

# PERENCANAAN DAN IMPLEMENTASI WIRELESS MESH NODE PADA RASPBERRY PI

Mochammad Luthfi febriadi<sup>1</sup>, Adian Fatchur Rochim<sup>2</sup>, Eko Didik Widiyanto<sup>3</sup>  
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro,  
Jln. Prof. Sudharto, Tembalang, Semarang, Indonesia  
email : lutpii@yahoo.com

## ABSTRAK

Perkembangan teknologi jaringan nirkabel / wireless membuat banyak perusahaan atau organisasi yang menggunakan teknologi nirkabel, teknologi ini lebih efisien karena hanya menggunakan media gelombang radio. Melihat wilayah Indonesia memiliki kontur yang ekstrim dimana banyak bukit-bukit dan pegunungan maka diperlukan teknologi nirkabel yang handal dan efisien. Teknologi wireless mesh tepat untuk digunakan tetapi masih sedikit perangkat yang diimplementasikan sebagai wireless mesh node. Oleh karena itu penelitian tugas akhir ini akan merancang dan mengimplementasikan sebuah wireless mesh node pada perangkat komputer Raspberry Pi. Metodologi penelitian tugas akhir ini antara lain studi literatur dimana tahap penggalan konsep dan semua bahan yang terkait dengan wireless mesh dan Raspberry Pi. Perancangan dan implementasi yaitu melakukan perancangan wireless mesh node pada komputer Raspberry Pi yang menggunakan sistem operasi Debian ARM serta mengimplementasikan wireless mesh node dengan protokol routing OLSR. Terakhir adalah pengujian dan analisis terhadap wireless mesh node pada Raspberry Pi. Hasil penelitian tugas akhir yaitu perangkat wireless mesh node pada Raspberry Pi yang efisien dimana membutuhkan sumber daya komputer (memori dan CPU) yang sedikit dan sumber daya listrik yang kecil yaitu sebesar 5 volt serta memiliki Quality of Service (QoS) yang baik.

**Kata Kunci :** Debian ARM, OLSR, Raspberry Pi, Wireless Mesh

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Indonesia merupakan salah satu dari negara berkembang, dimana negara kita ini sedang berjuang untuk bangkit dari kondisi krisis ekonomi yang melanda sejak tahun 1998. Ekonomi suatu negara akan maju jika memiliki infrastruktur yang baik. Sebagai contoh, infrastruktur di Indonesia masih tertinggal dari negara lain seperti Singapura yang sudah sangat maju dan negara-negara tetangga lainnya. Salah satu infrastruktur yang penting yang menentukan majunya perekonomian suatu negara adalah infrastruktur teknologi informasi khususnya bidang telekomunikasi. Keberadaan infrastruktur telekomunikasi akan membantu meningkatkan laju perekonomian, dimana akses masyarakat terhadap informasi akan cepat dan terbuka sehingga sumber daya manusia negara tersebut akan meningkat.

Penyediaan sarana telekomunikasi di Indonesia sangatlah kompleks. Teknologi nirkabel merupakan teknologi yang tepat digunakan untuk penyediaan sarana telekomunikasi. Teknologi nirkabel tidak membutuhkan biaya yang banyak, karena tidak memerlukan media kabel dan hanya menggunakan gelombang radio sebagai media perantara. Dan juga cocok untuk wilayah yang memiliki medan berat sehingga sulit untuk dijangkau, serta keterbatasan sumber daya listrik (bagi daerah yang belum terjangkau PLN).

Solusi yang tepat dalam penerapan teknologi nirkabel pada wilayah yang telah diterangkan di atas adalah jaringan berbasis *wireless mesh network*. Komponen utama jaringan *wireless mesh* adalah suatu perangkat yang selain berfungsi sebagai sumber trafik juga dapat berperan sebagai router yang mampu merutekan trafik dari

1 : Mahasiswa Sistem Komputer UNDIP

2 : Dosen Sistem Komputer UNDIP

3 : Dosen Sistem Komputer UNDIP

sumber ke tujuan, perangkat tersebut disebut juga *wireless mesh node* (WMN). *Wireless mesh node* (WMN) yang digunakan pada penelitian tugas akhir ini adalah Raspberry Pi yaitu komputer mini yang memerlukan daya rendah yang dikembangkan oleh Raspberry Pi Foundation di Inggris.

### 1.2 Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah untuk mengimplementasikan jaringan *wireless mesh* dengan protokol *routing* OLSR pada komputer Raspberry Pi sebagai *wireless mesh node* sehingga akan dihasilkan sebuah node *wireless mesh* yang multifungsi dan efisien.

