



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI JARINGAN VERTICAL HANDOVER MOBILE
IPV6 MENGGUNAKAN BIDIRECTIONAL TUNNELING DAN ROUTE
OPTIMIZATION DENGAN APLIKASI HTTP**

SKRIPSI

BAGUS WIRA REZA P

0806321511

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK KOMPUTER
JUNI 2012**



UNIVERSITAS INDONESIA

**ANALISA PERFORMANSI JARINGAN VERTICAL HANDOVER MOBILE
IPV6 MENGGUNAKAN BIDIRECTIONAL TUNNELING DAN ROUTE
OPTIMIZATION DENGAN APLIKASI HTTP**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

BAGUS WIRA REZA P

0806321511

FAKULTAS TEKNIK

PROGRAM STUDI TEKNIK KOMPUTER

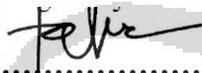
JUNI 2012

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Bagus Wira Reza P

NPM : 0806321511

Tanda Tangan : 

Tanggal: 26 Juni 2012

HALAMAN PENGESAHAN

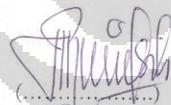
HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Bagus Wira Reza P
NPM : 0806321511
Program Studi : Teknik Komputer
Judul Skripsi : ANALISA PERFORMANSI JARINGAN
VERTICAL HANDOVER MOBILE IPV6
MENGUNAKAN BIDIRECTIONAL
TUNNELING DAN ROUTE OPTIMIZATION
DENGAN APLIKASI HTTP

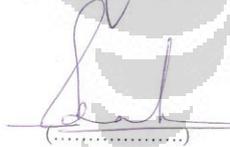
Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia

Pembimbing : Ir. Endang Sriningsih, M.T., Si.



(.....)

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



(.....)

Penguji 2 : Prima Dewi Purnamasari, S.T., M.T., M.Sc.



(.....)

Ditetapkan di : Depok

Tanggal : 26 Juni 2012

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya sehingga saya dapat menyelesaikan skripsi ini. Sebagaimana diketahui bahwa penulisan skripsi ini merupakan salah satu syarat dalam menuntaskan pendidikan Sarjana Teknik Komputer Jurusan Teknik Elektro Fakultas Teknik Universitas Indonesia.

Tidak lupa saya mengucapkan terima kasih kepada seluruh pihak yang telah membantu dalam proses skripsi maupun penyelesaian penulisan skripsi:

1. Ibu Ir. Endang Sriningsih M.T, selaku dosen pembimbing skripsi yang telah senantiasa menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penulisan skripsi;
2. Orang tua dan keluarga, yang telah memberikan dukungan moral;
3. Teman-teman kelompok skripsi, Ezra Diamond, James Daniel E, Rhaka Naufan A, Rian Constantino, Salman Alfarisyi, Suwega D, dan Yulius Dimas yang telah membantu dan mengingatkan saya selama pengerjaan skripsi;
4. Tb. Ari Wibawa M dan M. Charles Ringgo F.D sebagai sahabat yang mendukung dan memberikan semangat dalam pengerjaan skripsi;
5. Teman-teman seperjuangan Dept. Teknik Elektro angkatan 2008 yang telah bersama-sama melewati masa perkuliahan di Universitas Indonesia.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna dan mohon maaf apabila terjadi kesalahan penulisan nama ataupun gelar, untuk itu saya mengharapkan saran dan kritik yang bersifat membangun untuk kesempurnaan penulisan di masa yang akan datang. Akhir kata saya berharap semoga penulisan skripsi ini dapat bermanfaat bagi semua pihak.

Depok, 26 Juni 2012


Bagus Wira Reza P

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

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

Nama : Bagus Wira Reza P
NPM : 0806321511
Program Studi : Teknik Komputer
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi 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 JARINGAN VERTICAL HANDOVER MOBILE
IPV6 MENGGUNAKAN BIDIRECTIONAL TUNNELING DAN ROUTE
OPTIMIZATION DENGAN APLIKASI HTTP**

Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok,

Pada tanggal : 26 Juni 2012,

Yang menyatakan



(Bagus Wira Reza P)

ABSTRAK

Nama : Bagus Wira Reza P
Program Studi : Teknik Komputer
Judul : ANALISA PERFORMANSI JARINGAN VERTICAL
HANDOVER MOBILE IPV6 MENGGUNAKAN
BIDIRECTIONAL TUNNELING DAN ROUTE
OPTIMIZATION DENGAN APLIKASI HTTP

Mobile IPv6 memiliki dua metode dalam komunikasi antara *mobile node* dengan *correspondent node*, yaitu *Bidirectional Tunneling* dan *Route Optimization*. *Bidirectional Tunneling* tidak membutuhkan bantuan *correspondent node* dan dapat tersedia walaupun *mobile node* tidak mendaftarkan *binding* terbarunya terlebih dahulu. *Route Optimization* memerlukan dukungan *mobile node* untuk mendaftarkan *binding*-nya pada *correspondent node*. Proses *handover* pada *MIPv6* dibagi menjadi dua, yakni *horizontal handover* dan *vertical handover*. *Horizontal handover* merupakan *handover* yang terjadi pada saat *mobile node* berpindah *access point* namun masih berada pada *Home Network* yang sama, sedangkan *vertical handover* merupakan *handover* yang terjadi pada saat *mobile node* berpindah dari *Home Network* ke *Foreign Network*. Akan dijelaskan mengenai perbandingan performansi *throughput*, *packet loss*, dan *delay* antara *Bidirectional Tunneling* dengan *Route Optimization* menggunakan aplikasi HTTP. Hasil pengukuran pada *home link*, didapatkan *throughput* pada *bidirectional* dan *route optimization* memiliki perbandingan 15.7%, pada pengukuran *packet loss* memiliki perbandingan 0.007%, dan pada pengukuran *delay* memiliki perbandingan 1.77%. Hasil pengukuran pada *foreign link*, didapatkan *throughput* pada *route optimization* 36.77% lebih cepat dibandingkan *bidirectional tunneling*. *Packet loss* pada *route optimization* 1.76% lebih sedikit dibandingkan *bidirectional tunneling*, dan pengukuran *delay* pada *route optimization* 41.45 lebih cepat dibandingkan *bidirectional tunneling*.

Kata kunci: *Bidirectional Tunneling*, *Route Optimization*, *Handover*, *Mobile IP*, HTML, parameter QoS.

ABSTRACT

Name : Bagus Wira Reza P
Study Program : Teknik Komputer
Title : ANALYSIS OF NETWORK PERFORMANCE VERTICAL
MOBILE IPV6 HANDOVER USING BIDIRECTIONAL
TUNNELING AND ROUTE OPTIMIZATION WITH HTTP
APPLICATION

Mobile IPv6 has two methods in communication between mobile node with correspondent node, those are Bidirectional Tunneling and Route Optimization. Bidirectional Tunneling does not need help of correspondent node and no need registration of binding from mobile node. Route Optimization requires support of mobile node to register the binding on correspondent node. MIPv6 handover process is divided into two, those are horizontal handover and vertical handover. Horizontal handover is a handover that occurs when the mobile node moves to the other access point but still in the same Home Network, while vertical handover is a handover that occurs when the mobile node moves from Home Network to Foreign Network. Will be explained about the comparative performance of throughput, packet loss, and delay between Bidirectional Tunneling with Route Optimization using HTTP applications. Measurement result on home link, shows the throughput on bidirectional tunneling and route optimization having comparison about 15.7%, packet loss about 0.007%, and delay about 1.77%. Measurement result on foreign link, shows that throughput on route optimization is 36.77% faster than on bidirectional tunneling. Packet loss on route optimization is 1.76% lesser than bidirectional tunneling, and measurement of delay on route optimization is 41.45% faster than bidirectional tunneling.

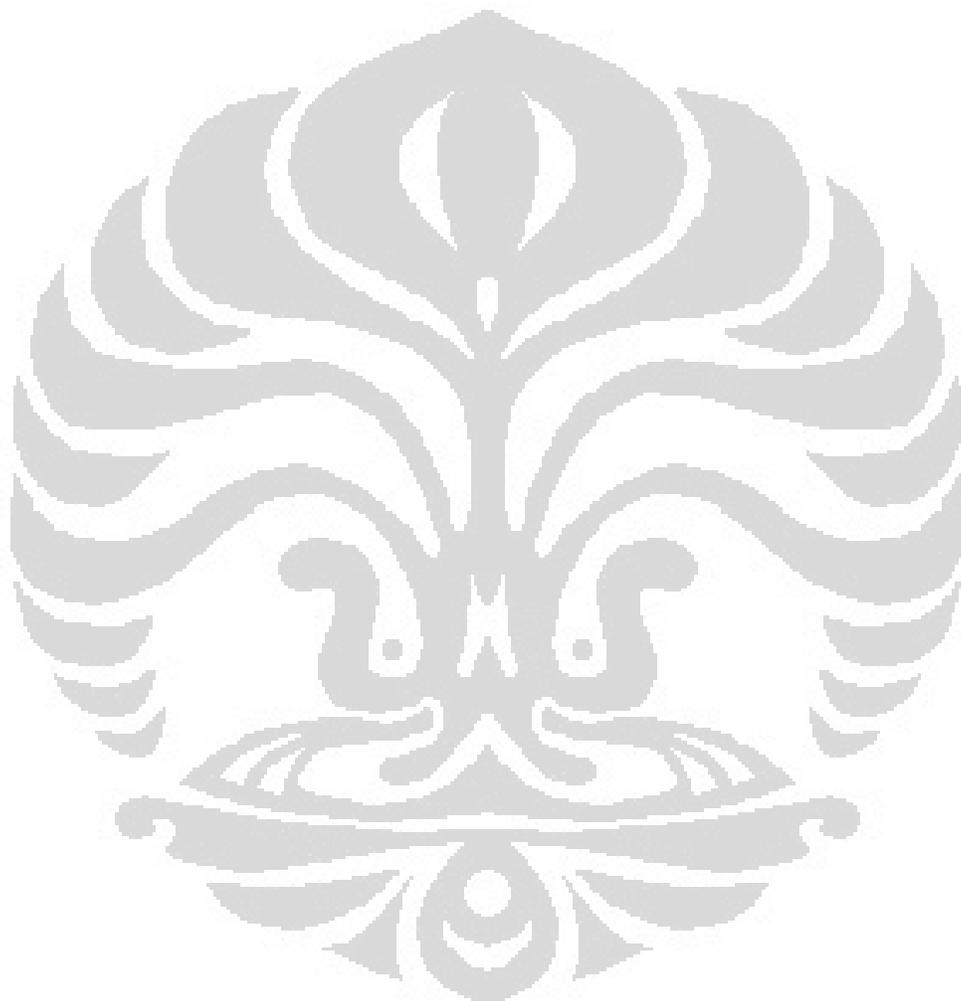
Keywords: Bidirectional Tunneling, Route Optimization, Handover, Mobile IP, HTML, QoS parameters

DAFTAR ISI

SKRIPSI.....	i
HALAMAN PERNYATAAN ORISINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
SKRIPSI UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR SINGKATAN.....	xiv
BAB 1.....	1
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Metode Penelitian.....	2
1.4. Batasan Masalah.....	3
1.5. Sistematika Penulisan.....	3
BAB 2.....	5
2.1. <i>Internet Protocol Version 6</i>	5
2.1.1. <i>Datagram IPv6</i>	7
2.1.2. <i>Format Header IPv6</i>	8
2.1.3. <i>Format Alamat IPv6</i>	9
2.1.4. <i>Format Prefiks IPv6</i>	10
2.1.5. <i>Protokol-protokol pada IPv6 Standar</i>	11
2.2. <i>Mobile Internet Protocol Version 6</i>	12
2.2.1. <i>MIPv6 Handover</i>	13
2.2.2. <i>Proses Handover</i>	15
2.2.3. <i>Prosedur Proses Handover MIPv6</i>	16
2.2.4. <i>Mekanisme MIPv6</i>	19
2.3. <i>Hypertext Transfer Protocol (HTTP)</i>	21
2.3.1. <i>Hypertext Transfer Protocol Secure (HTTPS)</i>	22

BAB 3	23
3.1 Jaringan Topologi.....	23
3.2 Spesifikasi Sistem.....	24
3.2.1 Spesifikasi Perangkat Keras.....	24
3.2.2 Spesifikasi Perangkat Lunak.....	25
3.3 Skenario Pengukuran Pada Jaringan Vertical Mobile IPv6	27
3.4 Pembuatan sistem	31
3.4.1 Instalasi Kernel Linux-2.6.32.32	31
3.4.2 Instalasi UMIP 4.0 dan Radvd.....	34
3.4.3 Instalasi LAMP.....	34
3.4.4 Konfigurasi <i>Node</i>	35
3.4.5 Konfigurasi <i>Multi Generator</i>	36
BAB 4	39
4.1 Pengukuran Jaringan <i>Mobile IPv6</i>	39
4.2 Analisa <i>Home Network</i>	40
4.2.1 Analisa <i>Throughput</i>	41
4.2.2 Analisa <i>Packet Loss</i>	44
4.2.3 Analisa <i>Delay</i>	48
4.3 Analisa <i>Foreign Network</i>	51
4.3.1 Analisa <i>Throughput</i>	52
4.3.2 Analisa <i>Packet Loss</i>	54
4.3.3 Analisa <i>Delay</i>	57
4.4 Analisa Perbandingan Skenario 1 dan Skenario 3.....	60
4.4.1 Analisa Perbandingan <i>Throughput</i>	60
4.4.2 Analisa Perbandingan <i>Packet loss</i>	62
4.4.3 Analisa Perbandingan <i>Delay</i>	63
4.5 Analisa Perbandingan Skenario 2 dan Skenario 4.....	65
4.5.1 Analisa Perbandingan <i>Throughput</i>	65
4.5.2 Analisa Perbandingan <i>Packet Loss</i>	66
4.5.3 Analisa Perbandingan <i>Delay</i>	68
BAB 5	71

DAFTAR ACUAN 73
LAMPIRAN 74



DAFTAR GAMBAR

Gambar 2.1 <i>Datagram IPv4 & IPv6</i>	7
Gambar 2.2 <i>Format Header IPv4</i>	8
Gambar 2.3 <i>Format Header IPv6</i>	8
Gambar 2.5 <i>Horizontal Handover</i>	14
Gambar 2.6 <i>Vertical Handover</i> pada ISP yang sama.....	14
Gambar 2.7 <i>Vertical Handover</i> pada ISP yang berbeda	15
Gambar 2.3 <i>Bidirectional Tunneling</i>	20
Gambar 2.4 <i>Route Optimization</i>	20
Gambar 2.8 <i>Request dan response HTTP</i>	22
Gambar 3.1 Topologi Jaringan <i>Vertical Mobile IPv6</i>	23
Gambar 3.2 Skenario 1.....	28
Gambar 3.3 Skenario 2.....	29
Gambar 3.3 Skenario 3.....	29
Gambar 3.4 Skenario 4.....	30
Gambar 4.1 <i>Tampilan web</i>	40
Gambar 4.2 <i>Throughput</i> pada <i>summary wireshark</i> (skenario 1).....	41
Gambar 4.3 <i>Throughput</i> pada <i>summary wireshark</i> (skenario 3).....	43
Gambar 4.4 <i>Packet loss</i> pada <i>capture wireshark</i> (skenario 1).....	45
Gambar 4.5 <i>Packet loss</i> pada <i>summary wireshark</i> (skenario 1)	45
Gambar 4.6 <i>Packet loss</i> pada <i>capture wireshark</i> (skenario 3).....	47
Gambar 4.7 <i>Packet loss</i> pada <i>summary wireshark</i> (skenario 3)	47
Gambar 4.8 <i>Delay</i> pada <i>summary wireshark</i> (skenario1).....	49
Gambar 4.9 <i>Delay</i> pada <i>summary wireshark</i> (skenario3).....	50
Gambar 4.10 <i>Throughput</i> pada <i>summary wireshark</i> (skenario 2).....	52
Gambar 4.11 <i>Throughput</i> pada <i>summary wireshark</i> (skenario 4).....	53
Gambar 4.12 <i>Packet loss</i> pada <i>capture wireshark</i> (skenario 2).....	54
Gambar 4.13 <i>Packet loss</i> pada <i>summary wireshark</i> (skenario 2)	55
Gambar 4.14 <i>Packet loss</i> pada <i>capture wireshark</i> (skenario 4).....	56
Gambar 4.15 <i>Packet loss</i> pada <i>summary wireshark</i> (skenario 4)	56
Gambar 4.16 <i>Delay</i> pada <i>summary wireshark</i> (skenario 2).....	58

Gambar 4.17 <i>Delay</i> pada <i>summary wireshark</i> (skenario 4).....	59
Gambar 4.18 Perbandingan <i>throughput</i> pada skenario 1 dan skenario 3.....	61
Gambar 4.19 Perbandingan <i>packet loss</i> pada skenario 1 dan skenario 3.....	62
Gambar 4.20 Perbandingan <i>delay</i> pada skenario 1 dan skenario 3.....	64
Gambar 4.21 Perbandingan <i>throughput</i> pada skenario 2 dan skenario 4.....	65
Gambar 4.22 Perbandingan <i>packet loss</i> pada skenario 2 dan skenario 4.....	67
Gambar 4.23 Perbandingan <i>delay</i> pada skenario 2 dan skenario 4.....	69
Gambar B.1 Grafik perbandingan <i>throughput</i> skenario 1.....	77
Gambar B.2 Grafik perbandingan <i>throughput</i> skenario 3.....	77
Gambar B.3 Grafik perbandingan <i>packet loss</i> skenario 1.....	78
Gambar B.4 Grafik perbandingan <i>packet loss</i> skenario 3.....	78
Gambar B.5 Grafik perbandingan <i>delay</i> skenario 1.....	79
Gambar B.6 Grafik perbandingan <i>delay</i> skenario 3.....	79
Gambar B.7 Grafik perbandingan <i>throughput</i> pada skenario 2.....	80
Gambar B.8 Grafik perbandingan <i>throughput</i> pada skenario 4.....	80
Gambar B.9 Grafik perbandingan <i>packet loss</i> pada skenario 2.....	81
Gambar B.10 Grafik perbandingan <i>packet loss</i> pada skenario 4.....	81
Gambar B.11 Grafik perbandingan <i>delay</i> pada skenario 2.....	82
Gambar B.12 Grafik perbandingan <i>delay</i> pada skenario 4.....	82

DAFTAR TABEL

Tabel 2.1 Alamat asli IPv6.....	10
Tabel 2.2 Alamat asli yang disederhanakan dan alamat setelah dikompres	10
Tabel 4.1 Data nilai rata-rata <i>throughput</i> pada skenario 1	42
Tabel 4.2 Data nilai rata-rata <i>throughput</i> pada skenario 3	44
Tabel 4.3 Data nilai rata-rata <i>packet loss</i> pada skenario 1	46
Tabel 4.4 Data nilai rata-rata <i>packet loss</i> pada skenario 3	48
Tabel 4.5 Data nilai rata-rata <i>delay</i> pada skenario 1	49
Tabel 4.6 Data nilai rata-rata <i>delay</i> pada skenario 3	51
Tabel 4.7 Data nilai rata-rata <i>throughput</i> pada skenario 2	52
Tabel 4.8 Data nilai rata-rata <i>throughput</i> pada skenario 4	54
Tabel 4.8 Data nilai rata-rata <i>packet loss</i> pada skenario 2	55
Tabel 4.10 Data nilai rata-rata <i>packet loss</i> pada skenario 4	57
Tabel 4.11 Data nilai rata-rata <i>delay</i> pada skenario 2	58
Tabel 4.12 Data nilai rata-rata <i>delay</i> pada skenario 4	60
Tabel A.1 Hasil Pengukuran <i>Throughput</i> pada <i>Home Network</i>	74
Tabel A.2 Hasil Pengukuran <i>Throughput</i> pada <i>Foreign Network</i>	74
Tabel A.3 Hasil Pengukuran <i>Packet loss</i> pada <i>Home Network</i>	75
Tabel A.4 Hasil Pengukuran <i>Packet loss</i> pada <i>Foreign Network</i>	75
Tabel A.5 Hasil Pengukuran <i>Delay</i> pada <i>Home Network</i>	76
Tabel A.6 Hasil Pengukuran <i>Delay</i> pada <i>Foreign Network</i>	76

DAFTAR SINGKATAN

CN: *Correspondent Node*

CoA: *Care of Address*

DHCP: *Dynamic Host*

HA: *Home Agent*

IP: *Internet Protocol*

MAC: *Media Access Control*

MN: *Mobile Node*

MTU: *Maximum Transmission Unit*

NAR: *Next Access Router*

PAR: *Previous Access Router*

TCP: *Transfer Control Protocol*

UDP: *User Datagram Protocol*



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini perkembangan teknologi dan informasi semakin pesat dan membawa berbagai dampak pada kehidupan manusia. Internet adalah salah satu teknologi yang berkembang pesat di mana dapat melebihi perkembangan teknologi-teknologi lain. Tanpa disadari internet sudah menjadi bagian hidup dari setiap kegiatan yang dilakukan manusia. Seiring dengan meningkatnya penggunaan internet, hal ini menyebabkan banyaknya permintaan alokasi alamat IP. IPv4 atau *Internet Protokol versi 4* adalah protokol standar yang digunakan untuk internet. IPv4 mempunyai jumlah alamat IP sebanyak 2^{32} bits atau 4.294.967.296.

Namun terdapat beberapa faktor yang kurang diperhatikan dan diantisipasi pada saat pengembangan IPv4 sehingga menjadikan kelemahan pada IPv4. Kenyataan yang dihadapi saat ini adalah tinggal sedikitnya persediaan alamat IPv4 yang dapat dialokasikan, *routing tables* yang semakin besar akibat banyaknya alamat IPv4 yang digunakan. *Network Address Translator* atau NAT dapat digunakan untuk mengatasi masalah keterbatasan alamat IP namun NAT tidak dapat dijadikan solusi akhir terhadap masalah tersebut. Berbagai negara sudah menyadari akan masalah ini dan mengembangkan IP *next generation*(IPng).

IPng diharapkan mampu mengatasi masalah persediaan alamat IP pada IPv4. IPng atau yang saat ini dikenal sebagai IPv6 (internet protokol versi 6) telah dirancang sedemikian rupa sehingga dapat melebihi kemampuan IPv4. Ipv6 memiliki jumlah alamat IP sebanyak 2^{128} bits atau $3,4 \times 10^{38}$. Selain itu IPv6 memiliki beberapa keunggulan lain dibanding IPv4, yaitu keamanan jaringan yang terintergrasi, kemampuan *multicasting*, *stateless address configuration*, *header* yang lebih sederhana, mobilitas, dan adanya beberapa *option* tambahan.

Internet protocol juga memiliki kemampuan dalam menyediakan konektivitas tanpa terputus yang memungkinkan *host* tetap terkoneksi dengan internet walaupun sudah pindah ke *subnet* lain, yang disebut *mobile IP*. *Mobile IP* terdapat baik di IPv4 maupun IPv6, akan tetapi terdapat beberapa masalah pada *mobile IPv4*. Teknologi *mobile IP* dikembangkan untuk memenuhi kebutuhan *mobile users* yang memiliki mobilitas tinggi dan membutuhkan akses internet yang tidak terbatas.

1.2. Tujuan

Tujuan penulisan skripsi ini adalah untuk mengukur performansi sebuah jaringan *mobile IPv6* dengan merancang jaringan *mobile IPv6* terlebih dahulu menggunakan *vertical handover* dengan aplikasi *web server/HTTP (Hypertext Transfer Protocol)*. Akan diimplementasikan dua tipe metode komunikasi antara *mobile node* dengan *correspondent node* yaitu *Bidirectional Tunneling* dan *Route Optimization*.

1.3. Metode Penelitian

Metode penelitian yang digunakan pada penulisan skripsi adalah

1. Studi Pustaka

Pemahaman tentang bagaimana konsep jaringan dan pemahaman tentang IPv6 serta bagaimana konsep *mobile IP* dan metode kerja komunikasi antar *mobile node* dan *correspondent node*.

2. Perancangan dan Implementasi

Perancangan jaringan meliputi perancangan sebuah jaringan *mobile IPv6* menggunakan *vertical handover*. Setelah perancangan jaringan selesai dilakukan, kemudian diimplementasikan dua tipe komunikasi *mobile node* dengan *correspondent node* yang berbeda. Implementasi sistem akan dibuat berdasarkan perangkat keras dan perangkat lunak yang mendukung *mobile IPv6*.

3. Pengukuran dan Analisa

Data yang akan diukur berupa parameter-parameter QoS dengan beberapa skenario yang berbeda-beda. Hal ini dilakukan untuk mendapatkan pengukuran data yang bervariasi sehingga dapat diketahui pengaruh dari mobilitas *mobile* IPv6 yang kemudian akan dianalisa lebih lanjut.

