



# ANALISIS PERFORMANSI *WIRELESS MESH NETWORK* DENGAN MENGGUNAKAN PROTOKOL *HYBRID WIRELESS MESH PROTOCOL (HWMP)* PADA JARINGAN *MOBILE AD HOC NETWORK (MANET)*

Muh. Darman Saleh<sup>\*1</sup>, Muh. Yamin<sup>2</sup>, LM. Fid Aksara<sup>3</sup>

<sup>\*1,2,3</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari  
e-mail : <sup>\*1</sup>[darman.saleh10@gmail.com](mailto:darman.saleh10@gmail.com), <sup>2</sup>[muh\\_yamin@uho.ac.id](mailto:muh_yamin@uho.ac.id), <sup>3</sup>[fid.aksara@uho.ac.id](mailto:fid.aksara@uho.ac.id),

## Abstrak

Semakin berkembangnya teknologi nirkabel dalam komunikasi antar pengguna maka dibutuhkan pula jaringan yang berkualitas tinggi. Penelitian ini bertujuan untuk merancang sebuah jaringan dan sebagai parameter uji pada jaringan *Wireless Mesh Network (WMN)* dengan menggunakan protokol *routing Hybrid Wireless Mesh Protocol (HWMP)*.

Jaringan ini dihubungkan dengan menggunakan beberapa *access point* yang telah dimodifikasi dengan menggunakan *firmware OpenWRT*. Parameter uji yang akan dilakukan yaitu dengan melakukan pengujian terhadap kinerja dari protokol *routing HWMP* dalam *self-configure* dan *self-healing* serta beberapa data penunjang dari *Throughput* dan *Jitter*.

Data yang dihasilkan dengan menggunakan beberapa *access point* dan dua (2) pengguna menghasilkan data dimana pada *self-configure* dapat dilakukan dengan sangat cepat, beda halnya dengan *self-healing* yang membutuhkan waktu lebih lama, pengaruh letak *access point* yang kurang memadai menimbulkan kinerja jaringan yang belum maksimal, dengan adanya *routing HWMP* mempengaruhi *Throughput* dan *Jitter* yang dihasilkan user apabila melakukan pertukaran informasi dalam kondisi selalu berpindah dari tempat satu ke tempat yang lainnya. Hasil rata – rata dari *self-configure* ialah 100 bytes/64 ms, *self-healing* ialah 100 bytes/187 ms. *Bandwidth User 1* dan *2* ialah 5,930 Kbyte/s dan 5,210,170 Kbyte/s. Nilai *Jitter* pada *User 1* dan *2* ialah 1,554 ms dan 1,699 ms. *Troughput* pada *User 1* dan *2* ialah 0,283 Mbit/s dan 4 Mbit/s.

**Kata Kunci**—WMN, HWMP, *Firmware openWRT*, *Self-Configure*, *Self-Healing*

## Abstract

*The growing development of wireless technology in communication between users requires high-quality networks. This study aims to design a network and as a test parameter on the Wireless Mesh Network (WMN) using the routing protocol Hybrid Wireless Mesh Protocol (HWMP).*

*This network is connected by using several access points that have been modified using OpenWRT software. The test parameters to be carried out are by testing the performance of the HWMP routing protocol in self-configure and self-healing as well as some supporting data from Throughput and Jitter.*

*The data generated by using several access points and two (2) users produce data whereon self-configure can be done very quickly, unlike the case with self-healing which requires more time, the effect of inadequate access point location gives rise to network performance, not yet maximal, the existence of routing HWMP affects Throughput and Jitter generated by the user if the exchange of*



information in the condition always changes from one place to another. The average result of self-configuration is 100 bytes / 64ms, self-healing is 100 bytes / 187ms. Users of bandwidth 1 and 2 are 5,930 Kbyte / s and 5,210,170 Kbyte / s. Jitter values for users 1 and 2 are 1.554 ms and 1,699 ms. Throughput on users 1 and 2 is 0.283 Mbit / s and 4 Mbit/s.

**Keywords**—WMN, HWMP, Firmware openWRT, Self-Configure, Self-Healing

## 1. PENDAHULUAN

**W**ireless Mesh Network (WMN) merupakan jaringan *wireless* yang terbentuk dari *node* radio dimana minimal terdapat dua atau lebih jalur komunikasi data pada setiap *node*. WMN merupakan cara paling efektif untuk mengoptimalkan jaringan *nirkabel* ketika *node nirkabel* dalam posisi *mobile* dan *terkoneksi* dalam jaringan sekaligus memonitor semua *node* lain dalam jaringan. Jaringan *mobile peer-to-peer* ini merupakan persyaratan konektivitas dari sebuah sistem *mesh network* dan kombinasi dari WMN sehingga mampu menyediakan jaringan tanpa kabel yang fleksibel dan handal. *Hybrid Wireless Mesh Protocol* (HWMP) merupakan *default protocol routing* dari IEEE 802.11s WLAN *mesh networking*. Sebagai salah satu jenis *protocol routing* yang *hybrid*. HWMP mendukung dua *model* dalam pencarian rutenya yaitu *on-demand mode* dan *proactive tree building mode*. Kombinasi reaktif dan proaktif dalam HWMP yang optimal dan efisien memungkinkan pemilihan path pada berbagai jaringan *mesh* dengan atau tanpa *infrastructure*. *Routing* di HWMP menggunakan mekanisme nomor urut untuk menjaga konektivitas bebas *loop* setiap saat [1].

