



UNIVERSITAS INDONESIA

**IMPLEMENTASI DAN UNJUK KERJA
HYBRID WIRELESS MESH NETWORK DENGAN
MENGUNAKAN PROTOKOL *ROUTING*
AODV-UU DAN UOBWINAODV**

SKRIPSI

**VEBBY APRILYAN ALHADI
0606042954**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2008**



UNIVERSITAS INDONESIA

**IMPLEMENTASI DAN UNJUK KERJA
HYBRID WIRELESS MESH NETWORK DENGAN
MENGUNAKAN PROTOKOL *ROUTING*
AODV-UU DAN UOBWINAODV**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

**VEBBY APRILYAN ALHADI
0606042954**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
KEKHUSUSAN TELEKOMUNIKASI
DEPOK
DESEMBER 2008**

HALAMAN PERNYATAAN ORISINALITAS

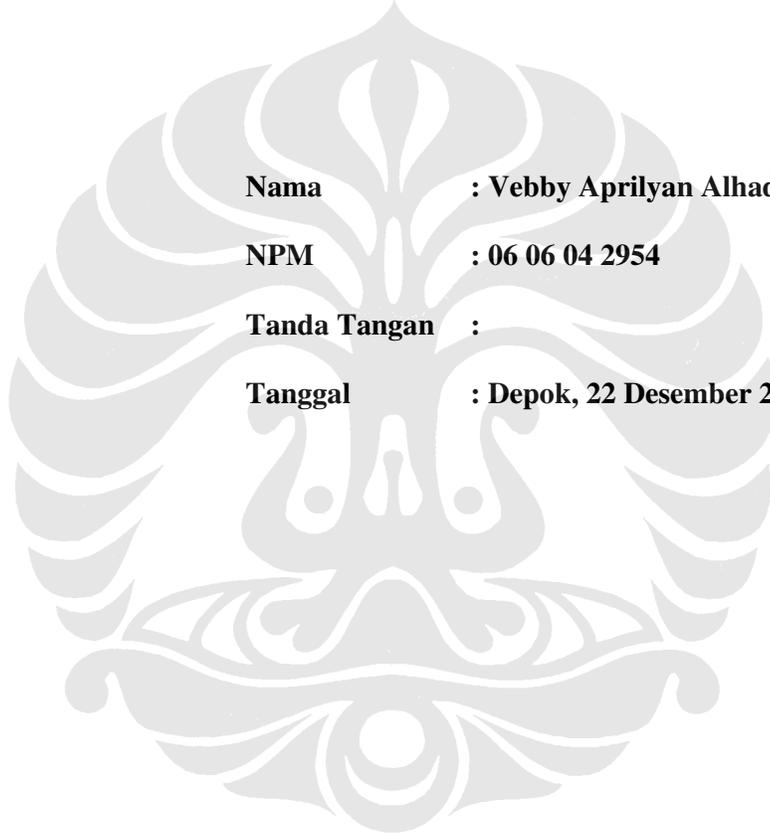
**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Vebby Aprilyan Alhadi

NPM : 06 06 04 2954

Tanda Tangan :

Tanggal : Depok, 22 Desember 2008



HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Vebby Aprilyan Alhadi
NPM : 06 06 04 2954
Program Studi : Teknik Elektro
Judul Skripsi : Implementasi Dan Unjuk Kerja *Hybrid Wireless Mesh Network* Dengan Menggunakan Protokol *Routing AODV-UU* dan *UoBWinAODV*

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

DEWAN PENGUJI

Pembimbing : Muhammad Salman ST, MIT (.....)

Penguji : Dr. Ir. AAP Ratna, M.Eng. (.....)

Penguji : Ir. Endang Sriningsih, MT. (.....)

Ditetapkan di : Depok

Tanggal : 22 Desember 2008

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Muhammad Salman ST., MIT sebagai dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini.
- (2) F. Astha Ekadiyanto ST., Msc yang telah memberikan ide judul serta masukan dalam mengerjakan skripsi ini.
- (3) Kedua orang tua, kakak, dan adik yang telah memberikan bantuan material dan moral.
- (4) Untuk rekan-rekan kelompok, Bagus Mardani dan Ashadi Budiawan yang telah menemani saya selama pengerjaan skripsi dan telah meluangkan banyak waktu dan tenaga dalam pengambilan data.
- (5) Serta semua pihak yang telah membantu dalam pembuatan dan penyusunan skripsi.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 22 Desember 2008

Penulis

Vebby Aprilyan Alhadi

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

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

Nama : Vebby Aprilyan Alhadi
NPM : 06 06 04 2954
Program Studi : Teknik Elektro
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 :

**IMPLEMENTASI DAN UNJUK KERJA
HYBRID WIRELESS MESH NETWORK DENGAN
MENGUNAKAN ROUTING PROTOCOL
AODV-UU DAN UOBWINAODV**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa meminta izin dari 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 : 22 Desember 2008
Yang menyatakan

(Vebby Aprilyan Alhadi)

ABSTRAK

Nama : VEBBY APRILYAN ALHADI
Program Studi : Teknik Elektro
Judul : IMPLEMENTASI DAN UNJUK KERJA *HYBRID WIRELESS MESH NETWORK* DENGAN MENGGUNAKAN *ROUTING PROTOCOL AODV-UU* DAN *UOBWINAODV*

Wireless Mesh Network merupakan teknologi jaringan *wireless* yang dipercaya dapat meningkatkan peranan penting dalam *wireless mobile network* dimasa yang akan datang. Teknologi ini memiliki kemampuan mengkonfigurasi dan mengorganisasi dirinya sendiri, sehingga mampu membuat dan menjaga konektivitasnya serta memiliki jangkauan luas karena menggunakan sistem *multihop*. Dalam penulisan skripsi ini akan dibangun *testbed wireless mesh network* tipe *hybrid* menggunakan perangkat *mesh client* dan *mesh router* dengan *routing protocol AODV-UU* dan *UoBWinAODV*. *Mesh router* dimodifikasi dengan menggunakan *firmware opensource OpenWrt*. *Testbed* tersebut digunakan untuk menguji performansi *self configure*, *self healing* serta parameter-parameter seperti *throughput*, *latency* dan *jitter* melalui beberapa skenario pengujian tertentu.

Kata kunci :
wireless mesh network, *AODV-UU*, *UoBWinAODV*, *OpenWrt*, *multihop*.

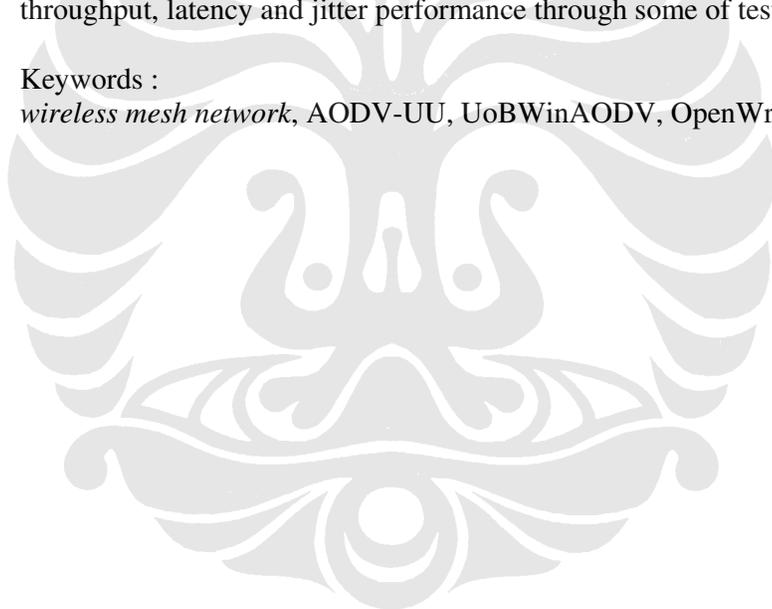
ABSTRACT

Name : VEBBY APRILYAN ALHADI
Study Program: Electrical Engineering
Title : IMPLEMENTATION AND PERFORMANCE SYSTEM
HYBRID WIRELESS MESH NETWORK USING ROUTING
PROTOCOL AODV-UU AND UOBWINAODV

Wireless Mesh Network is a wireless network technology that trusted can increase important role in the future of wireless mobile network. Its has an ability in self configured and self organized, so that can make and maintain the connectivity and also has a large range because its used a multihop system. In this final project will be built a hybrid wireless mesh network testbed using a mesh client and mesh router device with AODV-UU and UoBWinAODV routing protocol. The mesh router device is modified using opensource firmware OpenWrt. The testbed will be used to test a self configure, self healing and also a network parameters such as throughput, latency and jitter performance through some of testing skenario.

Keywords :

wireless mesh network, AODV-UU, UoBWinAODV, OpenWrt, multihop.



DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
UCAPAN TERIMA KASIH	iii
LEMBAR PERNYATAAN ORISINALITAS	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	v
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR SINGKATAN	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 TUJUAN PENELITIAN	2
1.3 PERUMUSAN MASALAH	2
1.4 BATASAN MASALAH	2
1.5 METODOLOGI PENELITIAN	3
1.6 SISTEMATIKA PENELITIAN	3
BAB 2 LANDASAN TEORI	4
2.1 TEKNOLOGI <i>WIRELESS</i>	4
2.1.1 Keuntungan Dan Keterbatasan Pada Teknologi <i>Wireless</i>	4
2.2 <i>WIRELESS MESH NETWORK</i>	5
2.2.1 Arsitektur <i>Wireless Mesh Network</i>	7
2.2.1.1 <i>Infrastructure Wireless Mesh Network</i>	8
2.2.1.2 <i>Client Wireless Mesh Network</i>	9
2.2.1.3 <i>Hybrid Wireless Mesh Network</i>	10
2.2.2 Karakteristik <i>Wireless Mesh Network</i>	10
2.2.3 Protokol Dalam <i>Wireless Mesh Network</i>	11
2.2.3.1 <i>Physical Layer Protocol</i>	11
2.2.3.2 <i>Medium Access Control (MAC) Layer Protocol</i>	13
2.2.3.3 <i>Network Layer Protocol</i>	13
2.2.3.4 <i>Transport Layer Protocol</i>	14
2.2.3.5 <i>Application Layer Protocol</i>	15
2.2.4 <i>Routing</i> Dalam <i>Wireless Mesh Network</i>	15
2.2.5 Protokol <i>Routing Ad hoc On-demand Distance Vector</i> (<i>AODV</i>).....	15
2.2.5.1 <i>Route Request (RREQ)</i>	17
2.2.5.2 <i>Route Reply (RREP)</i>	18
2.2.5.3 <i>Pesan HELLO</i>	19
2.2.5.4 <i>Route Error (RERR)</i>	20
2.2.5.5 <i>Informasi Yang Terdapat Pada Tabel Routing</i>	20
2.3 <i>AD HOC ON-DEMAND DISTANCE VECTOR – UPPSALA</i> <i>UNIVERSITY (AODV-UU)</i>	21

2.3.1 Konfigurasi	22
2.3.2 Interaksi Dengan Internet Protokol	23
2.3.3 Netfilter <i>Framework</i>	23
2.3.4 Penanganan Paket	25
2.3.5 Kedatangan Paket	25
2.3.6 Pemrosesan Paket	26
2.3.7 Pemrosesan Paket Data	26
2.3.8 Pemrosesan AODV <i>Control Message</i>	26
2.3.9 Pengiriman AODV <i>Control Message</i>	27
2.4 OPENWRT	27
2.4.1 WhiteRussian	28
2.4.2 Kamikaze	28
2.5 UoBWinAODV.....	29
BAB 3 PERANCANGAN JARINGAN	30
3.1 PERENCANAAN	30
3.2 PEMBANGUNAN <i>TESTBED</i>	31
3.2.1 Penentuan Topologi Jaringan	31
3.2.2 Menentukan Spesifikasi Perangkat Yang Digunakan.....	32
3.2.2.1 Perangkat <i>Mesh Router</i>	32
3.2.2.2 Perangkat <i>Mesh Client</i>	34
3.2.3 Instalasi <i>Software</i>	35
3.2.3.1 Instalasi OpenWrt	35
3.2.3.2 Instalasi Paket AODV-UU	36
3.2.3.3 Instalasi UoBWinAODV.....	38
3.2.4 Penentuan Lokasi <i>Testbed</i>	39
3.2.5 Konfigurasi Jaringan.....	41
3.2.5.1 Pengalamatan	41
3.2.5.2 Konfigurasi Parameter <i>Wireless</i> Pada <i>Mesh Router</i> ...	43
3.2.5.3 <i>Packet Forwarding Rules</i> Pada <i>Mesh Router</i>	43
3.2.5.4 Konfigurasi Pada UoBWinAODV	44
3.2.5.5 Menjalankan AODV-UU Dan UoBWinAODV.....	45
BAB 4 PENGUJIAN DAN ANALISA.....	47
4.1 HASIL RANCANGAN <i>TESTBED</i>	47
4.2 PENGUJIAN <i>TESTBED</i> JARINGAN.....	48
4.2.1 Pengujian <i>Self-configure</i>	48
4.2.2 Pengujian <i>self-healing</i>	50
4.2.3 Pengujian Pemakaian <i>Bandwidth</i>	51
4.2.4 Pengujian Sistem <i>Multihop</i>	52
4.3 ANALISA SISTEM.....	53
4.3.1 Analisa Performansi <i>Self Configure</i>	53
4.3.2 Analisa Performansi <i>Self Healing</i>	54
4.3.3 Analisa Performansi Pemakaian <i>Bandwidth</i>	56
4.3.3.1 Pemakaian <i>Bandwidth In</i>	56
4.3.3.2 Pemakaian <i>Bandwidth Out</i>	57
4.3.3.3 Pemakaian <i>Bandwidth Total</i>	58
4.3.4 Analisis Performansi <i>Latency</i>	59

4.3.5 Analisa Performansi <i>End-to-End Throughput</i>	61
4.3.6 Analisa Performansi <i>Jitter</i>	63
BAB 5 KESIMPULAN	67
DAFTAR ACUAN	69
DAFTAR PUSTAKA	71
LAMPIRAN	72



DAFTAR GAMBAR

Gambar 2.1.	Struktur jaringan <i>mesh</i> dengan tiga buah <i>node</i> , empat buah <i>node</i> , dan lima buah <i>node</i> dimana setiap <i>node</i> saling terhubung satu sama lain	5
Gambar 2.2.	<i>Node</i> pada <i>wireless mesh network</i> terdiri atas <i>mesh routers</i> dan <i>mesh clients</i>	7
Gambar 2.3.	Arsitektur <i>infrastructure wireless mesh network</i>	9
Gambar 2.4.	Arsitektur <i>client wireless mesh network</i>	9
Gambar 2.5.	Arsitektur <i>hybrid wireless mesh network</i>	10
Gambar 2.6.	Perbandingan antara FDM dan OFDM.....	12
Gambar 2.7.	CSMA/CA dengan paket kontrol RTS/CTS	13
Gambar 2.8.	Klasifikasi protokol <i>routing</i> dalam <i>wireless mesh network</i> ...	14
Gambar 2.9.	Pencarian rute AODV: disebelah kiri adalah rute permintaan (<i>route request</i>) dan disebelah kanan adalah rute balasan (<i>route reply</i>).....	16
Gambar 2.10.	Format paket pesan <i>route request</i> (RREQ).....	18
Gambar 2.11.	Format paket pesan <i>route reply</i> (RREP).....	19
Gambar 2.12.	Format paket pesan <i>HELLO</i>	19
Gambar 2.13.	Format paket pesan <i>RREQ</i>	20
Gambar 2.14.	Netfilter <i>hooks</i> untuk IP. Pengiriman paket pada <i>hook</i> ini dapat diterima dan dimodifikasi dengan kode <i>segment</i> yang telah disesuaikan (modul <i>kernel</i>).....	24
Gambar 2.15.	Penanganan paket pada AODV-UU. Paket data dan pesan kontrol AODV ditangani secara terpisah	25
Gambar 3.1.	Diagram alir tahapan pembangunan <i>testbed</i>	30
Gambar 3.2.	Perencanaan topologi jaringan.....	32
Gambar 3.3.	<i>Wireless router</i> linksys WRT54GL versi 1.1 dan perlengkapannya	33
Gambar 3.4.	Tampak depan <i>wireless router</i> WRT54GL versi 1.1	33
Gambar 3.5.	Tampak belakang <i>wireless router</i> WRT54GL versi 1.1	34
Gambar 3.6.	Laptop sebagai perangkat <i>client user</i>	34
Gambar 3.7.	Jendela <i>firmware upgrade</i> dari linksys WRT54GL versi 1.1.....	35
Gambar 3.8.	Halaman depan <i>web interface</i> OpenWrt.....	36
Gambar 3.9.	Halaman depan <i>command line</i> OpenWrt	36
Gambar 3.10.	Tampilan program WinSCP.....	37
Gambar 3.11.	<i>Passthru driver</i> pada <i>wireless network connections</i>	39
Gambar 3.12.	Denah lokasi penempatan	40
Gambar 3.13.	Lokasi penempatan <i>mesh router</i> dan <i>mesh client</i>	41
Gambar 3.14.	Pengalamatan pada <i>mesh client</i>	42
Gambar 3.15.	<i>Interface</i> yang aktif pada <i>mesh client</i>	44
Gambar 3.16.	Menjalankan <i>wireless interface</i> pada mode <i>ad-hoc</i>	45
Gambar 3.17.	Jendela <i>wireless network connection</i>	46
Gambar 4.1.	<i>Testbed</i> yang dibangun untuk <i>wireless mesh network</i>	48
Gambar 4.2.	Skenario pengujian <i>self-configure</i>	49
Gambar 4.3.	Skenario pengujian <i>self-healing</i>	50

Gambar 4.4.	Skenario pengujian pemakaian <i>bandwidth</i>	51
Gambar 4.5.	Skenario pengujian kinerja sistem <i>multihop</i>	52
Gambar 4.6.	Grafik perbandingan performansi <i>self configure</i>	54
Gambar 4.7.	Grafik perbandingan performansi <i>self configure</i>	55
Gambar 4.8.	Grafik perbandingan pemakaian <i>bandwidth in</i>	56
Gambar 4.9.	Grafik perbandingan pemakaian <i>bandwidth out</i>	57
Gambar 4.10.	Grafik perbandingan pemakaian <i>bandwidth total</i>	58
Gambar 4.11.	Perbandingan <i>latency</i> pada setiap percobaan.....	60
Gambar 4.12.	Perbandingan besarnya <i>latency</i> dari setiap <i>node</i>	60
Gambar 4.13.	Grafik perbandingan <i>throughput</i> pada sistem <i>multihop</i>	61
Gambar 4.14.	Grafik perbandingan <i>jitter</i> pada sistem <i>multihop</i>	64



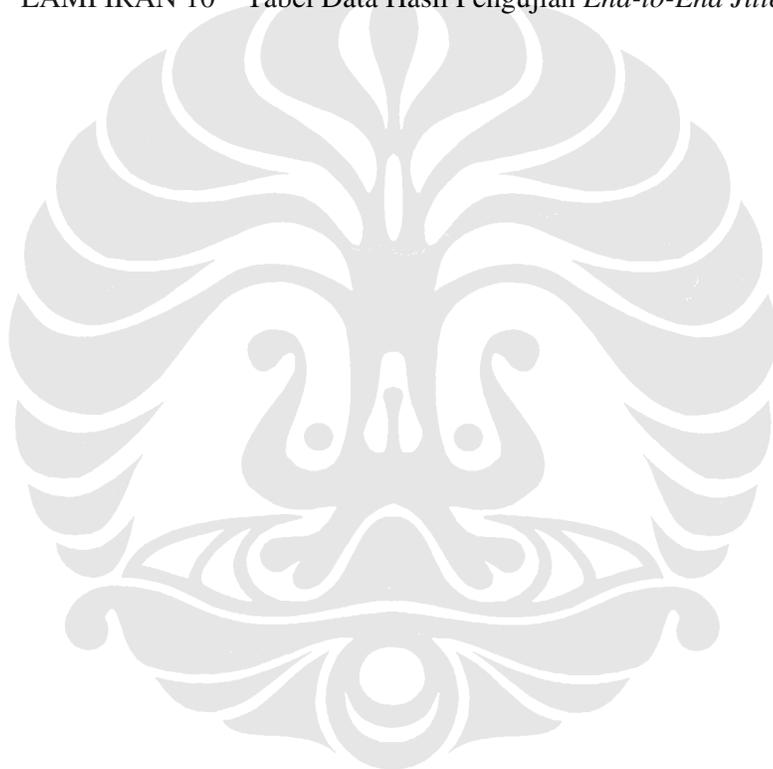
DAFTAR SINGKATAN

ACK	Acknowledgement
AODV	Ad hoc On-demand Distance Vector
AODV-UU	Ad hoc On-demand Distance Vector – Uppsala University
AP	Access Point
CCK	Complementary Code Keying
CDMA	Code Division Multiple Access
CP	Cyclic Prefix
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
DFC	Distributed Coordination Function
DHCP	Dynamic Host Configuration Protocol
DIFS	Distributed Inter Frame Space
DMZ	Demilitarized Zone
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FHSS	Frequency Hopping Spread Spectrum
GPL	General Public License
GUI	Graphic User Interface
ICI	Inter Carrier Interference
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISI	Inter Symbol Interference
ITU	International Telecommunication Union
LAN	Local Area Network
LED	Light Emitted Diode
MAC	Medium Access Control
Mad-hoc	Multicast Ad-hoc
MANET	Mobile Ad hoc Network
MCM	Multi Carrier Modulation
MD5	Message Digest 5
MIMO	Multiple Input Multiple Output
NAV	Network Allocation Vector
NLOS	Non Line of Sight
NVRAM	Non Volatile Random Access Memory
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing
OSI	Open System Interconnection
PDA	Personal Digital Assistant
PHY	Physical Layer

PSK	Phase Shift Keying
RAM	Random Access Memory
RERR	Route Error
RFC	Request For Comment
RP-TNC	Reverse Polarity-Threaded Neill Concelman
RREP	Route Reply
RREQ	Route Request
RTS/CTS	Ready to Send / Clear to Send
SIFS	Short Inter Frame Space
SNR	Signal to Noise Ratio
SSH	Secure Shell
SSID	Service Set Identifier
TCP	Transmission Control Protocol
TCP / IP	Transmission Control Protocol / Internet Protocol
TELNET	Telecommunication Network
TTL	Time To Live
UDP	User Datagram Protocol
UWB	Ultra Wideband
VCS	Virtual Carrier Sense
VLAN	Virtual Local Area Network
WAN	Wide Area Network
WECA	Wireless Ethernet Compatibility
WiFi	Wireless Fidelity
WiMAX	World-wide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WMN	Wireless Mesh Network
ZRP	Zone Routing Protocol

DAFTAR LAMPIRAN

LAMPIRAN 1	Spesifikasi <i>Wireless Router</i> WRT54GL Versi 1.1.....	71
LAMPIRAN 2	Tabel Hasil Pengujian <i>Self Configure</i> dan <i>Self Healing</i> ...	72
LAMPIRAN 3	Grafik Hasil Pengujian Pemakaian <i>Bandwidth</i>	73
LAMPIRAN 4	Grafik Hasil Pengamatan <i>Latency</i>	81
LAMPIRAN 5	Grafik Hasil Pengujian <i>Throughput</i> pada sistem <i>multihop</i>	89
LAMPIRAN 6	Grafik Hasil Pengujian <i>Jitter</i> pada sistem <i>multihop</i>	90
LAMPIRAN 7	Tabel Data Hasil Pengujian Pemakaian <i>Bandwidth</i>	91
LAMPIRAN 8	Tabel Data Hasil Pengujian <i>Latency</i>	97
LAMPIRAN 9	Tabel Data Hasil Pengujian <i>End-to-End Throughput</i> ...	99
LAMPIRAN 10	Tabel Data Hasil Pengujian <i>End-to-End Jitter</i>	100



BAB 1

PENDAHULUAN

1.1 LATAR BELAKANG

Kemajuan teknologi dibidang telekomunikasi dan komunikasi data berkembang dengan pesat diseluruh dunia. Salah satunya adalah dengan berkembangnya teknologi jaringan *wireless* yang telah banyak digunakan dalam berbagai macam aplikasi. Dalam perkembangannya jaringan *wireless* semakin dituntut untuk memberikan kualitas layanan yang lebih baik dalam area lokal maupun area yang luas dan dapat memberikan mobilitas yang tinggi, tingkat realibilitas yang tinggi, serta penghematan waktu serta biaya instalasi.

Wireless Mesh Network (WMN) merupakan teknologi jaringan *wireless* yang dipercaya dapat meningkatkan peranan penting dalam *wireless mobile network* di masa yang akan datang. *Wireless mesh network* memiliki kemampuan dalam mengkonfigurasi dan mengorganisasi dirinya sendiri (*self-configured/self-organized*), sehingga mampu membuat dan menjaga konektivitasnya apabila terjadi kerusakan pada salah satu *node*. Kemampuan ini selain membantu para pengguna untuk dapat selalu terhubung kapan saja dan dimana saja, juga akan membawa keuntungan lain seperti biaya pembuatan yang rendah, kemudahan dalam perawatan jaringan, tingkat *robustness* serta reliabilitas tinggi. *Wireless mesh network* juga memiliki jangkauan yang luas karena menggunakan sistem *multi-hop*.

Pada penelitian ini akan dibangun *testbed wireless mesh network* dengan menggunakan arsitektur tipe *hybrid wireless mesh network*. Pada tipe arsitektur ini *wireless mesh network* dibangun dari beberapa *mesh router* dan *mesh client*, dimana *mesh client* dapat terhubung pada jaringan melalui *mesh router* sekaligus tetap berhubungan langsung dengan *mesh client* lainnya. Untuk memudahkan dalam memodifikasi dan agar dapat bekerja sesuai dengan spesifikasi *wireless mesh network*, pada *wireless router* digunakan sebuah perangkat lunak *opensource* OpenWrt sebagai sistem operasinya. Untuk memenuhi konsep *multi-hop* pada *mesh router* digunakan protokol *routing AODV-UU* (*Ad hoc On-demand Distance Vector-Uppsala University*), dan pada *mesh client* digunakan

UoBWinAODV yang merupakan implementasi protokol *routing* AODV untuk microsoft windows XP.

1.2 TUJUAN PENELITIAN

Tujuan penelitian pada tugas akhir ini adalah sebagai berikut :

- 1) merancang dan membangun sebuah *testbed wireless mesh network* dengan tipe arsitektur *hybrid wireless mesh network* yang dibentuk dari beberapa perangkat *wireless router* sebagai *mesh router* dan laptop sebagai *mesh client*,
- 2) menerapkan AODV-UU pada *mesh router* sebagai protokol *routing* yang akan digunakan dan UoBWinAODV pada *mesh client* sebagai *protocol handler*.

1.3 PERUMUSAN MASALAH

Permasalahan yang ditangani pada penelitian ini adalah penerapan konsep *wireless mesh network* pada perangkat *wireless router* yang menggunakan OpenWrt sebagai sistem operasinya dan AODV-UU sebagai protokol *routing*-nya, serta pada *mesh client* yang menggunakan UoBWinAODV sebagai penanganan protokolnya.

1.4 BATASAN MASALAH

Penelitian ini mempunyai batasan-batasan masalah sebagai berikut :

- 1) arsitektur *wireless mesh network* yang akan dibangun adalah tipe *hybrid wireless mesh network*,
- 2) sistem operasi pada perangkat *wireless router* yang digunakan adalah OpenWrt dengan versi WhiteRussian 0.9,
- 3) protokol *routing* yang digunakan pada perangkat *wireless router* adalah AODV-UU versi 0.9.1-1 yang dikembangkan oleh Universitas Uppsala, Swedia,
- 4) penanganan protokol pada *mesh client* digunakan UoBWinAODV versi 0.15 yang dikembangkan oleh Universitas Bremen.

1.5 METODOLOGI PENELITIAN

Langkah-langkah yang ditempuh dalam penelitian ini dimulai dari studi literatur mengenai *wireless mesh network*, penggunaan *firmware* OpenWrt sebagai sistem operasinya, penerapan konsep *multi-hop* pada *wireless mesh network* dengan AODV-UU sebagai protokol *routing* pada *mesh router* dan UoBWinAODV pada *mesh client*. Perancangan yang dilakukan adalah membuat *testbed hybrid wireless mesh network* dengan menggunakan perangkat *wireless router* yang memiliki sistem operasi OpenWrt dan AODV-UU sebagai protokol *routing*-nya dan perangkat *mesh client* (laptop) yang menggunakan UoBWinAODV sebagai penanganan protokolnya.

1.6 SISTEMATIKA PENELITIAN

Penulisan penelitian ini disajikan dalam 5 (lima) bab, yang dapat dilihat sebagai berikut ini.

1. **Bab 1 Pendahuluan**, berisi penjelasan mengenai latar belakang, tujuan penelitian, perumusan masalah, batasan masalah, metodologi penelitian, dan sistematika penelitian yang digunakan untuk menyusun laporan penelitian tugas akhir ini.
2. **Bab 2 Landasan Teori**, berisi dasar teori yang menunjang dalam penelitian tugas akhir ini. Teori dasar yang diberikan meliputi teknologi *wireless*, *wireless mesh network*, protokol *routing* AODV-UU, *firmware* OpenWrt, *protocol handler* UoBWinAODV serta teori lain yang berkaitan dengan penelitian ini.
3. **Bab 3 Perancangan Testbed**, berisi perencanaan dan tahapan-tahapan yang dilakukan dalam perancangan dari sistem *wireless mesh network*. Tahapan-tahapan tersebut meliputi penentuan spesifikasi perangkat, instalasi *software*, penentuan lokasi *testbed*, dan konfigurasi jaringan.
4. **Bab 4 Pengujian dan Analisa**, berisikan pengujian *testbed* yang akan dibangun dan analisa pada setiap pengujian.
5. **Bab 5 Kesimpulan**, berisi kesimpulan yang didapat dari keseluruhan penelitian tugas akhir ini.

BAB 2 LANDASAN TEORI

2.1 TEKNOLOGI WIRELESS

Teknologi jaringan *wireless* terbentang luas mulai dari komunikasi suara sampai dengan jaringan data, yang mana membolehkan pengguna untuk membangun koneksi *wireless* pada jarak tertentu. Ini termasuk teknologi infrared, frekuensi radio, dan lain sebagainya. Perangkat yang umum digunakan untuk jaringan *wireless* termasuk di dalamnya adalah laptop, PDA, telepon selular, dan lain sebagainya. Teknologi *wireless* ini memiliki beberapa kegunaan, misalnya pengguna bergerak dapat menggunakan telepon selular untuk mengakses *e-mail*, pengguna dengan laptopnya biasa terhubung ke internet ketika berada di bandara, kafe, kereta api, dan tempat umum lainnya.

Karena kemampuannya dalam pengiriman data, manusia di seluruh dunia menggunakannya dalam berbagai aplikasi yang berkaitan dengan jaringan *wireless* dimana salah satunya adalah penggunaan dalam komunikasi data. Untuk mengatasi perkembangan penggunaan komunikasi data dengan jaringan *wireless*, diharapkan jaringan komunikasi di masa mendatang dapat menggunakan jaringan *wireless* dengan lebih baik dalam area lokal maupun area yang luas.

2.1.1 Keuntungan Dan Keterbatasan Pada Teknologi Wireless

Keuntungan utama dari jaringan *wireless* bagi pengguna adalah tidak diperlukannya ada tempat yang tetap dan memiliki kemampuan untuk berpindah secara bebas di dalam sebuah bangunan atau bahkan di luar bangunan. Dalam segi biaya jaringan *wireless* juga lebih menguntungkan dibandingkan dengan jaringan kawat, sebab pada jaringan kawat dibutuhkan kawat tembaga dalam jumlah yang cukup banyak, sehingga biaya yang dibutuhkan juga lebih besar.

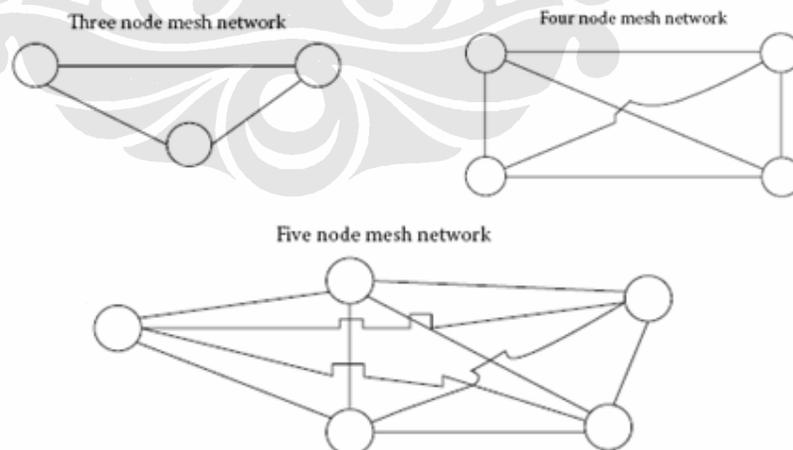
Selain keuntungan-keuntungan tersebut, jaringan *wireless* juga memiliki keterbatasan. Ada beberapa keterbatasan dari penggunaan jaringan *wireless* dibandingkan jaringan kawat. Permasalahan utama adalah keterbatasan bandwidth, menurut standar IEEE 802.11a dan 802.11g yang memiliki kecepatan transfer data 54 Mbps. Dengan kata lain, dengan menggunakan jaringan *wireless* jauh lebih

lambat daripada jaringan kawat tembaga yang bekerja pada kecepatan 100 Mbps. Selain itu juga terdapat kekhawatiran pada jangkauan area yang mampu dicakupnya (*coverage*). Misalnya seperti berapa banyak *access point* yang diperlukan untuk satu bangunan. Satu *access point* dapat menjangkau 30 sampai 150 meter dan lebih tergantung pada struktur dari bangunan tersebut.

Namun semua keterbatasan-keterbatasan yang ada ini dapat ditangani dengan perencanaan dan desain yang baik pada jaringan *wireless* yang akan dibangun pada area tersebut.

2.2 WIRELESS MESH NETWORK

Untuk memahami jaringan *mesh*, pertama kita perlu memiliki pemahaman tentang topologi *mesh*. Jika kita memiliki *node* dalam suatu jaringan, dimana istilah “*node*” menunjukkan perangkat komunikasi yang dapat membawa data dari satu *interface* ke *interface* yang lainnya, kemudian kemampuan dari setiap *node* untuk berkomunikasi dengan tiap-tiap *node* yang lain dalam jaringan yang menggambarkan topologi jaringan *mesh*. Kita dapat melihat struktur jaringan *mesh* dengan menyederhanakan jumlah *node* yang ada dalam jaringan dengan nilai n . Gambar 2.1 mengilustrasikan tiga, empat, dan lima buah *node* dalam struktur jaringan *mesh*, setiap *node* memiliki hubungan komunikasi dengan semua *node* ada dalam jaringan.



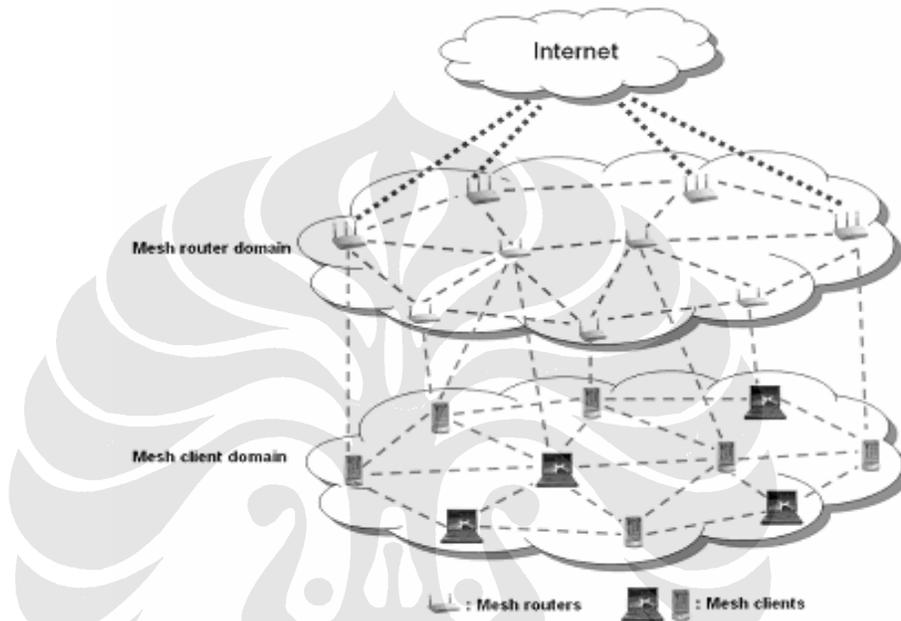
Gambar 2.1. Struktur jaringan *mesh* dengan tiga buah *node*, empat buah *node*, dan lima buah *node* dimana setiap *node* saling terhubung satu sama lain.

Hubungan antara setiap *node* menggambarkan sebuah *link*. Jika diperhatikan jumlah *link* terasosiasi dengan jaringannya, jumlah *link* meningkat seiring dengan meningkatnya jumlah *node*. Tiga buah *link* dibutuhkan untuk menghubungkan jaringan *mesh* yang memiliki tiga buah *node*, enam buah *link* dibutuhkan untuk menghubungkan jaringan *mesh* yang memiliki empat buah *node*, dan sepuluh buah *link* dibutuhkan untuk menghubungkan jaringan *mesh* yang memiliki lima buah *node*. Struktur jaringan *mesh* klasik yang setiap *node*-nya terhubung dengan *node* yang lain menjadi tidak praktis saat jumlah *node* dalam jaringan meningkat, karena *link* yang dibutuhkan untuk menghubungkan setiap *node* akan semakin meningkat pula. [1].

