv4 (sudah diterangkan pada tulisan sebelumnya).
2. Autokonfigurasi : mampu mengatur pengalamatan (konfigurasi) secara otomatis pada DHCP pada sisi *server*. Sedangkan pada IPv4 memiliki *dynamic address* dan *static address* yang dikonfigurasi secara manual. IPv6 mengkonfigurasi alamat dengan menggunakan DHCP *server* yang dinamakan *stateful address configuration*, sedangkan apabila ada alamat IPv6 yang tidak memiliki DHCP *server* dinamakan *stateless address configuration*.

3. Keamanan : kewananaan yang dimiliki IPv6 lebih baik dari IPv4 karena mendukung IPSec (IP security).

4. *Quality of Service* : Field baru yang ada pada header IPv6 menunjukkan bagaimana suatu *traffic* khusus diidentifikasi untuk ditangani. Penanganan khusus tersebut adalah dengan menggunakan *flow label* sehingga yang butuh penanganan dapat teridentifikasi.

2.2.3 Fitur – fitur IPv6

IPv6 memiliki fitur – fitur yang memperbarui IPv4. Fitur – fitur tersebut adalah sebagai berikut :

1. Format Header Baru

IPv6 memiliki header yang efisien dari pada IPv4 karena header IPv6 diperoleh dengan menghilangkan beberapa bagian yang tidak penting atau opsional.

2. Pengalamatan yang lebih besar

IPv6 memiliki jumlah alamat 2^{128} yang lebih banyak dari IPv4 yang dapat memenuhi kebutuhan lebih lama atau masih dapat digunakan dimasa mendatang.

3. Keamanan yang Built-in

Pada IPv6, IPsec menjadi spesifikasi standar dan sudah secara langsung diamankan pada layer network. Sangat berbeda dengan IPv4 yang memiliki IPsec bersifat opsional.

4. Dukungan QoS lebih bagus

Pada bagian header IPv6 yang memiliki bagian atau *field* baru yang mengidentifikasi trafik (*Flow lable*) dan *Traffic Class* untuk prioritas traffic. Kedua *field* ini menghasilkan QoS yang lebih baik dan tejamin.

5. Ekstensibilitas

Fitur – fitur IPv6 masih dapat dikembangkan lagi dengan cara menambahkan pada extension header.

2.3 MOBILE IPv6

2.3.1 Mobile IP

Mobile IP merupakan suatu standart yang dirancang oleh *Internet Engineering Task Force (IETF)*. *Mobile IP* berkerja pada *network layer* yaitu pada *network layer 3* yang memiliki beberapa karakteristik yang saling berhubungan dalam mendukung *mobile node*. *Mobile IP* memiliki karakteristik *mobile* atau dapat bergerak kemanapun selama dapat terhubung dengan jaringan tanpa harus terhubung secara fisik. *Mobile IP* memiliki keamanan yang sudah didukung oleh *mobile IP* sendiri. Kemampuan *mobile IP* ini didukung dengan adanya *mobile node (MN)*, *correspondent node (CN)*, *home agent*, *foreign agent*, akses point dan *care-of-address (COA)*.

2.3.2 Mobile IPv6

Sama halnya dengan *mobile IP*, *mobile IPv6* juga merupakan suatu standart yang dirancang oleh *Internet Engineering Task Force (IETF)*. *Mobile IPv6* dirumuskan dalam RCF (*Request For Comment*). *Mobile IPv6* sama dengan *mobile IP*. Hanya saja protokol yang digunakan adalah IPv6. *Mobile IPv6* ini dikembangkan untuk menggantikan *mobile IPv4*. *Mobile IPv6* memungkinkan suatu *host* tetap terkoneksi ke suatu jaringan meskipun *host* tersebut berpindah – pindah tempat atau berpindah dari suatu *subnet* ke *subnet* lain.

Sama seperti *mobile IP* yang lain, *mobile IPv6* terdapat di *network layer 3*. Ketika berpindahnya *host* dari suatu *subnet* ke *subnet* yang lain, layer – layer yang lain di atas *network layer* tidak perlu mengetahuinya.

Pada *mobile IPv6*, perangkat yang mendukung adalah *mobile node* yang memiliki dua alamat yaitu *home address* adalah alamat unik yang diberikan pada *home link*-nya dan *care-of-address* adalah alamat yang didapat ketika berpindah *link*. *Care-of-address* ini digunakan sebagai alamat pengganti dari *home address* ketika perangkat *mobile* yaitu *mobile node* berada di luar *home link*. Kedua alamat tersebut penggunaannya diatur oleh *Home Agent* dan *Foreign Agent*.

2.3.2.1 Mekanisme *Mobile IPv6*

Mobile node merupakan *host* yang dapat bergerak *mobile* atau berpindah dari *link* satu ke *link* yang lain dalam suatu jaringan. *Mobile node* ini dapat berupa laptop dan menggunakan wireless LAN agar dapat terkoneksi secara *mobile*. *Mobile node* ini memiliki dua alamat agar tetap dapat terhubung dengan jaringan *mobile IPv6*. Alamat yang digunakan *mobile node* adalah *home address* dan *care-of-address*.

Home address merupakan alamat IP awal yang diberikan untuk *mobile node* sebagai IP yang unik pada saat *mobile node* berada pada *home link* (jaringan asal). Sedangkan ketika *mobile node* berpindah link ke *foreign link* atau diluar *home link*, *home address* ini digunakan sebagai alamat *host* lain apabila ingin terkoneksi pada jaringan dan tidak memiliki *care-of-address*.

Care-of-address merupakan alamat IP yang didapatkan oleh *mobile node* ketika *mobile node* berada di *foreign link*. *Mobile node* akan mendapatkan *care-of-address* dengan cara mekanisme pengalamatan IPv6. Mekanisme yang digunakan adalah *stateless auto-configuration* (mekanisme *router advertisement*) atau *stateful auto-configuration* (melalui DHCPv6).

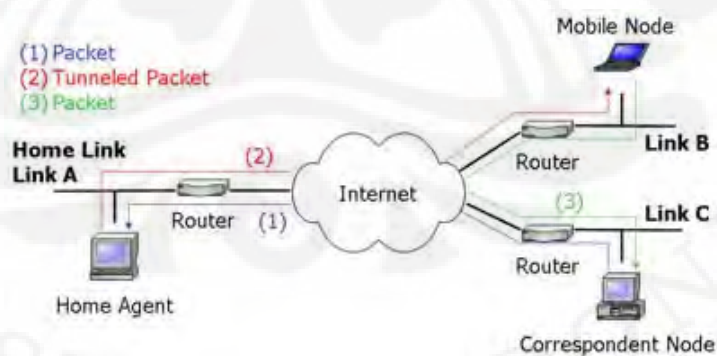
Binding merupakan kombinasi dari *home address* dan *care-of-address* yang dimiliki oleh *mobile node*. Ketika *mobile node* berada pada *foreign link*, *mobile node* memberitahukan *care-of-address* terbaru yang didapat kepada *home agent* yang berfungsi seperti *router* yang ada pada *home link*. Kemampuan *home agent* berbeda dengan *router – router advertisement* yang hanya berfungsi seperti *router* biasa. Proses dimana *mobile node* memberitahukan *care-of-address* terbaru dinamakan *binding update*. Setiap *mobile node* mengirimkan *care-of-address* terbarunya, *home agent* akan selalu mengetahuinya. Ketika *home agent* menerima informasi mengenai *care-of-address* terbaru, *home agent* akan memberikan balasan berupa *binding acknowledgement* ke *mobile node*.

Coresspondent node merupakan *node* atau *host* yang melakukan koneksi terhadap *mobile node*. *Coresspondent node* ini dapat berupa *host* yang *mobile* atau *host* yang bersifat statis atau tidak bergerak. Pada skripsi ini, *coresspondent node* berupa *host* yang statis. Setiap *mobile node* mengalami perpindahan link, *mobile node* akan memberitahukan kedudukan terbarunya pada *coresspondent node*

dengan menggunakan *coresspondent registration*. Proses registrasi tersebut akan dilakukan melalui *return routability procedure*. *Return routability procedure* merupakan prosedur yang dilakukan agar *correspondent node* dapat memastikan bahwa *mobile node* benar – benar dialamatkan pada *care-of-address* dan *home agent* yang sesuai.

1. Bidirectional Tunneling

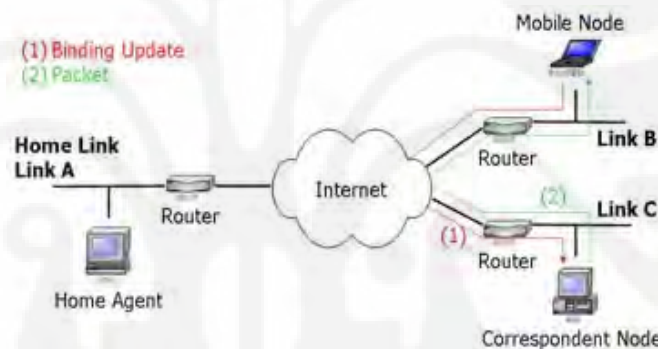
Bidirectional Tunneling merupakan salah satu cara komunikasi *mobile node* dengan *correspondent node* ketika berada diluar *home network* (*foreign network*) dalam jaringan *mobile IPv6*. Apabila *correspondent node* tidak memiliki kemampuan *mobile IPv6* atau memiliki kemampuan *mobile* tetapi saat melakukan *binding registration* untuk *mobile node* belum sempurna, *mobile node* dapat terkoneksi dengan *correspondent node* tanpa harus melakukan *correspondent registration* melalui *routability procedure*. Jadi paket – paket data dari *correspondent node* ke *mobile node* dikirim melalui *home agent* dan dialamatkan menggunakan *home address*. Paket – paket tersebut dirutekan ke *home agent* dan disampaikan ke *mobile node* dengan menggunakan *tunneling*. Sedangkan paket data yang dikirim ke *correspondent node* dari *mobile node* akan ditunnel melalui *home agent* (*reverse tunneling*) dan dirutekan secara normal dari *home network* ke *correspondent node*. Mekanisme bidirectional tunneling ditunjukkan pada Gambar 2.4.



Gambar 2.4 Mekanisme bidirectional tunneling [3]

2. Route Optimization

Route Optimization merupakan salah satu cara komunikasi *mobile node* dengan *correspondent node* ketika berada diluar *home network (foreign network)* dalam jaringan *mobile IPv6*. Dengan menggunakan cara *Route Optimization* ini, *mobile node* akan dapat langsung berkomunikasi dengan *correspondent node* dengan melakukan *correspondent registration*. Paket – paket data dapat dikirim dari *correspondent node* ke *mobile node* dengan menggunakan *care-of-address*. Dengan menggunakan *Route Optimization*, jalur pengiriman paket data dari *correspondent node* ke *mobile node* yang akan digunakan adalah jalur komunikasi terpendek. Dengan jalur komunikasi singkat ini dapat mengurangi terjadinya *overload* pada *home agent* dan mengurangi *congestion* pada *home link*. Mekanisme *Route Optimization* pada *mobile IPv6* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Mekanisme *Route Optimization* [4]

2.3.2.2 Perbedaan Mobile IPv6 dan Mobile IPv4

Perbedaan *mobile IPv6* dan *IPv4 mobile* adalah sebagai berikut :

1. Pengalamatan IPv6 untuk *mobile IPv6* tentunya sangat banyak. Berbeda dengan *mobile IPv4* yang memiliki pengalamatan yang terbatas.
2. Pada *mobile IPv4* memerlukan router khusus yang terdapat pada *foreign link*. *Router* ini bertindak sebagai *foreign agent*. Sedangkan *mobile IPv6* tidak memerlukan *router* khusus dan dapat beroperasi dilokasi atau *link* manapun.
3. *Route Optimization* pada *mobile IPv6* dapat berlangsung secara aman tanpa melakukan proses pengamanan terlebih dahulu terhadap jalur komunikasi.
4. Paket – paket yang dikirimkan ke *mobile node* oleh *home agent* ataupun *correspondent node* dijalurkan dengan menggunakan IPv6 *routing header* dari

pada enkapsulasi IPv6, sehingga dapat mengurangi overload pada pemrosesan paket. Lain halnya dengan mobile IPv4 yang pengiriman pakatnya menggunakan enkapsulasi IP.

2.3.2.3 Proses *Handover* pada Mobile IPv6

Proses *handover* pada jaringan mobile IPv6 dilakukan secara manual atau disebut dengan *hard handover*. *Hard handover* merupakan proses *handover* yang terjadi ketika terminal *mobile node* melakukan pemutusan koneksi secara mendadak dengan jaringan *home link* (jaringan asalnya) sebelum terkoneksi dengan jaringan yang baru (*foreign link*).

Movement detection merupakan proses pendeteksian ketika terjadinya *handover* pada mobile IPv6. Ketika *mobile node* mendeteksi, *mobile node* akan memilih *default router* baru dengan menggunakan metode *router discovery* dan melakukan pemeriksaan pada *link* baru untuk mencegah adanya *link local address* yang sama. Setelah itu *mobile node* akan melakukan *prefix discovery* dengan *default router* yang baru untuk mendapatkan *care-of-address* yang baru yang kemudian *mobile node* akan melakukan proses *binding update* ke *home agent*.

Ketika mendeteksi router advertisement untuk melakukan proses *handover* atau tidak, *mobile node* perlu mempertimbangkan hal – hal sebagai berikut :

1. Adanya kemungkinan terdapat banyak router pada link yang sama. Saat *mobile node* mendeteksi *router* yang baru pada link yang sama maka *mobile node* belum tentu mengalami proses *handover* (berpindah link).
2. Adanya banyak router dalam suatu link memungkinkan router – router tersebut memiliki prefix yang berbeda sehingga perubahan prefix berdasarkan perubahan router tidak berarti terjadi perpindahan link.
3. *Link-local address* yang dimiliki *router* tidak unik membuat *mobile node* masih mendapatkan *router advertisement* dari *link-local address* yang sama ketika sudah melakukan proses *handover*.

Handover terjadi tidak hanya pada saat *mobile node* berpindah *link* ke *foreign link*, akan tetapi *handover* terjadi ketika *mobile node* kembali ke *link* awalnya yaitu *home link*. Proses kembalinya *mobile node* ke *home link* disebut dengan proses *returning home*. Proses *returning home* ini terjadi ketika *mobile*

node mendeteksi subnet prefix dari *home link*-nya yang kemudian akan mengirimkan *binding update* ke *home agent*. Saat *mobile node* berada di *home link*, *mobile node* akan mengubah *care-of-address* miliknya menjadi *home address* dan mengirimkan pesan pada *home agent* dan *correspondent node* untuk menghapus *binding cache* yang berisi *care-of-address* milik *mobile node* sebelumnya.