### 1.3 Batasan Masalah

Batasan masalah penelitian tugas akhir ini adalah :

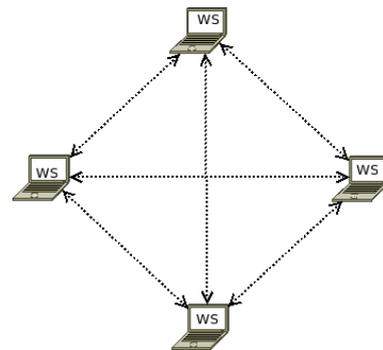
1. perancangan *wireless mesh node* menggunakan komputer Raspberry Pi.
2. Penggunaan sistem operasi Debian pada komputer Raspberry Pi.
3. Penggunaan protokol *routing* OLSR.

## 2. DASAR TEORI

### 2.1 Jaringan Nirkabel

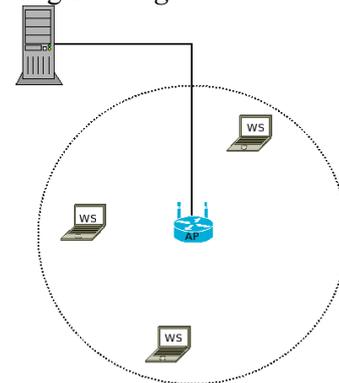
Jaringan nirkabel adalah suatu bentuk jaringan komputer dimana komunikasi yang terjadi antar perangkat komputer menggunakan frekuensi radio sebagai sarana transmisinya, memungkinkan *workstation* atau peralatan *mobile* untuk mengakses jaringan. Ada 2 jenis mode operasi *wireless LAN* yaitu :

1. *Ad-hoc Mode*, terdiri dari beberapa *wireless station* yang berkomunikasi secara langsung (*peer-to-peer*) tanpa menggunakan AP sebagai konfigurasi independen. Seperti pada gambar 1.



Gambar 1: Ad-hoc Mode

2. *Infrastructure Mode* seperti ditunjukkan pada gambar 2 konfigurasi *Infrastructure Mode* minimal terdiri dari sebuah *Access-Point* (AP) yang terhubung ke jaringan kabel atau internet. AP ini dikenal juga sebagai *managed network*.



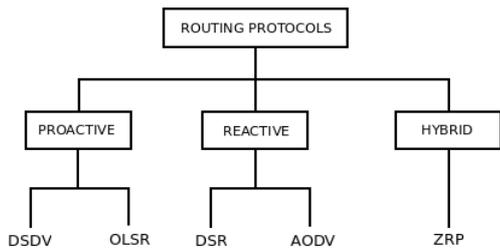
Gambar 2: Infrastruktur Mode

### 2.2 Konsep Dasar TCP/IP

Protokol-protokol yang digunakan dalam suatu sistem *wireless network* dapat dibagi menjadi beberapa bagian lapisan (*layer*) menurut pada standar model TCP/IP. Lapisan *Physical* yang utama adalah teknologi radio yang digunakan yang meliputi *data rate physical layer* dan kemampuan beroperasi saat ada *interferensi*.

Protokol pada *Link 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 control RTS/CTS (*Ready to Send/Clear to Send*) [3] [5].

Protokol yang dipakai dalam *network layer* pada suatu *wireless mesh network* banyak mengadaptasi *routing protocol* yang digunakan dalam *wireless adhoc network* yang seperti terlihat pada Gambar 3 dibagi menjadi tipe *proactive*, *reactive* dan *hybrid* [2].



Gambar 3: Klasifikasi *Ad-hoc Routing protocols*

Dua protokol utama *transport layer* adalah *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP).

Protokol 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 antar jaringan teknologi *wireless* [5].

### 2.3 Wireless Mesh Network

*Wireless mesh network* adalah jaringan komunikasi *wireless* yang terbentuk dari *node* radio dimana minimal terdapat dua atau lebih jalur komunikasi data pada setiap *node* [9]. Setiap *node* *Wireless mesh network* tidak hanya bertindak sebagai sebuah *host* tetapi juga dapat berfungsi sebagai sebuah *router* untuk meneruskan paket-paket informasi yang akan dikirim menuju *node* lain [4].

#### 2.3.1 Karakteristik *Wireless Mesh Network*

*Wireless mesh network* memiliki beberapa karakteristik umum yang sangat mempengaruhi kinerjanya.

1. *Multi-hop wireless network*
2. Kemampuan *self-forming*, *self-healing*, *self-organizing* serta

mendukung *ad-hoc networking*.

3. Tingkat mobilitas tergantung dari jenis *node*.
4. Dapat mengakses ke berbagai jenis teknologi jaringan lainnya.
5. Kebutuhan terhadap pemakaian daya tergantung dari jenis *node*.