*Mobile Ad Hoc Network* (MANET) merupakan teknologi jaringan *Ad Hoc* yang memiliki kemampuan untuk beradaptasi dengan kondisi *mobile* pada tiap *node*-nya. Berdasarkan dokumen *Request for Comments* menjelaskan bahwa terdapat beberapa karakteristik dari MANET. Dijelaskan bahwa MANET terdiri dari *mobile platform* (seperti *router* dan perangkat *wireless*) dalam hal ini disebut dengan “*node*” yang bebas berpindah-pindah ke mana saja. *Node* tersebut bisa saja berada di pesawat, kapal, mobil dan dimana saja [2].

Penelitian ini bertujuan untuk menerapkan dan mengevaluasi protokol *Dynamic Mobile Ad Hoc Network* (DYMO) pada sistem *monitoring* lahan pertanian

berbasis jaringan *sensor nirkabel*. Rancangan jaringan terdiri dari *node sensor* yang ditempatkan secara statik di lahan pertanian dan *node router* serta *gateway* sebagai *node* bergerak yang dibawa oleh petani/mobil pertanian. Hasil evaluasi menunjukkan bahwa protokol DYMO dapat digunakan pada jaringan *sensor nirkabel* untuk sistem *monitoring* lahan pertanian. Evaluasi rancangan dan skenario tersebut pada lingkungan simulasi OPNET memperlihatkan kinerja jaringan menggunakan protokol DYMO dapat diterima [3].

WMN adalah salah satu cara untuk memperluas jaringan komputer tanpa harus terbatas dengan jarak yang jauh seperti jaringan *komputerkabel*, *self configure* dan *self healing* merupakan karakteristik utama dari WMN. Perangkat yang mendukung WMN adalah WRT54GL dengan menggunakan *freifunk firmware* untuk mendukung *optimized link state protocol* yang merupakan *routing protocol* yang sangat cocok diterapkan pada WMN untuk menjamin kualitas komunikasi data yang baik. WMN memberikan kemudahan dan kepatan untuk instalasi jaringan komputer yang memiliki area jaringan luas [4].

Transmisi video melalui jaringan *nirkabel* semakin penting dan dibutuhkan. Umumnya, pembahasan transmisi *video* melalui teknologi jaringan *nirkabel* masih secara umum, sehingga memberikan tantangan dalam penggunaan *Throughput*, *end to end Delay*, *Packet Loss* dan PSNR. *Combined Scalable Video Coding* (CSCV) merupakan skema baru perkembangan dari H.264/MPEG-4. Pada penelitian ini CSCV digunakan untuk pengiriman video dengan *standard IEEE 802.11* pada lingkungan WMN dan diaplikasikan pada *protocol AODV*, *DSDV* dan *DSR*. Untuk mengetahui kinerja protokol *routing* mana yang lebih baik dalam transmisi video CSVC digunakan metrik pengujian *end to end Delay*, *Throughput*, *Packet Loss* dan PSNR. Untuk analisis dan pengukuran data digunakan *simulator NS-2*, dan hasil

percobaan simulasi yang dilakukan menunjukkan AODV memiliki nilai *end to end Delay* yang rendah yaitu 0,57 detik, DSR memiliki nilai yang baik di dalam *Throughput* yaitu 31 kbps dan PSNR yaitu 22,7 dB, sedangkan DSDV memiliki *Packet Loss* yang rendah yaitu 56,4 [5].

Pada penelitian yang dilakukan oleh [6], analisis peformansi kerja WMN dengan menggunakan *protocol* HWMP dan tidak meneliti tentang performansi kerja pada jalur *routing* HWMP, dari penelitian penelitian yang sudah dijabarkan bahwa WMN banyak di gunakan sebagai sensor dan juga dalam penelitian lain ada masih banyak menggunakan simulasi dalam pengujiannya. Dalam penelitian ini akan membahas tentang bagai mana jalaurnya akses yang dilalui tiap *node*. Dengan dilakukan penelitian tentang performansi kerja pada jalur HWMP dapat membantu dalam penelitian selanjutnya

Selanjutnya, penelitian yang dilakukan [7] menyatakan bahwa *wireless mesh* tepat untuk digunakan tetapi masih sedikit perangkat yang diimplementasikan sebagai *wireless mesh node*. Metodologi penelitian ini antara lain studi literatur dimana tahap penggalian 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 ini 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.