2.4 FILE TRANSFER PROTOCOL (FTP)

File Transfer protocol (FTP) adalah suatu protokol yang berfungsi untuk transfer file (*mendownload* dan *mengupload*) antar *host* dalam suatu jaringan yang berbasis TCP/IP. Dua hal yang terpenting untuk transfer file dalam FTP adalah FTP *server* (menjalankan suatu sistem diatas komputer yang dapat merespon perintah – perintah yang diberikan oleh FTP *client* dalam melakukan transfer file) dan FTP *client* (aplikasi dalam komputer yang memberikan perintah atau merequest file dari FTP *server*). *Client* dan *server* pada FTP dapat saling bertukar file dan saling terkoneksi dengan menggunakan Protokol TCP (*Transmission Control Protocol*) yang merupakan suatu protokol yang berada pada layer transport. TCP dipakai sebagai protokol transport pada FTP karena protokol TCP membuat pengiriman FTP agar memungkinkan *user* mengakses file dan direktori secara interaktif. FTP *client* dapat mengakses daftar file pada direktori remote dan lokal, mengganti nama dan menghapus file, *download* (transfer file dari *host* remote ke lokal), *upload* (transfer file dari *host* lokal ke remote). Mekanisme transfer file dari *host* lokal ke remote atau *upload* dilakukan ketika *client* memasuki jaringan TCP/IP kemudian komputer remote akan ditunjukan ke *host* FTP yang harus memiliki aplikasi FTP *server* yang telah di-*install* agar dapat terhubung dengan sistem file yang berada pada *host*.

Metode yang digunakan pada FTP adalah metode autentikasi standar yang hanya menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi, sehingga *user* atau pengguna yang sudah terdaftar dapat menggunakan *password* dan *username* untuk dapat mengakses beberapa direktori dan membuat file, menambah dan menghapus file, membuat direktori serta *men-download* dan *men-upload* file. Sedangkan untuk pengguna atau *user* yang belum terdaftar

dapat menggunakan metode *anonymous* login hanya dengan menggunakan nama pengguna *anonymous* dan *password* yang digunakan adalah alamat e-mail. Dengan metode autentikasi standar ini, pengiriman data atau transfer file tidak aman karena hanya melalui *clear text* tanpa enkripsi data. Metode text yang digunakan dalam mentransfer file adalah dengan format ASCII atau menggunakan format Binary. Secara default, FTP menggunakan mode ASCII dalam mentransfer file. Karena tidak menggunakan enkripsi dalam pengirimannya, maka *username*, *password*, file yang ditransfer dan perintah – perintah dikirim dapat di *sniffing* yaitu dapat melihat jalur atau lalu lintas data pada suatu jaringan dan menangkap data atau paket untuk menguraikan isi RFC (*request for comments*) dan spesifikasi lainnya dengan menggunakan protokol analyzer (*sniffer*). Hal tersebut dapat ditangani dengan SFTP (SSH FTP) yang merupakan FTP berbasis pada SSH (transfer file atau data secara aman antara dua perangkat jaringan dalam suatu jaringan) atau menggunakan FTPS (FTP over SSL) yang menggunakan enkripsi data sebelum dikirim.

FTP digunakan dalam proses transfer file melalui jaringan TCP/IP. Pada FTP, port yang biasa digunakan adalah port 20 dan port 21. Port 20 digunakan untuk transfer data antara client dan *server*. Pada port FTP *server* terdapat koneksi aktif mode dan pasif mode. Sedangkan port 21 (FTP *server Listen*) yang digunakan untuk *incoming connection* atau penerimaan koneksi dari FTP *client* (*request*) digunakan untuk *command port*. Pada Gambar 2.6, Sebelum melakukan transfer file dari *client* ke *server* menggunakan protokol TCP, pada FTP akan dilakukan pengkoneksian terlebih dahulu. Pada port 21, *server* akan mendengarkan uji koneksi yang dikirim oleh FTP *client* yang kemudian akan digunakan sebagai port pengatur atau *control port* yang berfungsi untuk menghubungkan antara *client* dan *server*, memperbolehkan *client* untuk mengirimkan perintah FTP kepada *server* dan mengembalikan *respon server* ke perintah tersebut. Saat koneksi control telah dibuat, maka port 20 pada *server* akan membuka koneksi baru dengan *client* untuk mentransfer file yaitu pada saat *upload* dan *download*. [5] Proses kerja FTP ditunjukkan pada Gambar 2.6.



Gambar2.6 Proses kerja FTP [5]

Dalam penggunaan FTP, ada dua cara agar dapat melakukan transfer file atau *download* yaitu sebagai *user* yang memiliki *authentication* atau sebagai *user* yang login sebagai *Anonymous*.

2.4.1 *Anonymous*

FTP yang menggunakan *user* anonymous dibuat agar setiap orang yang ingin terkoneksi melalui internet maupun jaringan lokal dapat saling berbagi file tanpa harus melakukan proses autentikasi. Untuk terkoneksi dengan *server* dan melakukan *upload* maupun *download* *user* *anonymous* dapat menggunakan *account* yang digunakan secara umum (*public account*) seperti menggunakan *username* sebagai *anonymous* tanpa memasukkan *password*.

Dengan penggunaan *user* anonymous yaitu dengan *public account*, hak yang dimiliki *user* sangat terbatas kepada aturan – aturan yang dimiliki oleh pemilik *server*. Batasan akses direktori dan file yang tidak diperbolehkan biasanya pengguna atau *user* hanya dapat *men-download*, membaca file tertentu dan tidak diperbolehkan untuk mengedit file, tidak dapat melakukan *uploading data* ke *server* dan memindahkan file ke direktori lain maupun berpindah – pindah direktori akses yang tidak diizinkan oleh pemilik *server*.

2.4.2 *Authentication User*

FTP yang menggunakan *authentication user* merupakan cara lain untuk dapat terkoneksi dengan FTP *server* melalui internet maupun jaringan lokal dengan menggunakan *account* yang sudah teregistrasi oleh pemilik *server*.

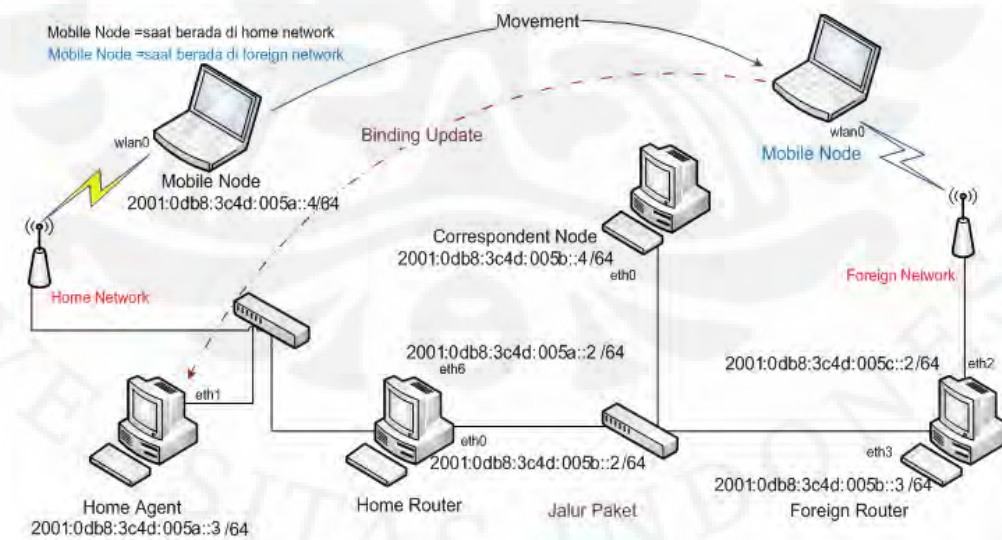
Dengan adanya *account* saat mengakses FTP *server*, seorang pengguna memiliki hak akses yang berbeda dengan *user anonymous*. *User* yang memiliki *account* dapat mengakses file direktori *server* yang dituju dan dapat melakukan *download*, berpindah pindah direktori, membaca file tertentu, *uploading* dan membuat sebuah file direktori.

BAB 3

PEMBANGUNAN JARINGAN DAN METODE PENGAMBILAN DATA

3.1 TOPOLOGI JARINGAN

Skripsi ini menggunakan jaringan berskala kecil yang digunakan sebagai uji coba. Topologi jaringan *mobile IPv6* terdiri dari empat PC yang masing-masing memiliki fungsi dan konfigurasi berbeda dan satu laptop sebagai *mobile node*. Keempat PC tersebut antara lain sebagai *home agent*, *correspondent node*, *home router* dan *foreign router*. Sistem operasi yang digunakan dalam membuat jaringan *mobile IPv6* ini adalah Linux Ubuntu 8.04. Pada sisi *server* yaitu pada *correspondent node* di-install *FTP server*. *FTP server* yang digunakan adalah yang dapat mendukung IPv6 dan sistem Operasi linux Ubuntu yaitu VSFTPD (Very Secure FTP Daemon). Sedangkan pada sisi *client* menggunakan *browser* untuk mengakses *FTP server*. Topologi jaringan *mobile IPv6* adalah seperti pada Gambar 3.1 berikut :



Gambar 3. 1 Jaringan *Mobile IPv6*

Jaringan *mobile IPv6* yang dibuat untuk analisa performansi FTP ini merupakan jaringan IPv6 murni. Dari Gambar 3.1 komponen – komponen jaringan *mobile IPv6* terdiri dari :

1. *Mobile Node*

Mobile Node merupakan suatu *host* atau *node* yang bergerak secara *mobile* atau berpindah – pindah dari *subnet* ke *subnet* yang lain. Dari topologi diatas, *mobile node* adalah sebuah laptop yang dapat bergerak *mobile*.

Spesifikasi perangkat keras pada *Mobile node* dapat dilihat pada Tabel 3.1.

Tabel 3. 1 Spesifikasi Mobile node

No.	Hardware	Spesifikasi
1.	Jenis	Laptop Toshiba M200
2.	Prosesor	Intel Core 2 Duo
3.	Memori	DDR2 2 GB
4.	Hardisk	120 GB

2. *Correspondent Node*

Correspondent Node merupakan *host* yang berhubungan dengan *host mobile node*. *Host mobile node* dapat bergerak atau *mobile*, sedangkan *host correspondent node* merupakan *host* yang diam atau statis akan tetapi bisa menjadi *host* yang bergerak.

Spesifikasi perangkat keras pada *Correspondent Node* dapat dilihat pada Tabel 3.2.

Tabel 3. 2 Spesifikasi Correspondent Node

No.	Hardware	Spesifikasi
1.	Jenis	Desktop PC
2.	Prosesor	AMD Athlon XP
3.	Memori	2 @ DDR PC2100 256 MB
4.	Hardisk	Maxtor 40 GB

3. *Home Agent*

Home agent merupakan sebuah *router* yang berada pada *home link*. *Home agent* ini memiliki fungsi sebagai pemantau gerakan *mobile node* dan

mencatat *care-of-address* yang baru dari *mobile node* setiap kali *mobile node* menginformasikannya kepada *home agent*.

Spesifikasi perangkat keras pada *Home Agent* dapat dilihat pada tabel sebagai berikut :

Tabel 3. 3 Spesifikasi Home Agent

No.	Hardware	Spesifikasi
1.	Jenis	Desktop PC
2.	Prosesor	AMD Athlon XP
3.	Memori	2 @ DDR PC2100 256 MB
4.	Hardisk	Maxtor 40 GB

4. *Home Router*

Home router merupakan *router* yang sangat berguna dalam proses penjaluran paket – paket IP. *Home router* ini adalah *router* yang terdapat pada *home link*.

5. *Foreign Router*

Foreign router merupakan *access router* yang terdapat pada *foreign link*. *Access router* ini berbeda dengan *router* pada *home router*, *access router* ini memiliki mekanisme *advertisement* yang digunakan *mobile node* untuk mendeteksi jaringan yang sedang ditempati, biasanya bila *mobile node* tidak berada di *home link*.

6. *Akses Point*

Akses point merupakan sebuah perangkat komunikasi wireless atau nirkabel yang memungkinkan antar perangkat terhubung secara nirkabel dengan menggunakan wireless LAN. *Akses point* ini dapat berfungsi sebagai hub atau switch yang menghubungkan jaringan lokal dengan jaringan wireless. *Akses point* ini menghubungkan *mobile node* dengan jaringan IPv6 *mobile*.

Spesifikasi perangkat keras pada *Akses Point* dapat dilihat pada tabel sebagai berikut :

Tabel 3. 4 Spesifikasi Akses Point 1

No.	Hardware	Spesifikasi
1.	Jenis	Linksys Router WRVS4400N
2.	Port	4 port 10/100/1000

Tabel 3. 5 Spesifikasi Akses Point 2

No.	Hardware	Spesifikasi
1.	Jenis	TP-LINK WR941ND
2.	Port	4 port 10/100

7. Switch

Spesifikasi perangkat keras pada *Switch* dapat dilihat pada tabel sebagai berikut :

Tabel 3. 6 Spesifikasi *Switch*

No.	Hardware	Spesifikasi
1.	Jenis	Linksys SD208
2.	Port	8 port 10/100

3.2 PERANGKAT LUNAK YANG DIGUNAKAN

Perangkat lunak yang digunakan dalam membuat jaringan *mobile IPv6* ini adalah sebagai berikut :

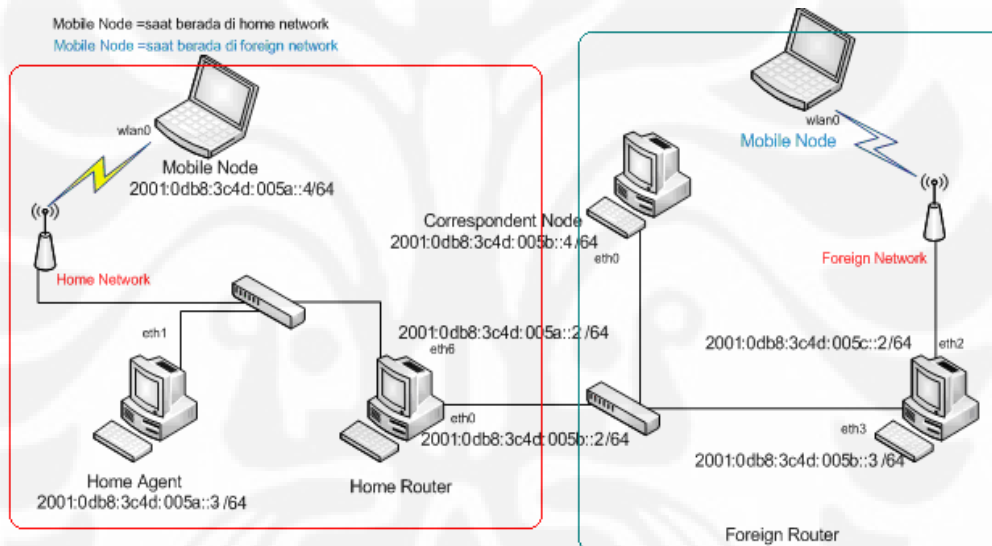
1. Sistem Operasi Linux Ubuntu 8.04 dengan kernel 2.6.22.14-mip6 yang mendukung operasi Mobile IPv6. Sistem operasi ini di-*install* pada setiap komponen jaringan, yaitu pada *home agent*, *correspondent node*, *home router*, *foreign router* dan *mobile node*.
2. VSFTPD merupakan aplikasi FTP *server* yang di-*install* pada *correspondent node*. VSFTPD ini merupakan aplikasi yang akan diujikan dan digunakan untuk *file transfer*.
3. Wireshark merupakan *tools* aplikasi yang digunakan dalam pengukuran parameter – parameter kualitas layanan berupa *throughput*, *transfer time* dan *delay*. Wireshark ini di- *install* pada *mobile node*.
4. Bind9

Bind9 merupakan sebuah aplikasi DNS server. DNS server ini yang nantinya akan digunakan untuk me-resolv sebuah nama domain ke dalam alamat IPv6. Agar sistem *Mobile IPv6* dapat berjalan pada jaringan uji coba, maka sistem operasi linux ubuntu pada bagian */etc/apt/source.list* ditambahkan :

```
deb http://software.nautilus6.org/packages/ubuntu gutsy/  
deb-src http://software.nautilus6.org/packages/ubuntu gutsy/
```

3.3 KONFIGURASI JARINGAN

Setelah topologi jaringan dan aplikasi – aplikasi yang dibutuhkan sudah ter-install kemudian setiap komponen pada jaringan tersebut dikonfigurasi agar dapat dijalankan sebagai jaringan *mobile IPv6*.



