

**ANALISIS KINERJA VOIP (VOICE OVER INTERNET PROTOCOL)
PADA WIRELESS MESH NETWORK**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Galuh Yudha Mahardika

NIM: 115060807113017

UNIVERSITAS BRAWIJAYA



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

ANALISIS KINERJA VOIP (VOICE OVER INTERNET PROTOCOL) PADA WIRELESS
MESH NETWORK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar
Sarjana Komputer

Disusun Oleh :
Galuh Yudha Mahardika
NIM: 115060807113017

Skripsi ini telah diuji dan dinyatakan lulus pada
4 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Adhitya Bhawiyuga, S.Kom, M.S
NIK. 20140589072011001

Kasyful Amron, S.T, M.Sc
NIP. 19750803 200312 1 003

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, ST., MT., Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

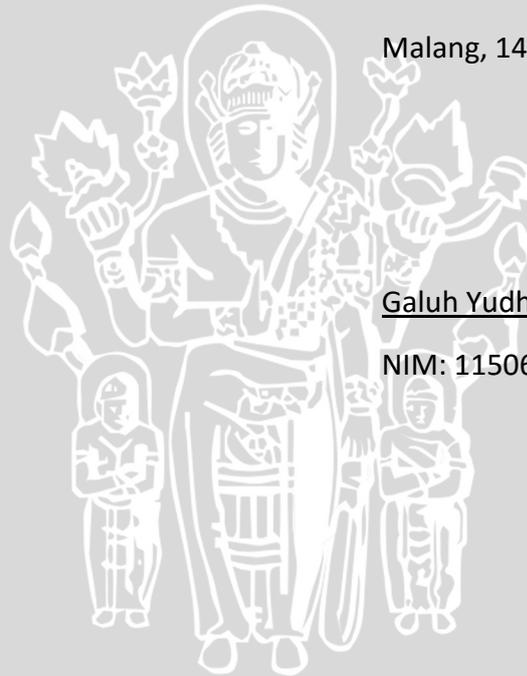
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 14 Juli 2016

Galuh Yudha Mahardika

NIM: 115060807113017



KATA PENGANTAR

Dengan menyebut nama Allah Yang Maha Pengasih dan Penyayang. Segala puji bagi Allah Subhanahu wa ta'ala karena hanya atas limpahan nikmat dan rahmat karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Analisis Kinerja VoIP (*Voice Over Internet Protocol*) Pada *Wireless Mesh Network*".

Penyusunan skripsi tidak lepas dari bantuan semua pihak yang terus memberikan bimbingan, kritik, saran, dukungan, motivasi maupun doa dari orang tua. Oleh karena itu, ucapan terima kasih penulis sampaikan kepada:

1. Ibunda Sutjiati selaku orang tua penulis dan seluruh keluarga yang telah memberikan doa dan dukungan penuh dalam penyusunan skripsi ini.
2. Bapak Adhitya Bhawiyuga, S.Kom, M.S. selaku pembimbing 1 dan Bapak Kasyful Amron, S.T, M.Sc selaku pembimbing 2, atas segala bimbingan dan waktu yang telah diluangkan serta kritik dan saran yang diberikan kepada penulis.
3. Seluruh bapak dan ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya atas segala bimbingan dan ilmu yang telah diajarkan kepada penulis.
4. Teman-teman penulis khususnya Fredy Nendra Pranata S.Kom, Muh. Ari Nurpermadi, Perdana Frischo Arganata, Bima Surya Nugraha, yang memberikan dukungan, semangat dan motivasi.
5. Teman-teman grup bimbingan dan grup riset WMN, Mikko, Arius, Redhi, Bayu dan Fika atas masukan, motivasi, diskusi, bantuan, pertemanan dan kelucuan yang membuat pengerjaan skripsi ini lebih berwarna.
6. Seluruh mahasiswa Program Studi Teknik Informatika angkatan 2011.
7. Semua pihak yang tidak dapat disebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah Subhanahu wa ta'ala. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna serta banyak kekurangan disebabkan oleh keterbatasan kemampuan dan ilmu yang dimiliki penulis. Semoga laporan skripsi ini bermanfaat bagi pembaca dan pengembang selanjutnya.

Malang, 14 Juli 2016

Penulis

galuh.y.mahardika@gmail.com

ABSTRAK

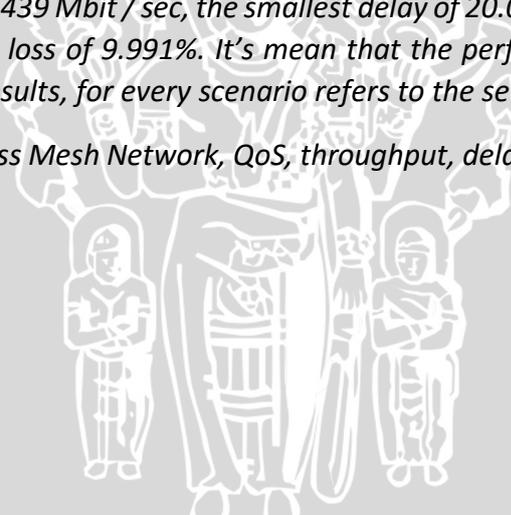
Voice over Internet Protocol (juga disebut VoIP, IP Telephony, Internet telephony atau Digital Phone) adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. Dengan adanya teknologi VoIP, biaya untuk melakukan komunikasi antara satu pengguna ke pengguna lainnya menjadi lebih efisien. Hal ini disebabkan karena VoIP tidak bergantung pada jarak dan bersifat global. Namun timbul permasalahan jika VoIP dijalankan di Wireless Mesh Network. WMN memiliki sifat self-configure dan self healing yang memungkinkan membangun konfigurasi diri sendiri dan membenahi diri sendiri serta tetap berfungsi jika satu atau lebih node mengalami kerusakan. Kedua sifat tersebut menyebabkan gangguan proses komunikasi dari VoIP. Maka dari itu, perlu dilakukan analisis VoIP pada WMN untuk menguji kinerja dan mengetahui kualitas VoIP berdasarkan parameter Quality of Service (QoS). Pengujian kinerja menggunakan QoS dilakukan dalam beberapa skenario pengujian yaitu, pengujian komunikasi VoIP antar client, pengujian komunikasi dengan 1 node disconnected, pengujian komunikasi dengan 2 node disconnected secara bergantian dan pengujian komunikasi dengan client bergerak (mobile). Nilai dari setiap parameter QoS terhadap VoIP pada setiap codec berupa throughput, delay, jitter dan packet loss kemudian dibandingkan untuk dijadikan tolok ukur baik tidaknya komunikasi pada teknologi WMN. Berdasarkan hasil pengujian yang telah dilakukan diperoleh nilai throughput tertinggi sebesar 0.1439 Mbit/sec, delay terkecil sebesar 20.0049 ms, jitter terendah sebesar 3.178 dan packet loss sebesar 9.991%. Hal tersebut menunjukkan kinerja dari VoIP yang dijalankan pada WMN mendapatkan hasil baik, untuk setiap skenario yang mengacu pada standart yang telah ditetapkan.

Kata Kunci: VoIP, Wireless Mesh Network, QoS, throughput, delay, jitter, paket loss.

ABSTRACT

Voice over Internet Protocol (called VoIP, IP Telephony, Internet telephony or Digital Phone) is a technology voice conversation that long range via the Internet. The VoIP technology, make the cost communication between the user to another more efficient. This is because VoIP doesn't depend with the distance and global. But there are still any problems, if VoIP running with Wireless Mesh Network. Characteristics of WMN were self-configure and self-healing allowed to build configuration yourself, fix themselves and continue to work if one or more nodes damaged. There were caused disruption of VoIP communication. So, there was to analyze required VoIP on WLAN to performance test and know the VoIP quality based on the parameters Quality of Service (QoS). The QoS performance testing conducted in several test scenario. There were VoIP communication test between client, one node disconnected, second node alternately disconnected and the client moved (mobile). The value of parameter QoS to VoIP on each codec such as throughput, delay, jitter and packet loss then compared, whether or not the communication on WMN technology. Based on test, that have been the highest throughput value of 0.1439 Mbit / sec, the smallest delay of 20.0049 ms, the lowest jitter 3.178 and packet loss of 9.991%. It's mean that the performance of VoIP is run on WMN is good results, for every scenario refers to the set standard.

keywords: VoIP, Wireless Mesh Network, QoS, throughput, delay, jitter, paket loss.



DAFTAR ISI

| | |
|--|-----|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| ABSTRAK..... | v |
| <i>ABSTRACT</i> | vi |
| DAFTAR ISI..... | vii |
| DAFTAR TABEL..... | x |
| DAFTAR GAMBAR..... | xi |
| DAFTAR KODE..... | 4 |
| DAFTAR LAMPIRAN | 5 |
| BAB 1 PENDAHULUAN..... | 6 |
| 1.1 Latar Belakang | 6 |
| 1.2 Rumusan Masalah | 7 |
| 1.3 Batasan Masalah | 7 |
| 1.4 Tujuan | 8 |
| 1.5 Manfaat | 8 |
| 1.6 Sistematika Penulisan..... | 8 |
| BAB 2 LANDASAN KEPUSTAKAAN | 10 |
| 2.1 <i>Wireless Mesh Network</i> | 10 |
| 2.1.1 Arsitektur Wireless Mesh Network..... | 11 |
| 2.1.1.1 Infrastruktur Wireless Mesh Network | 12 |
| 2.1.1.2 Client Wireless Mesh Network..... | 12 |
| 2.1.1.3 Hybrid Wireless Mesh Network | 13 |
| 2.1.2 <i>Babel Routing</i> | 13 |
| 2.2 VoIP..... | 14 |
| 2.2.1 Sistem Kerja VoIP | 14 |
| 2.2.1.1 <i>Session Initiation Protocol (SIP)</i> | 15 |
| 2.2.2 Codec | 16 |

| | |
|--|-----------|
| 2.2.3 Kualitas layanan VoIP (QOS) | 17 |
| 2.2.3.1 Throughput..... | 17 |
| 2.2.3.2 Delay..... | 17 |
| 2.2.3.3 Jitter..... | 18 |
| 2.2.3.4 Paket Loss..... | 19 |
| 2.3 Perangkat Pendukung | 19 |
| 2.3.1 Asterisk | 20 |
| 2.3.2 Raspberry Pi | 20 |
| BAB 3 METODOLOGI | 22 |
| 3.1 Studi Literatur..... | 22 |
| 3.2 Lingkungan Penelitian | 23 |
| 3.2.1 Topologi Jaringan | 24 |
| 3.2.1.1 Skenario Pengujian 1..... | 25 |
| 3.2.1.2 Skenario Pengujian 2..... | 26 |
| 3.2.1.3 Skenario Pengujian 3..... | 27 |
| 3.2.1.4 Skenario Pengujian 4..... | 28 |
| 3.2.2 Implementasi | 29 |
| 3.2.2.1 Konfigurasi Wireless Mesh Network Interfaces | 29 |
| 3.2.2.2 Implementasi Asterisk 13..... | 30 |
| 3.2.2.3 Implementasi Protokol..... | 31 |
| 3.2.2.4 Implementasi Softphone..... | 32 |
| 3.3 Pengambilan Data | 33 |
| 3.4 Hasil dan Pembahasan | 34 |
| 3.5 Kesimpulan dan Saran | 35 |
| BAB 4 HASIL DAN PEMBAHASAN | 36 |
| 4.1 Hasil Pengujian VoIP Skenario 1 | 36 |
| 4.1.1 Hasil Pengujian Koneksi <i>Client – Server</i> | 36 |
| 4.1.2 Hasil Pengujian Koneksi <i>Client-Client</i> | 37 |
| 4.1.3 Hasil Pengujian Panggilan VoIP..... | 37 |
| 4.1.4 Pengambilan Data Kualitas VoIP..... | 38 |
| 4.1.4.1 Pengambilan Data Kualitas VoIP dengan <i>Codec G.711</i> | 38 |

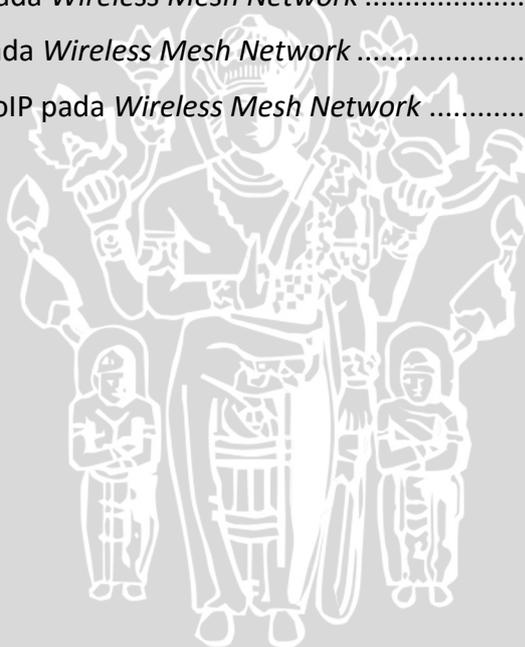


| | |
|--|----|
| 4.1.4.2 Pengambilan Data Kualitas VoIP dengan <i>Codec</i> G.722..... | 40 |
| 4.1.4.3 Pengambilan Data Kualitas VoIP dengan <i>Codec</i> G.726-32..... | 42 |
| 4.2 Hasil Pengujian VoIP Skenario 2 | 44 |
| 4.3 Hasil Pengujian VoIP Skenario 3 | 45 |
| 4.4 Hasil Pengujian VoIP Skenario 4 | 47 |
| 4.5 Pembahasan | 48 |
| 4.5.1 Throughput | 48 |
| 4.5.2 <i>Delay</i> | 49 |
| 4.5.3 <i>Jitter</i> | 51 |
| 4.5.4 <i>Paket Loss</i> | 53 |
| BAB 5 PENUTUP | 56 |
| 5.1 Kesimpulan | 56 |
| 5.2 Saran | 56 |
| DAFTAR PUSTAKA..... | 57 |
| LAMPIRAN | 59 |



DAFTAR TABEL

| | |
|--|----|
| Tabel 2.1 Macam – Macam CODEC..... | 17 |
| Tabel 2.2 Kategori Delay | 18 |
| Tabel 2.3 Kategori Jitter | 19 |
| Tabel 2.4 Kategori <i>Paket loss</i> | 19 |
| Tabel 3.1 Daftar ruang pemasangan Rpi..... | 23 |
| Tabel 3.2 Daftar Alamat IP Wlan0..... | 30 |
| Tabel 4.1 Hasil Pengujian Skenario 4 Codec G.711 | 47 |
| Tabel 4.2 Hasil Pengujian Skenario 4 Codec G.726-32 | 47 |
| Tabel 4.3 <i>Throughput VoIP</i> pada <i>Wireless Mesh Network</i> | 48 |
| Tabel 4.4 <i>Delay VoIP</i> pada <i>Wireless Mesh Network</i> | 50 |
| Tabel 4.5 <i>Jitter VoIP</i> pada <i>Wireless Mesh Network</i> | 51 |
| Tabel 4.6 <i>Paket Loss VoIP</i> pada <i>Wireless Mesh Network</i> | 53 |



DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 <i>Arsitecture Wireless Mesh Network</i> | 11 |
| Gambar 2.2 <i>Infrastruktur Wireless Mesh Networks</i> | 12 |
| Gambar 2.3 <i>Client Wireless Mesh Network</i> | 13 |
| Gambar 2.4 <i>Hybrid Wireless Mesh Network</i> | 13 |
| Gambar 2.5 <i>VoIP secara umum</i> | 15 |
| Gambar 2.6 <i>Arsitektur SIP</i> | 15 |
| Gambar 2.8 <i>Raspberry Pi model B+</i> | 20 |
| Gambar 3.1 <i>Diagram Alir Penelitian</i> | 22 |
| Gambar 3.2 <i>Denah Lokasi Penelitian di Gedung C FILKOM UB</i> | 23 |
| Gambar 3.3 <i>Topologi Wireless Mesh Netwok di Gedung C</i> | 24 |
| Gambar 3.4 <i>Pengujian Komunikasi VoIP</i> | 25 |
| Gambar 3.5 <i>Pengujian Komunikasi Saat 1 Node Disconected</i> | 26 |
| Gambar 3.6 <i>Pengujian dengan Node Rpi 4 Disconected</i> | 27 |
| Gambar 3.7 <i>Pengujian dengan Node Rpi 3 Disconected</i> | 27 |
| Gambar 3.8 <i>Pengujian dengan Client Bergerak (Mobile)</i> | 28 |
| Gambar 3.9 <i>Konfigurasi Host dan Hostname</i> | 29 |
| Gambar 3.10 <i>Hasil Instalasi Asterisk Server</i> | 30 |
| Gambar 3.11 <i>Console CLI Asterisk Server</i> | 31 |
| Gambar 3.12 <i>Mengaktifkan Protokol RTP</i> | 31 |
| Gambar 3.13 <i>Konfigurasi Client 100</i> | 32 |
| Gambar 3.14 <i>Pemilihan Codec</i> | 32 |
| Gambar 3.15 <i>Diagram alir pengujian koneksi server</i> | 33 |
| Gamabr 3.16 <i>Diagram alir pengujian</i> | 34 |
| Gambar 4.1 <i>Hasil Uji koneksi client-server</i> | 36 |
| Gambar 4.2 <i>Hasil Konektifitas Server</i> | 36 |
| Gambar 4.3 <i>Uji Koneksi Antar Client</i> | 37 |
| Gambar 4.4 <i>Proses Panggilan VoIP</i> | 37 |
| Gambar 4.5 <i>Hasil Capturing Aplikasi Wireshark untuk data codec G.711</i> | 38 |
| Gambar 4.6 <i>Hasil Wireshark Summary RTP codec G.711</i> | 39 |
| Gambar 4.7 <i>Wireshark RTP stream</i> | 39 |

| | |
|---|----|
| Gambar 4.8 Hasil Wireshark RTP Stream Analysis..... | 40 |
| Gambar 4.9 Hasil Capturing <i>codec</i> G.722 dengan Wireshark..... | 40 |
| Gambar 4.10 Hasil Wireshark <i>Summary codec</i> G.722 | 41 |
| Gambar 4.11 Hasil Wireshark RTP Streams <i>Codec</i> G.722 | 41 |
| Gambar 4.12 Hasil Wireshark RTP Stream Analysis <i>Codec</i> G.722..... | 42 |
| Gambar 4.13 Hasil <i>Capturing codec</i> G.726-32 dengan Wireshark | 42 |
| Gambar 4.14 Hasil Wireshark <i>Summary codec</i> G.726-32 | 43 |
| Gambar 4.15 Hasil Wireshark RTP Streams <i>Codec</i> G.726-32..... | 43 |
| Gambar 4.16 Hasil Wireshark RTP Stream Analysis <i>Codec</i> G.726-32 | 44 |
| Gambar 4.17 Hasil <i>Traceroute</i> komunikasi dari <i>client</i> 1 ke <i>client</i> 2..... | 45 |
| Gambar 4.18 Hasil <i>Traceroute</i> dengan Rpi 2 di <i>disconnected</i> | 45 |
| Gambar 4.19 Hasil <i>Traceroute</i> dengan node Rpi 4 di <i>disconnected</i> | 46 |
| Gambar 4.20 Hasil <i>Traceroute</i> dengan node Rpi 3 di <i>disconnected</i> | 46 |
| Gambar 4.21 Diagram Perbandingan <i>Throughput</i> Keseluruhan <i>Codec</i> | 48 |
| Gambar 4.22 Diagram Rata-rata Perbandingan <i>Throughput</i> Keseluruhan <i>Codec</i> | 49 |
| Gambar 4.23 Diagram Perbandingan <i>Delay</i> Keseluruhan <i>Codec</i> | 50 |
| Gambar 4.24 Diagram Rata-rata Perbandingan <i>Delay</i> Keseluruhan <i>Codec</i> | 51 |
| Gambar 4.25 Diagram Perbandingan <i>Jitter</i> Keseluruhan <i>Codec</i> | 52 |
| Gambar 4.26 Diagram Rata-rata Perbandingan <i>Jitter</i> Keseluruhan <i>Codec</i> | 52 |
| Gambar 4.27 Diagram Perbandingan <i>Paket Loss</i> Keseluruhan <i>Codec</i> | 54 |
| Gambar 4.28 Diagram Rata-rata Perbandingan <i>Paket Loss</i> Keseluruhan <i>Codec</i> .. | 54 |



DAFTAR KODE

Kode 3.1 Konfigurasi Network Interfaces 29



DAFTAR LAMPIRAN

Lampiran 1 Konfigurasi sip.conf dan extensions.conf 59



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Voice over Internet Protocol (juga disebut VoIP, *IP Telephony*, *Internet telephony* atau *Digital Phone*) adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. Data suara diubah menjadi kode digital dan dialirkan melalui jaringan yang mengirimkan paket-paket data, dan bukan lewat sirkuit analog telepon biasa. Dengan adanya teknologi VoIP, biaya untuk melakukan komunikasi antara satu pengguna ke pengguna lainnya menjadi lebih efisien. Hal ini disebabkan karena VoIP tidak bergantung pada jarak dan bersifat global. Akan tetapi kemudahan dan efisiensi yang ditawarkan layanan VoIP juga memiliki kekurangan, seperti kualitas panggilannya sangat ditentukan oleh *bandwidth* yang tersedia. Jika *bandwidth* tidak memadai atau kualitas koneksi internet buruk, maka kualitas suara pun akan buruk dan panggilan biasanya menjadi tidak jelas, bahkan tidak menutup kemungkinan panggilan bisa terputus dan juga VoIP tidak toleran terhadap *delay*, *packet loss*, dan *jitter* (Mehta dkk, 2001). *Delay* didefinisikan sebagai waktu yang dibutuhkan untuk mengirimkan data dari sumber ke tujuan, *packet loss* merupakan kehilangan paket data pada proses transmisi (Iskandarsyah, 2003), sedangkan *jitter* adalah variasi total *delay* yang disebabkan perbedaan waktu kedatangan antar paket satu dengan paket lainnya (Mehta dkk, 2001).