## 2. METODE PENELITIAN

Metode yang dilakukan dalam proses pengumpulan data dilakukan dengan beberapa tahap yaitu :

### 2.1 Kajian Pustaka

Metode ini digunakan untuk mencari literatur atau sumber pustaka yang berkaitan

dengan pengimplementasian WMN pada jaringan MANET. Sumber literatur berupa buku, *paper*, jurnal, karya ilmiah, dan beberapa situs penunjang [8].

### 2.2 Metode Perancangan Sistem

Perancangan sistem arsitektur jaringan WMN terbagi menjadi 4 yaitu perancangan topologi WMN, Pengimplementasian WMN serta analisis jaringan MANET, Simulasi perancangan topologi WMN berisikan gambaran topologi WMN yang akan di implementasikan untuk melakukan simulasi agar dapat dijadikan bahan pertimbangan di kemudian hari apabila akan diterapkan pada topologi jaringan yang ada di Jurusan Teknik Informatika, pengimplementasian topologi WMN berisikan pengimplementasian jaringan WMN pada Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo, analisis WMN berisikan hasil analisis jaringan WMN yang telah diimplementasikan beserta apa saja *output* yang akan didapatkan dari hasil perubahan arsitektur jaringan tersebut [9].

### 2.3 Analisis

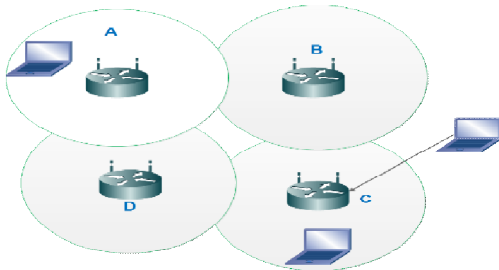
Tahap analisis tersiri atas :

- a. Analisis terhadap pengguna jaringan yang selalu bergerak dari titik satu ke titik yang lain ini menunjukkan *user* akan terputus dengan jaringan yang digunakan dan dapat mengganggu dalam melakukan transfer data.
- b. Dalam segi *hardware access point* yang digunakan memiliki jangkauan yang sangat dekat dan banyaknya penghalang berupa tembok dan gedung-gedung tinggi akan mempengaruhi kualitas jaringan yang dihasilkan, sedangkan dalam segi *software*, kemampuan *framework* dalam *access point default* atau bawaan memiliki kemampuan yang sangat terbatas, terkecuali *access point* yang digunakan merupakan merek mahal maka akan memadai pula dalam melakukan kinerjanya. Untuk itu digunakanlah *framework OpenWRT* agar dapat memaksimalkan kinerja dari *access point* yang digunakan.
- c. Dalam jumlah data pengguna jaringan yang banyak *access point* tidak dapat menampung semua pengguna dalam satu waktu

- d. Konfigurasi jaringan yang diatur memiliki maksimal pengguna hanya sampai 100 pengguna dalam waktu yang bersamaan, ini dapat mempengaruhi kecepatan dan kinerja dari *access point* akan kewalahan
- e. Peletakan *access point* yang diletakan pada ruangan terbuka dapat membahayakan, mulai dapat dicuri atau di rusak oleh pengguna yang tidak bertanggung jawab, dengan adanya penelitian ini *access point* dapat diletakan di mana saja terutama pada ruangan yang kerja atau semacamnya [10].

#### 2.4 Desain

Dari data-data yang sudah didapatkan sebelumnya, pada tahap ini akan dibuatkan gambar desain topologi jaringan yang akan dibangun. Sebelum melakukan pengimplementasian, pertama-tama dilakukan perancangan jaringan WMN, pada perancangan ini digunakan Edraw MAX untuk mempermudah dalam proses perancangan topologi dan dapat dijadikan sebagai patokan dalam pemasangan topologi WMN pada Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo nantinya. Gambar 1 menunjukkan desain dan topologi jaringan yang akan digunakan [3].



Gambar 1 Topologi WMN

#### 2.5 Implementasi

Setelah proses *design* dan *simulation prototyping* selesai, dihasilkan topologi WMN yang bisa digunakan dan perancangan itu dapat diimplementasikan langsung pada Jurusan Teknik Informatika Fakultas Teknik Universitas Halu Oleo. Hasil yang diharapkan dalam pengimplementasian jaringan WMN ini adalah :

1. Dapat dihasilkan topologi jaringan WMN yang dapat dijadikan panduan ketika diterapkan pada Jurusan Teknik

Informatika Fakultas Teknik Universitas Halu Oleo.

2. Dengan menganalisis hubungan jumlah dan kecepatan *node station* pada *protocol routing* HWMP, kita dapat mengetahui kecepatan akses suatu *node* pada topologi WMN yang digunakan.
3. Dengan WMN pengelolaan instalasi jaringan lebih mudah, karena setiap *access point* yang terhubung, tidak lagi menggunakan kabel sebagai *backbone*, sehingga dapat mencakup area-area yang tidak memungkinkan untuk melakukan instalasi jaringan menggunakan kabel [6].