Gambar 3. 2 Konfigurasi Jaringan *Mobile IPv6*

Dari Gambar 3.2, jaringan *mobile IPv6* ini terdiri dari *home link* dan *foreign link*. *Foreign link* ditandai dengan adanya *foreign router* dan *home link* ditandai dengan adanya *home router*. Konfigurasi pada tiap – tiap komponen yang digunakan dalam *mobile IPv6* adalah sebagai berikut :

1. Konfigurasi *Home Agent*

Home agent pada jaringan *mobile IPv6* ini memiliki alamat 2001:0db8:3c4d:005a::3. *Home agent* pada jaringan ini adalah komputer *desktop* yang dihubungkan dengan akses *point*.

2. Konfigurasi *Correspondent Node*

Correspondent node yang terdapat pada jaringan *mobile IPv6* ini dapat diletakkan pada *home link* maupun *foreign link*. Pada jaringan ini, *correspondent node* diletakkan pada *foreign link*. *Correspondent node* memiliki alamat 2001:0db8:3c4d:005b::4. *Correspondent node* pada jaringan berupa komputer *desktop* biasa yang dihubungkan pada *switch* antara *home router* dan *foreign router*. Aplikasi *FTP server* akan di-install di *correspondent node*. *FTP server* yang digunakan adalah *VSFTPD server*.

3. Konfigurasi *Home Router*

Home router merupakan router yang terapat pada *home link*. Pada *home router* memiliki dua alamat yaitu 2001:0db8:3c4d:005a::2 sebagai *wireless interface* yang terhubung ke akses *point* dan 2001:0db8:3c4d:005b::2 sebagai *wired interface*.

4. Konfigurasi *Foreign Router*

Foreign router pada jaringan *mobile IPv6* ini merupakan *access router* yang terdapat di *foreign link*. Sama seperti konfigurasi *home router*, pada *foreign router* tersebut memiliki dua alamat yaitu 2001:0db8:3c4d:005b::3 sebagai *wired interface* dan 2001:0db8:3c4d:005c::2 sebagai *wireless interface* yang terhubung ke akses *point*.

5. Konfigurasi *Mobile Node*

Mobile node merupakan *host* yang bergerak *mobile* yang difungsikan sebagai *client* dan berupa sebuah laptop karena memiliki dukungan *interface* berupa *wireless*. Pada *mobile node*, *interface* yang digunakan adalah hanya *wireless interface*. *Home address* yang diberikan pada *mobile node* adalah 2001:0db8:3c4d:005a::4. *Mobile node* akan mendapatkan *care-of-address* apabila *mobile node* berpindah tempat atau link. *Mobile node* yang berfungsi sebagai *client* tidak perlu menginstal aplikasi lain lagi karena untuk mendownload file dari *FTP server*, *mobile node* hanya perlu menggunakan *browser* untuk mengakses.

3.4 APLIKASI UJI

Aplikasi uji merupakan aplikasi yang diimplementasikan pada jaringan mobile IPv6. Aplikasi uji yang digunakan adalah VSFTPD yang merupakan aplikasi dari FTP (*File Transfer Protocol*).

3.4.1 Aplikasi VSFTPD

VSFTPD merupakan kepanjangan dari *Very Secure File Transfer Protocol Daemon*. VSFTPD ini merupakan salah satu aplikasi dari FTP yang dapat berjalan pada protokol IPv6 yang menggunakan *Operating System* Ubuntu. Aplikasi VSFTPD ini termasuk didalam ubuntu saat di-*install*. Fitur – fitur yang dimiliki VSFTPD adalah:

1. *Virtual IP configuration*, VSFTPD memberikan kemudahan dalam mengkonfigurasi. Konfigurasi ini dilakukan secara virtual melalui *command form* dan disimpan dalam bentuk file VSFTPD.conf.
2. *Virtual users*, dalam membuat dan meregistrasikan pengguna sangat mudah dan juga dilakukan secara virtual menggunakan *command form*.
3. *Standalone or inetd operation*, dapat dijalankan pada *inetd* atau IPv4 sendiri.
4. *Bandwidth throttling*, dengan adanya pembatasan *bandwidth* agar dapat mencegah atau mengurangi kemacetan dan mengatur lalu lintas saat ada transfer file.
5. Dapat membatasi *user* atau *client* menggunakan per-source-IP.
6. *Standalone IPv6*, dapat berjalan pada jaringan IPv6.

VSFTPD ini memiliki kelebihan dalam *security*-nya, performansinya dan kestabilannya.

3.4.2 Tahap Instalasi dan Konfigurasi VSFTPD

FTP *server* yang digunakan adalah VSFTPD. VSFTPD *server* di-*install* pada *correspondent node*. Konfigurasi yang dilakukan pada VSFTPD adalah pada file VSFTPD.conf dengan mengubah atau menambah konfigurasi yang dibutuhkan. Pada file VSFTPD.conf ini dapat menentukan *user anonymous* atau *user authentication* yang dapat mengakses FTP *server* pada *correspondent node*.

Pada VSFTPD *server* yang dibuat, *user* yang dapat mengakses adalah *user* yang sudah teregistrasi pada *server*.

Tahap – tahap menggunakan VSFTPD adalah:

1. *Install* VSFTPD. VSFTPD diinstal dengan menggunakan *command* `apt-get install VSFTPD` pada PC yang digunakan sebagai *correspondent node*.
3. Konfigurasi VSFTPD. Konfigurasi ini dilakukan dengan menggunakan *command* `gedit etc/VSFtpD.conf`. Konfigurasi VSFTPD sangat mudah karena hanya mengubah atau menambahkan konfigurasi pada file VSFTPD.conf. Untuk memastikan konfigurasi VSFTPD berjalan dengan baik, ketika merubah konfigurasi lakukan *restart* VSFTPD.
4. Konfigurasi *firewall*. Konfigurasi *firewall* ini dilakukan agar tidak semua *user* dapat mengakses direktori pada *server*. Tahap – tahap yang dilakukan adalah sebagai berikut :
 1. Membuat *group* untuk ftp-account dengan menggunakan *command* `# groupadd ftp-account`. Ftp-account ini adalah sebuah direktori yang *file*-nya dapat diakses oleh pengguna yang teregistrasi didalam *group* yang dibuat oleh pemilik *server*. Ftp-account ini terdapat di direktori `/home/ftp/ftp-account`.
 2. Konfigurasi *user*. Konfigurasi *user* ini dilakukan untuk mengubah hak akses direktori ftp-account agar dapat diakses *user*. Command yang digunakan adalah `# chmod 777 /home/ftp/ftp-account`. Sedangkan untuk mengubah kepemilikan direktori ftp-account adalah dengan menggunakan *command* `# chown root.ftp-account /home/ftp/ftp-account`.

Dalam file transfer protokol, yang dapat mengakses file pada *server* adalah *admin*, *user* dan pemilik *server*. *Admin* merupakan penyedia *file*, *user* adalah pengguna yang mengakses *file* dan pemilik *server* adalah yang dapat melakukan apapun termasuk yang mengatur siapa yang dapat mengakses dan siapa yang tidak. Dengan menggunakan `chmod 777`, ketika *men-download user* dapat mengakses dan membaca *file* yang disediakan oleh admin VSFTPD *server* serta mengedit file tersebut.

3. Membuat *user account* untuk dapat mengakses direktori. *Command* yang digunakan adalah `# useradd -g ftp-account -d /home/ftp/ftp-account acta1`. Maksud dari *command* tersebut adalah membuat *user* pada (-g) *group* untuk direktori ftp-account di direktori (-d) /home/ftp/ftp-account dengan nama acta1. *User* inilah yang nantinya dapat mengakses file direktori *server* dan digunakan untuk pengambilan data.

Tidak ada *user anonymous*. Untuk dapat mengakses *server* hanya *user* yang memiliki *account*.

3.4.3 Testing VSFTPD

Agar pengambilan data tidak mengalami kesalahan atau kegagalan, VSFTPD harus dites atau di uji terlebih dahulu dengan cara menjalankan pada jaringan *mobile IPv6*. *Testing* dilakukan dengan menggunakan *account acta1*. Dengan melakukan *testing VSFTPD*, pengujian ini bertujuan agar dapat memastikan apakah *user* yang dibuat dapat mengakses file dalam direktori *server*. Berikut adalah *testing VSFTPD* melalui *command form* (local host).

```
# ftp localhost

Connected to localhost.
220 Welcome to WINDA FTP service....
Name (localhost:ubuntu): "menggunakan user acta1 disini"
Name (localhost:ubuntu): acta1
331 Please specify the password.
Password:"password actarina disini"
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rwxrwxrwx   1 0   0   51200 Nov 03 08:49 Bab_6_ebook.doc
-rwxrwxrwx   1 0   0   142189 Nov 03 05:54 Paper.pdf
-rwxrwxrwx   1 0   0   112990 Nov 03 06:32 kupu2-gradasi.jpg
```

```

-rwxrwxrwx  1 0  0   2004466 Nov 03 05:24 messy2.AVI
-rwxrwxrwx  1 0  0    30 Nov 09 04:05 user.txt
-rwxrwxrwx  1 0  0   264842 Nov 18 06:14 winda.tar.gz

226 Directory send OK.
ftp>
ftp> get user.txt "get digunakan untuk mendownload salah satu file
pada direktori"
local: user.txt remote: user.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for user.txt (30 bytes).
226 File send OK.
30 bytes received in 0.01 secs (2.1 kB/s)
ftp>

```

Testing tidak hanya dilakukan pada *local host* saja. Agar dapat memastikan bahwa FTP *server* terkoneksi dengan *client*, testing VSFTPD dilakukan pada *mobile node* dengan menggunakan *browser*.

Mobile node merupakan host yang berfungsi sebagai *client* yang melakukan *download* dari direktori file yang berada di *correspondent node* yang telah dikonfigurasi VSFTPD *server*. Pada browser, untuk terkoneksi pada FTP *server* yang harus dilakukan adalah ftp://[alamat IPv6 pada *correspondent node*] yaitu [ftp://\[2001:0db8:3c4d:005b::4\]](ftp://[2001:0db8:3c4d:005b::4]) akan muncul tampilan seperti Gambar 3.3 berikut :



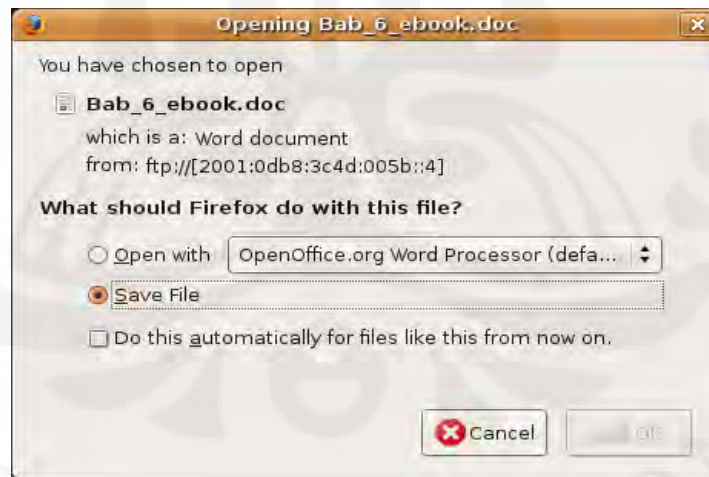
Gambar 3. 3 Autentikasi FTP

Alamat IPv6 pada *correspondent node* merupakan alamat dimana file atau direktori ftp dapat diakses. Autentikasi meminta *account* yang telah teregistrasi. Masukkan username dan password agar dapat masuk ke direktori yang dituju. Tampilan direktori yang akan muncul adalah seperti Gambar 3.4 berikut :



Gambar 3. 4 Tampilan Direktori File FTP

ketika men-*download* file yang diinginkan maka akan muncul tampilan seperti Gambar 3.5 berikut :



Gambar 3. 5 Tampilan *save file* saat men-*download*

3.5 METODE PENGAMBILAN DATA

Pengambilan data ini dilakukan untuk analisa performa FTP pada jaringan *mobile* IPv6. Pengambilan data pada jaringan *mobile* IPv6 ini dilakukan

berdasarkan parameter – parameter uji coba. Parameter uji coba yang digunakan adalah *delay*, *throughput* dan *transfer time*. Metode pengambilan data dilakukan dengan men-*download* file dari *server* ke *client* pada jaringan *mobile IPv6*. Bersamaan ketika *client* melakukan *download*, dengan menggunakan aplikasi Wireshark akan dilakukan penangkapan paket – paket yang lewat pada sisi *client*. File yang akan di-*download* dari FTP server akan dibedakan dalam tiga jenis file yaitu pdf, doc dan jpg dengan bermacam – macam ukuran, yaitu 16 Mbytes, 32 Mbytes, 64 Mbytes, 128 Mbytes dan 256 Mbytes. Dengan membedakan ukuran file yang akan di-*download* ini ditujukan untuk membandingkan ukuran file dengan parameter - parameter uji yang digunakan. Pengukuran parameter – parameter akan dilakukan di *mobile node*.

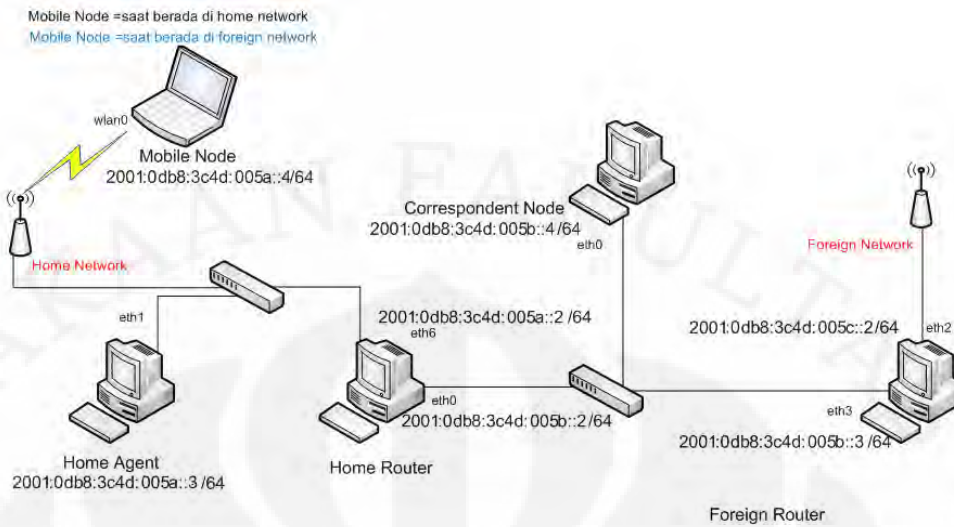
Pengambilan data pada masing – masing parameter akan dilakukan sebanyak 1 kali pada setiap ukuran file yang berbeda dan berdasarkan tiga jenis file yang berbeda. Ketika pengambilan data dilakukan ping 64Bytes dari server ke *client* yang bertujuan untuk memberikan beban kepada *client*.