#### 2.3.2 *Optimized Link State Routing* (OLSR)

OLSR (*Optimized Link State Protocol*) merupakan salah satu jenis dari *proactive routing protocol* yang biasa digunakan dalam jaringan *ad hoc*. Protokol ini melakukan pertukaran pesan secara periodik dalam rangka menjaga informasi topologi jaringan yang ada pada setiap *node* [7] [10]. Secara umum langkah-langka kerja dalam OLSR dapat diurutkan sebagai berikut :

1. *Link Sensing* (Pendeteksian hubungan).
2. *Neighbour detection* (pendeteksian *node* tetangga).
3. *MPR selection* (pemilihan MPR).
4. Pengiriman TC (*Topology Control*) Messages.
5. *Route calculation* (penghitungan jalur).

#### 2.3.3 Message OLSR

OLSR menggunakan beberapa jenis pesan yang berbeda untuk saling bertukar informasi antara lain *HELLO\_Message* untuk mendeteksi *node-node* tetangga dan *TC\_Message* untuk menginformasikan set/kumpulan jalur-jalur.

##### 2.3.3.1 HELLO Message

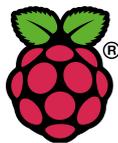
Setiap *node* pada OLSR harus mendeteksi *node-node* tetangga yang ada pada daerah jangkauannya. Untuk melakukan hal tersebut, setiap *node* akan mengirimkan paket *HELLO message* secara *broadcast* dalam periode waktu tertentu. Paket *HELLO* berisi informasi tentang *node-node* tetangga serta *link status*.

### 2.3.3.2 TC (Topology Control) Message

Setiap *node* yang telah terpilih sebagai MPR dalam jaringan mengirimkan TC Messages (pesan TC) untuk mendeklarasikan sebuah *set/kumpulan* jalur-jalur yang disebut *advertised link set* yang harus disertai minimal jalur ke seluruh *node* dari MPR *selector set*-nya atau *node-node* tetangga yang telah memilih *node* tersebut sebagai MPR-nya.

### 2.4 Raspberry Pi

Raspberry Pi adalah sebuah SBC (*Single Board Computer*) seukuran kartu kredit yang dikembangkan oleh Yayasan Raspberry Pi di Inggris (UK) dengan maksud untuk memicu pengajaran ilmu komputer dasar di sekolah-sekolah. Tetapi Raspberry Pi banyak digunakan untuk penelitian oleh para ilmuwan, hal ini karena Raspberry Pi sangat efisien, mudah digunakan, dan multifungsi.



Gambar 4: Logo Raspberry Pi

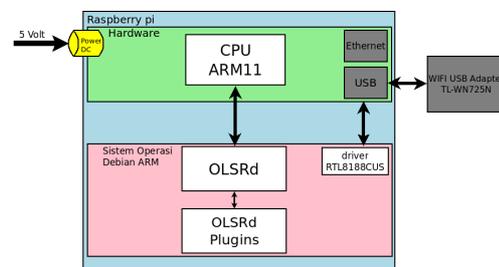
Tabel 1: Spesifikasi Raspberry Pi

Komponen	Keterangan
CPU	700 MHz ARM1176JZF-S core (ARM6 family)
Memori	512 MB/256 MB (shared with GPU)
USB ports	2 (via integrated USB hub)
Onboard storage	SD / MMC / SDIO card slot
Onboard network	10/100 Ethernet (RJ45)
Low-level peripherals	8 × GPIO, UART, I <sup>2</sup> C bus, SPI bus with two chip selects, +3.3 V, +5 V, ground[58][63]
Power source	5 volt via MicroUSB or GPIO header
Planned operating systems	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS, OpenWRT

## 3. PERANCANGAN DAN IMPLEMENTASI

### 3.1 Perancangan *Wireless Mesh Node*

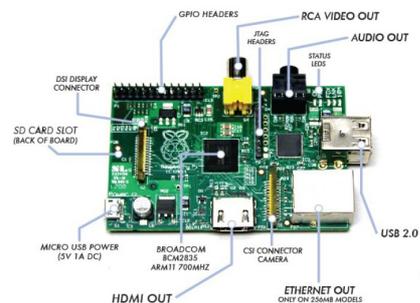
Perancangan sebuah *node wireless mesh*, langkah pertama adalah menentukan langkah-langkah pengerjaan *wireless mesh node* menggunakan Raspberry Pi. Untuk memudahkan dalam perancangan *wireless mesh node* pada Raspberry Pi maka dibuat diagram blok perancangan seperti pada gambar 5 yang berupa bagian-bagian dari *wireless mesh node* baik dari segi perangkat keras maupun perangkat lunak.