Disisi lain teknologi komunikasi di bidang jaringan *nirkabel* telah mengalami perkembangan yang biasa dikenal dengan *Wireless Mesh Network*. *Wireless mesh network* (WMN) adalah jaringan komunikasi *wireless* yang terbentuk dari node radio dimana minimal terdapat dua atau lebih jalur komunikasi data pada setiap *node*. 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 (Febriadi, Rochim, & Widiyanto, 2013). Sebuah WMN memiliki sifat *self-configure* yaitu merupakan kemampuan node di jaringan untuk membangun dan mengkonfigurasi diri sendiri. Ketika perangkat dinyalakan maupun mendapati topologi yang berubah, ia akan menginformasikan alamatnya ke node tetangga agar kemudian dapat dikenali di jaringan. Ini juga termasuk mencari rute secara otomatis tanpa campur tangan manusia dengan memperhatikan kualitas link, tetangga, topologi dan konektivitas. Sifat-sifat inilah yang memungkinkan WMN untuk dapat secara otomatis membangun dan memelihara konektivitas *mesh* diantara *node – node* dalam jaringan. Selain itu, WMN juga memiliki sifat *self-healing* yaitu, sifat yang memungkinkan WMN dapat membenahi dirinya sendiri dan tetap berfungsi bahkan jika satu atau lebih node mengalami kerusakan atau dipindahkan dari satu lokasi ke lokasi lain. Sehingga kehilangan satu atau lebih node tidak menimbulkan downtime pada jaringan mesh. Dengan begitu jaringan WMN dapat mengatasi permasalahan *routing* jika ada salah satu *node* yang mati, atau biasa disebut

failure tolerance. Dengan *self-configure node* baru yang bergabung ke dalam jaringan WMN dapat langsung terhubung dengan *node-node* lainnya.

Akan tetapi timbul permasalahan apabila VoIP dijalankan di *wireless mesh network*, contohnya seperti saat komunikasi terjadi diantara kedua *client*, kemudian salah satu dari *node* WMN terputus dan melakukan *self-configure* karena pada saat itu *node* lain mencoba untuk membangun dan menkonfigurasi diri yang menyebabkan terhambatnya atau terlambatnya proses komunikasi yang terjadi. Bukan hanya itu saja WMN juga melakukan *self-healing* yang kemungkinan menyebabkan komunikasi dari VoIP terputus karena *self-healing* sendiri merupakan proses *re-routing* atau membenahi diri sendiri apabila salah satu node mengalami kerusakan.

Dari permasalahan tersebut penulis melakukan analisis kinerja VoIP pada *wireless mesh network* dengan melakukan pengujian komunikasi VoIP yang dijalankan di *wireless mesh network*. Lingkungan pengujian dilakukan di gedung C Filkom dengan menggunakan mini PC yaitu Raspberry Pi. Untuk parameter dalam pengujian kinerja menggunakan QoS (*Quality of Service*) yaitu kualitas dari jaringan saat komunikasi terjadi dengan melakukan beberapa skenario pengujian.

Hasil dari penelitian ini adalah berupa hasil kualitas jaringan yang diuji dengan menggunakan parameter QoS berupa *Throughput*, *delay*, *jitter*, serta *paket loss*. Dari hasil penelitian diharapkan dapat melihat tingkat keberhasilan komunikasi VoIP apabila dijalankan di *wireless mesh network*.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang diatas maka dirumuskannya suatu permasalahan yaitu:

1. Bagaimana kinerja dari VoIP yang di jalankan pada *wireless mesh network*?
2. Bagaimana kinerja codec VoIP G.711, G722 dan G.726-32 pada *wireless mesh network*?
3. Bagaimana melakukan analisis terhadap hasil pengujian VoIP menggunakan parameter QoS?

1.3 Batasan Masalah

Penelitian ini dibatasi oleh hal-hal sebagai berikut:

1. Perangkat lunak yang akan digunakan dalam analisis kinerja VoIP menggunakan *Raspbian-wheezzy*.
2. Analisis kinerja VoIP dilakukan di jaringan *wireless mesh network* dengan menggunakan *hardware Raspberry Pi*.
3. Seluruh perangkat adapter *wireless* yang digunakan pada *Raspberry Pi* memiliki spesifikasi yang sama.
4. Tidak membahas masalah *signalling*.

5. Pembahasan difokuskan untuk menguji QoS VoIP pada jaringan *wireless mesh network*.

1.4 Tujuan

Tujuan dari penulisan skripsi ini adalah sebagai berikut:

1. Melakukan analisis VoIP pada *wireless mesh network*.
2. Menguji kinerja VoIP pada *wireless mesh network*.
3. Mengetahui kualitas kinerja VoIP berdasarkan *parameter Quality of Service (QoS)*.

1.5 Manfaat

Adapun manfaat dari penelitian ini antara lain:

1. Untuk mengetahui kinerja dari VoIP pada *wireless mesh network*.
2. Untuk mengetahui jenis *codec* mana yang dapat di jalankan pada *wireless mesh network*.
3. Dapat membantu komunikasi masyarakat melalui jaringan internet dengan biaya yang terjangkau dalam pembuatannya.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini adalah sebagai berikut.

BAB I PENDAHULUAN

Pendahuluan menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan skripsi.

BAB II LANDASAN KEPUSTAKAAN

Membahas berbagai landasan teori dasar yang diambil dari kajian pustaka dan referensi literatur pendukung yang terkait dengan *Wireless Mesh Network, VoIP, Routing VoIP, codec* dan parameter QoS (*Quality of Service*).

BAB III METODOLOGI PENELITIAN DAN PERANCANGAN

Menjelaskan tentang tipe penelitian, perancangan, metode, dan implementasi sistem yang digunakan pada penelitian *VoIP* ini.

BAB IV HASIL DAN ANALISIS

Bab ini menjelaskan mengenai hasil pengujian dan analisis yang dilakukan pada jaringan *Wireless Mesh Network* untuk menganalisa hasil performa *VoIP* yang dijalankan pada jaringan tersebut.

BAB V PENUTUP

Memuat kesimpulan yang merangkum jawaban dari rumusan masalah berdasarkan hasil penelitian yang sudah didapat, dan saran untuk pengembangan penelitian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 *Wireless Mesh Network*

Wireless mesh network merupakan suatu bentuk jaringan komunikasi *wireless* yang terbentuk dari susunan node radio dimana setidaknya terdapat dua atau lebih jalur komunikasi pada setiap node. Node pada sebuah *wireless mesh network* dapat berupa sebuah *mesh router* ataupun *mesh client*. Setiap node tidak hanya bertindak sebagai sebuah host tetapi juga berfungsi sebagai *router* untuk meneruskan paket-paket pengiriman informasi bagi sebuah node lain yang mungkin tidak dapat menjangkau tempat yang ingin ditujunya.

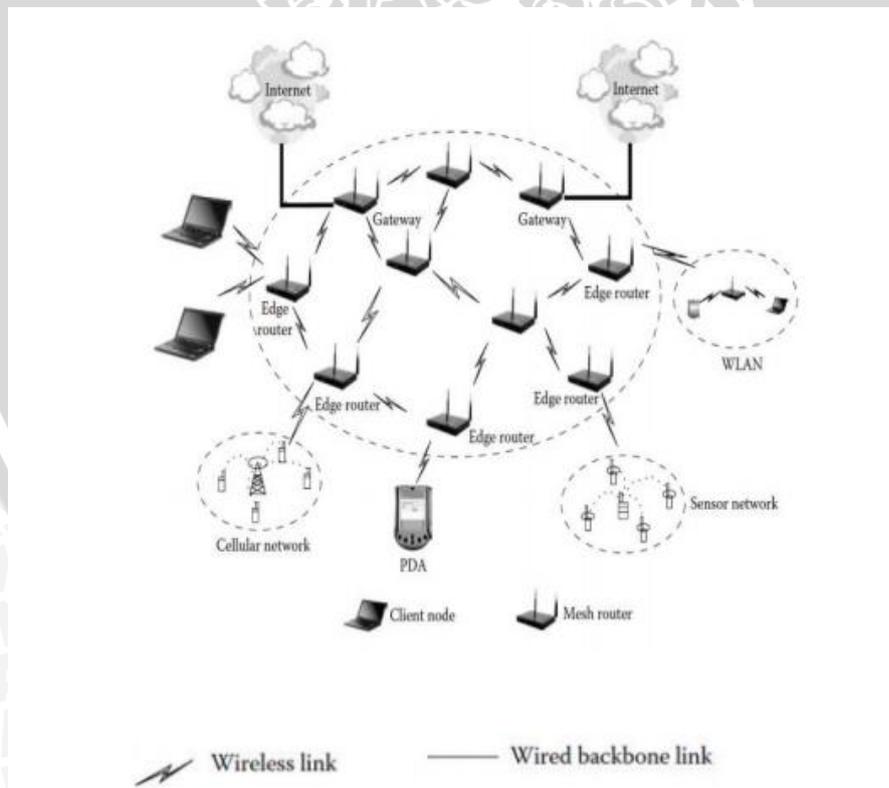
Karakteristik utama dari *wireless mesh network* adalah kemampuannya dalam mengkonfigurasi dan mengorganisasi dirinya sendiri (*self-configure/self-organize*), atau dengan kata lain mampu membuat dan menjaga konektivitasnya apabila terjadi kerusakan pada salah satu node. Kemampuan ini selain membantu para pengguna untuk dapat selalu on-line kapan saja dan dimana saja, juga akan membawa keuntungan lain seperti biaya pembuatan yang rendah, kemudahan dalam perawatan jaringan, tingkat *robustness* serta reliabilitas tinggi.

Node-node konvensional seperti desktop PC, laptop, PDA dan sebagainya yang telah dilengkapi dengan *wireless network interface card* (NIC) dapat tersambung langsung dengan *wireless mesh routers*. Sedangkan pengguna yang tidak mempunyai *wireless NIC*, tetap dapat terhubung dengan *mesh router* dengan menggunakan bentuk jaringan lain seperti *Ethernet*. Selain itu dengan menggunakan fungsi *mesh router* sebagai gateway atau bridge, maka suatu *wireless mesh network* dapat berintegrasi dengan jaringan *wireless* lainnya seperti jaringan seluler, Wi-fi, Wimax dan lain sebagainya (Akyildiz, Wang, & Wang, 2004).

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 kelebihanannya, *wireless mesh network* dapat digunakan untuk menyediakan layanan *wireless* untuk berbagai keperluan dan aplikasi baik untuk kepentingan pribadi, area lokal, kampus ataupun area metropolitan.

2.1.1 Arsitektur Wireless Mesh Network

Seperti telah disebutkan sebelumnya bahwa *wireless mesh network* mempunyai dua jenis node yaitu : *mesh router* dan *mesh client*. Untuk lebih meningkatkan fleksibilitas dari *mesh network*, maka suatu *mesh router* juga dapat dilengkapi dengan multiple *wireless interface*. Apabila dibandingkan dengan *wireless router* konvensional, maka *mesh router* dapat memiliki jangkauan area yang sama namun dengan daya transmisi yang jauh lebih rendah melalui komunikasi *multi-hop*. Walaupun dengan perbedaan tersebut, *wireless router* untuk *mesh* maupun konvensional biasanya tetap 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 networking*, 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 terdiri dari desktop PC, laptop, PDA, RFIP reader dan lain sebagainya (Zhang, Luo , & Hu, 2007).



Gambar 2.1 Arsitektur Wireless Mesh Network

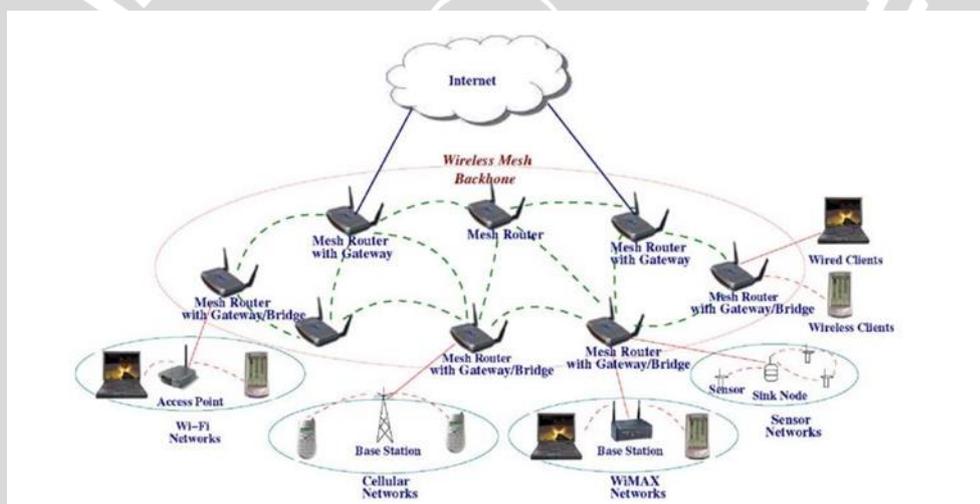
Sumber : (Pradhan, 2013)

Arsitektur WMN dapat diklasifikasikan menjadi tiga grup utama berdasarkan fungsionalitas *node*, yaitu:

2.1.1.1 Infrastruktur Wireless Mesh Network

Pada tipe ini, *mesh router* sebagai infrastruktur utama yang akan memberikan akses Internet bagi *client* seperti pada Gambar 2.1. Infrastruktur WMN dapat dibangun dengan berbagai tipe radio seperti yang biasa digunakan pada IEEE 802.11. Perangkat yang dapat digunakan sebagai *mesh router* adalah *wireless router* dan *embedded system*.

Mesh router membentuk jaringan *mesh* yang statis antar nodenya. *Mesh router* dapat tersambung ke Internet karena ia juga berperan sebagai *gateway*. Perangkat *client* dapat mengakses jaringan melalui *mesh router* terdekat yang juga berperan sebagai AP. Pada arsitektur tipe ini, *client* memainkan peran pasif dan tidak memberikan kontribusi dengan menyediakan fungsi routing.

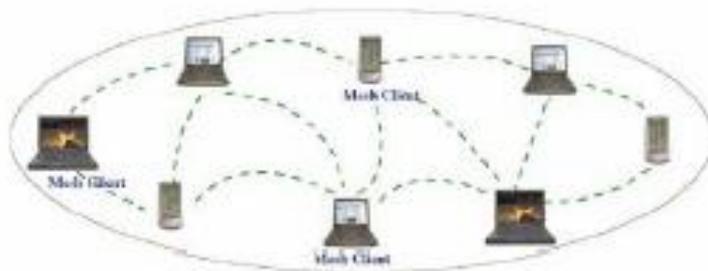


Gambar 2.2 Infrastruktur *Wireless Mesh Networks*

Sumber: (Pradhan, 2013)

2.1.1.2 Client Wireless Mesh Network

Client WMN atau bisa juga disebut *Ad hoc Mesh Network* bertujuan menyediakan jaringan untuk tujuan tertentu dengan pengaturan sementara sesuai dengan keperluan. Pada arsitektur ini, *mesh client* melakukan fungsi routing dan konfigurasi jaringan seperti pada Gambar 2.2. *Client* WMN biasanya dibentuk menggunakan satu tipe frekuensi radio pada tiap perangkatnya. Tipe arsitektur ini dapat berdiri walaupun tidak ada dukungan dari infrastruktur jaringan seperti *mesh router*. Pengguna dapat memperluas cakupan area sesuai dengan keberadaan perangkat *wireless* mereka.

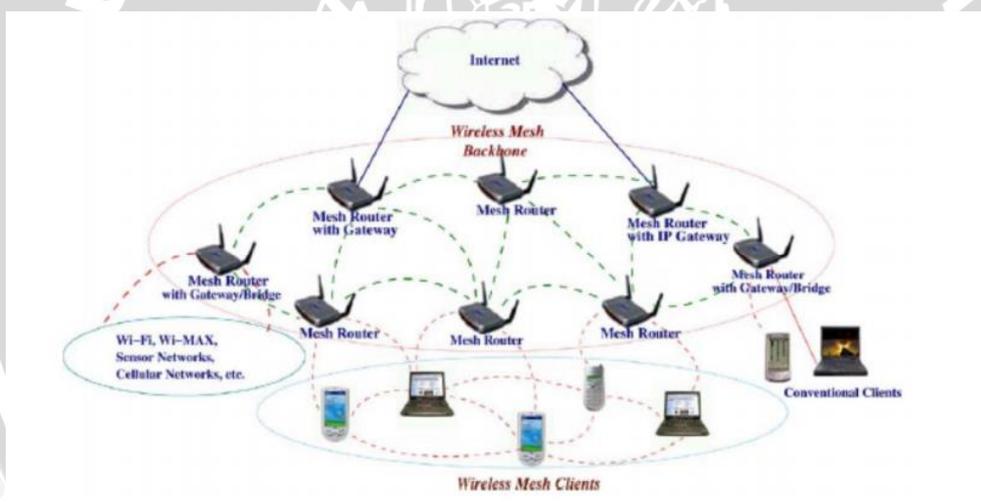


Gambar 2.3 Client Wireless Mesh Network

Sumber : (Ismail, Faroqi, Kamelia, & Mardiaty , 2011)

2.1.1.3 Hybrid Wireless Mesh Network

Arsitektur ini merupakan kombinasi dari tipe Infrastruktur dan *Client*. *Mesh client* dapat mengakses jaringan melalui *mesh router* seperti halnya secara langsung berkomunikasi dengan *mesh client* lainnya, seperti yang ditunjukkan pada Gambar 2.4.



Gambar 2.4 Hybrid Wireless Mesh Network

Sumber : (Ismail, Faroqi, Kamelia, & Mardiaty , 2011)

Tipe infrastruktur berperan untuk menyediakan konektivitas ke jaringan lain seperti Internet, Wi-Fi, seluler dan *sensor network*. Dan proses routing dari tipe *client* berguna untuk meningkatkan cakupan WMN. Tipe *hybrid* menggabungkan pendekatan statis dan dinamis dari tipe-tipe sebelumnya.

2.1.2 Babel Routing

Babel adalah protokol *Routing Proactive* berdasarkan algoritma Distance Vector yang merupakan evolusi dari harapan jumlah transmisi (ETX) algoritma dengan pemilihan rute yang lebih cerdas dari pengguna pendekatan *hop-count* sederhana (Abolhasan, Hagelstein, Chun, & Wang, 2009). Babel mempunyai dua

karakteristik khas untuk mengoptimalkan kinerja relay. Pertama, menggunakan *history-sensitive* pemilihan rute untuk meminimalkan dampak *route flaps* (Abolhasan, Hagelstein, Chun, & Wang, 2009). Kedua, babel mengesekusi *update* reaktif dan memaksa untuk meminta informasi ruting ketika mendeteksi kegagalan ruting dari salah satu *node* tetangga (Abolhasan, Hagelstein, Chun, & Wang, 2009).

2.2 VoIP

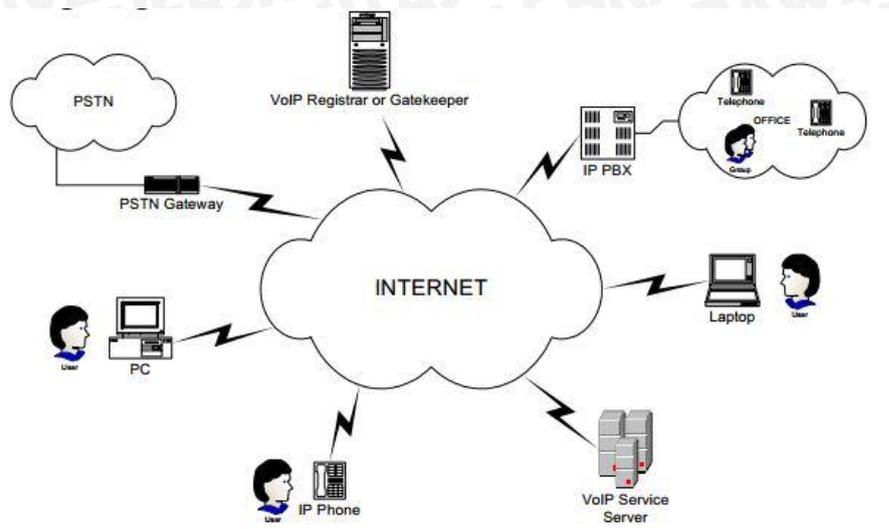
VoIP (*Voice over Internet Protocol*) adalah teknologi yang mampu melewati trafik suara, video dan data yang berbentuk paket secara *real-time* dengan jaringan *Internet Protocol*. VoIP ini dapat memanfaatkan infrastruktur internet yang sudah ada untuk berkomunikasi seperti layaknya menggunakan telepon biasa dan tidak dikenakan biaya telepon biasa untuk berkomunikasi dengan pengguna VoIP lainnya dimana saja dan kapan saja (Yuniati, Fitriawan, & Jaya Patih, 2014).

Teknik dasar *Voice over Internet Protokol* atau yang biasa dikenal dengan sebutan VoIP adalah teknologi yang memungkinkan kemampuan melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan (*networking*). Teknologi ini memungkinkan komunikasi suara menggunakan jaringan berbasis IP (*internet protokol*) untuk dijalankan diatas infrastruktur jaringan packet network. Jaringan yang digunakan data, dikirimkan dan dipulihkan kembali dalam bentuk voice dipenerima. *Voice* diubah dulu kedalam format digital karena lebih mudah dikendalikan dalam hal ini dapat dikompresi, dan dapat diubah keformat yang lebih baik dan data digital lebih tahan terhadap *noise* daripada *analog* (Yuniati, Fitriawan, & Jaya Patih, 2014). Sehingga kita bisa menggunakan VoIP di mana saja selama kita terhubung ke internet atau ke jaringan yang menggunakan *protokol* TCP/IP (Purbo & Raharja, 2011).