#### 2.6 Monitoring

Pada tahap ini dilakukan *monitoring* terhadap kualitas dari perangkat yang digunakan dan bagai mana kinerja dari alat yang digunakan terhadap proses pengiriman data dan kenyamanan pengguna, yaitu :

- a. Melihat kinerja dari *access point* dari segi *self-configure* dan *self-healing*
- b. Melihat kinerja *access point* terhadap pengguna yang selalu berpindah pindah dari titik satu ketitik yang lainnya
- c. Melakukan pengujian terhadap pengiriman data yang yang dilakukan secara terus menerus dengan posisi *user* yang selalu berpindah-pindah [2].

#### 2.7 Manajemen

Pada tahap ini dilakukan pengaturan *access point* dibuat dengan menggunakan *OpenWRT* agar dapat menghasilkan konfigurasi yang jaringan yang diinginkan dengan memanfaatkan protokol HWMP agar manajemen jaringan yang dihasilkan lebih stabil dan dapat teratur [4].

### 3. HASIL DAN PEMBAHASAN

Berdasarkan rancang bangun topologi, dibuatlah sebuah gambaran titik atau lokasi penempatan *access point* yang dapat dipakai untuk pegujian WMN dengan menggunakan protokol *routing* HWMP. Seperti ditunjukkan pada Gambar 2.

#### 3.1 Pengujian *Self-Configure* pada Topologi WMN

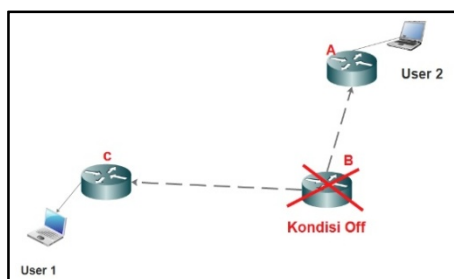
Pengujian ini bertujuan untuk melihat waktu yang dibutuhkan WMN dapat

melakukan *self-healing*, pengujian dilakukan dengan cara sebagai berikut :



Gambar 2 Topologi Jaringan WMN

1. Tiap *user* pada masing-masing *access point* harus melakukan *ping* secara terus menerus dimana *User 1* di *access point C* melakukan *ping* pada *User 2* di *access point A* begitu sebaliknya
2. Kondisi yang terjadi pada pengujian ini di mana *access point A* dan *C* saling berjauhan, maka hasil *ping* yang dilakukan sebelumnya akan menghasilkan *request time out*.
3. Selanjutnya *access point B* dinyalakan dengan bertujuan agar *access point A* dan *C* bisa saling berhubungan
4. Dengan dinyalakannya *access point B* maka *access point A* dan *C* bisa saling terhubung dengan menggunakan *system multihop*
5. Kemampuan *access point B* untuk mengkonfigurasi inilah yang dikatakan sebagai *self-configure*
6. Data yang dicatat adalah waktu yang dibutuhkan *router B* untuk melakukan *self-configure* dengan mencatat waktu yang diperlukan agar *User 1* dapat berhubungan dengan *User 2* atau mencatat waktu mulainya respon *reply* pada *User 1* semenjak dinyalakannya *router B* seperti yang terlihat pada Gambar 3.



Gambar 3 Kondisi Pengujian *Self-Configure*

Hasil dari proses pengujian *self-configure* dapat dilihat pada Tabel 1.

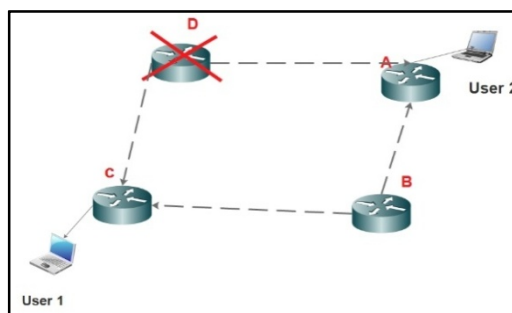
Tabel 1 Tabel Hasil *Ping/Self-Configure*

Percobaan	Banyak Data(bytes)	Waktu/(ms)
1	100	67
2	100	56
3	100	114
4	100	23
5	100	60

Pengujian ini bertujuan untuk mengetahui seberapa cepat sebuah *access point* melakukan *self-configure* pada dirinya sendiri untuk bergabung dengan *access point* WMN yang sudah ada, dari Tabel 1 hasil pengujian dengan menggunakan *ping* rata – rata waktu yang diperlukan untuk sebuah *access point* WMN melakukan *self-configure* ialah 100bytes/64 ms. Pengujian ini dilakukan secara terus menerus sebanyak lima kali dikarenakan *access point* yang digunakan yaitu *access point portable*.