Gambar 5: Diagram blok perancangan *wireless mesh node* pada Raspberry Pi

#### 3.1.1 Spesifikasi *Hardware*

Perangkat keras yang digunakan sebagai *wireless mesh node* adalah komputer Raspberry Pi seperti pada gambar 6. Raspberry Pi ideal digunakan sebagai *node* karena hanya membutuhkan daya yang kecil (5 volt) dan memiliki ukuran yang kecil sehingga mudah untuk ditempatkan dan dipindahkan (*mobile*).



Gambar 6: Tampilan lengkap Raspberry Pi

### 3.1.1.1 Perancangan Media

#### Penyimpanan

Raspberry Pi menggunakan *secure digital (SD) memory card*, yaitu sistem penyimpanan *solid-state*. Dikarenakan media penyimpanan pada Raspberry Pi memegang penuh seluruh sistem operasi maka minimal besar SD *card* setidaknya 2 GB.

### 3.1.1.2 Perancangan Antarmuka

#### Jaringan

Antarmuka jaringan yg terdapat secara *default* pada Raspberry Pi adalah *port ethernet*, sedangkan untuk antarmuka *wireless* dapat ditambahkan menggunakan USB Wifi *adapter*. Dimana Raspberry Pi terdapat 2 antarmuka USB yang bisa digunakan. USB Wifi *adapter* yang digunakan adalah merk TP-Link seri TL-WN725N yang memiliki standar IEEE 802.11 b/g/n seperti pada gambar 7.



### 3.1.2 Spesifikasi Software

Perangkat lunak yang digunakan untuk *wireless mesh node* pada Raspberry Pi terdiri dari beberapa bagian, yaitu sistem operasi yang digunakan, aplikasi *routing* protokol yang digunakan serta *package-package* tambahan yang diperlukan untuk mendukung *routing* protokol tersebut. Untuk sistem operasi yang digunakan adalah *debian armel* dengan aplikasi *routing* protokol OLSRd.

#### 3.1.2.1 Sistem Operasi

Sistem operasi yang digunakan adalah *Debian ARM* karena Raspberry Pi menggunakan arsitektur *chip ARM11*. Varian *Debian ARM* terbaru adalah *debian-wheezy-armel*.

#### 3.1.2.2 Package OLSR

Protokol *routing* OLSR dapat dijalankan pada sistem operasi *debian* dengan menginstal aplikasi *olsrd (optimized link state routing daemon)*. Selain aplikasi utama *olsrd* dibutuhkan juga aplikasi tambahan untuk menunjang proses *routing* yaitu *olsrd-plugins*.

*Olsrd-plugins* memuat beberapa *plugin-plugin* yaitu *httpinfo, dyn\_gw, dot\_draw, secure, nameservice, bmf, quagga, txtinfo*. *Plugin* yang sering digunakan yaitu *txtinfo* dan *httpinfo*, *plugin* tersebut berfungsi untuk menampilkan sambungan *node*, *neighbors*, tabel topologi, dan tabel *routing* secara aktual. *Plugin txtinfo* menampilkan informasi dalam bentuk *text*, sedangkan *httpinfo* menampilkan informasi dalam bentuk laman *web*.

*Olsrd* dan *olsrd-plugins* sudah terpasang pada Raspberry Pi, untuk file konfigurasi *olsrd* terdapat pada */etc/olsrd.conf*. File tersebut digunakan untuk mengkonfigurasi semua elemen-elemen yang digunakan dalam *routing olsr*, seperti *interface*, variabel-variabel dan *plugin-plugin* yang digunakan dalam *routing olsr*.

### 3.2 Implementasi Wireless Mesh Node

Setelah perancangan sistem *wireless mesh node* menggunakan Raspberry Pi selesai langkah selanjutnya adalah implementasi hasil rancangan *wireless mesh node* pada Raspberry Pi. Adapun langkah-langkah dalam implementasi *wireless mesh node* pada Raspberry Pi adalah konfigurasi jaringan pada *wireless mesh node* dan Konfigurasi *olsrd*.

#### 3.2.1 Konfigurasi Jaringan

Beberapa hal penting yang perlu diperhatikan dalam mengkonfigurasi sebuah *node* dalam jaringan *wireless mesh* adalah konfigurasi *wireless* dari *node* tersebut, pengalamatan yang digunakan sesuai dengan ketentuan yang berlaku, dan *data packet forwarding* agar

*node-node* tersebut dapat meneruskan data yang berlangsung dalam trafik *network*.