Dalam lingkungan *wireless*, frekuensi radio (pengirim atau penerima) dalam sebuah *node* memiliki kemampuan untuk berkomunikasi dengan *node* lainnya pada jumlah yang tak terbatas. Dengan demikian batasan fisik yang terasosiasi pada konektivitas kawat bukan menjadi persoalan dalam lingkungan *wireless*. Ini berarti proses untuk sebuah *node* berkomunikasi dengan *node* lainnya menjadi praktis dan relatif sederhana, karena sebuah *interface* tunggal dalam bentuk frekuensi radio (pengirim atau penerima) dapat menggantikan *interface-interface* yang dibutuhkan dalam lingkungan kawat. Dengan ketentuan *node* lainnya harus berada pada jarak transmisi agar komunikasi dapat terjadi.

Wireless mesh network merupakan jaringan komunikasi yang memiliki kemampuan dalam mengkonfigurasi dan mengorganisasi dirinya sendiri (*self-configured and self-organized*), dengan kata lain mampu membuat dan menjaga konektivitasnya apabila terjadi kerusakan pada salah satu *node*. Kemampuan ini selain membantu para pengguna untuk dapat selalu terhubung kapan saja dan dimana saja, juga akan membawa keuntungan lain seperti biaya pembuatan yang rendah, kemudahan dalam perawatan jaringan, tingkat *robustness* dan reliabilitas tinggi. *Wireless mesh network* terbentuk dari susunan node radio yang dapat saling terhubung satu sama lain. Node pada *wireless mesh network* terdiri dari *mesh router* dan *mesh client* (Gambar 2.2). Dalam *wireless mesh network* setiap *node* dapat berfungsi sebagai *router* dan *repeater*, yang akan meneruskan data melalui *node* lainnya sampai ke *node* tujuannya.

Wireless mesh network dikembangkan untuk mengantisipasi keterbatasan dan juga meningkatkan performansi dari *wireless adhoc network*, *wireless local area network* (WLAN), dan *wireless metropolitan area network* (WMAN). Dengan berbagai kelebihannya, *wireless mesh network* dapat digunakan untuk menyediakan layanan *wireless* untuk berbagai keperluan dan aplikasi baik untuk kepentingan pribadi, area lokal, kampus ataupun area metropolitan [2].



Gambar 2.2. Node pada *wireless mesh network* terdiri atas *mesh routers* dan *mesh clients*.

2. 2. 1 Arsitektur *Wireless Mesh Network*

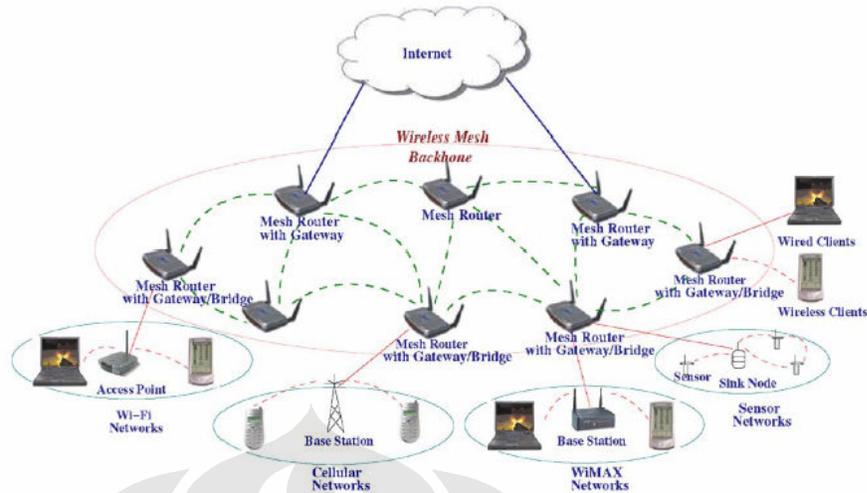
Pada umumnya *wireless mesh network* terdiri dari *mesh router* dan *mesh client* seperti yang terlihat pada Gambar 2.2. *Mesh router* biasanya dilengkapi dengan *multiple wireless interfaces* untuk meningkatkan fleksibilitas dari *mesh network*. Dibandingkan dengan *wireless router* konvensional, *wireless mesh router* dapat memiliki jangkauan area yang sama dengan daya transmisi yang jauh lebih rendah melalui komunikasi *multi-hop*. Walaupun dengan perbedaan tersebut, *wireless router* untuk *mesh* dan konvensional biasanya dibuat berdasarkan *platform hardware* yang sama. *Mesh router* dapat dibangun menggunakan *dedicated computer systems (embedded systems)*, ataupun juga *general purpose systems (desktop PC dan laptop)*.

Untuk *mesh client* juga mempunyai fungsi tertentu pada suatu *mesh network*, dimana suatu *mesh client* juga mempunyai fungsi *router*. Namun *node* ini tidak mempunyai fungsi *gateway* ataupun *bridge* seperti yang ada pada *mesh router*. Oleh karena itu biasanya *mesh client* cukup dilengkapi dengan satu *wireless interface* sehingga *platform hardware* dan *software* yang dipakaipun menjadi lebih sederhana dibandingkan dengan *mesh router*. *Mesh client* dapat menjadi *node* bergerak yang umumnya dijalankan dengan baterai, sehingga pemakaian daya pada *mesh client* harus dibatasi. *Mesh client* dapat terdiri dari *desktop PC*, *laptop*, *PDA*, dan lain sebagainya. Berdasarkan fungsionalitas dari *node-node* tersebut arsitektur pada *wireless mesh network* dapat diklasifikasikan kedalam tiga jenis [2], yaitu :

- 1) *infrastructure wireless mesh network*,
- 2) *client wireless mesh network*,
- 3) *hybrid wireless mesh network*.

2. 2. 1. 1 *Infrastructure Wireless Mesh Network*

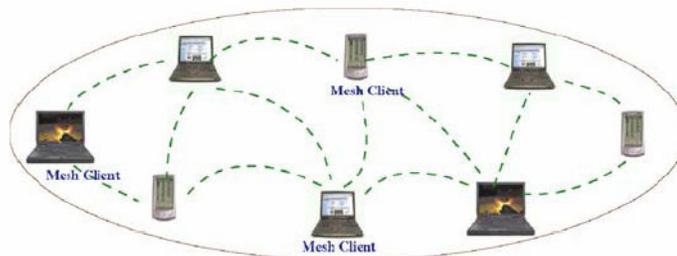
Pada tipe arsitektur ini terdapat beberapa *mesh router* yang membentuk sebuah infrastruktur bagi *client-client* yang terhubung dengannya. Seperti terlihat pada Gambar 2.3 dimana *wireless mesh network* dapat terkoneksi dengan *mesh client* ataupun dengan berbagai jenis teknologi *wireless* lainnya yang dalam hal ini bertindak sebagai *client* dari jaringan infrastruktur yang dibentuk oleh *mesh router*. Dengan kemampuan sebagai *gateway*, *mesh router* juga dapat dihubungkan ke internet. *Mesh router* harus mempunyai kemampuan untuk mengkonfigurasi serta memperbaiki hubungan antara *router* secara mandiri, sehingga hubungan tidak sampai terputus. Untuk *client* konvensional yang memiliki teknologi yang sama dengan *wireless mesh network*, mereka dapat terhubung langsung dengan *mesh router*. Sedangkan untuk *client* yang menggunakan teknologi *wireless* selain *wireless mesh network* maka mereka dapat terhubung dengan *mesh router* melalui sebuah *base station* yang terhubung dengan *mesh router* melalui kabel.



Gambar 2.3. Arsitektur *infrastructure wireless mesh network*.

2. 2. 1. 2 *Client Wireless Mesh Network*

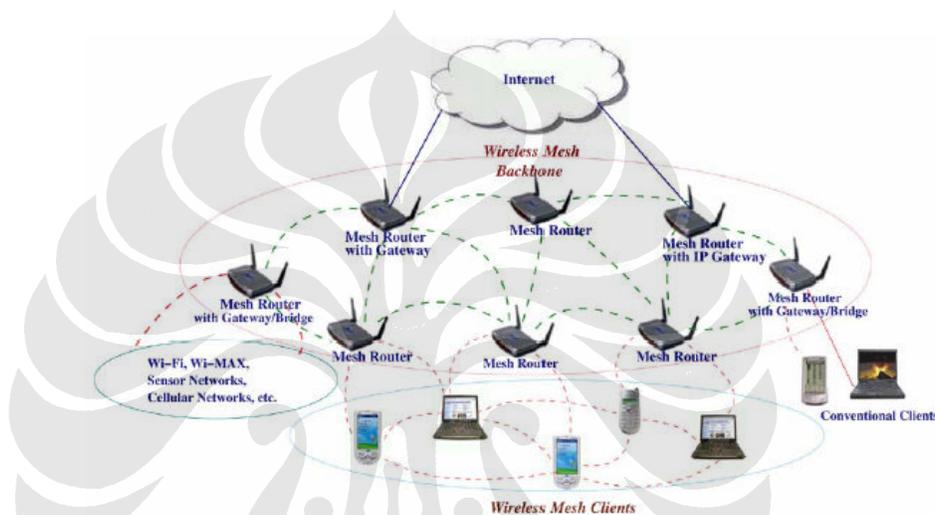
Client wireless mesh network menyediakan jaringan *peer-to-peer* diantara *node mesh client*, dimana *mesh client* dapat bertindak sebagai *host* dan *router*. Pada jenis arsitektur ini, jaringan terbentuk dari sekumpulan *node mesh client* yang dapat melakukan fungsi *routing* dan konfigurasi serta menyediakan aplikasi *end-user* pada pengguna jaringan, seperti yang terlihat pada Gambar 2.4. Tipe arsitektur ini tidak memerlukan *mesh router*, oleh karena itu tingkat mobilitasnya menjadi lebih tinggi bila dibandingkan dengan tipe arsitektur *infrastructure wireless mesh network*. Pada *client wireless mesh network*, paket yang dikirimkan ke suatu *node* tujuan akan melalui serangkaian lompatan (*hops*) melalui beberapa *node* untuk mencapai tujuan. Oleh karena itu, apabila dibandingkan dengan tipe *infrastructure wireless mesh network*, maka *node mesh client* pada tipe ini memerlukan kebutuhan akan aplikasi *end-user* yang lebih tinggi karena harus memiliki kemampuan *routing* serta konfigurasi sendiri.



Gambar 2.4. Arsitektur *client wireless mesh network*.

2. 2. 1. 3 Hybrid Wireless Mesh Network

Tipe ini merupakan bentuk gabungan dari dua tipe arsitektur lainnya yaitu *client* dan *Infrastructure wireless mesh network*. Seperti yang terlihat dalam Gambar 2.5, *mesh client* dapat terhubung pada jaringan melalui *mesh router* sekaligus tetap berhubungan langsung dengan *mesh client* lainnya. Selain itu jaringan *infrastructure* dari *mesh router* juga dapat terhubung dengan jaringan teknologi *wireless* lainnya.



Gambar 2.5. Arsitektur *hybrid wireless mesh network*.

2. 2. 2 Karakteristik *Wireless Mesh Network*

Wireless mesh network memiliki beberapa karakteristik umum yang sangat mempengaruhi kinerjanya [2], seperti berikut ini.

1. *Multi-hop wireless network*.

Karakteristik ini berguna untuk meningkatkan area jangkauan dari jaringan tanpa harus mengorbankan kapasitas kanal. Selain itu berguna pula untuk menyediakan bentuk layanan *Non Line-of-Sight* (NLOS).

2. Kemampuan *self-forming*, *self-healing*, *self-organizing* serta mendukung *ad-hoc networking*.

Karakteristik ini menambah performansi dari *wireless mesh network* karena membawa sifat fleksibel dalam jaringan, pembuatan dan konfigurasi yang mudah, serta *fault tolerance*.

3. Tingkat mobilitas tergantung dari jenis *node*.

Mesh router biasanya memiliki tingkat mobilitas yang lebih rendah dibanding dengan sebuah *mesh client* yang dapat bersifat *fixed* maupun *mobile*.

4. Dapat mengakses ke berbagai jenis teknologi jaringan lainnya.

Dalam *wireless mesh network* sebuah jaringan infrastruktur dari *mesh router* dapat terhubung baik ke *mesh client*, internet maupun dengan berbagai jenis teknologi jaringan lainnya.

5. Dependensi terhadap pemakaian daya tergantung dari jenis *node*.

Mesh router tidak memiliki suatu batasan tertentu dalam pemakaian daya karena biasanya bersifat *fixed* dan terhubung langsung dengan sumber daya. Sedangkan bagi *mesh client* kemungkinan memerlukan suatu protokol tertentu untuk mengatur pemakaian daya karena seringkali *mesh client* bersifat *mobile*.

2. 2. 3 Protokol Dalam Wireless Mesh Network

Protokol-protokol yang digunakan dalam suatu sistem *wireless mesh network* dapat dibagi menjadi beberapa bagian menurut lapisan (*layer*), mulai dari *physical layer* hingga *application layer*.

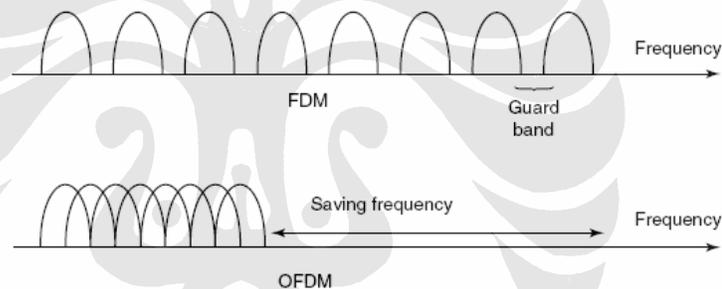
2. 2. 3. 1 Physical Layer Protocol

Permasalahan utama pada *physical layer* untuk *wireless mesh network* terletak pada pemilihan teknologi *radio* yang tepat. Pemilihan teknologi *radio* dapat didasarkan pada pertimbangan teknologi dan pertimbangan ekonomi.

Pertimbangan teknologi mengacu pada efisiensi spektrum, *physical layer data rate*, dan kemampuan beroperasi pada saat terjadi interferensi. Jenis *protocol* yang paling banyak digunakan untuk *physical layer* dalam sebuah sistem *wireless mesh network* saat ini adalah *Orthogonal Frequency Division Multiplexing* (OFDM) yang memungkinkan pengiriman data berkecepatan tinggi dalam lingkungan yang *mobile*. OFDM memiliki *physical layer data rate* maksimum sebesar 54 Mbps, pada jaringan yang padat dengan tingkat interferensi yang tinggi hal ini mungkin tidak dapat tercapai. OFDM merupakan perkembangan dari

sistem *Frequency Division Multiplexing* (FDM) dimana yang dapat mengirimkan berbagai sinyal secara simultan dengan cara membagi-bagi mereka ke dalam beberapa *band* frekuensi yang berbeda (*subcarrier*).

Dalam FDM terdapat *guard band* yang berfungsi untuk mengurangi interferensi antar frekuensi yang berbeda, hal ini menyebabkan adanya bandwidth yang terbuang percuma, sedangkan OFDM menerapkan sistem yang lebih efisien dimana OFDM tidak lagi menggunakan *guard band* untuk mengurangi interferensi melainkan dengan memodulasi sinyal secara *orthogonal* (lihat Gambar 2.6). Dengan cara ini OFDM dapat menyediakan lebih banyak ruang pada bandwidth untuk dipakai. OFDM menggunakan *Fast Fourier Transform* (FFT) yang dan *Inverse FFT* untuk mengubah serial data ke dalam bentuk *multiple channel*. OFDM mempunyai 256 *subchannel* (*carrier*), dan sinyal asli akan dibagi-bagi menjadi 256 *subcarrier* dan ditransmisikan secara paralel.



Gambar 2.6. Perbandingan antara FDM dan OFDM.

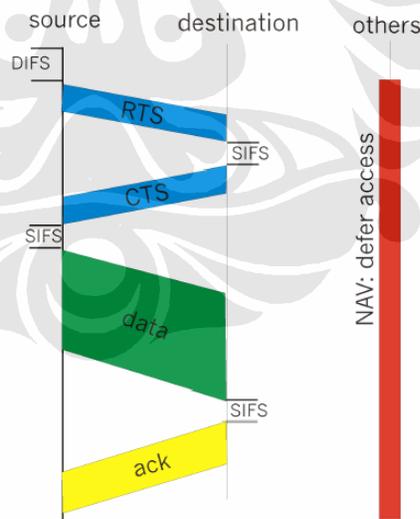
Pertimbangan ekonomi atau sosial mengarah pada harga peralatan yang murah dan dapat memberikan kemudahan bagi pengguna. Sebagai contoh adalah bukti suksesnya IEEE 802.11b berbasis *wireless mesh network* dimana murahanya kartu *interface* jaringan memberikan kontribusi atas suksesnya perkembangan *wireless mesh network*. Oleh sebab itu pada saat pemilihan teknologi *physical layer*, desain jaringan harus memperhatikan aplikasi dan kebutuhan penggunanya.

Namun dalam perkembangannya, lebih banyak lagi protokol yang dapat digunakan pada *physical layer* dari sebuah sistem *wireless mesh network* seperti *Code Division Multiple Access* (CDMA), *Ultra Wideband Access* (UWB) ataupun *Multiple Input Multiple Output* (MIMO) yang dapat digunakan sesuai dengan kebutuhan.

2. 2. 3. 2 *Medium Access Control (MAC) Layer Protocol*

Protokol yang dikembangkan untuk digunakan pada *MAC layer* untuk sebuah sistem *wireless mesh networking* adalah *protocol MAC* berbasis IEEE 802.11 yaitu *Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)* dengan paket kontrol *RTS/CTS (Ready to Send/Clear to Send)*. *Frame* yang akan dikirimkan membawa informasi waktu perkiraan untuk *frame* tersebut agar dapat sampai pada tujuan. Waktu perkiraan inilah yang akan dipakai oleh *node* yang lain dalam menentukan waktu minimum pemakaian jalur atau *network location vector (NAV)*.

Dengan menggunakan paket kontrol *RTS/CTS*, saat pengirim akan mengirimkan *frame* maka ia akan mengirimkan paket *RTS* yang berisi informasi waktu yang dibutuhkan untuk mengirimkan data dan sinyal *Acknowledgement (ACK)*. Penerima yang menerima paket *RTS* tersebut akan mengirimkan paket *CTS* sebagai balasan yang menandakan bahwa pengirim dapat mulai mengirimkan data. *Node* lain yang mengetahui adanya sinyal *RTS/CTS* ini dapat menunggu hingga waktu pengiriman selesai untuk menghindari adanya *collision*. Hal tersebut diperlihatkan pada Gambar 2.7 dibawah ini.



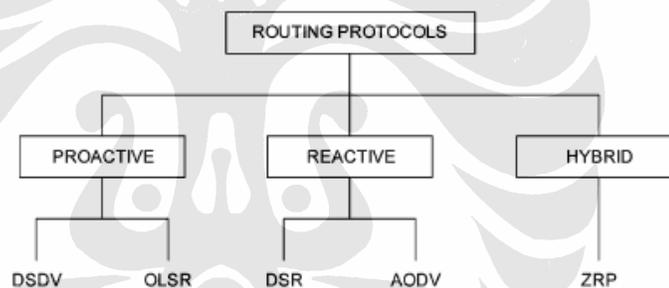
Gambar 2.7. CSMA/CA dengan paket kontrol *RTS/CTS* [3].

2. 2. 3. 3 *Network Layer Protocol*

Protokol yang dipakai dalam *network layer* pada suatu *wireless mesh network* banyak mengadaptasi *routing protocol* yang digunakan dalam *wireless*

ad-hoc network yang dibagi menjadi tipe *proactive*, *reactive* dan *hybrid* (lihat Gambar 2.8). Protokol *routing* tipe *proactive* mengirimkan informasi seperti keterangan *node* tetangga, rute dan lain-lain secara *broadcast* dalam periode tertentu. Hal ini memungkinkan waktu *set-up* yang cepat, namun meningkatkan penggunaan *overhead*. Contoh dari tipe *protocol* ini adalah *Destination Sequenced Distance Vector* (DSDV) dan *Optimized Link State Routing* (OLSR).

Pada protokol *routing* tipe *reactive* rute hanya dibangun berdasarkan adanya permintaan, sehingga mengurangi pemakaian *overhead* pemilihan rute namun menimbulkan *delay* yang cukup besar pada saat pengiriman *frame* pertama. Contoh dari protokol *routing* ini adalah *Dynamic Source Routing* (DSR) dan *Ad-hoc On-demand Distance Vector* (AODV). Sedangkan untuk protokol *routing* tipe *hybrid* merupakan gabungan dari kedua tipe tersebut diatas. Contoh dari protokol *routing* ini adalah *Zone Routing Protocol* (ZRP).



Gambar 2.8. Klasifikasi protokol *routing* dalam *wireless mesh network*.

2. 2. 3. 4 *Transport Layer Protocol*

Protokol TCP yang selama ini diadaptasi untuk *wireless mesh network* walaupun tetap digunakan namun tidak memberikan hasil yang maksimal. Kekurangan dari TCP dalam *wireless mesh network* karena pada TCP yang *standard* tidak membedakan *loss* yang diakibatkan oleh *congestion* maupun *non-congestion*, sehingga bila terjadi *non-congestion loss* maka *throughput* jaringanpun menurun drastis. Selain itu apabila terjadi perbaikan jalur setelah mengalami kerusakan, protokol TCP konvensional tidak dapat diperbaiki dengan cepat. Untuk itu menutupi kekurangan-kekurangan ini, sedang dikembangkan *protocol-protocol* berbasis TCP yang dapat dipakai untuk *wireless mesh network*.

2. 2. 3. 5 *Application Layer Protocol*

Hampir semua dari *protocol-protocol* yang bekerja pada *layer 7* pada OSI *layer* dapat digunakan dengan baik pada *wireless mesh network*. Terdapat tiga fungsi utama aplikasi yang dapat digunakan dalam suatu *wireless mesh network* yaitu akses internet, penyimpanan informasi terdistribusi, pertukaran informasi antara jaringan teknologi *wireless* lainnya.

2. 2. 4 *Routing Dalam Wireless Mesh Network*

Wireless mesh network merupakan *trend* baru dalam komunikasi *wireless* yang menjanjikan fleksibilitas tinggi, keandalan, dan performa diatas konvensional *wireless local network* (WLAN). *Wireless mesh network* dan jaringan *ad-hoc* bergerak menggunakan konsep komunikasi yang sama antara *node*-nya, namun menitikberatkan pada aspek yang berbeda. Jaringan *ad-hoc* bergerak (*mobile ad-hoc network* atau MANET) memiliki latar belakang akademis dan memusatkan pada perangkat pengguna, mobilitas, dan kemampuan *ad-hoc*. WMN memiliki latar belakang bisnis dan memusatkan pada perangkat statis (biasanya infrastruktur), keandalan, dan kapasitas jaringan. Fungsi utama dari jaringan *multihop ad-hoc* sama seperti *wireless mesh network* yaitu kemampuan *routing*-nya. Protokol *routing* menyediakan jalur yang dibutuhkan melalui *wireless mesh network*, sehingga *node* dapat terhubung dengan baik atau memiliki jalur terbaik pada *multiple wireless hop*. Protokol *routing* harus memperhitungkan kesulitan pada lingkungan *radio* dengan keadaan yang sering kali berubah dan harus mendukung komunikasi yang handal serta yang efisien pada jaringan *mesh*.

Karena *wireless mesh network* memiliki fitur serupa dengan jaringan *ad-hoc wireless*, protokol *routing* yang dikembangkan untuk MANET dapat diterapkan pada *wireless mesh network*. Sebagai contoh, *microsoft mesh network* yang dibuat berdasarkan pada *Dynamic Source Routing* (DSR), dan masih banyak perusahaan lainnya seperti penggunaan *Ad-hoc On-demand Distance Vector* (AODV).

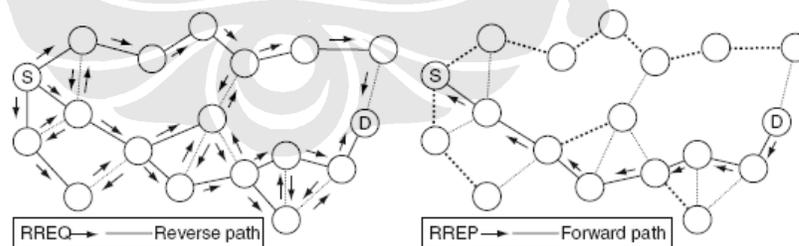
2. 2. 5 *Protokol Routing Ad hoc On-demand Distance Vector (AODV)*

AODV adalah protokol *routing* yang populer dari MANET, yang merupakan protokol *routing reactive*. Pada protokol *routing* ini jalur diatur

berdasarkan permintaan, dan hanya jalur yang aktif yang akan dipertahankan. AODV sudah distandarisasi oleh IETF dalam percobaan RFC 3561 [4].

AODV menggunakan mekanisme permintaan-balasan (*request-reply*) sederhana untuk menemukan jalurnya. Saat jalur ke tujuan tidak diketahui, AODV membuat paket permintaan jalur dan menyiarkannya ke tetangganya. Pesan permintaan jalur berisi identitas sumber (*source ID*), identitas tujuan (*destination ID*), urutan nomor sumber (*source sequence number*), urutan nomor tujuan (*destination sequence number*), penghitung *hop* (*hop count*), dan identitas penyiaran (*broadcast ID*) [5].

Pada saat ada permintaan jalur dari sebuah *node*, maka *node* tersebut akan memancarkan (*broadcast*) sebuah paket untuk meminta rute. Setiap *node* lain yang berada pada jaringan akan meneruskan paket tersebut serta mencatat dari mana paket tersebut berasal dan berapa *hop* yang telah dilalui. Apabila paket permintaan tersebut telah sampai pada *node* yang dituju atau sampai pada *node* yang mengetahui jalur menuju *node* yang dituju maka *node* tersebut akan mengirimkan paket balasan berisi informasi rute yang diperlukan. *Node* yang ingin mengirimkan paket akan menyeleksi paket balasan yang masuk dengan melihat jumlah *hop* terkecil, atau bila memiliki jumlah yang sama maka yang dipilih adalah paket yang paling dahulu datang. Hal ini dapat dilihat pada Gambar 2.9. Apabila ada kerusakan rute pada saat pengiriman maka akan ada pesan kerusakan yang dikirimkan ke *node* pengirim.



Gambar 2.9. Pencarian rute AODV: disebelah kiri adalah rute permintaan (*route request*) dan disebelah kanan adalah rute balasan (*route reply*) [6].

Setiap *node* mengamati status setiap *link*-nya, dan saat terjadi perubahan koneksi pada *link* atau apabila ada kerusakan rute pada saat pengiriman *node* akan membuat pesan kesalahan rute dan memberitahukan anggotanya tentang rute yang

tidak dapat dicapai. AODV mengandalkan skema *layer medium access control* (MAC) atau menggunakan paket *beacon* dengan *periodic interval* untuk mengetahui status koneksi dengan tetangganya.

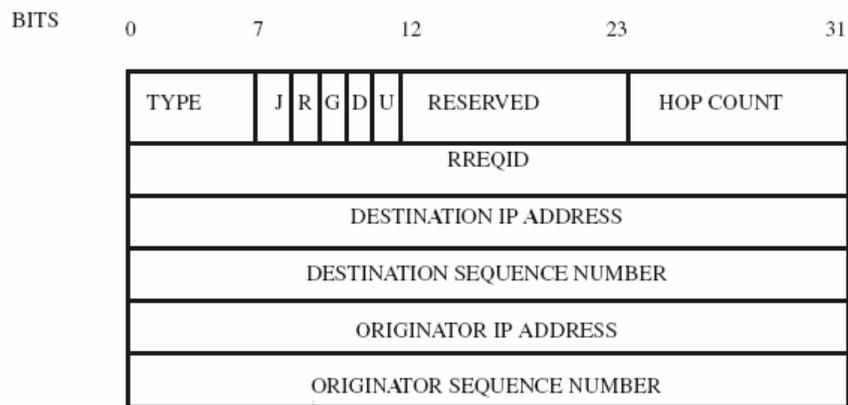
Keunggulan dari AODV adalah membuat rute hanya dalam daerah sebarannya, yang mengurangi pesan kontrol periodik berlebih yang terasosiasi dengan protokol *routing proactive*. Kekurangan dari AODV adalah terletak pada *node* penghubung (*hop*) dapat mengarahkan ke rute yang kurang efisien jika urutan nomor sumber berumur tua (rendah) dan *node* penghubung memiliki urutan nomor tujuan yang lebih tinggi. Juga rute paket balasan (RREP) yang banyak dalam menanggapi satu rute paket permintaan (RREQ) dapat menyebabkan kontrol berlebih yang tinggi [5].

Dalam proses pencarian jalurnya AODV menggunakan mekanisme format pesan (*message formats*) dalam bentuk paket-paket. Format pesan tersebut adalah *Route Request* (RREQ), *Route Reply* (RREP), *Route Error* (RERR), dan HELLO [1] [4].

2. 2. 5. 1 *Route Request* (RREQ)

Pada saat sebuah *node* ingin berkomunikasi dengan *node* lain, *node* tersebut akan memancarkan (*broadcast*) sebuah paket *route request* (RREQ) ke *node* tetangganya yang akan diteruskan ke *node* lainnya sampai ke *node* tujuannya. Paket RREQ yang dikirimkan suatu *node* ke *node* yang lain berisi RREQ ID, *destination IP address*, *destination sequence number*, *originator IP address*, dan *originator sequence number* (Gambar 2.10).

Pada saat proses pengiriman, setiap *node* yang menerima paket RREQ akan memeriksa *Destination IP Address*-nya apakah *node* tersebut adalah *node* tujuan. Jika *node* tersebut adalah *node* tujuan maka penjaluran berhenti sampai di *node* tersebut, dan sebaliknya apabila *node* tersebut bukanlah *node* tujuan, ia akan meneruskan paket RREQ tersebut ke *node* tetangganya. Proses ini akan berulang terus hingga paket RREQ menemui *node* tujuan atau apabila jumlah *hop count* telah memenuhi batas. *Sequence number* digunakan untuk menunjukkan tingkat *up-to-date* informasi pada paket tersebut, makin besar nilai *sequence number* maka makin baru informasi yang ada pada paket tersebut.



Legend

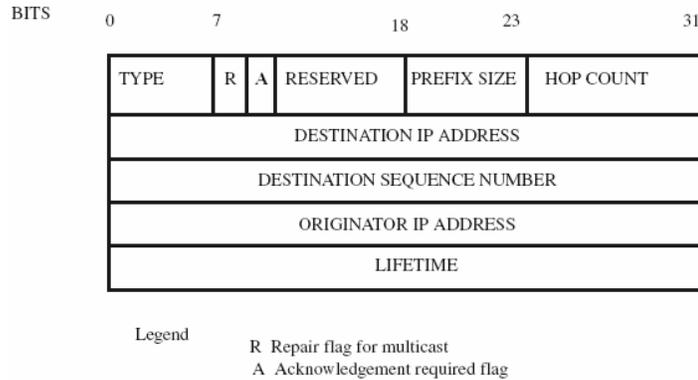
- J Join flag, reserved for multicasting
- R Repair flag, reserved for multicast
- G Gratuitous flag, indicates if a gratuitous RREP should be unicast to the node specified in the destination IP address field
- D Destination only flag, indicates only the destination should respond to this RREQ message
- U Unknown sequence number, indicates destination number unknown

Gambar 2.10. Format paket pesan *route request* (RREQ) [1].

2. 2. 5. 2 *Route Reply (RREP)*

Setelah paket RREQ sampai pada *node* tujuannya akan membentuk jalur yang menuju ke *node* yang mengirimnya, jalur ini akan digunakan kembali untuk mengirim paket RREP. Paket RREP merupakan paket yang dikirimkan oleh *node* tujuan atau *node* yang mempunyai informasi jalur menuju *node* tujuan. *Node* yang mengirimkan paket akan menyeleksi paket balasan yang masuk dengan melihat jumlah *hop* terkecil, atau bila memiliki jumlah yang sama maka yang dipilih adalah paket yang paling dahulu datang.

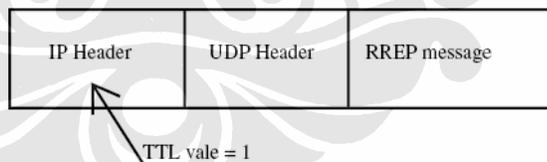
Pada Gambar 2.11 paket format RREP, terdapat bit A yang dapat diset apabila *node* tujuan memerlukan sinyal *Acknowledgement* dari RREP yang telah dikirimkan. Ada dua jenis *field* yang berbeda dari format paket RREP yaitu *Prefix Size* dan *Lifetime*. *Prefix Size* apabila diset dengan nilai diluar nol maka akan menyatakan bahwa *node* selanjutnya akan dapat dipakai oleh setiap *node* yang memiliki rute yang sama dengan *node* tujuan. Sedangkan *Lifetime* yang terdiri dari 32 bit menunjukkan waktu *valid* dari suatu rute (dalam *millisecond*) bagi setiap *node* yang menerima paket RREP.



Gambar 2.11. Format paket pesan route reply (RREP) [1].

2. 2. 5. 3 Pesan HELLO

Sebuah *node* dapat memeriksa informasi konektifitas dengan *node* tetangganya dengan cara memancarkan (*broadcasting*) paket pesan *HELLO* secara periodik selama jalur tersebut merupakan jalur yang aktif. Paket *HELLO* merupakan sebuah paket RREP dengan nilai *Time To Live* (TTL) adalah 1 yang ditentukan pada *IP header* dari pesan. Dalam paket RREP *destination IP address* diisi dengan *IP address* dari *node* yang mengirimkan pesan *HELLO*. Kemudian dengan nilai *field destination sequence number* diisi dengan nilai *sequence* terakhir dari *node* tersebut serta nilai *field hop count* diset dengan nilai 0. Gambar 2.12 menunjukkan format dari paket pesan *HELLO*.

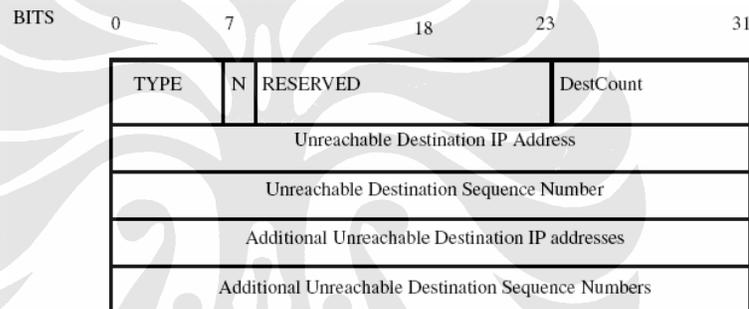


Gambar 2.12. Format paket pesan *HELLO* [1].

Jika suatu *node* tidak dapat menerima paket *HELLO* dari *node* tetangganya maka tabel rute (*routing table*) yang ada tersebut akan dihapus datanya dari *node* tetangga tersebut. Dan apabila hal tersebut terjadi pada saat pembuatan hubungan maka *node* tersebut akan mengirimkan paket RRER.

2. 2. 5. 4 Route Error (RRER)

Paket RRER memungkinkan AODV untuk menyesuaikan rute saat *node* tersebut berpindah tempat ataupun terjadi kerusakan pada suatu rute. Saat suatu *node* tidak menerima lagi paket *HELLO* dari *node* tetangganya, maka *node* tersebut akan menghapus rute dari *node* yang tidak mengirim paket *HELLO* itu dari tabel jalurnya. Kemudian *node* tersebut akan mengirimkan paket pesan RRER ke *node* tetangganya yang menggunakan *node* tersebut sebagai jalur ke *node* yang tidak mengirim paket *HELLO* tadi dan diteruskan ke *node-node* yang lainnya. Setelah RRER diterima maka jalur yang ada pada setiap *node* untuk menuju pada *node* yang tidak mengirim paket *HELLO* tersebut akan dihapus dari tabel jalurnya. Gambar 2.13 memperlihatkan format untuk paket pesan RRER.



Gambar 2.13. Format paket pesan RRER [1].

Dest count field menandakan jumlah *node* tujuan yang tidak dapat dicapai yang termasuk di dalam paket tersebut. *Dest count field* ini harus memiliki nilai minimal adalah 1. Apabila suatu *node* menerima paket RRER maka *node* tersebut akan menghapus semua informasi rute pada tabel jalur yang berkaitan dengan *node* tersebut.

2. 2. 5. 5 Informasi Yang Terdapat Pada Tabel Routing

Setiap tabel *routing* yang terdapat pada AODV terdiri dari informasi sebagai berikut ini [4].

1. *Destination IP address*

Bagian ini berisi informasi alamat IP (internet protokol) dari *node* yang merupakan tujuan *routing*-nya.

2. *Destination sequence number*

Bagian ini berisi informasi yang menunjukkan nomor urut *routing* dari *node* tujuan pada tabel *routing*.

3. *Valid destination sequence number flag*

Bagian ini berisi informasi yang menunjukkan status *flag* pada *destination sequence number*.