3.5.1 Skenario Pengujian

Proses pengambilan data ini dilakukan sesuai dengan beberapa skenario. Skenario yang dirancang adalah sebagai berikut :

1. Skenario 1

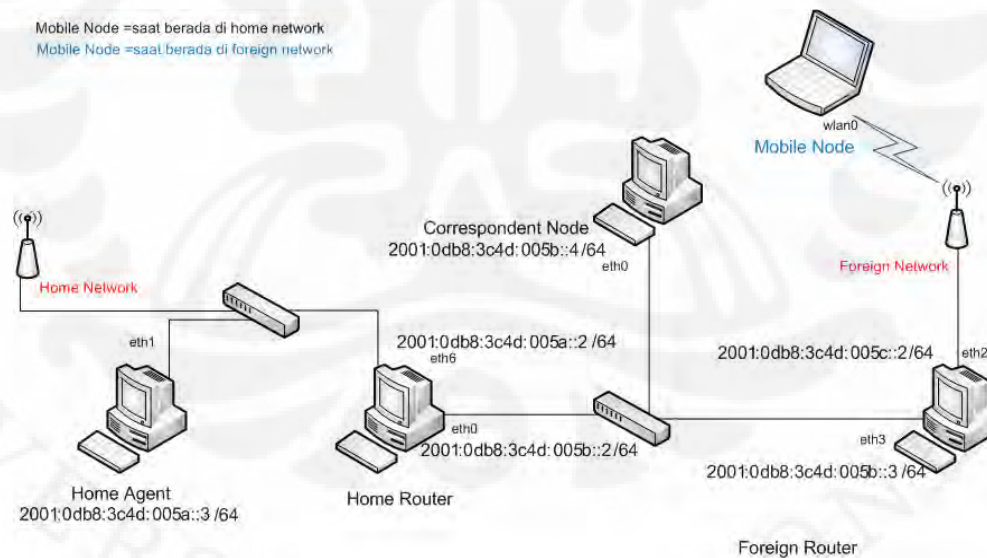
Pada skenario 1 dilakukan pengujian performa atau kualitas layanan pada saat *mobile node* tidak berpindah dari *home link*-nya. *Coresspondent node* tetap berada pada *foreign link* dan *mobile node* berada pada *home link*. Gambar 3.6 merupakan jaringan ketika diterapkan skenario 1 adalah sebagai berikut :



Gambar 3. 6 Skenario 1

2. Skenario 2

Pada skenario 2 dilakukan pengujian performa atau kualitas layanan pada saat *mobile node* berpindah ke *foreign link*. *Coresspondent node* tetap berada pada *foreign link* dan *mobile node* berada pada *foreign link*. Gambar 3.7 merupakan jaringan ketika diterapkan skenario 2.



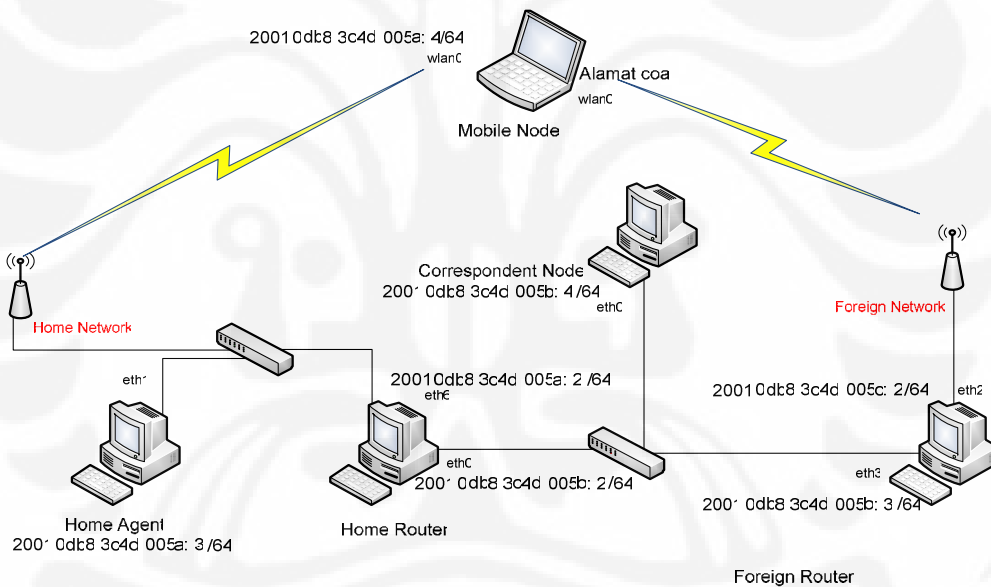
Gambar 3. 7 Skenario 2

Pada Gambar 3.7 *mobile node* berada pada *foreign link*. *mobile node* yang sebelumnya berada pada *home link* secara manual terkoneksi pada *foreign link*.

3. Skenario 3

Pada skenario 3 ini pengujian dilakukan ketika *mobile node* yang masih berada pada *home link* melakukan perpindahan *link* dari *home link* menuju *foreign link* atau proses *handover*. Perpindahan link ini dilakukan secara manual.

Sebelum pengambilan data, pada *mobile node* ketika berada pada *home link* akan dijalankan perintah `mip6d -c /etc/mip6d.conf` yang bertujuan untuk menjalankan *mobile IPv6*. Begitu juga pada *correspondent node* dan *home agent*. Untuk melakukan proses *handover* secara manual, dilakukan koneksi pada *foreign-network* sehingga muncul *care-of-address* pada tampilan perintah `mip6d -c /etc/mip6d.conf`. Pengambilan data dilakukan pada saat terjadi *handover* dari *home link* menuju *foreign link*. Skenario 3 dapat di lihat pada Gambar 3.8 sebagai berikut:



Gambar 3.8 Skenario 3

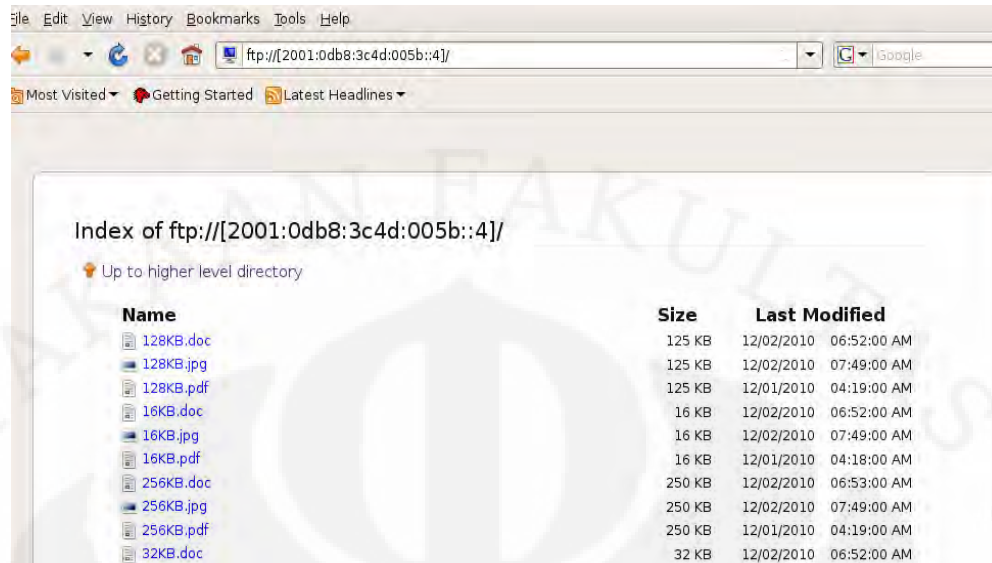
BAB 4

ANALISA DATA DAN KUALITAS LAYANAN MOBILE IPV6 DENGAN FTP

4.1 ANALISA PERFORMA JARINGAN PADA FTP

Jaringan uji yang digunakan dalam menganalisa performansi *mobile* IPv6 berdasarkan berbagai parameter uji dan merupakan jaringan *mobile* IPv6 murni. Jaringan *mobile* IPv6 ini terdiri dari dua *Router*, *Home Agent*, *Mobile Node* dan *Correspondent Node*. Pengujian dilakukan dengan melakukan *download* dari FTP *server* ke FTP *client* menggunakan *mobile node* yang berfungsi sebagai *client* dan *correspondent node* sebagai FTP *server*.

VSFTPD merupakan salah satu aplikasi dari FTP yang digunakan untuk *transfer file* data dari suatu *server* ke *client* atau sebaliknya. FTP merupakan suatu jenis aplikasi yang bekerja dengan memanfaatkan protocol TCP/IP yang terhubung menggunakan port 21. FTP *server* digunakan untuk menyimpan file – file yang akan di-*download* oleh *client*. Sedangkan FTP *client* berfungsi untuk melakukan proses *download* file data dari *server*. Proses yang dilakukan pertama adalah mengkoneksikan *client* ke *server*. Setelah terkoneksi maka *client* diminta mengisi *username* dan *password* oleh *server*. Setelah terhubung oleh *server*, *client* siap untuk men-*download*. Gambaran tampilan koneksi FTP *server* dan *client* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Tampilan FTP *client* saat terkoneksi dengan *server*

Pengambilan data dilakukan dengan menggunakan tiga jenis file, yaitu jenis file .pdf, .doc dan .jpg. Nama masing – masing data disesuaikan dengan ukuran file yang digunakan. Ukuran file yang digunakan adalah 16KB, 32KB,64KB, 128KB, 256KB dan 512KB. Pengambilan data dilakukan dengan menggunakan beban ping 64 bytes dari *server* ke *client* dan dilakukan berdasarkan tiga skenario. File – file yang diujikan ditunjukkan pada Tabel 4.1.

Tabel 4.1 File Uji Coba

Nama File			Ukuran File (Bytes)
Jenis PDF	Jenis DOC	Jenis JPG	
16KB.pdf	16KB.doc	16KB.jpg	16000
32KB.pdf	32KB.doc	32KB.jpg	32768
64KB.pdf	64KB.doc	64KB.jpg	65536
128KB.pdf	128KB.doc	128KB.jpg	131072
256KB.pdf	256KB.doc	256KB.jpg	262144

Parameter – parameter yang digunakan dalam analisa jaringan uji ini, yaitu *throughput*, *delay* dan *transfer time*. Analisa jaringan uji dengan menggunakan parameter tersebut dilakukan sesuai skenario yang dibuat, yaitu analisa jaringan ketika *mobile node* berada pada *home link* (skenario 1), pada *foreign link* (skenario 2) dan ketika *mobile node* mengalami *handover* (skenario 3). Pada

skenario 3, proses *handover* dalam pengujian ini dilakukan dengan manual atau *hard handover*.

4.2 ANALISA PADA *HOME LINK*

Analisa pada *home link* ini merupakan analisa yang dilakukan pada saat *mobile node* berada pada *home link* yang artinya *mobile node* tidak mengalami perpindahan *link* atau mengalami proses *handover*. Analisa ini dilakukan pada skenario 1. Dengan melakukan analisa ini, hasilnya diharapkan dapat mengetahui kualitas layanan yang diterima oleh *mobile node* pada saat berada pada *home link*.

4.2.1 Analisa *Throughput* (skenario 1)

Throughput merupakan kecepatan rata – rata transfer data yang berarti besarnya data rata – rata yang dapat ditransfer per detik pada suatu jaringan dengan pengukuran bps (*bit per second*). *Throughput* dapat dinyatakan sebagai *bytes per second*, *packet per second* atau *bit per second*. Pengambilan data untuk analisa *throughput* dilakukan saat *client* berada pada *home link* dan pada saat bersamaan pada sisi *client* melakukan penangkapan paket – paket yang masuk melalui *interface* WLAN ketika *client* sedang *men-download file* data dari *server*. Penangkapan paket – paket dilakukan dengan menggunakan aplikasi Wireshark. Hasil *capture* paket yang didapatkan Wireshark tidak hanya paket FTP yang berjalan pada protocol TCP/IP saja, merupakan paket – paket yang sudah di filter terlebih dahulu. Setelah melakukan *filtering* pada paket – paket FTP dapat ditentukan besar *throughput* ketika melakukan *download*. Pada paket – paket FTP yang sudah dilakukan *filtering*, *throughput* dapat ditentukan dengan melihat *summary*. Hasil *throughput* dapat dilihat pada Gambar 4.2.

Display			
Display filter:	ftp		
Traffic	Captured	Displayed	Marked
Packets	357	9	0
Between first and last packet	10.015 sec	0.294 sec	
Avg. packets/sec	35.646	30.644	
Avg. packet size	818.129 bytes	120.667 bytes	
Bytes	292072	1086	
Avg. bytes/sec	29163.004	3697.663	
Avg. MBit/sec	0.233	0.030	

Gambar 4.2 *Throughput* pada *Summary* Wireshark (Skenario 1)

Data yang ditangkap pada Gambar 4.2 merupakan data pada file 256KB.pdf ketika di-*download*. Terlihat bahwa hasil capture diatas masih dalam satuan *bytes per second* sehingga diubah terlebih dahulu kedalam satuan *bit per second* dengan cara mengalikan delapan pada data yang diambil dari Wireshark. Hasil data yang didapat secara keseluruhan untuk perhitungan nilai rata – rata *throughput* sesuai jenis file yang di-*download* dapat ditampilkan pada Tabel 4.2.

Tabel 4.2 Data Nilai *Throughput*

File Size (Kbytes)	Throughput (Bytes/sec)		
	File Pdf	File Doc	File jpg
16	19576.10	15244.80	11102.70
32	19004.40	5135.82	5014.42
64	8229.07	13391.00	3273.84
128	4994.37	7397.42	6629.31
256	3697.66	3595.20	4569.52
Rata -rata	11100.30	8952.85	6117.95

Dari Tabel 4.2, terlihat bahwa nilai *throughput* dari setiap jenis file yang berbeda dengan ukuran yang berbeda ternyata memiliki hasil berbeda. Berdasarkan ukuran file yang digunakan untuk men-*download*, semakin besar ukuran file maka umumnya semakin kecil nilai *throughput*-nya. Berdasarkan rata – ratanya, file jenis jpg memiliki nilai *throughput* paling kecil. Secara persentase, file jenis doc

lebih kecil 19.35 % dari file jenis pdf. Sedangkan file jenis jpg, file jenis ini lebih kecil dari file pdf sebesar 44.88 %.