VoIP menjadi pilihan komunikasi yang menarik bagi konsumen karena VoIP menggunakan layanan internet dimana biaya untuk memperoleh layanan broadband dasarnya cenderung lebih rendah (Desantis, 2006).

2.2.1 Sistem Kerja VoIP

Pengiriman sebuah sinyal ke *remote destination* dapat dilakukan secara digital, yaitu sebelum dikirim data yang berupa sinyal analog, diubah dulu ke bentuk data digital dengan ADC (*analog to digital converter*) kemudian ditransmisikan, dan dipenerima dipulihkan kembali menjadi data analog dengan DAC (*digital to analog converter*). Begitu juga dengan VoIP, digitalisasi *voice* dalam bentuk packet data, dikirimkan dan dipulihkan kembali dalam bentuk *voice* dipenerima. *Voice* diubah dulu kedalam format digital karena lebih mudah dikendalikan dalam hal ini dapat *dikompresi*, dan dapat diubah *keformat* yang lebih baik dan data digital lebih tahan terhadap *noise* daripada analog (Yuniati, Fitriawan, & Jaya Patih, 2014).

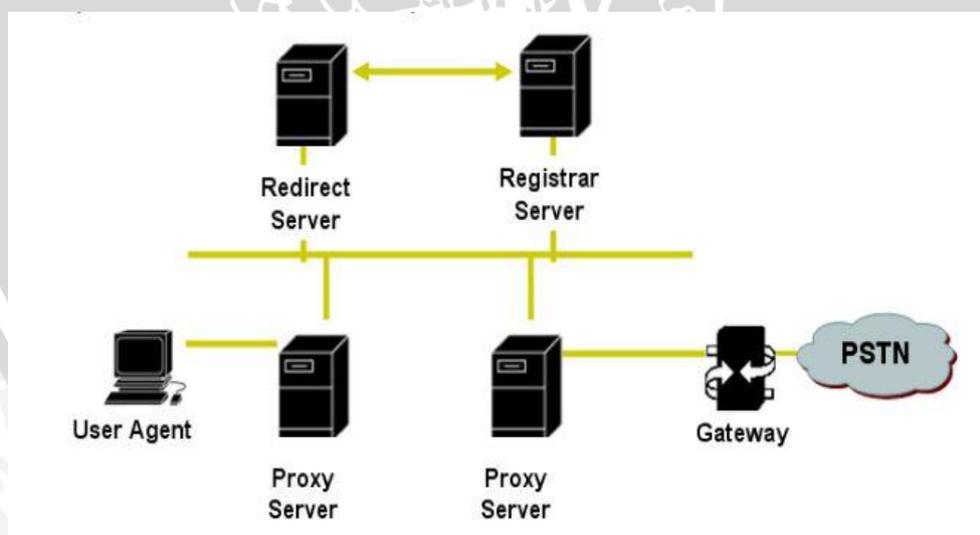


Gambar 2.5 VoIP secara umum

Sumber : (Noeranbia, 2009)

2.2.1.1 Session Initiation Protocol (SIP)

Session Initiation Protocol (SIP) merupakan sebuah protokol standart multimedia dimana merupakan produk dari *Internet Engineering Task Force (IETF)* dan telah digunakan menjadi suatu standart penggunaan VoIP. SIP merupakan protokol yang berada pada *layyer* aplikasi dimana mendefinisikan proses awal, pengubahan, dan pengakhiran (pemutusan) suatu sesi komunikasi multimedia.



Gambar 2.6 Arsitektur SIP

Sumber : (Noeranbia, 2009)

Dapat dikatakan juga SIP ini memiliki karakteristik *client-server*, dimana berarti *request* diberikan oleh *client* dan request ini diberikan ke *server*. Kemudian *server* mengolah *request* dan memberikan tanggapan terhadap *request* yang

diberikan *client*. Secara default, SIP menggunakan protokol UDP tetapi dapat juga menggunakan TCP sebagai *protokol transport*. *Request* dan Protokol yang mendukung SIP antara lain :

1. **Real-time Transport Protocol (RTP)**

Protokol RTP menyediakan *transfer* media secara terus-menerus pada jaringan paket. Protokol RTP menggunakan protokol UDP dan header RTP mengandung informasi kode bit yang spesifik pada tiap paket yang dikirimkan dimana hal ini membantu penerima untuk melakukan antisipasi jika terjadi paket yang hilang.

2. **Real-time Control Transport Protocol (RTCP)**

Protokol RTCP merupakan protokol yang mengendalikan *transfer media*. Protokol ini bekerja sama dengan *protokol* RTP dalam proses *transfer* media yang terjadi. Dalam satu sesi komunikasi, *protokol* RTP mengirimkan paket RTCP secara periodik untuk memperoleh informasi *transfer* media dalam perbaikan kualitas layanan.

3. **Session Description Protocol (SDP)**

Protokol SDP merupakan protokol yang mendeskripsikan media dalam suatu komunikasi. Tujuan *protokol* SDP adalah untuk memberikan informasi aliran media dalam satu sesi komunikasi agar penerimaan yang menerima informasi tersebut dapat berkomunikasi tanggapan terhadap *request* tersebut disebut transaksi SIP.

2.2.2 Codec

Codec adalah metode untuk mengompres sinyal digital agar ukurannya lebih kompak (padat). *Codec* bertujuan untuk mengurangi penggunaan bandwidth di dalam transmisi sinyal pada setiap panggilan dan sekaligus berfungsi untuk meningkatkan jumlah panggilan.

Konversi codec bekerja dengan cara memotong bagian sinyal (sampling) audio dalam jumlah tertentu perdetiknya. Sebagai contoh, *codec* G.711 melakukan sampling audio sebanyak 64.000 kali per detiknya. Jika data hasil *kompresi* berhasil diterima di titik lain, proses selanjutnya adalah melakukan perakitan ulang. Data yang dirakit tidak selengkap data saat pertama kali dikirim, ada beberapa bagian yang hilang. Akan tetapi bagian yang hilang sangat kecil sehingga tidak terdeteksi oleh telinga manusia.

Codec juga bekerja menggunakan algoritma tertentu untuk membantunya memecah, mengurutkan, mngkompresi, dan merakit ulang audio data yang ditransmisikan. Salah satu algoritma yang populer digunakan dalam teknologi VoIP adalah CS-ACELP (*Conjugate-Structure Algebraic-Code-Excited Linear Prediction*).

Tabel 2.1 Macam – Macam CODEC

| Tipe Codec | Bandwidth /Kbps | Algorithm |
|------------|-----------------|-----------|
| G.711 | 64 | PCM |
| GSM FR | 13 | RPE LTP |
| G.722 | 64 | ADPCM |
| G.723.1 | 5.3/6.4 | ACELP |
| G.726 | 16/24/32/40 | ADPCM |
| G.728 | 16 | LD-CELP |
| G.729A | 8 | CS-ACELP |

Sumber: Diadaptasi dari (Al-Quzwini & Sharafali, 2014).

2.2.3 Kualitas layanan VoIP (QoS)

Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang lebih baik pada trafik data tertentu pada berbagai jenis platform teknologi. *QoS* tidak diperoleh langsung dari infrastruktur yang ada, melainkan diperoleh langsung dengan mengimplementasikannya pada jaringan bersangkutan (Thabratas & Purbo, 2001).

Aplikasi VoIP merupakan aplikasi real time, sehingga tidak dapat mentolerir *delay* (dalam batasan tertentu) dan *packet loss*. *Delay* dapat diminimalkan dengan menggunakan teknologi *packet switching* sebagai pengganti data *switching*. Cara lain yang dapat ditempuh adalah mengoptimalkan penggunaan *bandwidth*, mengatur metode antrian yang dipakai dan menggunakan protokol-protokol manajemen untuk mengatur paket data yang dilewatkan. *QoS* pada IP Telephony adalah parameter-parameter yang menunjukkan kualitas paket data jaringan, agar didapatkan hasil suara sama dengan menggunakan telepon tradisional (PSTN) (Sudiarta & Sukadarmika, 2009).

2.2.3.1 Throughput

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

Untuk mencari besar *throughput* dalam komunikasi suara dapat dicari menggunakan Persamaan 2.1.

$$throughput = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}} \quad (2.1)$$

2.2.3.2 Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari suatu titik ke titik lain yang menjadi tujuan. *One Way Delay* (OWD) adalah waktu yang dibutuhkan oleh satu paket dari tempat sumber ke tujuan.



Waktu dari sumber ke tujuan kembali lagi ke sumber disebut *Round Trip Time* (RTT). Dalam *delay* terdapat beberapa besaran *delay* maksimum yang direkomendasikan oleh ITU untuk aplikasi suara adalah 150 ms, sedangkan *delay* maksimum dengan kualitas suara yang masih dapat diterima pengguna adalah 250 ms. *Delay end to end* adalah jumlah *delay* konversi suara analog-digital, *delay* waktu paketisasi atau bisa disebut juga *delay* panjang paket dan *delay* jaringan pada saat t (waktu). Ada beberapa penyebab terjadinya *delay* antara lain :

1. Kongesti (kelebihan beban data).
2. Kekurangan pada metode traffic shaping.
3. Penggunaan paket-paket data yang besar pada jaringan berkecepatan rendah
4. Adanya paket-paket data dengan ukuran berbedabeda
5. Perubahan kecepatan antar jaringan WAN
6. Pemadatan bandwidth secara tiba-tiba

Trafik suara merupakan trafik *realtime* sehingga jika *delay* dalam pengiriman paket suara terlalu besar, ucapan yang disampaikan tidak dapat dikenali. *Delay* maksimum yang ditolerir pada transmisi sinyal suara sesuai dengan standar ITU G.114 yang direkomendasikan bahwa *delay* kumulatif harus ≤ 150 mdetik (1-way *delay*) (Walker, 2002).

Delay yang berpengaruh dalam komunikasi VoIP adalah *delay* propagasi. Untuk mencari besar *delay* dalam komunikasi suara dapat dicari menggunakan Persamaan 2.2.

$$delay = \frac{waktu}{jumlah\ paket} \tag{2.2}$$

Tabel 2.2 Kategori Delay

| Kategori Degradasi | Besar <i>Delay</i> |
|--------------------|--------------------|
| <i>Good</i> | < 150ms |
| <i>Acceptable</i> | 150 s/d 300 ms |
| <i>Poor</i> | 300 s/d 450 ms |

Sumber : (Suryawan, Husni, & dkk, 2012)

2.2.3.3 Jitter

Jitter merupakan variasi *delay* yang terjadi akibat adanya selisih waktu atau *interval* antar kedatangan paket di penerima. Parameter ini dapat ditangani dengan mengatur metode antrian pada *router* saat terjadi kongesti atau saat perubahan kecepatan. Paket data yang datang dikumpulkan dulu dalam *jitter buffer* selama waktu yang telah ditentukan sampai paket dapat diterima pada sisi penerima dengan urutan yang benar. Hanya saja *jitter* tidak mungkin dihilangkan sebab metode antrian yang paling baik tetap saja tidak dapat mengatasi semua kasus antrian. Untuk meminimalisasi *jitter* ini, diusahakan agar pengiriman tiap-tiap paket data melalui jalur yang sama dan jangan sampai terjadi *paket loss* atau

kongesti jaringan. Selain *jitter* QoS juga dipengaruhi oleh *echo*. *Echo* disebabkan perbedaan impedansi dari jaringan yang menggunakan *four-wire* dengan *two-wire*. Efek *echo* adalah suatu efek yang dialami mendengar suara sendiri ketika sedang melakukan percakapan. Mendengar suara sendiri pada waktu lebih dari 25 ms dapat menyebabkan terhentinya pembicaraan (Sudiarta & Sukadarmika, 2009).

Tabel 2.3 Kategori Jitter

| Kategori Degradasi | Jitter |
|--------------------|-------------|
| <i>Good</i> | 0 s/d 20 ms |
| <i>Acceptable</i> | 0 s/d 50 ms |
| <i>Poor</i> | > 50 ms |

Sumber : (Suryawan, Husni, & dkk, 2012)

2.2.3.4 Paket Loss

Merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena collision dan congestion pada jaringan dan hal ini berpengaruh pada semua aplikasi karena retransmisi akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah bandwidth cukup tersedia untuk aplikasi aplikasi tersebut. *Packet loss* pada jaringan untuk untuk IP telephony sangat besar pengaruhnya, dimana bila terjadi packet loss dalam jumlah tertentu, akan menyebabkan terjadi interkoneksi TCP melambat (Sudiarta & Sukadarmika, 2009).

Tabel 2.4 Kategori Paket loss

| Kategori Degradasi | Paket Loss |
|--------------------|------------|
| Sangat Bagus | 0 % |
| Bagus | 3 % |
| Sedang | 15 % |
| Jelek | 25 % |

Sumber : (Suryawan, Husni, & dkk, 2012)

2.3 Perangkat Pendukung

Dalam pengimplementasian VoIP dibutuhkan 2 jenis perangkat yaitu beberapa *software* (perangkat lunak) dan beberapa *hardware* (perangkat keras) agar VoIP dapat berjalan dengan baik. Berikut adalah penjelasan beberapa perangkat lunak tersebut:

2.3.1 Asterisk

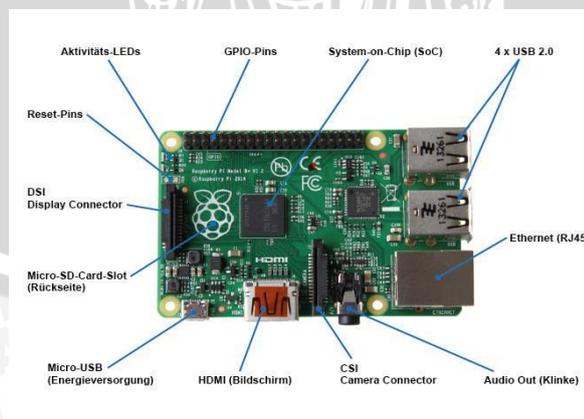
Asterisk merupakan perangkat lunak PBX opensource, diciptakan dari mendesakannya kebutuhan dibidang telekomunikasi. Mark Spencer, pencipta Asterisk membutuhkan sistem PBX khusus untuk memungkinkan model bisnis "Linux Support Service", sebuah perusahaan support opensource miliknya. Mark memerlukan suatu sistem untuk menerima pesan suara dan memberikan pesan ke orang terdekatnya. Karena ia hanya memiliki 4000 dollar pada saat itu, akhirnya dia berusaha menjadi seorang hacker kode dan pekerjaannya hanya duduk dan menulis sistemnya sendiri. Menurut Mark, nama Asterisk didapat dari wildcard UNIX yaitu simbol *. Mark membayangkan Asterisk sebagai semua jenis platform telepon, yang akan mampu melakukan lebih dari sekedar telepon (Simionovich, 2008).

2.3.2 Raspberry Pi

Perkembangan teknologi komputer dewasa ini semakin pesat. Dengan menggunakan komputer, berbagai macam pekerjaan dapat dilakukan dengan mudah. Oleh karena kebutuhan komputer yang tinggi, berbagai macam pengembangan dilakukan untuk menciptakan komputer yang dapat disesuaikan dengan kebutuhan-kebutuhan tertentu. Salah satu produk perkembangan komputer yang banyak diminati adalah Raspberry Pi.

Raspberry Pi (RPI) merupakan komputer dengan ukuran sebesar kartu kredit yang dapat ditancapkan ke monitor komputer ataupun layar TV serta dapat menggunakan keyboard dan mouse standar untuk mengoperasikannya.

Walaupun memiliki ukuran yang jauh lebih kecil dibandingkan komputer pada umumnya, tidak mengurangi kemampuan RPI dalam melakukan pengolahan kata, memutar video beresolusi tinggi bahkan memainkan permainan seperti yang bisa dilakukan komputer desktop (Cleevely, Lang, & Lomas, 2016).



Gambar 2.8 Raspberry Pi model B+

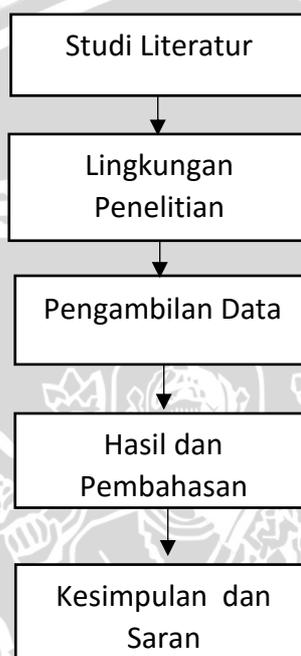
Sumber: (Cleevely, Lang, & Lomas, 2016)

Dalam *Raspberry Pi* ini sudah dilengkapi dengan prosesor ARM1176JZF-S 700 MHz, RAM sebesar 256 MB dan juga sebuah GPU VideoCore IV. Dan untuk penyimpanan data, *Raspberry Pi* tidak menggunakan Hard Disk namun *Raspberry Pi* dapat menggunakan SD Card untuk menyimpan data, baik itu data *Operating System* ataupun untuk media penyimpanan data jangka panjang.



BAB 3 METODOLOGI

Tipe penelitian ini merupakan penelitian *non-implimentatif analitik*. Langkah – langkah yang dilakukan dalam penelitian analisis kinerja VoIP pada *wireless mesh network* yaitu, study literatur, lingkungan penelitian, pengambilan data, hasil, pembahasan, kesimpulan dan saran. Adapun diagram alir metodologi penelitian seperti pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan teori pendukung dan ilmu-ilmu dasar yang digunakan dalam penelitian. Penelusuran literatur dasar pengetahuan mengenai hal-hal yang berkaitan dengan penelitian ini diperoleh dari buku, paper, *ebook*, dan jurna-jurnal ilmiah. Beberapa referensi terkait dalam penelitian ini, diantaranya adalah:

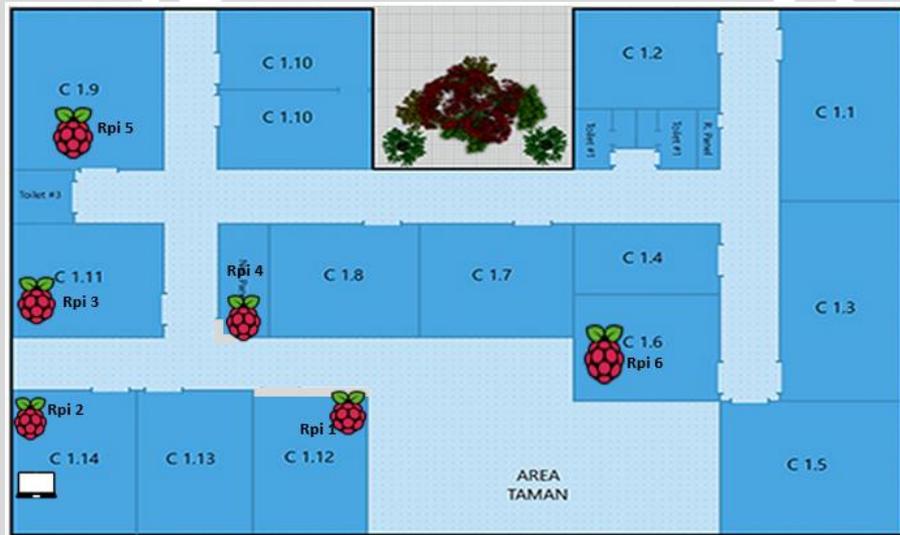
1. Wireless Mesh Network
 - a. Arsitekture *wireless mesh network*
2. VoIP (Voice Over Internet Protocol)
 - a. Sistem kerja VoIP
 - b. Protocol VoIP
 - c. Codec
 - d. Kualitas layanan VoIP

3. Raspberry Pi

- a. Pengenalan Raspberry Pi
- b. Spesifikasi Raspberry Pi

3.2 Lingkungan Penelitian

Lingkungan penelitian dilakukan di Gedung C FILKOM Universitas Brawijaya, untuk Lokasi pengujian dapat dilihat pada **Gambar 3.2**, dimana masing – masing Rpi ditempatkan di beberapa ruangan yang telah ditentukan dan laptop sebagai server diletakan pada ruangan C1.14 bersamaan dengan salah satu RPi, pada lokasi ini hanya digunakan 1 topologi karena pemasangan Rpi dilakukan dengan permanen atau tidak dapat diubah lokasi penempatannya.



Gambar 3.2 Denah Lokasi Penelitian di Gedung C FILKOM UB

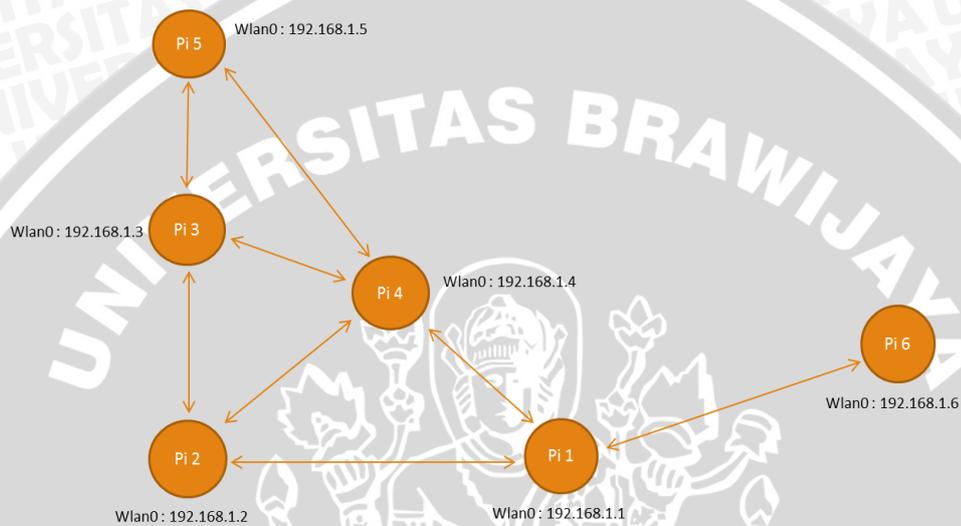
Berdasarkan Gambar 3.2 ditentukan ruang pemasangan Rpi. Sebelum melakukan pemasangan, Rpi diberikan nomor fisik di setiap casing dengan jumlah nomor 1 hingga 6 agar mudah untuk dikenali.