1.4. Batasan Masalah

Permasalahan akan difokuskan pada pembahasan pengukuran dan perbandingan performansi *throughput*, *packet loss*, dan *delay* pada aplikasi HTTP yang menggunakan *vertical mobile* IPv6. Metode yang digunakan untuk komunikasi antara *mobile node* dengan *correspondent node* yang akan digunakan adalah *bidirectional tunneling* dan *route optimization*.

1.5. Sistematika Penulisan

Skripsi ini tersusun atas beberapa bab pembahasan. Sistematika pembahasan tersebut adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi penjelasan singkat mengenai latar belakang permasalahan, tujuan, metode penelitian, batasan masalah dan sistematika penulisan.

BAB II IPV6, MOBILE IPV6 DAN HTTP

Bab ini berisi pembahasan secara garis besar mengenai jaringan, IPv6, *mobile* IPv6 serta fitur-fitur yang terdapat pada *mobile* IPv6.

BAB III PERANCANGAN DAN IMPLEMENTASI JARINGAN VERTICAL HANDOVER MOBILE IPV6 UNTUK WEB SERVER (HTTP)

Bab ini berisi rancangan sistem *mobile* IPv6 yang akan dibangun serta penjelasan mengenai komponen-komponen sistem yang digunakan

BAB IV PENGUKURAN DAN ANALISA PARAMETER QoS JARINGAN MOBILE IPV6

Bab ini berisi pengukuran parameter-parameter QoS dan analisa mengenai data yang telah didapatkan.

BAB V KESIMPULAN

Bab ini berisi kesimpulan tentang hasil akhir dari pengujian yang telah didapat.



BAB 2

IPV6, MOBILE IPV6 DAN HTTP

2.1. *Internet Protocol Version 6*

IPv6 adalah *protocol next generation* yang didesain sebagai pengganti IPv4 . Terjadinya transisi antara IPv4 dengan IPv6 disebabkan oleh menipisnya persediaan alamat IP pada IPv4 dan juga terdapatnya beberapa kelemahan pada IPv4. Perubahan dari IPv4 ke IPv6 dikelompokkan ke dalam beberapa kategori, yaitu:

1. Kapasitas perluasan alamat

IPv6 meningkatkan jumlah ukuran dan jumlah alamat yang mampu didukung, yang tadinya dari 32 bit pada IPv4 menjadi 128 bit pada IPv6. Terjadinya peningkatan jumlah alamat IP ini bertujuan untuk mengatasi masalah yang sedang terjadi pada IPv4.

2. Penyederhanaan *format header*

Header pada IPv6 mengalami beberapa perubahan dan penyederhanaan dari *header* IPv4. Beberapa *header* IPv4 dihilangkan atau dijadikan sebagai *header* pilihan. Hal ini dilakukan untuk mengurangi biaya proses dan mengatasi masalah *bandwidth* pada *header* IPv6, sehingga proses *header* pada paket IPv6 dapat dilakukan secara efisien.

3. Perubahan proses *header checksum*

Proses *header checksum* tidak dilakukan pada tingkat akhir lagi, melainkan secara *end to end*. Isec menjamin keamanan pada proses ini.

6. Fitur *stateless auto configuration*

Memiliki fitur *stateless auto configuration* dimana ketika sebuah *host* terhubung ke sebuah jaringan, konfigurasi dilakukan secara otomatis. Berbeda dengan IPv4 yang dilakukan secara manual.

5. *Built in security*

Ipssec memberikan dukungan pada keamanan jaringan dan menawarkan interoperabilitas antara implementasi IPv6 yang berbeda.

6. Ekstensibilitas

IPv6 dapat dengan mudah ditambahkan fitur baru dengan menambahkan *header* ekstensi setelah *header* IPv6. Ukuran dari *header* ekstensi Ipv6 ini hanya dibatasi oleh ukuran dari paket Ipv6 .

7. Kualitas layanan berdasarkan *best level of effort*

Memakai mekanisme *best level of effort* yang memastikan kualitas layanan. Penentuan prioritas pengiriman paket data berdasarkan kebutuhan akan kecepatan tinggi atau tingkat *delay* tinggi.

Alamat IPv6 memiliki tiga jenis pengalamatan:

1. *Unicast*

Alamat yang menyediakan komunikasi secara *point to point* secara langsung antara dua *host* dalam sebuah jaringan. Alamat *unicast* dibagi menjadi tiga jenis, yaitu alamat *link lokal*, alamat *site lokal*, alamat *global*.

- a. Alamat *link lokal* adalah alamat yang digunakan pada satu *link* yaitu jaringan lokal yang saling terhubung dalam satu *level* atau *subnet*.

- b. Alamat *site lokal* mirip dengan alamat *private* yang dipakai terbatas di dalam satu *site* sehingga terbatas penggunaannya hanya dalam satu *site* dan tidak dapat digunakan untuk mengirimkan alamat diluar *site*.
- c. Alamat *global* adalah alamat yang dipakai untuk *Internet Service Provider*.

2. Multicast

Paket yang dikirimkan ke alamat ini akan dikirimkan ke semua *interface* yang ditunjukkan oleh alamat ini. Alamat *multicast* didesain untuk menggantikan alamat *broadcast* pada IPv4 yang banyak mengkonsumsi *bandwidth*.

3. Anycast

Alamat yang menunjukkan beberapa *interface*. Paket akan dikirimkan ke alamat yang akan dikirimkan ke salah satu alamat *interface* yang paling dekat dengan *router*. Alamat *anycast* tidak memiliki alokasi khusus, karena jika beberapa *node* atau *interface* diberikan prefiks yang sama maka alamat tersebut merupakan alamat *anycast*.

2.1.1. Datagram IPv6

Datagram IPv6 berbeda dengan *datagram* pada IPv4. Perbedaan tersebut terdapat pada tambahan *extension header*. Secara umum *datagram* IPv4 terdiri dari *IP header* dan *Data*, sedangkan *datagram* IPv6 terdiri dari *IP header*, *Extension header* dan *Data*. *Datagram* IPv4 dan IPv6 dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Datagram* IPv4 & IPv6 [4]

Penjelasan *datagram* IPv6 adalah sebagai berikut:

- *IP Header*, 40 bit yang memiliki satu komponen per *datagram*. Berisikan informasi penting mengenai paket yang dikirimkan diantaranya sumber dan tujuan alamat.
- *Extension Header*, terdiri dari 0 atau lebih komponen. Berisikan fitur-fitur tambahan. Satu fitur tambahan berarti satu komponen *Extension Header*. Adapun fitur-fitur tambahan antara lain: fragmentasi dan keamanan.
- *Data*, satu komponen per *datagram*. Muatan dari *upper layer* yang akan dikirimkan dalam *datagram* seperti, ICMPv6, pesan UDP atau sebuah segmen TCP.

2.1.2. Format Header IPv6

Header pada IPv6 mengalami penyederhanaan dari *header* IPv4. Perbandingan antara *format header* IPv4 dengan IPv6 dapat dilihat pada Gambar 2.2 dan Gambar 2.3.

Ver.	<i>header</i>	TOS	<i>Total length</i>	
<i>Identification</i>			<i>flag</i>	<i>Fragment offset</i>
TTL		<i>Protocol</i>	<i>Checksum</i>	
<i>32 bit Source Address</i>				
<i>32 bit Destination Address</i>				

Gambar 2.2 Format Header IPv4 [4]

Ver.	<i>TrafficClass</i>	<i>Flow</i>		
<i>Payload Length</i>		<i>Next Header</i>	<i>Hop Limit</i>	
<i>128 bit Source Address</i>				
<i>128 bit Destination Address</i>				

Gambar 2.3 Format Header IPv6 [4]

Tampak dari kedua gambar diatas perbedaan antara *header* IPv6 dengan *header* IPv4. Beberapa opsi diganti dan beberapa opsi lain dipertahankan.

Penjelasan *format header* IPv6 ,sebagai berikut:

- *Version*, 4-bit versi IPv6
- *Traffic Class*, 8-bit *field traffic class*
- *Payload Length*, 16-bit *unsigned integer*, panjang dari *payload* IPv6
- *Next Header*, 8-bit *selector*. Mengidentifikasi tipe *header* yang langsung mengikuti *header* IPv6. Menggunakan nilai yang sama seperti *field* protokol IPv4
- *Hop Limit*, 8 bit *unsigned integer*, dikurangi dengan 1 oleh setiap *node* yang meneruskan paket
- *Source Address*, 128 bit alamat asal paket
- *Destination Address*, 128 bit alamat tujuan paket

2.1.3. *Format Alamat IPv6*

Alamat IPv6 terdiri dari 128 bit dan dibagi ke dalam 8 blok berukuran 16 bit, yang dikonversikan ke dalam heksadesimal berukuran 4 digit. Setiap blok akan dipisahkan dengan karakter “:”. *Format* notasi pada IPv6 disebut *colon-hexadecimal format*. Berikut adalah contoh alamat IPv6:

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

Alamat IPv6 dapat disederhanakan dengan menghilangkan angka 0 pada awal setiap blok dengan menyisakan satu *digit* terakhir. Setelah angka 0 dihilangkan maka alamat IPv6 diatas akan menjadi:

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

Konversi alamat IPv6 memperbolehkan untuk penyederhanaan lebih jauh lagi, dengan cara membuang karakter 0 pada sebuah alamat yang memiliki banyak angka 0. Alamat tersebut dapat dinotasikan dengan dua karakter titik dua, "::". Tabel 2.1 dan Tabel 2.2 dibawah akan menunjukkan hasil penyederhanaan alamat IPv6:

Tabel 2.1 Alamat asli IPv6

Alamat asli
DA6C:0000:0000:0000:03FF:00FF:FC5B:2A6E
FC02:0000:0000:0000:0000:0000:0000:0008

Tabel 2.2 Alamat asli yang disederhanakan dan alamat setelah dikompres

Alamat asli yang disederhanakan	Alamat setelah dikompres
DA6C:0:0:0:3FF:FF:FC5B:2A6E	DA6C::3FF:FF:FC5B:2A6E
FC02:0:0:0:0:0:0:8	FC02::8

2.1.4. Format Prefiks IPv6

Berbeda dengan IPv4 yang memiliki prefiks yang merujuk pada *subnet mask*, IPv6 pun memiliki prefiks hanya saja tidak digunakan sebagai hal yang merujuk pada *subnet mask* karena IPv6 tidak mendukung *subnet mask*. Prefiks pada IPv6 dapat dinotasikan dengan cara yang sama seperti halnya pada prefiks IPv4, yaitu **[alamat]/[angka panjang prefiks]**. Panjang prefiks menentukan jumlah bit terbesar paling kiri yang membuat prefiks subnet, sebagai contoh:

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

Pada contoh diatas, 64 bit pertama dari alamat tersebut dinyatakan sebagai prefiks alamat, sementara 64 bit sisanya dinyatakan sebagai *interface ID*.

2.1.5. Protokol-protokol pada IPv6 Standar

Protokol ini memecahkan beberapa masalah yang berhubungan dengan interaksi antara *node-node* yang terhubung dalam *network* yang sama. Protokol ini mendefinisikan mekanisme dalam memecahkan masalah tersebut:

1. *Router Discovery*, bagaimana *host-host* mencari *router* yang berkoneksi pada *link*
2. *Prefix Discovery*, bagaimana *host-host* menemukan alamat-alamat prefix yang merupakan alamat atau pengidentifikasi *link* tempat *host-host* tersebut saling terinterkoneksi
3. *Parameter Discovery*, bagaimana *node* mempelajari parameter-parameter pada *link* seperti *link MTU (Maximum Transmission Unit)* atau parameter-parameter internet seperti jumlah batasan *hop* yang akan ditempatkan pada paket yang akan dikirimkan
4. *Address Autoconfiguration*, *node-node* secara otomatis akan mengkonfigurasi alamat IPv6 untuk *interface*.
5. *Address Resolution*, *node-node* mencari alamat *link layer* dari *node* yang akan dituju yang masih berada pada *link* yang sama hanya dengan diberikan alamat IP *node* tujuannya saja
6. *Next Hop*, algoritma untuk memetakan alamat IPv6 dari *node* tujuan ke dalam alamat IPv6 *node* tetangga. Trafik untuk *node* tujuan tersebut akan dikirimkan ke *node* tetangga tersebut. *Next hop* dapat berupa *router* atau *host* itu sendiri.
7. *Neighbor Unreachability Detection*, *node* mempelajari bahwa salah satu tetangga sudah tidak aktif lagi.
8. *Duplicate Address Detection*, *node* mempelajari bahwa alamat yang ingin digunakan masih tersedia (tidak sedang digunakan oleh *node* lain)

9. *Redirect, router* memberitahu *host* tentang *node* pertama jalur mana yang baik digunakan sebagai *hop* berikutnya untuk mencapai tujuannya.

2.2. *Mobile Internet Protocol Version 6*

Mobile IPv6 adalah standar yang dibuat oleh IETF yang didefinisikan pada RFC 3775. Tujuan utama dari *MIPv6* adalah *mobile node* tidak perlu mengganti alamat IP saat berpindah jaringan dan dapat berpindah dari satu jaringan ke jaringan lainnya dengan tetap mempertahankan koneksi jaringan.

MIPv6 dirancang untuk mengatasi masalah pada jaringan IPv6 yang tidak dapat mempertahankan koneksi jaringan saat berpindah jaringan. Terdapat beberapa komponen yang dimiliki oleh *mobile IPv6*, diantaranya:

1. *Foreign Link*, mendefinisikan *link* yang bukan merupakan *MN home link*.
2. *Home Address*, merupakan alamat IP yang diberikan ke *MN* sebagai identitas unik.
3. *Home Link*, merupakan *sub network* dari *home agent*.
4. *Mobile Node (MN)*, merupakan sebuah *node* yang melakukan perpindahan posisi dari sebuah jaringan ke satu jaringan yang lain tanpa mengubah alamat IP dan masih tetap dapat terhubung dengan *Correspondent Node*. *Mobile node* juga dapat berkomunikasi dengan *node* lain yang berada pada suatu lokasi tertentu dengan menggunakan alamat IP konstan.
5. *Correspondent Node (CN)*, sebuah *node* yang melakukan komunikasi dengan *Mobile Node*. *Correspondent Node* dapat berfungsi sebagai *Mobile Node*.
6. *Home Agent (HA)*, *router* pada *Home Link* yang dapat mengirimkan paket data untuk *Mobile Node* saat berpindah dari asalnya dan juga memelihara informasi lokasi dari *Mobile Node*.

7. *Care of Address (CoA)*, alamat yang mengidentifikasi lokasi *Mobile Node* saat ini.

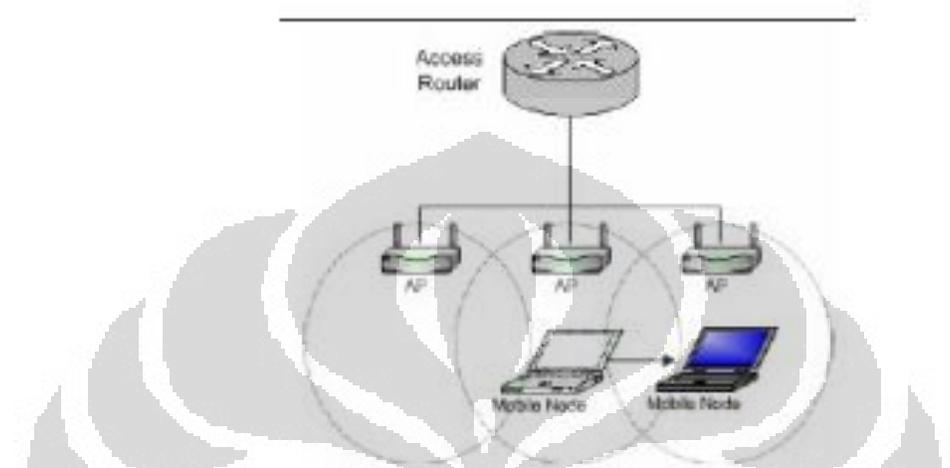
Beberapa layanan yang mendukung *Mobile IP* adalah:

- a. *Agent Solicitation*, permintaan atau permohonan kiriman layanan dari *Home Agent*, *Foreign Agent* dan *Access Point* oleh *Mobile Node* yang berisi permintaan *link* untuk mendeteksi adanya paket hilang tetapi masih memiliki *Care of Address* yang valid.
- b. *Registration*, ketika *Mobile Node* menjauhi *Home Agent*, *Mobile Node* akan melakukan *register* ke *Home Agent* dengan *Care of Address* yang dimiliki. Dengan demikian *Home Agent* dapat mengetahui lokasi terbaru *Mobile Node* dan mengirimkan paket data.
- c. Enkapsulasi, proses penumpangan IP *datagram* dengan *header* IP lain yang berisi *Care of Address* pada *Mobile Node*. IP *datagram* tetap utuh dan tidak tersentuh seluruhnya pada saat penumpangan.
- d. Dekapsulasi, proses pemisahan *header* IP terluar pada paket yang datang, sehingga *datagram* yang ditumpangkan dapat diakses dan dikirimkan ke tujuan yang sebenarnya.

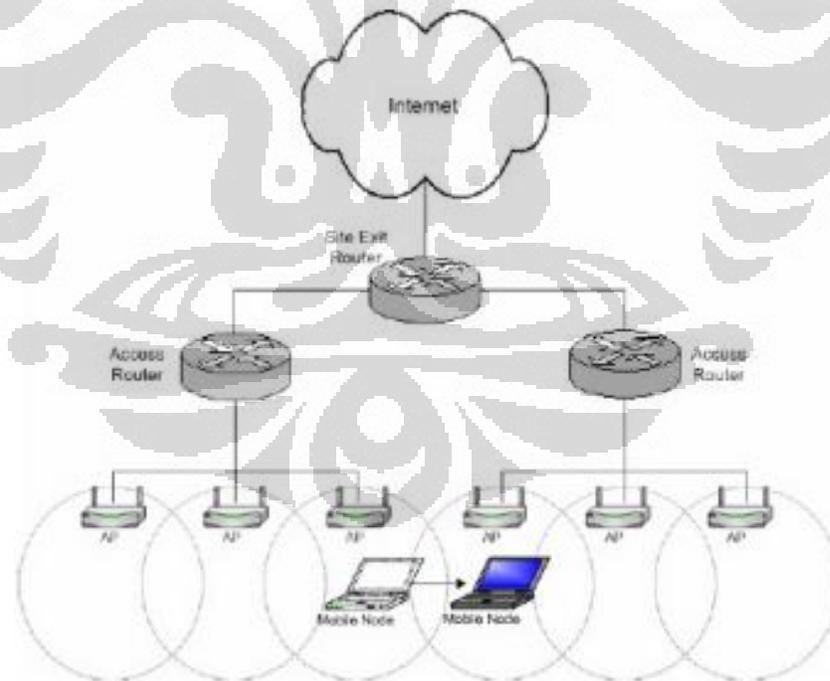
2.2.1. MIPv6 Handover

Handover adalah proses di mana MN berpindah dari satu jaringan *access point* (*home network*) ke jaringan *access point* yang lain (*foreign network*). *Handover* yang hanya mengalami perubahan pada *data link layer* tanpa mengubah alamat IP dinamakan *horizontal handover*. Sebagai contoh adalah ketika MN berpindah AP *wireless LAN* yang dilayani oleh IP *Access Router* yang sama. Dalam hal ini kedua AP berada pada *Extended Service Set (ESS)* yang sama. *Handover* yang terjadi ketika MN berpindah diantara AP yang berbeda ESS dan dilayani oleh AP yang berbeda dinamakan *Vertical Handover*. Proses ini dapat terjadi pada ISP yang sama maupun

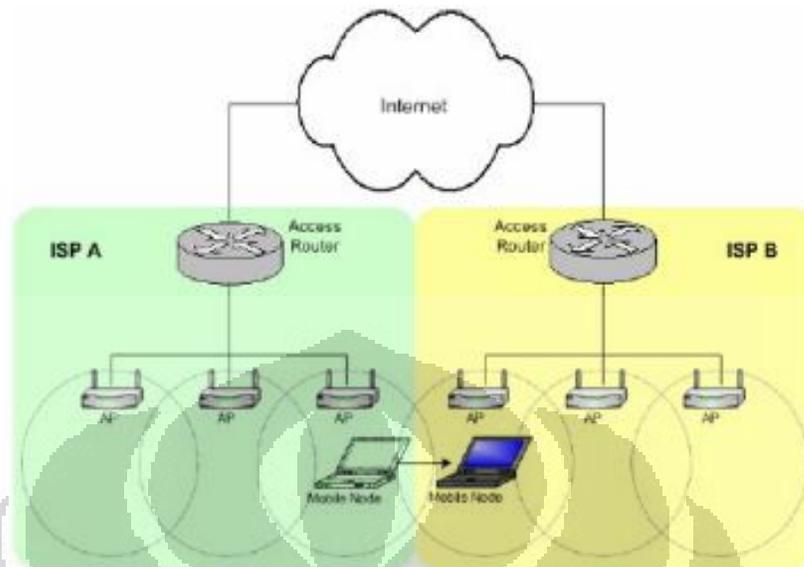
ISP yang berbeda. Gambar 2.5 menggambarkan tentang *horizontal handover*, gambar 2.6 menggambarkan tentang *vertical handover* pada ISP yang sama, dan gambar 2.7 menggambarkan tentang *vertical handover* pada ISP yang berbeda.



Gambar 2.5 *Horizontal Handover* [5]



Gambar 2.6 *Vertical Handover* pada ISP yang sama [5]



Gambar 2.7 Vertical Handover pada ISP yang berbeda [5]

2.2.2. Proses Handover

Karakteristik MIPv6 pada *wireless* LAN mengidentifikasi bahwa *delay* mempengaruhi *handover* selama proses *handover*. Proses *handover delay* meliputi:

- Waktu deteksi perpindahan (t_d)

Waktu yang diperlukan MN untuk mendeteksi dan memproses perpindahan ke titik akses baru.

- Waktu konfigurasi CoA (t_a)

Waktu antara proses perpindahan dan waktu untuk mendapatkan alamat IPv6 *global*.

- Waktu registrasi *binding* (t_r)

Waktu selama pengiriman *Binding Update* ke HA sampai menerima *binding acknowledgement*.

- Waktu *route optimization* (to)

Selang waktu dari registrasi CoA baru dengan HA dan CN sehingga metode *route optimization* dapat digunakan.

2.2.3. Prosedur Proses *Handover* MIPv6

Prosedur *handover* pada MIPv6 hampir sama dengan prosedur *autoconfiguration* pada saat IPv6 *booting up* ke sebuah jaringan, namun terdapat beberapa perbedaan:

- MN harus mengetahui bahwa dirinya telah berpindah ke jaringan baru.
- Setelah konfigurasi, MN harus menginformasikan HA dan CN mengenai lokasi terbarunya.
- Prosedur *handover* harus dilakukan secepat mungkin untuk meminimalisir terjadinya *packet loss* dan *packet delay*.

Prosedur *handover* pada MIPv6 sebagai berikut:

1. Deteksi perpindahan (*movement detection*)

MN memiliki kemampuan untuk mendeteksi apakah dirinya masih berada di *home network* atau sudah berpindah ke *foreign network*. Untuk mengetahui apakah CAR (*Current Access Router*) masih terjangkau secara dua arah, MN melakukan NUD (*Neighbour Unreachability Detection*) secara berkala. Apabila CAR tidak terjangkau, maka MN akan mengirimkan *router solicitation* untuk mencari *router* yang baru.

2. *Router Discovery*

Pada proses *router discovery* terjadi ketika MN menerima *router advertisement* dari *access router* yang baru (NAR). MN akan mengirimkan *router solicitation*

jika CAR tidak dapat dijangkau dan akan menerima *socialited advertisement* dari NAR yang dikirim secara periodik.

3. Konfigurasi CoA

MN mengkonfigurasi dirinya dengan alamat IPv6 pada jaringan baru, proses ini dinamakan CoA. Konfigurasi alamat IPv6 dapat dilakukan dengan dua cara, yaitu:

a) *Stateless Autoconfiguration Address*

MN secara otomatis mengkonfigurasi alamat IP dengan menggabungkan prefiks NAR dengan alamat MAC NIC.

b) *Stateful Configuration*

Menggunakan DHCPv6 pada mekanismenya. DHCPv6 dapat mengontrol dan mendokumentasikan penggunaan alamat IPv6.