Bila dilihat dari file doc dengan ukuran 64 dan 128 Kbytes seharusnya memiliki nilai *throughput* yang lebih kecil dari file doc berukuran 32 KBytes. Setelah dianalisa, besarnya nilai *throughput* pada file 128 KBytes tersebut dapat dilihat pada Gambar 4.3. Pada saat TCP mengirimkan ACK dari *client* ke *server* dan *server* mengirimkan FTP-DATA ke *client*, pada saat itulah terjadi kesalahan *checksum*. Kesalahan *checksum* yang terjadi saat pengiriman data diakibatkan TCP *checksum* mengalami *offload*, hal tersebut juga dapat diakibatkan karena beratnya suatu traffic, adanya masalah dengan network card dan lainnya. Ketika transfer data pada FTP mengalami *bad checksum*, pada sisi *server* akan melakukan transfer kembali file data. File yang mengalami *bad checksum* ketika pentransferan berlangsung adalah file pdf 64 KB yang mengalami 1 kali *bad checksum* dan file doc ukuran 128 KB yang mengalami 3 kali *bad checksum*.

```

54 3.107711 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
55 3.107725 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
56 3.108045 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
57 3.108385 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
58 3.108394 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
59 3.109255 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
60 3.109264 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
61 3.109598 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
62 3.109606 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
63 3.109942 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
64 3.109951 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
65 3.110284 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
66 3.110292 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
67 3.110627 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
68 3.111070 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
69 3.111414 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes
70 3.111422 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 47892 > 39052 [ACK] Seq=3909826331 Ack=292292667 Win=336 Len=0 TSV=528
71 3.111758 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP-DATA FTP Data: 1428 bytes

Transmission Control Protocol, Src Port: 39052 (39052), Dst Port: 47892 (47892), Seq: 292292667, Ack: 3909826331, Len: 1428
Source port: 39052 (39052)
Destination port: 47892 (47892)
Sequence number: 292292667 (relative sequence number)
[Next sequence number: 292294095 (relative sequence number)]
Acknowledgement number: 3909826331 (relative ack number)
Header length: 32 bytes
Flags: 0x10 (ACK)
Window size: 179
Checksum: 0xb5f7 [Incorrect, should be 0x1e3e (maybe caused by "TCP checksum offload"?)]
[Good Checksum: False]
[Bad Checksum: True]
Options: (12 bytes)
FTP Data

```

Gambar 4.3 Capture wireshark ketika terjadi *Bad Checksum*

Diantara jenis file doc dan jpg pada Tabel 4.2, yang mengalami *bad checksum* terbanyak adalah pada file jenis jpg. Untuk file jpg ukuran 128 KBytes mengalami 1 kali *bad checksum*, untuk file 256 Kbytes mengalami *bad checksum* 2 kali dan 32 KBytes selain mengalami *bad checksum* 1 kali, pada saat TCP mengalami *delay* dalam mengirimkan ACK ke *client* terjadi TCP *retransmission* pada saat file FTP selesai terkirim. Sehingga byte yang diterima lebih besar yang berakibat

throughput lebih besar. TCP *retransmission* dapat dilihat pada Gambar 4.4 dibawah.

```

92 5.577259 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 TCP 27358 > 40803 [ACK] Seq=206426788 Ack=3806696807 win=
93 5.577670 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP Response: 226 File send OK.
94 5.577686 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP [TCP Dup ACK 44#1] 39855 > ftp [ACK] Seq=118 Ack=80
97 5.765882 2001:db8:3c4d:5b::4 2001:db8:3c4d:5a::4 FTP [TCP Retransmission] Response: 150 Opening BINARY m
98 5.765919 2001:db8:3c4d:5a::4 2001:db8:3c4d:5b::4 TCP 39855 > ftp [ACK] Seq=118 Ack=190 Win=45 Len=0 TSv=

... .. = Push: Not set
... ..0.. = Reset: NOT set
... ..0.. = Syn: NOT set
... ..0.. = Fin: NOT set
window size: 45
[Checksum: 0xabbb0 [correct]
[Good Checksum: True]
[Bad Checksum: False]
options: (24 bytes)
[SEQ/ACK analysis]
[TCP Analysis Flags]
[This is a TCP duplicate ack]
[duplicate ACK #: 1]
[duplicate to the ACK in frame: 44]

```

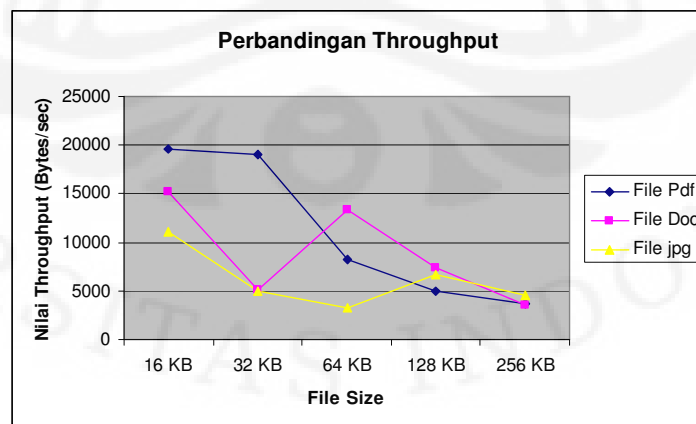
Gambar 4.4 *capture* wireshark ketika mengalami TCP *retransmission*

Secara detail dapat dilihat pada Gambar 4.5 dibawah ini.

No.	Sever.	Group	Protocol	Summary
37	Chat	Sequence	TCP	Connection establish request (SYN); server port 27358
39	Chat	Sequence	TCP	Connection establish acknowledge (SYN+ACK); server port 27358
45	Error	Checksum	TCP	Bad checksum
90	Chat	Sequence	TCP	Connection finish (FIN)
91	Chat	Sequence	TCP	Connection finish (FIN)
94	Note	Sequence	TCP	Duplicate ACK (#1)
97	Note	Sequence	TCP	Retransmission (suspected)

Gambar 4.5 Detail TCP *Bad Checksum* pada *Expert Infos*

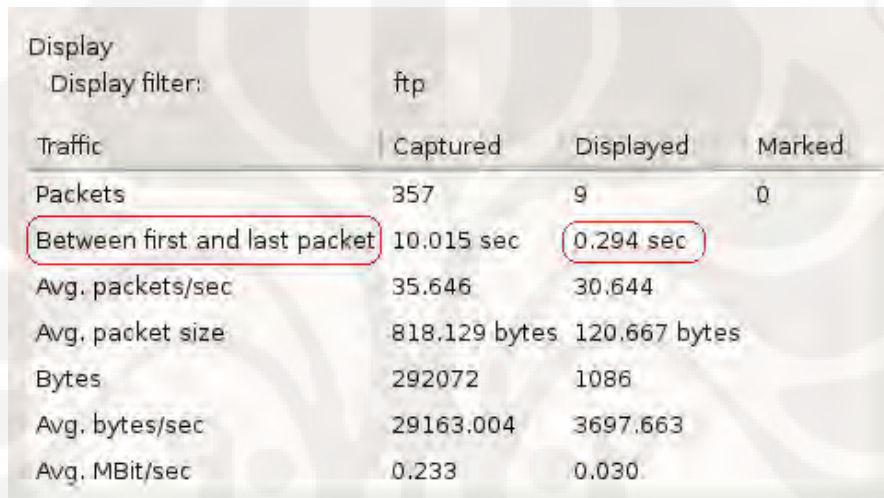
Hasil grafik dari *throughput* pada *Home link* dapat dilihat pada Gambar 4.6 sebagai berikut.



Gambar 4.6 Grafik *Throughput* pada *Home Link*

4.2.2 Analisa *Transfer Time* (skenario 1)

Transfer time merupakan jumlah waktu yang dibutuhkan untuk mengirimkan file data dari *FTP server* ke *FTP client*. Seperti halnya pengukuran parameter *throughput*, parameter *transfer time* di ambil dari hasil *capture* wireshark yang sebelumnya dilakukan *filtering* untuk paket – paket *FTP*. *Transfer time* didapatkan dari perbedaan waktu dari paket pertama sampai dengan paket terakhir. *Transfer time* dapat dilihat pada *summary* wireshark pada Gambar 4.7 dibawah ini.



Traffic	Captured	Displayed	Marked
Packets	357	9	0
Between first and last packet	10.015 sec	0.294 sec	
Avg. packets/sec	35.646	30.644	
Avg. packet size	818.129 bytes	120.667 bytes	
Bytes	292072	1086	
Avg. bytes/sec	29163.004	3697.663	
Avg. MBit/sec	0.233	0.030	

Gambar 4.7 *Transfer Time* pada *Summary* Wireshark (Skenario 1)

Pengambilan data dilakukan dengan ukuran file berbeda dan tiga jenis file yang berbeda. Hasil pengambilan data dari nilai *transfer time* dapat dilihat pada Tabel 4.3 dibawah ini.

Tabel 4.3 Data Nilai *Transfer Time*

File Size (Kbytes)	Transfer Time (s)		
	File Pdf	File Doc	File jpg
16	0.066	0.071	0.097
32	0.057	0.210	0.251
64	0.131	0.081	0.330
128	0.217	0.147	0.164
256	0.294	0.302	0.238
Rata – rata	0.1530	0.1622	0.2160

Dari data Tabel 4.3 secara umum, terlihat bahwa semakin besar ukuran file yang di-*download*, semakin besar nilai *transfer time* yang didapat. Data yang didapat file jenis doc dan file pdf memiliki rata – rata yang tidak terlalu signifikan. Pada file jpg memiliki nilai rata – rata *transfer time* yang paling besar. Hal tersebut diakibatkan karena pada saat transfer data, file jpg langsung tampil pada layar *browser*. Berbeda dengan file doc dan pdf, pada saat *download* keluar tampilan *save as* dahulu. Secara perbandingan persentase, file jenis pdf memiliki nilai *transfer time* lebih kecil dari file jenis doc sebesar 6.01 % dan lebih kecil dari file jpg sebesar 41.17 %.

4.2.3 Analisa Delay (skenario 1)

Delay merupakan waktu yang dibutuhkan untuk setiap bit data dalam melewati jaringan dari terminal sumber ke terminal tujuan. Parameter *delay* didapatkan dari hasil pembagian nilai *transfer time* dengan jumlah bit data. Rumusnya adalah sebagai berikut

$$\dots \quad \text{Delay (s)} = \frac{\text{Transfer Time (s)}}{\text{Jumlah bit}} \quad [8]$$

Hasil pengambilan data untuk parameter *delay* dapat dilihat pada Tabel 4.4 sebagai berikut.

Tabel 4.4 Data Perhitungan *Delay*

File Size (Kbytes)	Delay (µs)		
	File Pdf	File Doc	File jpg
16	0.0063	0.0082	0.0112
32	0.0066	0.0243	0.0250
64	0.0150	0.0094	0.0380
128	0.0250	0.0169	0.0188
256	0.0340	0.0347	0.0273
Rata - rata	0.01738	0.01870	0.02406

Berdasarkan data pada Tabel 4.4, rata – rata *delay* yang didapat untuk jenis file pdf dan doc memiliki rata – rata yang stabil, tetapi pada file jpg memiliki nilai *delay* yang paling tinggi. Hal tersebut dikarenakan pada proses transfer filenya berbeda dengan file pdf dan doc. Ketika file jpg di-*download*, file langsung ditampilkan tanpa harus menyimpan terlebih dahulu sehingga proses transfer data lebih lama dengan *delay* yang lebih besar.

Berdasarkan persentasenya, file jenis pdf memiliki *delay* rata – rata yang lebih kecil 7.59 % dibandingkan dengan file jenis doc. Jika dibandingkan dengan file jenis jpg, file pdf memiliki nilai *delay* jauh lebih kecil 38.43 % dari file jpg. Diantara ketiga file tersebut, file pdf dapat dikatakan lebih dari yang lainnya karena memiliki *delay* yang lebih kecil.

4.3 ANALISA PADA *FOREIGN LINK*

Analisa pada *foreign link* ini merupakan analisa yang dilakukan pada saat *mobile node* berada pada *foreign link* yang artinya *mobile node* mengalami perpindahan *link* atau mengalami proses *handover*. Analisa ini dilakukan sesuai skenario 2, yaitu pada saat *mobile node* terkoneksi dengan *foreign link*. Dengan melakukan analisa ini, hasilnya diharapkan dapat mengetahui kualitas layanan yang diterima oleh *mobile node* pada saat berada pada *foreign link*.

4.3.1 Analisa *Throughput* (skenario 2)

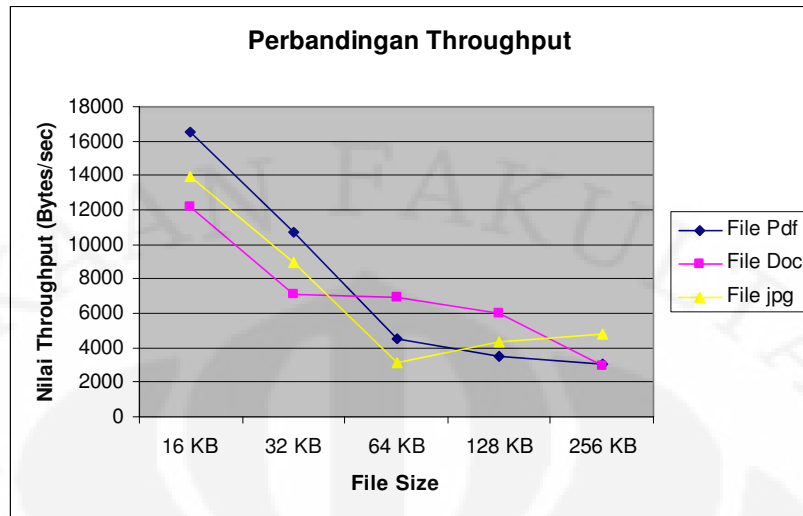
Pengambilan data dari Wireshark dilakukan berdasarkan skenario 2, yaitu pada saat *mobile node* terkoneksi dengan *foreign link*. *Mobile node* dikoneksikan secara manual dengan *foreign link*. Hasil penangkapan paket FTP pada Wireshark dapat dilihat pada Gambar 4.8 dan Hasil pengambilan data rata – rata dari nilai *throughput* dapat dilihat pada Tabel 4.5.

Traffic	Captured	Displayed	Marked
Packets	328	9	0
Between first and last packet	9.957 sec	0.422 sec	
Avg. packets/sec	32.941	21.339	
Avg. packet size	896.512 bytes	144.667 bytes	
Bytes	294056	1302	
Avg. bytes/sec	29532.305	3087.065	
Avg. MBit/sec	0.236	0.025	

Gambar 4.8 *Throughput* pada *Summary* Wireshark (Skenario 2)Tabel 4.5 Data Nilai *Throughput*

File Size (Kbytes)	Throughput (Bytes/sec)		
	File Pdf	File Doc	File jpg
16	16501.90	12167.90	13952.32
32	10685.00	7082.61	8987.63
64	4522.47	6918.38	3136.09
128	3545.89	6023.46	4330.01
256	3087.07	2962.48	4839.45
Rata – rata	7668.47	7030.87	7049.10

Dari data pada Tabel 4.5, dapat dilihat bahwa semakin besar ukuran file yang di *download*, maka semakin kecil nilai *throughput* yang didapat. Nilai *throughput* dapat mengalami perubahan yang signifikan ataupun tidak sesuai dengan kecepatan transfer file pada saat men-*download*. Rata – rata *throughput* pada setiap jenis file dapat dilihat bahwa nilai *throughput* terkecil adalah pada file doc, tetapi pada file doc, jpg dan pdf memiliki nilai *throughput* yang tidak jauh berbeda atau stabil. Berdasarkan persentasenya, file doc memiliki *throughput* lebih kecil 8.31 % dari file jenis pdf. Jika dibandingkan dengan file jenis jpg, file jenis ini memiliki nilai *throughput* lebih kecil 8.08 % dari file pdf. Hasil grafik dari *throughput* pada *foreign link* dapat dilihat pada Gambar 4.9 sebagai berikut.