Tabel 3.1 Daftar ruang pemasangan Rpi

| Nomor Rpi | Ruang |
|-----------|-------------|
| 1 | C1.12 |
| 2 | C1.14 |
| 3 | C1.11 |
| 4 | Ruang Panel |
| 5 | C1.9 |
| 6 | C1.6 |

3.2.1 Topologi Jaringan

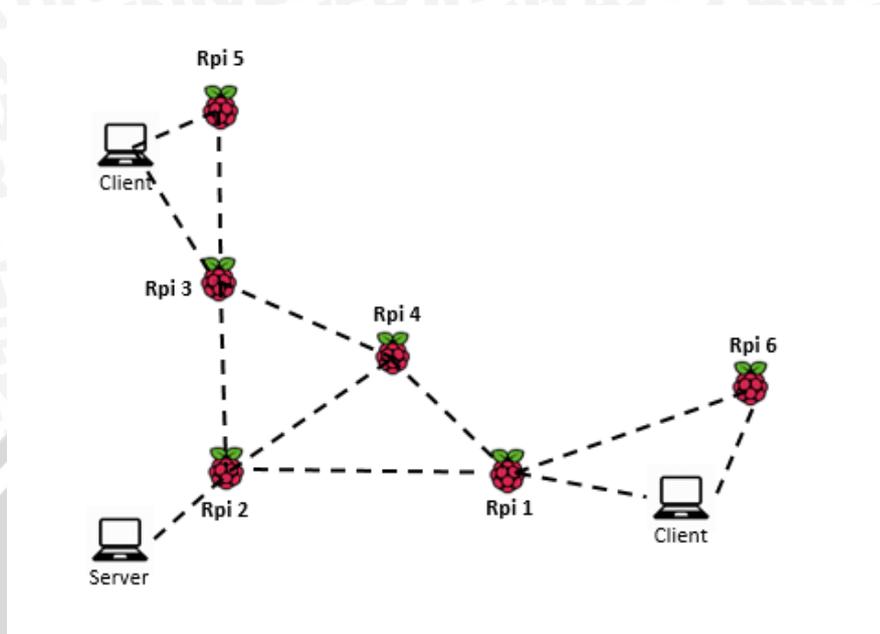
Pada pembangunan Wireless Mesh Network yang perlu diperhatikan adalah infrastruktur yang tersedia guna mewujudkan komunikasi antara node yang bersifat *multihop*, maka diperlukan lebih dari 2 node yang terdapat di jaringan. Pada *hardware dongle* dari edimax raspberry pi untuk *channel radio wireless* diatur dengan channel 4 untuk mengurangi interferensi dari jaringan *wireless* yang lain di lokasi implementasi. Pengalamatan pada tiap node diatur satu subnet untuk memudahkan identifikasi.



Gambar 3.3 Topologi Wireless Mesh Network di Gedung C

Dari gambar 3.3 dapat dilihat bahwa di dalam Gedung C terdapat 6 buah *Raspberry Pi* yang digunakan sebagai akses point.

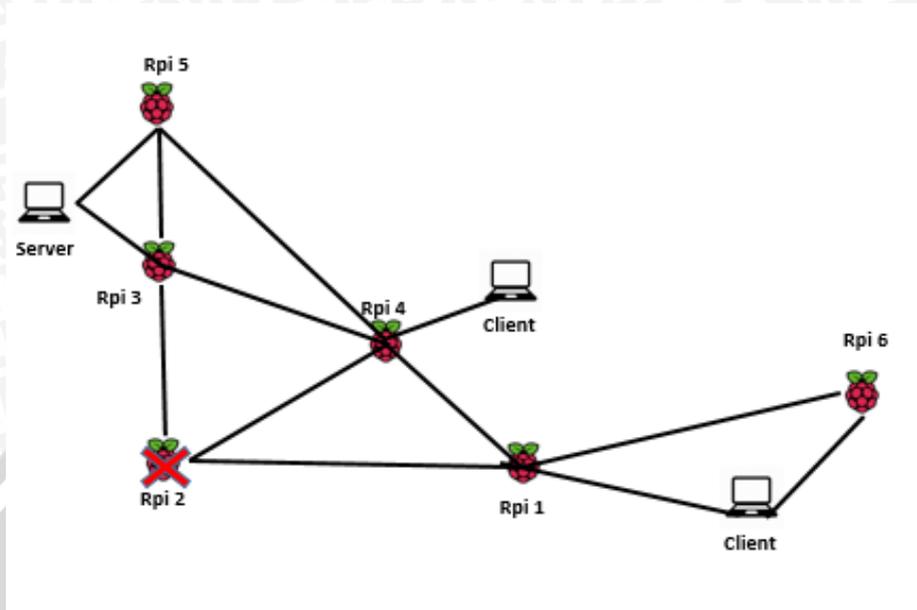
3.2.1.1 Skenario Pengujian 1



Gambar 3.4 Pengujian Komunikasi VoIP

Pada penelitian ini, sejumlah Rpi dipasang secara permanen di 6 ruangan yang berbeda, pada tabel 3.1 sudah disebutkan tempat-tempat dimana Rpi dipasang. Laptop yang bertindak sebagai *server* diletakkan di ruangan C1.14 yang juga merupakan tempat dari Rpi 2, untuk *client* 1 berada di belakang antara ruangan C1.11 dan C1.9 yang merupakan tempat Rpi 3 dan Rpi 5 berada, sedangkan *client* 2 berada di samping ruangan C1.12 yang merupakan tempat Rpi 1 berada. Sebelum melakukan komunikasi VoIP dilakukan pengujian koneksi antara *client-server* dengan melakukan *ping* kepada *ip client* dan *ip sever* untuk memastikan bahwa *client* dapat mengakses *server* dan *server* dapat mengakses *client* menggunakan *wireless mesh network*, setelah *server* mengenali semua *client* maka pengujian koneksi *client-client* diperlukan untuk memastikan bahwa kedua *client* dapat terhubung. Pengujian tersebut dilakukan seperti seperti sebelumnya yaitu dengan mengetikkan perintah *ping* pada *ip client* 1 dan *ip client* 2. Apabila koneksi sudah terbentuk maka dilakukan pengujian VoIP yang menggunakan *protocol* RTP untuk melakukan komunikasi selama 5 menit dan dilakukan sebanyak 10 kali panggilan dengan menggunakan 3 *codec* yang berbeda-beda yaitu G.711, G.722 dan G.726-32. Pada saat komunikasi berlangsung akan dilakukan pengambilan data QoS dengan melakukan *capture* data menggunakan *Wireshark*.

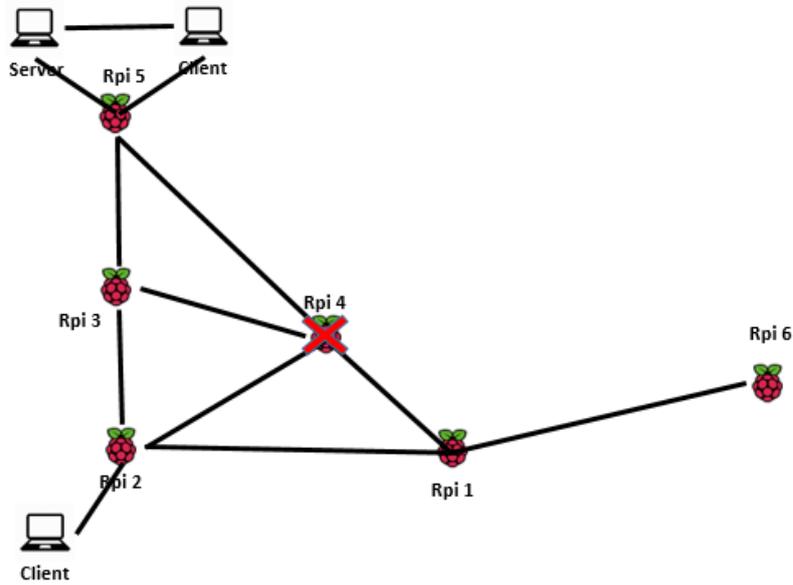
3.2.1.2 Skenario Pengujian 2



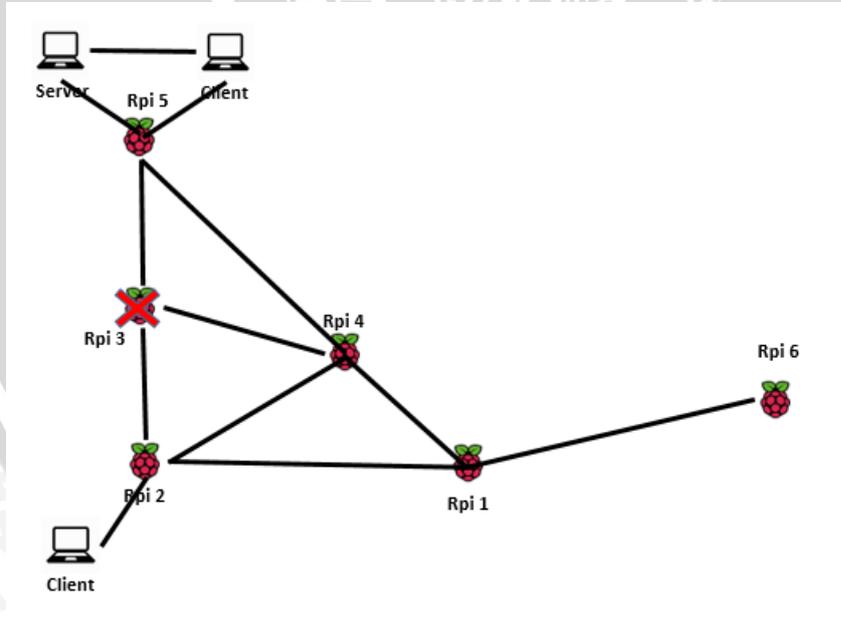
Gambar 3.5 Pengujian Komunikasi Saat 1 Node Disconnected

Pada pengujian VoIP skenario ini bentuk dari topologi sebelumnya tidak dirubah akan tetapi posisi dari *server* dan salah satu *client* sudah diubah, dimana *server* diletakkan di antara Rpi 3 dan Rpi 5, sedangkan posisi untuk kedua *client* yang pertama berada diantara Rpi 1 dan Rpi 6, *client* kedua berada di ruang panel yaitu di Rpi 4. Bentuk topologi ini untuk mengetahui ketika komunikasi berlangsung antara *client* 1 dan *client* 2 kemudian 1 *node* dari *wireless* Rpi diputus maka apakah komunikasi tersebut masih dapat berjalan dengan normal atau terjadi kendala. Pengujian diawali dengan melakukan panggilan antara *client* 1 dan *client* 2 selama 6 menit disaat memasuki menit ke 3 *wireless* Rpi yang berada di ruang C1.14 di buat *down* atau *disconnected*.

3.2.1.3 Skenario Pengujian 3



Gambar 3.6 Pengujian dengan Node Rpi 4 Disconnected

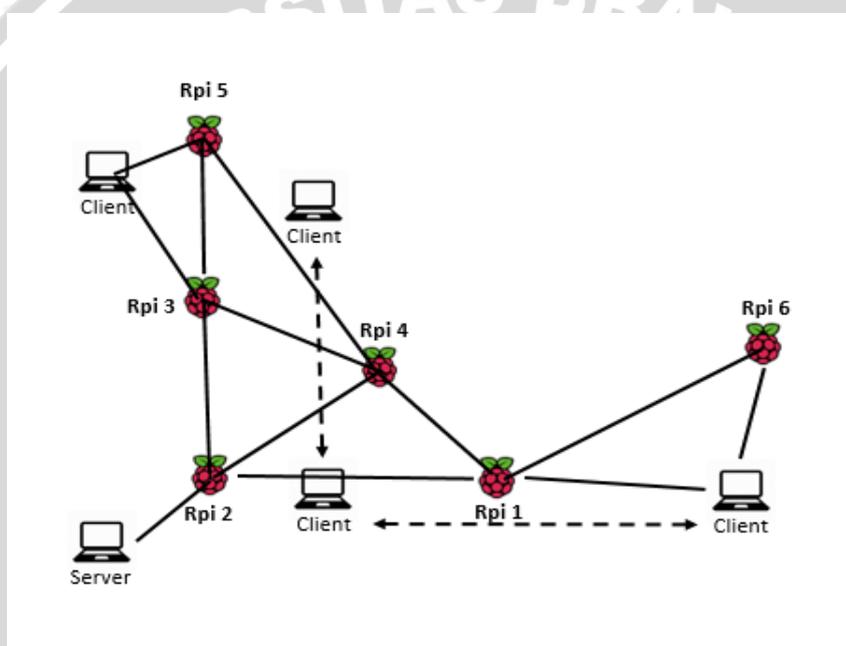


Gambar 3.7 Pengujian dengan Node Rpi 3 Disconnected

Pada Gambar 3.6 bentuk topologi Rpi tidak berubah dengan tetap menggunakan 6 node yang saling berkomunikasi. Akan tetapi posisi dari client 1, client 2 dan server yang diubah dengan menempatkan client 1 berada di Rpi 5

begitu juga dengan server dan client berada di dekat Rpi 2. Proses tersebut dimulai ketika client 1 menghubungi client 2 melalui server dengan rentang waktu 10 menit tanpa diputus. Pada saat komunikasi tersebut sedang berlangsung maka wireless pada Rpi 4 diputuskan pada menit ke 4 dan di menit ke 5 wireless Rpi 4 kembali dinyalakan. Topologi dengan posisi Rpi tetap serta *client server* juga masih berada di tempat yang sama dengan komunikasi tetap berjalan. Proses yang dilakukan sama seperti yang sebelumnya disaat komunikasi berlangsung maka wireless pada Rpi 3 diputus pada menit ke 7 dan pada menit ke 10 proses komunikasi dihentikan.

3.2.1.4 Skenario Pengujian 4



Gambar 3.8 Pengujian dengan *Client* Bergerak (Mobile)

Pada penelitian ini, terdapat satu komunikasi yang bejalan dan dilakukan oleh 2 *client*. Pada saat 2 *client* sedang melakukan komunikasi maka salah satu dari *client* akan berada pada kondisi bergerak bolak – balik melewati server dengan kecepatan yang konstan. Proses pengujian yang telah dilakukan pada salah satu *client* bergerak secara konstan bolak-balik mendekati dan menjauhi *server* yaitu *client* 1 berjalan diantara Rpi 1, Rpi 2 dan Rpi 6, proses komunikasi tersebut dijalankan dengan rentang waktu 5 menit dan dilakukan 6 kali komunikasi menggunakan 2 *codec* yang berbeda yaitu *codec* G.711 dan *codec* G.726-32. Proses yang kedua *client* tetap berjalan secara konstan dengan melewati antara Rpi 2, Rpi 3 dan Rpi 5 yang komunikasi tersebut sama dengan yang pertama dijalankan selama 5 menit dan dilakukan 6 kali komunikasi menggunakan 2 *codec* seperti yang sudah digunakan sebelumnya.

3.2.2 Implementasi

Lingkungan penelitian pada sub-bab ini membahas tentang *konfigurasi* dari paket-paket yang dibutuhkan untuk mendukung penelitian. Karena penelitian difokuskan pada analisis kinerja voip pada *wireless mesh network* yang merupakan suatu komunikasi antar *client* dan antar *node*, maka mode *wireless* diatur menjadi mode *ad-hoc*.

3.2.2.1 Konfigurasi Wireless Mesh Network Interfaces

Konfigurasi dilakukan terlebih dahulu sebelum perangkat terpasang permanen. Dalam hal ini mengganti *hostname* diperlukan karena untuk memberikan kemudahan dalam mengenali perangkat *Raspberry Pi* yang dipasang. Konfigurasi dapat dilakukan pada file `/etc/hostname` dan `/etc/hosts`.

```
pi@jarkom3 ~ $ sudo su
root@jarkom3:/home/pi# nano /etc/hosts
root@jarkom3:/home/pi# nano /etc/hostname
root@jarkom3:/home/pi# █
```

Gambar 3.9 Konfigurasi Host dan Hostname

Konfigurasi *network interfaces* diatur untuk setiap perangkat yang digunakan sebagai *mesh cilent*. Pemberian alamat IP dibutuhkan sebagai identitas perangkat dalam proses routing. Pada penelitian ini, tiap perangkat diatur dalam satu *subnet* untuk memudahkan indentikasi. Sedangkan untuk *wireless* diatur ke mode *ad-hoc*, di mana mode ini digunakan agar perangkat dapat berkomunikasi secara langsung dalam jangkauan sinyal yang satu dengan yang lain tanpa melalui *Access Point* terlebih dahulu. Agar setiap perangkat mendapatkan sinyal yang baik, konfigurasi *wireless channel* diatur pada *channel* yang mempunyai *interferensi* minimal di lokasi pengujian. Konfigurasi *network interfaces* bisa dilihat pada file `/etc/network/interfaces` seperti yang ditunjukkan pada Kode 3.1.

```
auto lo
iface lo inet loopback

auto eth0
allow-hotplug eth0
iface eth0 inet manual

auto wlan0
iface wlan0 inet static
address 192.168.1.3
netmask 255.255.255.0
wireless-channel 4
wireless-essid mesh
wireless-mode ad-hoc
```

Kode 3.1 Konfigurasi Network Interfaces

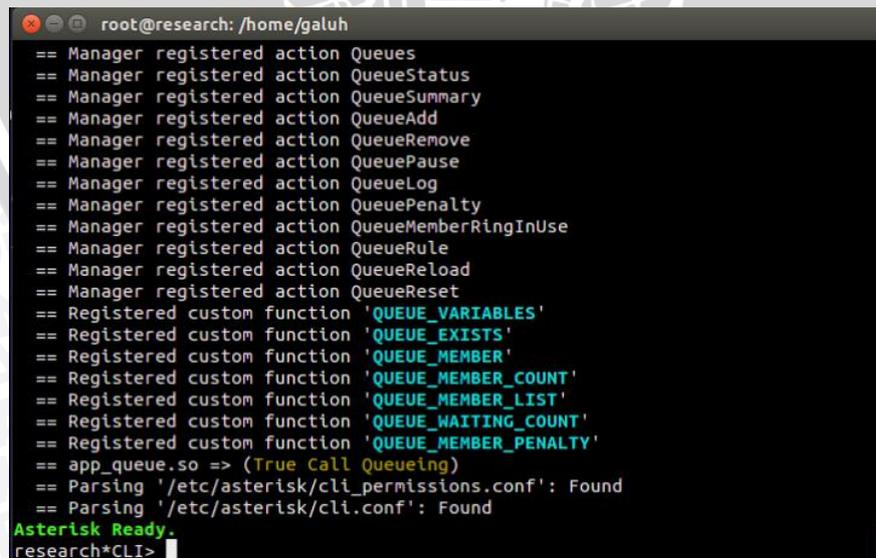
Pemilihan *channel* didapat dari pemindaian sinyal *wireless* melalui aplikasi *wifi Analyzer*. ESSID dan MAC *address* setiap perangkat dibuat sama untuk sebagai indentifikasi perangkat yang akan bergabung di jaringan yang sama. Seperti yang terlihat pada Tabel 3.2.