Hal penting dalam *wireless mesh* adalah pastikan mode *wireless* yang digunakan adalah *ad-hoc*, karena dalam *wireless mesh* semua *node* terhubung satu dengan lainnya atau disebut *multipoint-to-multipoint*. Jadi antarmuka yang bisa digunakan hanya antarmuka *wireless* yang mendukung mode *ad-hoc*. Berikut tampilan konfigurasi *wireless*.

```

molutpi@molutpi:~$ iwconfig
lo                no wireless extensions.

wlan0             IEEE 802.11abgn  ESSID:"LUMESH"
                  Mode:Ad-Hoc   Frequency:2.412 GHz  Cell: 02:11:87:EA:C5:FF
                  Tx-Power=15 dBm
                  Retry  long limit:7   RTS thr:off   Fragment thr:off
                  Power Management:on

eth0              no wireless extensions.

molutpi@molutpi:~$

```

Agar setiap paket data yang dikirimkan melalui jaringan mesh dapat diteruskan melalui *interface-interface node* khususnya dengan jenis *interface* yang berbeda maka perlu dilakukan pengaturan *forwarding rules*.

### 3.2.2 Konfigurasi Olsrd

Konfigurasi OLSR dapat dilakukan dengan mengedit file *olsrd.conf* yang terdapat pada */etc/olsrd.conf*. Konfigurasi dalam *olsrd.conf* secara *default* sudah terdapat beberapa konfigurasi routing OLSR serta *plugin-plugin* yang akan digunakan.

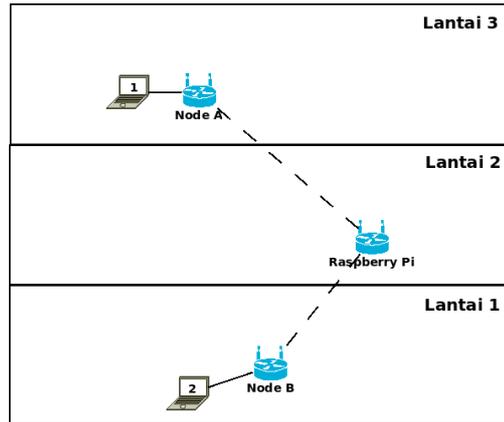
Terdapat dua *option* untuk Hna yaitu Hna4 dan Hna6. Hna6 digunakan jika jaringan yang menggunakan Ipv6 sedangkan Hna4 digunakan jika jaringan yang menggunakan Ipv4.

*Plugin* yang digunakan dalam penelitian tugas akhir ini hanya *plugin txtinfo*. Variabel *port* berisi nilai *port* yang digunakan untuk mengakses *plugin* tersebut nantinya menggunakan *telnet* ataupun *browser*. Variabel *accept* untuk menentukan *host* mana yang di ijinakan mengakses *plugin* tersebut.

## 4. PENGUJIAN DAN ANALISIS

### 4.1 Konfigurasi Pengujian

Pengujian ini menitik beratkan pada pengujian *wireless mesh node* Raspberry Pi, mulai dari pengujian penggunaan CPU dan memori, pengujian *throughput* dan pengujian *latency* dan *jitter*.



Gambar 9: Topologi pengujian *wireless mesh network*

Selain *node* Raspberry Pi dibutuhkan juga *node-node* lain untuk pengujian yang dijelaskan berikut ini.

- *Node A*  
*Devices* : TP-Link MR3420 v.1  
*OS* : OpenWRT  
*IP Adress* : 10.123.123.4/8
- *Node Raspberry Pi*  
*Devices* : Raspberry Pi Model B  
*OS* : Debian ARM  
*IP Adress* : 10.123.123.1/8
- *Node B*  
*Devices* : Thinkpad T410  
*OS* : Ubuntu 12.04 LTS  
*IP Adress* : 10.123.123.132/8

### 4.2 Pengujian *Wireless Mesh Node*

*Wireless mesh node* Raspberry Pi yang telah dibuat, maka langkah berikutnya adalah melakukan beberapa skenario pengujian untuk mengetahui secara menyeluruh dari *wireless mesh node* tersebut.

#### 4.2.1 Pengujian Penggunaan CPU Dan Memori

Pengujian ini dilakukan untuk mengamati performansi dari sistem berupa pemakaian memori dan CPU pada saat *wireless mesh node* pada Raspberry Pi berjalan dalam menangani dua buah *node* seperti pada topologi pengujian gambar 9.

#### 4.2.2 Pengujian Kinerja *Self-configure*

Skenario untuk pengujian kemampuan *self-configure* dari *wireless mesh network* digunakan untuk mendapatkan data waktu yang diperlukan suatu *node mesh* untuk melakukan *self-configure* pada dirinya sendiri dan bergabung dengan *wireless mesh network* yang sudah ada. Pengujian dilakukan berulang-ulang dengan merubah variabel besarnya interval dari HELLO *message* dan TC *message* dari paket OLSR yang dapat diubah pada file `etc/olsrd.conf`.