Gambar 4.9 Grafik *Throughput* pada *Foreign Link*

4.3.2 Analisa *Transfer Time* (skenario 2)

Hasil penangkapan paket FTP pada wireshark untuk skenario 2 dapat dilihat pada Gambar 4.10 dan Hasil pengambilan data rata – rata dari nilai *transfer time* dapat dilihat pada Tabel 4.6.

Display			
Display filter:	ftp		
Traffic	Captured	Displayed	Marked
Packets	328	9	0
Between first and last packet	9.957 sec	0.422 sec	
Avg. packets/sec	32.941	21.339	
Avg. packet size	896.512 bytes	144.667 bytes	
Bytes	294056	1302	
Avg. bytes/sec	29532.305	3087.065	
Avg. MBit/sec	0.236	0.025	

Gambar 4.10 *Transfer Time* pada *Summary* Wireshark (Skenario 2)

Tabel 4.6 Data Nilai *Transfer Time*

File Size (Kbytes)	Transfer Time (s)		
	File Pdf	File Doc	File jpg
16	0.065	0.107	0.100
32	0.121	0.183	0.144
64	0.287	0.187	0.413
128	0.367	0.216	0.301
256	0.422	0.439	0.269
Rata – rata	0.2524	0.2264	0.2454

Berdasarkan data – data pada Tabel 4.6 secara umum, semakin besar ukuran file yang di-*download* maka semakin besar transfer time yang didapat. Nilai *transfer time* pada setiap jenis file hampir sama dan tidak mengalami perbedaan yang signifikan. Dapat dilihat dari rata – rata *transfer time*, bahwa file jenis doc memiliki nilai *transfer time* terkecil dan file jenis pdf memiliki nilai *transfer time* terbesar. Secara persentasenya, file doc memiliki *transfer time* lebih kecil 10.30 % dari file pdf. Jika dibandingkan dengan file jpg, file jpg memiliki *transfer time* lebih kecil 2.77 % dari file pdf.

4.3.3 Analisa *Delay* (skenario 2)

Tabel 4.7 Data Perhitungan *Delay*

File Size (Kbytes)	Delay (μ s)		
	File Pdf	File Doc	File jpg
16	0.0075	0.0103	0.0089
32	0.0116	0.0177	0.0139
64	0.0276	0.0180	0.0398
128	0.0352	0.0207	0.0289
256	0.0424	0.0421	0.0258
Rata – rata	0.02486	0.02176	0.02346

Berdasarkan pada hasil perhitungan *delay* Tabel 4.7, semakin besar ukuran file maka semakin besar nilai *delay* yang didapat. Jika dibandingkan rata – rata pada setiap jenis file yang berbeda, file jenis doc memiliki nilai *delay* yang lebih sedikit dari yang lainnya. Secara persentase, file doc memiliki *delay* lebih kecil dari file

jenis pdf sebesar 12.47 %. Jika dibandingkan dengan file jenis jpg, maka nilai *delay* pada file jpg lebih kecil dari file pdf sebesar 5.63 %.

4.4 ANALISA PERBANDINGAN SKENARIO 1 DAN SKENARIO 2

Analisa ini dilakukan untuk membandingkan hasil dari skenario 1 dan skenario 2 berdasarkan *delay*, *throughput* dan *transfer time*. Analisa perbandingan ini bertujuan untuk mengetahui baik atau tidaknya suatu *network* dari *network* lainnya pada suatu jaringan *mobile IPv6*. Selain itu untuk membandingkan jenis file yang memiliki performa *transfer file* yang paling baik.

4.4.1 Analisa Perbandingan *Throughput*

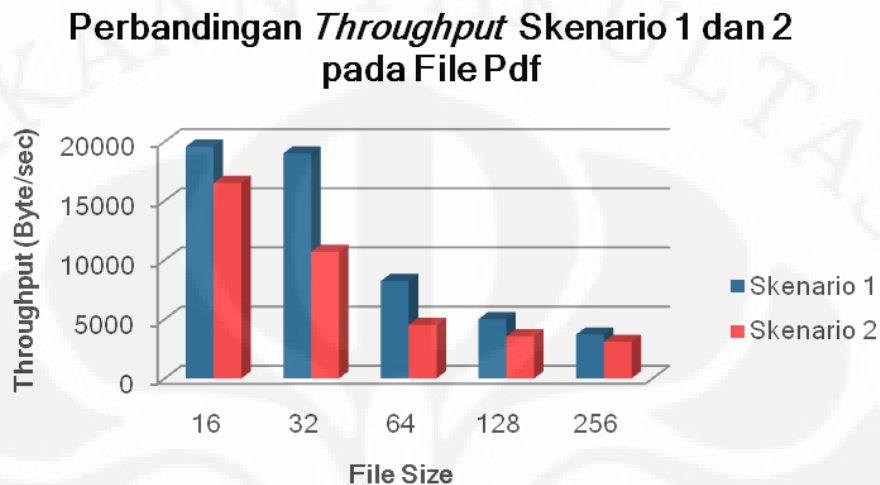
Dari pengambilan data yang dilakukan sesuai skenario 1 dan skenario 2, hasil data secara keseluruhan nilai *throughput* dapat dilihat pada Tabel 4.8 sebagai berikut.

Tabel 4.8 Data Nilai *Throughput* Keseluruhan

File Size (KBytes)	Throughput (Bytes/sec)					
	Skenario 1			Skenario 2		
	File Pdf	File Doc	File jpg	File Pdf	File Doc	File jpg
16	19576.10	15244.80	11102.7	16501.90	12167.90	13952.32
32	19004.40	5135.82	5014.42	10685.00	7082.61	8987.63
64	8229.07	13391.00	3273.84	4522.47	6918.38	3136.09
128	4994.37	7397.42	6629.31	3545.89	6023.46	4330.01
256	3697.66	3595.20	4569.52	3087.07	2962.48	4839.45
Rata -rata	11100.30	8952.85	6117.95	7668.47	7030.87	7049.10

Pada Tabel 4.8, jika nilai *throughput* dibandingkan antara skenario 1 dan skenario 2 dapat dilihat bahwa pada keseluruhan rata – rata setiap jenis file pada skenario 1 memiliki nilai *throughput* lebih besar dari skenario 2. Hal tersebut dikarenakan *mobile node* tetap berada pada link-nya yaitu *home link*. Diketahui bahwa semakin besar ukuran file yang digunakan untuk di-*download*, maka semakin besar nilai *throughput* yang didapat. Semakin besar nilai *throughput*, maka semakin besar data yang dapat ditransfer per detik. Jadi, *mobile node* akan mendapatkan performansi lebih baik jika berada pada *home link* dari pada *foreign link*.

Secara persentase, file pdf pada skenario 2 memiliki nilai *throughput* lebih kecil 30.91 % dari file pdf pada skenario 1. Perbandingan *throughput* file pdf pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.11.

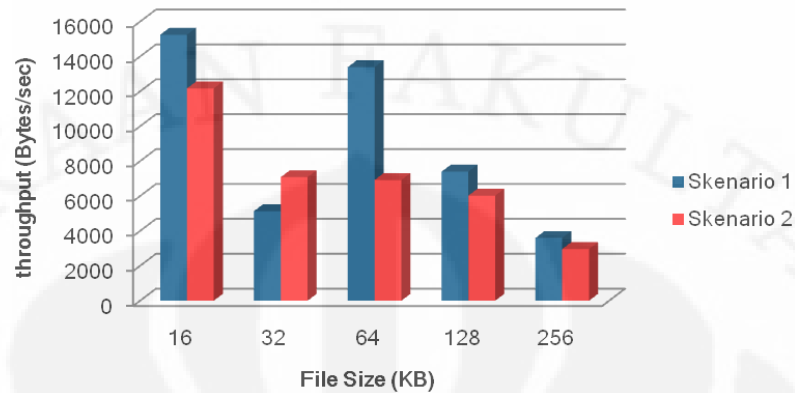


Gambar 4.11 Perbandingan *Throughput* Skenario 1 dan 2 pada File Pdf

Bila dilihat dari Gambar 4.11, grafik perbandingan *throughput* yang dihasilkan membentuk grafik linear. Hal tersebut menunjukkan bahwa file jenis pdf dikatakan memiliki performa yang baik didalam jaringan ketika dilakukan transfer.

Pada file doc pada skenario 2 memiliki nilai *throughput* lebih kecil 21.47 % dari file doc pada skenario 1. Perbandingan *throughput* file doc pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.12.

Perbandingan *Throughput* Skenario 1 dan 2 pada File Doc

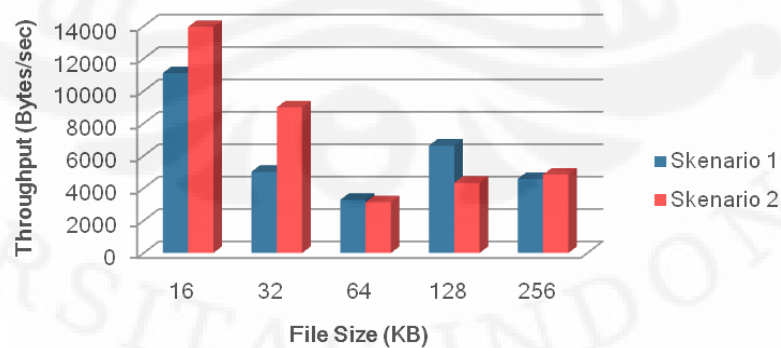


Gambar 4.12 Perbandingan *Throughput* Skenario 1 dan 2 pada File Doc

Dilihat dari Gambar 4.12, grafik perbandingan *throughput* pada file jenis doc ternyata tidak linear. Hal tersebut diakibatkan pada file ukuran 32 KBytes skenario 1 memiliki nilai *throughput* lebih kecil dari skenario 2. Pada skenario 1 pada saat pentransferan file 32 KB jaringan mengalami performa yang kurang baik. Menurunnya performa suatu jaringan dapat dibuktikan dari banyaknya *bad checksum* yang didapatkan ketika melakukan transfer file.

Pada file jpg pada skenario 1 memiliki nilai *throughput* lebih kecil 15.22 % dari file jpg pada skenario 2. Perbandingan *throughput* file jpg pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.13.

Perbandingan *Throughput* Skenario 1 dan 2 pada File Jpg



Gambar 4.13 Perbandingan *Throughput* Skenario 1 dan 2 pada File Jpg

Pada Gambar 4.13, grafik perbandingan *throughput* pada file jpg tidak linear sama seperti pada file doc. Tetapi pada grafik diatas skenario 1 terlihat memiliki performa yang kurang baik bila dibandingkan dengan scenario 2.

4.4.2 Analisa Perbandingan *Transfer Time*

Dari pengambilan data skenario 1 dan skenario 2 didapatkan tabel perbandingan nilai transfer time yang ditunjukkan pada tabel 4.9 sebagai berikut.

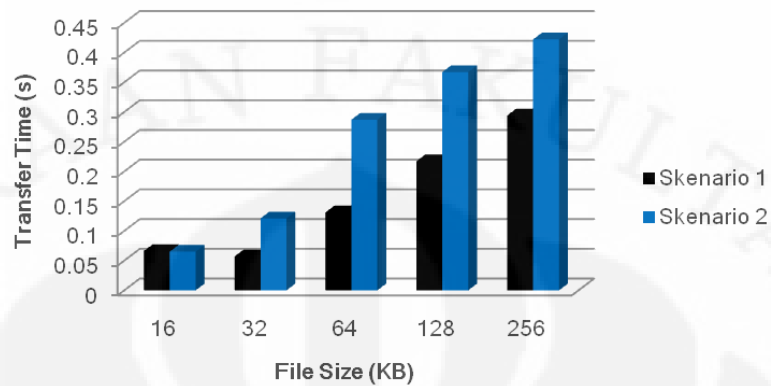
Tabel 4.9 Data Nilai *Transfer Time* Keseluruhan

File Size (KBytes)	Transfer Time (sec)					
	Skenario 1			Skenario 2		
	File Pdf	File Doc	File jpg	File Pdf	File Doc	File jpg
16	0.066	0.071	0.097	0.065	0.107	0.100
32	0.057	0.210	0.251	0.121	0.183	0.144
64	0.131	0.081	0.330	0.287	0.187	0.413
128	0.217	0.147	0.164	0.367	0.216	0.301
256	0.294	0.302	0.238	0.422	0.439	0.269
Rata-rata	0.153	0.1622	0.216	0.2524	0.2264	0.2454

Berdasarkan data rata – rata pada Tabel 4.9, didapatkan bahwa setiap jenis file pada skenario 1 memiliki *transfer time* lebih kecil dari skenario 2. Dapat dilihat bahwa mulai dari file pdf skenario 1 sampai file jpg skenario 2 mengalami kenaikan nilai *transfer time*. Diketahui bahwa semakin besar ukuran file yang di-*download*, maka semakin besar nilai *transfer time* yang didapat. Semakin kecil nilai *transfer time*, maka semakin bagus performansi jaringan yang digunakan untuk men-*download*. Rata – rata nilai *transfer time* pada skenario 1 lebih kecil dari skenario 2 diakibatkan karena *mobile node* tidak berpindah dari *network* asalnya yaitu *home link*. Ini menunjukkan bahwa *mobile node* akan bekerja lebih baik pada *home link* dari pada *foreign link*.

Secara persentase, file pdf pada skenario 1 memiliki nilai *transfer time* lebih kecil 66.97 % dari file pdf pada skenario 2. Perbandingan transfer time file pdf pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.14.

Perbandingan *Transfer Time* Skenario 1 dan 2 pada File Pdf

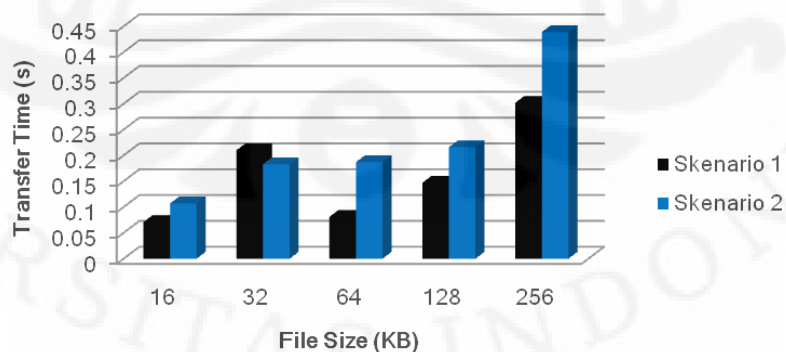


Gambar 4.14 Perbandingan *Transfer Time* Skenario 1 dan 2 pada File Pdf

Pada Gambar 4.14, dapat dilihat bahwa grafik perbandingan *transfer time* berbentuk linear. Setiap ukuran file yang berbeda menghasilkan grafik dengan nilai *transfer time* pada skenario 1 lebih kecil dari skenario 2. Hal tersebut membuktikan bahwa skenario 1 untuk waktu transfer file pdf lebih baik dari skenario 2.

Pada file doc pada skenario 1 memiliki nilai *transfer time* lebih kecil 39.58 % dari file doc pada skenario 2. Perbandingan transfer time file doc pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.15.