### 3.2 Pengujian *Self-Healing* pada Topologi WMN

Pengujian kemampuan *self-healing* digunakan untuk mendapatkan data waktu yang diperlukan jaringan untuk mencari jalur baru atau apa bila salah satu dari *access point* rusak atau terputus, seperti yang terlihat pada Gambar 4.



Gambar 4 Kondisi Pengujian *Self-Healing*

Adapun langkah – langkah yang dilakukan dalam pengujian ini adalah sebagai berikut :

1. Pengujian dilakukan dengan kondisi *User 1* pada *access point C* berusaha terus menerus terhubung dengan *User 2* pada *Access Point A* dengan melalui *access point D*.
2. Kemudian *access point D* dimatikan, dalam arti kondisi rusak, dengan demikian jalur pada *Access Point D* tidak dapat dilalui maka *ping* akan *request time out*.

3. Protokol HWMP dengan otomatis akan mencari jalur baru melalui *router C* yang masih aktif, kemampuan inilah yang disebut dengan *self-healing*.
4. Data yang dicatat adalah waktu yang dibutuhkan *user 1* pada *access point C* untuk dapat berhubungan kembali dengan *user 2* pada *router A* melalui jalur yang baru yaitu dengan munculnya kembali pesan *reply* pada perintah *ping* di *User 1*.

Adapun hasil dari pengujian *self-healing* ditunjukkan pada Tabel 2.

Tabel 2 Tabel Hasil Ping/Self-Healing

Percobaan	Banyak Data(bytes)	Waktu/(ms)
1	100	134
2	100	431
3	100	141
4	100	114
5	100	112

Pengujian ini bertujuan untuk melihat kecepatan *access point* WMN bergabung dengan jaringan WMN terdekat apa bila dari empat *access point* salah satunya mengalami kerusakan atau mati maka *access point* tersebut akan berusaha mencari jaringan WMN yang masi aktif dan berada paling dekat dengannya serperti yang telah dijelaskan sebelumnya *access point* yang digunakan adalah *access point portable*, dari hasil pengujian *self-healing* pada Tabel 2 menghasilkan rata – rata *access point* dapat melakukan *self-healing* ialah 100 bytes/187 ms.

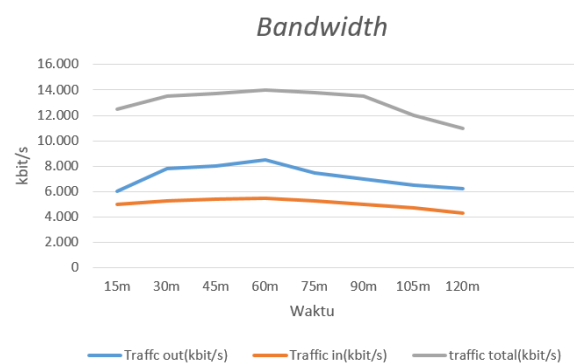
### 3.3 Pengujian Bandwith

*Bandwidth* adalah cakupan luas atau lebar frekuensi yang digunakan oleh sinyal pada media transmisi. Dengan kata lain, *bandwidth* adalah perbedaan pada komponen sinyal frekuensi yang tinggi dan yang rendah.

#### a. Pengujian Bandwith pada User 1

Pada pengujian ini dilakukan untuk melihat kemampuan jaringan WMN dalam menjalankan berbagai aplikasi serta dimana keadaan interferensi rendah. Dapat dilihat pada grafik 15 menit pertama *bandwidth* yang masuk sebanyak 12,665 Kbyte/s selanjutnya 30 menit dengan aplikasi *chat* telah dijalankan dan hasilnya menunjukkan grafik langsung naik ke puncak data total *bandwidth* yang diterima

sampai 3,062,914 Kbyte/s selanjutnya masuk pada 45 menit mencapai 3,011,113 Kbyte/s setelah 60 menit berikutnya dilakukan pengiriman *file* atau data yang ukurannya  $\pm 10\text{Gb}$  dan hasil yang didapatkan ialah 5,417,101 Kbyte/s semakin berjalannya waktu semakin besar jumlah data yang diterima oleh *User 1*, masuk 75 menit data yang masuk mencapai 476,839 Kbyte/s, 90 menit berikutnya aplikasi audio *converence* dijalankan dan hasil yang didapatkan sebesar 37,878 Kbyte/s, masuk 105 menit 37,887 Kbyte/s, dan sampai masuk pada 120 menit data yang masuk mencapai 5,930 Kbyte/s, seperti yang ditunjukkan pada Gambar 5.