Tabel 3.2 Daftar Alamat IP Wlan0

| Nomor <i>Raspberry Pi</i> | Alamat IP Wlan0 |
|---------------------------|-----------------|
| 1 | 192.168.1.1 |
| 2 | 192.168.1.2 |
| 3 | 192.168.1.3 |
| 4 | 192.168.1.4 |
| 5 | 192.168.1.5 |
| 6 | 192.168.1.6 |

3.2.2.2 Implementasi Asterisk 13.

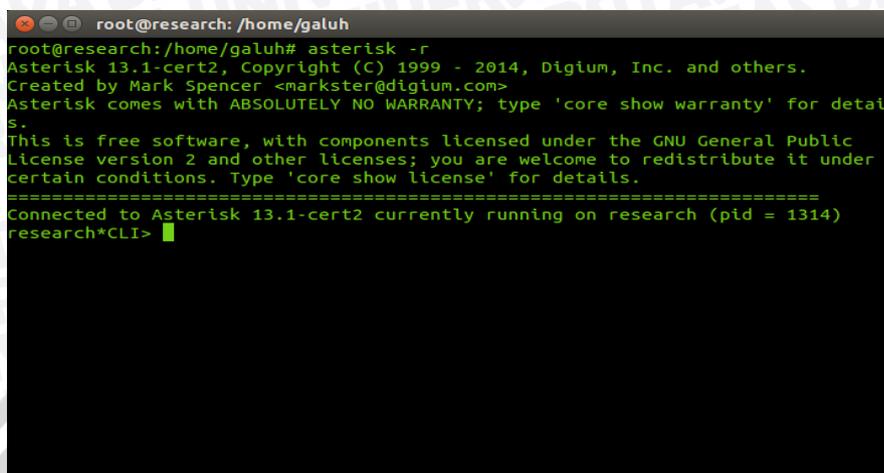
Proses instalasi dilakukan pada sebuah laptop yang digunakan untuk implemetasi server asterisk. Proses awal yang harus dilakukan setelah melakukan update sistem operasi pada laptop adalah meinstall *additional packages* yang dibutuhkan. Ketika melakukan proses install ada beberapa hal yang perlu menjadi perhatian seperti pemilihan *packages* pada saat proses *#make menuselect*. *Packages* yang dipilih harus sesuai dengan kebutuhan seperti paket dalam *Add-ons*, *Core Sound Packages*, dan *Extras Sound Packeges*. Jika proses instalasi selesai, maka perlu untuk memeriksa keberhasilan implementasi dengan mengetikkan perintah *asterisk – vvvvvvvc* pada terminal, seperti yang terlihat pada Gambar 3.10.



```
root@research: /home/galuh
== Manager registered action Queues
== Manager registered action QueueStatus
== Manager registered action QueueSummary
== Manager registered action QueueAdd
== Manager registered action QueueRemove
== Manager registered action QueuePause
== Manager registered action QueueLog
== Manager registered action QueuePenalty
== Manager registered action QueueMemberRingInUse
== Manager registered action QueueRule
== Manager registered action QueueReload
== Manager registered action QueueReset
== Registered custom function 'QUEUE_VARIABLES'
== Registered custom function 'QUEUE_EXISTS'
== Registered custom function 'QUEUE_MEMBER'
== Registered custom function 'QUEUE_MEMBER_COUNT'
== Registered custom function 'QUEUE_MEMBER_LIST'
== Registered custom function 'QUEUE_WAITING_COUNT'
== Registered custom function 'QUEUE_MEMBER_PENALTY'
== app_queue.so => (True Call Queueing)
== Parsing '/etc/asterisk/cli_permissstons.conf': Found
== Parsing '/etc/asterisk/cli.conf': Found
Asterisk Ready.
research*CLI>
```

Gambar 3.10 Hasil Instalasi Asterisk Server

Sedangkan untuk masuk *console CLI* asterisk dapat mengetikkan perintah *asterisk -r* pada terminal.



```

root@research:/home/galuh
root@research:/home/galuh# asterisk -r
Asterisk 13.1-cert2, Copyright (C) 1999 - 2014, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 13.1-cert2 currently running on research (pid = 1314)
research*CLI>

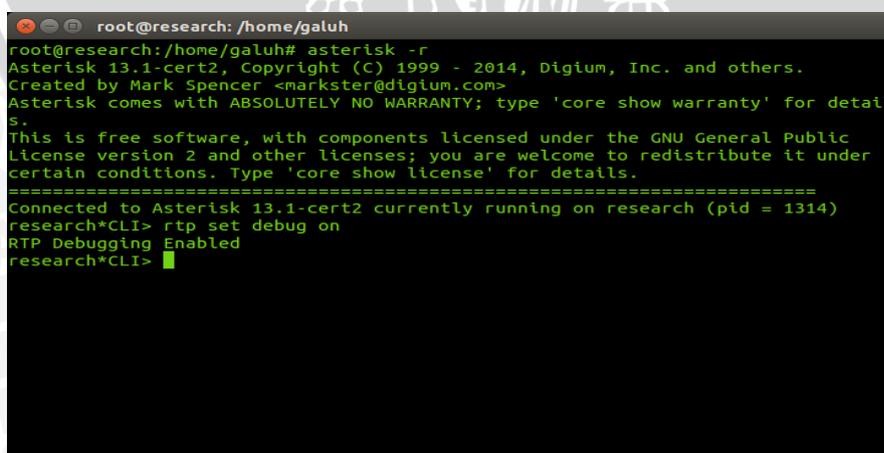
```

Gambar 3.11 Console CLI Asterisk Server

Kemudian proses selanjutnya setelah implementasi asterisk adalah melakukan konfigurasi *client* di server dengan cara yaitu memasukkan beberapa file tambahan secara langsung di dua file berekstensi *.conf* yang terdapat pada direktori */etc/asterisk*. File tersebut untuk mengatur, menampung dan mendaftarkan data *client* di server. Untuk konfigurasi datanya dapat mengetikkan */etc/asterisk/sip.conf* pada terminal. Sedangkan file untuk mengatur extension antar *client* yang telah terdaftar pada *sip.conf* terletak pada */etc/asterisk/extensions.conf*. File konfigurasi yang dibutuhkan pada *sip.conf* dan *extensions.conf* dapat dilihat pada lampiran.

3.2.2.3 Implementasi Protokol

Di dalam penelitian ini hanya menggunakan satu protokol yaitu protokol RTP dalam implementasinya. Untuk mengaktifkan protocol RTP di server asterisk dengan menggunakan perintah *rtp set debug on* pada *console CLI*.



```

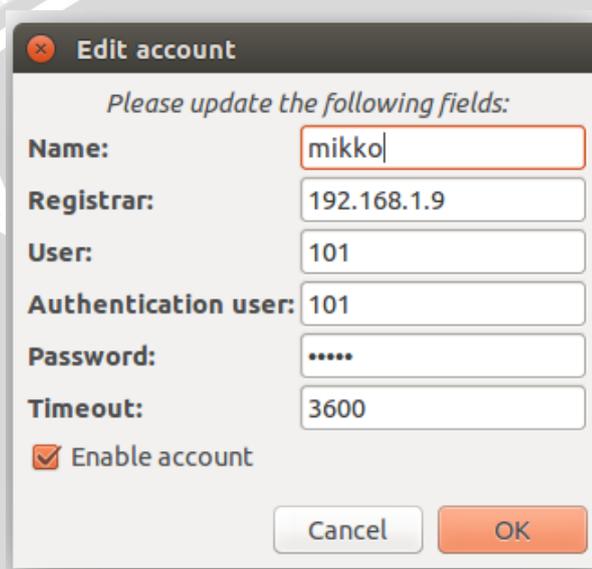
root@research:/home/galuh
root@research:/home/galuh# asterisk -r
Asterisk 13.1-cert2, Copyright (C) 1999 - 2014, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 13.1-cert2 currently running on research (pid = 1314)
research*CLI> rtp set debug on
RTP Debugging Enabled
research*CLI>

```

Gambar 3.12 Mengaktifkan Protokol RTP

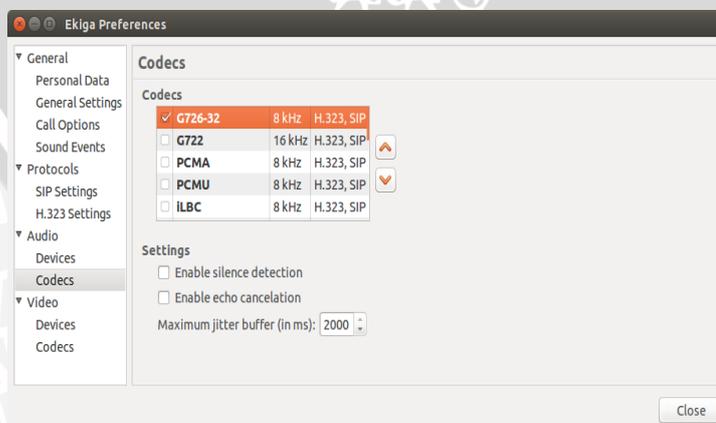
3.2.2.4 Implementasi Softphone

Client yang digunakan pada penelitian ini adalah *Softphone* Ekiga. Ekiga diinstall pada sistem operasi Ubuntu di setiap cilent yang digunakan. Konfigurasi dalam ekiga diisi sesuai dengan informasi yang sudah berada di *server*. Informasi yang dibutuhkan dalam ekiga ialah seperti domain sever, username, dan password *client*. Konfigurasi Informasi tersebut berada pada *account setting* di menu *softphone ekiga*.



Gambar 3.13 Konfigurasi Client 100

Selain mengatur konfigurasi setiap *client*, ekiga juga menyediakan menu untuk mengatur dalam pemilihan *codec* yang akan digunakan. Informasi tersebut berada pada menu *preference*. Terdapat beberapa pilihan *codec* yang tersedia di ekiga, tetapi untuk penelitian hanya *codec* G.711, *codec* G.722 dan *codec* G.726-32.



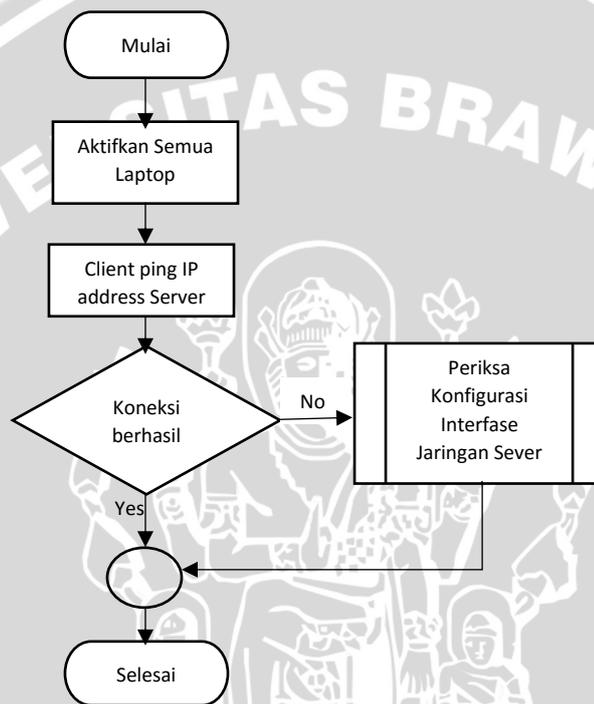
Gambar 3.14 Pemilihan Codec

3.3 Pengambilan Data

Untuk mengetahui bagaimana kehandalan dari suatu jaringan *mesh* serta kinerja dari VoIP yang berjalan di jaringan *mesh* maka pada tahap ini pengambilan data sistem VoIP ada beberapa cara yaitu :

1. Pengujian koneksi jaringan *client-server*

Tahap pengujian ini dilakukan dengan melakukan ping dan traceroute dari *client* ke *server* yang ditunjukkan diagram alir pada Gambar 3.15.



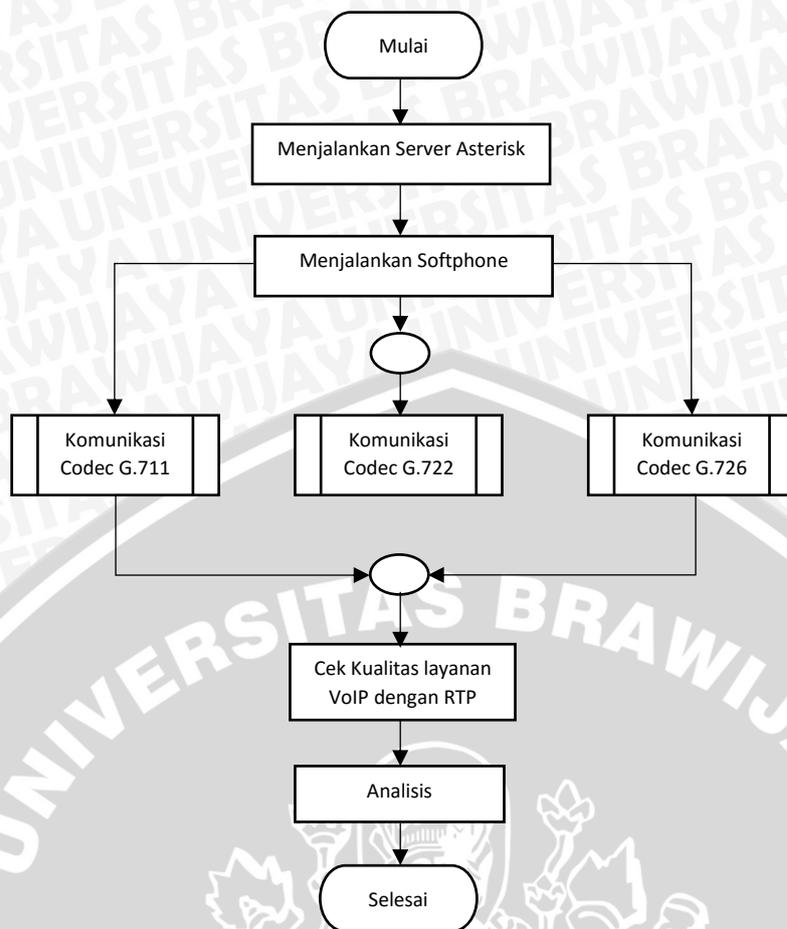
Gambar 3.15 Diagram alir pengujian koneksi server

2. Pengujian koneksi jaringan *client-client*

Tahap pengujian ini dilakukan dengan melakukan ping antar *client*.

3. Pengujian Panggilan VoIP dengan menggunakan Codec G.711, G.722 dan G.726-32.

Pada tahap pengujian ini akan dilakukan dengan melakukan panggilan komunikasi VoIP antar *client* selama 4 menit dengan 10 kali panggilan untuk setiap kondisi. Ketika proses panggilan VoIP berlangsung maka dilakukan pengambilan data menggunakan *wireshark*. Dari hasil *capture* tersebut dapat dianalisis nilai masing – masing parameter dengan menggunakan RTP. Seperti terlihat pada Gambar 3.16.



Gamabr 3.16 Diagram alir pengujian

4. Pengambilan data kualitas VoIP meliputi *Throughput*, *delay*, dan *paket loss*.

Pengambilan data kualitas VoIP akan dilakukan selama pengujian komunikasi VoIP sedang berlangsung. Pengamatan untuk pengambilan data kualitas VoIP menggunakan aplikasi wireshark.

3.4 Hasil dan Pembahasan

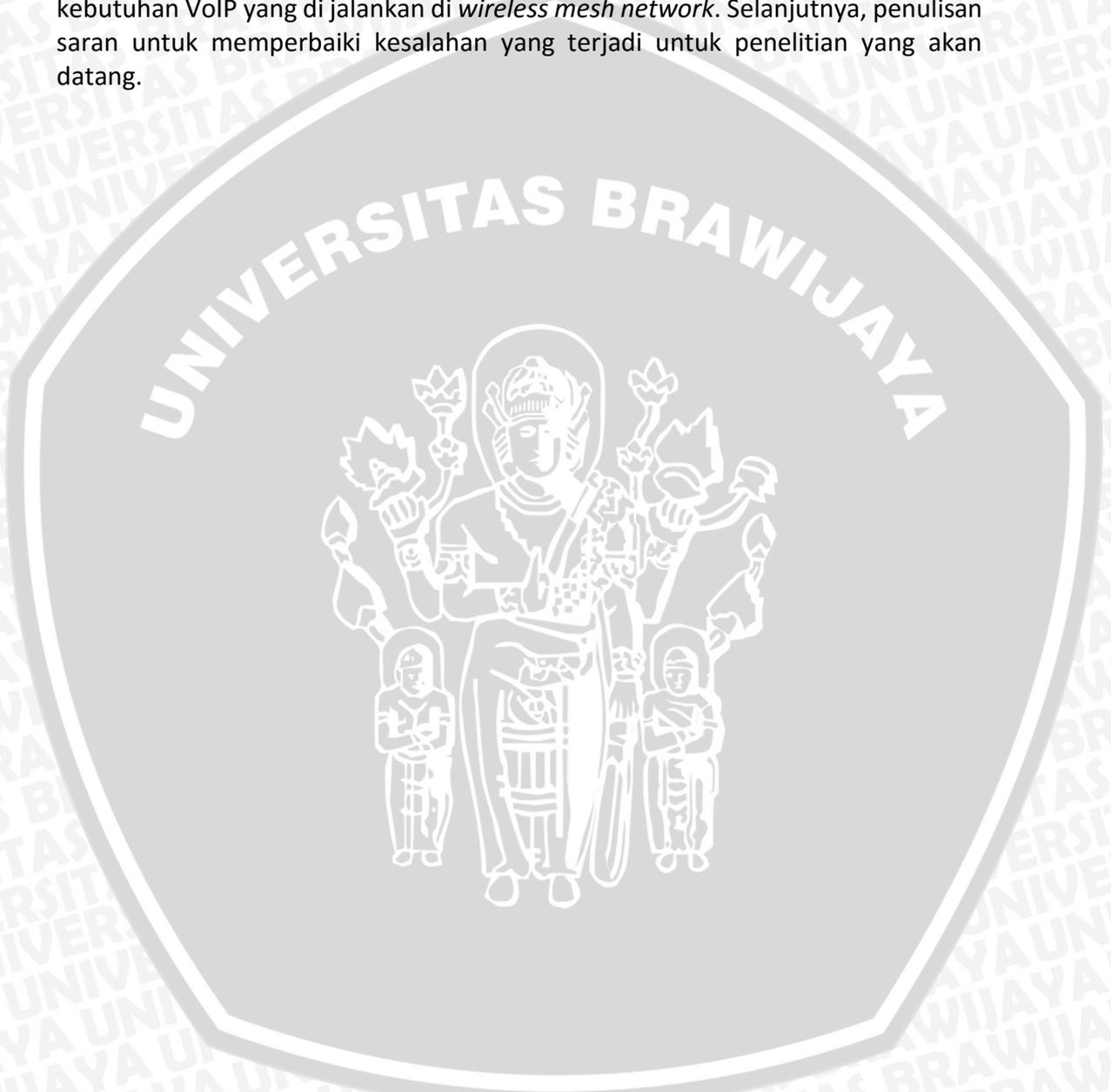
Untuk menjelaskan dan melaporkan hasil dari pengambilan data yang diolah dan disajikan secara terurut dan logis dengan menggunakan tabel dan gambar.

Hasil dari pembahasan ini dapat dijadikan sebagai pengetahuan dan referensi lebih lanjut dalam pemanfaatan WMN untuk layanan aplikasi yang sesuai dengan kinerja jaringan model *client* WMN. Untuk mengukur kinerja dari VoIP apabila dijalankan akan terdapat beberapa data dari parameter hasil pengujian yang diukur, dibahas serta dijadikan acuan dalam pembahasan yaitu:

1. Menghitung data dari hasil pengujian Qos yaitu *Throughput*, *Delay* dan *Paket loos*. Perhitungan yang dilakukan untuk mencari nilai perbandingan codec yang dijalankan VoIP pada jaringan *mesh*.
2. Melihat pengaruh *codec* yang dijalankan VoIP di jaringan *mesh*, untuk melihat manakah *codec* yang stabil untuk komunikasi VoIP di *wireless mesh network*.

3.5 Kesimpulan dan Saran

Kegiatan pengambilan kesimpulan dilakukan setelah mendapatkan data dari hasil pengujian dan analisis yang telah dilakukan. Hasil kesimpulan diperoleh berdasarkan pada data yang diperoleh dari pengujian. Isi dari kesimpulan diharapkan dapat menjadi acuan untuk pemilihan jenis *codec* yang sesuai dengan kebutuhan VoIP yang di jalankan di *wireless mesh network*. Selanjutnya, penulisan saran untuk memperbaiki kesalahan yang terjadi untuk penelitian yang akan datang.



BAB 4 HASIL DAN PEMBAHASAN

4.1 Hasil Pengujian VoIP Skenario 1

Tujuan dari pengujian skenario 1 ini adalah untuk menguji kelancaran komunikasi VoIP pada setiap *codec* yang sudah ditentukan. Pengujian VoIP skenario 1 dengan menempatkan satu client berada diantara Rpi 5 dan Rpi 3 dan client 2 berada diantara Rpi 1 dan Rpi 6 dengan posisi server berada di dalam Gedung C Ruang C1.14. Hasil pengujian yang dilakukan dengan adalah sebagai berikut:

4.1.1 Hasil Pengujian Koneksi *Client* – *Server*

Pada tahap ini dilakukan pengujian koneksi dari *client* ke *server* untuk memastikan bahwa *client* dapat mengakses *server*. Untuk melakukan hal tersebut dapat mengetikkan perintah *ping* pada terminal *client* maupun *server*. Gambar 4.1 koneksi *client* – *server*.

```
mikko@mikko: ~
mikko@mikko:~$ ping 192.168.1.9
PING 192.168.1.9 (192.168.1.9) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_seq=1 ttl=63 time=2.29 ms
64 bytes from 192.168.1.9: icmp_seq=2 ttl=63 time=4.88 ms
64 bytes from 192.168.1.9: icmp_seq=3 ttl=63 time=3.63 ms
64 bytes from 192.168.1.9: icmp_seq=4 ttl=63 time=3.26 ms
64 bytes from 192.168.1.9: icmp_seq=5 ttl=63 time=3.03 ms
64 bytes from 192.168.1.9: icmp_seq=6 ttl=63 time=3.33 ms
^C
--- 192.168.1.9 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 2.298/3.407/4.882/0.778 ms
```

Gambar 4.1 Hasil Uji koneksi *client-server*

Pada Gambar 4.1 menunjukkan bahwa *client* telah terkoneksi dengan *server*. Hal ini menunjukkan bahwa *server* membalas *request client*. Untuk melihat status registrasi *client* pada *server* dapat dengan cara mengetikkan *show peers* pada *console CLI* asterisk. Seperti yang terlihat pada Gambar 4.2.

```
root@research: /home/galuh
research*CLI> sip show peers
Name/username      Host                Dyn Forcerport
Comedia ACL Port      Status Description
101/101            192.168.1.10       D Yes
Yes                5060              OK (14 ms)
102/102            192.168.1.11       D Yes
Yes                5060              OK (31 ms)
2 sip peers [Monitored: 2 online, 0 offline Unmonitored: 0 online, 0 offline]
```

Gambar 4.2 Hasil Konektifitas Server

Dari Gambar 4.2 dapat diketahui bahwa semua *client* sudah terkoneksi dengan *server* dan *server* mampu mengenali semua *client*, dapat dilihat dari status setiap *client* terdeteksi OK.