4. *Duplicate Address Detection (DAD)*

MN yang telah berpindah ke jaringan baru akan melakukan proses DAD untuk CoA yang didapatkan dari hasil konfigurasi *stateless* atau *stateful*. Hal ini dilakukan agar tidak ada alamat IPv6 yang sama pada jaringan (duplikasi alamat IPv6). Jika terdapat *node* yang menggunakan alamat yang sama seperti CoA, akan terjadi dua hal yakni:

- *Node* duplikat tersebut akan mendapatkan pesan *neighbour solicitation* dan akan membalas dengan *neighbour advertisement* dan memberitahukan alamatnya pada MN.
- MN akan menerima *neighbour solicitation* dari *node* duplikat yang juga melakukan proses DAD

DAD akan memberikan indikasi pada MN bahwa ada *node* lain yang menggunakan alamat yang sama dengan CoA dan salah satu harus mengganti alamat IP-nya. Kemungkinan terjadinya duplikasi dengan menggunakan mekanisme *stateless* hampir tidak ada karena merupakan penggabungan antara prefiks jaringan dan alamat MAC yang memiliki sifat yang unik.

5. Otentikasi dan Otorisasi (AAA)

MN yang telah berpindah jaringan harus melakukan proses AAA supaya dapat mendapatkan akses ke jaringan yang baru. Pada proses AAA terjadi interaksi yang melibatkan proses *encompass* dan *handshake* diantara MN, *server* lokal dan *home server* MN.

6. Registrasi CoA

Setelah MN menerima CoA dan mendapatkasn hak akses ke jaringan baru, MN menginformasikan kepada HA tentang lokasinya yang baru. Selama koneksi MN terputus dengan PAR (*Previous Access Router*) sampai dengan MN menginformasikan lokasi barunya kepada HA, semua paket yang dikirimkan MN akan hilang dan tidak dapat mengirimkan paket ke CN manapun. HA mendaftarkan CoA-nya kepada HA dengan mengirimkan *Binding Update*. HA akan membalasnya dengan mengirimkan *binding acknowledgement*. Kini HA telah dapat men-*tunnel* paket yang ditujukan ke *home address* MN ke lokasi barunya (alamat CoA baru).

7. *Binding Update* (BU)

MN menginformasikan kepada semua CN tentang lokasi baru MN dan keterjangkauan MN melalui alamat CoA yang baru. Hal ini dilakukan dengan cara mengirimkan BU kepada semua CN.

2.2.4. Mekanisme MIPv6

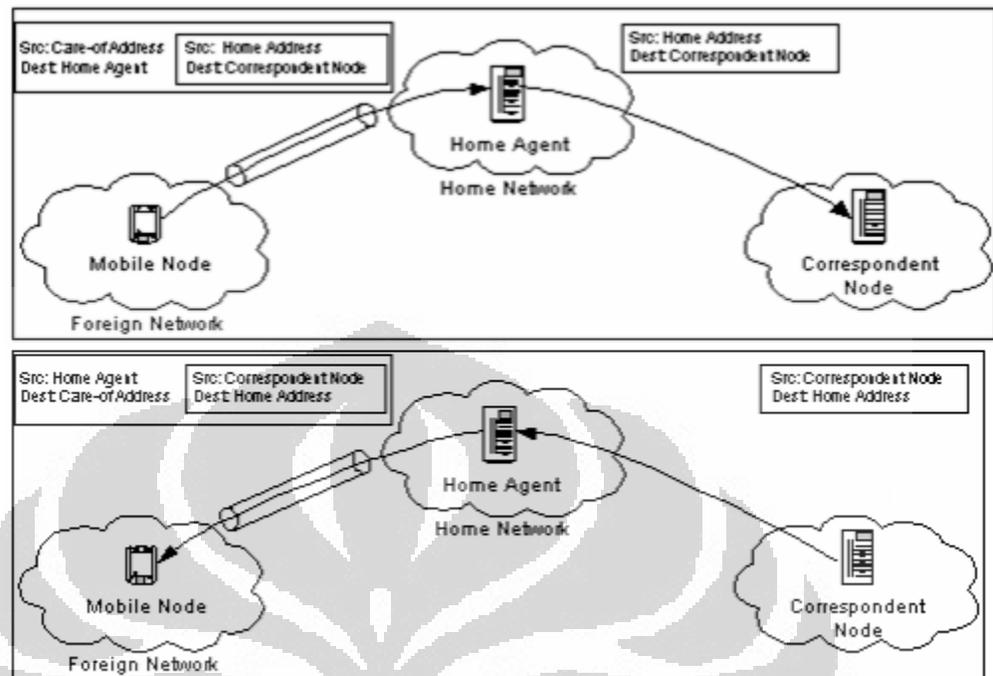
Mobile node pada mobile IP memiliki dua alamat, yaitu *home address* dan CoA. *Home address* adalah alamat IP yang diberikan *mobile node* sebagai identitas yang unik. Kombinasi dari *home address* dan CoA disebut juga *binding*.

Saat MN berpindah ke *foreign link*, MN akan menginformasikan *router* pada *home link* tentang status CoA-nya. HA akan merespon dan mengirimkan *binding acknowledgement* ke MN. Proses ini disebut juga sebagai *binding update*. MN menginformasikan lokasi baru ke CN melalui mekanisme *correspondent registration*.

Terdapat dua cara untuk melakukan komunikasi antara MN dengan CN, yaitu:

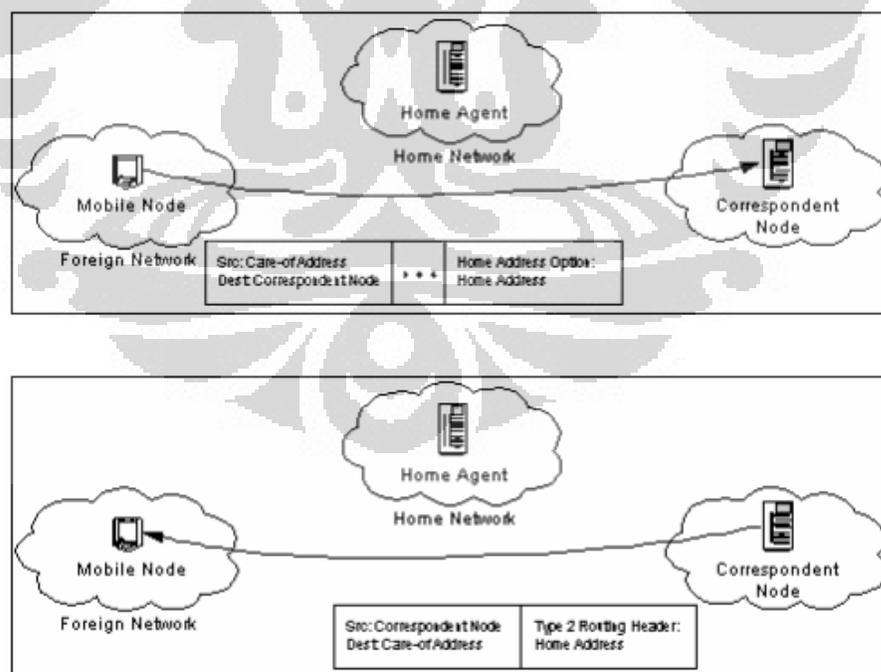
1. *Bidirectional Tunneling*

Bidirectional tunneling seperti pada Gambar 2.3 tidak memerlukan dukungan dari CN dan dapat tersedia walaupun MN tidak meregistrasi *binding* terbarunya dengan CN. Saat HA menerima paket yang dialamatkan pada *Home Address* dari MN dengan menggunakan mekanisme *Proxy Neighbour Discovery*, berarti HA mengirimkan pesan *multicast* berupa *Neighbour Advertisement* ke semua *host* pada *Home Network* atas nama MN. Pesan ini berisi alamat *link layer* dari HA kepada *Home Address* dari MN, kemudian HA akan membalas dengan mengirimkan pesan atas nama MN. Setiap paket yang diterima selanjutnya akan dikirimkan melalui sebuah *tunnel* ke alamat CoA dari MN dengan menggunakan mekanisme enkapsulasi paket Ipv6. Paket yang menuju CN dari MN juga akan melewati *tunnel* ke HA (*Reverse Tunneled*) sebelum dikirimkan secara normal ke CN.



Gambar 2.3 Bidirectional Tunneling [3]

2. Route Optimization



Gambar 2.4 Route Optimization [3]

Route optimization seperti pada Gambar 2.4 memerlukan dukungan MN untuk melakukan *registrasi binding*-nya pada CN. Paket-paket dari CN dapat dirutekan secara langsung ke CoA dari MN. Ketika mengirimkan sebuah paket ke beberapa tujuan, CN mengecek *binding* yang tertahan untuk masukan paket *destination address*. Jika *binding* yang tertahan untuk alamat tujuan ditemukan, *node* menggunakan sebuah tipe dari IPv6 *routing header* yang baru untuk melakukan *route* paket *mobile* ke MN dengan cara CoA menandai pada *binding* ini. *Routing* paket secara langsung ke *mobile node* CoA membolehkan penggunaan jalur komunikasi terpendek. Ini juga menghilangkan *congestion* HA dan *home link* pada MN. Kemungkinan terjadinya kegagalan dari HA atau *network* pada jalur dapat dikurangi. Ketika *routing* paket secara langsung ke MN, CN menyesuaikan *destination address* pada IPv6 *routing header* yang baru ditambahkan ke paket untuk dibawa ke *home address* yang ditentukan. Hampir mirip dengan MN menyesuaikan alamat asal dalam IPv6 paket *header* ke CoA yang baru. MN menambahkan pilihan tujuan IPv6 “*home address*” yang baru untuk membawanya ke *home address*. Pencantuman *home address* pada paket-paket ini membuat penggunaan CoA transparan di atas *network layer*.

2.3. *Hypertext Transfer Protocol (HTTP)*

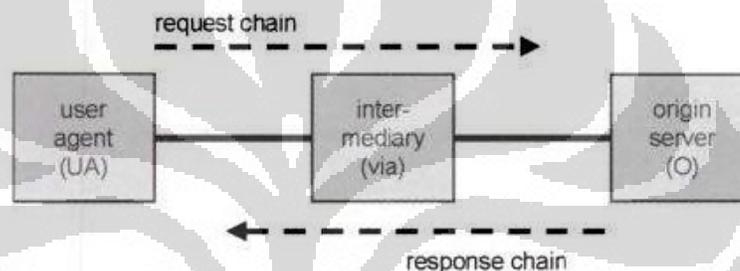
HTTP didefinisikan pada RFC 2068, adalah protokol yang memperbolehkan teks, grafik, multimedia, dan materi-materi lainnya untuk dapat diunduh dari HTTP *server*. HTTP menentukan *request* yang diminta oleh *client* dan bagaimana *server* membalas *request* tersebut. HTTP merupakan protokol *connectionless* yang menggunakan UDP sebagai protokol *transport*.

Pada implementasinya, HTTP *client* (*Web Browser*) membuat *request* dalam bentuk HTTP menggunakan koneksi TCP pada *port* 80 ke *server* yang menjalankan HTTP *server* (*Web Server*). *File* yang dibentuk menggunakan bahasa tertentu seperti *Hypertext Markup Language* (HTML) dikembalikan ke *client* dan koneksi akan tertutup.

HTTP menggunakan *Uniform Resource Locator* (URL) untuk menentukan halaman mana yang harus diunduh dari *remote server*. URL berisi tipe *request* (<http://>), nama *server* yang dituju (www.novell.com), dan halaman opsional yang di-*request* ([/support](http://www.novell.com/support)). Hasil halaman yang diminta akan menjadi:

<http://www.novell.com/support>

Proses *request* dan *response* pada HTTP dapat dilihat pada Gambar 2.8 sebagai berikut.



Gambar 2.8 Request dan response HTTP [4]

2.3.1. Hypertext Transfer Protocol Secure (HTTPS)

HTTPS merupakan salah satu cabang dari HTTP yang menggunakan sistem yang dikenal dengan *Secure Socket Layer* (SSL). SSL mengenkripsi informasi di antara *client* dan *host*. *Client* dan *server* harus mendukung HTTPS untuk dapat digunakan. *Browser-browser* saat ini sudah mendukung HTTPS, seperti *Microsoft Internet Information Server* (IIS), Apache dan aplikasi *web server* lainnya. Saat mengakses aplikasi HTTPS maka URL akan diawali dengan HTTPS daripada HTTP:

<https://www.mybankingonline.com>

BAB 3

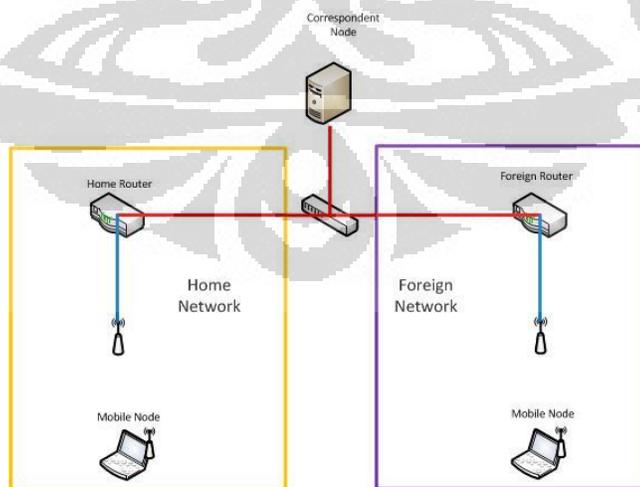
PERANCANGAN DAN IMPLEMENTASI JARINGAN VERTICAL HANDOVER MOBILE IPV6 UNTUK WEB SERVER (HTTP)

3.1 Jaringan Topologi

Rancangan jaringan yang akan diimplementasikan pada *Vertical Mobile IPv6* terdiri dari:

- 1 *Laptop*, berfungsi sebagai *Correspondent Node*,
- 1 *Laptop*, berfungsi sebagai *Mobile Node*,
- 2 *access point*,
- 1 *switch*.
- 2 *PC router*, berfungsi sebagai *Home Router/Home Agent* dan *Foreign Router*.

Sistem operasi yang akan digunakan adalah Linux Ubuntu versi 10.04. Pada *Vertical Handover* terdapat dua jaringan, di mana *Mobile Node* akan berada pada jaringan *Home Network* dan kemudian akan berpindah jaringan ke *Foreign Network*. Topologi jaringan yang akan diimplementasikan dapat dilihat pada Gambar 3.1 di bawah ini.



Gambar 3.1 Topologi Jaringan *Vertical Mobile IPv6*

3.2 Spesifikasi Sistem

3.2.1 Spesifikasi Perangkat Keras

Perangkat keras yang akan digunakan dalam perancangan jaringan *Vertical Handover Mobile IPv6* adalah :

1. *Home Router*

Home Router akan menggunakan sebuah *PC Router* yang mendukung *Mobile IPv6*, dan akan berfungsi sebagai *Home Agent* dan *Home Router*

Prosesor : Intel(R) Dual-Core CPU

Memori : 2 GB

Harddisk : 320 GB

2. *Foreign Router*

Foreign Router, akan menggunakan sebuah *PC Router* yang mendukung *Mobile IPv6*

Prosesor : Intel(R) Dual-Core CPU

Memori : 2 GB

Harddisk : 320 GB

3. *Mobile Node*

Mobile Node akan menggunakan sebuah *laptop* yang nantinya akan berpindah dari *access point* pada *Home Network* ke *access point* pada *Foreign Network*.

Prosesor : Intel(R) Core 2 Duo CPU

Memori : 2GB

Harddisk : 250GB

4. *Correspondent Node*

Correspondent node akan menggunakan sebuah *laptop* yang mendukung *Mobile IPv6* yang berhubungan dengan *mobile node*.

Prosesor : Intel(R) Core i7 CPU

Memori : 4 GB

Harddisk : 500 GB

5. *Access Point*

Access point akan digunakan sebagai penghubung *router* dengan *mobile node* dan *correspondent node*.

Tipe : TP-LINK Wireless-G Access Point

DataRates : 54 Mbps

6. *Switch*

Switch akan digunakan untuk menghubungkan antara *Correspondent Node*, *Home Router*, dan *Foreign Node*.

Tipe : TP-LINK TL-SF1005D

Ports : -ports 20/200 Mbps

3.2.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang akan digunakan dalam perancangan jaringan *Vertical Handover Mobile IPv6* adalah

1. Sistem operasi Linux Ubuntu 10.04 LTS (Lucid Linux)

Dalam perancangan jaringan *Vertical Handover Mobile IPv6* ini akan menggunakan sistem operasi Linux Ubuntu 10.04. Linux Ubuntu 10.04 belum mendukung fitur *mobile IPv6*, oleh karena itu perlu dilakukan instalasi kernel

baru sehingga Linux Ubuntu 10.04 mendukung fitur *mobile* IPv6. Kernel yang harus diinstalasi adalah kernel Linux-2.6.32.32.

2. UMIP 4.0

UMIP atau *mip6d* merupakan Linux Mobile IPv6 Daemon, yaitu perangkat lunak yang mampu menciptakan lingkungan *mobile* IPv6 pada jaringan *mobile* IPv6. UMIP membuat konfigurasi pada *Correspondent Node*, *Home Agent*, dan *Mobile Node* pada jaringan *mobile* IPv6.

3. Router Advertisement Daemon

Router Advertisement Daemon atau *radvd* adalah perangkat lunak yang digunakan untuk mengimplementasi *link-local advertisement* pada alamat *router* IPv6 dan *routing prefix* IPv6. Saat *mobile node* melakukan konfigurasi perangkat jaringannya, *mobile node* akan mengirimkan *router solicitation* (RS) ke *Home Router* dan *Foreign Router* untuk meminta alamat IPv6. *Radvd* menerima permintaan dari *mobile node* dan kemudian memberikan alamat IPv6 dengan *router advertisement* (RA). *Radvd* yang digunakan pada jaringan *mobile* IPv6 adalah *radvd* versi 1.3.

4. Wireshark

Wireshark adalah sebuah perangkat lunak *freeware* untuk memantau dan menganalisa protokol jaringan.

5. Traffic Generator

Traffic generator adalah sebuah perangkat lunak yang berfungsi untuk memberikan *traffic* pada jaringan sehingga lebih menyerupai jaringan internet yang sebenarnya. Perangkat lunak ini akan diinstal pada sebuah *node* dalam jaringan, yang nantinya akan terus mengirimkan *traffic*. Nama perangkat lunak yang digunakan adalah *Multi-Generator* (MGEN).

6. Mozilla Firefox

Web browser yang akan digunakan untuk melakukan proses komunikasi HTTP yang memanggil halaman *web* dan mengunduh *file* yang tersimpan didalam *server*.

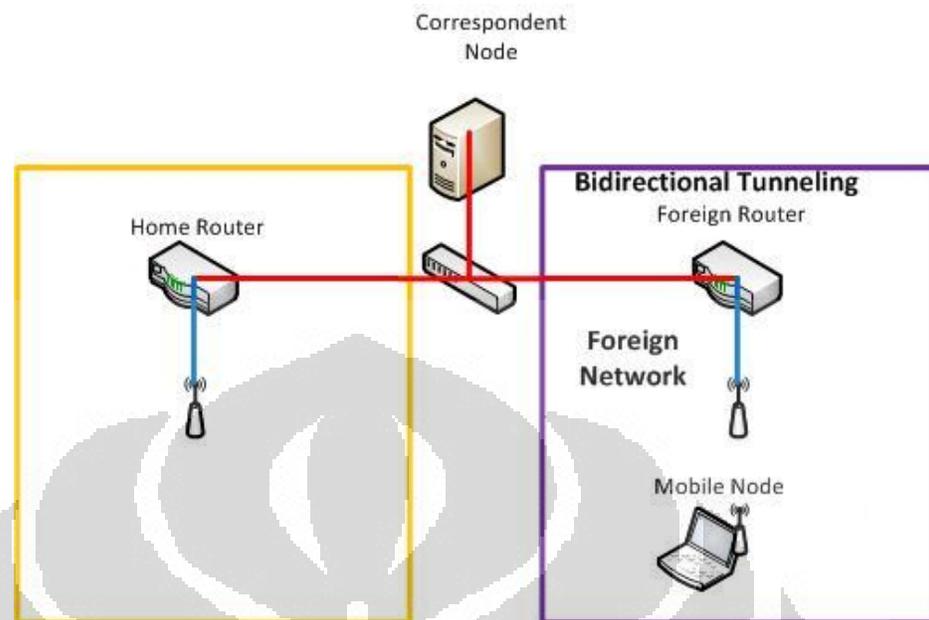
7. LAMP Server

LAMP (Linux, Apache, MySQL, PHP)-*Server* adalah perangkat lunak yang merupakan gabungan beberapa komponen untuk dapat membuat sebuah *web server*. Apache HTTP *Server* pada LAMP adalah sebuah perangkat lunak *open source* yang dapat menyediakan layanan HTTP dan menyediakan *server* yang efisien dan aman. MySQL adalah sistem basis data yang digunakan di dalam *web*. MySQL menyediakan pembuatan struktur basis data pada *web server* yang berfungsi untuk menyimpan atau mengotomatisasi prosedur. Komponen terakhir pada LAMP adalah PHP yang merupakan bahasa pemrograman berbasis *web* yang memiliki fitur proses data secara dinamis. PHP terdiri dari sintaks-sintaks dan perintah-perintah yang akan diproses oleh *server* sehingga dapat disebut *server-side embedded script*.

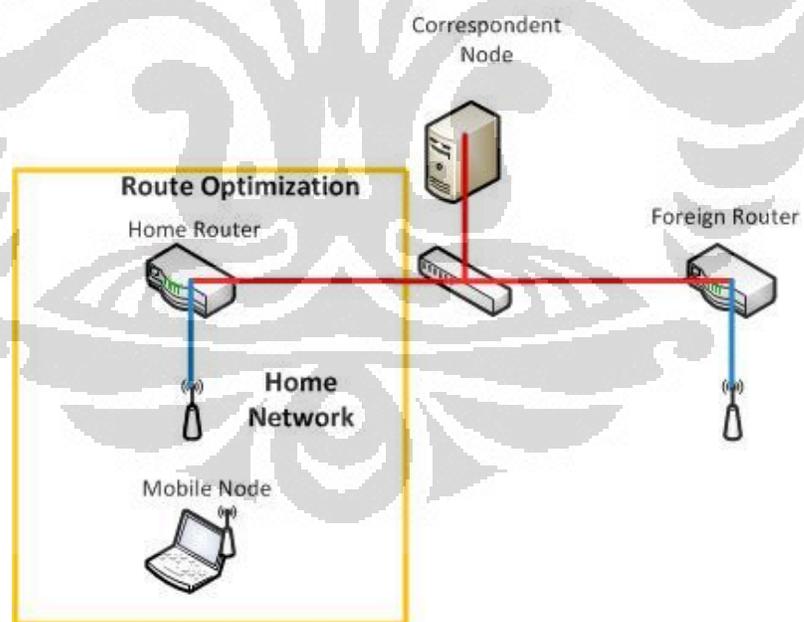
3.3 Skenario Pengukuran Pada Jaringan Vertical Mobile IPv6

Skenario yang direncanakan dalam sistem ini ada sebanyak empat skenario.

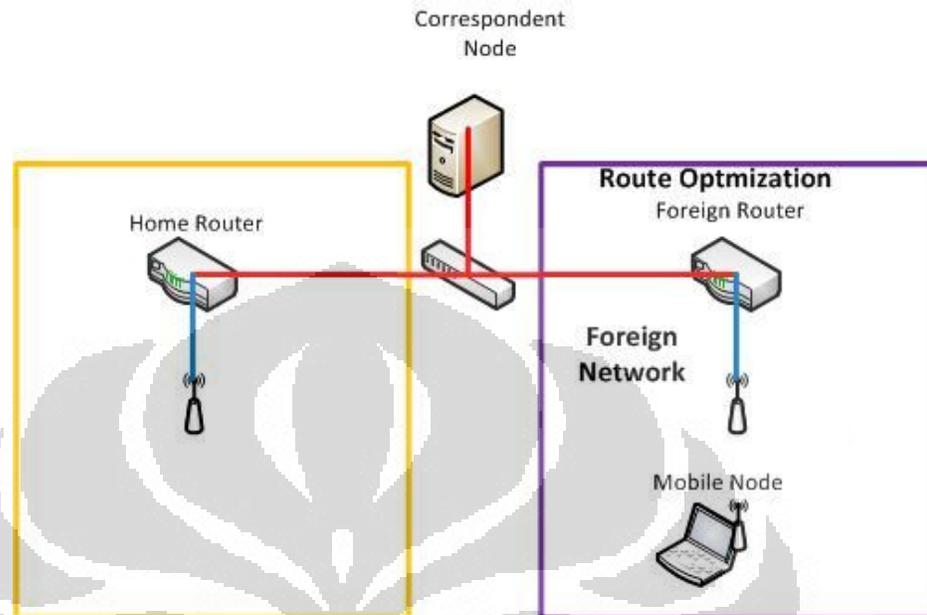
1. Skenario pertama akan menggunakan *bidirectional tunneling*, di mana *mobile node* berada dalam *home network* kemudian *mobile node* tersebut mengakses *web server* yang tersimpan di dalam *coresspondent node*.
2. Skenario kedua tetap akan menggunakan *bidirectional tunneling*, di mana *mobile node* berada dalam *foreign network* kemudian *mobile node* akan mengakses *web server* yang berada di dalam *coresspondent node*.