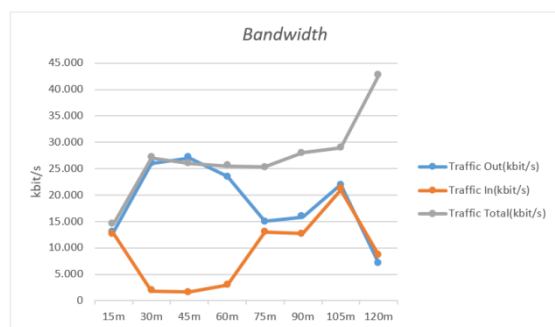


Gambar 5 Grafik Bandwith User 1

#### b. Pengujian Bandwith pada User 2

Dari hasil pengujian ini menunjukan bahwa besaran data yang masuk dari 15 menit pertama masih menunjukan data pada grafik tidak terlalu tinggi yaitu 188,965 Kbyte/s, masuk diwaktu 30 menit data yang masuk sampai 199,449 KByte/s, pada pengujian awal masuk menggunakan aplikasi *chat* sederhana selanjutnya dilakukan pengiriman *file* dengan banyaknya data sebesar  $\pm 10\text{Gb}$  dan hasil yang didapatkan pada 45menit berikutnya total data yang masuk sebesar 180,265 Kbyte/s masuk 75 menit berikutnya data yang dihasilkan sebesar 25,322 Kbyte/s 90 menit berikutnya data yang masuk sebesar 201,810 Kbyte/s, selanjutnya masuk pada 105 menit tahap pengujian dengan audio *converence* jadi data yang didapatkan sebesar 3,069,377 Kbyte/s masuk di 120 menit berikutnya data yang dihasilkan sebesar 5,210,170 Kbyte/s, jadi dari hasil *monitoring* yang dilakukan dapat diambil kesimpulan bahwa semakin banyaknya aplikasi yang berjalan pada *user* dua yang berada pada *access point A* maka *bandwidth* yang dihasilkan akan semakin besar pula ini

dikarenakan jaringan WMN yang dibangun tidak memiliki hambata atau dengan kata lain berada pada interferensi rendah seperti yang terlihat pada Gambar 6.



Gambar 6 Grafik *Bandwith User 2*

### 3.4 Pengujian *Jitter*

Pada pengujian *Jitter* dilakukan dengan dua kondisi di masing – masing *user* yaitu kondisi ketika mengunakan aplikasi berupa *chat*, pengiriman *file*, serta audio *converence* yang berikutnya dilakukan dengan tidak mengunakan aplikasi sama sekali, aplikasi yang digunakan dalam *monitoring Jitter* ialah Iperf.

#### a. Pengujian *Jitter* pada *User 1*

Pengujian pertama dilakukan dengan tidak adanya aktifitas apapun baik dari segi aplikasi *chat*, pengiriman *file* serta audio *converence* yang dijalankan pada pengujian ini, data yang dihasilkan dalam pengamatan selama satu jam tanpa henti, aplikasi *monitoring* yang digunakan yaitu aplikasi Iperf dengan cara *User 1* yang berada pada *access point C* melakukan *ping Jitter* pada *User 2* yang berada pada *access point A* dengan demikian data yang dihasilkan terlihat pada Tabel 3.

Tabel 3 Data Hasil *Monitoring Jitter* pada *User 1* Tanpa Aplikasi

Interval (s)	<i>Jitter</i> (ms)
300.00-301.00 sec	1.926 ms
600.00-601.00 sec	1.892 ms
900.00-901.00 sec	1.948 ms
1200.00-1201.00 sec	1.239 ms
1500.00-1501.00 sec	3.743 ms
1800.00-1801.00 sec	1.934 ms
2100.00-2101.00 sec	1.110 ms
2400.00-2401.00 sec	2.522 ms
2700.00-2701.00 sec	2.274 ms
3000.00-3001.00 sec	14.258 ms

*Jitter* yang dihasilkan menunjukkan tidak jauh berbeda tiap 5 menit *monitoring* yang dilakukan, total keseluruhan dari hasil *monitoring* selama satu jam tanpa henti *Jitter* yang dihasilkan sebanyak 1,554 ms dengan total datagrams 0,0035%. Selanjutnya pengujian *Jitter* akan dilakukan dengan kondisi dimana terdapat aktifitas berupa aplikasi *Chat*, transfer *file* dan audio *converence*, prosedur pengujian dilakukan secara bertahap dimana 5 menit pertama terdapat aktifitas *chat* yang dilakukan secara terus menerus sampai 30 menit, pengujian akan dilakukan dengan melakukan pengiriman *file* dengan ukuran *file*  $\pm 10$  Gb pada tahap ini pengujian dilakukan sampai masuk pada 45 menit lalu pengujian masuk ke tahap terakhir dengan adanya aktifitas audio *converence* pengujian ini berlangsung sampai 60 menit kemudian. Data yang didapatkan dapat dilihat pada Tabel 4.