4.1.2 Hasil Pengujian Koneksi *Client-Client*

Dalam pengujian koneksi antar *client* dilakukan untuk mengetahui dan memastikan bahwa antar *client* dapat terhubung dan dapat melakukan komunikasi VoIP. Hal ini dilakukan dengan cara mengirimkan perintah *ping* antar *client*.

```
mikko@mikko: ~
mikko@mikko:~$ ping 192.168.1.11
PING 192.168.1.11 (192.168.1.11) 56(84) bytes of data.
64 bytes from 192.168.1.11: icmp_seq=9 ttl=63 time=6.73 ms
64 bytes from 192.168.1.11: icmp_seq=10 ttl=63 time=4.41 ms
64 bytes from 192.168.1.11: icmp_seq=11 ttl=63 time=4.38 ms
64 bytes from 192.168.1.11: icmp_seq=12 ttl=63 time=4.85 ms
64 bytes from 192.168.1.11: icmp_seq=13 ttl=63 time=8.88 ms
^C
--- 192.168.1.11 ping statistics ---
13 packets transmitted, 5 received, 61% packet loss, time 12069ms
rtt min/avg/max/mdev = 4.385/5.853/8.881/1.742 ms
```

Gambar 4.3 Uji Koneksi Antar *Client*

4.1.3 Hasil Pengujian Panggilan VoIP

Pengujian VoIP dilakukan dengan antar *client* untuk mengetahui apakah sistem dapat berjalan dengan baik sesuai yang diharapkan. Tahap pengujian ini dilakukan dengan melakukan panggilan voip antar *client* selama 5 menit dengan 10 panggilan dan 1 kali panggilan dengan berbagai bentuk topologi jaringan. Proses pengujian komunikasi dengan menggunakan *codec* yang berbeda yaitu *codec* G.711, G.722 dan G.726-32.

Status keberhasilan panggilan di *server* asterisk dapat dilihat melalui *console* CLI asterisk ketika proses permintaan panggilan dimulai hingga panggilan diakhiri.

```
== Using SIP RTP CoS mark 5
-- Executing [102@default:1] Dial("SIP/101-00000002", "SIP/102,20") in new s
tack
== Using SIP RTP CoS mark 5
-- Called SIP/102
-- SIP/102-00000003 is ringing
-- SIP/102-00000003 answered SIP/101-00000002
-- Channel SIP/101-00000002 joined 'simple_bridge' basic-bridge <255db15c-40
d3-4c6c-9cd7-f31ecd057a99>
-- Channel SIP/102-00000003 joined 'simple_bridge' basic-bridge <255db15c-40
d3-4c6c-9cd7-f31ecd057a99>
> Bridge 255db15c-40d3-4c6c-9cd7-f31ecd057a99: switching from simple_brid
ge technology to native_rtp
> 0x7ffaa80060e0 -- Probation passed - setting RTP source address to 192.
168.1.11:7078
> 0x7ffaa80060e0 -- Probation passed - setting RTP source address to 192.
168.1.11:7078
```

Gambar 4.4 Proses Panggilan VoIP

Dari gambar 4.4 terlihat proses komunikasi *client* 101 memanggil *client* 102 ditunjukkan dengan perintah eksekusi `Dial(SIP/XXX)` dengan maksud proses panggilan tersebut telah menggunakan protocol SIP. Setelah perintah tersebut

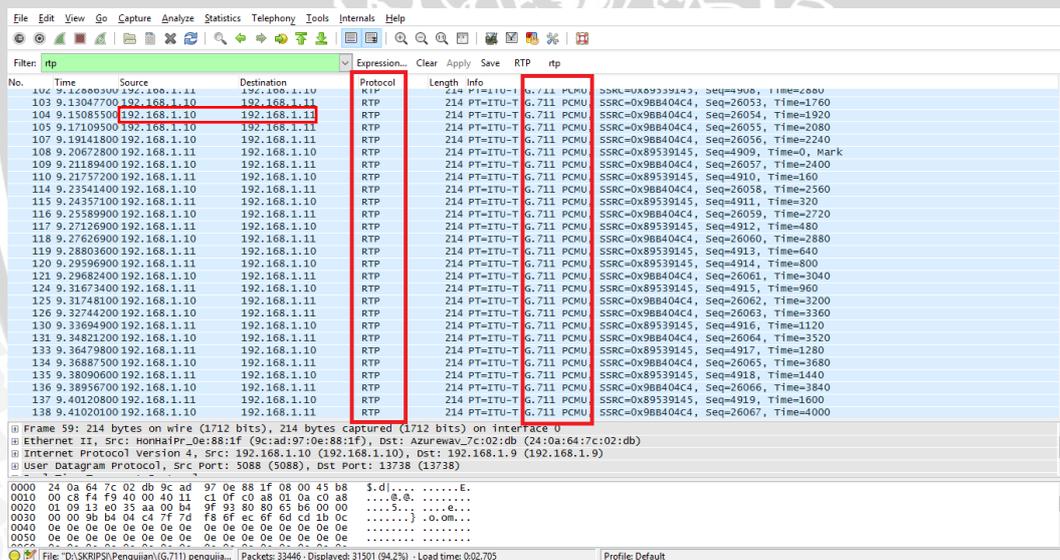
berhasil dieksekusi, maka proses panggilan tersebut ditandai dengan status *called* SIP/XXX pada CLI asterisk. Telepon penerima merespon dengan berdering ditandai dengan SIP/XXX *is ringing*, dan saat telepon diangkat oleh penerima, status di CLI akan berubah dengan menunjukkan SIP/XXX *answered* SIP/XXX. Proses komunikasi tersebut akan seperti itu hingga komunikasi antar *client* dimatikan.

4.1.4 Pengambilan Data Kualitas VoIP

Proses pengambilan data kualitas voip akan dilakukan dengan *codec* yang berbeda. Pengamatan untuk pengambilan data kualitas menggunakan aplikasi Wireshark, dan dilakukan ketika komunikasi berlangsung. Dari hasil yang didapatkan Wireshark, dapat diketahui nilai dari parameter QoS yang diuji seperti, *throughput*, *delay*, *paket loss*, dan *jitter*.

4.1.4.1 Pengambilan Data Kualitas VoIP dengan Codec G.711

Pengambilan data kualitas menggunakan protocol RTP. Sebelum melakukan panggilan antar *Client*, *codec* antar kedua *client* harus menggunakan *codec* yang sama yaitu menggunakan *codec* G.711. Untuk pengambilan data menggunakan aplikasi wireshark.



Gambar 4.5 Hasil Capturing Aplikasi Wireshark untuk data *codec* G.711

Dari gambar gambar 4.5 menunjukkan bahwa komunikasi voip menggunakan *codec* G.711 dan dapat diketahui berdasarkan hasil *filter* *codec* G.711 berjalan diatas protokol RTP. Pengambilan data untuk kualitas voip diambil dari *client* 101 dengan alamat IP 192.168.1.10 menuju *client* 102 dengan alamat IP 192.168.1.11. Dari data yang diperoleh dapat dianalisis nilai *throughput*, *paket loss*, *delay*, dan *jitter*. Untuk menganalisis nilai *throughput*, menggunakan wireshark terlebih dahulu dilakukan *filtering* paket seperti pada gambar 4.5.

| Traffic | Captured | Displayed | Displayed % | Marked | Marked % |
|-------------------------------|-------------|-------------|-------------|--------|----------|
| Packets | 33446 | 31501 | 94,185% | 0 | 0,000% |
| Between first and last packet | 311,033 sec | 301,199 sec | | | |
| Avg. packets/sec | 107,532 | 104,585 | | | |
| Avg. packet size | 208 bytes | 214 bytes | | | |
| Bytes | 6972559 | 6741214 | 96,682% | 0 | 0,000% |
| Avg. bytes/sec | 22417,451 | 22381,293 | | | |
| Avg. MBit/sec | 0,179 | 0,179 | | | |

Gambar 4.6 Hasil Wireshark Summary RTP codec G.711

Dari gambar 4.6 dapat dicari nilai *throughput* dengan menggunakan persamaan 2.1.

$$\text{throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}}$$

$$\text{throughput} = \frac{6741214}{301,199} = 22383,4833 \frac{\text{byte}}{\text{sec}} = 0,179 \text{ Mbit/sec}$$

Dari hasil perhitungan diperoleh nilai *throughput* sebesar 0,179 Mbit/sec. Untuk mengukur nilai delay dari proses tersebut, dengan cara membagi antara waktu dengan jumlah paket data. Untuk mendapatkan nilai *paket loss* dan *jitter* pilih menu *Telephony – RTP – Show All Streams* seperti yang ditunjukkan pada Gambar 4.7.

| Payload | Packets | Lost | Max Delta (ms) | Max Jitter (ms) | Mean Jitter (ms) | Pb? |
|---------|---------|----------|----------------|-----------------|------------------|-----|
| g711U | 3 | 0 (0,0%) | 27,723000 | 0,485910 | 2,592667 | |
| g711U | 16445 | 0 (0,0%) | 27,904000 | 61202415,693131 | 35701,976511 | X |
| g711U | 15035 | 1 (0,0%) | 61,014000 | 44055480,396513 | 35727,545380 | X |

Forward 192.168.1.10:5088 -> 192.168.1.11:5086, SSRC=0x9BB404C4
Reverse 192.168.1.11:5086 -> 192.168.1.10:5088, SSRC=0x89539145

Gambar 4.7 Wireshark RTP stream

Berdasarkan Gambar 4.7 dapat dilihat nilai paket loss dari 192.168.1.10 menuju 129.168.1.11 sebesar 0,0% dengan jumlah paket loss 0. Sedangkan nilai paket loss dari 192.168.1.11 menuju 192.168.1.10 sebesar 0,0% dengan jumlah paket loss 1.



Dari gambar gambar 4.9 menunjukkan bahwa komunikasi voip berjalan menggunakan *codec* G.722 dengan menambahkan kata RTP pada kolom filter maka dengan begitu dapat diketahui bahwa *codec* G.722 sudah berjalan diatas protokol RTP. Pengambilan data untuk kualitas voip diambil dari *client* 101 dengan alamat IP 192.168.1.10 menuju *client* 102 dengan alamat IP 192.168.1.11. Dari data yang diperoleh dapat dianalisis nilai *throughput*, *paket loss*, *delay*, dan *jitter*.

| Traffic | Captured | Displayed | Displayed % | Marked | Marked % |
|-------------------------------|-------------|-------------|-------------|--------|----------|
| Packets | 32393 | 31460 | 97,120% | 0 | 0,000% |
| Between first and last packet | 307,688 sec | 301,349 sec | | | |
| Avg. packets/sec | 105,279 | 104,397 | | | |
| Avg. packet size | 214 bytes | 214 bytes | | | |
| Bytes | 6934495 | 6732440 | 97,086% | 0 | 0,000% |
| Avg. bytes/sec | 22537,396 | 22341,027 | | | |
| Avg. MBit/sec | 0,180 | 0,179 | | | |

Gambar 4.10 Hasil Wireshark Summary codec G.722

Dari gambar 4.10 dapat dicari nilai *throughput* dengan persamaan 2.1

$$\text{throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}}$$

$$\text{throughput} = \frac{6732440}{301,349} = 22341,7 \frac{\text{byte}}{\text{sec}} = 0,179 \text{ Mbit/sec}$$

Dari hasil perhitungan diperoleh nilai *throughput* sebesar 2234,7 *byte/sec* dan menjadi 0,179 *Mbit/sec*. Untuk mengukur nilai *delay* dari proses tersebut, dengan cara membagi antara waktu dengan jumlah paket data.

| Payload | Packets | Lost | Max Delta (ms) | Max Jitter (ms) | Mean Jitter (ms) | Pb? |
|---------|---------|------------|----------------|-----------------|------------------|-----|
| g722 | 16598 | 0 (0,0%) | 31,186000 | 6,144773 | 1,064837 | X |
| g722 | 14862 | 379 (2,5%) | 7599,901000 | 59474425,015942 | 36562,458544 | X |

Forward: 192.168.1.10:5080 -> 192.168.1.11:5068, SSRC=0x6CC0A26C
Reverse: 192.168.1.11:5068 -> 192.168.1.10:5080, SSRC=0xC2B57A58

Gambar 4.11 Hasil Wireshark RTP Streams Codec G.722

Berdasarkan Gambar 4.11 dapat dilihat nilai *paket loss* dari 192.168.1.10 menuju 129.168.1.11 sebesar 0,0% dengan jumlah *paket loss* 0. Sedangkan nilai *paket loss* dari 192.168.1.11 menuju 192.168.1.10 sebesar 2,5% dengan jumlah *paket loss* 379.



| Packet | Sequence | Delta (ms) | Filtered Jitter (ms) | Skew (ms) | IP BW (kbps) | Marker | Status |
|--------|----------|------------|----------------------|-----------|--------------|--------|---------------------|
| 23 | 28851 | 0,00 | 0,00 | 0,00 | 1,60 | SET | [Ok] |
| 29 | 28852 | 103,37 | 6,46 | -103,37 | 3,20 | SET | Incorrect timestamp |
| 31 | 28853 | 21,41 | 6,14 | -104,78 | 4,80 | | [Ok] |
| 33 | 28854 | 17,98 | 5,89 | -102,75 | 6,40 | | [Ok] |
| 36 | 28855 | 21,78 | 5,63 | -104,53 | 8,00 | | [Ok] |
| 39 | 28856 | 15,40 | 5,57 | -99,93 | 9,60 | | [Ok] |
| 41 | 28857 | 20,36 | 5,24 | -100,29 | 11,20 | | [Ok] |
| 43 | 28858 | 21,03 | 4,98 | -101,33 | 12,80 | | [Ok] |

Max delta = 31,19 ms at packet no. 11263
 Max jitter = 6,14 ms. Mean jitter = 1,06 ms.
 Max skew = -113,59 ms.
 Total RTP packets = 15053 (expected 15053) Lost RTP packets = 0 (0,00%) Sequence errors = 0
 Duration: 301,12 (-2 ms clock drift, corresponding to 8000 Hz (-0,00%))

Gambar 4.12 Hasil Wireshark RTP Stream Analysis Codec G.722

Dari gambar 4.12 dapat dilakukan perhitungan *delay* dengan menggunakan dengan menggunakan persamaan 2.2.

$$delay = \frac{waktu}{jumlah\ paket}$$

$$delay = \frac{301,12}{15053} = 0,02\ sec = 20\ milisecond$$

Dari hasil perhitungan diperoleh nilai *delay* sebesar 20 ms.

4.1.4.3 Pengambilan Data Kualitas VoIP dengan Codec G.726-32

Untuk pengambilan data kualitas menggunakan *codec* G.726-32 mekanisme dan cara pengambilannya sama dengan *codec-codec* sebelumnya akan tetapi menghasilkan data dan hasil yang berbeda. *Client* yang menggunakan *codec* G.726-32 adalah *client* 101 dan *client* 102. Gambar 4.13 merupakan hasil *capturing* data menggunakan Wireshark.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|--------------|--------------|----------|--------|---|
| 40 | 6.58356200 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64013, Time=180 |
| 41 | 6.60928500 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64014, Time=160 |
| 42 | 6.62816000 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64015, Time=320 |
| 43 | 6.65168900 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64016, Time=480 |
| 44 | 6.66964500 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64017, Time=640 |
| 45 | 6.69109200 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64018, Time=800 |
| 46 | 6.71090200 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64019, Time=960 |
| 47 | 6.73502800 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64020, Time=1120 |
| 48 | 6.73866900 | 192.168.1.10 | 192.168.1.11 | RTP | 134 | PT=726-32, SSRC=0x5EA5E78C, Seq=21992, Time=0, Mark |
| 49 | 6.74524200 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64021, Time=1280 |
| 50 | 6.76580900 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64022, Time=1440 |
| 51 | 6.79149400 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64023, Time=1600 |
| 52 | 6.81083200 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64024, Time=1760 |
| 53 | 6.83146100 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64025, Time=1920 |
| 54 | 6.85286300 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64026, Time=2080 |
| 55 | 6.87362400 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64027, Time=2240 |
| 56 | 6.88252800 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64028, Time=2400 |
| 57 | 6.90732300 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64029, Time=2560 |
| 58 | 6.92727400 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64030, Time=2720 |
| 59 | 6.94738900 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64031, Time=2880 |
| 60 | 6.94786600 | 192.168.1.10 | 192.168.1.11 | RTP | 134 | PT=726-32, SSRC=0x5EA5E78C, Seq=21993, Time=0, Mark |
| 61 | 6.96238900 | 192.168.1.10 | 192.168.1.11 | RTP | 134 | PT=726-32, SSRC=0x5EA5E78C, Seq=21994, Time=160 |
| 62 | 6.96415800 | 192.168.1.10 | 192.168.1.11 | RTP | 134 | PT=726-32, SSRC=0x5EA5E78C, Seq=21994, Time=160 |
| 63 | 6.97281400 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64032, Time=3040 |
| 64 | 6.98788600 | 192.168.1.11 | 192.168.1.10 | RTP | 134 | PT=726-32, SSRC=0x460C417B, Seq=64033, Time=3200 |

Frame 37: 134 bytes on wire (1072 bits), 134 bytes captured (1072 bits) on interface 0
 Ethernet II, Src: azurewaw_7c:02:db (24:0a:64:7c:02:db), Dst: HonHaIPr_0e:88:1f (9c:ad:97:0e:88:1f)
 Internet Protocol Version 4, Src: 192.168.1.11 (192.168.1.11), Dst: 192.168.1.10 (192.168.1.10)
 User Datagram Protocol, Src Port: 5076 (5076), Dst Port: 5076 (5076)

Gambar 4.13 Hasil Capturing codec G.726-32 dengan Wireshark

Dari gambar gambar 4.13 menunjukkan bahwa komunikasi VoIP berjalan menggunakan *codec* G.726-32 dengan menambahkan kata RTP pada kolom filter maka dengan begitu dapat diketahui bahwa *codec* G.726-32 sudah berjalan diatas protokol RTP. Pengambilan data untuk kualitas voip diambil dari *client* 101 dengan alamat IP 192.168.1.10 menuju *client* 102 dengan alamat IP 192.168.1.11. Dari data yang diperoleh dapat dianalisis nilai *throughput*, *paket loss*, *delay*, dan *jitter*.

| Traffic | Captured | Displayed | Displayed % | Marked | Marked % |
|-------------------------------|-------------|-------------|-------------|--------|----------|
| Packets | 42645 | 40548 | 95,083% | 0 | 0,000% |
| Between first and last packet | 341,274 sec | 330,040 sec | | | |
| Avg. packets/sec | 124,958 | 122,858 | | | |
| Avg. packet size | 132 bytes | 134 bytes | | | |
| Bytes | 5623655 | 5433432 | 96,617% | 0 | 0.000% |
| Avg. bytes/sec | 16478,419 | 16462,926 | | | |
| Avg. MBit/sec | 0,132 | 0,132 | | | |

Gambar 4.14 Hasil Wireshark Summary codec G.726-32

Dari gambar 4.14 dapat dicari nilai *throughput* dengan menggunakan persamaan 2.1.

$$\text{throughput} = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}}$$

$$\text{throughput} = \frac{5433432}{330,040} = 16462,9 \frac{\text{byte}}{\text{sec}} = 0,132 \text{ Mbps}$$

Dari hasil perhitungan diperoleh nilai *throughput* sebesar 16462,9 *byte/sec* dan menjadi 0,132 *Mbps*. Untuk mengukur nilai *delay* dari proses tersebut, dengan cara membagi antara waktu dengan jumlah paket data.

| Payload | Packets | Lost | Max Delta (ms) | Max Jitter (ms) | Mean Jitter (ms) | Pb? |
|---------|---------|------------|----------------|-----------------|------------------|-----|
| G726-32 | 24058 | 0 (0,0%) | 28,402000 | 12,599949 | 1,145740 | X |
| G726-32 | 16490 | 870 (5,6%) | 17380,484000 | 61202425,344758 | 24403226,334500 | X |

Forward: 192.168.1.10:5076 -> 192.168.1.11:5076, SSRC=0x5EA5E78C
Reverse: 192.168.1.11:5076 -> 192.168.1.10:5076, SSRC=0x460C417B

Gambar 4.15 Hasil Wireshark RTP Streams Codec G.726-32

Berdasarkan Gambar 4.15 dapat dilihat nilai *paket loss* dari 192.168.1.10 menuju 129.168.1.11 sebesar 0,0% dengan jumlah *paket loss* 0. Sedangkan nilai *paket loss* dari 192.168.1.11 menuju 192.168.1.10 sebesar 5,6% dengan jumlah *paket loss* 870.



| Packet | Sequence | Delta(ms) | Filtered Jitter(ms) | Skew(ms) | IP BW(kbps) | Marker | Status |
|--------|----------|-----------|---------------------|----------|-------------|--------|----------------------------------|
| 48 | 21992 | 0,00 | 0,00 | 0,00 | 0,96 | SET | [Ok] |
| 60 | 21993 | 209,20 | 13,07 | -209,20 | 1,92 | SET | Incorrect timestamp |
| 61 | 21994 | 14,52 | 12,60 | -203,72 | 2,88 | | [Ok] |
| 62 | 21994 | 1,77 | 12,60 | -203,72 | 2,88 | | Suspected duplicate(MAC address) |
| 64 | 21995 | 20,50 | 11,84 | -204,22 | 3,84 | | [Ok] |
| 70 | 21996 | 22,50 | 11,26 | -206,71 | 4,80 | | [Ok] |
| 71 | 21997 | 18,28 | 10,66 | -204,99 | 5,76 | | [Ok] |
| 73 | 21998 | 20,49 | 10,03 | -205,48 | 6,72 | | [Ok] |

Analysing stream from 192.168.1.10 port 5076 to 192.168.1.11 port 5076 SSRC = 0x5EA5E78C

Max delta = 28,40 ms at packet no. 1646
 Max jitter = 12,60 ms. Mean jitter = 1,15 ms.
 Max skew = -210,90 ms.
 Total RTP packets = 16482 (expected 16482) Lost RTP packets = 0 (0,00%) Sequence errors = 0
 Duration: 329,80 s 15 ms clock drift, corresponding to 8000 Hz (-0,00%)

Gambar 4.16 Hasil Wireshark RTP Stream Analysis Codec G.726-32

Dari gambar 4.16 dapat dilakukan perhitungan *delay* dengan menggunakan persamaan 2.2.

$$delay = \frac{waktu}{jumlah\ paket}$$

$$delay = \frac{329,80}{16482} = 0,02\ sec = 20\ milisecond$$

Dari hasil perhitungan diperoleh nilai *delay* sebesar 20 ms.