4. *Interface*

Bagian ini berisi informasi tentang *interface* jaringan yang digunakan untuk mencapai *node* tujuan.

5. *Hop count*

Bagian ini berisi informasi yang menunjukkan keterangan jumlah *hop* yang akan dilalui untuk mencapai *node* tujuannya.

6. *Next hop*

Bagian ini berisi informasi yang menunjukkan keterangan *hop* selanjutnya yang akan dilalui saat menuju ke *node* tujuan.

7. *List of precursors*

Bagian ini berisi daftar tentang *node-node* tetangganya yang dapat memberikan jalur untuk meneruskan paket.

8. *Lifetime*

Bagian ini berisi informasi yang menunjukkan waktu berakhirnya atau waktu penghapusan jalur tersebut.

9. *State*

Bagian ini berisi informasi yang menunjukkan status jalur tersebut, seperti dapat digunakan (*valid*), tidak dapat digunakan (*invalid*), perlu perbaikan (*repairable*), dalam perbaikan (*being repaired*).

2.3 AD HOC ON-DEMAND DISTANCE VECTOR–UPPSALA UNIVERSITY (AODV-UU)

AODV-UU adalah implementasi protokol *routing* AODV pada Linux, yang dikembangkan oleh Universitas Uppsala, Swedia. AODV-UU ini berjalan sebagai *user-space daemon* dan memelihara *kernel* tabel *routing*. AODV-UU ditulis

dengan bahasa pemrograman C dan telah di terbitkan dibawah GNU *General Public License* (GPL).

AODV-UU mengimplementasikan hampir semua hal pada AODV. Salah satu tujuan dari AODV-UU adalah untuk memenuhi implementasi dari AODV yang sesuai dengan *draft* terakhir dan tujuan ini menopang pengembangan perangkat lunak yang berkelanjutan.

Kebutuhan sistem dari AODV-UU lebih sederhana, dengan menggunakan *distro* linux versi kernel 2.4.x bersama dengan menggunakan kartu jaringan *wireless* (ini juga memungkinkan digunakan pada jaringan kawat). Sebagai tambahan, AODV-UU dapat di *cross-compile* untuk digunakan pada *platform* ARM, sehingga AODV-UU ini dapat digunakan pada PDA yang populer seperti COMPAQ iPAQ dan Sharp Zaurus [7].

2.3.1 Konfigurasi

AODV-UU menyediakan banyak pilihan perintah untuk mengatur operasinya. Pilihan ini disediakan sebagai parameter pada *command line* untuk *aodvd routing daemon* [7]. Perintah pilihan yang tersedia adalah sebagai berikut:

- 1) *daemon mode* (-d, --daemon): menjalankannya di *background*,
- 2) *force gratuitous* (-g, --force-gratuitous): memaksa *gratuitous flag* untuk diterapkan pada semua RREQ,
- 3) *help* (-h, --help): menampilkan informasi bantuan,
- 4) *interface* (-i, --interface): menspesifikasikan *interface* jaringan mana yang akan digunakan pada AODV-UU. *Default*-nya adalah *interface* awal jaringan *wireless* tersebut,
- 5) *HELLO jittering* (-j, --hello-jitter): menonaktifkan *jitter* dari pesan *HELLO*.
- 6) *logging* (-l, --log): mengaktifkan *logging* ke AODV-UU *logfile*,
- 7) *routing table logging* (-r N, --log-rt-table N): mencatat isi dari tabel *routing* ke tabel *routing logfile* setiap N detik,
- 8) *N HELLOs* (-n N, --n-hellos N): memerlukan pesan *N HELLO* untuk diterima dari sebuah *node* sebelum ditetapkan sebagai *node* tetangganya,
- 9) *uni-directional hack* (-u, --unidir-hack): memperbolehkan pendeteksian dan penghindaran dari *link uni-directional*. Ini masih fitur percobaan,

- 10) *gateway mode* (-w, --gateway-mode): memperbolehkan dukungan internet *gateway*. Ini juga masih fitur percobaan,
- 11) *disabling of expanding ring search* (-x, --no-expanding-ring): menonaktifkan perluasan daerah pencarian untuk RREQ, yang normalnya digunakan untuk membatasi penyebaran RREQ didalam jaringan,
- 12) *no wait-on-reboot* (-D, --no-worb); menonaktifkan penundaan selama 15 detik *wait-on-reboot* pada saat *startup*,
- 13) *version information* (-V, --version); menampilkan informasi versi dan *copyright*.

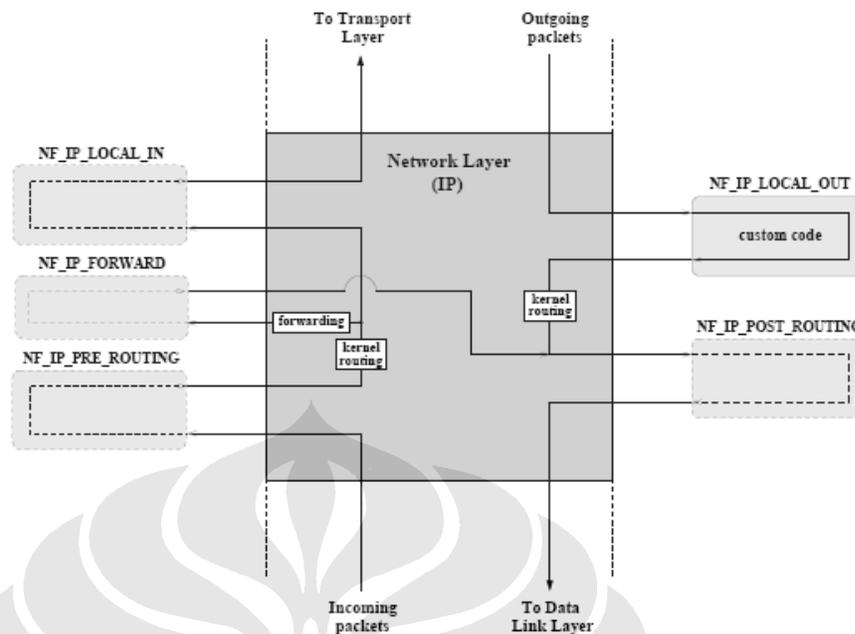
2.3.2 Interaksi Dengan Internet Protokol

Mengingat AODV merupakan protokol yang reaktif, maka penentuan rute dilakukan berdasar pada permintaan. Hal ini membutuhkan implementasi protokol *routing* yang mampu meneruskan permintaan (*request*) ke tujuan saat rute tujuan tersebut tidak ada. Penghapusan rute pada tabel *routing* yang telah lewat batas waktunya membutuhkan dukungan dari setiap *host* untuk pemantauan paketnya.

Implementasi AODV sebelumnya seperti *Multicast ad-hoc* (Mad-hoc) AODV tidak dapat melakukan hal tersebut diatas. Hal ini menghambat pengoperasian pada protokol yang berorientasi koneksi seperti TCP, dimana awal paket sangat vital untuk *setup* koneksi. Sejak itu, penanganan paket pada linux telah mengalami perkembang yang sangat cepat. Khususnya *software* yang disebut Netfilter telah dikembangkan, yang dapat memberikan penanganan paket yang sangat fleksibel. AODV-UU menggunakan Netfilter untuk pemrosesan semua paket dan keperluan modifikasi [7].

2.3.3 Netfilter Framework

Netfilter adalah sebuah *framework kernel* linux yang digunakan untuk mengelola paket. Untuk setiap protokol jaringan, beberapa *hook* diberikan. *Hook* ini berhubungan dengan tempat-tempat yang telah ditentukan dengan baik didalam *stack* protokol dan memperbolehkan memasukkan kode yang telah disesuaikan untuk pengelolaan paket dalam bentuk modul *kernel* (lihat Gambar 2.14).



Gambar 2.14. Netfilter *hooks* untuk IP. Pengiriman paket pada *hook* ini dapat diterima dan dimodifikasi dengan kode *segment* yang telah disesuaikan (modul *kernel*).

Setiap paket yang datang pada *hook* akan dihantarkan ke dalam sebuah *code segment* yang telah dicatat pada *hook* tersebut. Hal ini memungkinkan paket untuk dirubah, dibuang atau dirutekan ulang oleh kode *segment* modifikasi tersebut. Ketika sebuah paket diproses (dan mungkin dirubah), sebuah *verdict* (keputusan) harus dikembalikan ke *Netfilter*. *Verdict* ini menginstruksikan *Netfilter* untuk melakukan beberapa aksi pada paket yang berhubungan, seperti untuk membuang paket atau membiarkan untuk melanjutkan perjalanannya melalui *stack* protokol.

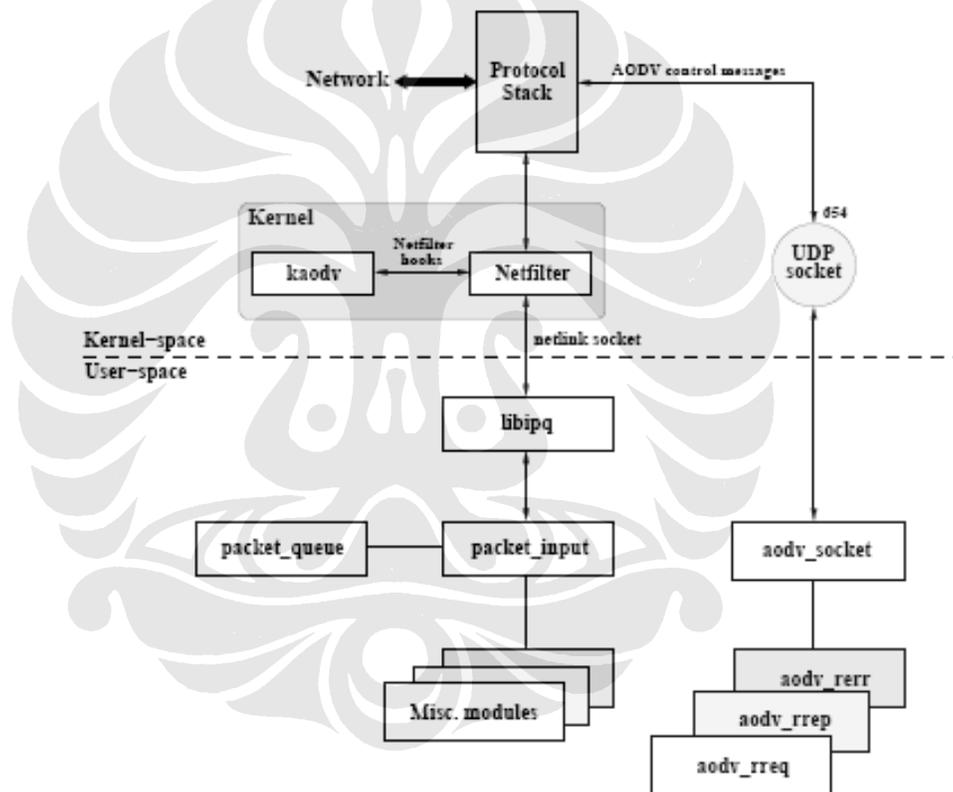
Netfilter juga menawarkan kemampuan untuk memproses paket dalam *user-space*. Dengan mengembalikan antrian keputusan khusus (*special queue verdict*), paket diantriakan pada *kernel* dan informasi tentang paket dikirim ke *user-space* melalui *netlink socket*. Paket yang antri akan tetap diantriakan sampai aplikasi *user-space* (disisi lain dari socket) mengembalikan *verdict* menandakan aksi yang diinginkan untuk paket-paket ini.

Didalam AODV-UU, sebuah komponen modul *kernel* mendengarkan secara konstan kedua paket *inbound* dan *outbound*, dengan mendaftarkan dirinya pada *netfilter*. Paket yang diantriakan sesuai kebutuhan memperbolehkan pemrosesan

user-space dilakukan oleh AODV-UU *routing daemon*. Hal ini memperbolehkan penentuan rute sesuai permintaan dari operasi AODV menjadi nyata. Pada akhirnya, *netfilter* memperbolehkan implementasi secara mandiri dari perubahan *kernel*. Ini adalah keuntungan besar didalam pemeliharaan *software* [7].

2.3.4 Penanganan Paket

Penanganan paket (*packet handling*) mampu membedakan antara paket data dan pesan kontrol AODV, dan menanganinya secara terpisah menggunakan modul perangkat lunak yang berbeda (lihat Gambar 2.15) [7].



Gambar 2.15. Penanganan paket pada AODV-UU. Paket data dan pesan kontrol AODV ditangani secara terpisah.

2.3.5 Kedatangan Paket

Ketika sebuah paket melintasi *stack* protokol, paket tersebut ditangkap oleh *netfilter hook* yang telah diatur oleh modul *kernel* AODV-UU, *kaodv*. Paket *non-*

IP selalu diterima, karena paket ini tidak memiliki hubungan dengan AODV-UU. Umumnya paket yang dibangkitkan selalu diantrikan, sebab sebuah rute harus ditentukan. Kontrol pesan AODV yang masuk juga selalu diterima, karena paket ini yang akan diproses pada socket UDP yang terpisah. Sedang paket yang masuk dalam antarmuka jaringan (*network interface*) lainnya, tetapi tidak diproses kemudian [7].

2.3.6 Pemrosesan Paket

Jika paket yang masuk adalah kontrol pesan AODV, verdict yang diterima dikembalikan ke modul libpq sehingga paket tersebut akan berakhir di socket UDP kontrol pesan AODV, diterima atau dikirim keluar, tergantung dari kondisi paket tersebut apakah paket *incoming* atau *outgoing* [7].

2.3.7 Pemrosesan Paket Data

Jika tujuan dari paket (ditentukan oleh alamat IP tujuan) adalah *host* yang sedang dikunjungi, maka paketnya merupakan paket *broadcast*, atau *mode internet gateway* telah di-*enable*-kan dan paket bukan merupakan paket *broadcast*, maka paket akan diterima. Ini berarti bahwa paket dalam kondisi seperti ini akan ditangani layaknya paket biasa oleh sistem operasi.

Selanjutnya, paket akan diteruskan, dimasukkan dalam antrian atau dibuang. Tabel *routing* internal dari AODV-UU digunakan untuk mengecek apakah rute yang aktif sekarang masih ada atau tidak. Jika rute masih ada, maka paket di-set dan diteruskan ke *hop* berikutnya. Dalam hal ini, paket di-*generate* secara lokal. ID paket yang unik diberikan dan akan digunakan untuk antrian paket yang tidak langsung sampai AODV-UU memutuskan sebuah aksi, dan *route discovery* dibuat. Jika paket tidak dibangkitkan (*generate*) secara lokal, dan tidak ada rute yang ditemukan, maka paket akan dibuang dan pesan RRER dikirimkan ke sumber [7].

2.3.8 Pemrosesan AODV Control Message

AODV *control message* diterima pada socket UDP (*port* 654) dan selanjutnya diproses. Tipe dari *field* AODV diperiksa, pesan dikonversi menjadi

tipe *message corresponding*, dan selanjutnya fungsi *correct handler* dijalankan [7].

2.3.9 Pengiriman AODV Control Message

Setiap AODV *control message* yang dibangkitkan oleh AODV-UU dikirimkan pada soket UDP. Ketika pesan ditangkap oleh *netfilter* untuk membangkitkan paket secara local kembali, *NF_IP_LOCAL_OUT*, diantrikan oleh *kernel* AODV dan diterima oleh modul *packet_input* dari AODV-UU melalui *libpq*. Modul *packet_input* akan mengembalikan sebuah *message* kembali *libpq*, dan paket selanjutnya akan ditangkap oleh *post-routing* dari *netfilter*, *NF_IP_POST_ROUTING*. Paket tersebut akan kembali diarahkan untuk memastikan penggunaan informasi *routing*, dan dikirim keluar oleh sistem [7].

2.4 OPENWRT

OpenWrt merupakan program *firmware* berbasis Linux untuk *embedded device* seperti *wireless router*. OpenWrt hanya menyediakan konfigurasi minimal namun dengan kemampuan untuk mendukung paket-paket fitur tambahan, ini berarti *user* memiliki kemampuan untuk memodifikasi fiturnya, menghilangkan paket yang tidak diinginkan untuk menyediakan ruang bagi paket lain, dan untuk pengembang ini berarti dapat terfokus pada paket tanpa harus menguji dan membuang keseluruhan *firmware* [8]. Pada umumnya OpenWrt menggunakan *command-line* sebagai *interface*-nya, namun juga tersedia fitur pilihan *interface* GUI berbasis *web* dan untuk mengatur paket-paket fiturnya OpenWrt menggunakan sistem *ipkg* seperti yang terdapat pada *distro* Linux Debian [9].

OpenWrt diterbitkan pada januari tahun 2004 oleh tim proyek OpenWrt yang dibentuk dalam rangka mengembangkan sebuah *third-party firmware* yang jumlahnya masih sangat terbatas pada saat itu. Sebelum OpenWrt, telah banyak orang memodifikasi *firmware* Linksys berbasis Linux untuk tujuan tertentu, sehingga menyebabkan banyaknya pembuatan *firmware* yang telah dimodifikasi [10].

Pada awalnya OpenWrt merupakan *firmware* pengganti untuk Linksys WRT54G dan sejenisnya (Broadcom BCM947xx) [8], tetapi saat ini telah dapat

digunakan pada *platform* yang berbeda seperti Netgear, ASUS, D-Link, DELL dan lain-lain. Perkembangan OpenWrt sangat dipengaruhi akan kemudahannya dalam memodifikasi fitur-fitur tambahan diluar fitur-fitur yang telah disediakan oleh pihak manufaktur agar dapat digunakan sesuai dengan keperluan tertentu dari para pengguna. Hal ini dapat terjadi karena OpenWrt bersifat *opensource* karena dibuat berdasarkan GNU *General Public License*/Linux, sehingga setiap perubahan yang dibuat oleh pihak manufaktur harus didaftarkan dan dirilis melalui lisensi GPL [9].

Berdasarkan sifat *opensource* ini pula maka para pengguna dapat dengan bebas memodifikasi ataupun menambahkan fitur-fitur lain pada *router* sesuai dengan kebutuhan. Pada perangkat lunak OpenWrt terdapat dua versi utama yang telah dikembangkan yaitu :

- 1) whiterussian, dan
- 2) kamikaze.

2. 4. 1 WhiteRussian

Ini merupakan versi awal dari OpenWrt, versi yang paling banyak digunakan karena memiliki tingkat kestabilan tinggi karena telah dikembangkan lebih lama. Selain itu versi ini juga telah banyak memiliki dokumentasi maupun tutorial yang tersedia untuk mendukung pemakaiannya. Pengembangan terakhir dari whiterussian adalah WhiteRussian 0.9 yang dirilis pada tanggal 5 Februari 2007 [11].

2. 4. 2 Kamikaze

Versi ini merupakan versi terbaru dari OpenWrt, walaupun sudah stabil namun masih dalam pengembangan. Dibuat berdasarkan desain berbeda dengan versi yang terdahulu sehingga dapat bekerja pada pilihan jenis *wireless router* yang lebih luas, selain itu mempunyai *kernel* yang lebih baru. Seri terakhir dari kamikaze adalah Kamikaze 7.09 yang dirilis pada September 2007 [11].

2.5 UoBWinAODV

UoBWinAODV Merupakan *protocol handler* AODV untuk Windows XP yang dikembangkan oleh Universitas Bremen. UoBWinAODV dibuat sesuai dengan RFC 3561 tentang protokol *routing* AODV. UoBWinAODV dibuat dalam bahasa C/C++ dengan menggunakan metode pembanding netfilter untuk mendapatkan informasi penting AODV. *Protocol handler* ini menggunakan PassThru *filter driver* yang dapat mengatur *driver* kartu jaringan secara langsung.

Melalui filter *driver* dapat dilakukan pengaturan distribusi paket dan pengumpulan informasi yang relevan. *Routing* daemon akan mengeksekusi program *user space* dan melewati IOCTL-*Systemcall* pada saat NDIS filter *driver* sedang berkomunikasi. Hal tersebut merupakan proses administrasi *routing* dari keseluruhan sistem setelah persyaratan AODV memungkinkan. NDIS filter *driver* yang digunakan berdasarkan pada yang dibuat oleh Thomas F. Divine [12].

Berdasarkan daftar bagian dari proyek yang terdapat pada RFC 3561, hal yang belum dapat di implementasikan adalah:

- 1) mencari tabel *routing* internal melalui pencocokkan *prefix* yang lama,
- 2) perbaikan lokal (*local repair*),
- 3) penjaluran subnet (*subnet routing*),
- 4) mengukur setelah *restarting*,
- 5) perbaikan dari hubungan *unidirectional*,
- 6) pemberitahuan ICMP tujuan tidak dapat dicapai (*destination unreachable*) setelah pencarian rute gagal.

BAB 3

PERANCANGAN *TESTBED* JARINGAN

Berdasarkan bab-bab sebelumnya, maka akan dibahas perancangan *wireless mesh network* dengan arsitektur tipe *hybrid wireless mesh network* yang menggunakan protokol *routing* AODV-UU pada *mesh router* dan UoBWinAODV sebagai penanganan protokol pada *mesh client*.

Dalam perancangan *wireless mesh network* ini dimulai dari tahap persiapan awal yang merupakan tahap perencanaan *testbed*, tahap pembangunannya, sampai *testbed* jaringan ini siap untuk digunakan.

3.1 PERENCANAAN *TESTBED*

Perencanaan *testbed* diperlukan dalam menentukan langkah-langkah pembangunan yang akan dikerjakan. Dalam perencanaan sistem ini akan dibagi dalam beberapa tahapan pembangunan, tahapan pembangunan ini merupakan urutan dari kegiatan pembangunan *testbed*. Gambar 3.1 dibawah ini memperlihatkan diagram alir dari tahapan pembangunannya.



Gambar 3.1. Diagram alir tahapan pembangunan *testbed*.

Tahapan-tahapan perencanaan yang dilakukan dalam pembangunan *testbed* jaringan ini adalah sebagai berikut:

1. Menentukan topologi dari jaringan.

Penentuan topologi jaringan ini dilakukan dengan merancang bentuk topologi jaringan yang akan sesuai dengan keperluan *testbed*.

2. Menentukan spesifikasi dari perangkat yang akan digunakan.

Penentuan spesifikasi dari perangkat yang digunakan dilakukan agar jaringan tersebut dapat bekerja dengan baik.

3. Melakukan instalasi *software*.

Agar konsep *wireless mesh network* yang diinginkan dapat tercapai, maka diperlukan dukungan dari beberapa perangkat lunak pada perangkat keras yang digunakan.

4. Menentukan lokasi *testbed*.

Penentuan lokasi dari *testbed* disesuaikan dengan bentuk topologi jaringan *testbed* yang diinginkan.

5. Mengkonfigurasi jaringan.

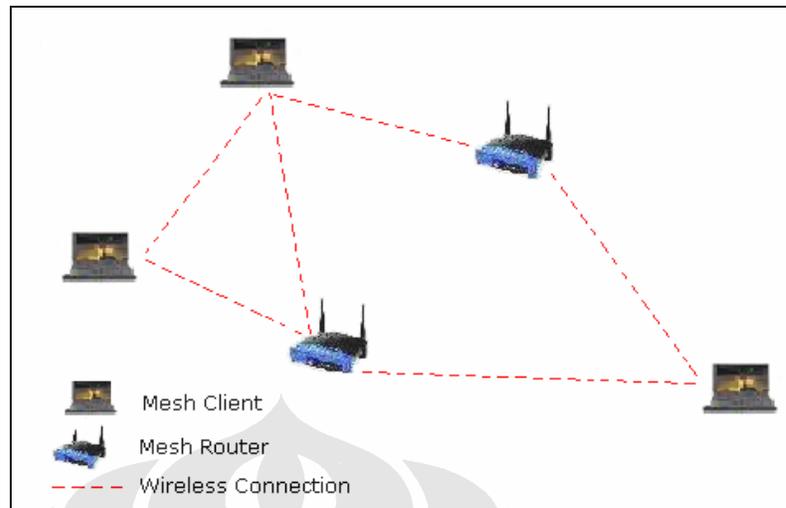
Melakukan konfigurasi jaringan seperti pengalamatan, konfigurasi parameter-parameter *wireless*, *packet forwarding rules*, konfigurasi parameter UoBWinAODV, dan sebagainya.

3.2 PEMBANGUNAN TESTBED

Pembangunan *testbed* jaringan dilakukan sesuai dengan apa yang telah direncanakan pada perencanaan *testbed* jaringan diatas, yang dilakukan dalam beberapa tahapan.

3.2.1 Penentuan Topologi Jaringan

Topologi *testbed* jaringan yang akan dibangun disesuaikan dengan konsep *wireless mesh network* dengan arsitektur tipe *hybrid wireless mesh network*. Gambar 3.2 memperlihatkan topologi *testbed* jaringan yang akan dibangun.



Gambar 3.2. Perencanaan topologi jaringan.

3. 2. 2 Menentukan Spesifikasi Perangkat Yang Digunakan

Pada *wireless mesh network* yang akan dibangun ini terdapat kebutuhan *hardware* untuk merealisasikannya. *Hardware* tersebut dibutuhkan untuk menyediakan fungsi *mesh router* dan *mesh cleint* dalam *hybrid wireless mesh network*.

3. 2. 2. 1 Perangkat *Mesh Router*

Untuk membangun arsitektur tipe *hybrid wireless mesh network* ini akan digunakan dua buah *wireless router* yang akan menjalankan fungsi *mesh router*. *Wireless router* yang akan digunakan adalah linksys model WRT54GL versi 1.1, yang memang dikhususkan untuk keperluan modifikasi dengan *platform opensource* berbasis linux dan sejenisnya, karena *firmware* bawaan dari pabrik milik linksys mempunyai keterbatasan dalam modifikasi aplikasi selain aplikasi yang telah tersedia oleh *firmware* tersebut. *Wireless router* ini telah dilengkapi dengan perlengkapan yang dibutuhkan seperti unit *power supply* dan kabel LAN ethernet standar, seperti yang terlihat pada Gambar 3.3.



Gambar 3.3. *Wireless router linksys WRT54GL versi 1.1 dan perlengkapannya.*

Spesifikasi dari *Wireless router linksys WRT54GL versi 1.1* dapat dilihat pada lampiran 1. Linksys WRT54GL versi 1.1 memerlukan suplai tegangan listrik sebesar 12V DC dengan arus sebesar 0,5 A. Kebutuhan daya ini merupakan kebutuhan *standard* untuk *embedded devices* dan *wireless access point*, sehingga menjadikannya *compatible* dengan dengan *Power over Ethernet (PoE)* [10].

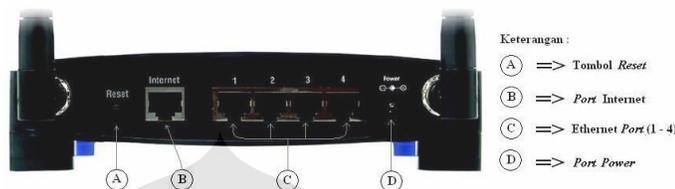
Seperti yang terlihat pada Gambar 3.4, dibagian depan *wireless router* ini terdapat tombol *secure easy setup* (dengan label “Cisco System”) bertujuan untuk memperbolehkan *user* dengan mudah mengatur enkripsi jaringan *wireless*, dan menetapkan bahwa *user* menggunakan perangkat keras dari *vendor* yang berpartisipasi dalam program *secure easy setup*. Selain itu terdapat juga beberapa lampu indikator (LED) pada *wireless router*. Lampu indikator *power* menandakan bahwa *wireless router* sedang aktif atau menerima suplai daya. Lampu DMZ menandakan bahwa *wireless router* sedang dalam proses *booting*. Lampu LED WLAN dan Ethernet (nomor 1 sampai 4) merupakan lampu yang menandakan bahwa *port* tersebut sedang digunakan (aktif), dan lampu LED internet menandakan adanya koneksi ke *port* WAN [10].



Gambar 3.4. *Tampak depan wireless router WRT54GL versi 1.1.*

Pada bagian belakang terdapat tombol *reset*, tombol ini adalah tombol yang *default*-nya akan me-*reset wireless router* ke *setting* awal dari pabriknya. *Port*

ethernet (1 – 4) digunakan untuk menghubungkan *router* dengan jaringan LAN yang ada dan dengan perangkat jaringan lainnya. *Port power* yang merupakan *input* untuk suplai daya juga terdapat pada bagian belakang ini, serta terdapat *port* internet yang dapat digunakan untuk menghubungkan ke internet. Hal ini dapat dilihat pada Gambar 3.5.



Gambar 3.5. Tampak belakang *wireless router* WRT54GL versi 1.1.

3. 2. 2. 2 Perangkat *Mesh Client*

Perangkat *mesh client* yang ideal untuk digunakan dalam *testbed* ini adalah *notebook/laptop* (lihat Gambar 3.6). Penggunaan laptop sebagai *mesh client* adalah agar dapat memperlihatkan fungsi sebagai *host* dan *router* yang dapat melakukan fungsi *routing* dan konfigurasi serta menyediakan aplikasi *end-user* pada pengguna jaringan. Spesifikasi minimal yang harus dimiliki oleh perangkat *mesh client* adalah sebagai berikut:

- 1) OS (*Operating System*) Windows XP,
- 2) Processor Pentium III,
- 3) Memory 256 Mb,
- 4) LAN *card* (port koneksi ethernet),
- 5) WLAN *card* / *Wireless modem*.

Laptop yang digunakan harus dilengkapi dengan kartu jaringan (LAN *card*) agar dapat dihubungkan dengan *wireless router* untuk masuk ke dalam *firmware*-nya, sedangkan *Wireless modem* dibutuhkan agar konsep *user* dapat terhubung dengan jaringan kapan saja dan dimana saja dapat terpenuhi.



Gambar 3.6. Laptop sebagai perangkat *mesh client*.

3. 2. 3 Instalasi Software

Instalasi *software* dilakukan pada perangkat *wireless router* dan pada laptop. Instalasi *wireless router* menggunakan *third party firmware* OpenWrt dan paket AODV-UU, sedangkan instalasi pada laptop menggunakan *software* UoBWinAODV. Instalasi *software* diperlukan agar konsep *wireless mesh network* yang akan dibangun dapat tercapai.

3. 2. 3. 1 Instalasi OpenWrt

Jenis *firmware* OpenWrt yang diinstall adalah WhiteRussian 0.9 yang dapat di *download* pada: <http://downloads.openwrt.org/whiterussian/0.9/default>. Versi ini digunakan karena merupakan versi OpenWrt yang paling stabil dan memiliki dokumentasi yang paling lengkap. Paket `openwrt-wrt54g-squashfs.bin` adalah paket OpenWrt WhiteRussian 0.9 yang akan diinstall pada *wireless router* linksys WRT54GL versi 1.1.

Instalasi dilakukan dengan menggunakan fasilitas *upgrade firmware* dari *firmware* melalui *web interface* bawaan linksys pada WRT54GL versi 1.1 dengan alamat *default*-nya yaitu 192.168.1.1 [13], seperti yang terlihat pada Gambar 3.7.

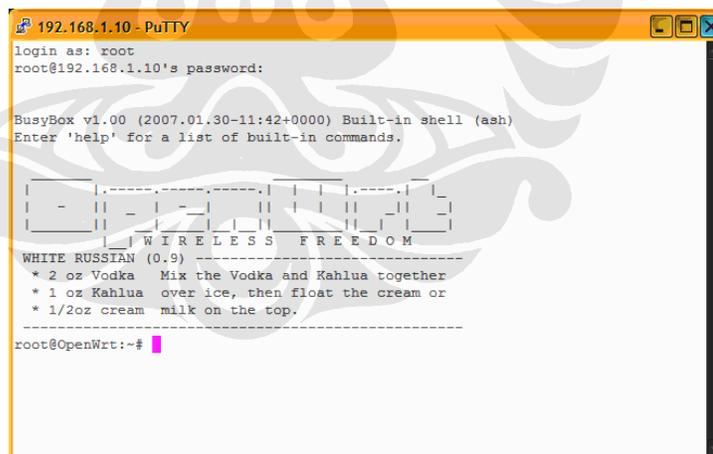


Gambar 3.7. Jendela *firmware upgrade* dari linksys WRT54GL versi 1.1.

Apabila *firmware* OpenWrt telah terinstall, maka dapat dimasuki melalui *web interface* atau Telnet. Gambar 3.8 memperlihatkan tampilan *web interface* OpenWrt dan Gambar 3.9 memperlihatkan jendela *command line* dari OpenWrt.



Gambar 3.8. Halaman depan *web interface* OpenWrt.

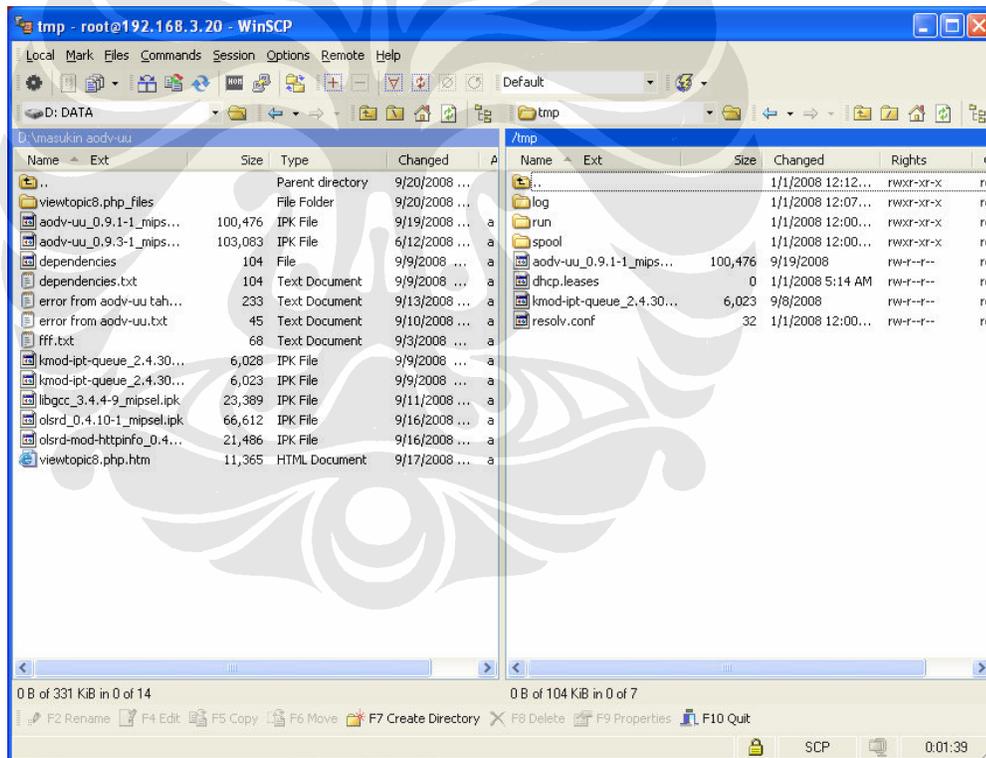


Gambar 3.9. Halaman depan *command line* OpenWrt.

3. 2. 3. 2 Instalasi Paket AODV-UU

Pada OpenWrt dapat digunakan paket-paket tertentu sesuai dengan kebutuhan *user*, salah satunya adalah paket protokol *routing* AODV-UU. Paket ini dapat digunakan pada OpenWrt untuk membangun sebuah jaringan *mesh*.

Paket AODV-UU yang telah disediakan untuk OpenWrt dapat di download pada: <https://www.comnets.uni-bremen.de>. Paket yang akan di install adalah aodv-uu 0.9.1-1. Instalasi yang dilakukan adalah dengan memasukkan paket AODV-UU kedalam *wireless router* menggunakan program WinSCP (Windows *Secure Copier*). Gambar 3.10 memperlihatkan tampilan program WinSCP yang digunakan untuk memasukkan paket AODV-UU. Selain itu, AODV-UU juga memerlukan beberapa paket pendukung dalam membangun jaringan *mesh*. Paket pendukung tersebut antara lain paket untuk netfilter (*kmod-ipt-queue*), dan paket untuk mengaktifkan SNMP (*Simple Network Monitoring Protocol*). Gambar 3.10 memperlihatkan tampilan program WinSCP yang digunakan untuk memasukkan paket AODV-UU dan paket pendukungnya.



Gambar 3.10. Tampilan program WinSCP.

Setelah paket AODV-UU dan paket-paket pendukungnya telah berada dalam *wireless router*, maka baru dapat diinstall dengan menjalankan perintah pada *command line* OpenWrt. Perintah tersebut adalah sebagai berikut:

ipkg install kmod-ipt-queue_2.4.30-brcm-5_mipsel.ipk

ipkg install aodv-uu_0.9.1-1_mipsel.ipk

ipkg install snmpd_mipsel.ipk

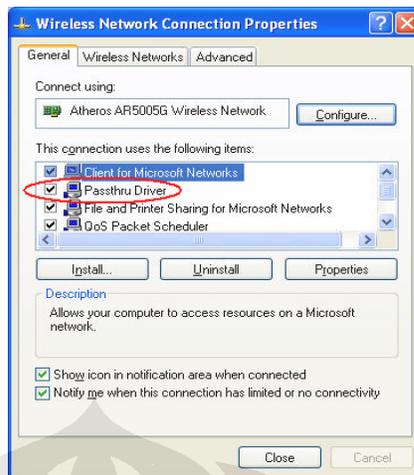
3. 2. 3. 3 Instalasi UoBWinAODV

Pada arsitektur *hybrid wireless mesh network* setiap *mesh client* selain bertindak sebagai *host* juga bertindak sebagai *router* yang dapat melakukan fungsi *routing* dan konfigurasi. Agar *mesh client* dapat bekerja sesuai fungsinya, maka pada laptop digunakan UoBWinAODV. Versi yang digunakan adalah UoBWinAODV versi 0.15.