Gambar 3.3 Skenario 2



Gambar 3.3 Skenario 3



Gambar 3.4 Skenario 4

Pengambilan data akan dilakukan sebanyak sepuluh kali pada setiap skenario untuk mendapatkan data yang akurat. Halaman *website* akan bervariasi dari *website* yang berukuran 20 KB, 40 KB, dan 80 KB, untuk membantu proses analisa jaringan *mobile IPv6* tersebut. Ukuran HTML tersebut akan ditambahkan dengan ukuran *file* gambar yang terdapat pada halaman *web*, sehingga variasi ukuran HTML yang akan diakses menjadi 20 KB + 925 KB, 40 KB + 2899 KB, dan 80 KB + 6542 KB. Setelah mendapatkan hasil data yang diambil pada setiap skenario kemudian akan dibandingkan *throughput*, *packet loss*, dan *delay* pada *home link* maupun *foreign link* menggunakan metode *bidirectional tunneling* dan *route optimization*.

Adapun parameter QoS yang akan diukur, antara lain:

1. *Throughput*, yaitu kecepatan pengiriman data (*transfer rate*) yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval tertentu dibagi oleh durasi interval waktu tersebut.

2. *Packet Loss*, merupakan sebuah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah paket yang hilang, dapat disebabkan oleh *colission* (tabrakan) dan *congestion* (penumpukan) pada jaringan.
3. *Delay*, adalah waktu yang dibutuhkan data untuk menempuh jarak dari sumber ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama.

3.4 Pembuatan sistem

Pembuatan sistem jaringan *vertical mobile IPv6* dilakukan melalui beberapa tahapan instalasi dan konfigurasi sehingga setiap *node* pada jaringan dapat bekerja sesuai dengan fungsinya masing-masing, yakni sebagai *Home Router*, *Foreign Router*, *Coresspondent Node*, maupun *Mobile Node*. Hal pertama yang harus dilakukan dalam pembuatan sistem jaringan *vertical mobile IPv6* adalah instalasi kernel linux 2.6.32.32 yang mendukung fitur *IPv6 mobility*. Setelah melakukan instalasi kernel-linux 2.6.32.32 kemudian dilanjutkan dengan konfigurasi UMIP dan radvd pada setiap node. Proses pembuatan sistem jaringan *mobile IPv6* berikutnya adalah penginstalan perangkat lunak LAMP yang berfungsi sebagai *web server* pada jaringan ini.

Node-node pada jaringan *mobile IPv6* akan dikonfigurasi sesuai dengan fungsi-fungsi yang telah ditentukan. Perangkat lunak MGEN yang berfungsi untuk menciptakan *traffic* akan dikonfigurasi pada *Home Router*, *Foreign Router*, dan *Correspondent Node*. Konfigurasi *bidirectional tunneling* dan *route optimization* dilakukan pada *Home Router* dan *Correspondent Node*.

3.4.1 Instalasi Kernel Linux-2.6.32.32

Kernel bawaan Linux 10.04 LTS belum mendukung fitur *IPv6 mobility*, sehingga diperlukan instalasi kernel baru yang memungkinkan sistem operasi linux dapat mengaktifkan fitur *mobile IPv6*. Instalasi kernel linux-2.6.32.32 dapat diunduh dari <http://kambing.ui.ac.id/linux/v2.6/longterm/v2.6.32/linux-2.6.32.32.tar.bz2>. File

kernel linux-2.6.32.32 yang telah diunduh, kemudian disimpan di dalam direktori /usr/src.

Setelah kernel linux-2.6.32.32 selesai terunduh dalam format tar.bz2, maka kernel linux-2.6.32.32 harus diekstrak terlebih dahulu karena tar.bz2 merupakan format arsip terkompresi.

Fitur IPv6 *mobility* pada kernel linux-2.6.32.32 dapat diaktifkan dengan mengubah beberapa konfigurasi *file* kernel. *File* konfigurasi kernel yang akan diubah menggunakan *file* konfigurasi kernel yang lama dan mengatur opsi *mobility* yang hanya diperlukan pada jaringan *mobile* IPv6 yang akan diimplementasikan.

Sebelum melakukan *upgrade* kernel baru, perlu dipastikan beberapa perangkat lunak sudah terinstal di dalam sistem operasi linux ubuntu 10.04 LTS untuk mendukung proses instalasi kernel baru. Perangkat lunak tersebut adalah sebagai berikut.

<i>Gnu C</i>	2.95.3	# <i>gcc --version</i>
<i>Gnu make</i>	3.78	# <i>make --version</i>
<i>binutils</i>	2.12	# <i>ld -v</i>
<i>util-linux</i>	2.10o	# <i>fdformat --version</i>
<i>module-init-tools</i>	0.9.10	# <i>depmod -V</i>
<i>e2fsprogs</i>	1.29	# <i>tune2fs</i>
<i>jfsutils</i>	1.1.3	# <i>fsck.jfs -V</i>
<i>reiserfsprogs</i>	3.6.3	# <i>reiserfsck -V 2>&1 grep reiserfsprogs</i>
<i>xfspgrog</i>	2.1.0	# <i>xfp_db -V</i>
<i>pcmcia-cs</i>	3.1.21	# <i>cardmgr -V</i>
<i>quota-tools</i>	3.09	# <i>quota -V</i>
<i>PPP</i>	2.4.0	# <i>pppd --version</i>
<i>isdn4k-utils</i>	3.1pre1	# <i>isdnctrl 2>&1 grep version</i>
<i>nfs-utils</i>	1.0.5	# <i>showmount --version</i>
<i>procps</i>	3.1.13	# <i>ps --version</i>
<i>oprofile</i>	0.5.3	# <i>oprofiled -version</i>

Setelah memastikan bahwa perangkat lunak tersebut terinstal pada kernel linux, selanjutnya memilih fitur-fitur pendukung fitur *mobile* IPv6. Proses pengaktifan fitur-fitur pendukung *mobile* IPv6 adalah sebagai berikut.

General setup

--> *Prompt for development and/or incomplete code/drivers*

[CONFIG_EXPERIMENTAL]

--> *System V IPC* **[CONFIG_SYSVIPC]**

Networking support **[CONFIG_NET]**

--> *Networking options*

--> *Transformation user configuration interface* **[CONFIG_XFRM_USER]**

--> *Transformation sub policy support* **[CONFIG_XFRM_SUB_POLICY]**

--> *Transformation migrate database* **[CONFIG_XFRM_MIGRATE]**

--> *PF_KEY sockets* **[CONFIG_NET_KEY]**

--> *PF_KEY MIGRATE* **[CONFIG_NET_KEY_MIGRATE]**

--> *TCP/IP networking* **[CONFIG_INET]**

--> *The IPv6 protocol* **[CONFIG_IPV6]**

--> *IPv6: AH transformation* **[CONFIG_INET6_AH]**

--> *IPv6: ESP transformation* **[CONFIG_INET6_ESP]**

--> *IPv6: IPComp transformation* **[CONFIG_INET6_IPCOMP]**

--> *IPv6: Mobility* **[CONFIG_IPV6_MIP6]**

--> *IPv6: IPsec transport mode*

[CONFIG_INET6_XFRM_MODE_TRANSPORT]

--> *IPv6: IPsec tunnel mode* **[CONFIG_INET6_XFRM_MODE_TUNNEL]**

--> *IPv6: MIPv6 route optimization mode*

[CONFIG_INET6_XFRM_MODE_ROUTEOPTIMIZATION]

--> *IPv6: IPv6-in-IPv6 tunnel* **[CONFIG_IPV6_TUNNEL]**

--> *IPv6: Multiple Routing Tables* **[CONFIG_IPV6_MULTIPLE_TABLES]**

--> *IPv6: source address based routing* **[CONFIG_IPV6_SUBTREES]**

File systems

--> Pseudo filesystems

--> /proc file system support [CONFIG_PROC_FS]

Setelah selesai memilih fitur yang akan diaktifkan pada kernel baru, maka akan dilakukan penyusunan konfigurasi (*upgrade*) dan instalasi kernel linux yang baru. Untuk memastikan bahwa kernel linux-2.6.32.32 sudah berhasil terinstal, dengan memeriksa kembali konfigurasi *bootloader* pada grub dengan me-*restart* PC.

3.4.2 Instalasi UMIP 4.0 dan Radvd

Proses instalasi UMIP dilakukan dengan menyusun kembali beberapa *source code* pada paket-paket pendukung yang terdapat di dalam UMIP, sehingga UMIP dapat menciptakan *environment mobile IPv6*. Paket-paket yang akan disusun kembali pada UMIP, antara lain *autoconf*, *bison*, *libssl-de*, *indent*, *ipsec-tools*, dan *radvd*. Sebelum melakukan proses instalasi paket-paket UMIP, harus dipastikan terlebih dahulu bahwa paket-paket tersebut sudah ada pada UMIP. Selanjutnya *source code* UMIP akan dipindahkan dari repository ke direktori */usr/src*, sehingga UMIP dapat diaktifkan dan disusun kembali *source code*-nya.

Source code pada paket-paket UMIP yang telah terinstal kemudian akan dipindahkan ke direktori */usr/src* dengan mengunduhnya pada repository. Setelah memindahkan UMIP yang terdapat pada repository ke direktori */usr/src*, langkah terakhir adalah menyusun kembali *source code* UMIP dan menginstalnya.

3.4.3 Instalasi LAMP

Proses instalasi LAMP pada Ubuntu 2.6.32.32, dengan memasukkan perintah sebagai berikut:

sudo apt-get install lamp-server^

Saat memasukkan perintah instalasi pada terminal tanda “^” harus ikut terketik, karena jika tidak ikut terketik maka proses instalasi tidak akan berjalan. Saat proses instalasi sedang berlangsung, akan terdapat perintah untuk memasukkan

password baru untuk *root user* pada MySQL database. APT Package Manager kemudian akan melanjutkan proses instalasi LAMP hingga selesai.

Untuk memastikan bahwa LAMP sudah terinstal, maka dapat dilakukan *test* dengan cara mengakses *localhost* melalui *web browser*. Masukkan alamat <http://localhost/> pada URL di *web browser*, jika muncul tulisan “*It works!*” pada halaman *web* maka LAMP telah berhasil terinstal.

3.4.4 Konfigurasi Node

Proses konfigurasi *node-node* pada jaringan *vertical handover mobile IPv6* terdiri dari beberapa tahap, diantaranya: konfigurasi alamat *node*, konfigurasi *static routing*, fungsi-fungsi *node*, dan konfigurasi UMIP.

1. Home Router

Home Router atau *Home Agent* memiliki dua *interface* yakni *interface* yang digunakan untuk *home link* dengan alamat IPv6 2001:db8:ffff:100a::2/64 pada eth1 dan *interface* yang digunakan ke *Foreign Router* dan *Correspondent Node* dengan alamat IPv6 2001:db8:ffff:100b::1/64 pada eth0. Konfigurasi *node Mobile IPv6* disimpan dalam *file* *mip6d.conf* pada direktori */usr/local/etc*. Sebelum mengubah konfigurasi *file* *mip6d.conf*, setiap *interface* dikonfigurasi terlebih dahulu dan mengaktifkan fungsi pada *Home Router*. *Home Router* kemudian diberikan *static routing* untuk membuat *home link* dapat mengakses *foreign link*. Setelah memberikan alamat IPv6 pada setiap *interface* pada *Home Router* dan menambahkan *static routing*, kemudian mengaktifkan beberapa fungsi *Home Router*. Perangkat lunak UMIP yang sebelumnya telah terinstal pada kernel 2.6.32.32 akan ditambahkan beberapa konfigurasinya yang terdapat didalam *file* *mip6d.conf* pada direktori */usr/local/etc*, untuk mendukung fitur *mobile IPv6*. Radvd yang berfungsi untuk memberikan *router advertisement* pada *mobile node*-pun akan disusun ulang beberapa konfigurasinya, sehingga Radvd dapat diaktifkan pada *Home Router*.

2. *Foreign Router*

Foreign Router memiliki dua *interface* yakni *interface* yang digunakan untuk *foreign link* dengan alamat IPv6 2001:db8:ffff:100c::2/64 pada eth1 dan *interface* yang digunakan ke *Home Router* dan *Correspondent Node* dengan alamat IPv6 2001:db8:ffff:100b::2/64 pada eth0. *Foreign Router* tidak memerlukan konfigurasi *file* mip6d.conf, karena *Foreign Router* tidak menjaga koneksi dengan *mobile node*. Setiap *interface* pada *Foreign Router* dikonfigurasi dan fungsi pada *Foreign Router* diaktifkan. Sama halnya seperti *Home Router*, pada *Foreign Router* harus ditambahkan *static routing* supaya *foreign link* dapat mengakses *home link*. Setelah itu, mengaktifkan beberapa fungsi *Foreign Router* dengan perintah yang sama seperti pada *Home Router*. Untuk menjalankan radvd pada *Foreign Router*, diperlukan *file* konfigurasi radvd.conf yang terletak pada direktori /etc.

3. *Correspondent Node*

Tidak seperti *Home Router* maupun *Foreign Router*, *Correspondent Node* hanya memiliki satu *interface* yaitu eth0 dengan alamat IPv6 2001:db8:ffff:100b::3/64 yang tersambung dengan *Home Router* dan *Foreign Router* melalui *switch*. Agar *Correspondent Node* dapat melakukan akses pada *home link* dan *foreign link*, maka perlu ditambahkan *static routing*. Pada *Correspondent Node* diperlukan *file* konfigurasi UMIP, yaitu mip6d.conf.

4. *Mobile Node*

Mobile Node tidak memerlukan pengalamatan IPv6 secara manual, tidak seperti pada *node-node* lainnya. Pengalamatan IPv6 pada *Mobile Node* menggunakan *file* konfigurasi mip6d.conf dan konfigurasi UMIP. *Mobile Node* perlu diaktifkan beberapa fungsi sehingga dapat mengakses *node-node* lainnya pada jaringan *mobile* IPv6.

3.4.5 **Konfigurasi *Multi Generator***

Multi Generator atau MGEN merupakan sebuah perangkat lunak yang dapat memberikan kemampuan untuk melakukan tes performansi jaringan IP dan merekayasa suatu trafik di dalam jaringan. Rekayasa trafik dilakukan dengan cara

menciptakan suatu pola trafik *real-time* sehingga jaringan dapat diberi beban yang bervariasi. Trafik yang diciptakan dapat dicatat dan disimpan untuk proses analisis selanjutnya. Data log MGEN dapat digunakan untuk menghitung statistik performansi pada *throughput*, *packet loss rates*, *delay*, dan lainnya.

Untuk dapat menjalankan MGEN sebagai pembuat suatu trafik pada jaringan, diperlukan *server* dan *client* yang berfungsi untuk mengirimkan dan mendengarkan paket. *Server* akan berfungsi untuk mengirimkan paket-paket yang akan diciptakan, sedangkan *client* berfungsi untuk mendengarkan permintaan koneksi dan merespon permintaan koneksi tersebut serta menerima paket-paket tersebut.

MGEN memiliki suatu format *command line*, sebagai berikut:

```
# mgen [ipv6][input <scriptFile>][output <logFile>]
```

[ipv6] pada *command line* menunjukkan bahwa *internet protocol* yang digunakan adalah IPv6. [input <scriptFile>] menunjukkan *script file* yang akan dijalankan oleh MGEN. <scriptFile> pada [input <scriptFile>] diubah dengan direktori *script file* tersebut disimpan. [output <logFile>] menunjukkan bahwa MGEN harus mencatat perintah-perintah yang dijalankan dalam sebuah *log file*. <logFile> pada [output <logFile>] diubah dengan direktori *log file* tersebut disimpan serta nama *log file*.

MGEN diaktifkan pada setiap *node* pada jaringan *mobile* IPv6, yaitu *Home Router/Home Agent*, *Foreign Router*, dan *Correspondent Node*. Isi dari setiap *script file* pada *node-node* tersebut adalah sebagai berikut:

a. *Home Router*

```
0.0 LISTEN UDP 5000
0.0 ON 1 UDP SRC 5001 DST 2001:db8:ffff:100b::2/5001
PERIODIC [32 8129]
10.0 ON 2 TCP DST 2001:db8:ffff:100b::3/8000 PERIODIC [1
5242880]
```

b. *Foreign Router*

0.0 LISTEN UDP 5001

0.0 ON 1 UDP SRC 5002 DST 2001:db8:ffff:100b::3/5002
PERIODIC [32 8192]

0.0 ON 2 UDP SRC 5000 DST 2001:db8:ffff:100b::1/5000
PERIODIC [32 8192]

c. *Correspondent Node*

0.0 LISTEN TCP 8000

0.0 LISTEN UDP 5002

Home Router/Home Agent dirancang untuk mengirimkan pesan TCP secara periodik sebanyak 1 pesan per detik dengan ukuran 5242880 byte. *Foreign Router* dirancang untuk mengirimkan paket UDP secara periodik sebanyak 32 pesan per detik dengan ukuran 8192 byte. *Correspondent Node* dirancang untuk mendengarkan kiriman pesan UDP di *port* 5002 dari *Foreign Router* dan mendengarkan permintaan sambungan dan kiriman pesan TCP di *port* 8000.

Home router dan *Foreign Router* juga saling bertukar pesan UDP, di mana *Home Router* mengirimkan pesan UDP sebanyak 32 pesan per detik dengan ukuran 8192 byte dari *port* 5001 pada *Home Router* ke *port* 5001 pada *Foreign Router* dan *Foreign Router* juga mengirimkan pesan UDP sebanyak 32 pesan per detik dengan ukuran 8192 byte dari *port* 5000 pada *Foreign Router* ke *port* 5000 pada *Home Router*.

BAB 4

PENGUKURAN DAN ANALISA PARAMETER QOS JARINGAN

MOBILE IPV6

4.1 Pengukuran Jaringan *Mobile IPv6*

Jaringan yang digunakan dalam analisa performansi *mobile IPv6* menggunakan metode *Bidirectional Tunneling* dan *Route Optimization*, di mana proses *handover* yang terjadi menggunakan *vertical handover*. Jaringan terdiri dari *Home Agent, Foreign Router, Correspondent Node,* dan *Mobile Node*. Pengujian akan dilakukan dengan mengakses halaman *web* yang tersimpan pada *correspondent node*.

LAMP merupakan salah satu perangkat lunak *open source* dari HTTP yang digunakan untuk mengakses halaman *web* yang terdapat di dalam *correspondent node*. LAMP terdiri dari Linux sebagai sistem operasi, Apache HTTP server sebagai *web server*, MySQL sebagai sistem basis data, dan PHP sebagai bahasa pemrograman yang digunakan. HTTP merupakan suatu jenis aplikasi yang bekerja dengan menggunakan protokol TCP/IP yang terhubung dengan *port 80*.

Proses pengambilan data dilakukan dengan mengakses halaman web ke *server* menggunakan *web browser*. Pengambilan data terjadi pada saat mengakses halaman *web* sampai halaman *web* selesai terakses oleh *web browser*. Data akan diambil sebanyak sepuluh kali untuk mendapatkan variasi data yang kemudian akan dianalisa. Halaman *web* yang telah berhasil terakses dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Tampilan web

Bahasa pemrograman yang digunakan pada halaman *web* adalah HTML. HTML atau *HyperText Markup Language* merupakan bahasa markup yang digunakan untuk membuat sebuah halaman *web* dan disimpan dalam format ASCII. Terdapat tiga jenis ukuran *file* HTML yang akan digunakan pada setiap skenarionya, yaitu 20 KB + 925, 40 KB + 2899, dan 80 KB + 6542 KB. Parameter-parameter yang akan diukur dalam analisa uji jaringan *mobile IPv6*, antara lain *throughput*, *packet loss*, dan *delay*. Analisa jaringan *mobile IPv6* menggunakan parameter-parameter tersebut dilakukan sesuai dengan skenario yang dibuat, yakni analisa jaringan ketika *mobile node* berada pada *home network* menggunakan *bidirectional tunneling* (skenario 1), ketika *mobile node* berada pada *foreign network* menggunakan *bidirectional tunneling* (skenario 2), ketika *mobile node* berada pada *home network* menggunakan *route optimization* (skenario 3), dan ketika *mobile node* berada pada *foreign network* menggunakan *route optimization* (skenario 4).

4.2 Analisa Home Network

Analisa pada *home network* merupakan analisa yang dilakukan pada saat *mobile node* berada pada *home network*. Analisa ini dilakukan mengikuti skenario 1 dan skenario 3, dimana akan terdapat perbedaan metode dalam komunikasi antara

mobile node dengan *correspondent node*. Metode tersebut adalah *bidirectional tunneling* yang digunakan pada skenario 1 dan *route optimization* yang digunakan pada skenario 3.

4.2.1 Analisa Throughput

Throughput merupakan kecepatan pengiriman data yang diukur dalam bps (*bit per second*), yang berarti jumlah kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu. Pengambilan data untuk analisa *throughput* skenario 1 dilakukan pada saat *mobile node* berada pada *home network* dan mengakses halaman *web* pada *server* menggunakan metode *bidirectional tunneling*. Ketika *mobile node* mengakses halaman *web*, pada saat yang bersamaan pada sisi *mobile node* melakukan penangkapan paket-paket yang masuk melalui *interface* WLAN. Penangkapan paket-paket menggunakan aplikasi *Wireshark*. Melalui hasil pemantauan (*capture*) paket-paket yang didapatkan *wireshark* dapat diketahui beberapa informasi mengenai protokol jaringan. Perlu dilakukan *filtering* protokol HTTP pada hasil *capture* jaringan untuk dapat melihat *summary* pada *wireshark*. Dapat dilihat berbagai parameter *traffic* jaringan pada bagian *summary wireshark*, yang kemudian dapat dilakukan analisa *throughput* pada skenario 1. Hasil *throughput* dapat dilihat pada Gambar 4.2 di bawah ini.

Display			
Display filter:	http		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	550	101	0
Between first and last packet	19.401 sec	0.968 sec	
Avg. packets/sec	28.349	104.288	
Avg. packet size	873.005 bytes	1185.475 bytes	
Bytes	480153	119733	
Avg. bytes/sec	24749.054	123631.350	
Avg. MBit/sec	0.198	0.989	

Gambar 4.2 *Throughput* pada *summary wireshark* (skenario 1)

Data pada Gambar 4.2 merupakan *throughput* hasil pencuplikan data pada *file* HTML berukuran 20 KB + 925 KB saat halaman *web* diakses menggunakan metode *bidirectional tunneling*. Terlihat pada hasil pencuplikan diatas bahwa satuan *throughput* yang diambil menggunakan satuan *Megabit per second* (Mbit/sec). Pada *summary wireshark* tersebut dapat dilihat bahwa rata-rata *throughput* untuk salah satu data pada skenario 1 sebesar 0.989 Mbit/sec. Nilai *throughput* pada setiap data yang diambil sebanyak sepuluh kali memiliki nilai *throughput* yang berbeda-beda. Setelah mencuplik data HTML 20 KB + 925 KB sebanyak sepuluh kali, maka akan diperoleh sepuluh data yang kemudian dari data-data tersebut dapat dibuat grafik perbandingan nilai *throughput*. Perbandingan nilai *throughput* pada *home network* untuk tiap ukuran *file* (skenario 1) dapat dilihat pada Lampiran B.

Setiap data yang telah diambil sebanyak sepuluh kali pada *file* HTML menggunakan metode *bidirectional tunneling*, akan dicari nilai rata-ratanya untuk kemudian dapat dibandingkan dengan ukuran *file* HTML lainnya. Rata-rata *throughput* yang diperoleh dari perhitungan keseluruhan data sesuai dengan ukuran *file* HTML yang telah ditentukan dapat dilihat pada Tabel 4.1.

Tabel 4.1 Data nilai rata-rata *throughput* pada skenario 1

File Size (Kbytes)	Throughput (bps)
20 + 925	0,9818
40 + 2899	0,8074
80 + 6542	0,4814

Terkait Tabel 4.1, terlihat bahwa nilai rata-rata *throughput* dari setiap ukuran yang berbeda ternyata memiliki hasil berbeda. Dilihat dari ukuran *file* HTML yang digunakan saat mengakses halaman *web*, semakin besar ukuran file maka semakin kecil nilai *throughput* yang diperoleh. Berdasarkan perhitungan menurut persentase, *file* HTML dengan ukuran 20 KB + 925 KB lebih besar 17.76% dari *file* HTML ukuran 40 KB + 2899 KB, sedangkan *file* dengan ukuran 40KB lebih besar 67.72%

dibandingkan dengan *file* HTML ukuran 80KB + 6542 KB. Terdapatnya perbedaan pada nilai *throughput* antar tiap ukuran *file* Perbedaan *throughput* lebih dipengaruhi oleh *bandwidth* pada jaringan, di mana semakin besar *bandwidth* maka semakin besar *throughput*.

