IMPLEMENTASI DAN ANALISIS QoS (QUALITY OF SERVICE) PADA VoIP (VOICE OVER INTERNET PROTOCOL) BERBASIS LINUX

Sutarti¹, Siswanto², Asep Subandi³

Program Studi Sistem Komputer Fakultas Teknologi Informasi Universitas Serang Raya <u>sutarti86@gmail.com¹</u>, <u>fitraakbar06@gmail.com²</u>, <u>andidavinchi@gmail.com³</u>

Abstrak – Voice over Internet Protocol atau VoIP adalah istilah yang menjelaskan penggunaan jaringan komputer untuk melakukan komunikasi suara. Teknologi VoIP pada saat ini sudah cukup maju sehingga membuatnya sebagai sarana komunikasi suara pilihan yang efisien. Selain itu, faktor penghematan biaya percakapan telpon juga menjadi daya tarik dari VoIP. Pada penelitian ini dilakukan analisis VoIP terhadap Quality of Service (QoS) pada layanan jaringan komunikasi VoIP (Voice over Internet Protocol) menggunakan softphone X-Lite sebagai software untuk melakukan panggilan dan wireshark sebagai Network analyzer untuk mengukur parameter QoS pada VoIP seperti delay, jitter dan Packet loss, jaringan yang digunakan adalah Indihome yang telah terpasang di SMK Karya Mandiri dengan tujuan untuk mengetahui besar kecilnya jaringan ketika panggilan berlangsung. Berdasarkan hasil penelitian dapat disimpulkan bahwa: 1) Kualitas jaringan terbaik ketika pengujian VoIP dengan waktu bervariasi adalah ketika pagi hari atau saat pengguna ISP di SMK Karya Mandiri tidak terlalu banyak, 2) rata-rata delay terbesar (0,019987068 ms) dan terkecil (0,011158369 ms), jitter terbesar (11,96 ms) dan terkecil (0,32 ms) sedangkan untuk packet loss terbesar (1,71%.) dan terkecil (0,00 %).

Kata Kunci: Delay, Jitter, Linux, Packet Loss, VoIP, Wireshark, X-Lite

I. PENDAHULUAN

Perkembangan teknologi informasi sekarang ini sudah sangat maju dan sangat pesat, sehingga aktifitas tidak terbatas oleh jarak dan waktu. Pada saat ini percakapan jarak jauh bisa menggunakan telepon. tetapi panggilan telepon di Indonesia biayanya masih terhitung mahal, apalagi jika terpisah jarak yang cukup jauh maka semakin mahal pula biaya yang diperlukan dalam melakukan komunikasi. Apalagi jika penulis akan melakukan panggilan nomor interlokal atau lintas operator biayanyapun tentunya akan semakin mahal. Tetapi dengan berkembangnya internet hal tersebut dapat sedikit diatasi, dengan memanfaatkan Voice Over Internet Protocol (VoIP). VoIP merupakan teknologi vang memiliki kemampuan melakukan percakapan telepon dengan menggunakan jalur komunikasi data pada suatu jaringan (networking) berbasis IP (Internet Protocol) yang dijalankan di atas infrastruktur jaringan packet network. Dengan semakin luasnya jaringan internet di Indonesia VoIP bisa dijadikan mode komunikasi yang murah. Karena VoIP ini dapat berjalan di jaringan komputer tentunya membutuhkan sebuah server untuk dapat melayani traffic VoIP yang ada suatu jaringan. Terdapat beberapa server VoIP, diantaranya adalah Asterisk, Trixbox dan Axon virtual pbx. Ketiga sofware server ini dapat berjalan pada sistem operasi windows.

Menurut Lestariningati (2013), VoIP disebut juga IP Telephony adalah teknologi yang memungkinkan percakapan suara jarak jauh melalui media internet. Data suara yang diubah menjadi kode digital dan dialirkan menlalui jaringan yang mengirimkan paketpaket data VoIP dapat digunakan pada conference room yang terintegerasi dengan slide, serta video berbasis web, seperti Bigbluebutton. Tentunya agar aplikasi ini dapat berjalan dengan baik dibutuhkan layanan web server serta modul VoIP yang berperan sebagai penyedia layanan. Modul tersebut sudah tersedia di sistem operasi Linux, dimana didalamnya sudah terdapat built-in modul untuk keperluan tersebut.

Berdasarkan data yang penulis terima di SMK Karya Mandiri belum adanya penerapan teknologi *VoIP*. Sistem komunikasi yang sedang berjalan saat ini adalah komunikasi yang masih menggunakan operator seluler. Baik untuk komunikasi antar ruangan di ruang tata usaha, maupun komunikasi antar gedung di lingkungan sekolah. Dengan kondisi seperti itu berakibat pada borosnya penggunaan pulsa pada setiap staf tata usaha dan guru untuk melakukan komunikasi. Tentu saja permasalahan yang terjadi ini menjadi sebuah masalah yang besar bagi sebagian guru. Terlebih lagi biaya komunikasi ke beda operator seluler jauh lebih mahal dibandingkan dengan sesama operator seluler.