Instalasi yang dilakukan adalah dengan menginstalasi sebuah *driver* dengan nama *passthru*. Berikut adalah langkah-langkah yang dilakukan:

- 1) buka jendela *network connections*,
- 2) pilih *interface* yang akan digunakan dalam hal ini dipilih *Wireless Network Connections* (karena menggunakan AODV *wireless interface*) lalu masuk ke dalam *properties*,
- 3) pilih *install*, kemudian pilih *service*, lalu pilih *Have Disk*,
- 4) pilih *browse* dan masuk kedalam direktori *Installabe* didalam direktori UoB-WinAODV lalu pilih *open* dan kemudian pilih *OK*,
- 5) selanjutnya pilih *Passthru Driver* dan kemudian pilih *OK*.
- 6) Dan terakhir tutup jendela *properties* dengan memilih *close*.

Setelah langkah-langkah diatas dilakukan maka UoBWinAODV telah terinstalasi pada *wireless interface*, yang ditandai dengan adanya *Passthru Driver* pada jendela *Wireless Network Connections Properties*. Gambar 3.11 memperlihatkan tampilan pada jendela *Wireless Network Connections Properties* yang telah memiliki *Passthru Driver*.



Gambar 3.11. *Passthru driver pada wireless network connections.*

3. 2. 4 Penentuan Lokasi *Testbed*

Dalam rangka membangun *testbed* yang sesuai dengan kondisi jaringan *mesh*, lokasi penempatan *mesh router* dan *mesh client* merupakan faktor yang menentukan. Oleh karena itu dalam penentuan lokasi dan penempatan perangkat yang digunakan didasarkan atas beberapa pertimbangan yang dinilai dapat menghasilkan kondisi sesuai dengan apa yang diinginkan, pertimbangan tersebut meliputi:

1. Pertimbangan teknis.

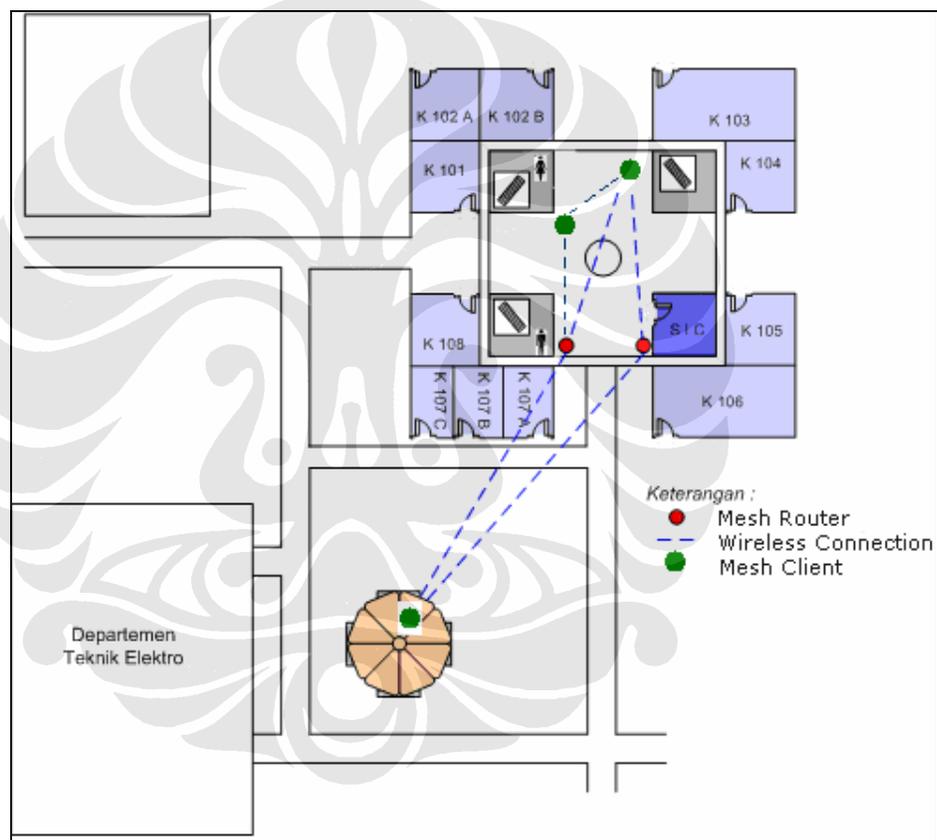
Yang merupakan pertimbangan teknis dalam penentuan lokasi dan penempatan perangkat adalah ketersediaan ruang yang cukup untuk memenuhi bentuk dari topologi jaringan *testbed* yang akan dibangun, ketersediaan sumber listrik (karena perangkat *mesh router* memerlukan sumber listrik untuk penggunaannya), dan lokasi tersebut mudah dijangkau untuk keperluan pengetesan dan konfigurasi.

2. Pertimbangan non teknis.

Pertimbangan non teknis mengacu pada pertimbangan keamanan baik yang berasal dari faktor manusia ataupun faktor alam. Faktor yang berasal dari manusia seperti keamanan penempatan perangkat dari hilir mudik manusia, dan pengawasan perangkat apabila telah terpasang. Sedangkan faktor dari alam meliputi perlindungan terhadap cuaca, seperti hujan dan gangguan petir.

Dari pertimbangan-pertimbangan tersebut maka lokasi yang dinilai cocok sebagai lokasi dalam membangun jaringan *testbed* ini adalah lobi Fakultas Teknik Universitas Indonesia.

Dalam proyek ini *mesh router* dan *mesh client* ditempatkan pada beberapa posisi yang berbeda dengan jarak tertentu, agar didapatkan fungsi *mesh* dan *multi-hop* dari jaringan. Lokasi penempatan pada lobi Fakultas Teknik Universitas Indonesia dapat dilihat pada Gambar 3.12 dan lokasi penempatan dari perangkat-perangkat yang digunakan dapat dilihat pada Gambar 3.13.



Gambar 3.12. Denah lokasi penempatan.



Gambar 3.13. Lokasi penempatan *mesh router* dan *mesh client*.

3. 2. 5 Konfigurasi Jaringan

Setelah melakukan proses instalasi *firmware* OpenWrt dan paket *aodv-uu*, tahap berikutnya adalah mengkonfigurasi jaringan dan *aodv-uu* agar dapat beroperasi sesuai dengan ketentuan *routing protocol* AODV. Langkah-langkah yang dilakukan adalah sebagai berikut ini:

- 1) memberikan pengalamatan pada setiap *mesh router* dan *mesh client*,
- 2) melakukan konfigurasi parameter-parameter *wireless router* agar berjalan sebagai jaringan *mesh*,
- 3) memberikan *packet forwarding rules* pada *mesh router*,
- 4) konfigurasi pada UoBWinAODV,
- 5) menjalankan AODV-UU dan UoBWinAODV.

3. 2. 5. 1 Pengalamatan

Wireless router Linksys WRT54GL secara *default* mempunyai alamat 192.168.1.1 pada *interface* br0, dimana *interface* br0 bersifat *bridge* sehingga alamat untuk LAN dan *wireless LAN* menggunakan satu alamat yang sama. Setiap *wireless router* yang digunakan akan dipisahkan antara alamat LAN *interface* dengan alamat *wireless interface*, karena *wireless mesh network* bersifat

ad-hoc. Adapun pengalamatan yang dilakukan untuk setiap *wireless router* adalah sebagai berikut:

Alamat untuk *interface* LAN:

nvrnm set lan_ifname=vlan0 (memberi nama interface lan = vlan0)
 nvrnm set lan_ipaddr=192.168.3.10 (memberi alamat lan =192.168.3.10)
 nvrnm set lan_netmask=255.255.255.0 (memberi netmask lan =255.255.255.0)
 nvrnm set lan_proto=static (menon-aktifkan DHCP)

Alamat untuk *interface wireless*:

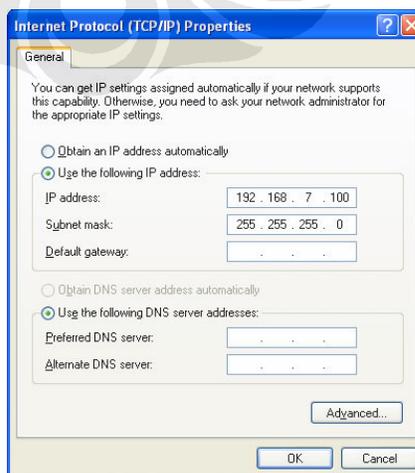
nvrnm set wifi_ifname=eth1 (memberi nama interface wifi = eth1)
 nvrnm set wifi_ipaddr=192.168.7.10 (memberi alamat wifi =192.168.7.10)
 nvrnm set wifi_netmask=255.255.255.0 (memberi netmask wifi =255.255.255.0)
 nvrnm set wifi_proto=static (menon-aktifkan DHCP)

Pengalamatan pada *mesh client* menggunakan pengalamatan *static*, dengan memasukkan secara langsung melalui jendela *internet protocol (TCP/IP) properties*. Pengalamatan yang dimasukkan adalah sebagai berikut:

Alamat IP: 192.168.7.100

Subnet mask: 255.255.255.0

Gambar 3.14. memperlihatkan pengalamatan *static* yang dilakukan melalui jendela *internet protocol (TCP/IP) properties* pada *mesh client*.



Gambar 3.14. Pengalamatan pada *mesh client*.

3. 2. 5. 2 Konfigurasi Parameter *Wireless* Pada *Mesh Router*

Agar dapat bekerja dalam kondisi *ad-hoc* parameter *wireless* pada *router* harus diatur dengan menjalankan perintah-perintah berikut ini:

```

nvrn set wl0_mode=sta           (beroperasi sebagai client mode )
nvrn set wl0_infra=0           (0 = ad-hoc mode, 1 = normal AP)
nvrn set wl0_ssid=AODV-Hybrid (memberi nama SSID)
nvrn set wl0_radio=1          (mengaktifkan koneksi radio wireless)
nvrn set wl0_channel=11       (menentukan channel yang dipakai 0 – 11)
nvrn set wl0_closed=0        (0 = broadcast SSID, 1 = hide SSID)
nvrn commit                   (menuliskan perubahan pada nvrn)
ifup lan                      (eksekusi perintah konfigurasi pada lan)
ifup wifi                     (eksekusi perintah konfigurasi pada wifi)
reboot                        (me-reboot router)

```

3. 2. 5. 3 *Packet Forwarding Rules* Pada *Mesh Router*

Untuk dapat meneruskan paket data yang dikirimkan melalui jaringan *mesh* melalui *interface* dengan jenis yang berbeda seperti *interface* LAN dan *wireless* LAN maka perlu dilakukan pengaturan *forwarding rules*.

Untuk memberikan *forwarding rules* pada setiap *mesh router* dapat dilakukan melalui *file* *firewall.user* yang terdapat pada direktori */etc/firewall.user*. Konfigurasi yang diberikan untuk mengatur *forwarding rules* antar *interface-interface* yang berbeda-beda seperti dari LAN ke wifi, WAN (internet) ke wifi, hubungan *interface* antar wifi *client* maupun hubungan *interface* antar LAN *client*. Konfigurasi *forwarding rules* untuk proyek ini pada *firewall.user* adalah sebagai berikut:

```

#!/bin/sh
. /etc/functions.sh

WAN=$(nvrn get wan_ifname)
LAN=$(nvrn get lan_ifname)
WIFI=$(nvrn get wifi_ifname)

iptables -F input_rule
iptables -F output_rule
iptables -F forwarding_rule

```

```
iptables -t nat -F prerouting_rule
iptables -t nat -F postrouting_rule
```

```
# For forwarding WAN (internet) to WIFI
iptables -A forwarding_rule -i $WIFI -o $WAN -j ACCEPT
#For forwarding LAN & WIFI in nodes
iptables -A forwarding_rule -i $LAN -o $WIFI -j ACCEPT
#For WIFI clients to connect to node
iptables -A forwarding_rule -i $WIFI -o $WIFI -j ACCEPT
#For connecting a Wired Lan client of node 1 to wired client of node 2
iptables -A forwarding_rule -i $LAN -o $LAN -j ACCEPT
```

3. 2. 5. 4 Konfigurasi Pada UoBWinAODV

Agar dapat bekerja dengan benar, pada UoBWinAODV harus dilakukan beberapa konfigurasi antara lain merubah parameter *IfaceName* dan *IP address* pada file konfigurasinya, dan menjalankan *wireless interface* pada mode *ad-hoc*.

1. Parameter IfaceName.

File konfigurasi terletak pada direktori *Installable* dengan nama *UoBWinAODV.cfg*. Isi parameter *IfaceName* yang dirubah disesuaikan dengan *interface* yang digunakan untuk AODV. *Interface* yang aktif dapat dilihat dengan menjalankan perintah berikut pada *command line*:

```
Factotum.exe /enum
```

Setelah menjalankan perintah tersebut maka akan menampilkan *interface* yang aktif pada *mesh client*, seperti yang terlihat pada Gambar 3.15. Dalam hal ini *interface* yang digunakan adalah *wireless interface*, isi baris kedua adalah yang harus diletakkan pada parameter *IfaceName* di file konfigurasi.

```
C:\WINDOWS\system32\cmd.exe
Copyright (c) 2004 ComNets, University of Bremen
Copyright (c) 2003 Printing Communications Assoc., Inc. (PCUSA)
Copyright (c) 1982, 1986, 1993 The Regents of the University of California.
All rights reserved.

Driver Bindings:
"\"Device\{327909CD-7F14-4CE3-BF6F-FD08D34C581E}\""
"\"Device\{2AE15F1A-1004-4FAA-898C-B48AF47B5CC3}\""
Description: "Athlon AR5005C Wireless Network Adapter"
Medium: 802.3; Mac address = 00-16-E3-47-F5-74

"\"Device\{042638F7-7D4D-49B3-BB5F-E960835D426D}\""
"\"DEVICE\NDISWANIP\""
Description: " NdisWan Adapter"
Medium: 802.3; Mac address = 5A-99-20-52-41-53

"\"Device\{913BB413-1BC6-4FB1-A1C2-505E74C20BC5}\""
"\"DEVICE\{181E22C3-BE0A-4920-8736-2D7DDFBA4433}\""
Description: " Realtek RTL8139/810x Family Fast Ethernet NIC
"
Medium: 802.3; Mac address = 00-16-36-51-0C-5C

C:\Documents and Settings\SMPN 146 JAKARTA\My Documents\UoBWinAODU-0.15\UoBWinAODU-0.15\Installable>
```

Gambar 3.15. *Interface* yang aktif pada *mesh client*.

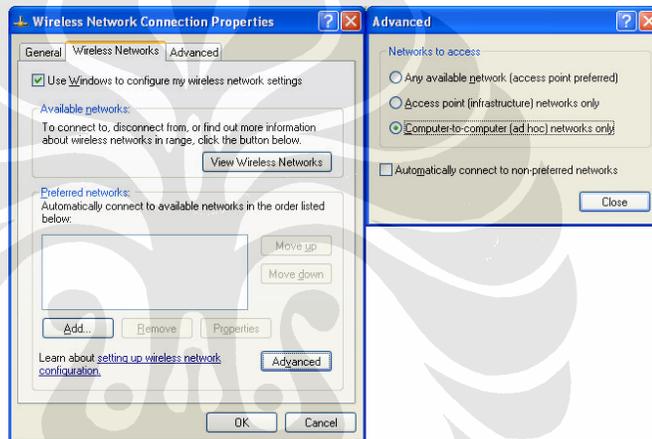
2. IP address.

Alamat IP yang berada pada file konfigurasi disesuaikan dengan alamat IP yang telah diberikan melalui jendela *internet protocol (TCP/IP) properties*, misal:

Alamat IP: 192.168.7.100

3. Menjalankan *wireless interface* pada mode *ad-hoc*.

Untuk menjalankan *wireless interface* pada mode *ad-hoc* dapat dilakukan melalui jendela *Wireless Network Connections Properties* pada tab *Wireless Networks*. Lalu memilih *advance* dan kemudian memilih *computer-to-computer (ad hoc) networks only*. Seperti yang diperlihatkan pada Gambar 3.16



Gambar 3.16. Menjalankan *wireless interface* pada mode *ad-hoc*.

3. 2. 5. 5 Menjalankan AODV-UU Dan UoBWinAODV.

Untuk menjalankan AODV-UU ataupun UoBWinAODV harus diberikan perintah-perintah untuk menjalankannya, karena keduanya tidak dapat berjalan secara otomatis.

1. Menjalankan AODV-UU.

AODV-UU merupakan protokol *routing* yang berada pada *mesh router*, dan agar dapat dijalankan sebagai jaringan *mesh* diperlukan perintah-perintah yang harus diberikan melalui *command line* OpenWrt. Perintah yang diberikan adalah sebagai berikut:

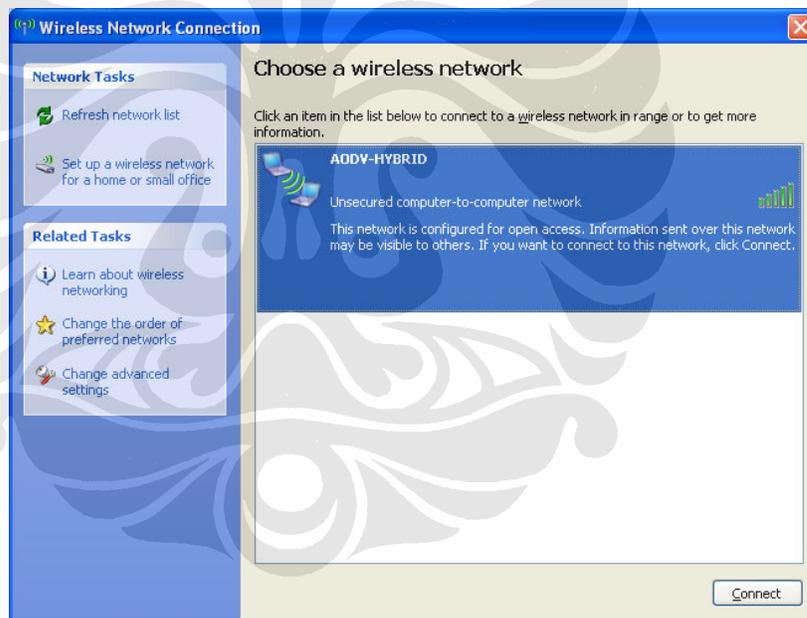
<code>aodvd -d</code>	(menjalankan aodv di <i>background</i>)
<code>insmod ip_queue</code>	(memasukkan modul kernel queue)
<code>insmod kaodv</code>	(memasukkan modul kernel kaodv)

2. Menjalankan UoBWinAODV.

Untuk menjalankan UoBWinAODV dapat dilakukan dengan memberikan perintah melalui *command line* pada Windows. Agar perintahnya dapat berjalan, pada *command line* Windows harus masuk ke dalam direktori *Installable* berada. Perintah yang diberikan untuk menjalankan UoBWinAODV adalah sebagai berikut:

```
UoBWinAODV.exe /config UoBWinAODV.cfg
```

Setelah semua langkah konfigurasi telah selesai dilakukan maka *wireless mesh network* dengan arsitektur *hybrid wireless mesh network* ini pun telah siap untuk digunakan dan pada jendela *Wireless Network Connection* jaringannya pun sudah siap di akses, seperti yang terlihat pada Gambar 3.17.



Gambar 3.17. Jendela *wireless network connection*.

BAB 4

PENGUJIAN DAN ANALISA

4.1 HASIL RANCANGAN TESTBED

Perancangan *testbed* jaringan yang dibangun berdasarkan topologi perencanaan dan penentuan lokasi yang digunakan untuk menguji kinerja dari *wireless mesh network* dengan menggunakan arsitektur tipe *hybrid wireless mesh network* dapat diperlihatkan pada gambar 4.1. *Testbed* jaringan ini terdiri tiga buah *mesh client* dan dua buah *mesh router* yang ditempatkan pada posisi tertentu agar sesuai dengan topologi jaringan yang dikehendaki. Setiap perangkat diberikan pengalamatan yang berbeda dengan keterangan sebagai berikut:

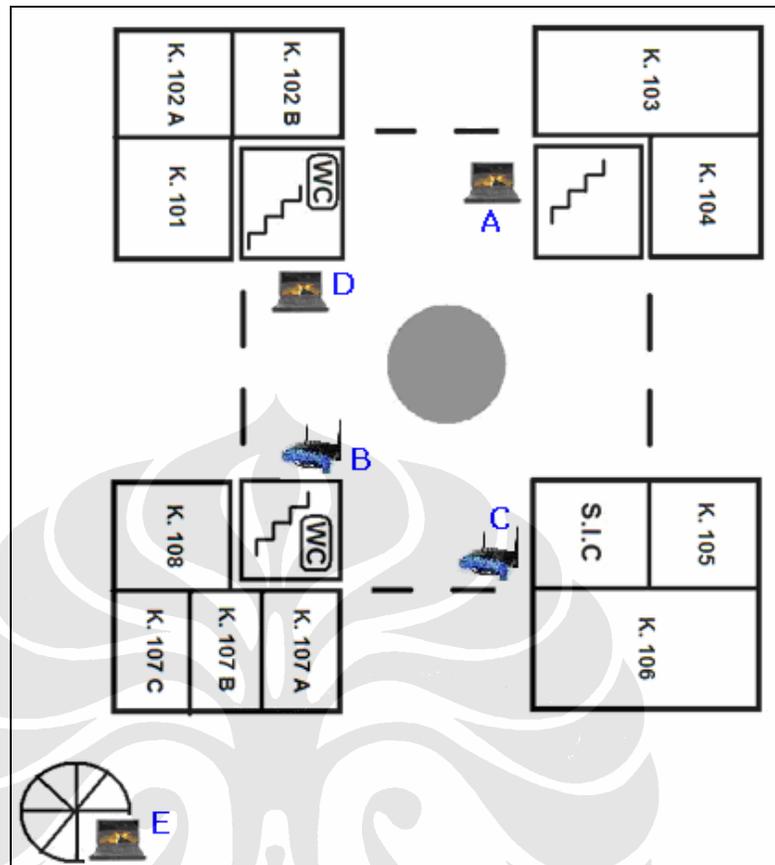
Mesh Client A: alamat IP = 192.168.7.100
 subnet mask = 255.255.255.0

Mesh Router B: alamat IP = 192.168.7.20
 subnet mask = 255.255.255.0

Mesh Router C: alamat IP = 192.168.7.10
 subnet mask = 255.255.255.0

Mesh Client D: alamat IP = 192.168.7.70
 subnet mask = 255.255.255.0

Mesh Client E: alamat IP = 192.168.7.50
 subnet mask = 255.255.255.0



Gambar 4.1. *Testbed yang dibangun untuk wireless mesh network.*

4. 2 PENGUJIAN TESTBED JARINGAN

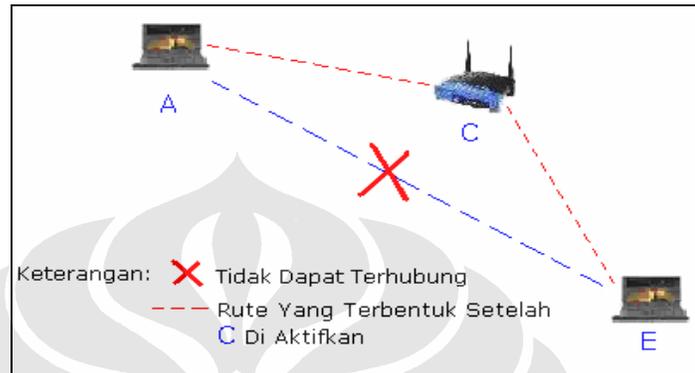
Testbed jaringan yang telah dibuat diberikan beberapa skenario pengujian. Skenario pengujian tersebut diharapkan dapat memberikan gambaran kinerja yang bagi jaringan *testbed* tersebut. Skenario pengujian tersebut antara lain:

- 1) pengujian untuk melihat kinerja *self-configure* dan *self-healing* pada *wireless mesh network*,
- 2) pengujian untuk melihat pemakaian *bandwidth*, *troughtput*, *latency* dan *jitter* dari *wireless mesh network*.

4. 2. 1 Pengujian *Self-configure*

Skenario pengujian *self-configure* bertujuan untuk mendapatkan data lamanya waktu yang diperlukan suatu *node* dalam mengkonfigurasi dirinya sendiri untuk dapat terhubung dengan jaringan yang ada. Skenario pengujian ini

dilakukan dengan menggunakan sebuah *mesh router* yang untuk menghubungkan dua buah *mesh client* yang ingin saling terhubung, namun kedua *mesh client* tersebut tidak berada pada jangkauan areanya. Skenario pengujian ini diperlihatkan pada gambar 4.2.



Gambar 4.2. Skenario pengujian *self-configure*.

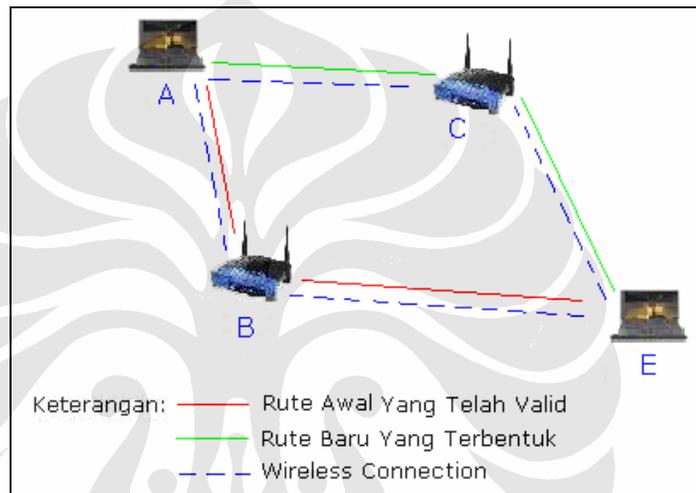
Untuk melakukan pengujian *self-configure* dilakukan langkah-langkah berikut:

- 1) *mesh client* A diletakkan diluar jangkauan *mesh client* E,
- 2) pada *mesh client* A dilakukan ping secara terus menerus untuk memeriksa hubungan dengan *mesh client* E,
- 3) kondisi awal *mesh router* C dalam keadaan tidak aktif (*off*),
- 4) kemudian *mesh router* C dinyalakan sehingga *mesh client* A dapat terhubung dengan *mesh router* E melalui *mesh router* C,
- 5) kemampuan *mesh client* A untuk mengkonfigurasi inilah yang dikatakan sebagai kemampuan *self-configure* dan hubungan yang terjadi menggunakan sistem *multihop*,
- 6) lamanya waktu yang diperlukan *mesh client* A untuk terhubung dengan *mesh client* E adalah data yang diambil.

Langkah tersebut dilakukan berulang-ulang dengan mengubah parameter HelloInterval dengan nilai: 100, 500, dan 1000 pada file konfigurasi UoBWinAODV. Data hasil pengujian *self-configure* dapat dilihat pada lampiran 2.

4. 2. 2 Pengujian *self-healing*

Skenario pengujian *self-healing* bertujuan untuk mendapatkan data lamanya waktu yang diperlukan jaringan dalam pencarian rute yang baru apabila rute yang lama sudah tidak aktif lagi atau rusak. Skenario pengujian ini dilakukan dengan menggunakan dua buah *mesh router*, dimana salah satunya telah memiliki rute yang *valid* untuk menghubungkan dua buah *mesh client* dengan menggunakan sistem *multihop*. Skenario pengujian *self-healing* ini diperlihatkan pada gambar 4.3.



Gambar 4.3. Skenario pengujian *self-healing*.

Untuk melakukan pengujian *self-healing* dilakukan langkah-langkah berikut:

- 1) *mesh client* A diletakkan diluar jangkauan *mesh client* E,
- 2) pada *mesh client* A dilakukan ping secara terus menerus untuk memeriksa hubungan dengan *mesh client* E,
- 3) rute awal yang *valid* didapatkan *mesh client* A untuk terhubung dengan *mesh client* E adalah melalui *mesh router* B,
- 4) *mesh router* B dimatikan untuk memberikan simulasi bahwa telah terjadi kerusakan pada *mesh router* B sehingga rute melalui *router* B tidak dapat dilalui yang menyebabkan ping akan mengeluarkan pesan *request time out*,
- 5) kemudian *mesh router* C dinyalakan sehingga *mesh client* A dapat terhubung dengan *mesh router* E melalui *mesh router* C dan memperoleh rute *valid* yang baru,

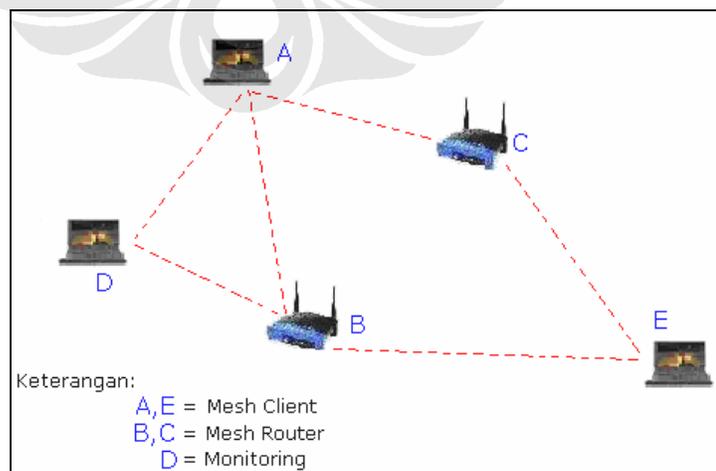
- 6) kemampuan *mesh client* A untuk mencari rute yang baru ini dilakukan secara otomatis oleh protokol AODV dan kemampuan ini disebut dengan *self-healing*,
- 7) lamanya waktu yang diperlukan *mesh client* A untuk mendapatkan rute yang baru agar dapat terhubung dengan *mesh client* E adalah data yang diambil.

Langkah tersebut dilakukan berulang-ulang dengan mengubah parameter HelloInterval pada file konfigurasi UoBWinAODV pada *mesh client*. Data hasil pengujian *self-healing* dapat dilihat pada lampiran 2.

4. 2. 3 Pengujian Pemakaian *Bandwidth*

Dalam skenario pengujian pemakaian *bandwidth* ini bertujuan untuk mendapatkan informasi tentang pemakaian *bandwidth* pada setiap perangkat *mesh router* dan *mesh client* yang digunakan. Untuk mendapatkan data pada pengujian ini dijalankan sebuah aplikasi *chatting* dan *audio-video conference* dan untuk memonitoring jaringan digunakan aplikasi PRTG.

Pengujian ini dilakukan dengan durasi waktu selama 1 jam dimana setelah 10 menit pertama jaringan diberikan beban berupa aplikasi *chatting*, memasuki menit ke 20 dijalankan aplikasi *voip/audio conference*, dan pada saat memasuki menit ke 40 dijalankan aplikasi *video conference*. Gambaran dari skenario pengujian pemakaian *bandwidth* pada jaringan ini diperlihatkan pada Gambar 4.4.



Gambar 4.4. Skenario pengujian pemakaian *bandwidth*.

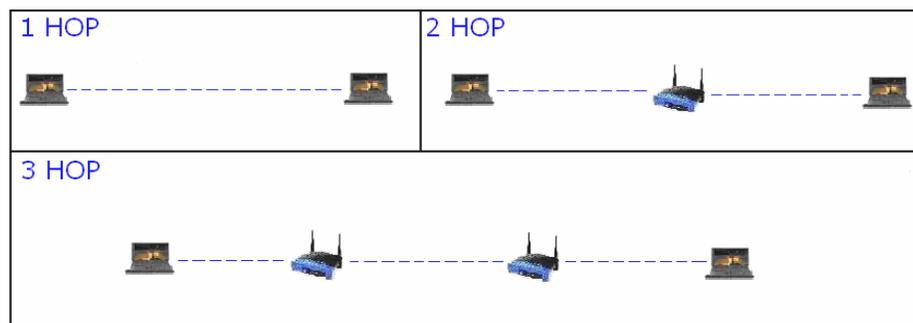
Untuk melakukan pengujian pemakaian *bandwidth* dilakukan langkah-langkah berikut:

- 1) pada *mesh client* A dilakukan ping secara terus menerus untuk memeriksa hubungan dengan *mesh client* E,
- 2) *mesh client* D digunakan untuk memonitoring jaringan dengan menjalankan aplikasi PRTG sehingga data pemakaian *bandwidth* dan *latency* dari setiap *mesh router* dan *mesh client* diperoleh,
- 3) setelah 10 menit pertama jaringan diberikan beban berupa aplikasi *chatting*,
- 4) memasuki menit ke 20 dijalankan aplikasi *audio conference*,
- 5) dan pada saat memasuki menit ke 40 dijalankan aplikasi *video conference*.

Grafik hasil pengujian pemakaian *bandwidth* untuk setiap *mesh router* dan *mesh client* dapat dilihat pada lampiran 3 dan hasil pengamatan *latency* untuk setiap *mesh router* dan *mesh client* dapat dilihat pada lampiran 4.

4. 2. 4 Pengujian Sistem *Multihop*

Skenario pengujian sistem *multihop* ini bertujuan untuk mendapatkan informasi tentang kinerja sistem *multihop* pada *wireless mesh network*. Pada skenario ini pengujian dibagi dalam tiga pengujian yang berbeda. Pengujian tersebut antara lain pengujian dengan 1 *hop*, pengujian dengan 2 *hop*, dan pengujian dengan 3 *hop*. Adapun parameter-parameter yang akan diamati pada pengujian ini adalah *throughput* dan *jitter* dari sistem *multihop*. Gambaran dari skenario pengujian sistem *multihop* pada jaringan ini diperlihatkan pada Gambar 4.5.



Gambar 4.5. Skenario pengujian kinerja sistem *multihop*.

Pengujian ini dilakukan pada saat aplikasi tidak dijalankan dan pada saat aplikasi dijalankan. Untuk melakukan pengujian kinerja sistem *multihop* dilakukan langkah-langkah dibawah ini (bagian ketiga dilakukan jika akan melakukan pengujian dengan aplikasi):

- 1) perangkat yang digunakan pada pengujian sesuai dengan yang terlihat pada gambar skenario pengujian untuk masing-masing jumlah *hop*,
- 2) menjalankan program *jperf* untuk mendapatkan data *throughput* dan *jitter* dari sistem *multihop*,
- 3) memberikan beban pada jaringan dengan menjalankan aplikasi *audio-video conference*.

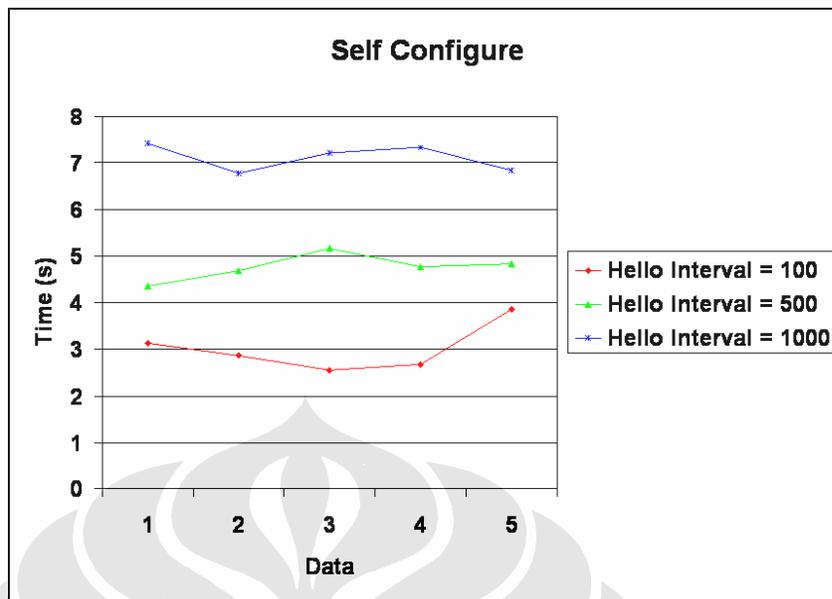
Grafik *throughput* hasil pengujian dari kinerja sistem *multihop* untuk setiap skenario dengan jumlah *hop* yang berbeda dapat dilihat pada lampiran 5, dan grafik *jitter* hasil pengujian dari kinerja sistem *multihop* untuk setiap skenario dengan jumlah *hop* yang berbeda dapat dilihat pada lampiran 6.

4.3 ANALISA SISTEM

Pada bagian ini hasil yang didapat dari skenario pengujian yang telah dilakukan akan dianalisa. Analisa dilakukan untuk melihat performansi dari *wireless mesh network*.

4.3.1 Analisa Performansi Self Configure

Dari data hasil pengujian *self configure* yang terlihat pada lampiran 2 dapat dibuat sebuah grafik perbandingan performansi *self configure*. Grafik tersebut dapat dilihat pada Gambar 4.6.