Analisa *throughput* pada *home network* berikutnya, yakni ketika *mobile node* mengakses halaman *web* saat berada di *home network* menggunakan metode *route optimization* (skenario 3). Proses pengambilan data *throughput* menggunakan *wireshark* saat *mobile node* mulai mengakses halaman *web* sampai halaman *web* selesai terakses. Hasil *capture* menggunakan *wireshark* pada skenario 3 dapat dilihat pada Gambar 4.3 sebagai berikut.

Display			
Display filter: http			
Ignored packets: 0			
Traffic	Captured	Displayed	Marked
Packets	547	94	0
Between first and last packet	17.475 sec	0.817 sec	
Avg. packets/sec	31.302	115.081	
Avg. packet size	876.673 bytes	1156.798 bytes	
Bytes	479540	108739	
Avg. bytes/sec	27441.624	133125.943	
Avg. MBit/sec	0.220	1.065	

Gambar 4.3 *Throughput* pada *summary wireshark* (skenario 3)

Berdasarkan Gambar 4.3 di atas, nilai *throughput* hasil pencuplikan menggunakan *wireshark* sebesar 1.065 Mbit/sec. *Throughput* dari pengambilan data dengan ukuran *file* yang berbeda memiliki nilai yang berbeda juga. Grafik perbandingan *throughput* pada pengambilan data sebanyak sepuluh kali untuk skenario 3 dapat dilihat pada Lampiran B.

Setelah mendapat seluruh data nilai *throughput* untuk setiap skenario, maka akan dicari nilai rata-rata *throughput* pada *file* HTML dengan ukuran 20 KB + 925

KB, 40 KB + 2899 KB, dan 80 KB + 6542 KB. Perbandingan nilai rata-rata *throughput* dapat dilihat pada Tabel 4.2 sebagai berikut.

Tabel 4.2 Data nilai rata-rata *throughput* pada skenario 3

File Size (Kbytes)	Throughput (bps)
20 + 925	1,0001893
40 + 2899	0,8186864
80 + 6542	0,4887189

Dari Tabel 4.2, secara umum terlihat bahwa nilai rata-rata *throughput* memiliki nilai yang berbeda. Terkait dengan Tabel 4.1 yang merupakan perbandingan nilai rata-rata *throughput* pada skenario 1, terlihat bahwa terdapat persamaan nilai rata-rata *throughput* yang semakin mengecil seiring dengan ukuran *file* HTML yang semakin besar. Dilihat dari perhitungan menggunakan persentase, diketahui bahwa ukuran *file* HTML 20 KB + 925 KB memiliki persentase 18.15% lebih besar dibandingkan dengan HTML yang berukuran 40 KB + 2899 KB, dan *file* HTML berukuran 80 KB + 6542 KB memiliki persentase 67.52% lebih kecil dibandingkan dengan 40 KB + 2899 KB. Sama halnya seperti skenario 1, terjadinya perbedaan *throughput* pada skenario 3 dapat disebabkan oleh kepadatan *traffic* jaringan yang dapat menyebabkan data pada jaringan menjadi *overload*.

4.2.2 Analisa *Packet Loss*

Packet loss merupakan sebuah parameter yang menggambarkan suatu kondisi mengenai jumlah paket yang hilang. *Packet loss* dapat terjadi akibat adanya tabrakan antar paket saat dikirim (*collision*) atau terdapat tumpukan pada jaringan (*congestion*) yang menyebabkan beban pada jaringan menjadi besar sehingga terdapat beberapa paket yang hilang pada saat proses pengiriman paket. Filterisasi yang digunakan untuk mengetahui apabila terjadi *packet loss* pada hasil *capture wireshark* menggunakan filter `tcp.analysis.lost_segment`. Data *packet loss* yang pertama akan

diambil adalah saat *mobile node* berada di *home network* menggunakan metode *bidirectional tunneling* (skenario 1) dan berikutnya *packet loss* saat *mobile node* masih berada di *home network* tetapi menggunakan metode *route optimization* (skenario 3). Ketika memantau *traffic* yang sedang terjadi saat *mobile node* mengakses halaman *web* pada *correspondent node*, akan terlihat berbagai proses pada protokol-protokol jaringan. paket-paket yang mengalami *packet loss*. Pada *capture wireshark*, paket yang mengalami *packet loss* akan ditandai dengan warna hitam pada *background capture wireshark*. *Packet loss* pada skenario 1 yang tertangkap menggunakan *wireshark* dapat dilihat pada Gambar 4.4 berikut.

No.	Time	Source	Destination	Protocol	Length	Info
151	1.940429	2001:db8:ffff:100b:	2001:db8:ffff:100a:	TCP	1514	[TCP Previous segment lost]
191	2.489962	2001:db8:ffff:100b:	2001:db8:ffff:100a:	TCP	1514	[TCP Previous segment lost]
238	3.073151	2001:db8:ffff:100b:	2001:db8:ffff:100a:	HTTP	1514	[TCP Previous segment lost]
258	3.091813	2001:db8:ffff:100b:	2001:db8:ffff:100a:	TCP	98	[TCP Previous segment lost]
393	3.285817	2001:db8:ffff:100b:	2001:db8:ffff:100a:	TCP	1514	[TCP Previous segment lost]

Gambar 4.4 *Packet loss* pada *capture wireshark* (skenario 1)

Dari Gambar 4.4 tersebut terlihat bahwa terdapat lima paket yang mengalami *packet loss*. *Packet loss* pada Gambar 4.4 merupakan data ketiga yang diambil pada ukuran *file* HTML 80 KB + 6542 KB. *Packet loss* pada *summary wireshark* dapat dilihat pada Gambar 4.5 di bawah.

Display			
Display filter:		tcp.analysis.lost_segment	
Ignored packets:		0	
Traffic	Captured	Displayed	Marked
Packets	649	5	0
Between first and last packet	23.162 sec	1.345 sec	
Avg. packets/sec	28.020	3.716	
Avg. packet size	775.492 bytes	1230.800 bytes	
Bytes	503294	6154	
Avg. bytes/sec	21729.346	4574.145	
Avg. MBit/sec	0.174	0.037	

Gambar 4.5 *Packet loss* pada *summary wireshark* (skenario 1)

Terkait dengan Gambar 4.5 di atas, dapat dilihat bahwa terjadi lima kali *packet loss* dari total jumlah paket sebesar 649 paket yang tertangkap menggunakan *wireshark*. *Packet loss* belum tentu akan terjadi pada setiap data *file* HTML yang akan di-*capture* pada *wireshark*. Pengambilan data dilakukan sebanyak sepuluh kali untuk tiga ukuran *file* yang berbeda pada *home network* menggunakan metode *bidirectional tunneling*. Perbandingan nilai *packet loss* yang diambil sebanyak sepuluh kali dapat dilihat pada bagian Lampiran B.

Dari data-data *packet loss* yang telah dicuplik menggunakan *wireshark* dapat dicari rata-rata *packet loss* untuk setiap ukuran *file* HTML. Nilai rata-rata *packet loss* dapat dilihat pada Tabel 4.3 berikut ini.

Tabel 4.3 Data nilai rata-rata *packet loss* pada skenario 1

File Size (Kbytes)	Packet Loss
20 + 925	0
40 + 2899	0
80 + 6542	0,79

Berdasarkan data nilai rata-rata *packet loss* pada Tabel 4.3 di atas, secara umum tidak banyak terjadi *packet loss* pada *home network* menggunakan metode *bidirectional tunneling*. *Packet loss* hanya terjadi satu kali saja, yaitu saat pengambilan data ketiga pada ukuran *file* HTML 80 KB + 6542 KB. Pada data ketiga yang diambil pada HTML 80 KB + 6542 KB, persentase perbandingan *packet loss* yang terjadi dengan jumlah paket yang tertangkap adalah 4.5%. Dari 649 paket yang tertangkap, terdapat 4.5 % atau sebanyak lima paket yang mengalami *packet loss*. Pada data-data lainnya di masing-masing ukuran *file* HTML, persentase perbandingan *packet loss* adalah 0% karena tidak terjadi paket yang mengalami *packet loss*. Dapat dikatakan bahwa perbedaan rata-rata *packet loss* pada skenario 1 tidak terlalu signifikan.

Pengambilan data *packet loss* selanjutnya adalah pada *home network* menggunakan metode *route optimization* (skenario 3). Sama halnya pada skenario 1, untuk melihat *packet loss* pada *capture wireshark* perlu dilakukan filterisasi *packet loss* terlebih dahulu. *Packet loss* yang di *capture* menggunakan *wireshark* pada skenario 3 dapat dilihat pada Gambar 4.6 berikut ini.

No.	Time	Source	Destination	Protocol	Length	Info
76	1.152732	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
99	1.176881	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	86	[TCP Previous segment lost]
100	1.180787	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
175	1.291575	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
271	1.396735	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
456	1.766845	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]

Gambar 4.6 *Packet loss* pada *capture wireshark* (skenario 3)

Dari Gambar 4.6 di atas, terlihat bahwa terjadi enam paket yang mengalami *packet loss* pada skenario 3. Hasil *capture packet loss* pada skenario 3 dapat dilihat pada 4.7 sebagai berikut.

Display			
Display filter:	tcp.analysis.lost_segment		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	607	6	0
Between first and last packet	26.387 sec	0.614 sec	
Avg. packets/sec	23.004	9.770	
Avg. packet size	800.783 bytes	1276.000 bytes	
Bytes	486075	7656	
Avg. bytes/sec	18421.186	12466.759	
Avg. MBit/sec	0.147	0.100	

Gambar 4.7 *Packet loss* pada *summary wireshark* (skenario 3)

Jika dilihat dari Gambar 4.7 di atas, dapat diketahui bahwa terjadi *packet loss* sebanyak enam kali pada hasil *capture wireshark* tersebut. Perbandingan hasil pengambilan data *packet loss* untuk setiap ukuran *file HTML* pada skenario 3 dapat dilihat pada Lampiran B.

Dari data-data *packet loss* yang telah dicuplik menggunakan *wireshark* dapat dicari rata-rata *packet loss* untuk setiap ukuran *file* HTML. Nilai rata-rata *packet loss* dapat dilihat pada Tabel 4.4 berikut ini.

Tabel 4.4 Data nilai rata-rata *packet loss* pada skenario 3

File Size (Kbytes)	Packet Loss
20 + 925	0
40 + 2899	0,99
80 + 6542	0

Packet loss pada skenario 3 hanya terjadi pada ukuran *file* HTML 40 KB + 2899 KB, yaitu pada saat pengambilan data ketujuh. Terdapat enam *packet loss* yang terjadi saat pengambilan data ketujuh pada HTML 40 KB + 2899 KB. Pada data tersebut, persentase perbandingan *packet loss* yang terjadi dengan jumlah paket yang tertangkap adalah 5.9% dari 607 total jumlah paket. Pada data-data lainnya di masing-masing ukuran *file* HTML, persentase perbandingan *packet loss* adalah 0% karena tidak terjadi paket yang mengalami *packet loss*. *Packet loss* yang terjadi tidak terlalu signifikan diantara setiap ukuran *file* HTML.

4.2.3 Analisa Delay

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari sumber (*source*) ke tujuan (*destination*). Parameter *delay* didapatkan melalui perhitungan antara nilai *transfer time* dengan jumlah bit data. Nilai *transfer time* akan dibagi dengan jumlah bit data, sehingga perumusan nilai *delay* adalah sebagai berikut.

$$Delay (s) = \frac{Transfer\ time\ (s)}{Jumlah\ bit}$$

Pengambilan data *delay* pertama dilakukan untuk skenario 1 dan kemudian untuk skenario 3. Nilai *delay* akan dicuplik menggunakan *wireshark* yang

sebelumnya telah difilter paket-paket HTTP saja. Hasil *capture delay* pada skenario 1 dapat dilihat pada Gambar 4.8 sebagai berikut

Display			
Display filter: http			
Ignored packets: 0			
Traffic	Captured	Displayed	Marked
Packets	550	101	0
Between first and last packet	19.401 sec	0.968 sec	
Avg. packets/sec	28.349	104.288	
Avg. packet size	873.005 bytes	1185.475 bytes	
Bytes	480153	119733	
Avg. bytes/sec	24749.054	123631.350	
Avg. MBit/sec	0.198	0.989	

Gambar 4.8 *Delay* pada *summary wireshark* (skenario1)

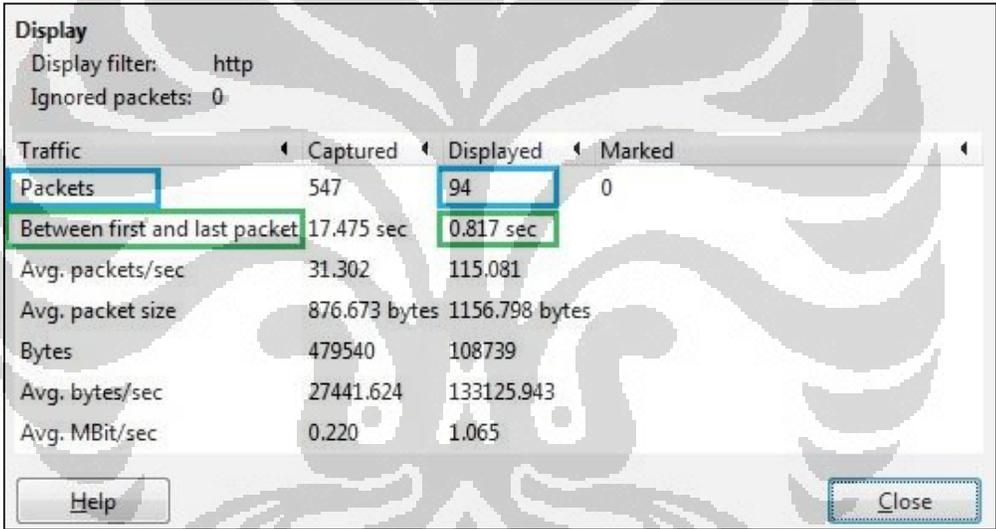
Pada Gambar 4.8 di atas, terlihat bahwa data salah satu *file* HTML 20 KB + 925 KB terdapat 101 paket dan *transfer time* atau pada *summary wireshark* di atas adalah *between first and last packet* selama 0.968 detik. Nilai *delay* didapat dari hasil pembagian *transfer time* selama 0.968 detik dengan paket sebanyak 101 paket sesuai dengan rumus perhitungan *delay* di atas. Sehingga *delay* yang terjadi pada data tersebut adalah selama 0.009 detik. Perbandingan nilai *delay* untuk setiap ukuran *file* HTML pada skenario 1 dapat dilihat pada Lampiran B. Berdasarkan data nilai *delay* pada Lampiran B, dapat dihitung rata-rata nilai *delay* untuk setiap ukuran *file* HTML. Nilai rata-rata *delay* skenario 1 dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.5 Data nilai rata-rata *delay* pada skenario 1

File Size (Kbytes)	Delay (second)
20 + 925	0,0103947
40 + 2899	0,0116943
80 + 6542	0,0200097

Dari Tabel 4.5 mengenai perbandingan rata-rata nilai *delay*, dapat dilihat bahwa semakin besar ukuran *file* HTML maka semakin besar *delay* yang terjadi pada jaringan. *Delay* terbesar terjadi saat ukuran *file* HTML 80 KB + 6542 KB sebesar 0.0200097. Berdasarkan persentase, HTML 80 KB + 6542 KB lebih besar 41.56 % dibandingkan dengan HTML 40 KB + 2899 KB dan lebih besar 48.51 % dibandingkan dengan HTML 20 KB + 925 KB.

Pengambilan data berikutnya dilakukan pada skenario 3, yaitu pada saat *mobile node* berada di *home network* menggunakan metode *route optimization*. Hasil *capture delay* pada skenario 3 dapat dilihat Gambar 4.9 di bawah ini.



The screenshot shows the 'Display' window in Wireshark with a filter set to 'http'. The traffic summary table is as follows:

Traffic	Captured	Displayed	Marked
Packets	547	94	0
Between first and last packet	17.475 sec	0.817 sec	
Avg. packets/sec	31.302	115.081	
Avg. packet size	876.673 bytes	1156.798 bytes	
Bytes	479540	108739	
Avg. bytes/sec	27441.624	133125.943	
Avg. MBit/sec	0.220	1.065	

Gambar 4.9 *Delay* pada *summary wireshark* (skenario3)

Perbandingan nilai *delay* untuk setiap ukuran *file* HTML pada skenario 3 dapat dilihat pada Lampiran B. Terkait dengan hasil data *delay* yang telah diambil sebanyak sepuluh kali pada setiap ukuran *file* HTML, dapat dicari nilai rata-rata *delay* untuk dibandingkan dengan ukuran *file* HTML lainnya pada skenario 3. Nilai rata-rata *delay* skenario 3 dapat dilihat pada Tabel 4.6 berikut.

Tabel 4.6 Data nilai rata-rata *delay* pada skenario 3

File Size (Kbytes)	Delay (second)
20 + 925	0,010133473
40 + 2899	0,011845199
80 + 6542	0,02030494

Pada Tabel 4.6 di atas, terlihat perbandingan nilai rata-rata *delay* setiap ukuran *file* HTML pada skenario 3, di mana semakin besar ukuran *file* HTML maka semakin besar nilai *delay*. Sama halnya seperti skenario 1, *delay* terbesar terjadi saat ukuran *file* HTML 80 KB + 6542 KB. HTML 80 KB + 6542 KB lebih besar 41.66% dibandingkan dengan HTML 40 KB + 2899 KB, dan lebih besar 50.09% dibandingkan dengan HTML 20 KB + 925 KB. Besar-kecilnya nilai *delay* dapat dipengaruhi oleh penumpukan data-data yang terjadi saat proses pengiriman data, sehingga data-data akan membutuhkan waktu lebih lama dibanding waktu aslinya.

4.3 Analisa *Foreign Network*

Analisa pada *foreign network* merupakan analisa yang dilakukan pada saat *mobile node* berada pada *foreign network*. Analisa ini dilakukan mengikuti skenario 2 dan skenario 4, dimana akan terdapat perbedaan metode dalam komunikasi antara *mobile node* dengan *correspondent node*. Metode tersebut adalah *bidirectional tunneling* yang digunakan pada skenario 2 dan *route optimization* yang digunakan pada skenario 4. Dengan melakukan analisa ini, diharapkan dapat mengetahui perbedaan kualitas layanan yang diterima *mobile node* pada saat berada di *foreign network*.

4.3.1 Analisa *Throughput*

Pengambilan data untuk analisa *throughput* skenario 2 dilakukan pada saat *mobile node* berada pada *foreign network* menggunakan metode *bidirectional tunneling*. Hasil pencuplikan *throughput* pada skenario 2 dapat dilihat pada Gambar 4.10.

Display			
Display filter:	http		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	672	123	0
Between first and last packet	18.015 sec	2.634 sec	
Avg. packets/sec	37.302	46.690	
Avg. packet size	752.943 bytes	1297.561 bytes	
Bytes	505978	159600	
Avg. bytes/sec	28086.289	60583.079	
Avg. MBit/sec	0.225	0.485	

Gambar 4.10 *Throughput* pada *summary wireshark* (skenario 2)

Terkait dengan Gambar 4.10 di atas, *throughput* salah satu data HTML 20 KB + 925 KB pada skenario 2 adalah sebesar 0.485 Mbit/sec. Selanjutnya akan dilakukan pengambilan data sebanyak sepuluh kali pada setiap ukuran *file* HTML. Grafik perbandingan nilai *throughput* untuk setiap ukuran *file* HTML dapat dilihat pada Lampiran B. Dari hasil *capture wireshark*, didapatkan nilai-nilai *throughput* yang kemudian akan dicari nilai rata-rata *throughput* pada setiap ukuran *file* HTML. Nilai rata-rata *throughput* pada skenario 2 ditunjukkan oleh Tabel 4.7.

Tabel 4.7 Data nilai rata-rata *throughput* pada skenario 2

File Size (Kbytes)	Throughput (bps)
20 + 925	0,8983
40 + 2899	0,7734
80 + 6542	0,6121

Berdasarkan data nilai rata-rata *throughput* pada Tabel 4.7 di atas, HTML berukuran 20 KB + 925 KB memiliki *throughput* paling besar dibandingkan dengan ukuran *file* HTML lainnya. Secara persentase, HTML 20 KB + 925 KB lebih besar 13.90% dibandingkan dengan HTML 40 KB + 2899 KB dan lebih besar 31.86% dibandingkan dengan HTML 80 KB + 6542 KB. Pengambilan data pada *foreign network* berikutnya dilakukan pada skenario 4. Hasil *capture wireshark* pada skenario 4 menggunakan metode *route optimization* dapat dilihat pada Gambar 4.11 berikut.

Display			
Display filter:	http		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	652	133	0
Between first and last packet	45.929 sec	0.846 sec	
Avg. packets/sec	14.196	157.146	
Avg. packet size	755.092 bytes	1234.180 bytes	
Bytes	492320	164146	
Avg. bytes/sec	10719.132	193946.215	
Avg. MBit/sec	0.086	1.552	

Gambar 4.11 *Throughput* pada *summary wireshark* (skenario 4)

Dari data yang telah dicuplik menggunakan *wireshark* sebanyak sepuluh kali, didapatkan bahwa nilai *throughput* untuk setiap ukuran *file* HTML berbeda-beda. Grafik nilai *throughput* untuk HTML 20 KB + 925 KB, 40 KB + 2899 KB, dan 80 KB + 6542 KB dapat dilihat pada Lampiran B. Dari hasil *capture wireshark* di atas, didapatkan nilai-nilai *throughput* yang kemudian akan dicari nilai rata-rata *throughput*. Nilai rata-rata *throughput* pada skenario 4 ditunjukkan oleh Tabel 4.8 sebagai berikut.

Tabel 4.8 Data nilai rata-rata *throughput* pada skenario 4

File Size (Kbytes)	Throughput (bps)
20 + 925	1,25968
40 + 2899	1,21391
80 + 6542	1,11971

HTML dengan ukuran 20 KB + 925 memiliki nilai rata-rata *throughput* paling besar dibandingkan *file* HTML lainnya. HTML 20 KB + 925 KB memiliki persentasi nilai rata-rata *throughput* 3.63% lebih besar dibandingkan HML 40 KB + 2899 KB dan 1.11% lebih besar dibandingkan dengan HTML 80KB.

4.3.2 Analisa *Packet Loss*

Pengambilan data untuk parameter *packet loss* akan dilakukan pada skenario 2 terlebih dahulu dan kemudian dilakukan pada skenario 4. *Packet loss* yang tertangkap menggunakan *wireshark* pada skenario 2 di mana menggunakan metode *bidirectional tunneling* dapat dilihat pada Gambar 4.12 berikut.

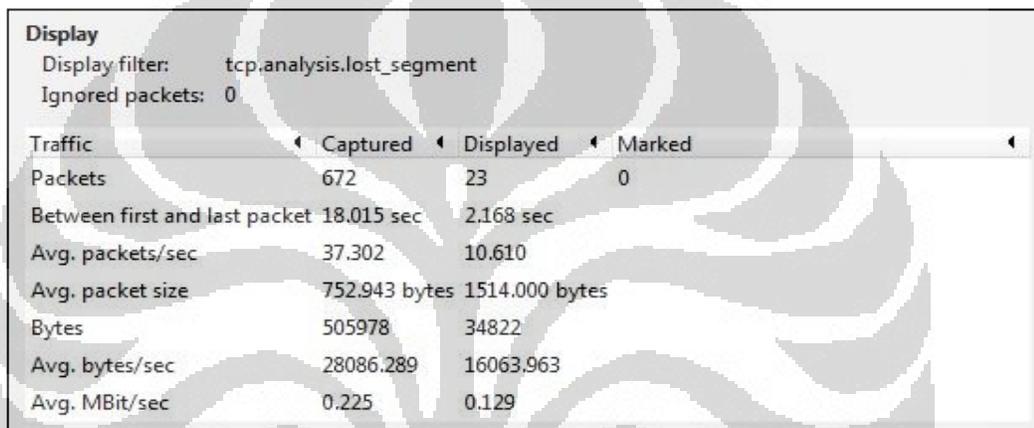
No.	Time	Source	Destination	Protocol	Length	Info
112	1.103760	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
116	1.106979	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
118	1.107306	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
124	1.111773	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
147	1.322483	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
152	1.443188	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
158	1.488478	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
211	1.561356	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
223	1.579901	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
247	1.601169	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
267	1.621441	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
291	1.858542	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
353	1.923243	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
367	2.043227	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
377	2.112899	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
389	2.243646	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]

Gambar 4.12 *Packet loss* pada *capture wireshark* (skenario 2)

Packet loss yang terjadi pada skenario 2 merujuk pada Gambar 4.12 di atas, terdapat banyak paket yang mengalami *packet loss*. Untuk mengetahui berapa jumlah

paket yang mengalami *packet loss*, dapat dilihat melalui *summary wireshark*. Hasil *capture wireshark* pada skenario 2 ditunjukkan oleh Gambar 4.13.