4.2 Hasil Pengujian VoIP Skenario 2

Topologi skenario ini bertujuan untuk menguji kestabilan komunikasi VoIP jika salah satu *node* diputus dalam waktu tertentu saat komunikasi berjalan. Pada pengujian VoIP skenario ini bentuk dari topologi sebelumnya tidak dirubah akan tetapi posisi dari *server* dan salah satu *client* sudah diubah, dimana *server* diletakkan di antara Rpi 5 dan Rpi 3, sedangkan posisi untuk kedua *client* yang pertama berada diantara Rpi 1 dan Rpi 6, *client* kedua berada di ruang panel yaitu di Rpi 4. Bentuk topologi ini untuk mengetahui ketika komunikasi berlangsung antara *client* 1 dan *client* 2 kemudian 1 *node* dari *wireless* Rpi diputus maka apakah komunikasi tersebut masih dapat berjalan dengan normal atau terjadi kendala. Proses ini diawali dengan melakukan komunikasi VoIP *client* 1 menghubungi *client* 2 yang saat itu transmisi jaringan melewati Rpi 2 menuju ke *server*.

```

root@mikko: /home/mikko
root@mikko:/home/mikko# traceroute 192.168.1.11
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 60 byte packets
 1 * 192.168.1.1 (192.168.1.1) 38.236 ms 59.870 ms
 2 192.168.1.2 (192.168.1.2) 82.731 ms 97.396 ms 127.141 ms
 3 192.168.1.3 (192.168.1.3) 150.185 ms 168.834 ms 198.364 ms
 4 192.168.1.9 (192.168.1.9) 219.162 ms 253.540 ms
 5 192.168.1.3 (192.168.1.3) 259.364 ms 268.646 ms
 6 192.168.1.4 (192.168.1.4) 287.240 ms 290.185 ms 299.664 ms
 7 192.168.1.11 (192.168.1.11) 338.236 ms 368.636 ms 388.536 ms

```

Gambar 4.17 Hasil Traceroute komunikasi dari client 1 ke client 2

Seperti terdapat pada gambar 4.17 yang menunjukkan bahwa jalur komunikasi voip dengan *client* 1 yang melewati IP dari Rpi 2.

Setelah itu Ketika kedua *client* sedang berkomunikasi selama rentang waktu 6 menit dan saat memasuki menit ke 3 *wireless* pada Rpi diputus, seperti yang terlihat pada Gambar 4.18.

```

root@mikko: /home/mikko
root@mikko:/home/mikko# traceroute 192.168.1.11
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 60 byte packets
 1 * 192.168.1.1 (192.168.1.1) 59.870 ms 67.211 ms
 2 192.168.1.4 (192.168.1.4) 82.731 ms 88.595 ms 95.396 ms
 3 192.168.1.3 (192.168.1.3) 127.141 ms 150.834 ms 168.096 ms
 4 192.168.1.9 (192.168.1.9) 181.494 ms 219.122 ms 226.940 ms
 5 192.168.1.5 (192.168.1.5) 255.285 ms 259.664 ms
 6 192.168.1.4 (192.168.1.4) 268.266 ms 269.967 ms
 7 192.168.1.11 (192.168.1.11) 287.240 ms 299.664 ms 338.236 ms

```

Gambar 4.18 Hasil Traceroute dengan Rpi 2 di disconnected

Dari gambar 4.18 menunjukkan dengan salah satu dari *node* Rpi dimatikan, ternyata komunikasi tersebut tidak mengalami kendala dengan kata lain komunikasi masih tetap berjalan dengan normal sampai pada menit ke 6 komunikasi diakhiri.

Berdasarkan hasil tersebut dapat disimpulkan bahwa dengan terputusnya jalur komunikasi VoIP maka *node* lain yang masih dalam satu jaringan akan menangani komunikasi tersebut.

4.3 Hasil Pengujian VoIP Skenario 3

Topologi skenario ini bertujuan untuk mengetahui pengaruh komunikasi antar *node* jika salah satu *node* diputus dengan bergantian ketika komunikasi berlangsung. Bentuk topologi Rpi tidak berubah dengan tetap menggunakan 6 *node* yang saling berkomunikasi. Akan tetapi posisi dari *client* 1, *client* 2 dan *server* yang diubah dengan menempatkan *client* 1 berada di Rpi 5 begitu juga dengan

server dan *client* berada di dekat Rpi 2. Proses tersebut dimulai ketika *client* 1 menghubungi *client* 2 melalui *server* dengan rentang waktu 10 menit tanpa diputus. Pada saat komunikasi tersebut sedang berlangsung maka *wireless* pada Rpi 4 diputuskan pada menit ke 4, dapat diketahui pada saat itu komunikasi yang berlangsung tidak terputus, dikarenakan *routing* mencari rute lainnya yang masih dalam satu jaringan untuk meneruskan komunikasi, seperti yang terlihat pada Gambar 4.19.

```
root@mikko: /home/mikko
root@mikko:/home/mikko# traceroute 192.168.1.11
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 60 byte packets
 1 * 192.168.1.2 (192.168.1.1) 63.834 ms 80.638 ms
 2 192.168.1.4 (192.168.1.4) 97.396 ms 137.211 ms *
 3 192.168.1.5 (192.168.1.3) 150.834 ms 162.221 ms 168.096 ms
 4 192.168.1.9 (192.168.1.9) 178.066 ms 181.494 ms 186.505 ms
 5 192.168.1.11 (192.168.1.11) 299.807 ms 313.708 ms 315.594 ms
```

Gambar 4.19 Hasil Traceroute dengan node Rpi 4 di *disconnected*

maka pada menit ke 5 *wireless* pada Rpi 4 kembali dinyalakan dan tetap komunikasi tersebut berjalan dengan lancar tanpa ada kendala. Setelah itu, beralih ke topologi dengan posisi Rpi tetap serta *client server* juga masih berada di tempat yang sama dengan komunikasi tetap berjalan. Proses yang dilakukan sama seperti yang sebelumnya disaat komunikasi berlangsung maka *wireless* pada Rpi 3 diputus pada menit ke 7 dan ternyata komunikasi masih tetap berjalan lancar tidak ada tanda-tanda bahwa komunikasi tersebut terputus seperti yang terlihat pada Gambar 4.20.

```
root@mikko: /home/mikko
root@mikko:/home/mikko# traceroute 192.168.1.11
traceroute to 192.168.1.11 (192.168.1.11), 30 hops max, 60 byte packets
 1 * 192.168.1.2 (192.168.1.2) 63.834 ms 80.638 ms
 2 192.168.1.3 (192.168.1.3) 97.396 ms 127.141 ms *
 3 192.168.1.5 (192.168.1.5) 157.211 ms 175.095 ms 195.395 ms
 4 * 192.168.1.9 (192.168.1.9) 219.162 ms 250.185 ms
 5 192.168.1.11 (192.168.1.11) 253.540 ms 259.364 ms 268.646 ms
```

Gambar 4.20 Hasil Traceroute dengan node Rpi 3 di *disconnected*

Berdasarkan hasil pengujian dapat disimpulkan bahwa apabila komunikasi sedang berlangsung kemudian salah satu *node* terputus maka *node* yang lain menangani *node* yang terputus tersebut agar jaringan tetap tersambung.

4.4 Hasil Pengujian VoIP Skenario 4

Tujuan dari pengujian ini adalah untuk mengetahui komunikasi dari *client* yang bergerak apakah mengalami gangguan komunikasi atau tidak saat melakukan panggilan. Hasil pengujian yang telah dilakukan pada salah satu *client* bergerak secara konstan bolak-balik mendekati dan menjauhi *server* yaitu client 1 berjalan diantara Rpi 1, Rpi 2 dan Rpi 6, proses komunikasi tersebut dijalankan dengan rentang waktu 5 menit dan dilakukan 6 kali komunikasi menggunakan 2 *codec* yang berbeda yaitu *codec* G.711 dan *codec* G.726-32. Proses yang kedua client tetap berjalan secara konstan dengan melewati antara Rpi 2, Rpi 3 dan Rpi 5 yang komunikasi tersebut sama dengan yang pertama dijalankan selama 5 menit dan dilakukan 6 kali komunikasi menggunakan 2 *codec* seperti yang sudah digunakan sebelumnya.

Tabel 4.1 Hasil Pengujian Skenario 4 Codec G.711

| Pengujian ke | Throughput (Mbps) | PaketLoss (%) | Delay (ms) | Jitter (ms) | Melalui Node Raspberry |
|------------------|--------------------|---------------|--------------------|--------------------|------------------------|
| 1 | 0,191 | 2,72 | 20,01 | 35239,02 | 1 & 2 & 3 |
| 2 | 0,16 | 0,2 | 20 | 35421,18 | 1 & 2 & 3 |
| 3 | 0,216 | 10,19 | 20 | 35570,52 | 1 & 2 & 3 |
| 4 | 0,107 | 0 | 23,48 | 323031,65 | 2 & 3 & 5 |
| 5 | 0,237 | 0 | 20,02 | 35312,19 | 2 & 3 & 5 |
| 6 | 0,242 | 0 | 20,01 | 35830,62 | 2 & 3 & 5 |
| Rata-rata | 0,192166667 | 2,185 | 20,58666667 | 83400,86333 | |

Tabel 4.2 Hasil Pengujian Skenario 4 Codec G.726-32

| Pengujian ke | Throughput (Mbps) | PaketLoss (%) | Delay (ms) | Jitter (ms) | Melalui Node Raspberry |
|------------------|--------------------|---------------|--------------------|-------------|------------------------|
| 1 | 0,113 | 0 | 20,01 | 1,64 | 1 & 2 & 3 |
| 2 | 0,116 | 0 | 20,01 | 1,18 | 1 & 2 & 3 |
| 3 | 0,097 | 0 | 23,01 | 5,82 | 1 & 2 & 3 |
| 4 | 0,114 | 0 | 20,02 | 3,61 | 2 & 3 & 5 |
| 5 | 0,112 | 0 | 20,01 | 2 | 2 & 3 & 5 |
| 6 | 0,13 | 0 | 20,01 | 1,23 | 2 & 3 & 5 |
| Rata-rata | 0,113666667 | 0 | 20,51166667 | 2,58 | |

4.5 Pembahasan

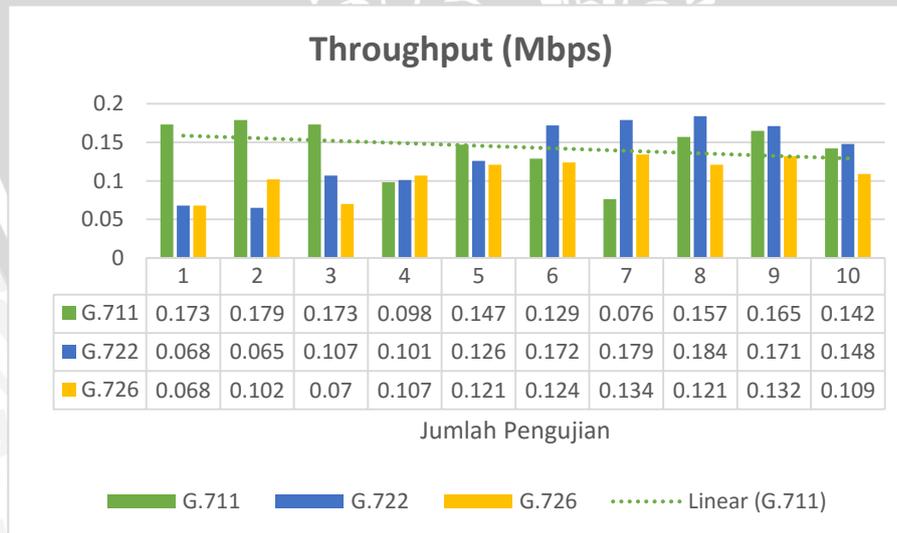
Berdasarkan hasil dari pegujian, dilakukan analisis terhadap nilai *throughput*, *paketloss*, *delay*, dan *jitter* sebagai berikut.

4.5.1 Throughput

Tabel 4.3 berikut menunjukkan nilai *throughput* dalam komunikasi voip menggunakan *wireless mesh network* dengan *codec* G.711, G.722 dan G.726-32.

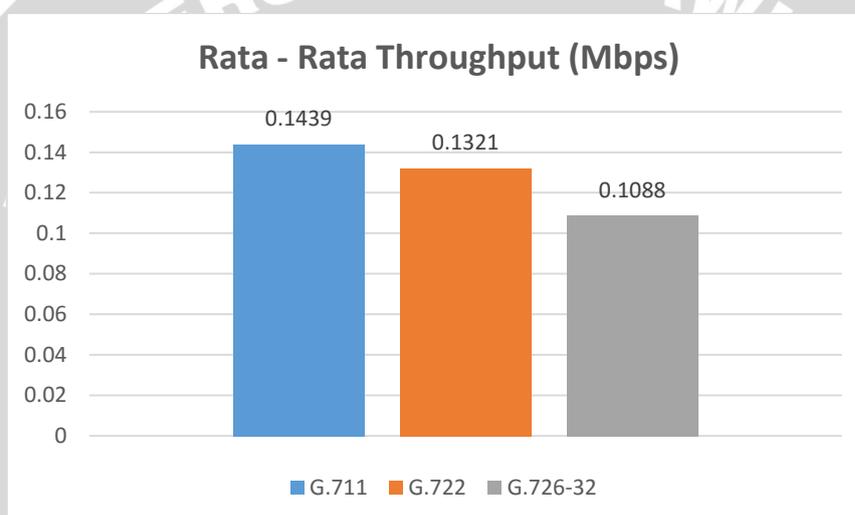
Tabel 4.3 Throughput VoIP pada Wireless Mesh Network

| Throughput (Mbit/sec) | | | |
|-----------------------|---------------|---------------|---------------|
| Pengujian Ke | G.711 | G.722 | G.726-32 |
| 1 | 0,173 | 0,068 | 0,068 |
| 2 | 0,179 | 0,065 | 0,102 |
| 3 | 0,173 | 0,107 | 0,07 |
| 4 | 0,098 | 0,101 | 0,107 |
| 5 | 0,147 | 0,126 | 0,121 |
| 6 | 0,129 | 0,172 | 0,124 |
| 7 | 0,076 | 0,179 | 0,134 |
| 8 | 0,157 | 0,184 | 0,121 |
| 9 | 0,165 | 0,171 | 0,132 |
| 10 | 0,142 | 0,148 | 0,109 |
| Rata - Rata | 0,1439 | 0,1321 | 0,1088 |



Gambar 4.21 Diagram Perbandingan Throughput Keseluruhan Codec

Pada gambar 4.21 dapat dilihat dengan menggunakan codec G.711, nilai *throughput* pada awal komunikasi sangat tinggi dengan nilai mencapai 0,173 Mbps. Apabila dibandingkan dengan codec yang lainnya, akan tetapi memasuki tahap pengujian komunikasi yang ke 4 dan ke 7 mengalami penurunan dengan jumlah paket yang dikirimkan sebesar 0,098 Mbps dan 0,076 Mbps, serta mengalami peningkatan pada pengujian ke 5 dan seterusnya dengan nilai *throughput* sebesar 0,147 Mbps. Dikarenakan pada pengujian ke 4 dan ke 7 komunikasi yang terjadi saat itu terputus-putus. Begitu juga yang terjadi pada codec G.722 yang mengalami peningkatan nilai *throughput* sebesar 0,184 Mbps pada pengujian ke 8 dikarenakan pada awal komunikasi dengan menggunakan codec G.722 suara yang dihasilkan tidak begitu bagus yang menyebabkan nilai *throughput* menjadi kecil sebesar 0,068 Mbps.



Gambar 4.22 Diagram Rata-rata Perbandingan *Throughput* Keseluruhan Codec

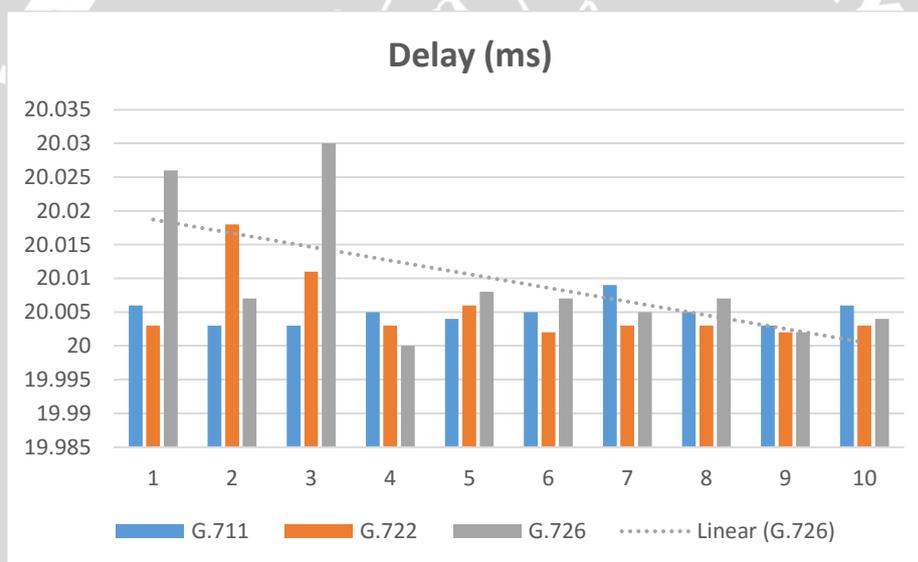
Diagram pada gambar 4.22 merupakan hasil dari *throughput* keseluruhan komunikasi dan semua codec yang digunakan yang dijadikan satu untuk diambil rata-rata *throughput* yang dihasilkan tiap-tiap codec. Dari diagram diatas disimpulkan bahwa dalam pengiriman paket yang dijalankan di *wireless mesh network* menggunakan ketiga codec yaitu, codec G.711, codec G.722 dan codec G.726-32. Diperoleh nilai *throughput* terbesar mencapai 0,1439 Mbps pada codec G.711. Disebabkan pada saat pengukuran tidak ada gangguan yang menyebabkan pengiriman *throughput* terganggu.

4.5.2 Delay

Tabel 4.4 berikut menunjukkan nilai *delay* dalam komunikasi voip menggunakan *wireless mesh network* dengan codec G.711, G.722 dan G.726-32.

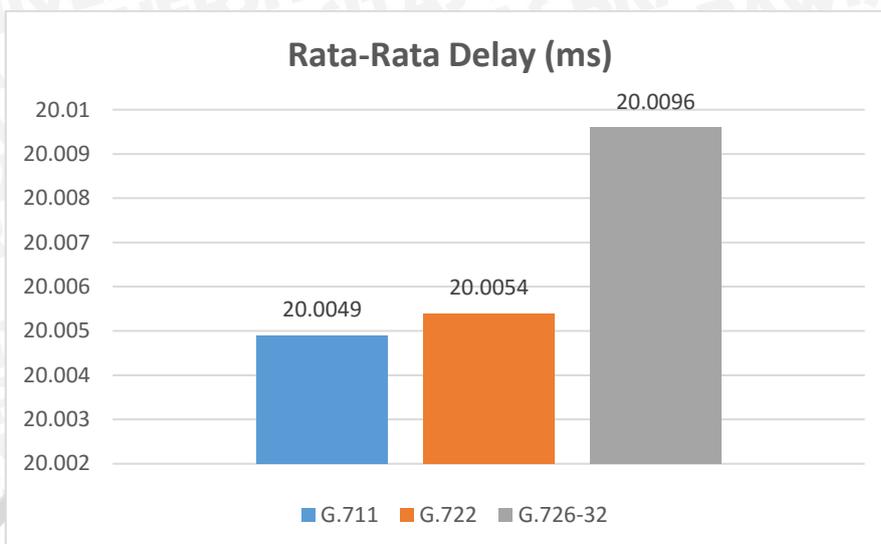
Tabel 4.4 Delay VoIP pada Wireless Mesh Network

| Pengujian Ke | Delay (ms) | | |
|--------------------|----------------|----------------|----------------|
| | G.711 | G.722 | G.726-32 |
| 1 | 20,006 | 20,003 | 20,026 |
| 2 | 20,003 | 20,018 | 20,007 |
| 3 | 20,003 | 20,011 | 20,03 |
| 4 | 20,005 | 20,003 | 20 |
| 5 | 20,004 | 20,006 | 20,008 |
| 6 | 20,005 | 20,002 | 20,007 |
| 7 | 20,009 | 20,003 | 20,005 |
| 8 | 20,005 | 20,003 | 20,007 |
| 9 | 20,003 | 20,002 | 20,002 |
| 10 | 20,006 | 20,003 | 20,004 |
| Rata - Rata | 20,0049 | 20,0054 | 20,0096 |



Gambar 4.23 Diagram Perbandingan Delay Keseluruhan Codec

Pada gambar 4.23 merupakan diagram hasil pengukuran yang dilakukan sebanyak 10 kali komunikasi panggilan dari VoIP Client dengan menggunakan codec G.711, G.722 dan G.726-32. Pada codec G.711 menunjukkan nilai delay rata-rata yang kecil dikarenakan pada saat komunikasi berlangsung komunikasi yang dihasilkan bagus dan suara yang dihasilkan jernih, sedangkan VoIP client dengan codec G.726-32 memiliki nilai rata-rata yang tinggi, keadaan itu bisa dilihat pada pengukuran yang ada pada 1 dan 3 dengan nilai delay mencapai 20,026 ms dan 20,03 ms.