Tabel 4 Data Hasil *Monitoring Jitter* pada *User 1* dengan Aplikasi

Interval (s)	<i>Jitter</i> (ms)
300.00-301.00 sec	1.826 ms
600.00-601.00 sec	17.018 ms
900.00-901.00 sec	13.105 ms
1200.00-1201.00 sec	11.201 ms
1500.00-1501.00 sec	14.380 ms
1800.00-1801.00 sec	9.095 ms
2100.00-2101.00 sec	14.529 ms
2400.00-2401.00 sec	14.008 ms
2700.00-2701.00 sec	14.729 ms
3000.00-3001.00 sec	16.393 ms
3300.00-3301.00 sec	9.109 ms
3599.00-3600.00 sec	25.463 ms

#### b. Pengujian *Jitter* pada *User 2*

Prosedur dalam penelitian ini sama halnya yg telah dilakukan pada *User 1* yang berada pada *access point C* yang melakukan *ping* kepada *User 2* yang berada di *access point A* namun sebaliknya untuk pengujian ini *User 2* yang berada pada *access point A* melakukan *ping Jitter* pada *User 1* yang berada pada *access point C*, pengujian ini dilakukan dengan aplikasi *monitoring* Iperf, dalam pengujian ini terdapat dua kondisi dimana kondisi pertama tidak ada aktifitas aplikasi yang akan dijalankan baik berupa *chat*, transfer *file* dan audio *converence* dan kondisi kedua dimana semua aplikasi akan dijalankan secara berkala, 5 menit pertama aplikasi *chat* akan dijalankan sampai masuk pada 30 menit

berikutnya, transfer *file* akan dilakukan selama 15 menit berikutnya dan terakhir aplikasi audio *converence* akan dilakukan sampai satu jam kemudian. Seperti terlihat pada Tabel 5.

Tabel 5 Data Hasil *Monitoring Jitter* pada *User 2* Tanpa Aplikasi

Interval (s)	<i>Jitter</i> (ms)
300.00-301.00 sec	0.823 ms
600.00-601.00 sec	3.069 ms
900.00-901.00 sec	1.490 ms
1200.00-1201.00 sec	1.350 ms
1500.00-1501.00 sec	1.003 ms
1800.00-1801.00 sec	2.107 ms
2100.00-2101.00 sec	2.287 ms
2400.00-2401.00 sec	1.895 ms
2700.00-2701.00 sec	3.046 ms
3000.00-3001.00 sec	2.617 ms
3300.00-3301.00 sec	1.903 ms
3599.00-3600.00 sec	1.699 ms

Dari hasil data pada Tabel 5 menunjukkan kecepatan *Jitter* tiap lima menit nya, dan total *Jitter* yang diperoleh dari hasil *monitoring* selama satu jam tanpa henti adalah 1,699 ms dengan datagrams yang dihasilkan mencapai 0,3%. Selanjutnya pengujian akan dilakukan dengan 3 kondisi dimana kondisi pertama aplikasi *chat* akan dijalankan mulai dari 5 menit pertama sampai masuk pada 30 menit kemudian akan dilakukan transfer *file* dengan ukuran *file*  $\pm 10$  Gb pada pengujian ini akan berlangsung sampai masuk menit ke 45, setelah masuk pada menit ke 45 masuk pada kondisi aplikasi audio *conversnce* dijalankan selama 60 menit berikutnya. Data yang dapat diperoleh dari pengujian ini dapat dilihat pada Tabel 6.

Tabel 6 Data Hasil *Monitoring Jitter* pada *User 2* dengan Aplikasi

Interval (s)	<i>Jitter</i> (ms)
300.00-301.00 sec	0.972 ms
600.00-601.00 sec	3.292 ms
900.00-901.00 sec	4.986 ms
1200.00-1201.00 sec	8.508 ms
1500.00-1501.00 sec	6.097 ms
1800.00-1801.00 sec	8.389 ms
2100.00-2101.00 sec	8.025 ms
2400.00-2401.00 sec	3.661 ms
2700.00-2701.00 sec	3.835 ms
3000.00-3001.00 sec	5.910 ms
3300.00-3301.00 sec	4.613 ms
3599.00-3600.00 sec	4.172 ms

### 3.5 Pengujian *Throughput* pada *User 1*

Proses penghitungan *Throughput* dilakukan setiap 5 menit waktu *monitoring* berlangsung tanpa henti yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{0,027344 \text{ KByte}}{0,09375 \frac{\text{KByte}}{\text{s}}} * 8\text{bit} \\ &= 0,00234 \text{ Mbit/s} \end{aligned}$$

Pengujian *Throughput* dengan kondisi adanya aktifitas aplikasi aplikasi ialah ini :

- a. Lima menit–lima belas menit penggunaan aplikasi *chat* yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{359 \text{ KByte}}{1,22625 \text{ KByte/s}} \\ &= 29,2763 \text{ Mbit/s} \end{aligned}$$

- b. Dua puluh menit–empat puluh lima menit dengan kondisi *file* transfer yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{205,513 \text{ KByte}}{0,70125 \text{ KByte/s}} \\ &= 2,3445 \text{ Mbit/s} \end{aligned}$$