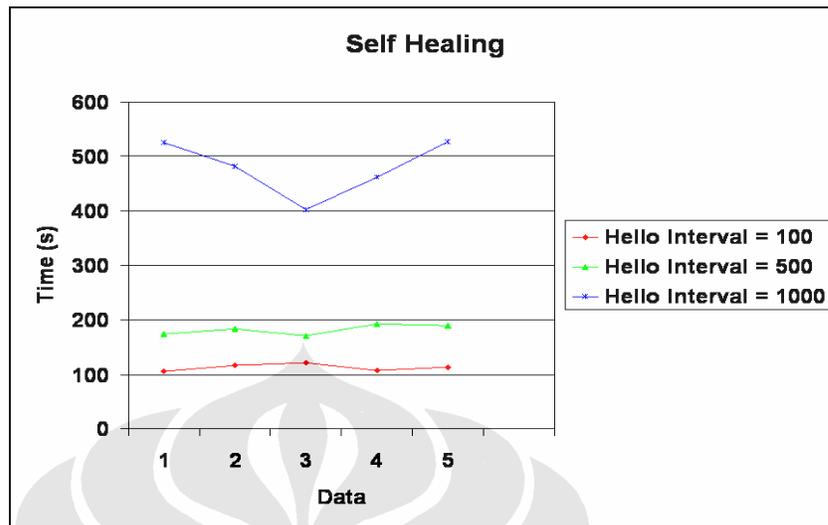


Gambar 4.6. Grafik perbandingan performansi *self configure*.

Dari grafik diatas dapat diketahui bahwa parameter *Hello Interval* dapat memberikan pengaruh terhadap performansi *self configure* pada jaringan *mesh*. Semakin besar nilai *Hello Interval* yang diberikan akan memperbesar waktu yang diperlukan sebuah *node* dalam melakukan *self configure*. Hal ini disebabkan karena *Hello Interval* merupakan *interval* waktu yang diperlukan oleh *Hello message* untuk memeriksa informasi konektifitas dengan *node* tetangganya. *Hello message* adalah paket RREP sehingga semakin besar nilai yang diberikan pada *Hello Interval* akan mempengaruhi kecepatan informasi yang diperlukan dalam proses pembentukan tabel *routing*.

4. 3. 2 Analisa Performansi *Self Healing*

Dari data hasil pengujian *self healing* yang terlihat pada lampiran 2 dapat dibuat sebuah grafik perbandingan performansi *self healing*. Grafik tersebut dapat dilihat pada Gambar 4.7.



Gambar 4.7. Grafik perbandingan performansi *self healing*.

Hello Interval juga memberikan pengaruh terhadap performansi *self healing* pada jaringan *mesh*. Semakin besar nilai *Hello Interval* yang diberikan akan memperbesar waktu yang diperlukan sebuah *node* dalam melakukan *self healing* apabila rute awal yang tersedia tidak dapat digunakan atau rusak, hal ini dapat dilihat pada grafik diatas dimana nilai *Hello Interval* memberikan pengaruh yang cukup besar.

Pada saat sebuah *node* ingin terhubung dengan *node* lain yang ada didalam jaringannya maka *node* tersebut akan mencari rute yang memiliki jumlah *hop* terkecil dan kemudian rute tersebut disimpan pada table *routing*. Jika sebuah rute tidak dapat digunakan maka *node* tersebut akan mencari rute lain untuk mencapai tujuannya, pencarian rute lain inilah yang disebut dengan *self healing*.

Waktu yang diperlukan saat sebuah *node* melakukan *self healing* lebih lama dibandingkan saat sebuah *node* melakukan *self cofigure*, karena pada saat sebuah *node* melakukan *self healing* informasi rute awal ketujuan yang tersimpan di tabel *routing* akan diperiksa terlebih dahulu. Pemeriksaan ini dilakukan oleh paket *Hello message*, dan jika statusnya tidak aktif atau rusak maka jalur tersebut akan dihapus. Selanjutnya *node* tersebut akan mencari rute lain yang akan digunakan untuk dapat terhubung dengan *node* tujuan, proses ini juga menggunakan *Hello*

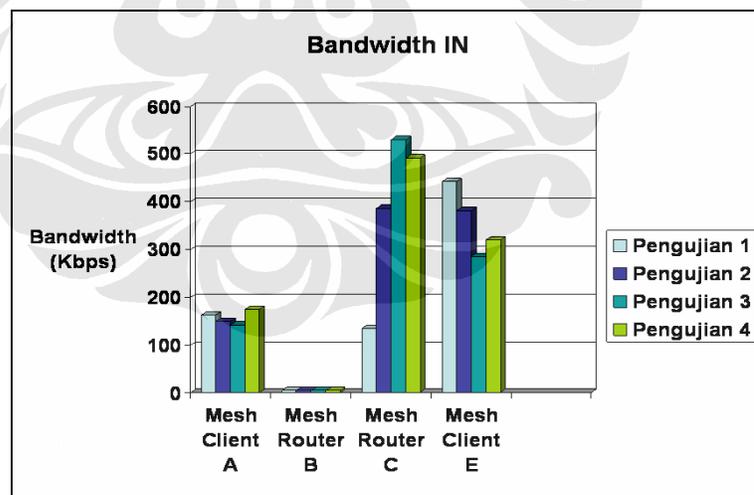
message sebagai paket RREP. Hal inilah yang membuat proses *self healing* jauh lebih lama dibandingkan dengan *self configure*.

4.3.3 Analisa Performansi Pemakaian *Bandwidth*

Analisa yang dilakukan adalah untuk mendapatkan nilai pemakaian *bandwidth* dari setiap *mesh router* dan *mesh client* pada *testbed* jaringan. Dari data hasil pengujian didapat nilai rata-rata *bandwidth* untuk setiap *mesh router* dan *mesh client*. Data pemakaian *bandwidth* ini dibagi dalam tiga bagian, yaitu *bandwidth* yang masuk ke dalam setiap perangkat atau yang berasal dari perangkat lain pada saat melakukan aktivitas (*bandwidth out*), *bandwidth* yang keluar dari dalam setiap perangkat atau yang berasal dari perangkat itu sendiri (*bandwidth in*), dan *bandwidth* total yang merupakan gabungan dari keduanya (*bandwidth in* dan *bandwidth out*).

4.3.3.1 Pemakaian *Bandwidth In*

Grafik hasil perbandingan pemakaian *bandwidth in* dari *mesh router* dan *mesh client* pada setiap pengujian dapat dilihat pada Gambar 4.8.



Gambar 4.8. Grafik perbandingan pemakaian *bandwidth in*.

Dari grafik hasil perbandingan pemakaian *bandwidth in* tersebut terlihat bahwa dari beberapa pengujian yang dilakukan *bandwidth in* pada *mesh client* A lebih kecil dibandingkan dengan *mesh client* E, hal ini disebabkan karena aplikasi

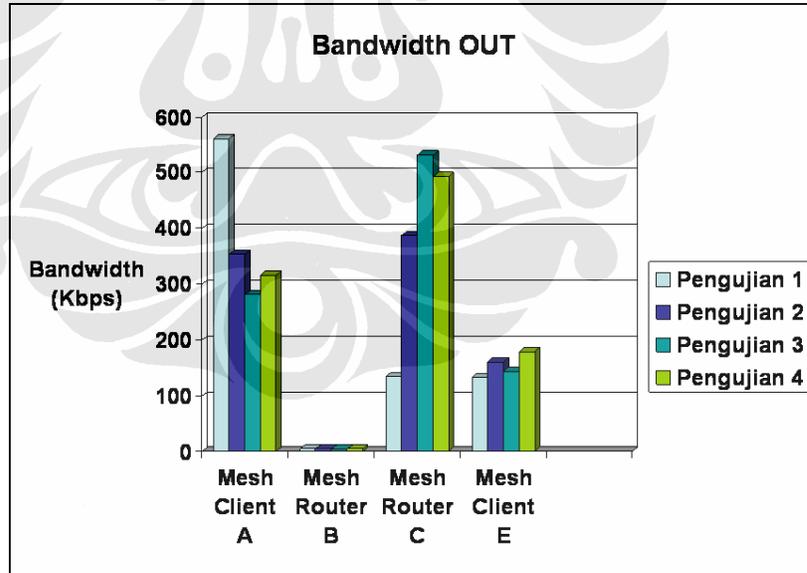
yang digunakan (*audio-video conference*) lebih banyak dijalankan dari *mesh client* E menuju ke *mesh client* A. Sedangkan *bandwidth in* pada *mesh router* C lebih besar dari *mesh router* B, hal ini disebabkan karena jalur yang digunakan untuk jalur distribusi data melalui *mesh router* C (*mesh client* A – *mesh router* C – *mesh client* E).

Besar pemakaian *bandwidth in* rata-rata dari setiap *mesh router* dan *mesh client* pada pengujian ini adalah:

- 1) *bandwidth in* rata-rata *mesh client* A = 157,422 Kbps
- 2) *bandwidth in* rata-rata *mesh router* B = 4,321 Kbps
- 3) *bandwidth in* rata-rata *mesh router* C = 385,459 Kbps
- 4) *bandwidth in* rata-rata *mesh client* E = 357,2 Kbps

4. 3. 3. 2 Pemakaian *Bandwidth Out*

Gambar 4.9 memperlihatkan grafik perbandingan pemakaian *bandwidth out* dari *mesh router* dan *mesh client* pada setiap pengujian.



Gambar 4.9. Grafik perbandingan pemakaian *bandwidth out*.

Dari grafik hasil perbandingan pemakaian *bandwidth out* tersebut terlihat bahwa dari beberapa pengujian yang dilakukan *bandwidth out* pada *mesh client* A lebih besar dibandingkan dengan *mesh client* E, hal ini disebabkan karena aplikasi

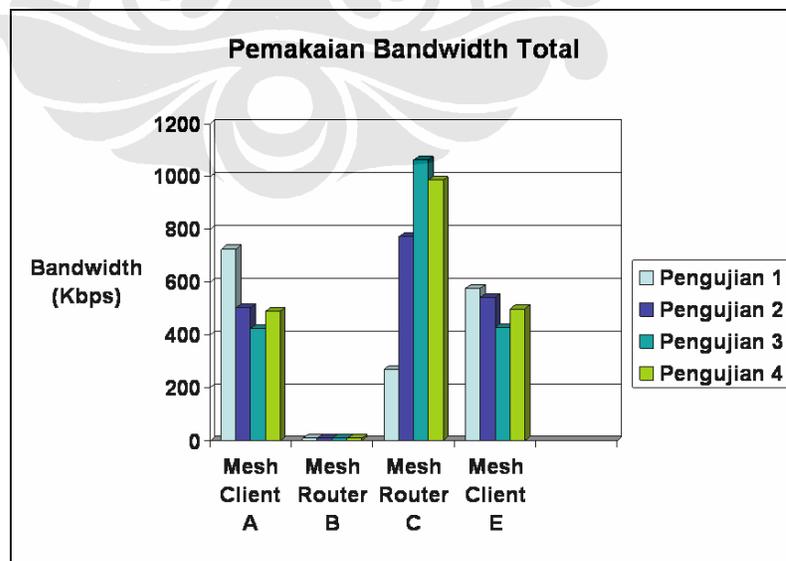
yang digunakan (*audio-video conference*) lebih banyak dijalankan dari *mesh client* E menuju ke *mesh client* A. Sedangkan *bandwidth out* pada *mesh router* C lebih besar dari *mesh router* B, hal ini disebabkan karena jalur yang digunakan untuk jalur distribusi data melalui *mesh router* C (*mesh client* A – *mesh router* C – *mesh client* E). Pada *mesh router* C besarnya *bandwidth in* dan *bandwidth out* tidak jauh berbeda, hal ini disebabkan karena *mesh router* C merupakan jalur yang dilalui oleh *mesh client* A dan *mesh client* E pada saat menjalankan aplikasi (*audio-video conference*) sehingga *bandwidth in* pada *mesh router* C sebanding dengan *bandwidth out*-nya.

Besar pemakaian *bandwidth out* rata-rata dari setiap *mesh router* dan *mesh client* pada pengujian ini adalah:

- 1) *bandwidth out* rata-rata *mesh client* A = 378,299 Kbps
- 2) *bandwidth out* rata-rata *mesh router* B = 3,9 Kbps
- 3) *bandwidth out* rata-rata *mesh router* C = 386,658 Kbps
- 4) *bandwidth out* rata-rata *mesh client* E = 153,931 Kbps

4.3.3.3 Pemakaian Bandwidth Total

Dan grafik perbandingan pemakaian *bandwidth* total dari setiap *mesh router* dan *mesh client* diperlihatkan pada Gambar 4.10.



Gambar 4.10. Grafik perbandingan pemakaian *bandwidth* total.

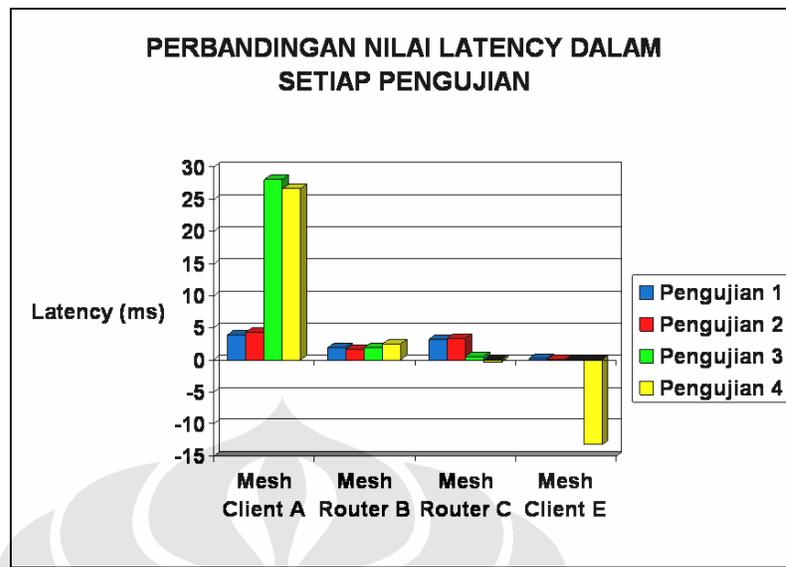
Grafik hasil perbandingan pemakaian *bandwidth* total tersebut merupakan gabungan dari *bandwidth in* dan *bandwidth out* dari setiap *mesh client* dan *mesh router* yang ada. Besarnya pemakaian *bandwidth* total antara *mesh client* A dan *mesh client* E tidaklah terlalu berbeda, karena merupakan akumulasi dari pemakaian *bandwidth* yang terpakai. Adanya perbedaan besarnya *bandwidth* total *mesh client* A dengan *mesh client* E disebabkan pada saat pengujian dilakukan *mesh client* A selalu melakukan perintah pemeriksaan koneksi secara terus menerus “*ping -t*” ke setiap *node* yang ada, sedangkan pada *mesh client* E tidak dilakukan hal demikian terhadap semua *node*. Dan *bandwidth* total pada *mesh router* C lebih besar dari *mesh router* B, hal ini disebabkan karena jalur yang digunakan untuk jalur distribusi data melalui *mesh router* C (*mesh client* A – *mesh router* C – *mesh client* E).

Besar pemakaian *bandwidth* total rata-rata dari setiap *mesh router* dan *mesh client* pada pengujian ini adalah:

- 1) *bandwidth* total rata-rata *mesh client* A = 535,722 Kbps
- 2) *bandwidth* total rata-rata *mesh router* B = 8,22 Kbps
- 3) *bandwidth* total rata-rata *mesh router* C = 772,117 Kbps
- 4) *bandwidth* total rata-rata *mesh client* E = 511,131 Kbps

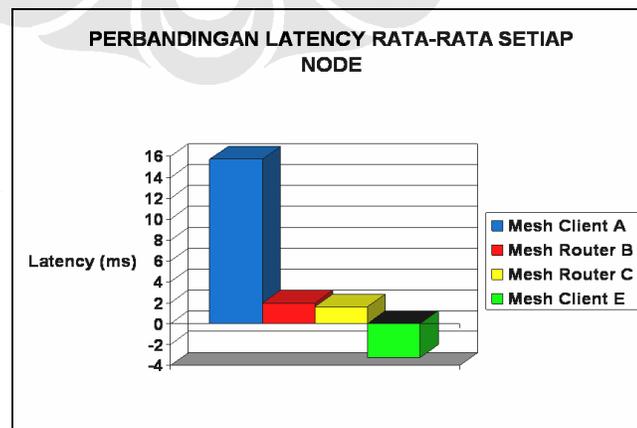
4.3.4 Analisis Performansi *Latency*

Pengujian *latency* ini dilakukan secara bersamaan dengan pengujian untuk pemakaian *bandwidth* dan dengan scenario yang sama pula. Pada pengujian ini akan dicari besarnya nilai *latency* rata-rata yang terjadi pada setiap *mesh client* dan *mesh router* dari pengujian yang dilakukan. Gambar 4.11 memperlihatkan besarnya rata-rata *latency* yang terjadi pada setiap percobaan.



Gambar 4.11. Perbandingan *latency* pada setiap percobaan.

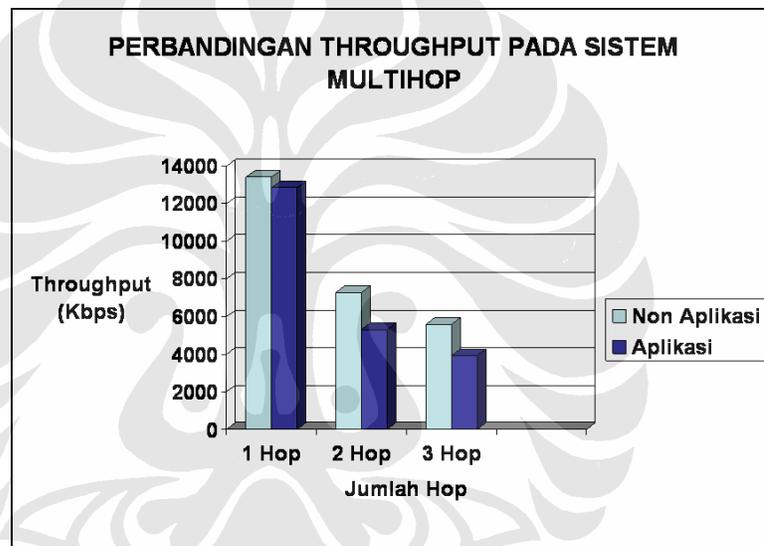
Dari grafik hasil perbandingan *latency* tersebut terlihat bahwa dari beberapa pengujian yang dilakukan *latency* pada *mesh client A* lebih besar dibandingkan dengan *mesh client E*, hal ini disebabkan karena aplikasi yang digunakan (*audio-video conference*) lebih banyak dijalankan dari *mesh client E* menuju ke *mesh client A*. *Latency* pada *mesh router C* lebih besar dari *mesh router B*, hal ini disebabkan karena jalur yang digunakan untuk jalur distribusi data melalui *mesh router C* (*mesh client A – mesh router C – mesh client E*). Perbandingan besarnya rata-rata *latency* dari setiap *node* dapat diperlihatkan pada Gambar 4.12.



Gambar 4.12. Perbandingan besarnya *latency* dari setiap *node*.

4.3.5 Analisa Performansi *End-to-End Throughput*

Pengujian ini dilakukan dengan menerapkan skenario yang terbentuk dengan sistem *multihop* pada *wireless mesh network*. Pengujian ditujukan untuk mengamati performansi *end-to-end throughput* pada sistem *multihop* dari *wireless mesh network*. Dari data yang diperoleh pada pengujian ini, didapatkan nilai rata-rata *end-to-end throughput* yang terdiri dari *throughput* pada sistem 1 *hop*, *throughput* pada sistem 2 *hop*, dan *throughput* pada sistem 3 *hop*. Perbandingan *throughput* yang didapatkan dari setiap jumlah *hop* yang dilalui ini diperlihatkan pada Gambar 4.13.



Gambar 4.13. Grafik perbandingan *throughput* pada sistem *multihop*.

Berdasarkan grafik perbandingan *throughput* pada sistem *multihop* tersebut dapat diketahui bahwa performansi *end-to-end throughput* dari setiap sistem *multihop* akan semakin menurun seiring dengan bertambahnya jumlah *hop* yang ada pada *wireless mesh network*. Penurunan performansi ini terjadi pada saat jaringan tanpa beban maupun saat jaringan diberikan beban aplikasi (*audio-video conference*).

Penurunan tersebut sebanding dengan nilai dari hasil pengukuran *bandwidth* yang terjadi pada pengujian pemakaian *bandwidth* pada setiap *node*, dimana terjadi perubahan nilai *bandwidth* pada setiap *node* yang dilalui sebagai jalur lalu

lintas data. Penurunan performansi yang terjadi pada setiap jumlah *hop* yang berbeda baik dalam keadaan tanpa beban maupun saat diberikan beban aplikasi (*audio-video conference*) ini dapat diperhitungkan secara persentase sebagai berikut.

1. Penurunan performansi pada saat jaringan tanpa beban.

Persentase antara 1 *hop* dengan 2 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 2 \text{ hop} &= \frac{13421,6 - 7270,433}{13421,6} \times 100\% \\ &= 45,83\% \end{aligned}$$

Persentase antara 2 *hop* dengan 3 *hop*.

$$\begin{aligned} 2 \text{ hop dengan } 3 \text{ hop} &= \frac{7270,433 - 5592,633}{7270,433} \times 100\% \\ &= 23,08\% \end{aligned}$$

Persentase antara 1 *hop* dengan 3 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 3 \text{ hop} &= \frac{13421,6 - 5592,633}{13421,6} \times 100\% \\ &= 58,33\% \end{aligned}$$

2. Penurunan performansi pada saat jaringan diberikan beban aplikasi (*audio-video conference*).

Persentase antara 1 *hop* dengan 2 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 2 \text{ hop} &= \frac{12862,37 - 5313,033}{12862,37} \times 100\% \\ &= 58,69\% \end{aligned}$$

Persentase antara 2 *hop* dengan 3 *hop*.

$$\begin{aligned} 2 \text{ hop dengan } 3 \text{ hop} &= \frac{5313,033 - 3914,867}{5313,033} \times 100\% \\ &= 26,32\% \end{aligned}$$

Persentase antara 1 *hop* dengan 3 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 3 \text{ hop} &= \frac{12862,37 - 3914,867}{12862,367} \times 100\% \\ &= 69,56\% \end{aligned}$$

Pemberian beban aplikasi (*audio-video conference*) pada jaringan juga mempengaruhi performansi dari *end-to-end throughput* yang dihasilkan dari setiap jumlah *hop* yang berbeda. Dengan diberikannya beban aplikasi (*audio-video conference*) performansi *end-to-end throughput* akan menurun, hal ini dapat diperlihatkan dengan menghitung persentase penurunan yang terjadi.

Persentase penurunan *end-to-end throughput* pada saat diberikan beban aplikasi.

$$\begin{aligned} \text{Throughput 1 hop} &= \frac{13421,6 - 12862,37}{13421,6} \times 100\% \\ &= 4,17\% \end{aligned}$$

$$\begin{aligned} \text{Throughput 2 hop} &= \frac{7270,433 - 5313,033}{7270,433} \times 100\% \\ &= 26,92\% \end{aligned}$$

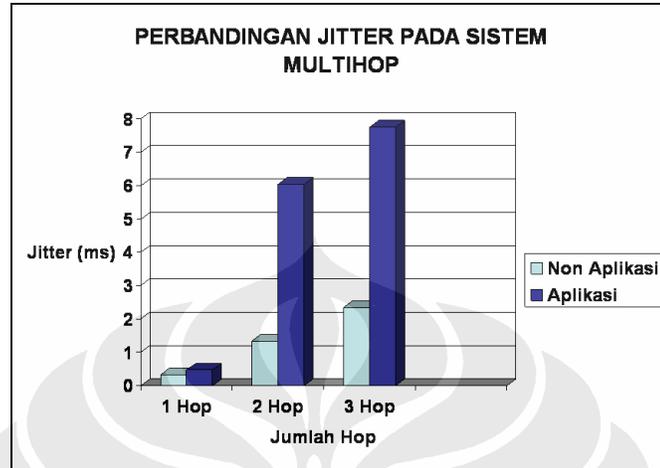
$$\begin{aligned} \text{Throughput 3 hop} &= \frac{5592,633 - 3914,867}{5592,633} \times 100\% \\ &= 29,99\% \end{aligned}$$

4.3.6 Analisa Performansi *Jitter*

Pada pengujian ini skenario yang dilakukan dengan menerapkan skenario yang terbentuk dengan sistem *multihop* pada *wireless mesh network*. Pengujian ditujukan untuk mengamati performansi *end-to-end jitter* pada sistem *multihop* dari *wireless mesh network*.

Dari data yang diperoleh pada pengujian ini, didapatkan nilai rata-rata *end-to-end jitter* yang terdiri dari *jitter* pada sistem 1 *hop*, *jitter* pada sistem 2 *hop*, dan

jitter pada sistem 3 *hop*. Perbandingan *jitter* yang didapatkan dari setiap jumlah *hop* yang dilalui ini diperlihatkan pada Gambar 4.14.



Gambar 4.14. Grafik perbandingan *jitter* pada sistem *multihop*.

Berdasarkan grafik perbandingan *jitter* pada sistem *multihop* tersebut dapat diketahui bahwa performansi *end-to-end jitter* dari setiap sistem *multihop* akan semakin meningkat seiring dengan bertambahnya jumlah *hop* yang ada pada *wireless mesh network*. Peningkatan performansi ini terjadi pada saat jaringan tanpa beban maupun saat jaringan diberikan beban aplikasi (*audio-video conference*).

Peningkatan tersebut sebanding dengan nilai dari hasil pengukuran *latency* yang terjadi pada pengujian *latency* pada setiap *node*, dimana terjadi perubahan nilai *latency* pada setiap *node* yang dilalui sebagai jalur lalu lintas data. Peningkatan performansi yang terjadi pada setiap jumlah *hop* yang berbeda baik dalam keadaan tanpa beban maupun saat diberikan beban aplikasi (*audio-video conference*) ini dapat diperhitungkan secara persentase sebagai berikut.

1. Penurunan performansi pada saat jaringan tanpa beban.

Persentase antara 1 *hop* dengan 2 *hop*.

$$\begin{aligned}
 1 \text{ hop dengan } 2 \text{ hop} &= \frac{1,344 - 0,3104}{1,344} \times 100\% \\
 &= 76,9\%
 \end{aligned}$$

Persentase antara 2 *hop* dengan 3 *hop*.

$$\begin{aligned} 2 \text{ hop dengan } 3 \text{ hop} &= \frac{2,347 - 1,344}{2,347} \times 100\% \\ &= 42,74\% \end{aligned}$$

Persentase antara 1 *hop* dengan 3 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 3 \text{ hop} &= \frac{2,347 - 0,3104}{2,347} \times 100\% \\ &= 86,77\% \end{aligned}$$

2. Penurunan performansi pada saat jaringan diberikan beban aplikasi (*audio-video conference*).

Persentase antara 1 *hop* dengan 2 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 2 \text{ hop} &= \frac{6,051 - 0,464}{6,051} \times 100\% \\ &= 92,33\% \end{aligned}$$

Persentase antara 2 *hop* dengan 3 *hop*.

$$\begin{aligned} 2 \text{ hop dengan } 3 \text{ hop} &= \frac{7,76 - 6,051}{7,76} \times 100\% \\ &= 22,02\% \end{aligned}$$

Persentase antara 1 *hop* dengan 3 *hop*.

$$\begin{aligned} 1 \text{ hop dengan } 3 \text{ hop} &= \frac{7,76 - 0,464}{7,76} \times 100\% \\ &= 94,02\% \end{aligned}$$

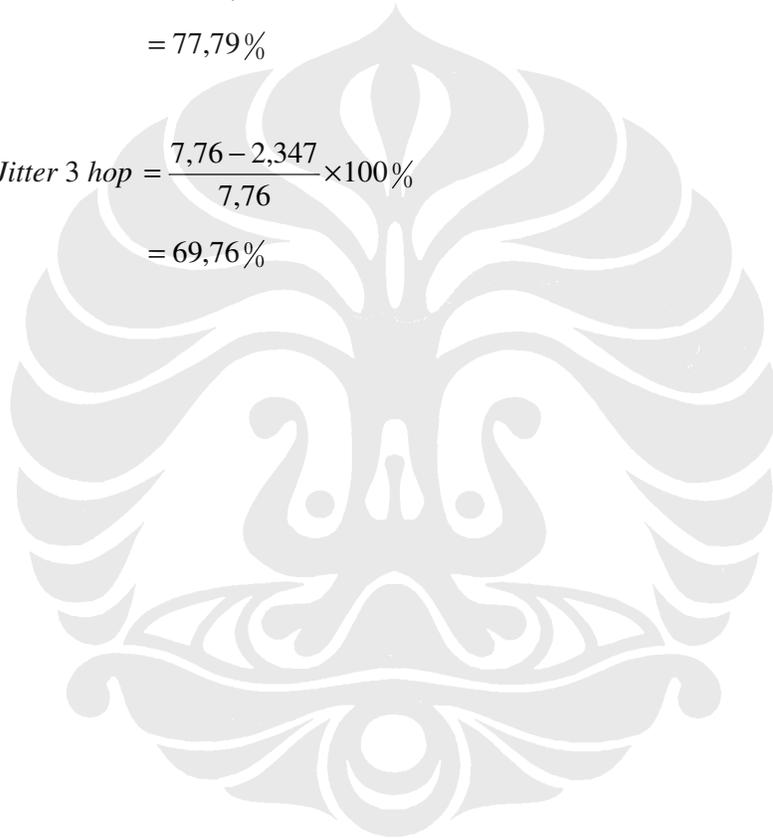
Pemberian beban aplikasi (*audio-video conference*) pada jaringan juga mempengaruhi performansi dari *end-to-end jitter* yang dihasilkan dari setiap jumlah *hop* yang berbeda. Dengan diberikannya beban aplikasi (*audio-video conference*) performansi *end-to-end jitter* akan menurun, hal ini dapat diperlihatkan dengan menghitung persentase peningkatan yang terjadi.

Persentase peningkatan *end-to-end jitter* pada saat diberikan beban aplikasi.

$$\begin{aligned} \text{Jitter 1 hop} &= \frac{0,464 - 0,3104}{0,464} \times 100\% \\ &= 33,1\% \end{aligned}$$

$$\begin{aligned} \text{Jitter 2 hop} &= \frac{6,051 - 1,344}{6,051} \times 100\% \\ &= 77,79\% \end{aligned}$$

$$\begin{aligned} \text{Jitter 3 hop} &= \frac{7,76 - 2,347}{7,76} \times 100\% \\ &= 69,76\% \end{aligned}$$



BAB 5

KESIMPULAN

Berdasarkan hasil pengujian dan analisa sistem yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut ini.

1. Performansi *self configure* dipengaruhi oleh *Hello Interval*, semakin besar nilai *Hello Interval* yang diberikan akan memperbesar waktu yang diperlukan sebuah *node* dalam melakukan *self configure*. Hal ini disebabkan karena *Hello Interval* merupakan *interval* waktu yang diperlukan oleh *Hello message* untuk memeriksa informasi konektifitas dengan *node* tetangganya.
2. Pada pengujian *self healing* juga dipengaruhi oleh *Hello Interval*. Waktu yang diperlukan saat sebuah *node* melakukan *self healing* lebih lama dibandingkan saat sebuah *node* melakukan *self cofigure*, karena pada saat sebuah *node* melakukan *self healing* informasi rute awal ketujuan yang tersimpan di tabel *routing* akan diperiksa terlebih dahulu.
3. Pada pengujian pemakaian *bandwidth* dan *latency* untuk setiap *node* performanya dipengaruhi statusnya, apakah merupakan *node* dalam jalur distribusi data atau tidak.
4. Pada pengujian pemakaian *bandwidth* untuk setiap *node*, besarnya nilai *bandwidth* yang diperoleh berbanding lurus dengan aktivitas yang dilakukan oleh *node*-nya.
5. Dalam pengujian pemakaian *bandwidth* total, nilai terbesar rata-rata terjadi pada *mesh router* C yaitu sebesar 772,117 Kbps dan nilai terkecil rata-rata terjadi pada *mesh router* B yaitu sebesar 8,22 Kbps.
6. Nilai *end-to-end throughput* untuk sistem *multihop* pada *wireless mesh network* sangat dipengaruhi oleh jumlah *hop* yang dilalui sebagai jalur datanya serta besarnya data dikirimkan, dengan bertambahnya jumlah *hop* performansi *end-to-end throughput* juga semakin berkurang.
7. Nilai rata-rata *throughput* paling besar didapatkan pada jalur 1 *hop* baik untuk kondisi tanpa beban maupun kondisi dengan beban. Untuk kondisi tanpa beban nilai rata-rata *throughput* paling besar adalah 13421,6 Kbps dan

untuk kondisi berbeban nilai rata-rata *throughput* paling besar adalah 12862,37 Kbps.

8. Nilai *end-to-end jitter* dipengaruhi oleh jumlah *hop* yang dilalui dan besarnya data yang dikirimkan, dengan bertambahnya jumlah *hop* maka nilai *jitter* yang didapatkan akan semakin meningkat.
9. Nilai rata-rata *jitter* terkecil didapatkan pada jalur 1 *hop* baik untuk kondisi tanpa beban maupun kondisi dengan beban. Untuk kondisi tanpa beban nilai rata-rata *jitter* terkecil adalah 0,3104 ms dan untuk kondisi berbeban nilai rata-rata *jitter* terkecil adalah 0,464 ms.



DAFTAR ACUAN

- [1] Gilbert Held, *Wireless Mesh Network* (Boca Raton: Auerbach Publications Taylor & Francis Group, 2005), hal.2 – 4, hal.63 – 69.
- [2] Ian F. Akyildiz, Xudong Wang, Weilin Wang (2005). “Wireless mesh networks: a survey”. *Computer Network*, 47, 445-487. Diakses 21 April 2008 dari Sciencedirect.
http://perso.citi.insa-lyon.fr/afrabout/ctr/docs/05_commun.pdf
- [3] Dan Keun Sung (2008). “Wireless Local Area Network (WLAN)”. *Communication Networks Research Lab*. Diakses 26 Mei 2008 dari KAIST.
http://pcnr.kaist.ac.kr/lecture/523_2008/download/Lab06_Prof_2008.ppt
- [4] C. Perkins, E. Belding Royer, S. Das (Juli 2003). “Ad hoc On-Demand Distance Vector (AODV) Routing”. *Request For Comments (RFC) 3561*. Diakses 26 Mei 2008 dari Internet Engineering Task Force.
<http://www.ietf.org/rfc3561.txt>
- [5] Palaniappan Annamalai. “*Comparative Performance Study of Standardized Ad-Hoc Routing Protocols and OSPF-MCDS*”. Thesis, Master of Science in Electrical Engineering Faculty of Virginia Polytechnic Institute and State University, Blacksburg, Virginia, Oktober, 2005, hal.9 – 12.
- [6] Yan Zhang, Jijun Luo, Honglin Hu, *Wireless Mesh Networking* (Boca Raton: Auerbach Publications Taylor & Francis Group, 2007), hal.15, 16, 17, 117, 123.
- [7] Bjorn Wiberg. “*Porting AODV-UU Implementation to ns-2 and Enabling Trace-based Simulation*”. Thesis, Information Technology Department of Computer Systems Uppsala University, Swedia, 18 Desember, 2002, hal.16 – 26.
- [8] Felix Fietkau (2005). “OpenWrt Hacking”. *Hacking*. Diakses 21 April 2008 dari 22nd Chaos Communication Congress Private Investigations.
http://events.ccc.de/congress/2005/fahrplan/attachments/567-Paper_HackingOpenWRT.pdf
- [9] Wikipedia Foundation Inc (2008). “OpenWrt”. *Free Encyclopedia*. Diakses 21 April 2008 dari Wikipedia.
<http://en.wikipedia.org/wiki/OpenWrt>
- [10] Paul Asadoorian, Larry Pesce, *Linksys WRT54G Ultimate Hacking*. (United State of America: Syngress Publishing Inc, 2007), hal. 5 – 6, hal. 69.

[11] Simon B (2007). “OpenWrtOverview”. OpenWrt. Diakses 15 April 2008 dari OpenWrt.

<http://wiki.openwrt.org/OpenWrtOverview>

[12] Karsten Renhak. “*Entwicklung einer Clientsoftware zur Unterstützung des kontextsensitiven Routings*”. Diplomarbeit, Fakultät für Elektrotechnik und Informationstechnik Technische Universität Ilmenau, Ilmenau, Maret, 2008, hal.31 – 32.

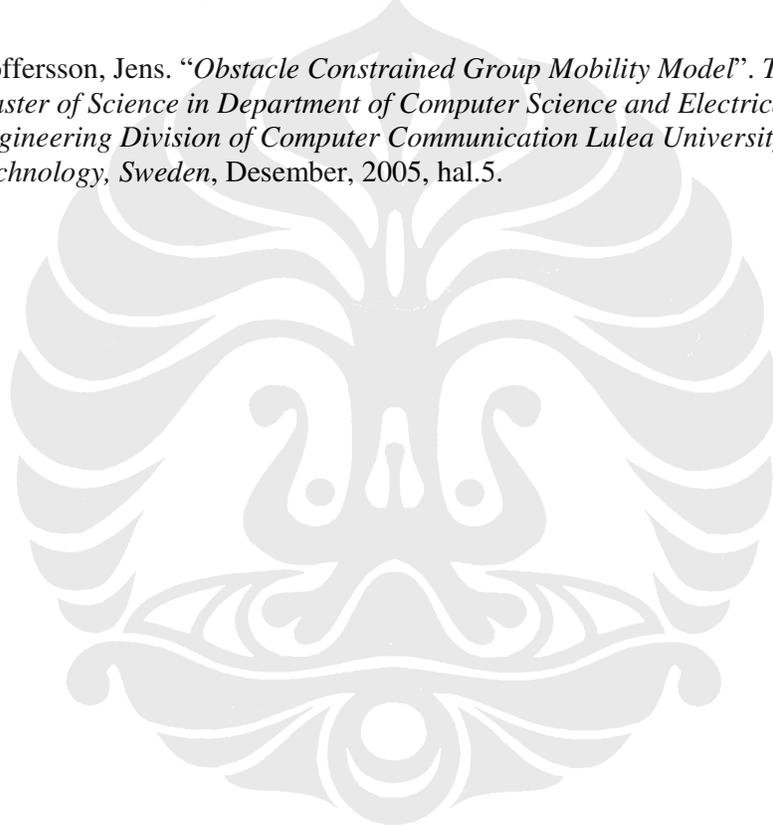
[13] Irawan (2007). “Instalasi OpenWRT di Linksys-WRT54GL”. Catatan Seorang Ophise Boy. Diakses 15 April 2008 dari Blog pada WordPress.