Berdasarkan *summary wireshark* pada Gambar 4.13, terlihat bahwa paket yang mengalami *packet loss* berjumlah 23 paket. Dapat diketahui persentase antara paket yang mengalami *packet loss* dengan jumlah paket yang tertangkap, yaitu sebesar 3.4%. Perbandingan *packet loss* yang terjadi di setiap ukuran *file HTML* dapat dilihat pada Lampiran B.



Display			
Display filter:	tcp.analysis.lost_segment		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	672	23	0
Between first and last packet	18.015 sec	2.168 sec	
Avg. packets/sec	37.302	10.610	
Avg. packet size	752.943 bytes	1514.000 bytes	
Bytes	505978	34822	
Avg. bytes/sec	28086.289	16063.963	
Avg. MBit/sec	0.225	0.129	

Gambar 4.13 *Packet loss* pada *summary wireshark* (skenario 2)

Packet loss yang terjadi berdasarkan grafik perbandingan pada Lampiran B, menunjukkan bahwa *packet loss* selalu terjadi di setiap ukuran *file HTML*. Nilai rata-rata *packet loss* yang terjadi pada skenario 2 dapat dilihat pada Tabel 4.9 di bawah ini.

Tabel 4.8 Data nilai rata-rata *packet loss* pada skenario 2

File Size (Kbytes)	Packet Loss
20 + 925	2,23
40 + 2899	1,96
80 + 6542	2,17

Berdasarkan Tabel 4.8 di atas, dapat diketahui nilai rata-rata *packet loss* tiap-tiap ukuran *file* HTML setelah dilakukan pengambilan data sebanyak sepuluh kali. Pada HTML 20 KB + 925 KB terdapat 14.1 paket yang mengalami *packet loss*, HTML 40 KB + 2899 KB terdapat 12.2 paket yang mengalami *packet loss*, dan HTML 80 terdapat 13.7 paket yang mengalami *packet loss*.

Pengambilan data *packet loss* berikutnya dilakukan pada skenario 4 yang menggunakan metode *route optimization*. *Packet loss* yang tertangkap menggunakan *wireshark* pada skenario 4 dapat dilihat pada Gambar 4.14 berikut.

No.	Time	Source	Destination	Protocol	Length	Info
80	1.235837	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
118	1.344119	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
120	1.344528	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	86	[TCP Previous segment lost]
135	1.353837	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
161	1.376517	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
239	1.459028	2001:db8:ffff:100b	2001:db8:ffff:100c	HTTP	1514	[TCP Previous segment lost]
275	1.516656	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
358	1.615074	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]
453	1.712379	2001:db8:ffff:100b	2001:db8:ffff:100c	TCP	1514	[TCP Previous segment lost]

Gambar 4.14 *Packet loss* pada *capture wireshark* (skenario 4)

Packet loss yang di *capture* menggunakan *wireshark* pada skenario 4 dapat dilihat pada Gambar 4.15 berikut ini.

Display			
Display filter:	tcp.analysis.lost_segment		
Ignored packets:	0		
Traffic	Captured	Displayed	Marked
Packets	620	9	0
Between first and last packet	22.192 sec	0.477 sec	
Avg. packets/sec	27.938	18.886	
Avg. packet size	785.853 bytes	1355.333 bytes	
Bytes	487229	12198	
Avg. bytes/sec	21954.826	25596.905	
Avg. MBit/sec	0.176	0.205	

Gambar 4.15 *Packet loss* pada *summary wireshark* (skenario 4)

Dari Gambar 4.15 diketahui bahwa terdapat sembilan paket yang mengalami *packet loss*. Persentase antara paket yang mengalami *packet loss* dengan jumlah paket

yang tertangkap dari data di atas, yaitu sebesar 1.4%. Untuk mengetahui lebih lanjut tentang perbandingan antara nilai *packet loss* di setiap ukuran *file* HTML dapat dilihat pada Lampiran B.

Packet loss pada skenario 4 terjadi pada HTML 40 KB + 2899 KB sebanyak satu kali dengan jumlah paket yang mengalami *packet loss* sebanyak sembilan paket saat pengambilan data keenam. *Packet loss* juga terjadi pada HTML 80 KB + 6542 KB sebanyak satu kali dengan jumlah paket yang mengalami *packet loss* sebanyak enam paket yang terjadi saat pengambilan data ketujuh. Nilai rata-rata *packet loss* yang terjadi pada tiap-tiap ukuran HTML dapat dilihat pada Tabel 4.10 berikut.

Tabel 4.10 Data nilai rata-rata *packet loss* pada skenario 4

File Size (Kbytes)	Packet Loss
20 + 925	0
40 + 2899	0,14
80 + 6542	0,98

Pada HTML 80 KB + 6542 KB dengan pengambilan data sebanyak sepuluh kali terdapat 0.6 paket yang mengalami *packet loss*. Pada HTML 40 KB + 2899 KB terdapat 0.4 paket yang mengalami *packet loss* dan HTML 20 KB + 925 KB tidak ada paket yang mengalami *packet loss*.

4.3.3 Analisa Delay

Pengambilan data *delay* pertama dilakukan untuk skenario 2 dan kemudian untuk skenario 4. Hasil *delay* untuk skenario 2 menggunakan metode *bidirectional tunneling* dapat dilihat pada *capture wireshark* ditunjukkan pada Gambar 4.16 sebagai berikut.

Display			
Display filter: http			
Ignored packets: 0			
Traffic	Captured	Displayed	Marked
Packets	672	123	0
Between first and last packet	18.015 sec	2.634 sec	
Avg. packets/sec	37.302	46.690	
Avg. packet size	752.943 bytes	1297.561 bytes	
Bytes	505978	159600	
Avg. bytes/sec	28086.289	60583.079	
Avg. MBit/sec	0.225	0.485	

Gambar 4.16 *Delay* pada *summary wireshark* (skenario 2)

Pada Gambar 4.16 *summary wireshark* di atas, dapat dilihat bahwa data HTTP tersebut memiliki jumlah paket sebanyak 123 paket dan waktu pengiriman seluruh paket selama 2.634 detik. *Delay* pada data tersebut kemudian dicari menggunakan rumus perhitungan *delay*, sehingga *delay* yang terjadi selama 0.0214 detik. Data akan diambil sebanyak sepuluh kali pada setiap ukuran *file* HTML yang kemudian akan dicari grafik perbandingan nilai *delay*. Grafik perbandingan nilai *delay* untuk setiap ukuran HTML pada skenario 2 menggunakan metode *bidirectional tunneling* dapat dilihat pada Lampiran B.

Terkait dengan hasil data *delay* yang telah diambil sebanyak sepuluh kali pada setiap ukuran *file* HTML, dapat dicari nilai rata-rata *delay* untuk dibandingkan dengan ukuran *file* HTML lainnya pada skenario 2. Nilai rata-rata *delay* skenario 2 dapat dilihat pada Tabel 4.11 berikut.

Tabel 4.11 Data nilai rata-rata *delay* pada skenario 2

File Size (Kbytes)	Delay (second)
20 + 925	0,012454562
40 + 2899	0,013341082
80 + 6542	0,018834178

Berdasarkan data pada Tabel 4.22 di atas, dapat dilihat bahwa *delay* terbesar terjadi pada saat mengakses halaman *web* berukuran 80 KB + 6542 KB. Dalam perhitungan persentase, HTML 80 KB + 6542 KB lebih besar 29.16% dibandingkan dengan HTML 40 KB + 2899 KB dan juga lebih besar 33.82% dibandingkan dengan HTML 20 KB + 925 KB.

Pengambilan data *delay* berikutnya dilakukan pada skenario 4 menggunakan metode *route optimization*. *Delay* pada skenario 4 dapat dilihat pada hasil *capture wireshark* pada Gambar 4.17.

Traffic	Captured	Displayed	Marked
Packets	652	133	0
Between first and last packet	45.929 sec	0.846 sec	
Avg. packets/sec	14.196	157.146	
Avg. packet size	755.092 bytes	1234.180 bytes	
Bytes	492320	164146	
Avg. bytes/sec	10719.132	193946.215	
Avg. MBit/sec	0.086	1.552	

Gambar 4.17 *Delay* pada *summary wireshark* (skenario 4)

Dari hasil *capture wireshark* pada data yang telah diambil sebanyak sepuluh kali untuk setiap ukuran *file HTML*, dapat dilihat perbandingan nilai-nilai *delay* pada skenario 4. Perbandingan tersebut dapat dilihat pada Lampiran B.

Nilai-nilai *delay* tersebut kemudian akan dicari nilai rata-rata pada setiap ukuran *file HTML*. Tabel 4.12 menunjukkan nilai rata-rata *delay* pada skenario 4.

Tabel 4.12 Data nilai rata-rata *delay* pada skenario 4

File Size (Kbytes)	Delay (second)
20 + 925	0,008819298
40 + 2899	0,009532501
80 + 6542	0,013139862

Berdasarkan data nilai rata-rata pada tabel 4.12, semakin besar ukuran *file* HTML maka akan semakin besar *delay* yang terjadi. HTML 80 KB + 6542 KB memiliki *delay* terbesar dibandingkan dengan HTML 20 KB + 925 KB dan 40 KB + 2899 KB. HTML 80 KB + 6542 KB lebih besar 27.45% dibandingkan HTML 40 KB + 2899 KB dan lebih besar 32.88% dibandingkan dengan HTML 20 KB + 925 KB.

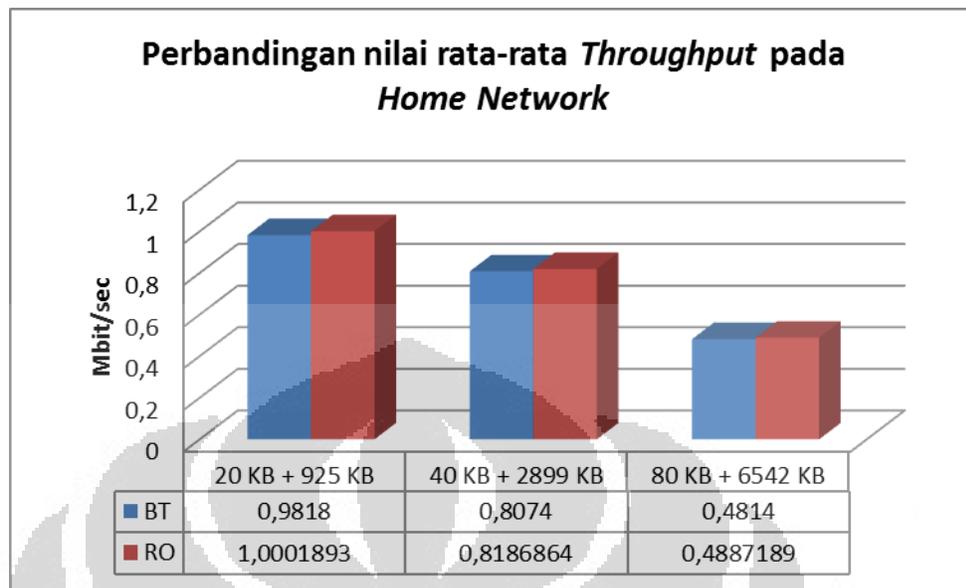
4.4 Analisa Perbandingan Skenario 1 dan Skenario 3

Analisa ini dilakukan untuk membandingkan hasil dari skenario 1 dan skenario 3 (*mobile node* berada di *home network*) berdasarkan parameter *throughput*, *packet loss*, dan *delay*. Analisa perbandingan ini bertujuan untuk mengetahui performansi jaringan *mobile* IPv6 mana yang lebih baik, ketika menggunakan *bidirectional tunneling* atau *route optimization*.

4.4.1 Analisa Perbandingan *Throughput*

Dari pengambilan data yang dilakukan pada skenario 1 dan skenario 3, akan didapatkan hasil data keseluruhan nilai *throughput* yang dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *throughput* pada skenario 1 dan skenario 3. Perbandingan tersebut dapat dilihat pada Gambar 4.18 sebagai berikut.



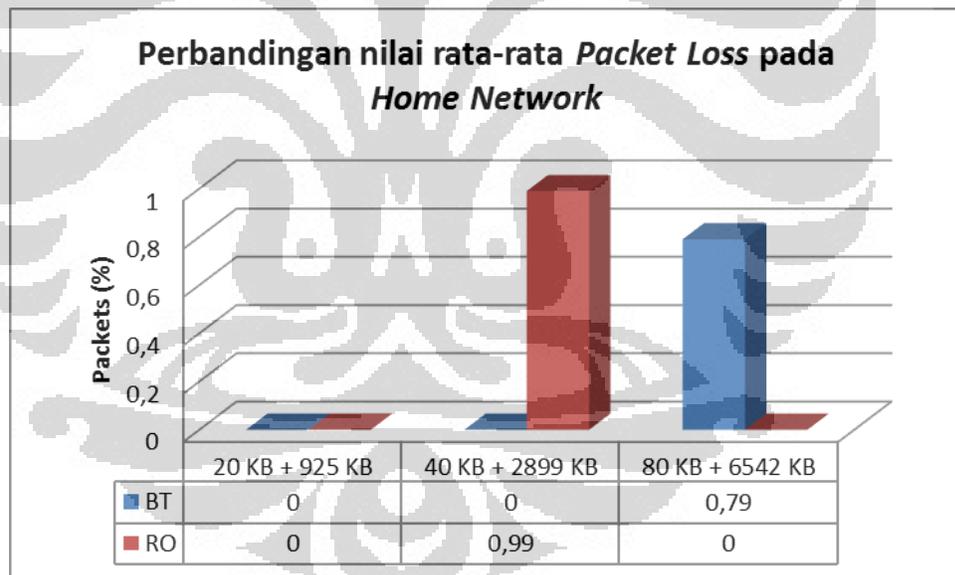
Gambar 4.18 Perbandingan *throughput* pada skenario 1 dan skenario 3

Terlihat pada Gambar 4.18, bahwa perubahan *throughput* pada masing-masing data HTML tidak terlalu signifikan. Pada HTML 20 KB + 925 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.9818 Mbit/sec dan saat menggunakan *route optimization* sebesar 1.0002 Mbit/sec. Terdapat selisih 0.0184 Mbit/sec (18.40% lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 40 KB + 2899 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.8074 Mbit/sec dan saat menggunakan *route optimization* sebesar 0.8187 Mbit/sec. Terdapat selisih 0.0113 Mbit/sec (13.80% lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 80 KB + 6542 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.4814 Mbit/sec dan saat menggunakan *route optimization* sebesar 0.4887 Mbit/sec. Terdapat selisih 0.0073 Mbit/sec (14.94% lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Perbedaan yang tidak terlalu signifikan pada masing-masing skenario disebabkan karena *mobile node* akan langsung mengakses halaman *web* pada *server*, sehingga proses yang terjadi pada kedua skenario hampir sama dan tidak terlalu mempengaruhi nilai *throughput* yang didapatkan. Selain itu perbedaan *throughput* lebih dipengaruhi oleh *bandwidth* pada jaringan, di mana semakin besar *bandwidth* maka semakin besar *throughput*.

4.4.2 Analisa Perbandingan *Packet loss*

Dari pengambilan data yang dilakukan pada skenario 1 dan skenario 3, akan didapatkan hasil data keseluruhan nilai *packet loss* yang dapat dilihat pada Tabel 4.3 dan Tabel 4.4. Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *packet loss* pada skenario 1 dan skenario 3. Perbandingan tersebut dapat dilihat pada Gambar 4.19.

Berdasarkan Gambar 4.19, secara umum hampir tidak terdapat *packet loss* yang terjadi baik pada skenario 1 dan skenario 3 pada masing-masing ukuran *file* HTML. Selama pengambilan data pada masing-masing skenario, paket yang mengalami *packet loss* hanya terjadi satu kali yaitu pada data ketiga HTML 80 KB + 6542 KB pada skenario 1 dan data ketujuh HTML 40 KB + 2899 KB pada skenario 3.



Gambar 4.19 Perbandingan *packet loss* pada skenario 1 dan skenario 3

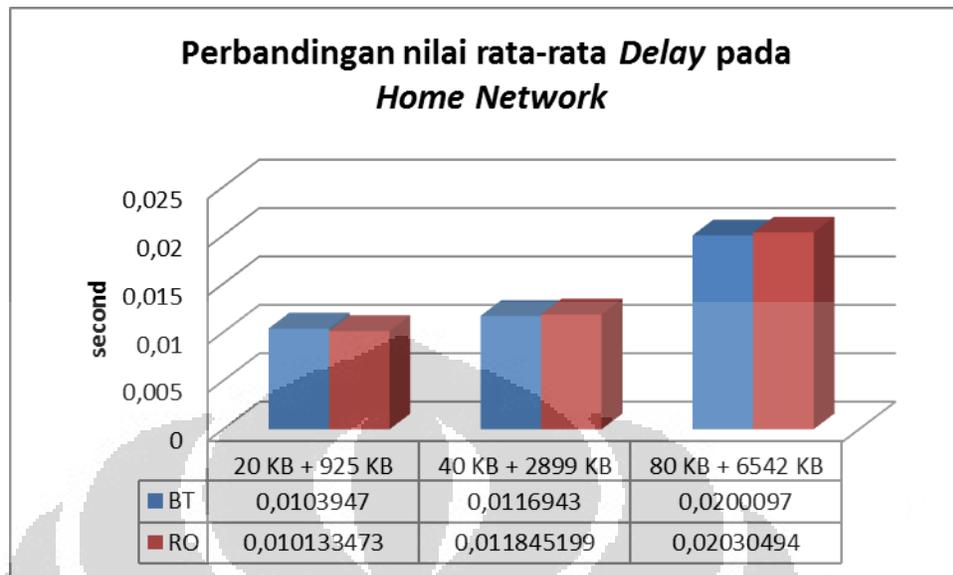
Pada HTML 20 KB + 925 KB, rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional tunneling* maupun *route optimization* sebesar 0%. Tidak ada perbedaan *packet loss* yang terjadi antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 40 KB + 2899 KB, rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional tunneling* sebesar 0% dan saat

menggunakan *route optimization* sebesar 0.99%. Paket yang mengalami *packet loss* pada *route optimization* yang diambil sebanyak sepuluh kali hanya terjadi satu kali saja, yaitu saat pengambilan data yang ketujuh (dapat dilihat pada Lampiran B) sebanyak enam paket. Pada HTML 80 KB + 6542 KB, rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional tunneling* sebesar 0.79% dan saat menggunakan *route optimization* sebesar 0%. Paket yang mengalami *packet loss* pada *bidirectional tunneling* yang diambil sebanyak sepuluh kali hanya terjadi satu kali saja, yaitu saat pengambilan data yang ketiga (dapat dilihat pada Lampiran B) sebanyak lima paket

Tidak banyaknya paket yang mengalami *packet loss* pada kedua skenario tersebut, disebabkan oleh tidak berbedanya proses pengambilan data dari *mobile node* ke *correspondent node* pada *bidirectional tunneling* maupun *route optimization*. Pada kedua metode tersebut, *mobile node* akan langsung mengakses data yang berada di *correspondent node*. Terdapatnya beberapa *packet loss* dapat disebabkan oleh kualitas performansi jaringan saat pengambilan data yang tidak stabil, sehingga terdapat paket yang mengalami *packet loss* saat melakukan proses akses data.

4.4.3 Analisa Perbandingan Delay

Dari pengambilan data yang dilakukan pada skenario 1 dan skenario 3, akan didapatkan hasil data keseluruhan nilai *delay* yang dapat dilihat pada Tabel 4.5 dan Tabel 4.6. Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *delay* pada skenario 1 dan skenario 3. Perbandingan tersebut dapat dilihat pada Gambar 4.20 berikut.



Gambar 4.20 Perbandingan *delay* pada skenario 1 dan skenario 3

Dari data di atas, dapat dilihat perubahan nilai *delay* pada kedua metode tersebut yang tidak terlalu jauh berbeda. Pada HTML 20 KB + 925 KB, *delay* ketika menggunakan *bidirectional tunneling* sebesar 0.0104 detik dan saat menggunakan *route optimization* sebesar 0.0101 detik. Terdapat selisih 0.0003 detik (2.97% lebih kecil) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 40 KB + 2899 KB, *delay* saat menggunakan *bidirectional tunneling* sebesar 0.0117 dan saat menggunakan *route optimization* sebesar 0.0118. Hanya terdapat selisih 0.0001 detik saja (0.85 lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 80 KB + 6542 KB, *delay* saat menggunakan *bidirectional tunneling* sebesar 0.02 detik dan saat menggunakan *route optimization* sebesar 0.0203 detik. Terdapat selisih 0.0003 detik (1.48% lebih besar) antara *route optimization* dengan *bidirectional tunneling*.

Selisih yang kecil pada nilai rata-rata *delay* di setiap skenario baik pada HTML 20 KB + 925 KB, HTML 40 KB + 2899 KB, maupun HTML 80 KB + 6542 KB, disebabkan oleh jarak *hop* antara *mobile node* dengan *correspondent node* yang hampir sama sehingga *transfer time* yang dibutuhkan untuk mengirim data juga hampir sama. *Delay* sangat dipengaruhi oleh waktu pengiriman paket (*transfer time*) dan juga jumlah paket yang dikirim. Kualitas jaringan pun dapat mempengaruhi nilai

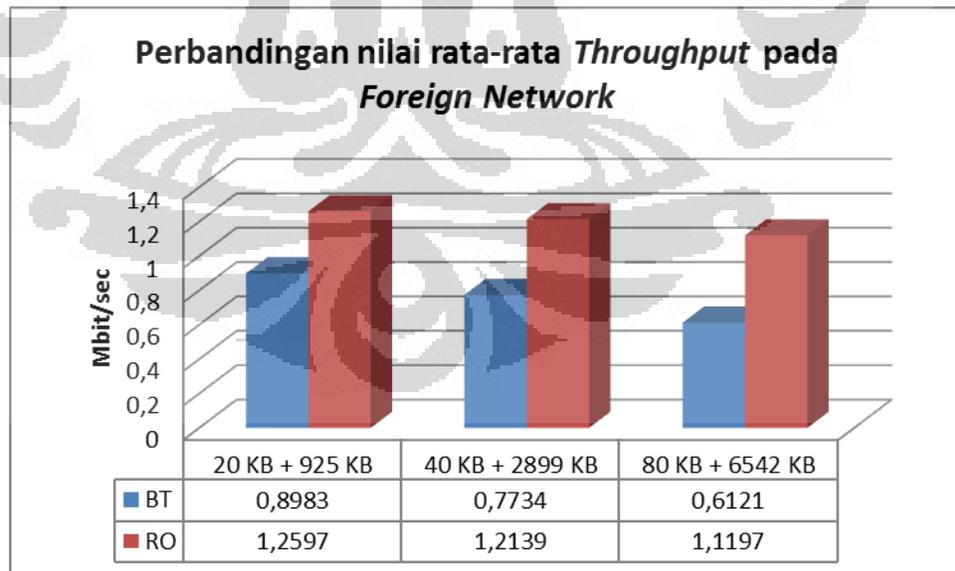
delay, karena jika performansi jaringan kurang bagus maka dapat terjadi tabrakan atau penumpukan data pada jaringan yang dapat menghambat proses pengiriman data. Hal ini akan menyebabkan pada bertambahnya waktu pengiriman data dan *delay*-pun akan ikut bertambah.

4.5 Analisa Perbandingan Skenario 2 dan Skenario 4

Analisa ini dilakukan untuk membandingkan hasil dari skenario 1 dan skenario 3 (*mobile node* berada di *foreign link*) berdasarkan parameter *throughput*, *packet loss*, dan *delay*. Analisa perbandingan ini bertujuan untuk mengetahui performansi jaringan *mobile IPv6* mana yang lebih baik, ketika menggunakan *bidirectional tunneling* atau *route optimization*.

4.5.1 Analisa Perbandingan *Throughput*

Dari nilai rata-rata *throughput* pada Tabel 4.7 dan Tabel 4.8, akan didapatkan hasil data keseluruhan nilai *throughput*. Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *throughput* pada skenario 1 dan skenario 3. Perbandingan tersebut dapat dilihat pada Gambar 4.21 sebagai berikut.