- c. Lima puluh lima menit–enam puluh menit dengan kondisi audio *converence* yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{317,203 \text{ KByte}}{1,0825 \text{ KByte/s}} * 8\text{bit} \\ &= 2,3443 \text{ Mbit/s} \end{aligned}$$

### 3.6 Pengujian *Throughput* pada *User 2*

Pengujian *Throughput* pada *User 2* tanpa adanya aktifitas aplikasi yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{5,17 \text{ KByte}}{0,08875 \text{ KByte/s}} * 8\text{bit} \\ &= 0,467 \text{ Mbit/s} \end{aligned}$$

Pengujian selanjutnya dengan kondisi adanya aktifitas aplikasi *chat*, *file* transfer dan audio *coverence* ialah sebagai berikut :

- a. Lima menit–lima belas menit penggunaan aplikasi *chat* yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{103,592 \text{ KByte}}{1,82238 \text{ KByte/s}} * 8\text{bit} \\ &= 56,8446 \text{ Mbit/s} \end{aligned}$$

- b. Dua puluh menit–empat puluh lima menit dengan kondisi *file* transfer yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{100,299 \text{ KByte}}{1,712 \text{ KByte/s}} * 8\text{bit} \\ &= 0,4687 \text{ Mbit/s} \end{aligned}$$

- c. Lima puluh lima menit–enam puluh menit dengan kondisi audio *converence* yaitu :

$$\begin{aligned} \text{Throughput} &= \frac{100,299 \text{ KByte}}{1,712 \text{ KByte/s}} * 8\text{bit} \\ &= 0,4662 \text{ Mbit/s} \end{aligned}$$



#### 4. KESIMPULAN

Dari hasil pengujian yang telah dilakukan dapat ditarik kesimpulan bahwa jaringan WMN sangat efektif digunakan pada area yang sangat luas dan jaringan ini pun bisa memulihkan dirinya sendiri dengan dengan sangat cepat, namun dalam hal pengiriman data sangat berpengaruh pada besar kecilnya data yang dikirim dengan waktu yang dibutuhkan untuk sampai ketujuan.

#### 5. SARAN

Adapun saran dalam penelitian ini yaitu diharapkan kedepannya menggunakan yang memiliki spesifikasi yang sama agar mendapatkan data yang lebih akurat. Diharapkan bisa dilakukan penelitian lebih lanjut mengenai WMN dengan menggunakan pengujian yang lebih kompleks

#### DAFTAR PUSTAKA

- [1] I. Chakeres and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing draft-ietf-manet-dymo-17," 2009. [Online]. Available: <http://tools.ietf.org/pdf/draft-ietf-manet-dymo-17.pdf>. [Accessed: 12-Sep-2018].
- [2] A. Halid, L. O. J. Takumansang, and M. Fajar, "Dynamic MANET on Demand Protocol pada Jaringan Sensor Nirkabel untuk Sistem Monitoring Lahan Pertanian," *Ilk. J. Ilm.*, Vol. 9, No. 2, pp. 211- 217, 2018.
- [3] M. Fajar, J. Litan, A. Munir, and A. Halid, "Energy Efficiency Using Data Filtering Approach on Agricultural Wireless Sensor Network," *Int. J. Comput. Eng. Inf. Technol.*, Vol. 9, No. 9, pp. 192–197, 2017.
- [4] M. Alilou and M. Dehghan, "Upgrading Performance of DSR Routing Protocol in Mobile Ad-Hoc Networks," Vol. 1, No. 5, pp. 38–40, 2005.
- [5] D. Amaxilatis and I. Chatzigiannakis, "Component Based Clustering in Wireless Sensor Networks," *arXiv*, Vol. 5, No. 12, pp. 1847–1854, 2011.
- [6] B. Mardani, "Analisis Unjuk Kerja Wireless Mesh Network dengan Routing Protocol OLSR," Universitas Indonesia, 2008.
- [7] A. Rahman and Z. Zukarnain, "Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad-Hoc Networks," *Eur. J. Sci. ...*, Vol. 31, No. 4, pp. 566–576, 2009.
- [8] F. M. Manullang, N. D. Wahyu Cahyani, and V. Suryani, "Analisis Performansi Routing Hybrid Wireless Mesh Protocol (HWMP) pada Wireless Mesh Network (WMN) Berdasarkan Standar IEEE 802.11S," *JSM (Jurnal SIFO Mikroskil)*, Vol. 13, No. 1, pp. 43–52, 2012.
- [9] S. Misra, S. C. Misra, and I. Woungang, Eds., *Guide to Wireless Mesh Networks*. London: Springer, 2009.
- [10] R. F. Sari, A. Syarif, and B. Budiardjo, "Analisis Kinerja Protokol Routing Ad Hoc on-Demand Distance Vector (AODV) pada Jaringan Ad Hoc Hybrid : Perbandingan Hasil Simulasi dengan NS-2 dan Implementasi," *Makara J. Technol.*, Vol. 12, No. 1, pp. 7–18, 2008.