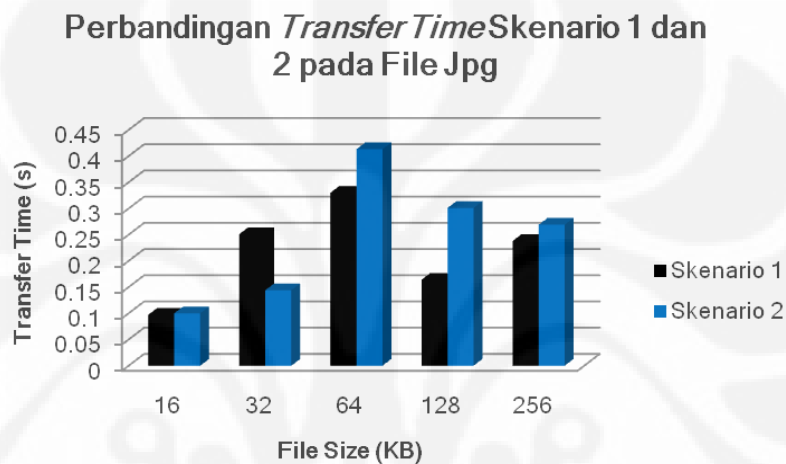
Perbandingan *Transfer Time* Skenario 1 dan 2 pada File Doc



Gambar 4.15 Perbandingan *Transfer Time* Skenario 1 dan 2 pada File Doc

Dari Gambar 4.15, pada file ukuran 32 KBytes skenario 1 memiliki nilai *transfer time* lebih besar dari skenario 2 karena pada skenario 1 mengalami banyak *bad checksum*. sehingga *bytes* yang diterima sama tetapi dengan waktu yang lebih lama. Sama seperti grafik pada file jpg pada Gambar 4.16 dibawah.

Pada file jpg pada skenario 1 memiliki nilai *transfer time* lebih kecil 13.61 % dari file jpg pada skenario 2. Perbandingan *transfer time* file jpg pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.16.



Gambar 4.16 Perbandingan *Transfer Time* Skenario 1 dan 2 pada File Jpg

4.4.1 Analisa Perbandingan *Delay*

Dari pengambilan data skenario 1 dan skenario 2 didapatkan tabel perbandingan nilai perhitungan *delay* yang ditunjukkan pada Tabel 4.9.

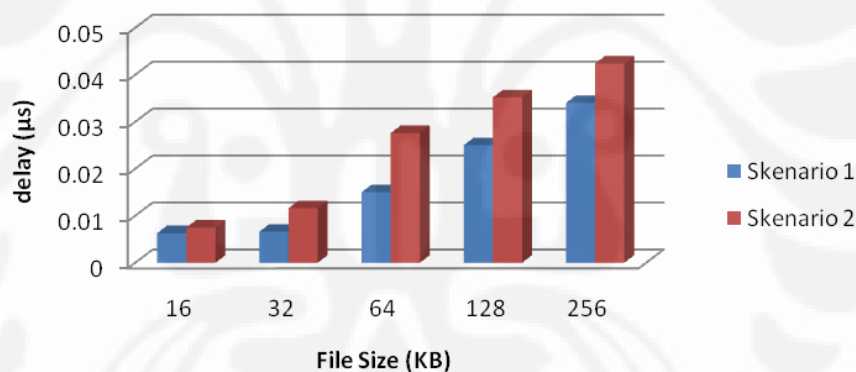
Tabel 4.10 Data Nilai *Delay* Keseluruhan

File Size (Kbytes)	Delay (μ s)					
	Skenario 1			Skenario 2		
	File Pdf	File Doc	File jpg	File Pdf	File Doc	File jpg
16	0.0063	0.0082	0.0112	0.0075	0.0103	0.0089
32	0.0066	0.0243	0.0250	0.0116	0.0177	0.0139
64	0.0150	0.0094	0.0380	0.0276	0.0180	0.0398
128	0.0250	0.0169	0.0188	0.0352	0.0207	0.0289
256	0.0340	0.0347	0.0273	0.0424	0.0421	0.0258
Rata -rata	0.01738	0.01870	0.02406	0.02486	0.02176	0.02346

Berdasarkan data rata – rata pada Tabel 4.10, didapatkan bahwa jenis file pdf memiliki nilai rata – rata delay terkecil. Setiap jenis file pada skenario 1 memiliki nilai *delay* lebih kecil dari skenario 2. Diketahui bahwa semakin besar ukuran file yang di-*download*, maka semakin besar nilai *delay* yang didapat. Semakin kecil nilai *delay*, maka semakin bagus performansi jaringan yang digunakan untuk men-*download*. Data rata – rata perhitungan nilai *delay* pada skenario 1 lebih kecil dari skenario 2, sehingga didapatkan bahwa *mobile node* memiliki performansi yang lebih baik ketika berada pada *home link* sesuai skenario 1.

Secara persentase, file pdf pada skenario 1 memiliki nilai *delay* lebih kecil 43.04 % dari file pdf pada skenario 2. Perbandingan *delay* file pdf pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.17.

Perbandingan Delay Skenario 1 dan 2 pada file Pdf

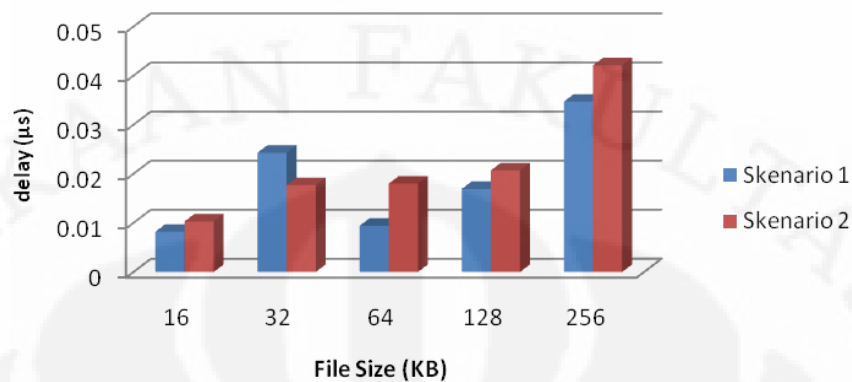


Gambar 4.17 Perbandingan *Delay* Skenario 1 dan 2 pada File Pdf

Berdasarkan Gambar 4.17, perbandingan *delay* pada skenario 1 dan 2 didapatkan grafik yang linear. Terlihat bahwa skenario 1 lebih baik dari pada skenario 2.

Pada file doc pada skenario 1 memiliki nilai *delay* lebih kecil 16.36 % dari file doc pada skenario 2. Perbandingan *delay* file doc pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.18.

Perbandingan Delay Skenario 1 dan 2 pada file Doc

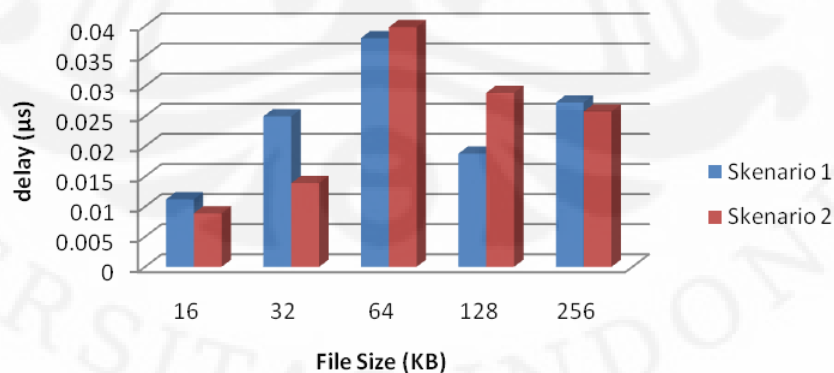


Gambar 4.18 Perbandingan *Delay* Skenario 1 dan 2 pada File Doc

Dari Gambar 4.18, pada file ukuran 32 KBytes skenario 1 memiliki nilai *delay* lebih besar dari skenario 2 karena pada skenario 1 mengalami banyak *bad checksum*. Hal tersebut diakibatkan nilai *transfer time* yang lebih besar dari skenario 2.

Pada file jpg pada skenario 1 memiliki nilai *delay* lebih besar 2.49 % dari file jpg pada skenario 2. Perbandingan *delay* file jpg pada skenario 1 dan skenario 2 dapat ditunjukkan pada Gambar 4.19.

Perbandingan Delay Skenario 1 dan 2 pada file Jpg



Gambar 4.19 Perbandingan *Delay* Skenario 1 dan 2 pada File Jpg

Dari Gambar 4.19 terlihat bahwa nilai *delay* pada skenario 1 dan skenario 2 berubah – ubah. Sehingga analisa didapatkan dari rata – rata pada setiap skenario

yang dinyatakan pada persentase. Besarnya nilai delay yang terjadi pada file ukuran 16, 32 dan 256 KB diakibatkan banyaknya bad checksum yang terjadi selama penransferan.

Berdasarkan delay, throughput dan transfer time pada analisa parameter – parameter diatas dapat dilihat sesuai grafik bahwa perbedaan jenis file ternyata mempengaruhi performa jaringan. Bila jenis file dibandingkan, jenis file pdf adalah file yang lebih baik dari pada jpg dan doc. Hal tersebut dikarenakan pada file pdf merupakan file hasil konversi, berbeda dengan file asli atau document dan file yang berupa gambar.

4.5 ANALISA PADA *HANDOVER*

Analisa pada *handover* ini merupakan analisa yang dilakukan pada saat *mobile node* melakukan perpindahan link dari *home link* ke *foreign link* yang artinya pengambilan data akan dilakukan pada saat *mobile node* melakukan proses perpindahan *link* atau mengalami proses *handover*. Analisa ini dilakukan sesuai skenario 3. Dengan melakukan analisa ini, hasilnya diharapkan dapat mengetahui kualitas layanan yang diterima oleh *mobile node* pada saat mengalami *handover*.

Pengambilan data pada skenario 3 sedikit lebih berbeda. Pengambilan data tidak dilakukan dengan 3 jenis file yang berbeda, tetapi file yang digunakan adalah jenis file rar. Hal tersebut dilakukan untuk meminimalisasi terjadinya hank terus menerus pada jaringan *mobile Node*. Dari analisa ketika pengambilan data dilakukan dengan menggunakan file yang berbeda dengan ukuran file 32 MB keatas akan menyebabkan hank. Sedangkan untuk *handover* dibutuhkan file yang berukuran besar. Karena hal tersebut maka pengambilan data dilakukan pada file yang berukuran 10 MB dan 16 MB.

Pada skenario 3 ini, *handover* dilakukan ketika file telah di-*download* dari 50 sampai 60 %. Hal tersebut dilakukan agar dapat dilihat ketika *mobile node* mengalami *handover*. Gambar 4.20 merupakan penangkapan paket mipv6 ketika mengalami *handover*.

No. -	Time	Source	Destination	Protocol	Info
10651	14.209125	2001:db8:3c4d:5a::4	2001:db8:3c4d:5a::3	MIPv6	Binding Update
10661	15.225513	2001:db8:3c4d:5a::3	2001:db8:3c4d:5a::4	MIPv6	Binding Acknowledgement
10721	49.445385	2001:db8:3c4d:5a::4	2001:db8:3c4d:5a::3	MIPv6	Binding Update
10722	49.448299	2001:db8:3c4d:5a::3	2001:db8:3c4d:5a::4	MIPv6	Binding Acknowledgement
10768	66.524513	2001:db8:3c4d:5a::4	2001:db8:3c4d:5b::4	MIPv6	Home Test Init
10769	66.524825	2001:db8:3c4d:5c:21c:	2001:db8:3c4d:5b::4	MIPv6	Care-of Test Init
10774	66.528070	2001:db8:3c4d:5b::4	2001:db8:3c4d:5a::4	MIPv6	Home Test
10775	66.529064	2001:db8:3c4d:5b::4	2001:db8:3c4d:5c:21c:	MIPv6	Care-of Test
10792	66.543620	2001:db8:3c4d:5a::4	2001:db8:3c4d:5b::4	MIPv6	Binding Update

+ Frame 10651 (110 bytes on wire, 110 bytes captured)
 + Ethernet II, Src: IntelCor_aa:e5:db (00:1c:bf:aa:e5:db), Dst: 3Com_dc:c5:ae (00:04:75:dc:c5:ae)
 + Internet Protocol Version 6
 + Mobile IPv6 / Network Mobility

Gambar 4.20 Capture Wireshark saat *handover*

Gambar 4.20 merupakan penangkapan yang dilakukan pada file berukuran 16 MB pada saat *handover*.

4.5.1 Analisa *Throughput* (skenario 3)

Pengambilan data dari Wireshark dilakukan berdasarkan skenario 3, yaitu pada saat *mobile node* melakukan *handover* ke *foreign link*. *Mobile node* melakukan *handover* secara manual dengan mengkoneksikannya ke *foreign link*. Hasil penangkapan paket FTP pada Wireshark dapat dilihat pada Gambar 4.21 dan Hasil pengambilan data dari nilai *throughput* dapat dilihat pada Tabel 4.11.

Traffic	Captured	Displayed	Marked
Packets	10635	9	0
Between first and last packet	52.253 sec	48.651 sec	
Avg. packets/sec	203.528	0.185	
Avg. packet size	1046.451 bytes	125.111 bytes	
Bytes	11129006	1126	
(Avg. bytes/sec)	212981.805	23.145	
Avg. MBit/sec	1.704	0.000	

Gambar 4.21 *Throughput* pada *Summary* Wireshark (Skenario 3)

Tabel 4.11 Data Nilai *Throughput*

File Size (MBytes)	Throughput (Bytes/sec)
10	23.145
16	16.106

Pada Tabel 4.11, terlihat bahwa file ukuran 10 MB memiliki nilai *throughput* lebih besar dari file ukuran 16 MB. Jadi semakin besar ukuran file, maka semakin kecil nilai *throughput* yang didapat. Secara persentase, file ukuran 16 lebih kecil atau mengalami penurunan nilai *throughput* 30.41 % dari 10 MB.

4.5.2 Analisa *Transfer Time* (skenario 3)

Hasil penangkapan paket FTP pada Wireshark untuk skenario 3 dapat dilihat pada Gambar 4.22 dan Hasil pengambilan data rata – rata dari nilai *transfer time* dapat dilihat pada Tabel 4.12.

Traffic	Captured	Displayed	Marked
Packets	10635	9	0
Between first and last packet	52.253 sec	48.651 sec	
Avg. packets/sec	203.528	0.185	
Avg. packet size	1046.451 bytes	125.111 bytes	
Bytes	11129006	1126	
Avg. bytes/sec	212981.805	23.145	
Avg. MBit/sec	1.704	0.000	

Gambar 4.22 *Transfer Time* pada Summary Wireshark (Skenario 3)Tabel 4.12 Data Nilai *Transfer Time*

File Size (MBytes)	Transfer Time (s)
10	48.651
16	69.910

Pada Tabel 4.12, terlihat bahwa file ukuran 10 MB memiliki nilai *transfer time* lebih kecil dari file ukuran 16 MB. Jadi semakin besar ukuran file, maka semakin

besar nilai transfer time yang didapat. Secara persentase, file ukuran 16 mengalami kenaikan *transfer time* 43.70 % dari file ukuran 10 MB.