Gambar 4.21 Perbandingan *throughput* pada skenario 2 dan skenario 4

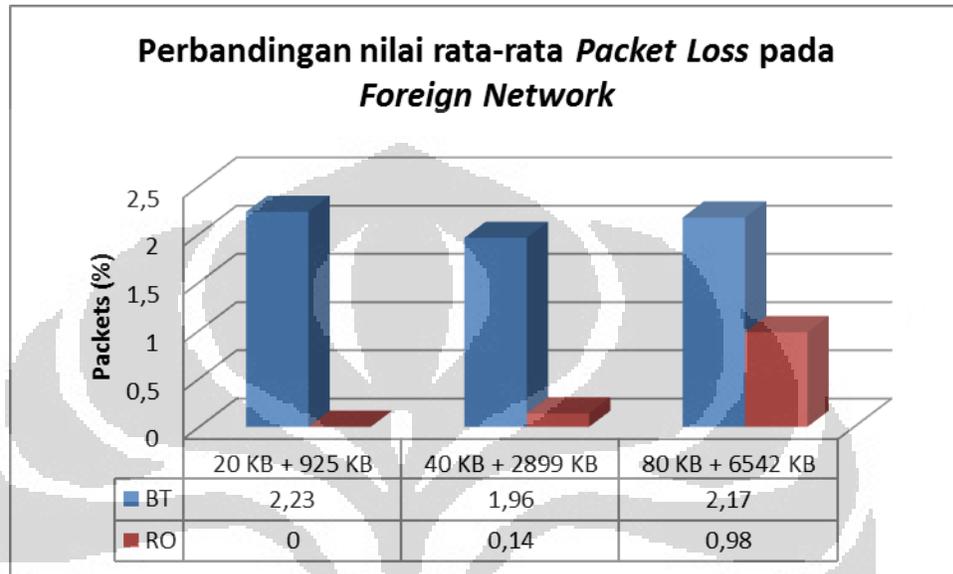
Pada Gambar 4.21 di atas, terlihat bahwa perubahan *throughput* saat berada di *foreign link* pada masing-masing data HTML cukup signifikan. Pada HTML 20 KB + 925 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.8983 Mbit/sec dan saat menggunakan *route optimization* sebesar 1.2597 Mbit/sec. Terdapat selisih 0.3614 Mbit/sec (28.69% lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 40 KB + 2899 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.7734 Mbit/sec dan saat menggunakan *route optimization* sebesar 1.2139 Mbit/sec. Terdapat selisih 0.4405 Mbit/sec (36.29% lebih besar) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 80 KB + 6542 KB, *throughput* saat menggunakan *bidirectional tunneling* sebesar 0.6121 Mbit/sec dan saat menggunakan *route optimization* sebesar 1.1197 Mbit/sec. Terdapat selisih 0.5076 Mbit/sec (45.33% lebih besar) antara *route optimization* dengan *bidirectional tunneling*.

Perbedaan *throughput* pada skenario 2 dan skenario 4 terjadi karena jarak *hop* antara *mobile node* dengan *correspondent node* yang berbeda, di mana pada *bidirectional tunneling* memiliki *hop* yang lebih banyak dibandingkan dengan *route optimization*. Pada *bidirectional tunneling*, *correspondent node* harus melewati *Home Agent* terlebih dahulu sebelum dapat mengirimkan permintaan dari *mobile node* untuk mengakses halaman *web* yang berada di *server*. Hal ini dapat terjadi dikarenakan *mobile node* tidak melakukan *binding update* pada *correspondent node* sehingga setiap kali *mobile node* akan mengakses data yang berada pada *correspondent node*, *mobile node* harus melewati *Home Agent* terlebih dahulu sebelum dikirimkan secara normal ke *correspondent node*. Setiap paket yang diterima kemudian akan dikirimkan ke alamat *Care of Address* dari *mobile node*. Selain itu kualitas jaringan dapat mempengaruhi *throughput* pada jaringan *mobile IPv6*.

4.5.2 Analisa Perbandingan *Packet Loss*

Dari pengambilan data yang dilakukan pada skenario 2 dan skenario 4, akan didapatkan hasil data keseluruhan nilai *packet loss* yang dapat dilihat pada Tabel 4.9 dan Tabel 4.10.

Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *packet loss* pada skenario 2 dan skenario 4. Perbandingan tersebut dapat dilihat pada Gambar 4.22.



Gambar 4.22 Perbandingan *packet loss* pada skenario 2 dan skenario 4

Berdasarkan Gambar 4.22 di atas, *packet loss* yang terjadi pada skenario 2 dan skenario 4 sangat signifikan pada masing-masing ukuran *file* HTML. Saat menggunakan metode *bidirectional tunneling*, *packet loss* selalu terjadi di setiap proses pengiriman paket yang dilakukan sebanyak sepuluh kali. Hal ini berbanding terbalik dengan saat menggunakan metode *route optimization*. Pada *route optimization* paket yang mengalami *packet loss* hanya terjadi beberapa kali saja. Banyaknya paket yang mengalami *packet loss* dapat dilihat pada Lampiran B untuk skenario 2 dan skenario 4.

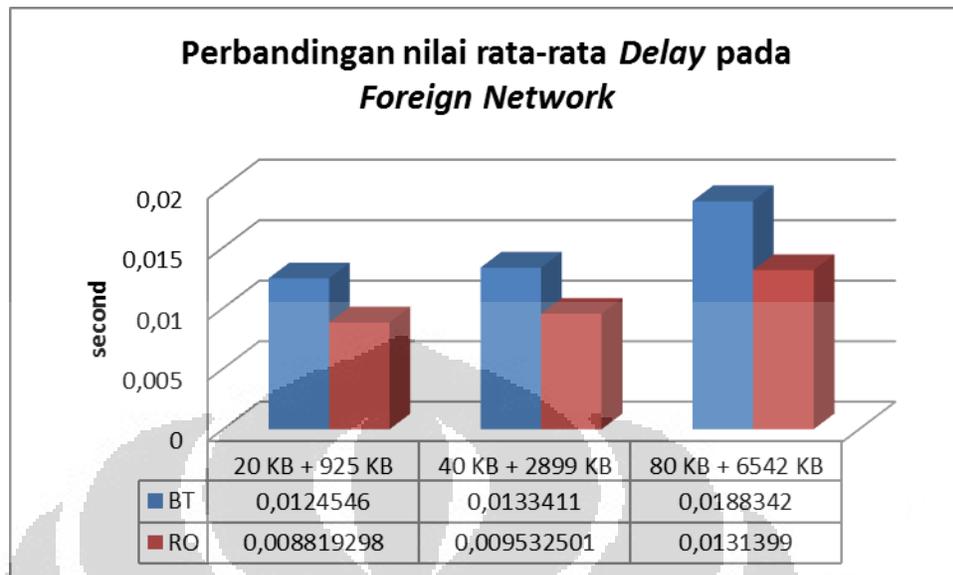
Pada HTML 20 KB + 925 KB rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional tunneling* sebesar 2.23% setiap sepuluh data yang diambil, sedangkan saat menggunakan *route optimization* tidak terdapat *packet loss* (0%). *Bidirectional tunneling* memiliki nilai rata-rata persentase *packet loss* 2.23% lebih besar dibandingkan dengan *route optimization*. Pada HTML 40 KB + 2899 KB rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional*

tunneling sebesar 1.96% dan saat menggunakan *route optimization* sebesar 0.14%. *Bidirectional tunneling* memiliki nilai rata-rata persentase *packet loss* 1.85% lebih besar dibandingkan dengan *route optimization*. Pada HTML 80 KB + 6542 KB, rata-rata persentase *packet loss* yang terjadi saat menggunakan *bidirectional tunneling* sebesar 2.17% dan saat menggunakan *route optimization* sebesar 0.98%. *Bidirectional tunneling* memiliki nilai rata-rata persentase *packet loss* 1.19% lebih besar dibandingkan dengan *route optimization*.

Perbedaan yang sangat signifikan disebabkan oleh beberapa faktor di antara lain, proses *binding update* pada *correspondent node* oleh *mobile node* yang tidak terdapat pada *bidirectional tunneling* sehingga untuk mengirim permintaan yang dikirimkan oleh *mobile node* ke *correspondent node* harus melalui *Home Agent* terlebih dahulu. Proses pengiriman paket menggunakan *bidirectional tunneling* dapat menyebabkan terjadinya tabrakan maupun penumpukan paket pada jaringan sehingga paket akan hilang saat sedang dikirim dan akhirnya paket tidak terjadi terkirim. Pada *route optimization* permintaan paket dapat langsung dikirimkan oleh *correspondent node* ke alamat CoA dari *mobile node* sehingga tabrakan maupun penumpukan paket pada jaringan dapat dihindari.

4.5.3 Analisa Perbandingan Delay

Dari pengambilan data yang dilakukan pada skenario 2 dan skenario 4, akan didapatkan hasil data keseluruhan nilai *delay* yang dapat dilihat pada Tabel 4.11 dan Tabel 4.12. Berdasarkan kedua tabel tersebut, kemudian dapat dibuat perbandingan nilai rata-rata *delay* pada skenario 2 dan skenario 4. Perbandingan tersebut dapat dilihat pada Gambar 4.23 berikut ini.

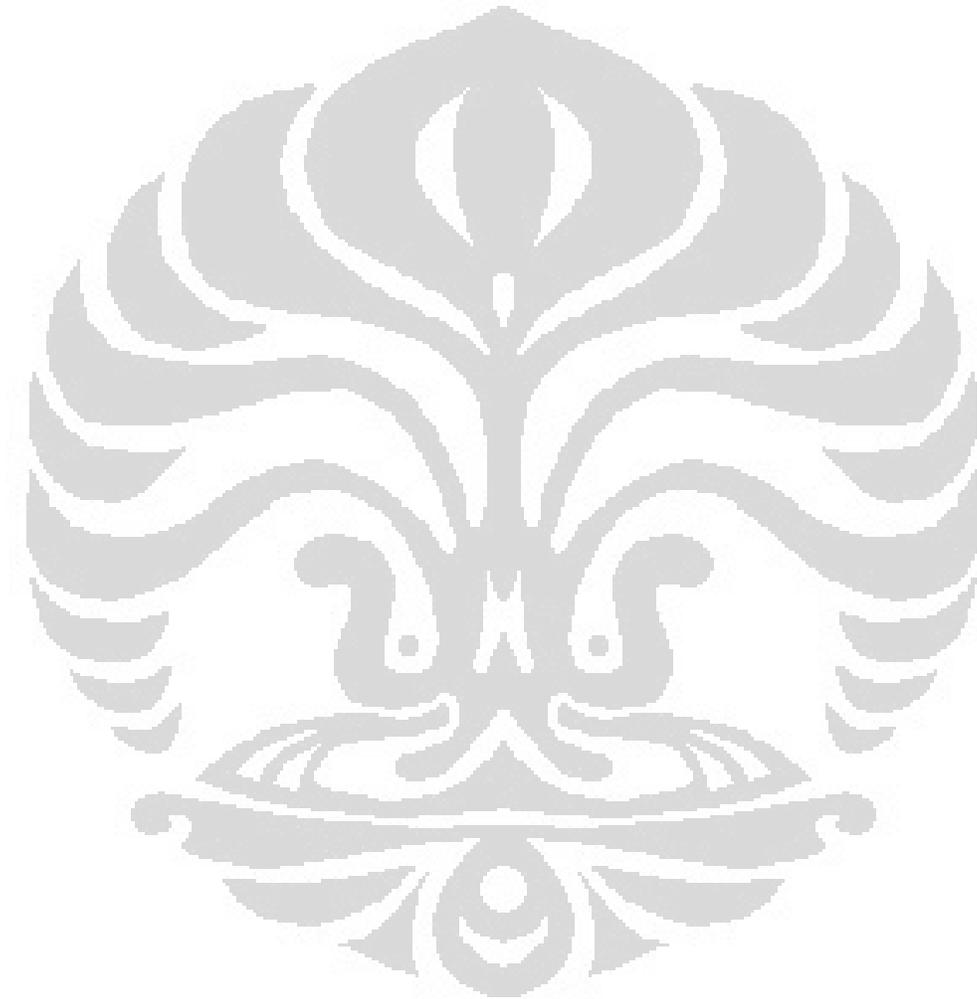


Gambar 4.23 Perbandingan *delay* pada skenario 2 dan skenario 4

Dari data di atas, dapat dilihat perubahan nilai *delay* yang signifikan pada kedua metode tersebut. Pada HTML 20 KB + 925 KB, *delay* ketika menggunakan *bidirectional tunneling* sebesar 0.0124 detik dan saat menggunakan *route optimization* sebesar 0.0088 detik. Terdapat selisih 0.0036 detik (40.90% lebih kecil) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 40 KB + 2899 KB, *delay* saat menggunakan *bidirectional tunneling* sebesar 0.0133 dan saat menggunakan *route optimization* sebesar 0.0095. Terdapat selisih 0.0038 detik (39.95% lebih kecil) antara *route optimization* dengan *bidirectional tunneling*. Pada HTML 80 KB + 6542 KB, *delay* saat menggunakan *bidirectional tunneling* sebesar 0.0188 detik dan saat menggunakan *route optimization* sebesar 0.0131 detik. Terdapat selisih 0.0057 detik (43.51% lebih kecil) antara *route optimization* dengan *bidirectional tunneling*.

Route optimization yang dapat mengirimkan dan menerima paket secara langsung dari *correspondent node* tanpa harus melewati *Home Agent* terlebih dahulu dapat menyebabkan perbedaan *delay* yang signifikan pada skenario 2 dan skenario 4, di mana perbedaan nilai rata-rata *delay* mencapai 40%. Dibandingkan dengan *bidirectional tunneling*, di mana *routing* paket harus melewati *Home Agent* terlebih dahulu dan menyebabkan pada penambahan *hop*. *Routing* pada *route optimization*

memungkinkan *routing* paket secara langsung ke alamat CoA dari *mobile node* menggunakan jalur komunikasi terpendek, sehingga kemungkinan terjadinya *collision* (tabrakan) dan *congestion* (penumpukan) paket pada jaringan dapat dihindari. Penggunaan jalur komunikasi terpendek dapat menyebabkan waktu pengiriman paket (*transfer time*) yang singkat dan berakibat pada *delay* saat proses pengiriman paket pada jalur komunikasi.



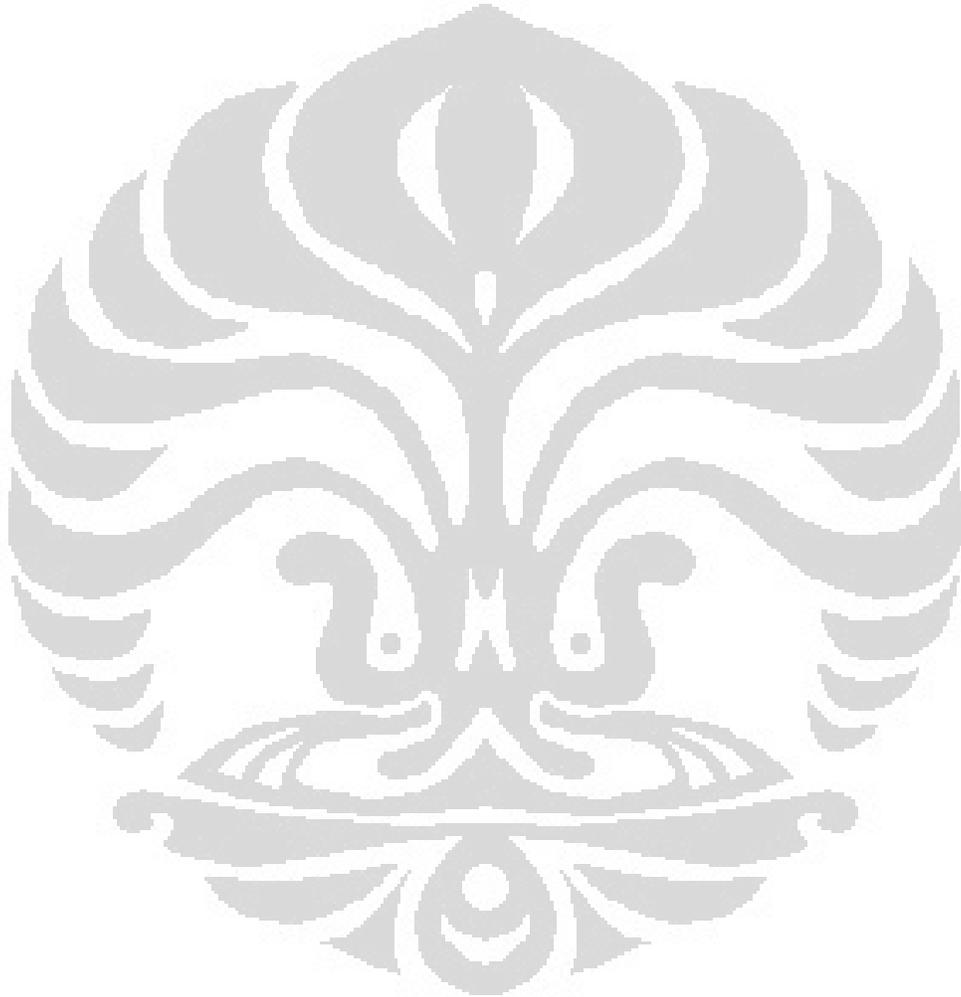
BAB 5

KESIMPULAN

Setelah melakukan pengukuran dan analisis jaringan *Mobile IPv6*, dapat disimpulkan beberapa poin sebagai berikut :

1. Secara umum pada *home network* menggunakan *route optimization* pada proses *vertical handover* memiliki performansi jaringan yang hampir sama dibandingkan *bidirectional tunneling*. Hal ini dapat dilihat dari persentase perbandingan parameter yang di uji, di mana pada pengukuran *throughput* memiliki perbandingan 15.7%, pada pengukuran *packet loss* memiliki perbandingan 0.007%, dan pada pengukuran *delay* memiliki perbandingan 1.77%.
2. Pada *foreign network* menggunakan *bidirectional tunneling* dan *route optimization*, menunjukkan performansi jaringan yang berbeda. Perbandingan performansi jaringan terlihat dari pengukuran parameter-parameter QoS.
3. Hasil pengukuran pada *foreign network* menggunakan *bidirectional tunneling* dan *route optimization* menunjukkan bahwa nilai *throughput* pada *route optimization* mengalami kenaikan sebesar 36.77% dibandingkan dengan *bidirectional tunneling*.
4. Hasil pengukuran pada *foreign network* menggunakan *bidirectional tunneling* dan *route optimization* menunjukkan bahwa nilai *packet loss* pada *route optimization* mengalami penurunan sebesar 1.76% dibandingkan dengan *bidirectional tunneling*.
5. Hasil pengukuran pada *foreign network* menggunakan *bidirectional tunneling* dan *route optimization* menunjukkan bahwa nilai *delay* pada *route optimization* mengalami kenaikan sebesar 41.45% dibandingkan dengan *bidirectional tunneling*.
6. *Binding update* oleh *mobile node* ke *correspondent node* mempengaruhi performansi jaringan. *Route Optimization* yang mendaftarkan *binding update* pada *correspondent node* memiliki performansi jaringan yang lebih baik dibandingkan

dengan *bidirectional tunneling* yang tidak mendaftarkan *binding update*. Hal ini dapat dilihat dari hasil perbandingan *throughput*, *packet loss*, dan *delay* pada butir 1, 3, 4, dan 5.



DAFTAR ACUAN

- [1] Pawan, K. Bhardwaj. 2007. *A+, Network+, Security+ Exams in a Nutshell*. O'Reilly Media
- [2] Bird, Drew, Mike Hardwood. 2002. *Network+ Training Guide*. Que Publishing
- [3] Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. Yogyakarta: Amikom
- [4] Clark, Martin. 2003. *Data Networks, IP, and the Internet*. John Wiley and Sons
- [5] Chohan, M. Arif. Dkk. 2007. *Analisa dan Perancangan Jaringan Mobile IPv6 pada PT. Indosat Mega Media dan BPPT*. Jakarta: Universitas Bina Nusantara
- [6] Fuad. 2008. *Implementasi Mobile IPv6 di PENS ITS (Studi kasus: Pada Home Agent - Foreign Agent)*. Surabaya: ITS
- [7] Heriyanto, Fikri. *Perbandingan Internet Protokol Versi 4 dan Versi 6*. Palembang: Universitas Sriwijaya
- [8] Holger, Zuleger (2005, Desember). *Mobile Internet Protocol v6 (MIPv6)*. HZNet
- [9] *IPv6 Extension Headers Review and Considerations*, Cisco White Paper, Oktober 2006.
- [10] *IPv6 Transition Technologies*. Microsoft Corporation, 2003.
- [11] S. Deering, R. Hinden (1998, Desember). "Internet Protocol, Version 6 (IPv6) Specification". IETF RFC 2460. Internet Engineering Task Force

LAMPIRAN A

Tabel A.1 Hasil Pengukuran *Throughput* pada *Home Network*

Data	Home Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Throughput (Mbit/sec)					
1	0,989	0,802	0,591	1,000868	0,811624	0,604002
2	1,099	0,789	0,547	0,809822	0,898128	0,553564
3	0,98	0,807	0,366	1,216512	0,818298	0,550672
4	1,038	0,795	0,542	1,059798	0,811695	0,374418
5	0,401	0,805	0,327	1,031234	0,721968	0,329289
6	1,188	0,891	0,548	1,00352	0,796101	0,557864
7	0,979	0,836	0,545	1,000538	0,849376	0,55372
8	1,061	0,8	0,523	1,084342	0,8096	0,529276
9	1,065	0,837	0,309	1,08417	0,855414	0,521676
10	1,018	0,712	0,516	1,111089	0,81466	0,312708
Rata-rata	0,9818	0,8074	0,4814	1,0401893	0,8186864	0,4887189

Tabel A.2 Hasil Pengukuran *Throughput* pada *Foreign Network*

Data	Foreign Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Throughput (Mbit/sec)					
1	1,13	0,797	0,494	0,8382	0,9198	0,9386
2	1,58	0,47	0,22	0,8245	0,752	1,17
3	1,163	0,511	0,65	1,582	0,6205	0,5833
4	0,834	0,365	1,063	1,3344	1,368	0,528
5	0,87	1,057	0,645	1,479	1,4798	1,0965
6	0,517	1,002	0,81	0,9306	1,503	1,458
7	0,762	0,912	0,307	1,4076	1,4346	0,936
8	0,782	0,718	0,312	1,896	1,2206	1,4882
9	0,485	0,89	0,825	1,8608	1,424	1,5675
10	0,86	1,012	0,795	1,634	1,4168	1,431
Rata-rata	0,8983	0,7734	0,6121	1,259683333	1,21391	1,11971

Tabel A.3 Hasil Pengukuran *Packet loss* pada *Home Network*

Data	Home Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Packet Loss (packets)					
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	5	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	0	6	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0
Rata-rata	0	0	0,5	0	0,6	0

Tabel A.4 Hasil Pengukuran *Packet loss* pada *Foreign Network*

Data	Foreign Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Packet Loss (packets)					
1	12	10	13	0	0	0
2	15	12	11	0	0	0
3	12	10	14	0	0	0
4	11	11	14	0	0	0
5	13	13	17	0	0	0
6	10	12	18	0	9	0
7	14	20	14	0	0	6
8	14	11	6	0	0	0
9	23	9	15	0	0	0
10	17	14	15	0	0	0
Rata-rata	14,1	12,2	13,7	0	0,9	0,6