<http://servas.wordpress.com/2007/12/01/instalasi-openwrt-di-linksys-wrt54gl/>



DAFTAR PUSTAKA

- Udugama, Asanga., *UoBWinAODV*. Diakses 20 November 2008 dari comnets kommunikationsnetze.
<http://www.aodv.org>
- Brenner, Pablo (1997). "A Technical Tutorial on the IEEE 802.11 Protocol". *Breezeecom Wireless Communications*. Diakses 26 Mei 2008 dari SSS Magazine.
http://sss-mag.com/pdf/802_11tut.pdf
- Kristoffersson, Jens. "*Obstacle Constrained Group Mobility Model*". Thesis, *Master of Science in Department of Computer Science and Electrical Engineering Division of Computer Communication Lulea University of Technology, Sweden*, Desember, 2005, hal.5.



LAMPIRAN 1: Spesifikasi *Wireless Router* WRT54GL Versi 1.1

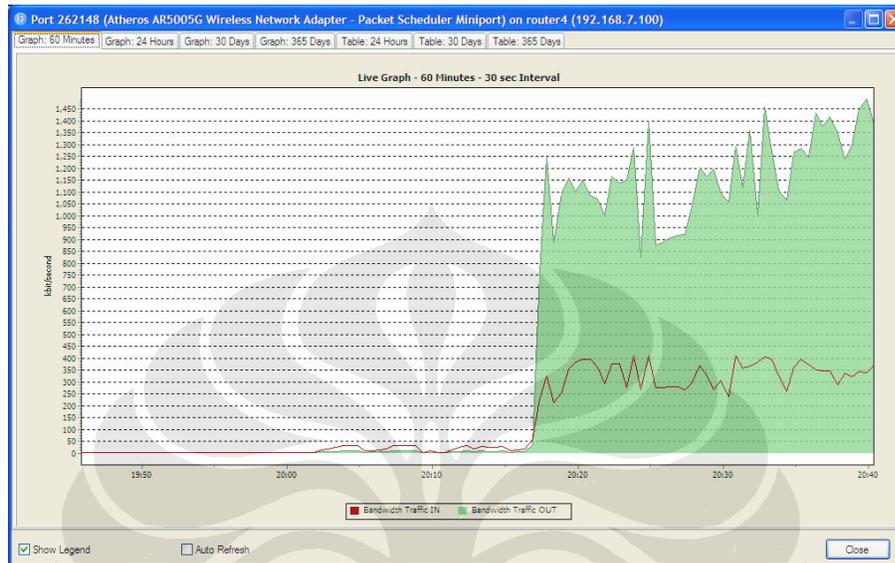
CPU type and version	:	BCM3302 V0.8
CPU speed	:	200 MB
RAM	:	16 MB
Flash memory	:	4 MB
Serial number prefix	:	CL7B
Ports	:	Internet (VLAN0) →1, RJ-45 Port LAN (VLAN1) → 1 – 4, RJ-45 Port 1 Power Port
Buttons	:	Reset, SecureEasySetup
Wireless interface	:	eth1
Networking bridge	:	br0
Antenna connector	:	RP-TNC
Standards	:	IEEE 802.3, IEEE 802.3u, IEEE 802.11g, IEEE 802.11b
Cabling Type	:	Ethernet Network Cable
LEDs	:	Power, DMZ, WLAN, LAN (1-4), Internet
RF Power Output	:	18 dBm
Dimensions	:	186 mm × 48 mm × 200 mm
Unit Weight	:	0,482 kg
Power	:	External, 12V, 0.5A
Certifications	:	FCC, IC-03, CE
Operating Temperature	:	32°F to 104°F
Storage Temperature	:	-4°F to 158°F

LAMPIRAN 2: Tabel Hasil Pengujian *Self Configure* Dan *Self Healing*.

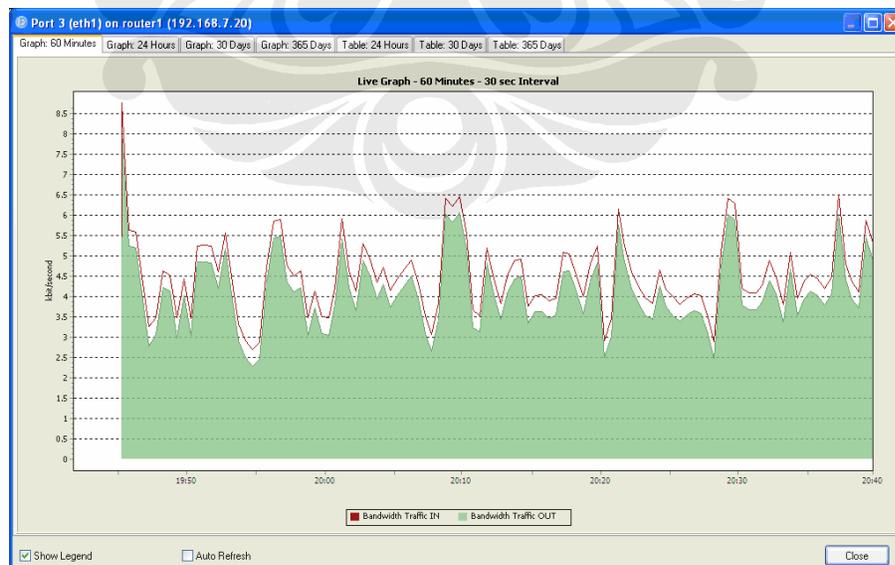
HelloInterval (ms)	Waktu Yang Diperlukan (s)	
	<i>Self Configure</i>	<i>Self Healing</i>
100	3,12	106,43
100	2,87	117,32
100	2,54	120,76
100	2,67	107,19
100	3,85	113,81
Rata-rata	3,01	113,102
500	4,35	174,73
500	4,69	183,21
500	5,17	170,29
500	4,77	193,42
500	4,82	189,32
Rata-rata	4,76	182,194
1000	7,43	525,27
1000	6,78	482,19
1000	7,21	402,24
1000	7,34	461,56
1000	6,83	562,65
Rata-rata	7,118	486,782

LAMPIRAN 3: Grafik Hasil Pengujian Pemakaian *Bandwidth*.

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian pertama.

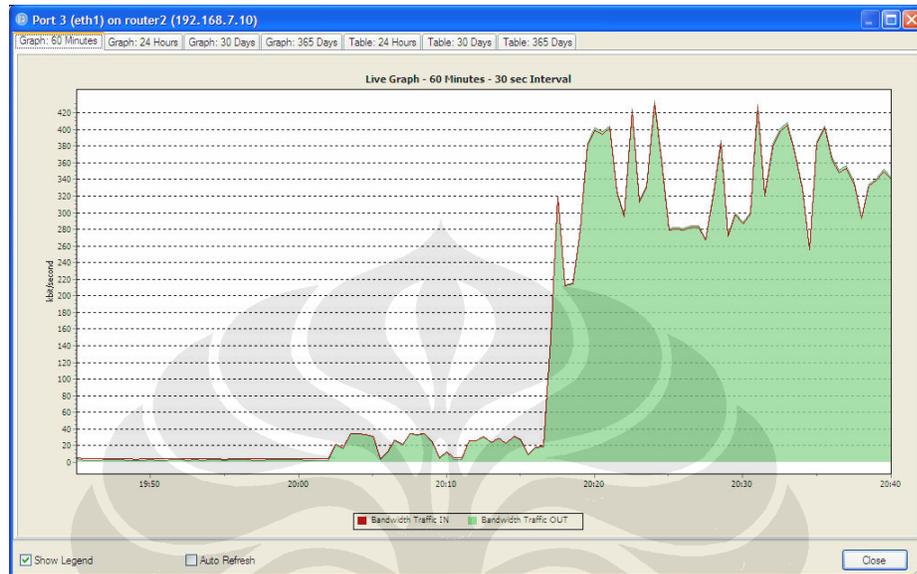


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian pertama.

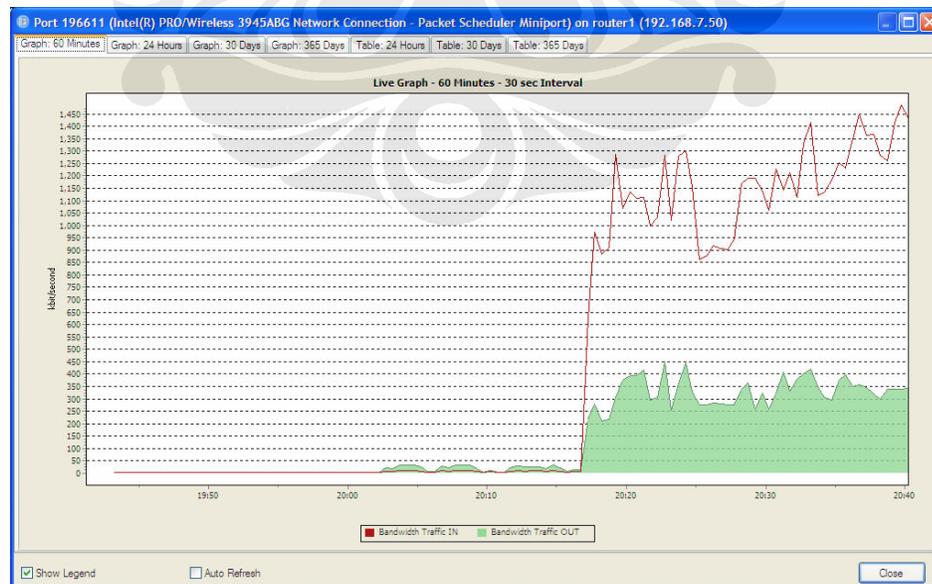


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router C* pada pengujian pertama.

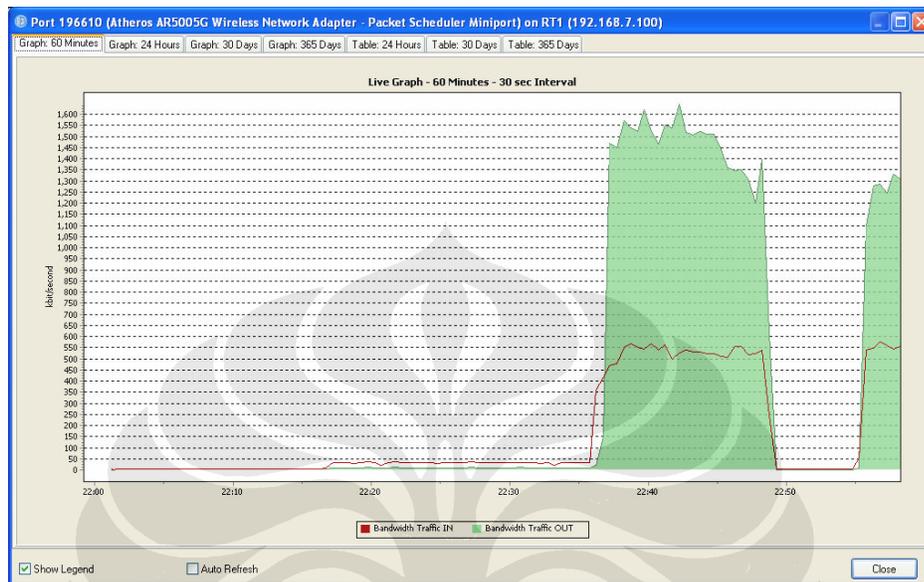


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client E* pada pengujian pertama.

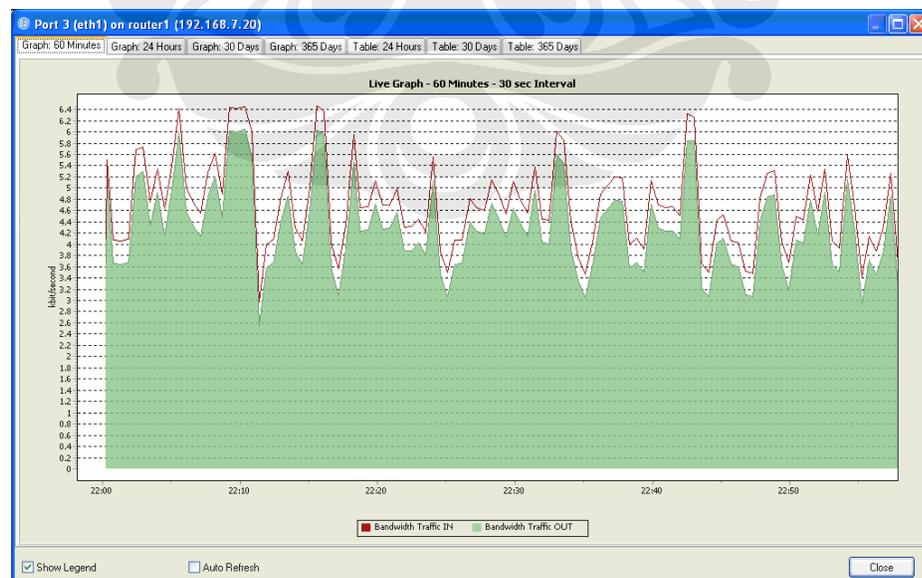


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian kedua.

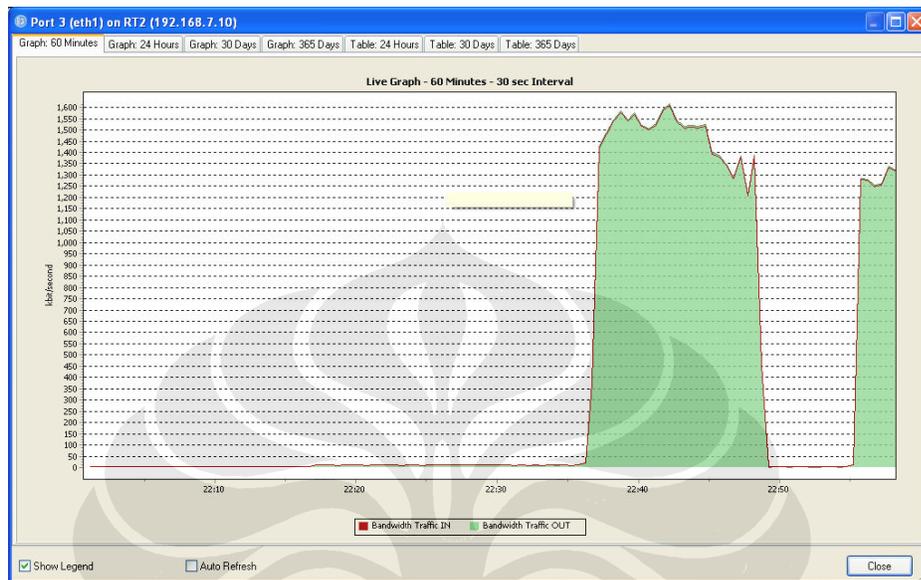


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian kedua.

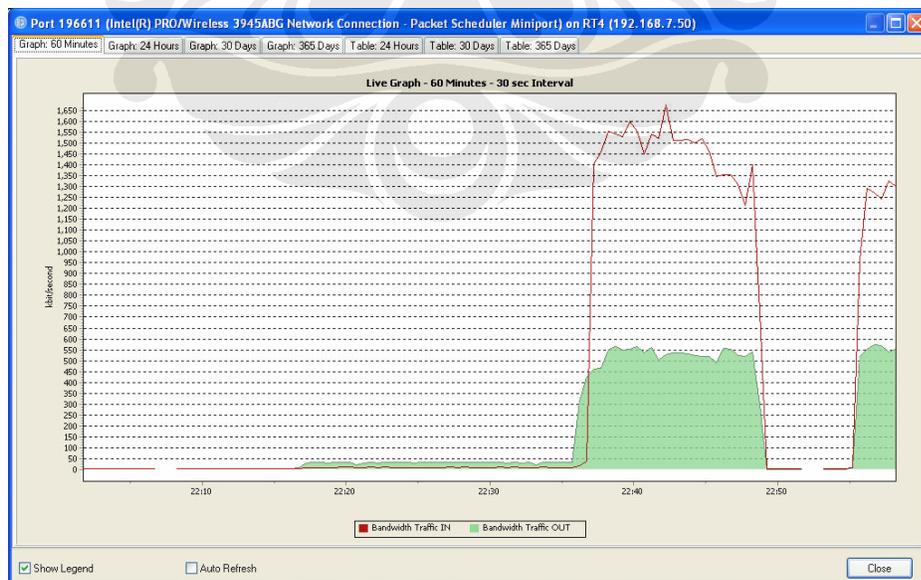


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router C* pada pengujian kedua.

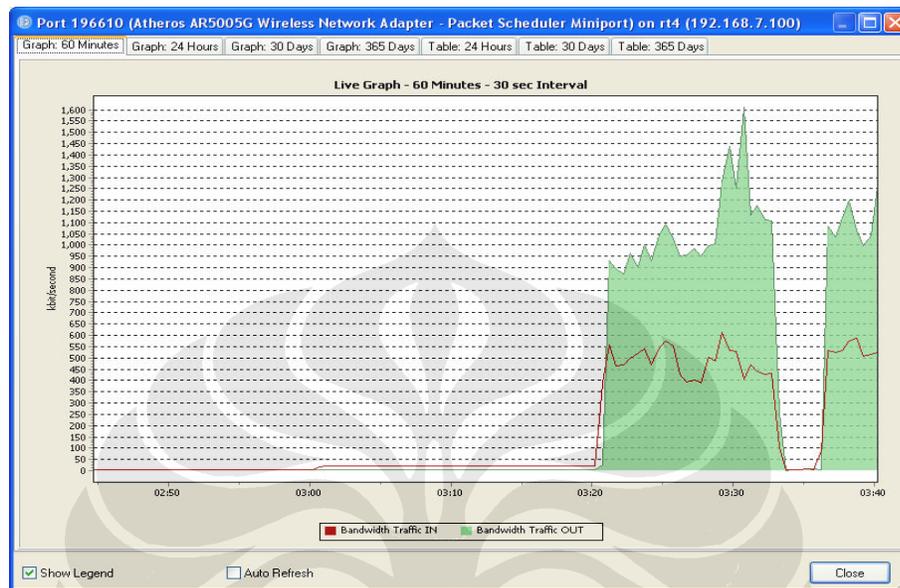


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client E* pada pengujian kedua.

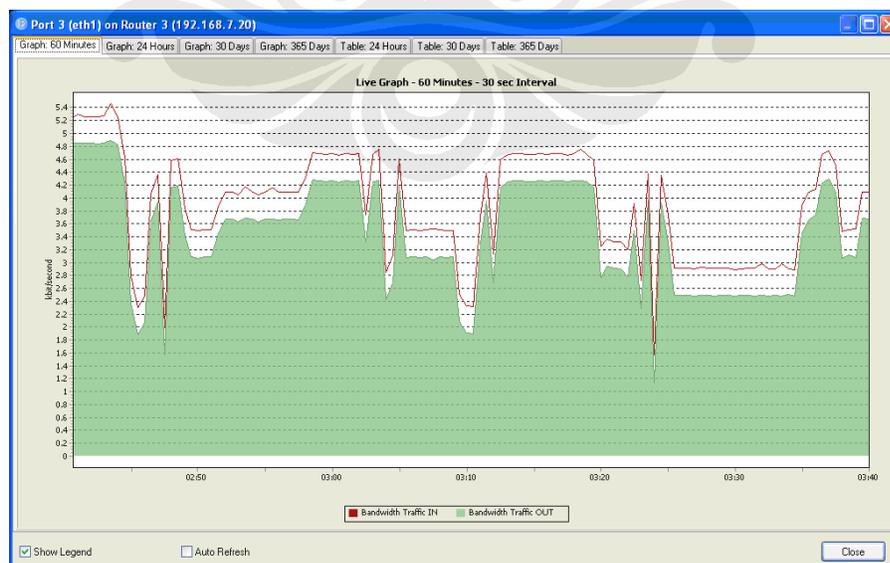


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian ketiga.

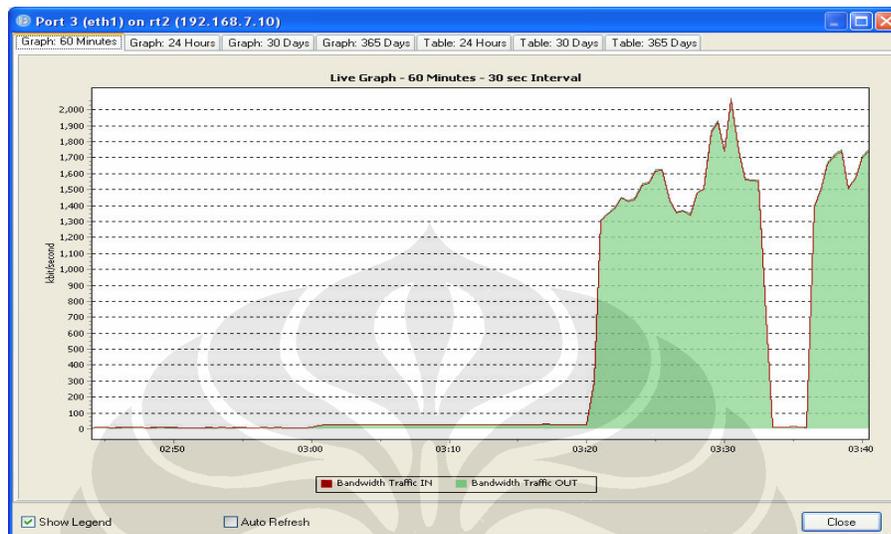


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian ketiga.

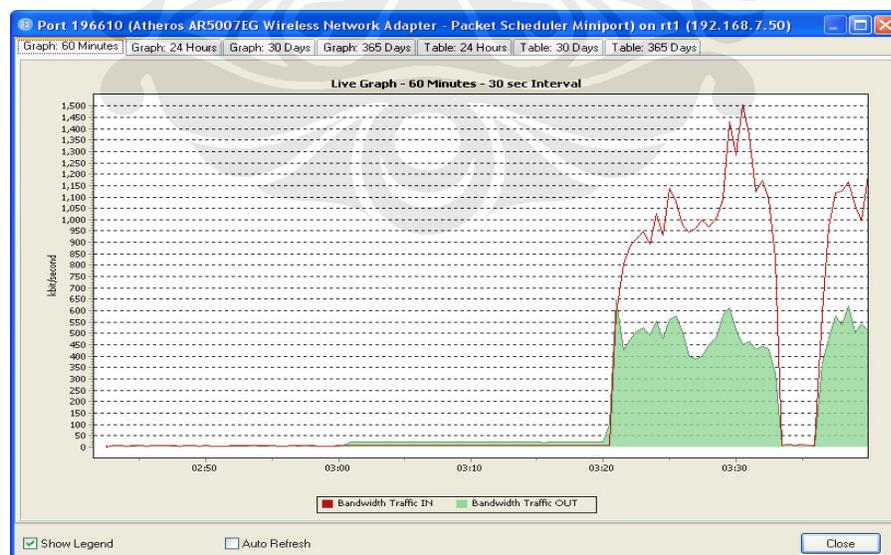


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router C* pada pengujian ketiga.

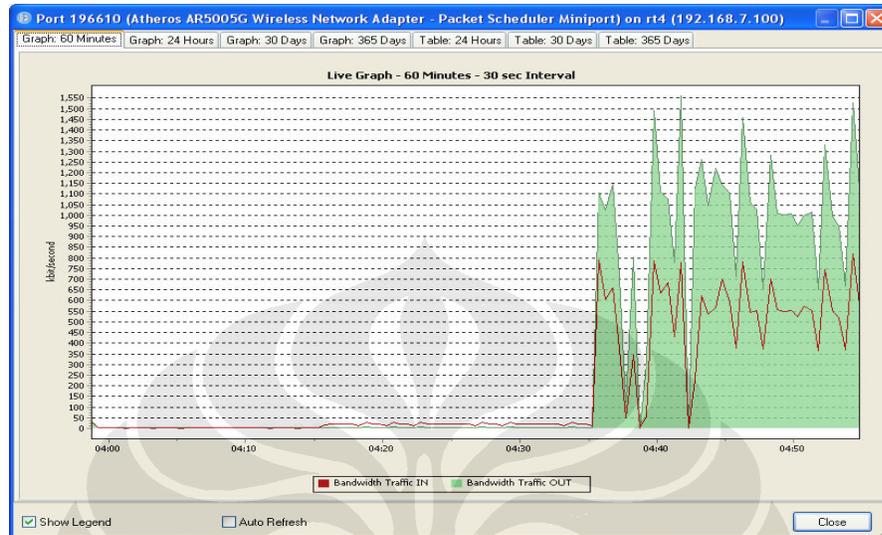


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client E* pada pengujian ketiga.

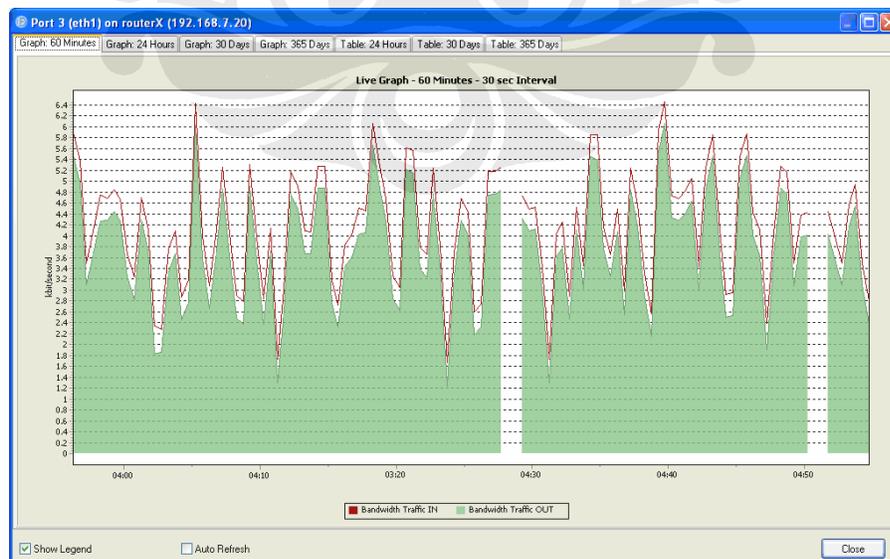


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian keempat.

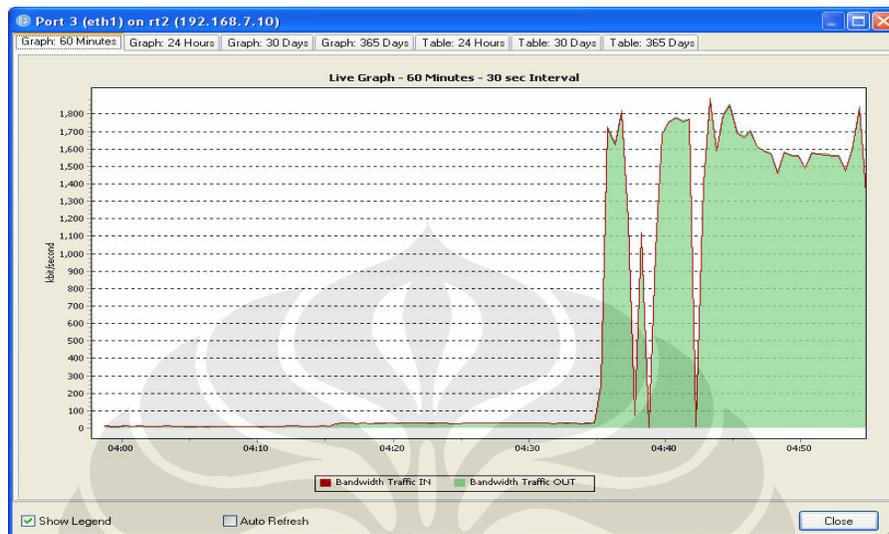


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian keempat.

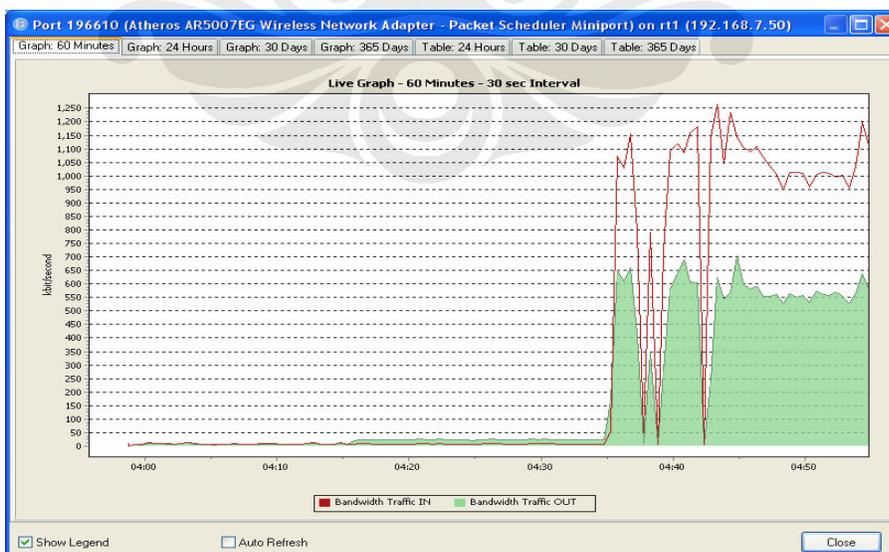


(lanjutan)

Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router C* pada pengujian keempat.

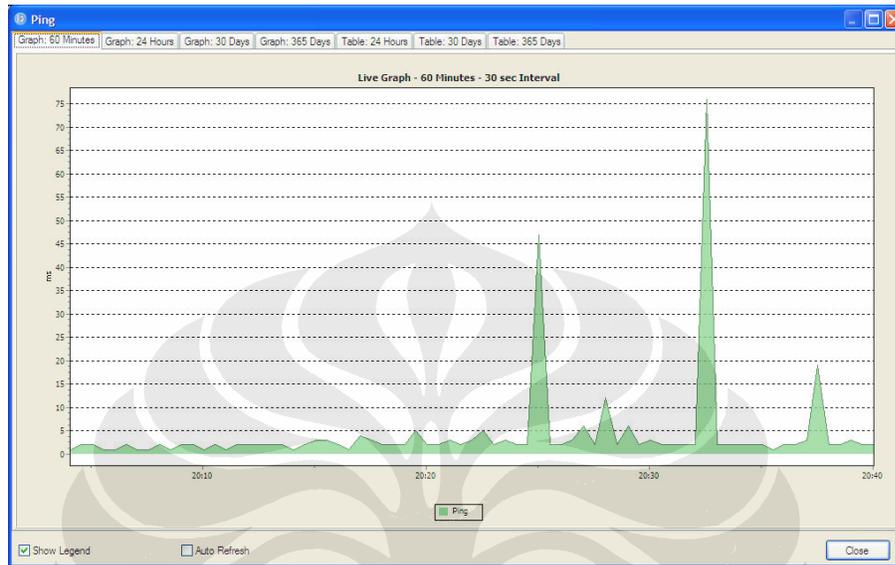


Grafik pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client E* pada pengujian keempat.

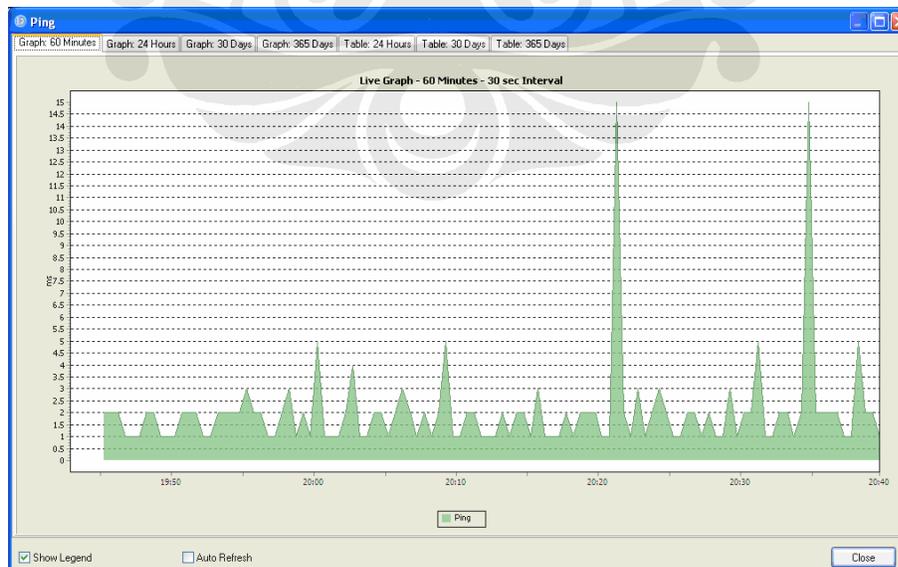


LAMPIRAN 4: Grafik Hasil Pengamatan *Latency*.

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client* A pada pengujian pertama.

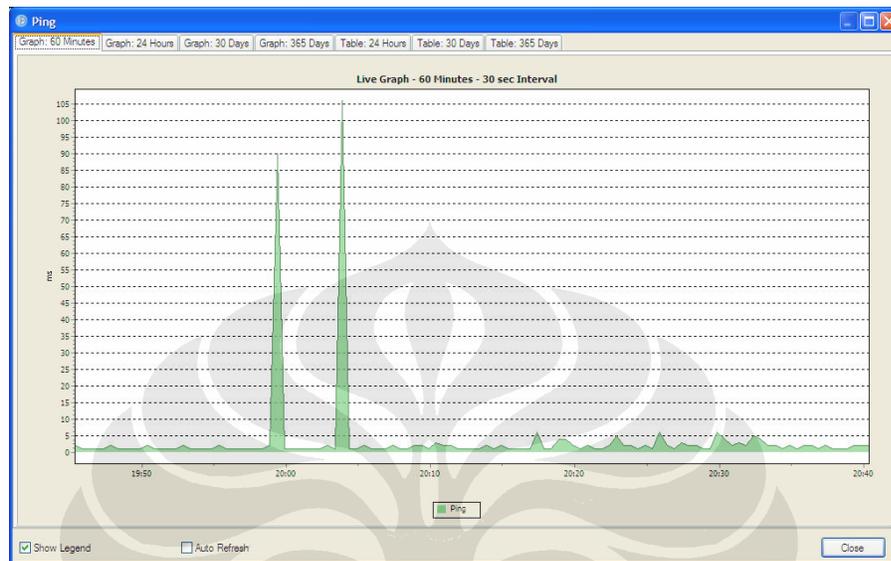


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client* B pada pengujian pertama.

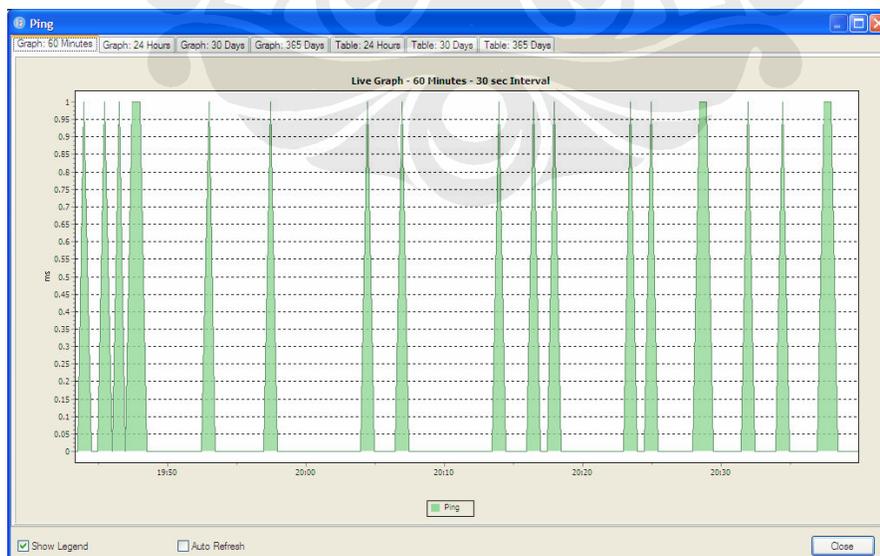


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh router C* pada pengujian pertama.