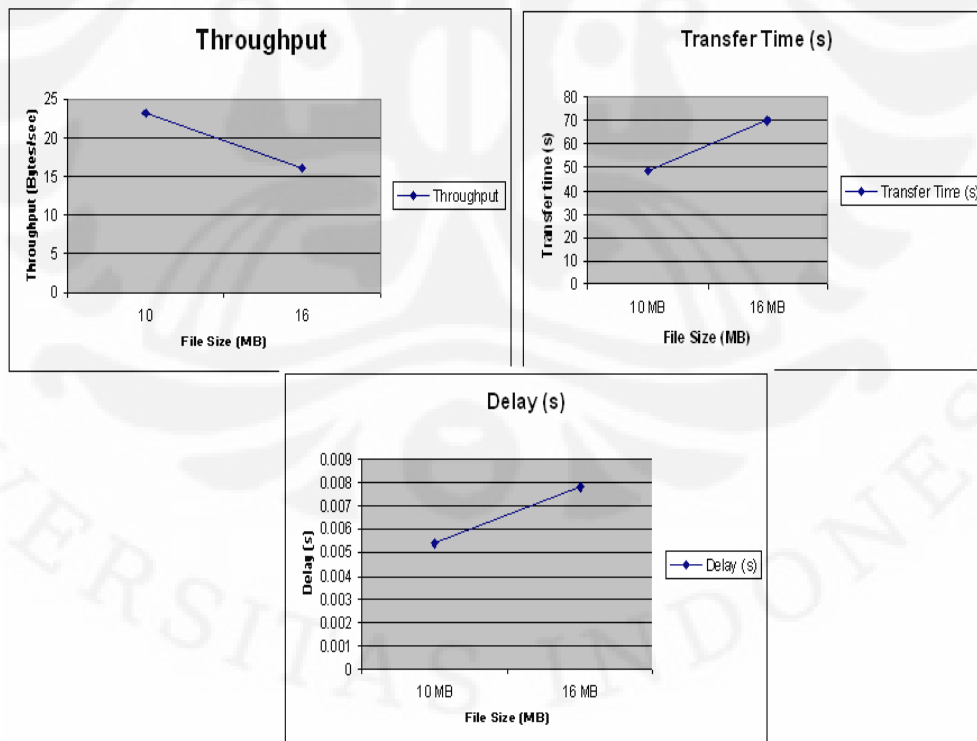
4.5.3 Analisa *Delay* (skenario 3)

Tabel 4.13 Data Nilai Perhitungan *Delay*

File Size (MBytes)	Delay (s)
10	0.0054
16	0.0078

Berdasarkan pada hasil perhitungan *delay* Tabel 4.13, terlihat bahwa file ukuran 10 MB memiliki nilai perhitungan *delay* lebih kecil dari file ukuran 16 MB. Jadi semakin besar ukuran file, maka semakin besar *delay* yang diterima. Secara persentase, file ukuran 16 mengalami kenaikan *delay* 44.4 % dari file ukuran 10 MB.

Berdasarkan analisa pada ketiga parameter tersebut, grafik analisa *throughput*, *delay* dan *transfer time* dapat dilihat pada Gambar 4.23 sebagai berikut.



Gambar 4.23 Grafik Analisa *Throughput*, *Transfer Time* dan *Delay* (Skenario 3)

Pada Gambar 4.23, sama halnya dengan skenario 1 dan skenario 2 grafik *throughput* mengalami penurunan pada saat ukuran file yang di-*download* semakin besar. Karena file yang digunakan pada skenario 3 yaitu proses *handover* ini lebih besar, maka nilai *throughput* yang muncul jauh lebih kecil dari skenario 1 dan 2. Jadi berdasarkan parameter *throughput*, performansi pada skenario 3 tidak lebih baik dari skenario 1 dan 2.

Pada grafik *transfer time*, semakin besar ukuran file maka semakin besar waktu yang digunakan untuk mentransfer. Karena file pada *handover* ini menggunakan file *download* yang jauh lebih besar, maka *transfer time* yang muncul jauh lebih besar dari skenario 1 dan 2.

Sedangkan pada grafik *delay*, semakin besar ukuran file maka semakin besar *delay* yang muncul. Pada skenario 3 dengan ukuran file yang lebih besar dapat dilihat bahwa nilai *delay* yang didapat adalah jauh lebih besar dari skenario 1 dan 2 dengan ukuran file yang lebih kecil. Sehingga dapat disimpulkan bahwa skenario 1 dan 2 lebih baik dari skenario 3.

BAB 5

KESIMPULAN

Berdasarkan hasil pengambilan data pada skenario – skenario yang ada, untuk penelitian ini didapatkan kesimpulan sebagai berikut :

1. Berdasarkan parameter *throughput*, file pdf pada skenario 2 memiliki nilai *throughput* lebih kecil 30.91 % dari file pdf pada skenario 1, file doc 21.47 % dan file jpg 15.22 %. Jadi bila dirata – ratakan, skenario 1 memiliki nilai *throughput* lebih besar 16.89 % dari pada skenario 2.
2. Berdasarkan parameter *transfer time*, file pdf pada skenario 1 memiliki nilai *transfer time* lebih kecil 66.67 % dari file pdf pada skenario 2, file doc 39.58 % dan file jpg 13.61 %. Jadi bila dirata – ratakan, skenario 1 memiliki nilai *transfer time* lebih kecil 36.38 % dari pada skenario 2.
3. Berdasarkan parameter delay, file pdf pada skenario 1 memiliki nilai *delay* lebih kecil 43.04 % dari file pdf pada skenario 2, file doc 16.36 % dan file jpg pada skenario 1 memiliki delay lebih besar 2.49 % dari skenario 2. Tetapi persentase rata – rata keseluruhan jenis file, skenario 1 memiliki nilai *delay* lebih kecil 16.8 % dari pada skenario 2
4. Pada skenario 3, file 10 Mbytes memiliki nilai *throughput* lebih besar 30.41 % *transfer time* lebih kecil 43.70 % dan delay lebih kecil 44.4 % dari pada file ukuran 16Mbytes.
5. Berdasarkan analisa pada parameter – parameter yang ada, perbedaan jenis file yang digunakan untuk transfer file ternyata dapat mempengaruhi performansi suatu jaringan.
6. Pada analisa keseluruhan, dapat dikatakan bahwa performa *mobile node* pada *home link* lebih baik dari pada *foreign link* dan *handover*.

DAFTAR REFERENSI

- [1]. "Hader Ipv6"
<http://www.maxim-ic.com/app-notes/index.mvp/id/4158>, diakses pada
September 2010
- [2]. "IPv6 over IPv4 tunneling"
<http://technet.microsoft.com/en-us/library/bb727021.aspx>, diakses pada
September 2010.
- [3]. Hasan, Fuad. "Implementasi Mobile IPv6 di PENS-ITS". Institut
Teknologi Sepuluh November, Surabaya 2008.
- [4]. "FTP". Wikipedia. 15 December 2010
<<http://en.wikipedia.org/wiki/FTP>>
- [5]. Miller, Mark A, P.E. "Implementing IPv6, Second Edition", Publish by
M&T Books. November 1997
- [6]. Perdana, Muamar Putra. "Analisa Performansi File Transfer Protocol Pada
Jaringan IPv6 Dengan Tunneling 6to4 dan ISATAP". Departemen Teknik
Elektro, Universitas Indonesia, Juni 2009.
- [7]. "Network Delay". Wikipedia. 26 October 2009
<http://en.wikipedia.org/wiki/Network_delay>
- [8]. "Transmission Delay." Wikipedia. 26 July 2010
<http://en.wikipedia.org/wiki/Transmission_delay>
- [9]. Davies, Joseph. "Understanding IPv6", Copyright © 2003 by Microsoft
Corporation.

LAMPIRAN 1 : Konfigurasi *Mobile Node*

1. Konfigurasi *Interface Mobile Node*

```
ifconfig wlan0 add 2001:0db8:3c4d:005a::4/64 up

echo "0" > /proc/sys/net/ipv6/conf/all/forwarding
echo "1" > /proc/sys/net/ipv6/conf/all/autoconf
echo "1" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "1" > /proc/sys/net/ipv6/conf/all/accept_redirects
```

2. Konfigurasi file “*etc/mip6d.conf*” *Mobile Node*

```
NodeConfig MN;
DebugLevel 7;
Interface "wlan0";
MnHomeLink "wlan0"
{
    HomeAgentAddress 2001:0db8:3c4d:005a::3;
    HomeAddress 2001:0db8:3c4d:005a::4/64;
}
UseMnHaIPsec disabled;

mip6d -c /etc/mip6d.conf
```

3. Konfigurasi file “*etc/network/interfaces*” *Mobile Node*

```
auto lo
iface lo inet loopback

auto wlan0
iface wlan0 inet dhcp

#iface wlan0 inet6 static
#address 2001:0db8:3c4d:005a::4
#netmask 64

wireless-essid home-network

#up echo 0 > /proc/sys/net/ipv6/conf/all/forwarding
#up echo 1 > /proc/sys/net/ipv6/conf/all/autoconf
#up echo 1 > /proc/sys/net/ipv6/conf/all/accept_ra
#up echo 1 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

LAMPIRAN 2 : Konfigurasi *Home Agent*

1. Konfigurasi *Interface Home Agent*

```
ifconfig eth1 inet6 add 2001:0db8:3c4d:005a::3/64

echo 1 >/proc/sys/net/ipv6/conf/all/forwarding
echo 1 >/proc/sys/net/ipv6/conf/all/proxy_ndp

ip route add ::/0 via 2001:0db8:3c4d:005a::2
```

2. Konfigurasi file “etc/mip6d.conf” *Home Agent*

```
NodeConfig HA;
DebugLevel 7;
Interface "eth1";
UseMnHaIPsec disabled;

mip6d -c /etc/mip6d.conf
```

3. Konfigurasi file “etc/radvd.conf” *Home Agent*

```
interface eth1
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 10;
    AdvHomeAgentInfo on;
    AdvHomeAgentFlag on;
    prefix 2001:0db8:3c4d:005a::3/64
    {
        AdvOnLink on;
        AdvAutonomous on;
        AdvRouterAddr on;
    };
};

radvd -C /etc/radvd.conf
```

LAMPIRAN 3 : Konfigurasi *Home Router*

1. Konfigurasi `etc/network/interface` *Home Router*

```
#the loopback network interfaces
auto lo
iface lo inet loopback

auto eth0

iface eth0 inet6 static
address 2001:0db8:3c4d:005b::2
netmask 64

auto eth6

iface eth6 inet6 static
address 2001:0db8:3c4d:005a::2
netmask 64

up echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
up echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
up echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
up echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects

up ip route add 2001:0db8:3c4d:005c::/64 via
2001:0db8:3c4d:005b::3
```

2. Konfigurasi "`rc.local`" *Home Router*

```
ifconfig eth0 inet6 add 2001:0db8:3c4d:005b::2/64 up
ifconfig eth6 inet6 add 2001:0db8:3c4d:005a::2/64 up

#route -A inet6 add 2001:0db8:3c4d:005c::/64 gw
2001:0db8:3c4d:005b::3 dev eth0

echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects

ip route add 2001:0db8:3c4d:005c::/64 via
2001:0db8:3c4d:005b::3

ip route add 2001:0db8:3c4d:005a::/64 via
2001:0db8:3c4d:005b::2
```


LAMPIRAN 4 : Konfigurasi *Foreign Router*

1. Konfigurasi `etc/network/interface` *Foreign Router*

```
auto lo
iface lo inet loopback

auto eth2
iface eth2 inet dhcp
iface eth2 inet6 static
address 2001:0db8:3c4d:005c::2
netmask 64

auto eth3
iface eth3 inet6 static
address 2001:0db8:3c4d:005b::3
netmask 64

up echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
up echo 0 > /proc/sys/net/ipv6/conf/all/autoconf
up echo 0 > /proc/sys/net/ipv6/conf/all/accept_ra
up echo 0 > /proc/sys/net/ipv6/conf/all/accept_redirects
```

2. Konfigurasi `rc.local` *Foreign Router*

```
ifconfig eth2 add 2001:0db8:3c4d:005c::2/64 up
ifconfig eth3 add 2001:0db8:3c4d:005b::3/64 up

route -A inet6 add 2001:0db8:3c4d:005a::/64 gw
2001:0db8:3c4d:005b::2 dev eth3

echo "1" > /proc/sys/net/ipv6/conf/all/forwarding
echo "0" > /proc/sys/net/ipv6/conf/all/autoconf
echo "0" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "0" > /proc/sys/net/ipv6/conf/all/accept_redirects

ip route add 2001:0db8:3c4d:005a::/64 via
2001:0db8:3c4d:005b::2
```

3. Konfigurasi file `etc/radvd.conf` *Foreign Router*

```
interface eth2
{
    AdvSendAdvert on;
    AdvIntervalOpt on;
```

LANJUTAN

```
MinRtrAdvInterval 1;
MaxRtrAdvInterval 3;

AdvHomeAgentFlag off;
prefix 2001:0db8:3c4d:005c::/64
{
  AdvRouterAddr on;
  AdvOnLink on;
  AdvAutonomous on;
};
};

radvd -C /etc/radvd.conf
```

LAMPIRAN 5 : Konfigurasi *Correspondent Node*

1. Konfigurasi *Interface Correspondent Node*

```
killall mip6d

ifconfig eth0 inet6 add 2001:0db8:3c4d:005b::4/64 up

echo "0" > /proc/sys/net/ipv6/conf/all/forwarding
echo "1" > /proc/sys/net/ipv6/conf/all/autoconf
echo "1" > /proc/sys/net/ipv6/conf/all/accept_ra
echo "1" > /proc/sys/net/ipv6/conf/all/accept_redirects

ip route add 2001:0db8:3c4d:005a::/64 via
2001:0db8:3c4d:005b::2 dev eth0
ip route add 2001:0db8:3c4d:005c::/64 via
2001:0db8:3c4d:005b::3 dev eth0

mip6d -c /etc/mip6d.conf
```

2. Konfigurasi file “*etc/mip6d.conf*” *Correspondent Node*

```
NodeConfig CN;
DebugLevel 7;
DoRouteOptimizationCN enabled;

mip6d -c /etc/mip6d.conf
```

LAMPIRAN 6 : Konfigurasi VSFTPD

```
# Run standalone? vsftpd can run either from an inetd or as a
standalone
# daemon started from an initscript.
listen=NO
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an IPv6
socket
# instead of an IPv4 one. This parameter and the listen parameter
are mutually
# exclusive.
listen_ipv6=YES
#
# Allow anonymous FTP? (Beware - allowed by default if you comment
this out).
anonymous_enable=NO
#
#the directory which vsftpd will try to change
#into after an anonymous login (Default = /home/ftp)
anon_root=/home/ftp
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change
this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files.
This only
# has an effect if the above global write enable is activated.
Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to
create
# new directories.
#anon_mkdir_write_enable=YES
# Activate directory messages - messages given to remote users
when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
```

LANJUTAN

```
# Make sure PORT transfer connections originate from port 20 (ftp-
data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be
owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The
default is shown below.
xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog
format
xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data
connection.
#data_connection_timeout=120
#nopriv_user=ftpsecure
#async_abor_enable=YES
# ASCII mangling is a horrible feature of the protocol.
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
ftpd_banner>Welcome to WINDA FTP service....
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#ls_recurse_enable=YES
secure_chroot_dir=/var/run/vsftpd
# This string is the name of the PAM service vsftpd will
use.
pam_service_name=vsftpd
#
# This option specifies the location of the RSA certificate
to #use for SSL encrypted connections.
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
```

LANJUTAN

```
# This option specifies the location of the RSA key to use
for SSL encrypted connections.
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key

max_clients=10
max_per_ip=2
```