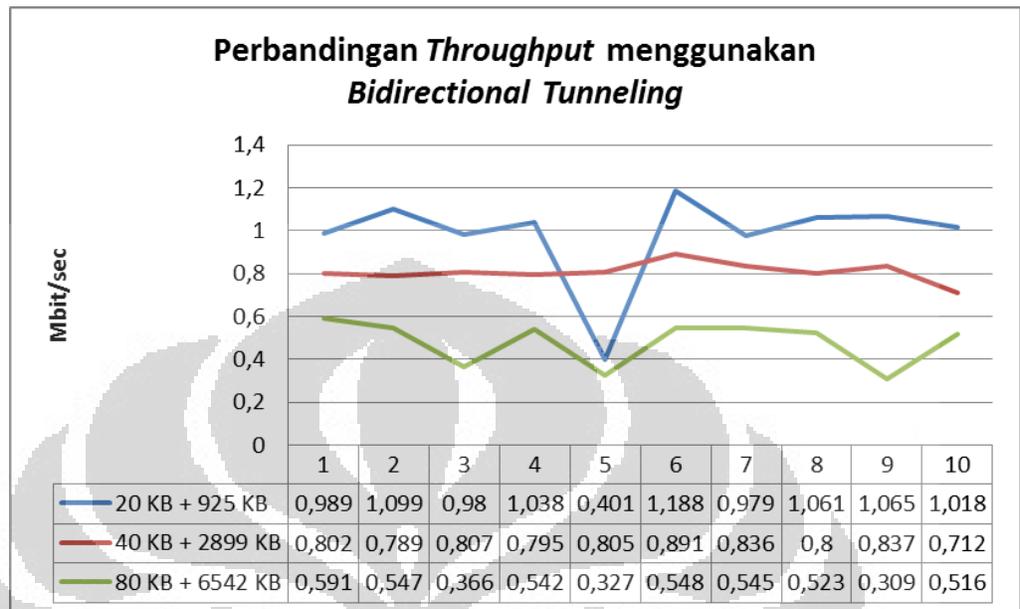
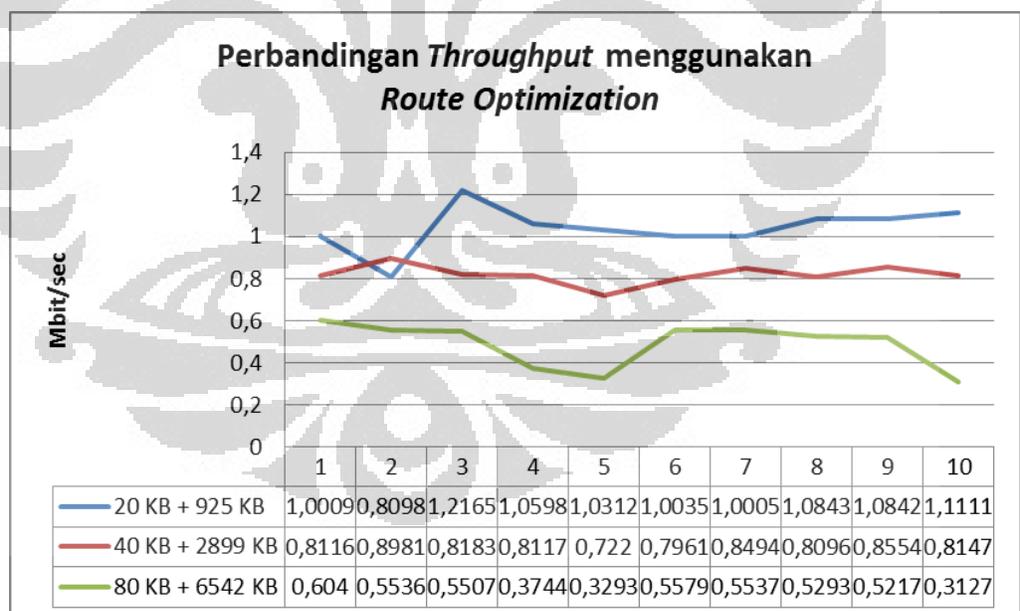
Tabel A.5 Hasil Pengukuran *Delay* pada *Home Network*

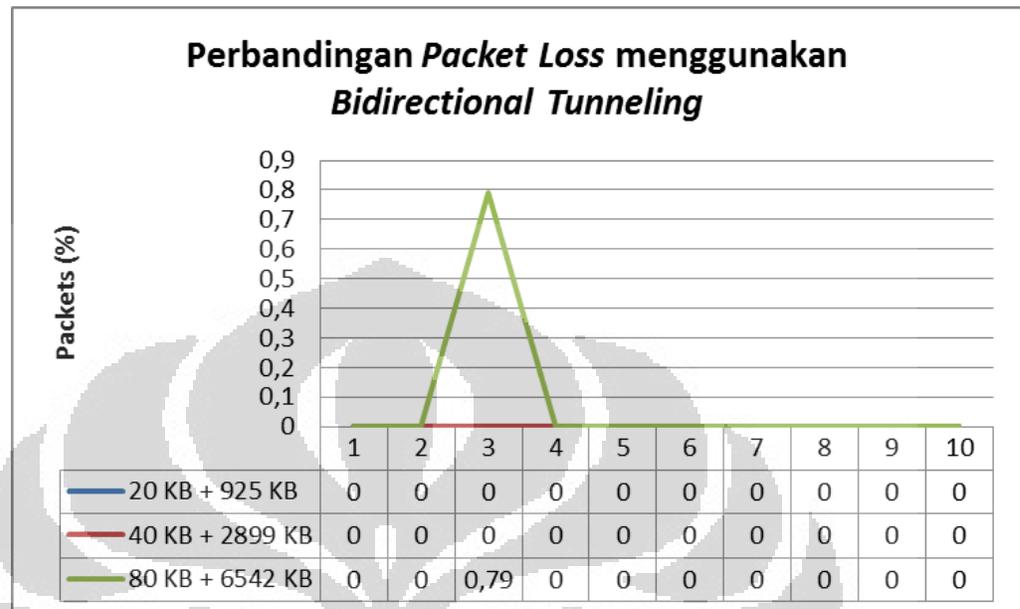
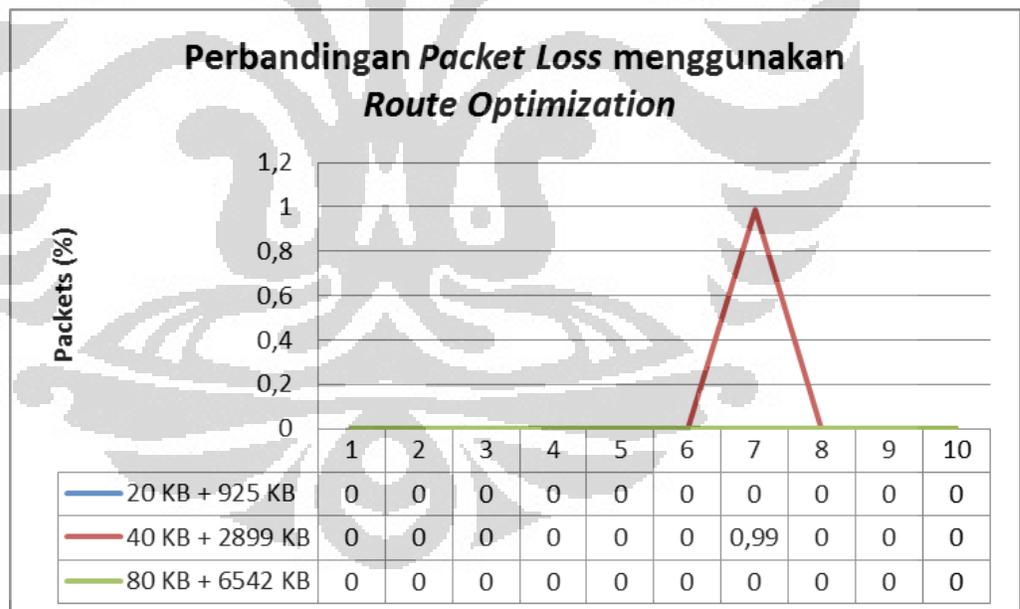
Data	Home Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Delay (second)					
1	0,009584158	0,011804348	0,016069307	0,009603327	0,011851565	0,016422832
2	0,008387097	0,011870968	0,017516484	0,008479355	0,011977806	0,017726681
3	0,009408602	0,011835165	0,026642857	0,009540323	0,012000857	0,027255643
4	0,009186813	0,011623656	0,017692308	0,009379736	0,011832882	0,017975385
5	0,023010526	0,011634409	0,025145161	0,023286653	0,011774022	0,025321177
6	0,008316832	0,01085	0,016978723	0,008416634	0,0109368	0,01728434
7	0,009598039	0,011326733	0,017236559	0,009751608	0,01150796	0,017512344
8	0,008698925	0,011548387	0,017731183	0,008846806	0,011686968	0,017943957
9	0,008691489	0,011417582	0,0268	0,008847936	0,011668769	0,0271216
10	0,009064516	0,013031915	0,018284314	0,009182355	0,013214362	0,018485441
Rata-rata	0,0103947	0,011694316	0,02000969	0,010533473	0,011845199	0,02030494

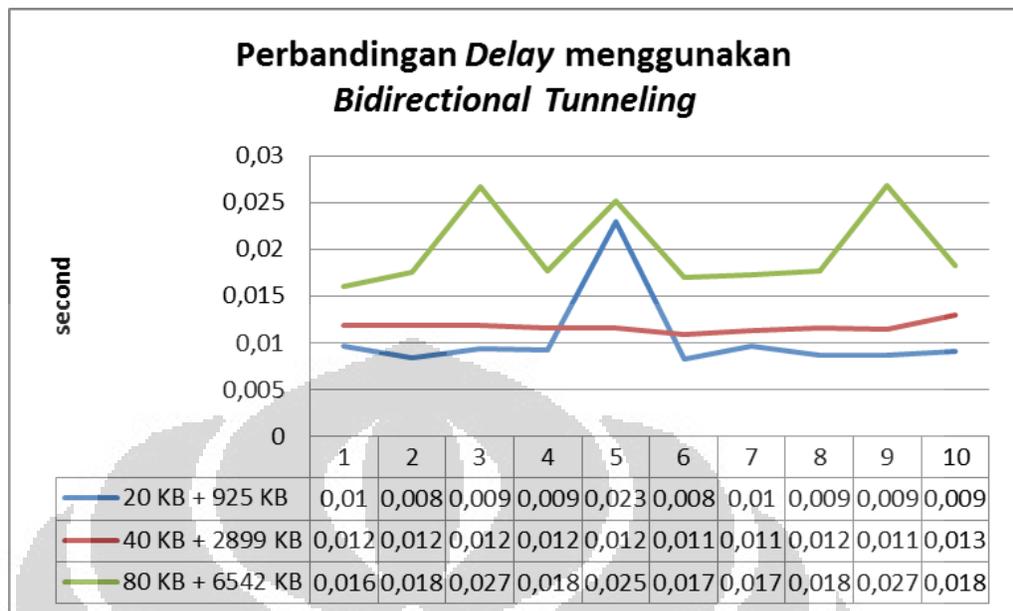
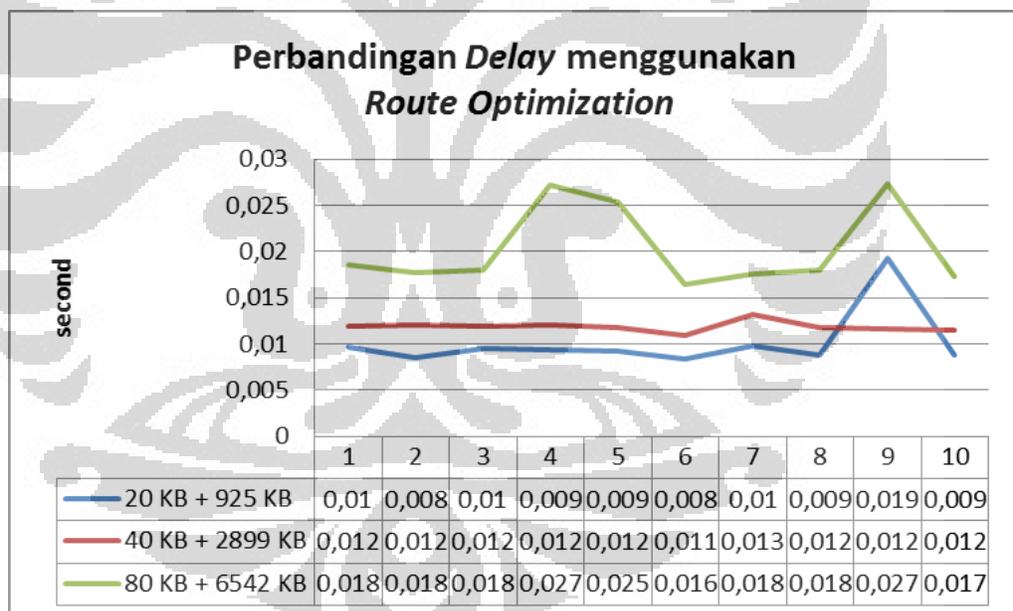
Tabel A.6 Hasil Pengukuran *Delay* pada *Foreign Network*

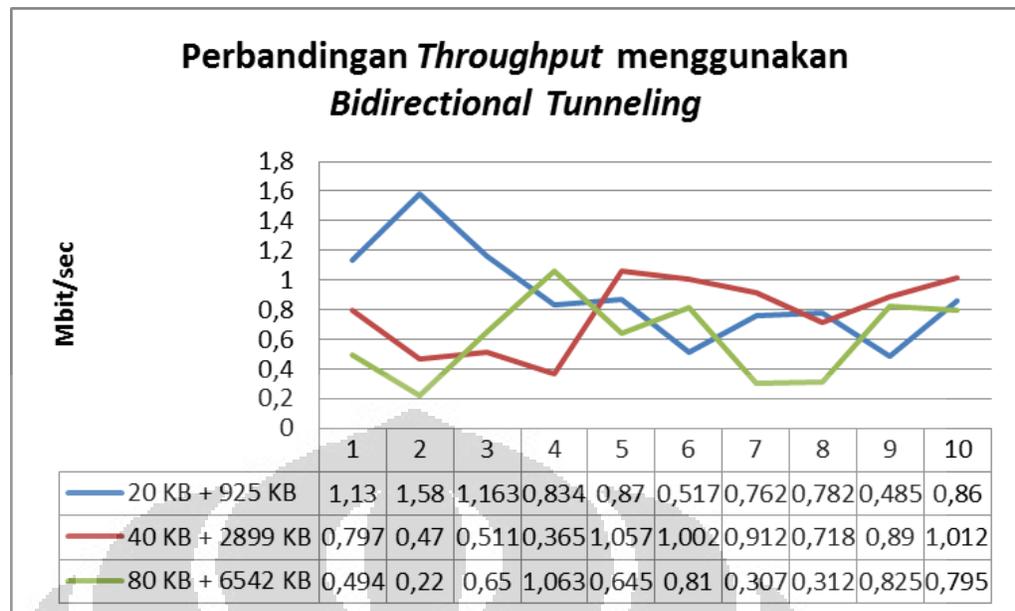
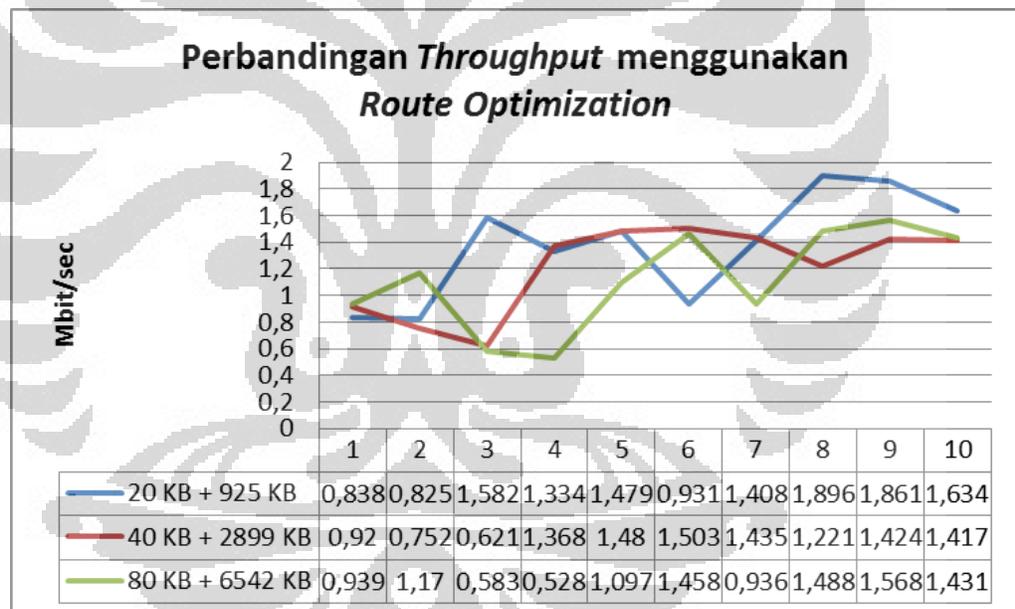
Data	Foreign Network					
	Bidirectional Tunneling			Route Optimization		
	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB	(20 + 925) KB	(40 + 2899) KB	(80 + 6542) KB
	Delay (second)					
1	0,008936937	0,01312766	0,020245614	0,006434595	0,009451915	0,014981754
2	0,006266667	0,017414286	0,037027778	0,004762667	0,012712429	0,025178889
3	0,008639706	0,017	0,015696429	0,005875	0,0119	0,010830536
4	0,011754545	0,020449275	0,010217143	0,008933455	0,015541449	0,007049829
5	0,010742268	0,0094	0,015598214	0,007519588	0,007238	0,011854643
6	0,01927451	0,010264151	0,012418919	0,013106667	0,006876981	0,008817432
7	0,012619048	0,010921739	0,026257143	0,009338095	0,007645217	0,017329714
8	0,012851852	0,013041237	0,025684932	0,008867778	0,008998454	0,017465753
9	0,021414634	0,011849206	0,012538462	0,014561951	0,007701984	0,008776923
10	0,012045455	0,009943262	0,012657143	0,008793182	0,007258582	0,009113143
Rata-rata	0,012454562	0,013341082	0,018834178	0,008819298	0,009532501	0,013139862

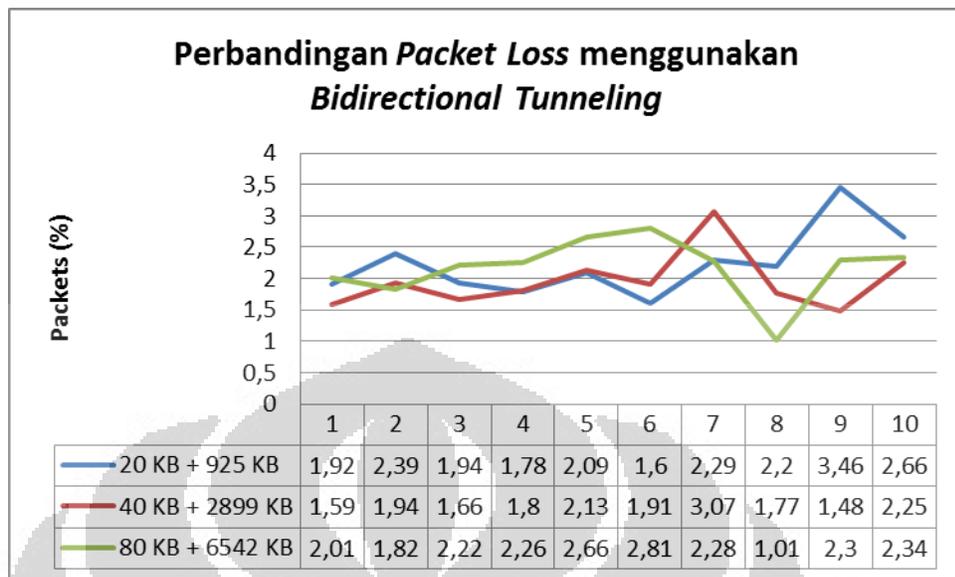
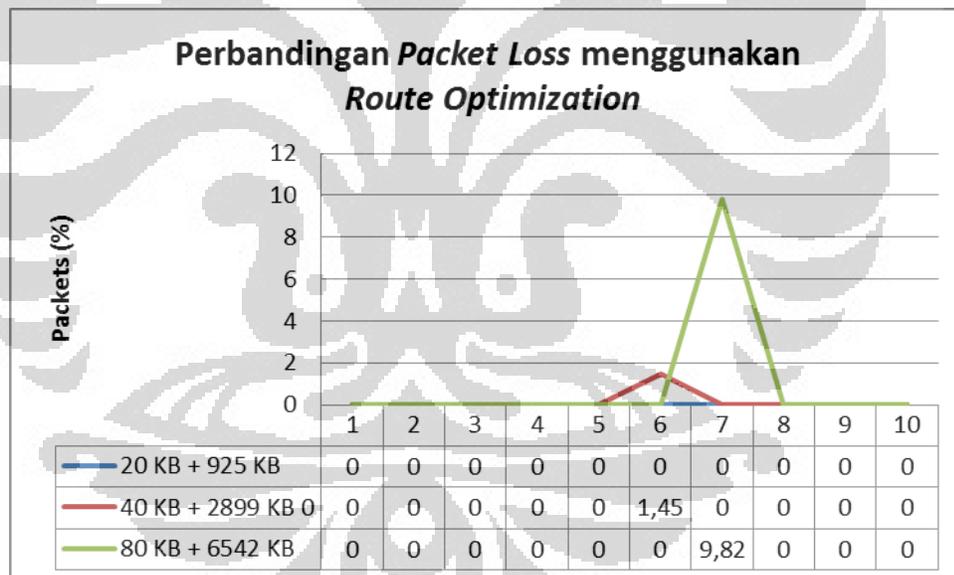
LAMPIRAN B

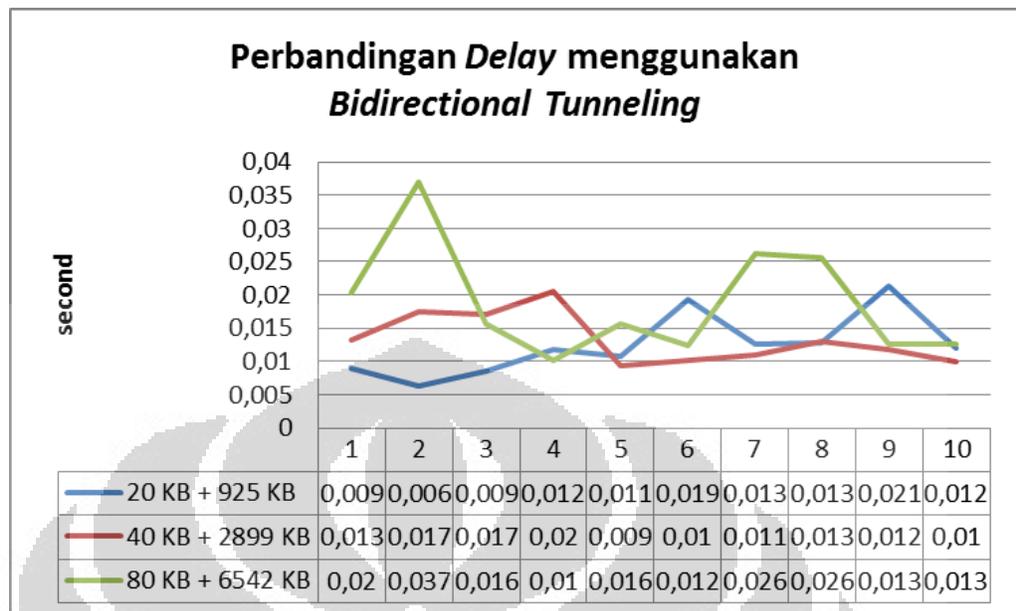
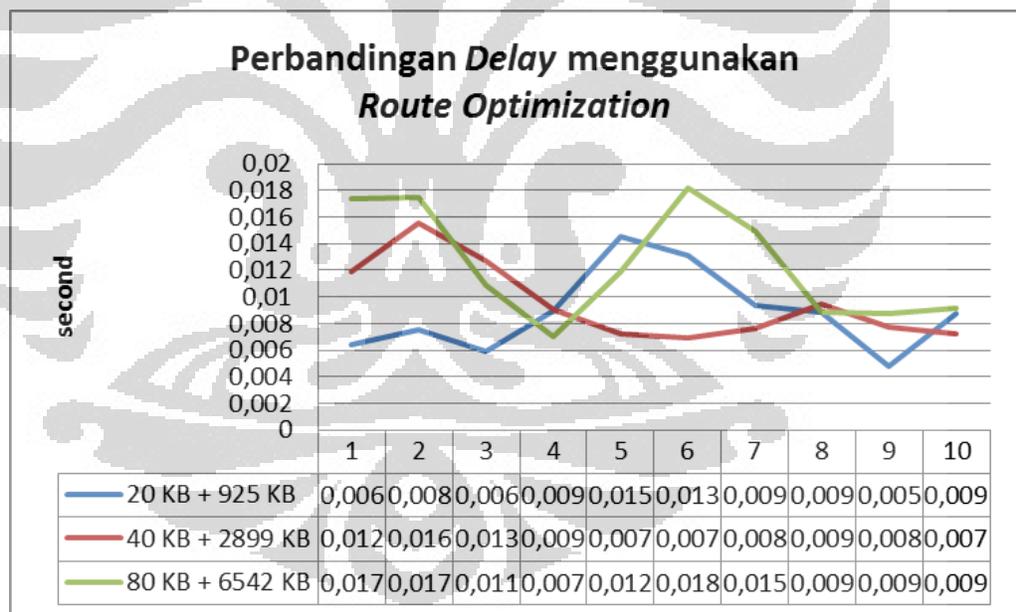
Gambar B.1 Grafik perbandingan *throughput* skenario 1Gambar B.2 Grafik perbandingan *throughput* skenario 3

Gambar B.3 Grafik perbandingan *packet loss* skenario 1Gambar B.4 Grafik perbandingan *packet loss* skenario 3

Gambar B.5 Grafik perbandingan *delay* skenario 1Gambar B.6 Grafik perbandingan *delay* skenario 3

Gambar B.7 Grafik perbandingan *throughput* pada skenario 2Gambar B.8 Grafik perbandingan *throughput* pada skenario 4

Gambar B.9 Grafik perbandingan *packet loss* pada skenario 2Gambar B.10 Grafik perbandingan *packet loss* pada skenario 4

Gambar B.11 Grafik perbandingan *delay* pada skenario 2Gambar B.12 Grafik perbandingan *delay* pada skenario 4