Gambar 4.24 Diagram Rata-rata Perbandingan Delay Keseluruhan Codec

Dari gambar 4.24 dapat dilihat rata-rata *delay* dari ketiga *codec* menunjukkan bahwa VoIP *client* dengan menggunakan *codec* G.711 memiliki *delay* paling kecil yaitu 20,0049 ms. Sedangkan VoIP *client* dengan menggunakan *codec* G.726-32 memiliki *delay* sebesar 20,0096 ms. *Delay* pada *codec* G.726-32 lebih besar dibandingkan *codec* G.711 karena suara yang dilewatkan terputus-putus atau terjadi gangguan komunikasi.

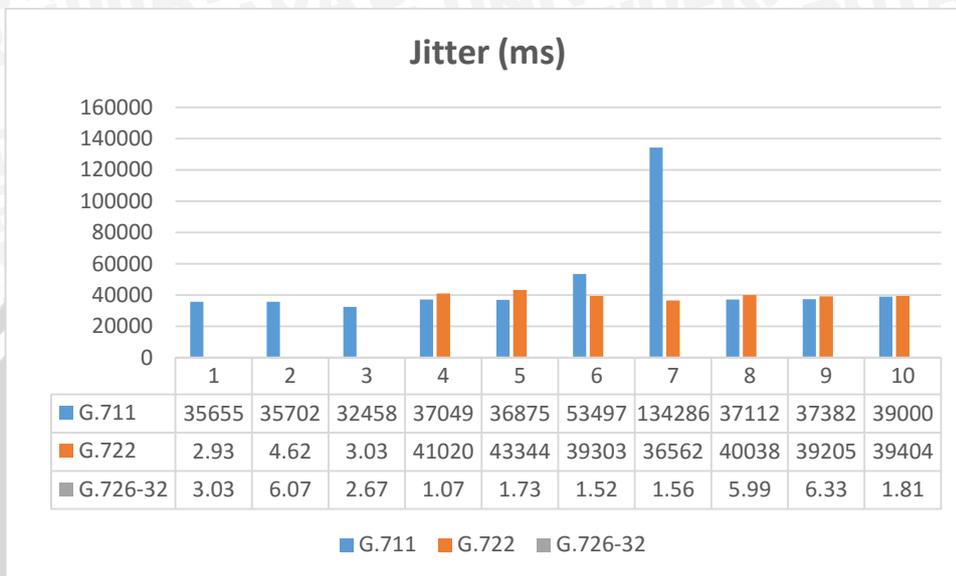
4.5.3 Jitter

Tabel 4.5 menunjukkan nilai *delay* dalam komunikasi voip menggunakan *wireless mesh network* dengan *codec* G.711, G.722 dan G.726-32.

Tabel 4.5 Jitter VoIP pada Wireless Mesh Network

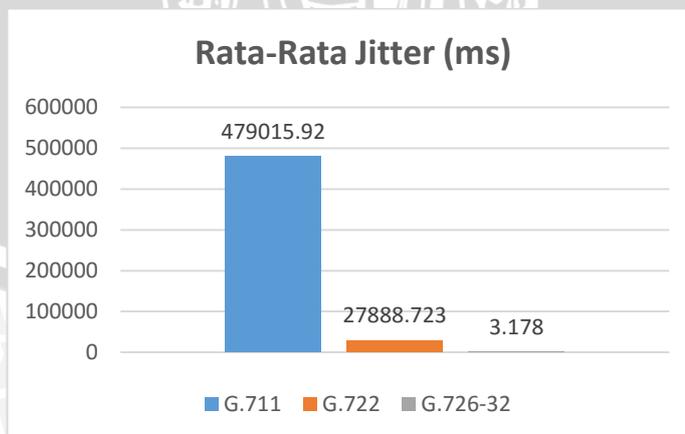
| Jitter (ms) | | | |
|--------------------|------------------|------------------|--------------|
| Pengujian ke 1 | G.711 | G.722 | G.726-32 |
| 1 | 35654,57 | 2,93 | 3,03 |
| 2 | 35701,98 | 4,62 | 6,07 |
| 3 | 32458,02 | 3,03 | 2,67 |
| 4 | 37049,23 | 41019,92 | 1,07 |
| 5 | 36874,64 | 43344,49 | 1,73 |
| 6 | 53497,09 | 39302,98 | 1,52 |
| 7 | 134285,91 | 36562,46 | 1,56 |
| 8 | 37112,36 | 40037,99 | 5,99 |
| 9 | 37382,22 | 39204,71 | 6,33 |
| 10 | 38999,9 | 39404,1 | 1,81 |
| Rata - Rata | 479015,92 | 27888,723 | 3,178 |

Dari hasil tabel 4.5 dapat diketahui bahwa implementasi VoIP menggunakan protocol RTP pada komunikasi codec G.711, G.722 dan G.726-32 didapatkan perubahan nilai-nilai *jitter* yang sangat signifikan yaitu pada tiap –tiap *codec* disetiap komunikasi.



Gambar 4.25 Diagram Perbandingan *Jitter* Keseluruhan *Codec*

Gambar 4.25 merupakan bentuk diagram dari pengukuran yang dilakukan sebanyak 10 kali komunikasi panggilan yang menunjukkan VoIP Client dengan menggunakan *codec* G.711, G.722 dan G.726-32. Pada *codec* G.711 memiliki nilai *jitter* yang sangat besar, sedangkan VoIP client dengan *codec* G.726-32 memiliki nilai yang sangat kecil seperti yang terlihat pada pengujian ke 4 dengan nilai *jitter* 1,07 ms serta keadaan itu bisa dilihat pada pengukuran yang ada pada setiap komunikasi yang terjadi.



Gambar 4.26 Diagram Rata-rata Perbandingan *Jitter* Keseluruhan *Codec*

Dari Gambar 4.26 diatas terlihat bahwa rata-rata *jitter* dari ketiga *codec* menunjukkan bahwa VoIP Client dengan *codec* G.711 memiliki nilai *jitter* paling besar yaitu 479015,92 ms. Sedangkan VoIP client dengan *codec* G.726-32 memiliki *jitter* paling kecil dengan nilai sebesar 3,178 ms. Karena pada saat pengiriman paket yang dikirimkan oleh *codec* G.711 lebih besar daripada paket yang dirimkan oleh G.726-32. Komunikasi saat menggunakan *codec* G.711 lebih jelas dengan suara yang tidak terdapat *noise* sama sekali apabila dibandingkan dengan *codec* G.726-32, dengan kejadian tersebut dapat dilihat bahwa *transfer* data dengan menggunakan *codec* G.711 lebih cepat daripada *codec* G.726-32 yang menyebabkan tingginya nilai *jitter* pada *codec* G.711. Bukan hanya itu nilai tersebut disebabkan karena ketika komunikasi lintasan tempuh paket yang digunakan berbeda-beda dan besarnya paket yang dibawa oleh *codec* G.711.

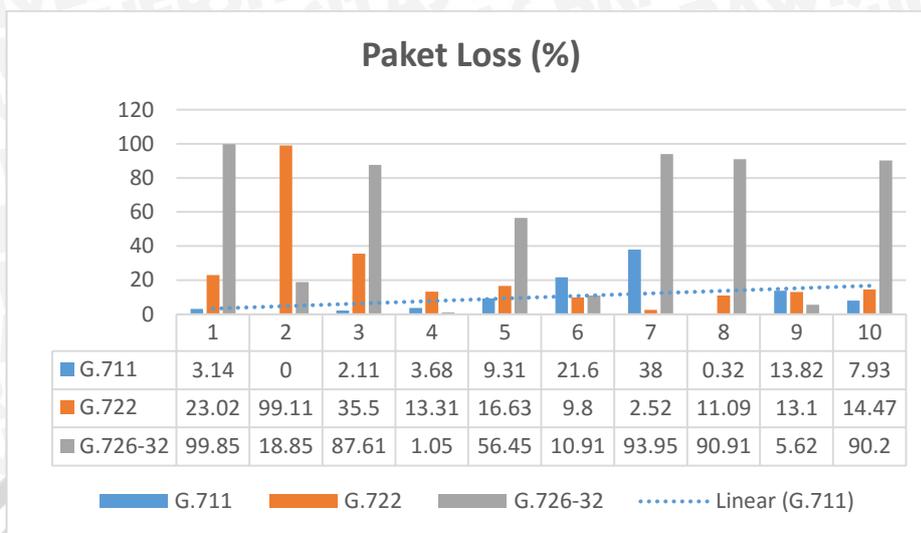
4.5.4 Paket Loss

Tabel 4.6 menunjukkan nilai paket *loss* dalam komunikasi *voip* menggunakan *wireless mesh network* dengan *codec* G.711, G.722 dan G.726-32.

Tabel 4.6 Paket Loss VoIP pada Wireless Mesh Network

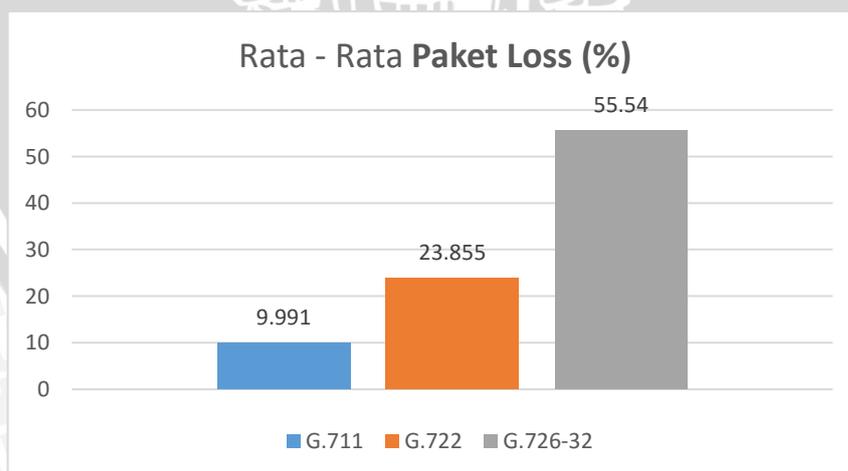
| Paket Loss (%) | | | |
|--------------------|--------------|---------------|--------------|
| Pengujian Ke | G.711 | G.722 | G.726-32 |
| 1 | 3,14 | 23,02 | 99,85 |
| 2 | 0 | 99,11 | 18,85 |
| 3 | 2,11 | 35,5 | 87,61 |
| 4 | 3,68 | 13,31 | 1,05 |
| 5 | 9,31 | 16,63 | 56,45 |
| 6 | 21,6 | 9,8 | 10,91 |
| 7 | 38 | 2,52 | 93,95 |
| 8 | 0,32 | 11,09 | 90,91 |
| 9 | 13,82 | 13,1 | 5,62 |
| 10 | 7,93 | 14,47 | 90,2 |
| Rata - rata | 9,991 | 23,855 | 55,54 |

Dari hasil tabel 4.5 dapat diketahui bahwa implementasi VoIP menggunakan protocol RTP pada komunikasi *codec* G.711, G.722 dan G.726-32 didapatkan nilai-nilai paket *loss* dengan berbeda-beda hampir pada setiap komunikasi.



Gambar 4.27 Diagram Perbandingan *Paket Loss* Keseluruhan *Codec*

Terdapat beberapa faktor yang menyebabkan terjadinya paket *loss* contohnya berupa penurunan *signal* dalam suatu jaringan, paket yang melebihi batas *saturasi* dari jaringan, paket *corrupt* yang menolak *transit* dan kesalahan *hardware* jaringan. Dari hasil pengujian terdapat pada gambar 4.27 dapat diketahui dengan menggunakan *codec* G.726-32, nilai *paket loss* pada awal pengukuran sampai akhir pengukuran sangat tinggi apabila dibandingkan dengan *codec* yang lainnya. Akan tetapi, memasuki tahap pengujian komunikasi yang ke 2 terjadi peningkatan paket *loss* pada *codec* G.722 dengan nilai paket *loss* mencapai 99,11% pada paket yang dikirimkan disebabkan pada saat komunikasi antar dua *client* komunikasi yang terjadi terputus-putus dan menyebabkan paket komunikasi yang dikirimkan menjadi banyak yang hilang.



Gambar 4.28 Diagram Rata-rata Perbandingan *Paket Loss* Keseluruhan *Codec*

Dari gambar 4.28 diatas menunjukkan bahwa rata-rata paket *loss* VoIP Client terkecil terjadi pada *codec* G.711 yaitu 9,991%. Sedangkan, VoIP client dengan

menggunakan *codec* G.726-32 memiliki *paket loss* terbesar dengan nilai sebesar 55,54 %. Karena pada saat komunikasi VoIP paket yang dikirimkan oleh *codec* G.726-32 lebih banyak yang hilang dari pada *codec* G.711. sebab *codec* G.711 mempunyai fitur yang dapat menyembunyikan *packet loss* selama transmisi, sehingga kualitas suara dapat dioptimalkan.



BAB 5 PENUTUP

5.1 Kesimpulan

Kesimpulan yang diperoleh dalam penelitian Analisis Kinerja VoIP (*Voice Over Internet Protocol*) pada *Wireless Mesh Network* antara lain:

1. Kinerja dari VoIP yang dijalankan pada WMN berdasarkan pengujian yang sudah dilakukan pada setiap scenario menghasilkan nilai rata-rata *throughput* sebesar 0,1439 Mbps, dengan begitu makin besar *throughput* yang didapat, makin baik komunikasi data antara pengguna VoIP. Dari hasil pengukuran didapatkan nilai rata-rata *delay* (ms) dari *client* ke *client* adalah sebesar 20,0049 *milisecond* serta hasil pengukuran nilai rata-rata *jitter* (ms) sangat besar dari *client* ke *client* adalah 479015,92 *milisecond*.
2. Kinerja Codec VoIP G.711, G.722 dan G.726-32 pada WMN menunjukkan hasil yang berbeda disebabkan oleh nilai QoS yang berbeda pada setiap *codec*, seperti halnya nilai *jitter* pada *codec* G.711 yang menunjukkan 479015,92 *milisecond*. Nilai tersebut disebabkan karena ketika komunikasi lintasan tempuh paket yang digunakan berbeda-beda serta besarnya paket yang dibawa oleh *codec* G.711.
3. Secara keseluruhan QoS masih sesuai dengan standart yang telah direkomendasikan ITU-T, kecuali nilai *jitter* pada salah satu *codec*, akan tetapi nilai tersebut tidak mempengaruhi kinerja jaringan dan tidak menyebabkan kegagalan komunikasi, karena besarnya nilai *jitter* dapat dikompensasi dengan nilai *delay* yang kecil.

5.2 Saran

Saran untuk penelitian Analisis Kinerja VoIP (*Voice Over Internet Protocol*) pada *Wireless Mesh Network*, antara lain:

1. Komunikasi VoIP di *wireless mesh* perlu dikembangkan ke media komunikasi berupa *video* atau *conference*.
2. Perlu dilakukan pengujian untuk mengetahui besar pengaruh interferensi pada *Wireless Mesh Network*.
3. Pengujian dilakukan pada skala jaringan yang lebih luas.

DAFTAR PUSTAKA

- Abolhasan, M., Hagelstein, B., Chun, J., & Wang, P. (2009). *Real-world performance of current proactive multihop mesh protocols*. Australia: University of Wollongong.
- Akyildiz, I. F., Wang, X., & Wang, W. (2004). *Wireless mesh network: a survey*. Atlanta: 2005.
- Al-Quzwini, M. M., & Sharafali, S. A. (2014). Performance Evaluation of MPLS TE Signal Protocols with Different Audio Codecs. *Journal of Emerging Trends in Computing and Information Sciences*.
- Chhabra, A., & Singh, D. (2011). Performance Evaluation and Delay Modelling of VoIP Traffic over 802.11 Wireless Mesh Network. *International Journal of Computer Applications*, 21.
- Cleevly, D., Lang, J., & Lomas, P. (2016, Januari 20). *Raspberry Pi Foundation*. Diambil kembali dari Raspberry Pi Foundation: <https://www.raspberrypi.org>
- Desantis, M. (2006). *Understanding Voice over Internet Protocol (VoIP)*. 2008.
- Febriadi, M. L., Rochim, A. F., & Widiyanto, E. D. (2013). *Perancangan dan Implementasi Wireless Mesh Node pada Raspberry Pi*. Semarang: Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro.
- Hossain, E., & Leung, K. (2008). *Wireless Mesh Network: Architectures and Protocols*. *Institute of Electrical and Electronics Engineers*, 2008.
- Iskandarsyah. (2003, Juni Senin). *Dasar-Dasar VoIP*. Diambil kembali dari IlmuKomputer.Com: <http://ilmukomputer.org/>
- Ismail, N., Faroqi, A., Kamelia, L., & Mardiaty, R. (2011). Simulasi Wireless Mesh Network (WMN) untuk Mendukung Implementasi Next Generation Network (NGN).
- Mehta, P., & Udany, S. (2001). Voice Over IP :Sounding good on the Internet. *IEEE*, 36.
- Noeranbia, Y. (2009). *Perbandingan Internet Protocol Telephony pada Voice Over Internet protocol (VoIP) Di Indonesia*. Depok: Departemen Teknik Elektro, Universitas Indonesia.
- Pradhan, P. (2013). *Wireless Mesh Network*. India: National Institute of Science & Technology.

- Purbo, O. W., & Raharja, A. (2011). *VoIP Cookbook: Building your own Telecommunication Infrastructure*. Jakarta: 2011.
- S, A. W., H, S. N., & Wahidah, I. (2007). Analisis Quality of Service (QoS) dari Layanan Video Streaming Pada Jaringan IP Multimedia Subsystem (IMS). *Seminar Nasional Sistem dan Informatika 2007*;
- Setiawan, E. B. (2012). Analisa Quality of Service (QoS) Voice Over Internet Protocol (VoIP) dengan Protocol H.323 dan Session Initial Protocol (SIP). *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*.
- Setiawan, R. A. (2015). Analysis Performance VoIP Call Application Android in MANET (Mobile Ad Hoc Network). *Buletin Pos dan Telekomunikasi Vol. 13 No.1 (2015)*, 79-96.
- Simionovich, N. (2008). *AsteriskNOW : a practical guide for deploying and managing an Asterisk-based telephony system using the AsteriskNOW software appliance*. Birmingham: Packt Publishing Ltd.
- Sudiarta, P. K., & Sukadarmika, G. (2009). Penerapan Teknologi VoIP Untuk Mengoptimalkan Penggunaan Jaringan Intranet Kampus Universitas Udayana.
- Suryawan, K. D., Husni, M., & dkk. (2012). *Analisis Layanan Kinerja VoIP Pada Protokol SRTP Dan VPN*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Thabratas, T., & Purbo, W. O. (2001). *Buku Pintar Internet : Teknologi VOIP (Voice Over Internet Protocol)* . Jakarta: Elex Media Komputindo.
- Walker, J. Q. (2002). *Assessing VoIP Call Quality Using the E-model*. NetIQ Corporation.
- Yuniati, Y., Fitriawan, H., & Jaya Patih, D. F. (2014). Analisis Perancangan Server VoIP (Voice Over Internet Protocol) dengan Open Source Asterisk dan VPN (Virtual Private Network) Sebagai Pengaman Jaringan Antar Client. *Jurnal Sains, Teknologi dan Industri, Vol. 12, No. 1,, 112 – 121*.
- Zhang, Y., Luo , J., & Hu, H. (2007). *Wireless Mesh Networking: Architectures, Protocols and Standards* . Boca Raton, New York: Auerbach .
- Zulham, Rosmansyah, Y., & Ismail, N. (2008). Pengembangan Wireless Mesh Network Berbasis Standar IEEE 802.11. *Konferensi dan Temu Nasional Teknologi Informasi dan Komunikasi untuk Indonesia*. Jakarta: Kementerian Perindustrian Republik Indonesia.

LAMPIRAN

Lampiran 1 Konfigurasi sip.conf dan extensions.conf

- *sip.conf*

| | |
|----|----------------------|
| 1 | [general] |
| 2 | context=default |
| 3 | allowoverlap=no |
| 4 | allowguest=yes |
| 5 | qualify=yes |
| 6 | canreinvite=yes |
| 7 | allowanonymous=yes |
| 8 | bindport=5060 |
| 9 | bindaddr=0.0.0.0 |
| 10 | notifyingringing=yes |
| 11 | notifyhold=yes |
| 12 | disallow=all |
| 13 | allow=ulaw |
| 14 | allow=alaw |
| 15 | allow=gsm |
| 16 | allow=g722 |
| 17 | allow=g726 |
| 18 | allow=g729 |
| 19 | |
| 20 | |
| 21 | [101] |
| 22 | context=default |
| 23 | type=friend |
| 24 | username=101 |
| 25 | secret=mesh1 |
| 26 | nat=yes |
| 27 | qualify=yes |
| 28 | host=dynamic |
| 29 | registersip=yes |
| 30 | dtmfmode=rfc2833 |
| 31 | callerid=101 |
| 32 | |
| 33 | |
| 34 | [102] |
| 35 | context=default |
| 36 | type=friend |
| 37 | username=102 |
| 38 | secret=mesh2 |
| 39 | nat=yes |
| 40 | qualify=yes |
| 41 | host=dynamic |
| 42 | registersip=yes |
| 43 | dtmfmode=rfc2833 |
| 44 | callerid=102 |

- *extensions.conf*

| | |
|---|------------------------------|
| 1 | [default] |
| 2 | exten => 100,1,Dial(SIP/101) |
| 3 | exten => 100,1,Hangup() |
| 4 | |
| 5 | exten => 102,1,Dial(SIP/102) |
| 6 | exten => 102,2,Hangup() |