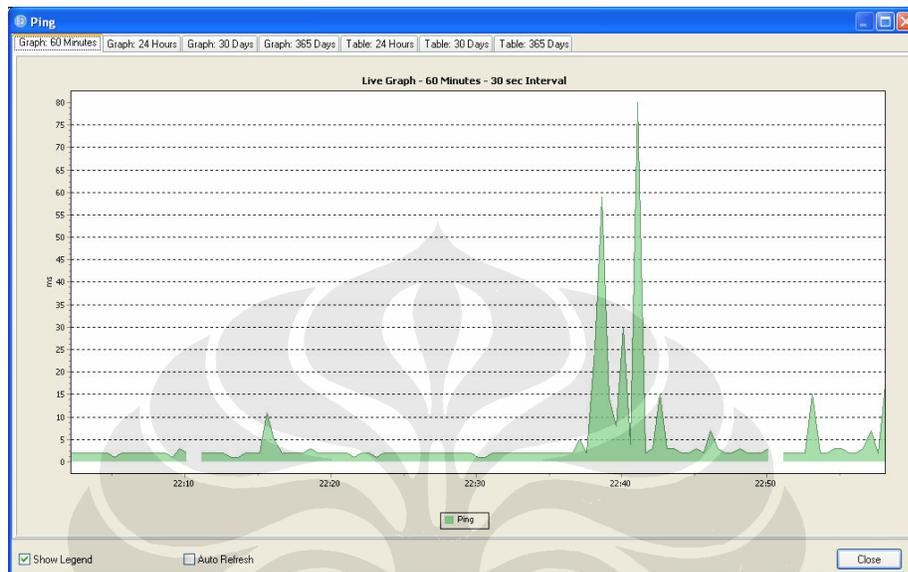


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client E* pada pengujian pertama.

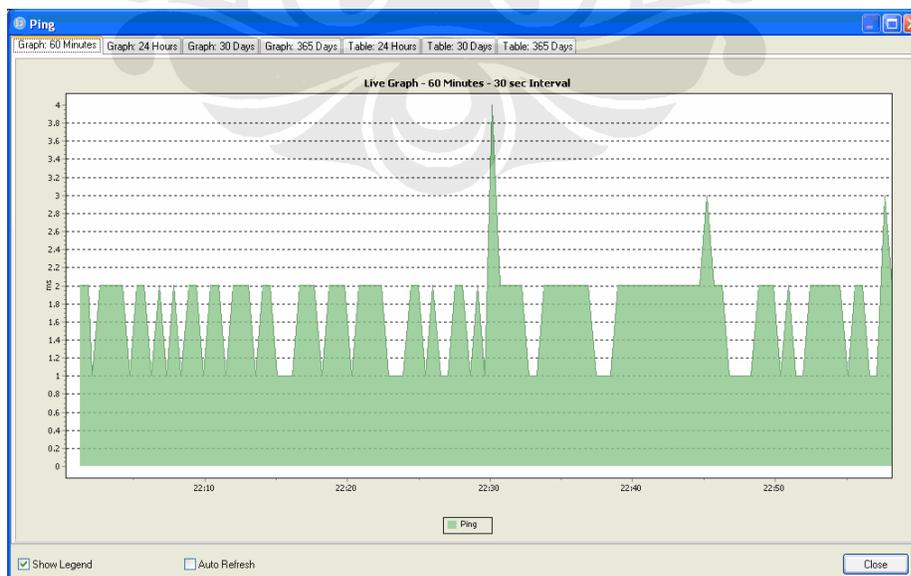


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client A* pada pengujian kedua.

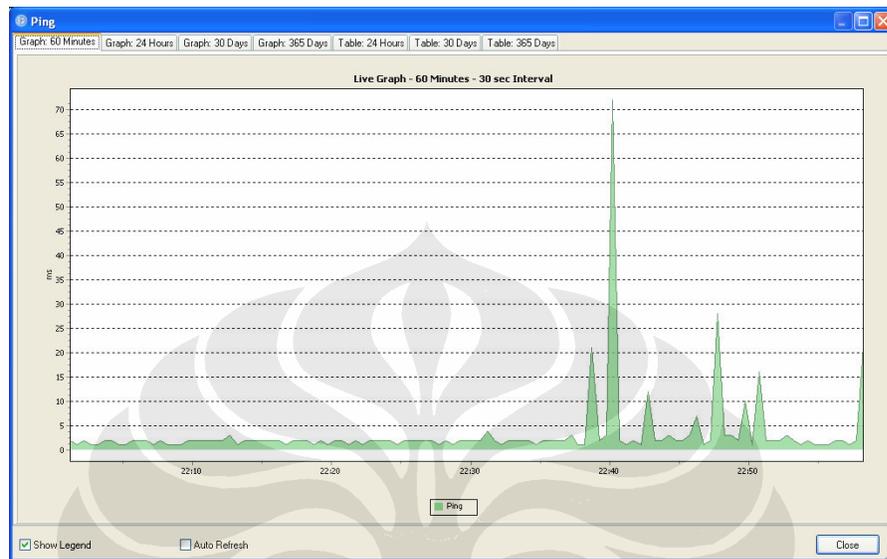


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client B* pada pengujian kedua.

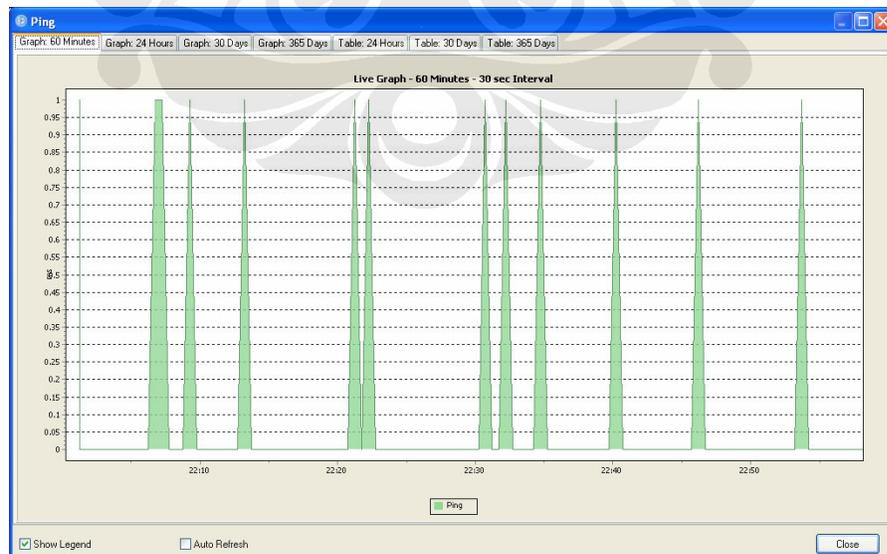


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh router C* pada pengujian kedua.

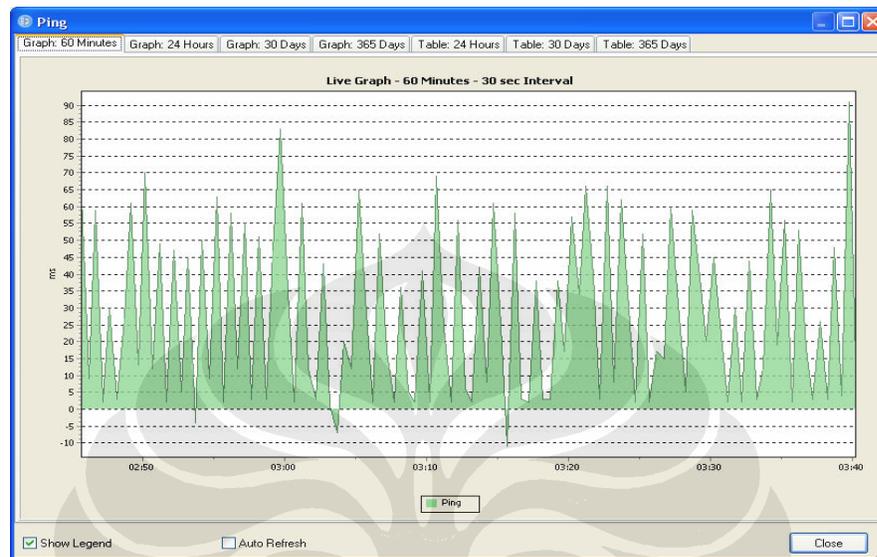


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client E* pada pengujian kedua.

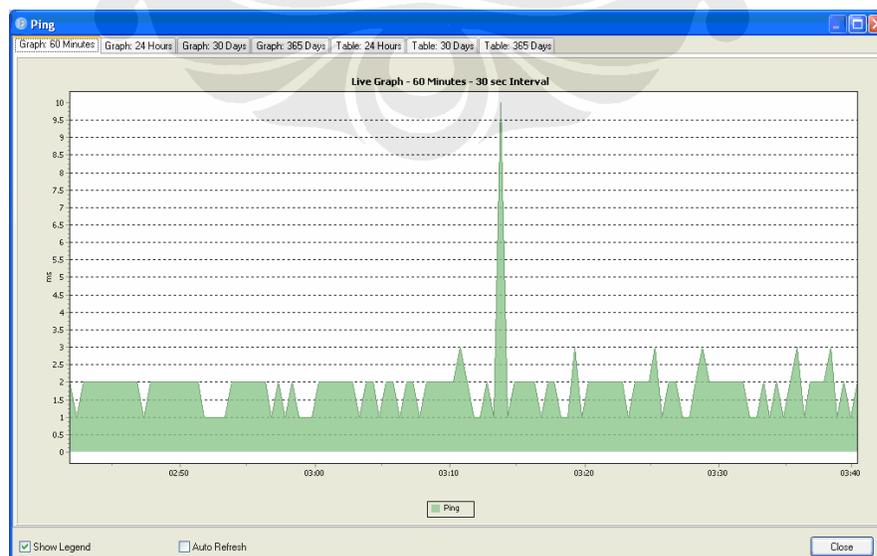


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client A* pada pengujian ketiga.

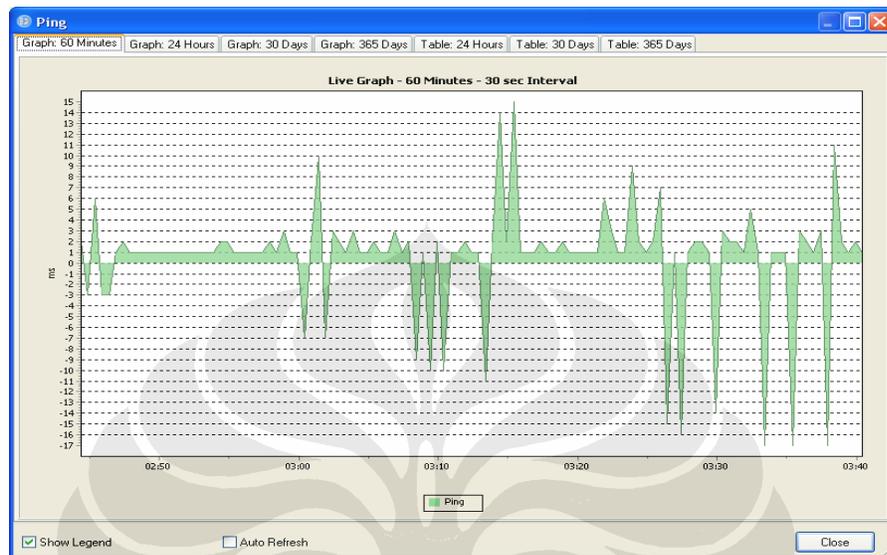


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client B* pada pengujian ketiga.

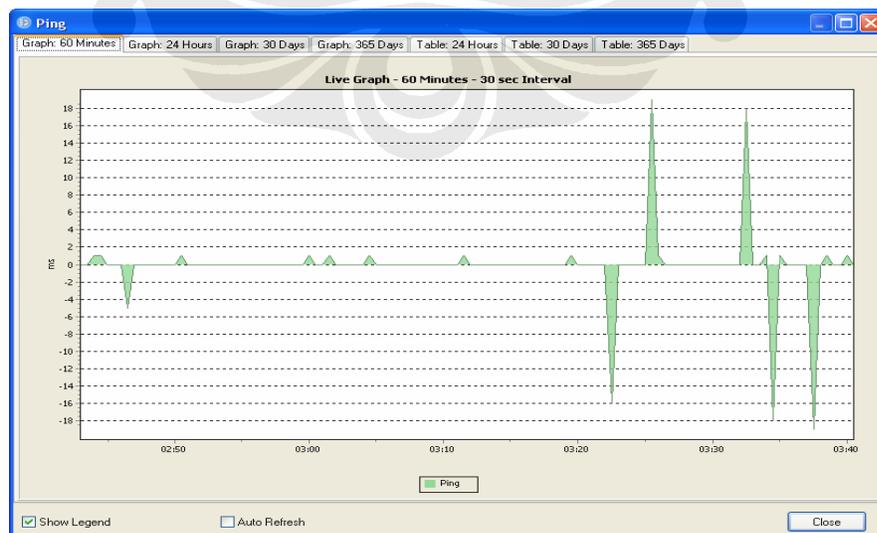


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh router C* pada pengujian ketiga.

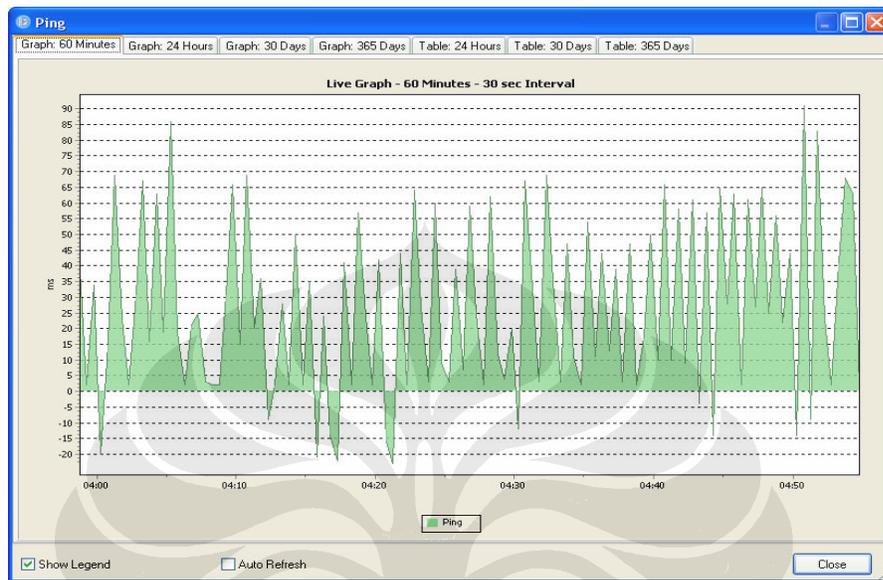


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client E* pada pengujian ketiga.

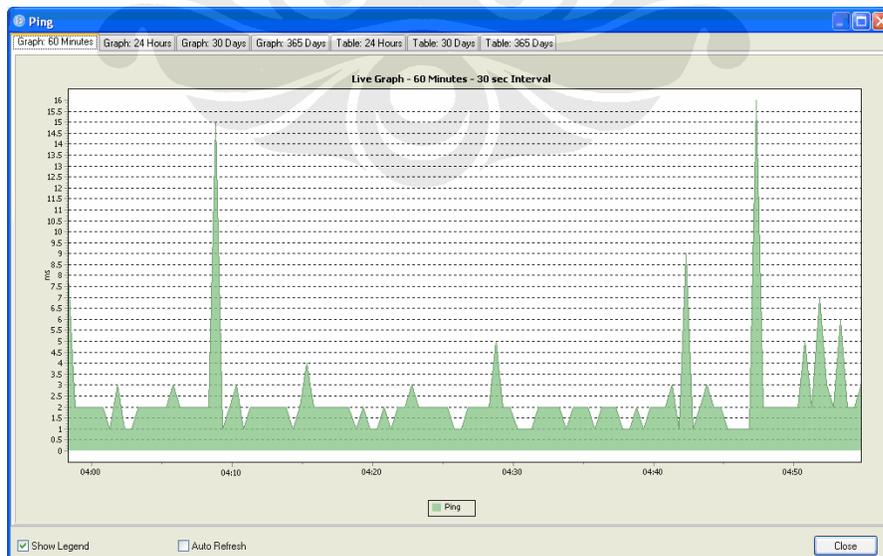


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client* A pada pengujian keempat.

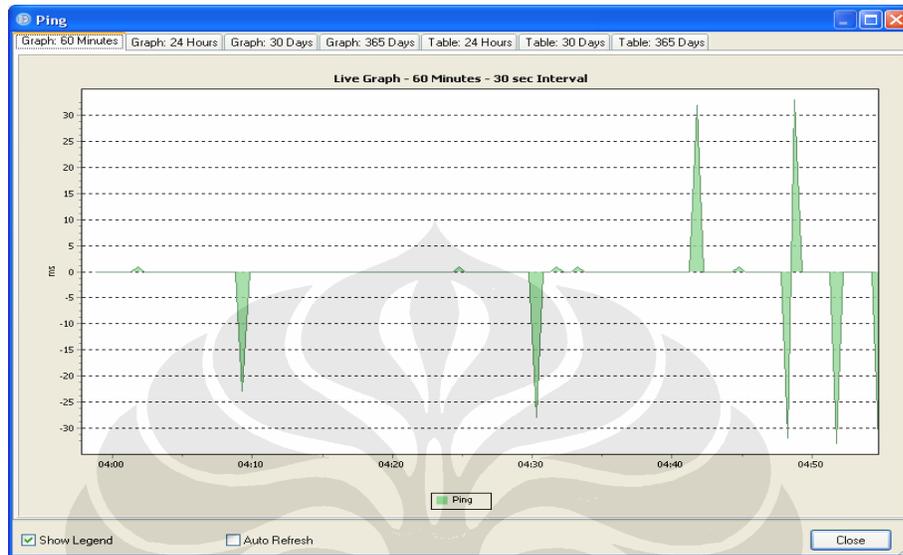


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client* B pada pengujian keempat.

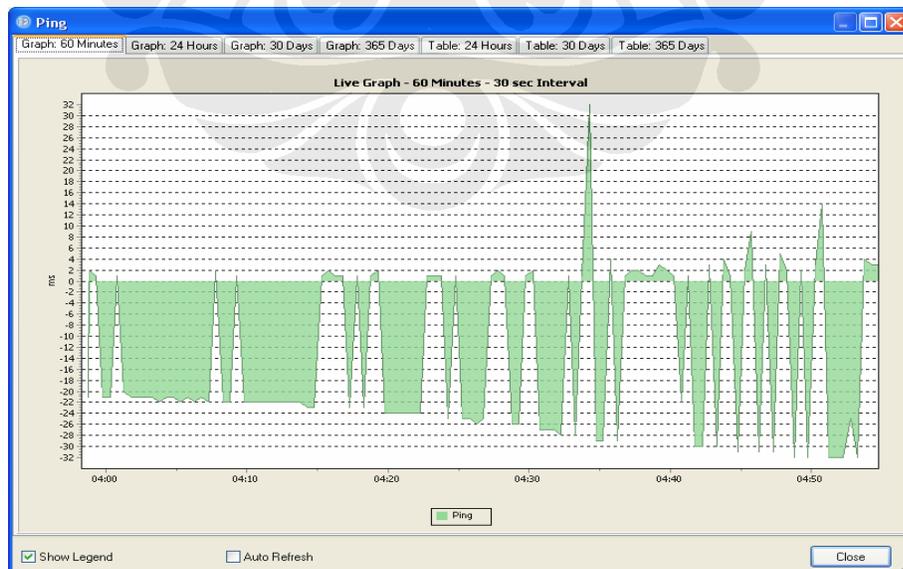


(lanjutan)

Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh router C* pada pengujian keempat.

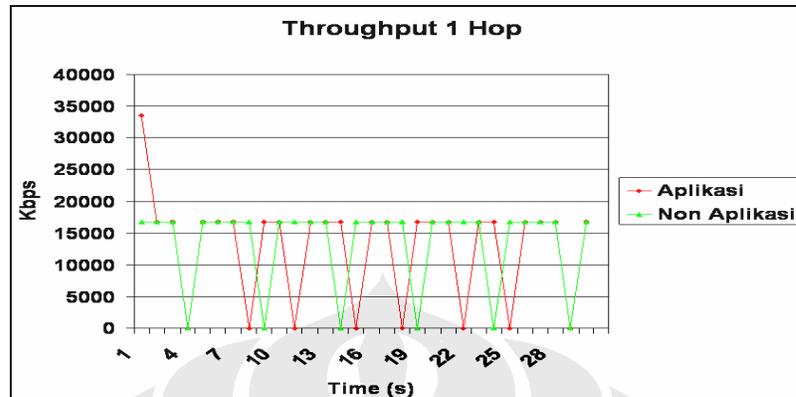


Grafik pembacaan PRTG untuk pengamatan *latency* pada *mesh client E* pada pengujian keempat.

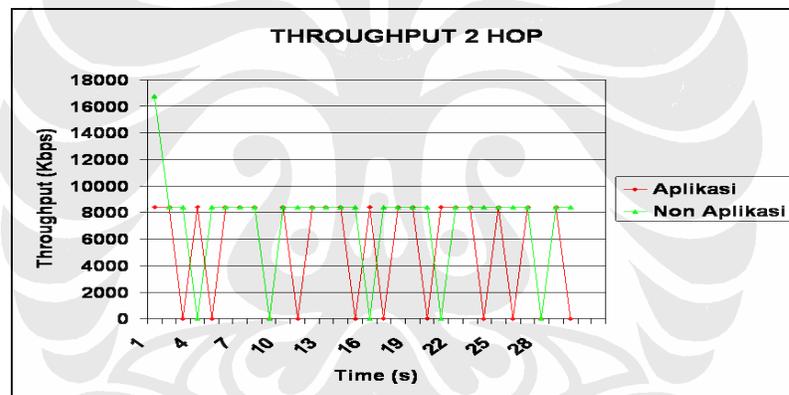


LAMPIRAN 5: Grafik Hasil Pengujian *Throughput* Pada Sistem *Multihop*.

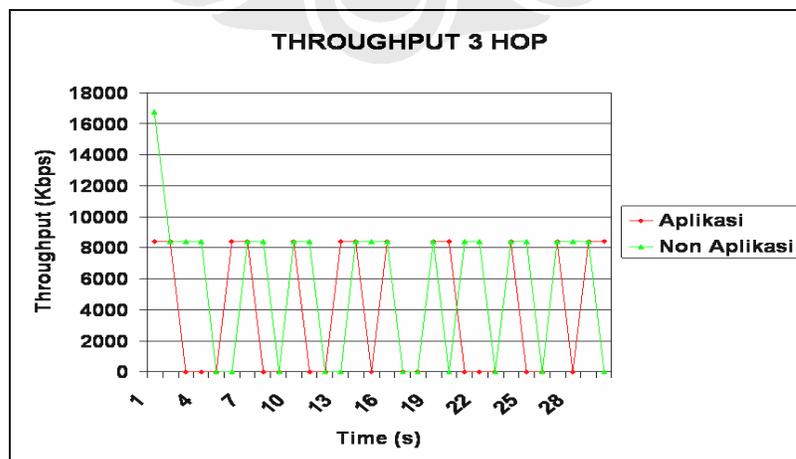
Grafik *throughput* hasil pengujian 1 *hop*.



Grafik *throughput* hasil pengujian 2 *hop*.

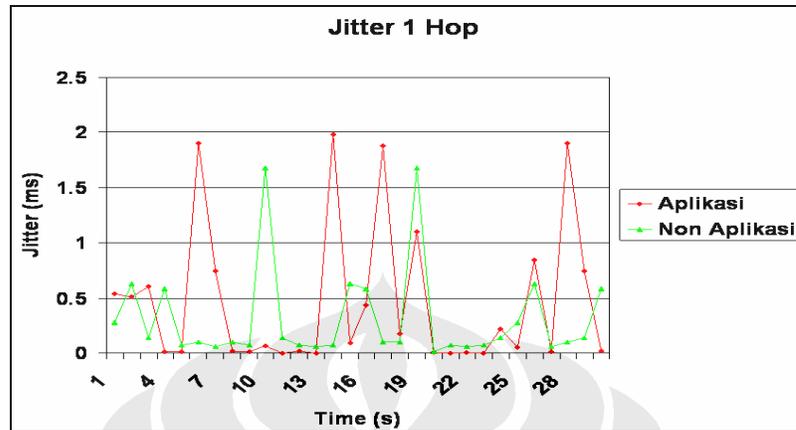


Grafik *throughput* hasil pengujian 3 *hop*.

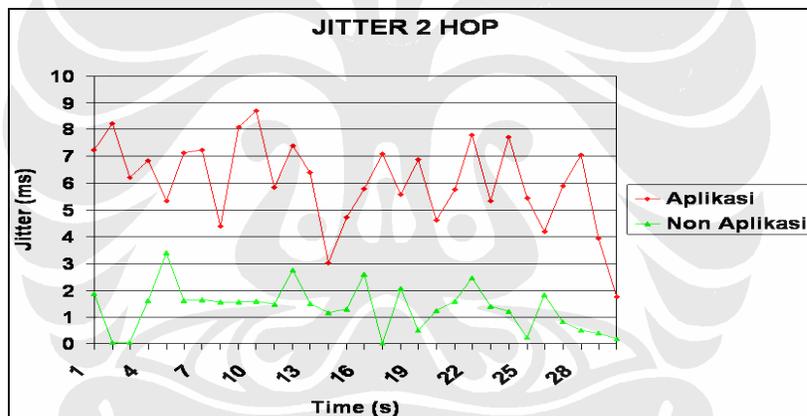


LAMPIRAN 6: Grafik Hasil Pengujian *Jitter* Pada Sistem *Multihop*.

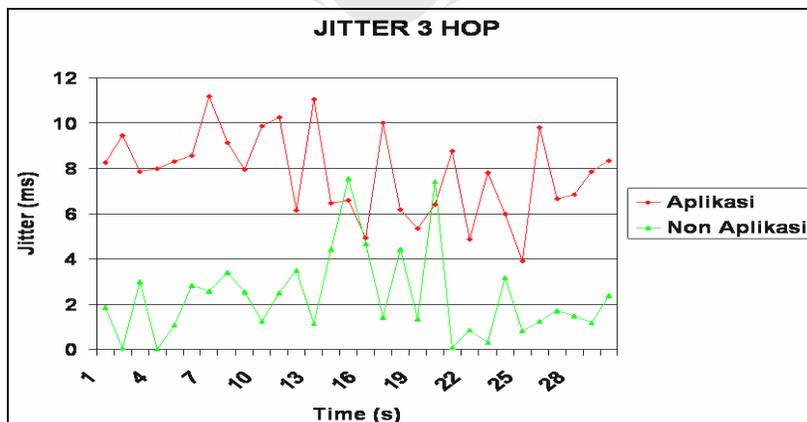
Grafik *Jitter* hasil pengujian 1 hop.



Grafik *Jitter* hasil pengujian 2 hop.



Grafik *Jitter* hasil pengujian 3 hop.



LAMPIRAN 7: Tabel Data Hasil Pengujian Pemakaian *Bandwidth*.

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian 1.

Port 262148 (Atheros AR5005G Wireless Network Adapter - Packet Scheduler Miniport) on mesh client A (192.168.7.100)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/08/2008 8:40 PM - 8:45 PM	1,623.01	365.167	6,245.28	1,405.15	7,868.30	1,770.31	12
12/08/2008 8:35 PM - 8:40 PM	12,653.34	345.578	49,543.07	1,353.08	62,196.40	1,698.66	100
12/08/2008 8:30 PM - 8:35 PM	12,825.29	350.263	43,750.68	1,194.85	56,575.97	1,545.11	100
12/08/2008 8:25 PM - 8:30 PM	10,906.28	297.884	37,566.87	1,026.07	48,473.15	1,323.95	100
12/08/2008 8:20 PM - 8:25 PM	13,078.71	357.184	41,084.99	1,122.04	54,163.70	1,479.23	100
12/08/2008 8:15 PM - 8:20 PM	6,540.92	178.652	22,262.07	608.045	28,802.98	786.697	100
12/08/2008 8:10 PM - 8:15 PM	764.232	20.873	274.225	7.49	1,038.46	28.362	100
12/08/2008 8:05 PM - 8:10 PM	767.758	20.968	273.527	7.47	1,041.29	28.439	100
12/08/2008 8:00 PM - 8:05 PM	603.533	16.483	237.439	6.485	840.973	22.968	100
12/08/2008 7:55 PM - 8:00 PM	103.538	2.828	123.024	3.36	226.563	6.188	100
12/08/2008 7:50 PM - 7:55 PM	103.066	2.815	120.785	3.299	223.852	6.114	100
12/08/2008 7:45 PM - 7:50 PM	92.383	2.729	107.238	3.167	199.621	5.896	92
Total	60,062.05		201,589.20		261,651.25		
Average	5,314.24	163.452	17,762.40	561.708	11,538.32	725.16	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router* B pada pengujian 1.

Port 3 (eth1) on mesh router B (192.168.7.20)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/08/2008 8:35 PM - 8:40 PM	176.977	4.851	161.721	4.433	338.697	9.284	100
12/08/2008 8:30 PM - 8:35 PM	158.01	4.316	142.627	3.895	300.637	8.211	100
12/08/2008 8:25 PM - 8:30 PM	161.27	4.405	145.956	3.986	307.226	8.391	100
12/08/2008 8:20 PM - 8:25 PM	158.415	4.327	143.291	3.913	301.706	8.24	100
12/08/2008 8:15 PM - 8:20 PM	163.157	4.456	147.87	4.039	311.027	8.495	100
12/08/2008 8:10 PM - 8:15 PM	161.879	4.421	147.052	4.016	308.931	8.438	100
12/08/2008 8:05 PM - 8:10 PM	175.213	4.785	160.309	4.378	335.521	9.164	100
12/08/2008 8:00 PM - 8:05 PM	167.559	4.576	152.035	4.152	319.594	8.728	100
12/08/2008 7:55 PM - 8:00 PM	162.256	4.432	147.153	4.019	309.409	8.451	100
12/08/2008 7:50 PM - 7:55 PM	155.867	4.257	140.895	3.848	296.762	8.105	100
12/08/2008 7:45 PM - 7:50 PM	164.213	4.485	148.721	4.062	312.934	8.547	100
12/08/2008 7:40 PM - 7:45 PM	46.871	5.484	43.369	5.074	90.24	10.558	23
Total	1,851.69		1,681.00		3,532.68		
Average	164.164	4.566	148.957	4.151	156.561	8.718	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router* C pada pengujian 1.

Port 3 (eth1) on mesh router C (192.168.7.10)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/08/2008 8:35 PM - 8:40 PM	12,727.74	347.622	12,817.75	350.08	25,545.49	697.702	100
12/08/2008 8:30 PM - 8:35 PM	12,897.84	352.279	12,984.78	354.654	25,882.62	706.933	100
12/08/2008 8:25 PM - 8:30 PM	10,803.29	295.061	10,877.80	297.096	21,681.09	592.157	100
12/08/2008 8:20 PM - 8:25 PM	13,212.63	360.841	13,302.06	363.283	26,514.69	724.124	100
12/08/2008 8:15 PM - 8:20 PM	6,594.80	180.124	6,615.73	180.696	13,210.53	360.82	100
12/08/2008 8:10 PM - 8:15 PM	806.867	22.037	779.341	21.285	1,586.21	43.321	100
12/08/2008 8:05 PM - 8:10 PM	808.68	22.087	781.305	21.339	1,589.98	43.426	100
12/08/2008 8:00 PM - 8:05 PM	639.157	17.456	607.265	16.585	1,246.42	34.041	100
12/08/2008 7:55 PM - 8:00 PM	145.349	3.97	102.758	2.806	248.106	6.776	100
12/08/2008 7:50 PM - 7:55 PM	144.676	3.951	100.387	2.742	245.063	6.693	100
12/08/2008 7:45 PM - 7:50 PM	143.932	3.931	100.24	2.738	244.172	6.669	100
12/08/2008 7:40 PM - 7:45 PM	32.17	4.136	22.598	2.905	54.768	7.041	21
Total	58,957.14		59,092.01		118,049.15		
Average	5,357.86	134.458	5,371.00	134.684	5,364.43	269.142	4

(lanjutan)

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* E pada pengujian 1.

Port 196611 (Intel(R) PRO/Wireless 3945ABG Network Connection - Packet Scheduler Miniport) on mesh client E (192.168.7.50)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	
12/08/2008 8:35 PM - 8:40 PM	49,283.63	1,346.09	12,686.90	346.518	61,970.52	1,692.60	100
12/08/2008 8:30 PM - 8:35 PM	43,736.16	1,194.53	12,722.76	347.486	56,458.91	1,542.01	100
12/8/2008 8:25 PM - 8:30 PM	36,994.00	1,010.39	10,801.74	295.019	47,795.75	1,305.40	100
12/08/2008 8:20 PM - 8:25 PM	41,752.02	1,140.34	13,301.63	363.296	55,053.65	1,503.63	100
12/08/2008 8:15 PM - 8:20 PM	21,196.27	578.935	6,166.16	168.417	27,362.43	747.351	100
12/08/2008 8:10 PM - 8:15 PM	287.084	7.84	751.221	20.516	1,038.31	28.356	100
12/08/2008 8:05 PM - 8:10 PM	288.006	7.866	775.45	21.179	1,063.46	29.045	100
12/08/2008 8:00 PM - 8:05 PM	245.204	6.697	565.438	15.442	810.642	22.139	100
12/08/2008 7:55 PM - 8:00 PM	136.017	3.715	96.508	2.636	232.524	6.351	100
12/08/2008 7:50 PM - 7:55 PM	135.166	3.692	95.791	2.616	230.957	6.308	100
12/08/2008 7:45 PM - 7:50 PM	134.658	3.678	97.498	2.663	232.156	6.341	100
12/08/2008 7:40 PM - 7:45 PM	51.104	4.019	37.802	2.973	88.905	6.992	35
Total	194,239.31		58,098.89		252,338.21		
Average	17,657.22	442.315	5,279.39	132.397	11,468.30	574.711	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian 2.

Port 262148 (Atheros AR5005G Wireless Network Adapter - Packet Scheduler Miniport) on mesh client A (192.168.7.100)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	
12/09/2008 10:55 PM - 11:00 PM	12,309.762	483.259	27,551.106	1,081.606	39,860.868	1,564.864	70
12/09/2008 10:50 PM - 10:55 PM	95.205	2.600	70.533	1.926	165.738	4.527	100
12/09/2008 10:45 PM - 10:50 PM	14,416.549	393.747	36,542.148	998.044	50,958.697	1,391.791	100
12/09/2008 10:40 PM - 10:45 PM	19,555.662	534.071	55,988.119	1,529.053	75,543.781	2,063.124	100
12/09/2008 10:35 PM - 10:40 PM	14,683.853	401.034	34,307.323	936.975	48,991.176	1,338.009	100
12/09/2008 10:30 PM - 10:25 PM	1,160.375	31.692	339.124	9.262	1,499.499	40.955	100
12/09/2008 10:25 PM - 10:20 PM	1,230.406	33.606	367.808	10.046	1,598.214	43.652	100
12/09/2008 10:20 PM - 10:25 PM	1,155.504	31.560	340.923	9.312	1,496.427	40.872	100
12/09/2008 10:15 PM - 10:20 PM	787.661	21.513	264.404	7.221	1,052.065	28.734	100
12/09/2008 10:10 PM - 10:15 PM	100.909	2.756	108.406	2.961	209.315	5.717	100
12/09/2008 10:05 PM - 10:10 PM	100.581	2.747	109.751	2.998	210.332	5.745	100
12/09/2008 10:00 PM - 10:05 PM	107.474	2.935	115.659	3.159	223.133	6.094	100
12/09/2008 09:55 PM - 10:00 PM	15.585	3.204	16.454	3.382	32.039	6.586	13
Total	65,719.526		156,121.76		221,841.28		
Average	5,055.348	149.594	12,009.37	353.534	17,064.71	503.128	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router* B pada pengujian 2.

Port 3 (eth1) on mesh router B (192.168.7.20)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	
12/09/2008 10:50 PM - 10:55 PM	156.876	4.293	141.22	3.864	298.096	8.157	100
12/09/2008 10:45 PM - 10:50 PM	172.766	4.719	157.118	4.291	329.884	9.01	100
12/09/2008 10:40 PM - 10:45 PM	160.272	4.377	144.651	3.951	304.924	8.328	100
12/09/2008 10:35 PM - 10:40 PM	168.729	4.608	153.363	4.189	322.092	8.797	100
12/09/2008 10:30 PM - 10:25 PM	170.065	4.644	154.412	4.217	324.478	8.861	100
12/09/2008 10:25 PM - 10:20 PM	177.574	4.85	161.909	4.422	339.483	9.272	100
12/09/2008 10:20 PM - 10:25 PM	158.674	4.334	143.285	3.914	301.959	8.247	100
12/09/2008 10:15 PM - 10:20 PM	176.187	4.812	160.286	4.378	336.473	9.189	100
12/09/2008 10:10 PM - 10:15 PM	172.292	4.706	156.718	4.28	329.01	8.986	100
12/09/2008 10:05 PM - 10:10 PM	204.348	5.581	188.985	5.161	393.333	10.742	100
12/09/2008 10:00 PM - 10:05 PM	175.386	4.79	159.542	4.358	334.928	9.148	100
12/09/2008 09:55 PM - 10:00 PM	11.193	4.092	10.037	3.669	21.23	7.761	7
Total	1,904.36		1,731.53		3,635.89		
Average	172.166	4.65	156.553	4.224	164.359	8.875	4

(lanjutan)

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router C* pada pengujian 2.