#### 4.2.3 Pengujian *Throughput*

Pengujian ini dilakukan untuk mengamati performansi dari *wireless mesh node* Raspberry Pi dalam *wireless mesh network*. Dalam skenario pengujian ini parameter yang akan diuji adalah *throughput*. *Throughput* adalah kecepatan rata-rata keberhasilan dalam satuan bit setiap detik pengiriman data melalui saluran komunikasi atau jaringan. Adapun topologi jaringan yang digunakan dalam skenario pengujian ini adalah seperti pada gambar 9.

#### 4.2.4 Pengujian *Latency* Dan *Packet Loss*

Pengujian ini dilakukan untuk mengamati performansi dari *wireless mesh node* Raspberry Pi dalam *wireless mesh network*. Dalam skenario pengujian ini parameter yang akan diuji adalah *latency* dan *packet loss*. *Latency* atau biasa disebut juga *delay* adalah jumlah waktu yang digunakan oleh sebuah paket ketika dikirim oleh sebuah *node* pengirim dan diterima di *node* tujuan. *Packet loss*

adalah banyaknya paket yang hilang selama proses transmisi oleh sebuah *node* pengirim dan diterima di *node* tujuan. Semakin kecil *latency* dan *packet loss* yang dialami, maka semakin baik *quantitative performance* dari *network* tersebut.

#### 4.3 Analisis *Wireless Mesh Node*

Setelah skenario pengujian yang dilakukan, kita dapat menganalisa data dari hasil pengujian tersebut. Analisa yang dilakukan adalah analisa penggunaan CPU dan memori, analisa pengujian hasil *throughput*, analisa pengujian *lency* dan *packet loss*, dan analisa kinerja *self-configure*.

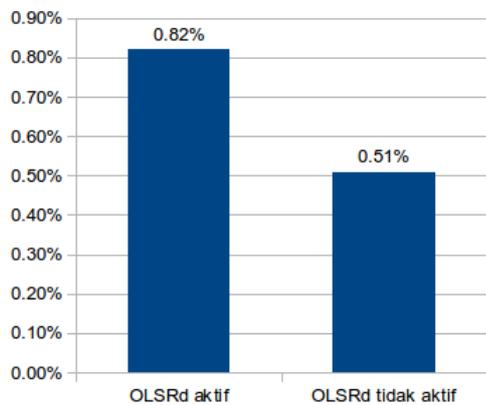
##### 4.3.1 Analisis Penggunaan CPU Dan Memori

Berdasarkan data pengujian penggunaan CPU dan memori, maka didapatkan grafik perbandingan penggunaan CPU dan memori saat aplikasi *routing* OLSR aktif dan saat aplikasi *routing* OLSR tidak aktif.

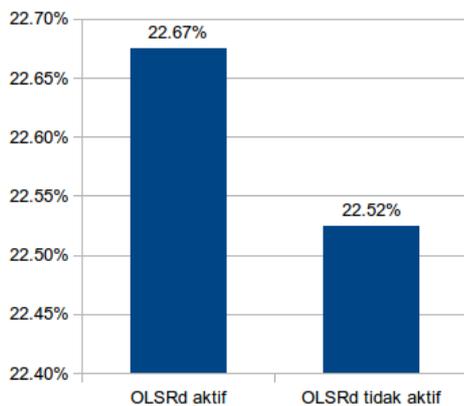
Pada gambar 10 ditampilkan grafik perbandingan penggunaan CPU pada saat aplikasi *routing* OLSR tidak aktif rata-rata sumber daya CPU yang digunakan sebesar 0.51% sedangkan pada saat *routing* OLSR aktif rata-rata sumber daya CPU yang digunakan sebesar 0.82%, perbandingan dua kondisi tersebut hanya selisih 0.31%, dari data pengujian tersebut dapat dianalisa bahwa pemakaian aplikasi *routing* OLSR tidak menggunakan banyak sumber daya CPU secara signifikan dalam menangani dua pengguna atau *node* yang terhubung dengan *wireless mesh node* pada Raspberry Pi.

Pada pengujian penggunaan memori yaitu perbandingan penggunaan memori pada saat aplikasi *routing* OLSR aktif dan saat aplikasi *routing* OLSR tidak aktif. Pada saat *routing* OLSR tidak aktif rata-rata memori yang digunakan sebesar 53528.4 KB sedangkan pada saat *routing* OLSR aktif rata-rata memori yang

digunakan sebesar 53886.4 KB. Menurut data tersebut disimpulkan bahwa aplikasi *routing* OLSR membutuhkan sumber daya memori rata-rata sebesar 358 KB dalam menangani dua pengguna atau *node* yang terhubung dengan *wireless mesh node* pada Raspberry Pi. Jika dalam grafik perbandingan penggunaan memori pada dua kondisi diatas akan dihasilkan grafik perbandingan dalam prosentase seperti pada gambar 11.