	Port 3 (eth1) on mesh router C (192.168.7.10)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/09/2008 10:55 PM - 11:00 PM	28,193.574	1,110.927	28,303.521	1,115.259	56,497.096	2,226.187	69
12/09/2008 10:50 PM - 10:55 PM	112.226	3.065	76.515	2.090	188.740	5.155	100
12/09/2008 10:45 PM - 10:50 PM	35,703.366	975.135	35,837.592	978.801	71,540.958	1,953.936	100
12/09/2008 10:40 PM - 10:45 PM	56,085.396	1,531.658	56,319.835	1,538.061	112,405.230	3,069.719	100
12/09/2008 10:35 PM - 10:40 PM	35,163.574	960.360	35,296.993	964.004	70,460.567	1,924.364	100
12/09/2008 10:30 PM - 10:25 PM	377.859	10.320	354.784	9.690	732.644	20.011	100
12/09/2008 10:25 PM - 10:20 PM	409.438	11.182	386.204	10.548	795.642	21.730	100
12/09/2008 10:20 PM - 10:25 PM	380.241	10.385	356.320	9.732	736.562	20.117	100
12/09/2008 10:15 PM - 10:20 PM	308.849	8.435	281.396	7.686	590.245	16.121	100
12/09/2008 10:10 PM - 10:15 PM	148.923	4.067	116.592	3.184	265.515	7.252	100
12/09/2008 10:05 PM - 10:10 PM	149.774	4.091	117.725	3.215	267.499	7.306	100
12/09/2008 10:00 PM - 10:05 PM	151.766	4.208	119.699	3.319	271.465	7.527	98
Total	157,184.99		157,567.18		314,752.16		
Average	13,098.75	386.153	13,130.60	387.132	26,229.35	773.285	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client E* pada pengujian 2.

	Port 196611 (Intel(R) PRO/Wireless 3945ABG Network Connection - Packet Scheduler Miniport) on mesh client E (192.168.7.50)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/09/2008 10:55 PM - 11:00 PM	26,998.360	1,061.993	12,079.921	475.169	39,078.281	1,537.162	69
12/09/2008 10:50 PM - 10:55 PM	66.106	2.257	57.964	1.979	124.070	4.237	80
12/09/2008 10:45 PM - 10:50 PM	36,998.687	1,010.513	14,582.147	398.269	51,580.834	1,408.782	100
12/09/2008 10:40 PM - 10:45 PM	55,992.687	1,529.279	19,534.328	533.524	75,527.015	2,062.804	100
12/09/2008 10:35 PM - 10:40 PM	33,698.598	920.319	14,503.314	396.090	48,201.912	1,316.409	100
12/09/2008 10:30 PM - 10:25 PM	349.270	9.540	1,143.772	31.240	1,493.042	40.780	100
12/09/2008 10:25 PM - 10:20 PM	377.709	10.316	1,213.134	33.134	1,590.843	43.451	100
12/09/2008 10:20 PM - 10:25 PM	351.661	9.604	1,140.204	31.140	1,491.865	40.745	100
12/09/2008 10:15 PM - 10:20 PM	272.448	7.441	760.735	20.777	1,033.184	28.218	100
12/09/2008 10:10 PM - 10:15 PM	119.932	3.276	84.510	2.308	204.441	5.584	100
12/09/2008 10:05 PM - 10:10 PM	94.758	3.228	67.061	2.285	161.818	5.513	80
12/09/2008 10:00 PM - 10:05 PM	106.680	3.399	76.747	2.446	183.427	5.845	86
Total	155,426.90		65,243.84		220,670.73		
Average	12,952.24	380.930	5,436.99	160.697	18,389.23	541.628	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client A* pada pengujian 3.

	Port 196610 (Atheros AR5005G Wireless Network Adapter - Packet Scheduler Miniport) on mesh client A (192.168.7.100)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/10/2008 3:35 AM - 3:40 AM	14,355.95	392.079	27,998.50	764.673	42,354.46	1,156.75	100
12/10/2008 3:30 AM - 3:35 AM	10,189.22	278.271	27,767.99	758.352	37,957.21	1,036.62	100
12/10/2008 3:25 AM - 3:30 AM	17,821.40	486.724	39,140.93	1,068.99	56,962.32	1,555.71	100
12/10/2008 3:20 AM - 3:25 AM	16,488.46	450.32	27,926.90	762.718	44,415.36	1,213.04	100
12/10/2008 3:15 AM - 3:20 AM	796.37	21.75	189.793	5.183	986.163	26.933	100
12/10/2008 3:10 AM - 3:15 AM	789.757	21.569	183.125	5.001	972.882	26.571	100
12/10/2008 3:05 AM - 3:10 AM	775.225	21.172	172.488	4.711	947.713	25.882	100
12/10/2008 3:00 AM - 3:05 AM	698.328	19.072	167.96	4.587	866.288	23.659	100
12/10/2008 2:55 AM - 3:00 AM	93.702	2.559	104.809	2.863	198.511	5.422	100
12/10/2008 2:50 AM - 2:55 AM	94.569	2.583	103.848	2.836	198.417	5.419	100
12/10/2008 2:45 AM - 2:50 AM	94.969	2.594	104.16	2.845	199.129	5.438	100
12/10/2008 2:40 AM - 2:45 AM	53.788	2.93	59.23	3.226	113.019	6.156	50
Total	62,251.74		123,919.73		186,171.47		
Average	5,655.27	141.802	11,261.84	282.165	8,458.55	423.967	4

(lanjutan)

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian 3.

Port 3 (eth1) on mesh router B (192.168.7.20)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/10/2008 3:35 AM - 3:40 AM	149.247	4.076	133.833	3.655	283.08	7.731	100
12/10/2008 3:30 AM - 3:35 AM	108.982	2.977	93.276	2.548	202.259	5.524	100
12/10/2008 3:25 AM - 3:30 AM	107.993	2.949	92.568	2.528	200.562	5.477	100
12/10/2008 3:20 AM - 3:25 AM	123.266	3.367	107.715	2.942	230.98	6.308	100
12/10/2008 3:15 AM - 3:20 AM	168.274	4.596	152.521	4.166	320.796	8.762	100
12/10/2008 3:10 AM - 3:15 AM	148.307	4.05	132.799	3.627	281.105	7.677	100
12/10/2008 3:05 AM - 3:10 AM	123.798	3.381	108.154	2.954	231.952	6.335	100
12/10/2008 3:00 AM - 3:05 AM	155.452	4.246	139.817	3.819	295.27	8.064	100
12/10/2008 2:55 AM - 3:00 AM	158.661	4.333	142.938	3.904	301.599	8.237	100
12/10/2008 2:50 AM - 2:55 AM	143.789	3.927	128.122	3.499	271.911	7.426	100
12/10/2008 2:45 AM - 2:50 AM	128.023	3.497	112.593	3.075	240.616	6.572	100
12/10/2008 2:40 AM - 2:45 AM	172.268	5.064	157.552	4.631	329.819	9.695	93
Total	1,688.06		1,501.89		3,189.95		
Average	141.792	3.872	126.181	3.446	133.986	7.317	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router* C pada pengujian 3.

Port 3 (eth1) on mesh router C (192.168.7.10)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/10/2008 3:40 AM - 3:45 AM	9,400.46	1,728.59	9,444.19	1,736.63	18,844.65	3,465.22	15
12/10/2008 3:35 AM - 3:40 AM	43,907.88	1,199.26	44,083.95	1,204.07	87,991.83	2,403.32	100
12/10/2008 3:30 AM - 3:35 AM	36,926.03	1,008.53	37,048.14	1,011.86	73,974.17	2,020.39	100
12/10/2008 3:25 AM - 3:30 AM	56,861.75	1,553.12	57,126.94	1,560.36	113,988.69	3,113.48	100
12/10/2008 3:20 AM - 3:25 AM	45,868.33	1,252.80	46,087.48	1,258.79	91,955.81	2,511.59	100
12/10/2008 3:15 AM - 3:20 AM	1,035.43	28.279	1,029.94	28.129	2,065.37	56.408	100
12/10/2008 3:10 AM - 3:15 AM	1,034.75	28.26	1,028.56	28.091	2,063.31	56.352	100
12/10/2008 3:05 AM - 3:10 AM	1,017.80	27.798	1,010.13	27.589	2,027.93	55.387	100
12/10/2008 3:00 AM - 3:05 AM	956.917	26.135	949.601	25.935	1,906.52	52.069	100
12/10/2008 2:55 AM - 3:00 AM	277.856	7.589	251.266	6.862	529.122	14.451	100
12/10/2008 2:50 AM - 2:55 AM	282.886	7.727	256.381	7.003	539.267	14.73	100
12/10/2008 2:45 AM - 2:50 AM	319.555	8.727	293.792	8.024	613.347	16.751	100
12/10/2008 2:40 AM - 2:45 AM	186.67	8.542	171.153	7.832	357.823	16.374	60
Total	198,076.31		198,781.52		396,857.82		
Average	17,139.42	529.642	17,200.98	531.629	17,170.20	1,061.27	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* E pada pengujian 3.

Port 196610 (Atheros AR5007EG Wireless Network Adapter - Packet Scheduler Miniport) on mesh client E (192.168.7.50)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/10/2008 3:35 AM - 3:40 AM	28,109.13	767.694	14,317.34	391.024	42,426.47	1,158.72	100
12/10/2008 3:30 AM - 3:35 AM	28,382.27	775.206	10,327.93	282.087	38,710.20	1,057.29	100
12/10/2008 3:25 AM - 3:30 AM	39,038.14	1,066.22	18,045.95	492.873	57,084.09	1,559.09	100
12/10/2008 3:20 AM - 3:25 AM	27,804.54	759.427	16,466.75	449.757	44,271.29	1,209.18	100
12/10/2008 3:15 AM - 3:20 AM	272.081	7.431	843.021	23.024	1,115.10	30.455	100
12/10/2008 3:10 AM - 3:15 AM	284.183	7.761	855.041	23.352	1,139.22	31.114	100
12/10/2008 3:05 AM - 3:10 AM	278.087	7.595	848.163	23.164	1,126.25	30.759	100
12/10/2008 3:00 AM - 3:05 AM	280.35	7.657	775.742	21.186	1,056.09	28.843	100
12/10/2008 2:55 AM - 3:00 AM	218.059	5.955	177.721	4.854	395.779	10.809	100
12/10/2008 2:50 AM - 2:55 AM	223.038	6.092	183.372	5.008	406.41	11.1	100
12/10/2008 2:45 AM - 2:50 AM	259.117	7.077	218.938	5.979	478.056	13.056	100
12/10/2008 2:40 AM - 2:45 AM	176.124	6.92	146.491	5.756	322.615	12.676	70
Total	125,325.11		63,206.47		188,531.57		
Average	11,379.54	285.419	5,733.90	144.005	8,556.72	429.425	4

(lanjutan)

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* A pada pengujian 4.

	Port 196610 (Atheros AR5005G Wireless Network Adapter - Packet Scheduler Miniport) on mesh client A (192.168.7.100)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/11/2008 4:50 AM - 4:55 AM	20,552.87	561.324	37,458.22	1,023.03	58,011.08	1,584.35	100
12/11/2008 4:45 AM - 4:50 AM	20,514.07	560.246	37,911.25	1,035.37	58,425.32	1,595.61	100
12/11/2008 4:40 AM - 4:45 AM	18,966.88	518.009	37,866.06	1,034.17	56,832.95	1,552.18	100
12/11/2008 4:35 AM - 4:40 AM	13,370.55	365.166	24,376.35	665.748	37,746.90	1,030.91	100
12/11/2008 4:30 AM - 4:35 AM	779.942	21.301	176.514	4.821	956.456	26.122	100
12/11/2008 4:25 AM - 4:30 AM	777.38	21.23	174.251	4.759	951.631	25.989	100
12/11/2008 4:20 AM - 4:25 AM	770.959	21.055	173.408	4.736	944.367	25.791	100
12/11/2008 4:15 AM - 4:20 AM	686.131	18.738	164.153	4.483	850.284	23.222	100
12/11/2008 4:10 AM - 4:15 AM	100.164	2.736	112.851	3.082	213.015	5.818	100
12/11/2008 4:05 AM - 4:10 AM	100.758	2.752	114.884	3.138	215.642	5.889	100
12/11/2008 4:00 AM - 4:05 AM	101.44	2.77	114.938	3.139	216.379	5.909	100
12/11/2008 3:55 AM - 4:00 AM	44.006	2.768	48.31	3.039	92.315	5.807	43
Total	76,765.15		138,691.19		215,456.34		
Average	6,975.74	174.841	12,605.88	315.792	9,790.81	490.634	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* B pada pengujian 4.

	Port 3 (eth1) on mesh router B (192.168.7.20)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/11/2008 4:50 AM - 4:55 AM	118.194	4.022	106.207	3.614	224.401	7.636	80
12/11/2008 4:45 AM - 4:50 AM	163.37	4.462	147.932	4.04	311.302	8.502	100
12/11/2008 4:40 AM - 4:45 AM	160.461	4.362	145.431	3.972	305.892	8.364	100
12/11/2008 4:35 AM - 4:40 AM	158.818	4.398	143.777	3.927	302.596	8.264	100
12/11/2008 4:30 AM - 4:35 AM	148.538	4.057	132.986	3.632	281.524	7.689	100
12/11/2008 4:25 AM - 4:30 AM	126.519	4.324	114.348	3.908	240.866	8.232	80
12/11/2008 4:20 AM - 4:25 AM	149.104	4.072	133.8	3.654	282.904	7.726	100
12/11/2008 4:15 AM - 4:20 AM	154.246	4.213	138.997	3.796	293.243	8.009	100
12/11/2008 4:10 AM - 4:15 AM	148.442	4.054	133.145	3.636	281.587	7.69	100
12/11/2008 4:05 AM - 4:10 AM	153.227	4.185	138.115	3.772	291.342	7.956	100
12/11/2008 4:00 AM - 4:05 AM	125.266	3.421	109.846	3	235.111	6.421	100
12/11/2008 3:55 AM - 4:00 AM	166.71	4.819	152.299	4.402	319.009	9.221	94
Total	1,772.90		1,596.88		3,369.78		
Average	153.65	4.196	138.409	3.779	146.03	7.975	4

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh router* C pada pengujian 4.

	Port 3 (eth1) on mesh router C (192.168.7.10)						
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/11/2008 4:50 AM - 4:55 AM	56,953.27	1,555.57	57,168.04	1,561.43	114,121.31	3,117.00	100
12/11/2008 4:45 AM - 4:50 AM	58,570.51	1,599.63	58,815.19	1,606.31	117,385.69	3,205.95	100
12/11/2008 4:40 AM - 4:45 AM	56,897.04	1,554.03	57,125.18	1,560.26	114,022.22	3,114.29	100
12/11/2008 4:35 AM - 4:40 AM	37,950.40	1,036.51	38,096.05	1,040.48	76,046.45	2,076.99	100
12/11/2008 4:30 AM - 4:35 AM	1,027.81	28.072	1,021.27	27.893	2,049.08	55.965	100
12/11/2008 4:25 AM - 4:30 AM	1,047.19	28.6	1,040.93	28.429	2,088.12	57.029	100
12/11/2008 4:20 AM - 4:25 AM	1,030.92	28.156	1,024.34	27.976	2,055.26	56.132	100
12/11/2008 4:15 AM - 4:20 AM	928.394	25.356	918.748	25.093	1,847.14	50.449	100
12/11/2008 4:10 AM - 4:15 AM	368.966	10.077	343.999	9.395	712.965	19.472	100
12/11/2008 4:05 AM - 4:10 AM	350.039	9.56	324.283	8.857	674.322	18.417	100
12/11/2008 4:00 AM - 4:05 AM	417.851	11.413	393.917	10.759	811.768	22.172	100
12/11/2008 3:55 AM - 4:00 AM	234.526	12.03	221.24	11.348	455.767	23.378	53
Total	215,776.93		216,493.18		432,270.11		
Average	19,598.84	491.583	19,665.18	493.187	19,632.01	984.77	4

(lanjutan)

Tabel pembacaan PRTG untuk pemakaian *bandwidth* pada *mesh client* E pada pengujian 4.

Port 196610 (Atheros AR5007EG Wireless Network Adapter - Packet Scheduler Miniport) on mesh client E (192.168.7.50)							
	Bandwidth Traffic IN		Bandwidth Traffic OUT		Sum		Coverage
	kbyte	kbit/second	kbyte	kbit/second	kbyte	kbit/second	%
12/11/2008 4:50 AM - 4:55 AM	37,591.66	1,026.78	20,707.97	565.616	58,299.63	1,592.39	100
12/11/2008 4:45 AM - 4:50 AM	38,086.71	1,040.23	20,686.45	564.991	58,773.16	1,605.22	100
12/11/2008 4:40 AM - 4:45 AM	37,967.30	1,036.97	19,083.72	521.217	57,051.02	1,558.19	100
12/11/2008 4:35 AM - 4:40 AM	24,544.07	670.351	13,517.92	369.203	38,061.98	1,039.55	100
12/11/2008 4:30 AM - 4:35 AM	284.689	7.775	855.367	23.361	1,140.06	31.136	100
12/11/2008 4:25 AM - 4:30 AM	305.979	8.357	876.523	23.939	1,182.50	32.296	100
12/11/2008 4:20 AM - 4:25 AM	297.587	8.127	862.973	23.568	1,160.56	31.695	100
12/11/2008 4:15 AM - 4:20 AM	277.321	7.574	766.739	20.941	1,044.06	28.515	100
12/11/2008 4:10 AM - 4:15 AM	304.132	8.306	260.427	7.113	564.559	15.419	100
12/11/2008 4:05 AM - 4:10 AM	284.142	7.76	239.203	6.533	523.345	14.293	100
12/11/2008 4:00 AM - 4:05 AM	351.934	9.612	307.041	8.386	658.975	17.998	100
12/11/2008 3:55 AM - 4:00 AM	227.501	9.797	200.429	8.631	427.93	18.427	63
Total	140,523.02		78,364.76		218,887.78		
Average	12,756.91	320.136	7,107.38	178.625	9,932.15	498.761	4

LAMPIRAN 8: Tabel Data Hasil Pengujian *Latency*.Tabel pembacaan PRTG untuk *latency* pada pengujian 1.

LATENCY							
Mesh Client A (192.168.7.100)		Mesh Router B (192.168.7.20)		Mesh Router C (192.168.7.10)		Mesh Client E (192.168.7.50)	
Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)
12/08/2008 8:35 PM - 8:40 PM	3.8	12/08/2008 8:35 PM - 8:40 PM	2	12/08/2008 8:40 PM - 8:45 PM	2	12/08/2008 8:35 PM - 8:40 PM	0.2
12/08/2008 8:30 PM - 8:35 PM	9.5	12/08/2008 8:30 PM - 8:35 PM	3.3	12/08/2008 8:35 PM - 8:40 PM	1.5	12/08/2008 8:30 PM - 8:35 PM	0.2
12/08/2008 8:25 PM - 8:30 PM	8.4	12/08/2008 8:25 PM - 8:30 PM	1.5	12/08/2008 8:30 PM - 8:35 PM	3.1	12/08/2008 8:25 PM - 8:30 PM	0.3
12/08/2008 8:20 PM - 8:25 PM	2.6	12/08/2008 8:20 PM - 8:25 PM	3.1	12/08/2008 8:25 PM - 8:30 PM	2.1	12/08/2008 8:20 PM - 8:25 PM	0.1
12/08/2008 8:15 PM - 8:20 PM	2.7	12/08/2008 8:15 PM - 8:20 PM	1.6	12/08/2008 8:20 PM - 8:25 PM	1.9	12/08/2008 8:15 PM - 8:20 PM	0.2
12/08/2008 8:10 PM - 8:15 PM	1.7	12/08/2008 8:10 PM - 8:15 PM	1.5	12/08/2008 8:15 PM - 8:20 PM	2.2	12/08/2008 8:10 PM - 8:15 PM	0.1
12/08/2008 8:05 PM - 8:10 PM	1.5	12/08/2008 8:05 PM - 8:10 PM	2	12/08/2008 8:10 PM - 8:15 PM	1.5	12/08/2008 8:05 PM - 8:10 PM	0.1
12/08/2008 8:00 PM - 8:05 PM	1.6	12/08/2008 8:00 PM - 8:05 PM	2	12/08/2008 8:05 PM - 8:10 PM	1.4	12/08/2008 8:00 PM - 8:05 PM	0.1
12/08/2008 7:55 PM - 8:00 PM	8.2	12/08/2008 7:55 PM - 8:00 PM	1.8	12/08/2008 8:00 PM - 8:05 PM	11.6	12/08/2008 7:55 PM - 8:00 PM	0.1
12/08/2008 7:50 PM - 7:55 PM	1.6	12/08/2008 7:50 PM - 7:55 PM	1.7	12/08/2008 7:55 PM - 8:00 PM	10.1	12/08/2008 7:50 PM - 7:55 PM	0.1
12/08/2008 7:45 PM - 7:50 PM	1.6	12/08/2008 7:45 PM - 7:50 PM	1.545	12/08/2008 7:50 PM - 7:55 PM	1.2	12/08/2008 7:45 PM - 7:50 PM	0.4
		12/08/2008 7:40 PM - 7:45 PM	1	12/08/2008 7:45 PM - 7:50 PM	1	12/08/2008 7:40 PM - 7:45 PM	0.25
				12/08/2008 7:40 PM - 7:45 PM	1		
Average	3.927	Average	1.92	Average	3.138	Average	0.179

Tabel pembacaan PRTG untuk *latency* pada pengujian 2.

LATENCY							
Mesh Client A (192.168.7.100)		Mesh Router B (192.168.7.20)		Mesh Router C (192.168.7.10)		Mesh Client E (192.168.7.50)	
Time	Ping Latency (ms)						
12/09/2008 10:55 PM - 11:00 PM	5	12/09/2008 10:50 PM - 10:55 PM	1	12/09/2008 10:55 PM - 11:00 PM	4	12/09/2008 10:55 PM - 11:00 PM	0
12/09/2008 10:50 PM - 10:55 PM	3.67	12/09/2008 10:45 PM - 10:50 PM	1.6	12/09/2008 10:50 PM - 10:55 PM	3.2	12/09/2008 10:50 PM - 10:55 PM	0.1
12/09/2008 10:45 PM - 10:50 PM	2.8	12/09/2008 10:40 PM - 10:45 PM	1.8	12/09/2008 10:45 PM - 10:50 PM	6.1	12/09/2008 10:45 PM - 10:50 PM	0.1
12/09/2008 10:40 PM - 10:45 PM	14.4	12/09/2008 10:35 PM - 10:40 PM	1.8	12/09/2008 10:40 PM - 10:45 PM	9.9	12/09/2008 10:40 PM - 10:45 PM	0.1
12/09/2008 10:35 PM - 10:40 PM	12.1	12/09/2008 10:30 PM - 10:25 PM	1.7	12/09/2008 10:35 PM - 10:40 PM	3.9	12/09/2008 10:35 PM - 10:40 PM	0
12/09/2008 10:30 PM - 10:25 PM	1.8	12/09/2008 10:25 PM - 10:20 PM	2	12/09/2008 10:30 PM - 10:25 PM	2	12/09/2008 10:30 PM - 10:25 PM	0.3
12/09/2008 10:25 PM - 10:20 PM	2	12/09/2008 10:20 PM - 10:25 PM	1.4	12/09/2008 10:25 PM - 10:20 PM	1.8	12/09/2008 10:25 PM - 10:20 PM	0
12/09/2008 10:20 PM - 10:25 PM	1.8	12/09/2008 10:15 PM - 10:20 PM	1.8	12/09/2008 10:20 PM - 10:25 PM	1.7	12/09/2008 10:20 PM - 10:25 PM	0.2
12/09/2008 10:15 PM - 10:20 PM	3.3	12/09/2008 10:10 PM - 10:15 PM	1.6	12/09/2008 10:15 PM - 10:20 PM	1.7	12/09/2008 10:15 PM - 10:20 PM	0
12/09/2008 10:10 PM - 10:15 PM	1.78	12/09/2008 10:05 PM - 10:10 PM	1.6	12/09/2008 10:10 PM - 10:15 PM	2	12/09/2008 10:10 PM - 10:15 PM	0.1
12/09/2008 10:05 PM - 10:10 PM	1.9	12/09/2008 10:00 PM - 10:05 PM	1.727	12/09/2008 10:05 PM - 10:10 PM	1.5	12/09/2008 10:05 PM - 10:10 PM	0.3
12/09/2008 10:00 PM - 10:05 PM	1.78			12/09/2008 10:00 PM - 10:05 PM	1.5	12/09/2008 10:00 PM - 10:05 PM	0.111
						12/09/2008 09:55 PM - 10:00 PM	0
Average	4.360	Average	1.639	Average	3.275	Average	0.101

(lanjutan)

Tabel pembacaan PRTG untuk *latency* pada pengujian 3.

LATENCY							
Mesh Client A (192.168.7.100)		Mesh Router B (192.168.7.20)		Mesh Router C (192.168.7.10)		Mesh Client E (192.168.7.50)	
Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)
12/10/2008 3:35 AM - 3:40 AM	30.4	12/10/2008 3:35 AM - 3:40 AM	1.9	12/10/2008 3:40 AM - 3:45 AM	1	12/10/2008 3:40 AM - 3:45 AM	0
12/10/2008 3:30 AM - 3:35 AM	24.4	12/10/2008 3:30 AM - 3:35 AM	1.6	12/10/2008 3:35 AM - 3:40 AM	-1	12/10/2008 3:35 AM - 3:40 AM	-1.7
12/10/2008 3:25 AM - 3:30 AM	30	12/10/2008 3:25 AM - 3:30 AM	1.9	12/10/2008 3:30 AM - 3:35 AM	-1.5	12/10/2008 3:30 AM - 3:35 AM	0.1
12/10/2008 3:20 AM - 3:25 AM	36.8	12/10/2008 3:20 AM - 3:25 AM	1.9	12/10/2008 3:25 AM - 3:30 AM	-1.4	12/10/2008 3:25 AM - 3:30 AM	2
12/10/2008 3:15 AM - 3:20 AM	17.6	12/10/2008 3:15 AM - 3:20 AM	1.7	12/10/2008 3:20 AM - 3:25 AM	2.6	12/10/2008 3:20 AM - 3:25 AM	-1.6
12/10/2008 3:10 AM - 3:15 AM	27.2	12/10/2008 3:10 AM - 3:15 AM	2.5	12/10/2008 3:15 AM - 3:20 AM	2.7	12/10/2008 3:15 AM - 3:20 AM	0.1
12/10/2008 3:05 AM - 3:10 AM	25.5	12/10/2008 3:05 AM - 3:10 AM	1.8	12/10/2008 3:10 AM - 3:15 AM	0.3	12/10/2008 3:10 AM - 3:15 AM	0.1
12/10/2008 3:00 AM - 3:05 AM	18.2	12/10/2008 3:00 AM - 3:05 AM	1.8	12/10/2008 3:05 AM - 3:10 AM	-0.7	12/10/2008 3:05 AM - 3:10 AM	0
12/10/2008 2:55 AM - 3:00 AM	37.7	12/10/2008 2:55 AM - 3:00 AM	1.5	12/10/2008 3:00 AM - 3:05 AM	0.9	12/10/2008 3:00 AM - 3:05 AM	0.3
12/10/2008 2:50 AM - 2:55 AM	28.5	12/10/2008 2:50 AM - 2:55 AM	1.6	12/10/2008 2:55 AM - 3:00 AM	1.4	12/10/2008 2:55 AM - 3:00 AM	0
12/10/2008 2:45 AM - 2:50 AM	27.8	12/10/2008 2:45 AM - 2:50 AM	1.9	12/10/2008 2:50 AM - 2:55 AM	1.1	12/10/2008 2:50 AM - 2:55 AM	0.1
12/10/2008 2:40 AM - 2:45 AM	32.833	12/10/2008 2:40 AM - 2:45 AM	3.222	12/10/2008 2:45 AM - 2:50 AM	0.4	12/10/2008 2:45 AM - 2:50 AM	-0.5
				12/10/2008 2:40 AM - 2:45 AM	0.667	12/10/2008 2:40 AM - 2:45 AM	0.286
Average	28.078	Average	1.944	Average	0.497	Average	-0.063

Tabel pembacaan PRTG untuk *latency* pada pengujian 4.

LATENCY							
Mesh Client A (192.168.7.100)		Mesh Router B (192.168.7.20)		Mesh Router C (192.168.7.10)		Mesh Client E (192.168.7.50)	
Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)	Time	Ping Latency (ms)
12/11/2008 4:50 AM - 4:55 AM	39.2	12/11/2008 4:50 AM - 4:55 AM	3.3	12/11/2008 4:50 AM - 4:55 AM	-3.3	12/11/2008 4:50 AM - 4:55 AM	-16.1
12/11/2008 4:45 AM - 4:50 AM	41.4	12/11/2008 4:45 AM - 4:50 AM	3	12/11/2008 4:45 AM - 4:50 AM	0.2	12/11/2008 4:45 AM - 4:50 AM	-10.2
12/11/2008 4:40 AM - 4:45 AM	30.3	12/11/2008 4:40 AM - 4:45 AM	2.7	12/11/2008 4:40 AM - 4:45 AM	3.2	12/11/2008 4:40 AM - 4:45 AM	-10
12/11/2008 4:35 AM - 4:40 AM	23.2	12/11/2008 4:35 AM - 4:40 AM	1.6	12/11/2008 4:35 AM - 4:40 AM	0	12/11/2008 4:35 AM - 4:40 AM	-7.3
12/11/2008 4:30 AM - 4:35 AM	27.6	12/11/2008 4:30 AM - 4:35 AM	1.6	12/11/2008 4:30 AM - 4:35 AM	-2.6	12/11/2008 4:30 AM - 4:35 AM	-10
12/11/2008 4:25 AM - 4:30 AM	22.2	12/11/2008 4:25 AM - 4:30 AM	2.1	12/11/2008 4:25 AM - 4:30 AM	0.1	12/11/2008 4:25 AM - 4:30 AM	-14.8
12/11/2008 4:20 AM - 4:25 AM	20.5	12/11/2008 4:20 AM - 4:25 AM	1.8	12/11/2008 4:20 AM - 4:25 AM	0	12/11/2008 4:20 AM - 4:25 AM	-16.6
12/11/2008 4:15 AM - 4:20 AM	12.8	12/11/2008 4:15 AM - 4:20 AM	2.1	12/11/2008 4:15 AM - 4:20 AM	0	12/11/2008 4:15 AM - 4:20 AM	-6
12/11/2008 4:10 AM - 4:15 AM	27.9	12/11/2008 4:10 AM - 4:15 AM	1.9	12/11/2008 4:10 AM - 4:15 AM	0	12/11/2008 4:10 AM - 4:15 AM	-22.1
12/11/2008 4:05 AM - 4:10 AM	21	12/11/2008 4:05 AM - 4:10 AM	3.3	12/11/2008 4:05 AM - 4:10 AM	-2.3	12/11/2008 4:05 AM - 4:10 AM	-17
12/11/2008 4:00 AM - 4:05 AM	30	12/11/2008 4:00 AM - 4:05 AM	1.8	12/11/2008 4:00 AM - 4:05 AM	0.1	12/11/2008 4:00 AM - 4:05 AM	-18.8
12/11/2008 3:55 AM - 4:00 AM	24.25	12/11/2008 3:55 AM - 4:00 AM	4.444	12/11/2008 3:55 AM - 4:00 AM	0	12/11/2008 3:55 AM - 4:00 AM	-9.75
Average	26.696	Average	2.47	Average	-0.383	Average	-13.221

LAMPIRAN 9: Tabel Data Hasil Pengujian *End-To-End Throughput*.Tabel pembacaan jperf untuk *end-to-end throughput* pada sistem *multihop*.

bin/jperf.exe -c 192.168.7.100 -P 1 -i 1 -p 5001 -l 1m -f k -t 30						
Client connecting to 192.168.7.100, TCP port 5001						
TCP window size: 8.00 KByte (default)						
Interval (sec)	Throughput (Kbits/sec)					
	1 Hop		2 Hop		3 Hop	
	Non Aplikasi	Aplikasi	Non Aplikasi	Aplikasi	Non Aplikasi	Aplikasi
0.0 - 1.0	16777	33554	16777	8389	16777	8389
1.0 - 2.0	16777	16777	8389	8389	8389	8389
2.0 - 3.0	16777	16777	8389	0.00	8389	0.00
3.0 - 4.0	0.00	0.00	0.00	8389	8389	0.00
4.0 - 5.0	16777	16777	8389	0.00	0.00	0.00
5.0 - 6.0	16777	16777	8389	8389	0.00	8389
6.0 - 7.0	16777	16777	8389	8389	8389	8389
7.0 - 8.0	16777	0.00	8389	8389	8389	0.00
8.0 - 9.0	0.00	16777	0.00	0.00	0.00	0.00
9.0 - 10.0	16777	16777	8389	8389	8389	8389
10.0 - 11.0	16777	0.00	8389	0.00	8389	0.00
11.0 - 12.0	16777	16777	8389	8389	0.00	0.00
12.0 - 13.0	16777	16777	8389	8389	0.00	8389
13.0 - 14.0	0.00	16777	8389	8389	8389	8389
14.0 - 15.0	16777	0.00	8389	0.00	8389	0.00
15.0 - 16.0	16777	16777	0.00	8389	8389	8389
16.0 - 17.0	16777	16777	8389	0.00	0.00	0.00
17.0 - 18.0	16777	0.00	8389	8389	0.00	0.00
18.0 - 19.0	0.00	16777	8389	8389	8389	8389
19.0 - 20.0	16777	16777	8389	0.00	0.00	8389
20.0 - 21.0	16777	16777	0.00	8389	8389	0.00
21.0 - 22.0	16777	0.00	8389	8389	8389	0.00
22.0 - 23.0	16777	16777	8389	8389	0.00	0.00
23.0 - 24.0	0.00	16777	8389	0.00	8389	8389
24.0 - 25.0	16777	0.00	8389	8389	8389	0.00
25.0 - 26.0	16777	16777	8389	0.00	0.00	0.00
26.0 - 27.0	16777	16777	8389	8389	8389	8389
27.0 - 28.0	16777	16777	0.00	0.00	8389	0.00
28.0 - 29.0	0.00	0.00	8389	8389	8389	8389
29.0 - 30.0	16777	16777	8389	0.00	0.00	8389
Average	13421.6	12862.367	7270.433	5313.033	5592.633	3914.867

LAMPIRAN 10: Tabel Data Hasil Pengujian *End-To-End Jitter*.Tabel pembacaan jperf untuk *end-to-end jitter* pada sistem *multihop*.

bin/iperf.exe -s -u -P 0 -i 1 -p 5001 -w 1000k -f k						
Server listening on UDP port 5001						
Receiving 1470 byte datagrams						
UDP buffer size: 1000 Kbyte						
Interval (sec)	Jitter (ms)					
	1 Hop		2 Hop		3 Hop	
	Non Aplikasi	Aplikasi	Non Aplikasi	Aplikasi	Non Aplikasi	Aplikasi
0.0 - 1.0	0.279	0.539	1.878	7.229	1.889	8.253
1.0 - 2.0	0.631	0.513	0.05	8.223	0.05	9.467
2.0 - 3.0	0.14	0.606	0.059	6.213	2.999	7.848
3.0 - 4.0	0.586	0.011	1.627	6.84	0.024	7.977
4.0 - 5.0	0.073	0.016	3.408	5.337	1.091	8.297
5.0 - 6.0	0.1	1.905	1.607	7.136	2.832	8.555
6.0 - 7.0	0.061	0.743	1.64	7.25	2.586	11.175
7.0 - 8.0	0.1	0.02	1.576	4.374	3.397	9.124
8.0 - 9.0	0.07	0.01	1.566	8.078	2.549	7.945
9.0 - 10.0	1.677	0.068	1.582	8.705	1.226	9.877
10.0 - 11.0	0.14	0.000	1.483	5.837	2.513	10.236
11.0 - 12.0	0.073	0.023	2.761	7.392	3.516	6.152
12.0 - 13.0	0.061	0.000	1.511	6.398	1.154	11.042
13.0 - 14.0	0.071	1.981	1.18	3.022	4.435	6.45
14.0 - 15.0	0.631	0.091	1.296	4.723	7.537	6.573
15.0 - 16.0	0.586	0.435	2.609	5.783	4.669	4.943
16.0 - 17.0	0.1	1.878	0.022	7.082	1.426	10.005
17.0 - 18.0	0.1	0.17	2.078	5.582	4.415	6.183
18.0 - 19.0	1.677	1.098	0.502	6.87	1.361	5.339
19.0 - 20.0	0.013	0.005	1.248	4.623	7.405	6.391
20.0 - 21.0	0.071	0.000	1.597	5.751	0.086	8.757
21.0 - 22.0	0.061	0.005	2.474	7.802	0.873	4.873
22.0 - 23.0	0.073	0.001	1.41	5.339	0.332	7.802
23.0 - 24.0	0.14	0.218	1.208	7.719	3.182	5.978
24.0 - 25.0	0.279	0.055	0.238	5.434	0.835	3.916
25.0 - 26.0	0.631	0.844	1.825	4.199	1.252	9.789
26.0 - 27.0	0.061	0.016	0.818	5.885	1.708	6.642
27.0 - 28.0	0.1	1.905	0.498	7.045	1.494	6.857
28.0 - 29.0	0.14	0.74	0.395	3.916	1.188	7.867
29.0 - 30.0	0.586	0.02	0.187	1.743	2.403	8.352
Average	0.3	0.464	1.344	6.051	2.348	7.756