Gambar 10: Grafik perbandingan penggunaan CPU



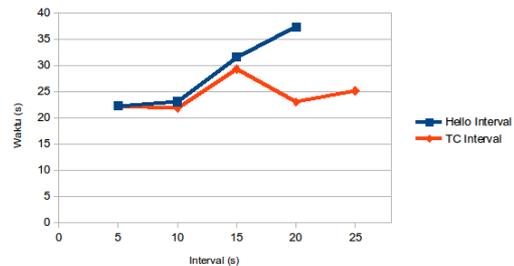
Gambar 11: Grafik perbandingan penggunaan memori

#### 4.3.2 Analisis Kinerja Self-configure

Berdasarkan hasil pengujian kemampuan *self-configure* pada *wireless mesh node* Raspberry Pi. Pengujian pertama adalah pengujian yang dilakukan dengan parameter *Hello Message*, dan pengujian kedua adalah pengujian yang

dilakukan dengan parameter *TC Message*.

Berdasarkan pada dua kali pengujian tersebut maka akan dihasilkan grafik perbandingan waktu yang diperlukan dalam kinerja *self-configure* oleh *wireless mesh node* Raspberry Pi seperti pada gambar 12.

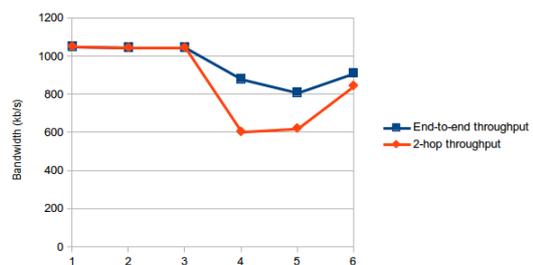


Gambar 12: Grafik perbandingan kinerja self-configure

Pada grafik perbandingan antara pengaruh parameter waktu interval Hello message dan TC message terhadap performansi self configure terlihat bahwa parameter waktu interval untuk Hello message lebih berpengaruh terhadap kinerja self-configure jaringan mesh dibandingkan dengan parameter interval TC message.

#### 4.3.3 Analisis Performansi Throughput

Pengujian *throughput* pada bagian sebelumnya maka akan didapatkan hasil pengukuran *throughput* pengujian *end-to-end throughput* antara *node A* dengan *node* Raspberry Pi. pengujian *throughput* dengan *node* Raspberry Pi berfungsi sebagai *node* perantara antara *node A* dengan *node B*, dalam hal ini dilakukan komunikasi jaringan dengan 2-hop. maka akan didapatkan grafik perbandingan seperti pada gambar 13.

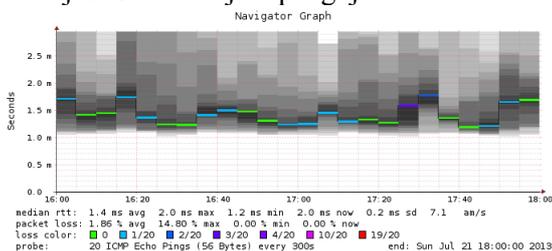


Gambar 13: Grafik pengujian throughput

Menurut grafik perbandingan dapat dianalisa bahwa nilai throughput jaringan wireless mesh menggunakan *node* Raspberry Pi sebagai *end-to-end devices* memiliki nilai rata-rata *throughput* yang lebih besar dibandingkan dengan nilai rata-rata *node* Raspberry Pi sebagai *node* perantara pada *throughput* 2-hop. Jadi nilai *throughput* pada jaringan *wireless mesh* dipengaruhi oleh jumlah *hop* dan besarnya data yang dikirim.

#### 4.3.4 Analisis Pengujian Latency Dan Packet Loss

Berdasarkan pengujian *letency* dan *packet loss* yang menggunakan aplikasi *SmokePing*, maka akan dihasilkan grafik seperti pada gambar 14. Pada gambar tersebut terdapat data untuk besarnya nilai tengah rata-rata rtt atau *round-trip time* dan *packet loss* yang terjadi selama 2 jam pengujian.



Gambar 14: Grafik pengujian *latency* dan *packet loss*

Gambar grafik diatas maka didapat nilai latency yang bervariasi selama 2 jam pengujian, kemudian akan dihasilkan nilai tengah rata-rata pengujian latency yaitu sebesar 1,4 ms. Dari data tersebut maka akan didapatkan rata-rata besar *packet loss* yang terjadi selama 2 jam pengujian adalah sebesar 1,86%.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Raspberry Pi tidak menyediakan fitur POE, maka dibutuhkan sebuah konverter tambahan agar POE dapat berjalan di Raspberry Pi.

2. Penggunaan Raspberry Pi sebagai *wireless mesh node* pada *routing* OLSR menggunakan sumber daya memori yang sedikit yaitu sebesar kurang lebih 358 KB dan sumber daya CPU sekitar 0.31%.
3. Nilai interval *Hello message* lebih berpengaruh terhadap performansi *self configure* pada jaringan *wireless mesh* daripada nilai interval *TC message*. Semakin besar nilai *Hello message* maka kinerja *self-configure* akan semakin lama.
4. Posisi setiap *node* dan mobilitas-nya antar *node* bertetangga mempengaruhi *QoS* layanan komunikasi data yang berjalan.
5. Nilai *throughput* jaringan *wireless mesh* menggunakan *node* Raspberry Pi sebagai *end-to-end devices* memiliki nilai rata-rata *throughput* yang lebih besar yaitu 985,83 Kbps dibandingkan dengan nilai rata-rata *node* Raspberry Pi sebagai *node* perantara pada *throughput* 2-hop 870,5 Kbps.
6. Nilai *throughput* pada jaringan *wireless mesh* dipengaruhi oleh jumlah *hop* dan besarnya data yang dikirim.
7. Nilai *latency* dari *node* Raspberry Pi terbilang cukup yaitu sebesar 1,4 ms dan terdapat *packet loss* sebesar 1,86%.

### 5.2 Saran

saran yang dapat diberikan penulis untuk pengembangan lebih lanjut dari penelitian ini adalah sebagai berikut:

1. Penerapan *routing* protokol *wireless mesh* selain OLSR pada Raspberry Pi dan pengujian dengan menggunakan lebih banyak *node* sehingga kinerja OLSR lebih terlihat.

2. Membangun node Wireless mesh yang multifungsi pada Raspberry Pi dengan memaksimalkan fitur-fitur dari Raspberry Pi.

#### DAFTAR PUSTAKA

- [1] Andrew S. Tanenbaum, "Computer Network" (New Jersey : Prentice Hall, 2003)
- [2] B.H Walke, S. Mangold, L. Berlemann, "IEEE 802 Wireless Systems : Protocols, Multi-Hop Mesh/Relaying, Performance and Spectrum Coexistence" (Chicester : John Wiley & Sons Ltd, 2006)
- [3] Gilbert Held, "Wireless Mesh Networks" (Boca Raton : Aurbach Publications, 2005)
- [4] Gunadi Dwi Hantoro (2005) "Mengenal Wireless Mesh". 2008 dari Telkom RDC Media. <http://www.telkomrdc-media.com/index.php?ch=8&lang=&s=26fdbeeba7e3ec6e6297247bc5fffe3a&n=321&page=5>. Diakses 5 Juni 2013
- [5] Ian F. Akyldiz, Xudong Wang, Weilin Wang, "Wireless mesh networks: a survey." Diakses 21 Juni 2013. [http://users.ece.gatech.edu/~wxudong/Xudong\\_Wang\\_WirelessMesh\\_COMNET\\_0305.pdf](http://users.ece.gatech.edu/~wxudong/Xudong_Wang_WirelessMesh_COMNET_0305.pdf)
- [6] T. Clausen, P. Jaquet, "Optimized Link State Protocol (OLSR)", RFC 3626, Oktober 2003
- [7] Thomas B. Krag, "Introduction to Wireless Mesh" Presentasi, <http://www.wireless.dk>. diakses 23 Juni 2013
- [8] Yan Zhang, Jijun Luo, Honglin Hu. "Wireless Mesh Networking Architecturs Protocols and Standards"(USA: Auerbach Publication Taylor & Francis Group., 2007), hal 15

- [9] --, Wireless Mesh Network [http://en.wikipedia.org/wiki/Wirelessmesh\\_network](http://en.wikipedia.org/wiki/Wirelessmesh_network). Diakses 2 Mei 2013

- [10] --, OLSR, [http://en.wikipedia.org/wiki/Optimized\\_Link\\_State\\_Routing\\_Protocol](http://en.wikipedia.org/wiki/Optimized_Link_State_Routing_Protocol). Diakses 23 Juni 2013

#### BIODATA PENULIS



Mochammad Luthfi Febriadi, lahir di kota Pontianak pada 21 Februari 1990, menempuh pendidikan di TK Tunas Mulia Semarang, kemudian melanjutkan di SD Kemala Bhayangkari 04 Akpol Semarang, SMP Negeri 5 Semarang, SMA Kesatrian 1 Semarang dan sekarang sedang melanjutkan pendidikan di Program Studi Sistem Komputer Fakultas teknik Universitas Diponegoro Semarang.