Tujuan dari penelitian di SMK Karya Mandiri adalah untuk mengetahui parameter QoS seperti delay, jitter dan packet loss, dengan cara menganalisis jaringan VoIP ketika melakukan panggilan dengan waktu bervariasi untuk mengetahui rata-rata nilai dari parameter QoS yang tadi disebut. Dengan aplikasi wireshark sebagai network analyzer dan X-Lite sebagai aplikasi untuk melakukan panggilan VoIP tersebut.

II. KAJIAN PUSTAKA

Yuniati *et all* (2014), menganalisis dan merancang *server* komunikasi melalui jaringan IP menggunakan *software asterisk* dan menggunakan *software VPN* sebagai pengamannya, merancang sentral komunikasi dengan jaringan *IP* agar dapat melakukan komunikasi antar *client* di seluruh dunia dengan hemat, mengetahui dan membandingkan *QoS* dan *MOS* pada jaringan yaitu pada saat menggunakan dan tidak menggunakan *VPN*.

Lestariningati dkk (2013), melakukan penelitian tentang analisis kualitas layanan VoIP dengan Linux zentyal sebagai penyedia layanan VoIP. Untuk memeriksa ketersediaan layanan VoIP dilakukan proses packet-capture untuk memperoleh informasi dari paket data mengenai layanan yang diterima, dari informasi yang akan dijadikan indikasi adanya layanan VoIP adalah informasi dari protokol RTMP (Real Time Message Protocol). Pengujian dilakukan dengan cara membangun conference dengan jumlah client sebanyak 2 client, 3 client dan seterusnya sampai conference diakhiri. Dari hasil pengujian diperoleh bahwa pembebanan terbesar adalah saat inisialisasi modul serta sesaat sebelum sistem mengalami masalah (down) dan pada pengujian aplikasi VoIP terhadap Linux Zentyal menunjukkan bahwa delay yang tinggi dapat menyebabkan terjadinya drop client.

Wahyuddin (2013), melakukan penelitian tentang implementasi VoIP computer to computer berbasis freeware menggunakan session initiation protocol. Model proses yang digunakan dalam pengembangan sistem VoIP ini adalah model sekuensial linear (classic life cycle), dimana model tersebut mengusulkan pendekatan yang sistematis dan berurut di dalam pengembangannya. Walaupun menggunakan aplikasiaplikasi yang bersifat freeware, secara umum sistem VoIP yang telah dibangun dapat beroperasi dengan baik serta memiliki kualitas suara yang memuaskan. Aplikasi yang digunakan pada penelitian tersebut adalah VOManager, karena VOManager memiliki fitur-fitur yang lengkap untuk memonitor suatu jaringan VoIP, seperti: Call Volume, Voice Quality dan Traffic Monitor. VQManager bekerja dengan cara menaruh sniffer pada sebuah interface proxy server, sehingga VQManager dapat melihat dan merekam paket-paket yang keluar masuk pada interface tersebut.

Voice Over Internet Protocol (VoIP)

Agustin et all (2015), menyatakan bahwa komunikasi suara melalui telepon dapat dilakukan melalui internet tanpa keterbatasan biaya dan jarak. *VoIP* (juga disebut *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.



Gambar 1. VoIP (Voice Over Internet Protocol)

Dalam komunikasi VoIP, pemakai melakukan hubungan telepon melalui terminal yang berupa PC atau android. Dengan bertelepon menggunakan VoIP, banyak keuntungan yang dapat diambil diantaranya adalah dari segi biaya jelas lebih murah dari tarif telepon tradisional, karena jaringan IP bersifat global. Selain itu, biaya maintenance dapat ditekan karena voice dan data network terpisah, sehingga IP Phone dapat ditambah, dipindah dan diubah. Hal ini karena VoIP dapat dipasang di sembarang ethernet dan IP address, tidak seperti telepon konvensional yang harus mempunyai port tersendiri di sentral atau PBX (Private branch exchange).

Cara Kerja *VoIP* adalah mengubah suara analog yang didapatkan dari *speaker* pada komputer menjadi paket data digital, kemudian dari *PC* diteruskan melalui *Hub/ Router/ ADSL Modem* dikirimkan melalui jaringan internet dan akan diterima oleh tempat tujuan melalui media yang sama. Atau bisa juga melalui melalui media telepon diteruskan ke *phone adapter* yang disambungkan ke internet dan bisa diterima oleh telepon tujuan.

Untuk Pengiriman sebuah sinyal ke *remote* destination dapat dilakukan secara digital yaitu sebelum dikirim data yang berupa sinyal analog diubah ke bentuk data digital dengan *ADC* (*Analog to Digital Converter*), kemudian ditransmisikan, dan di penerima dipulihkan kembali menjadi data analog dengan *DAC* (*Digital to Analog Converter*). Begitu juga dengan *VoIP*, digitalisasi *voice* dalam bentuk *packets* data, dikirimkan dan dipulihkan kembali dalam bentuk *voice* di penerima. Format digital lebih mudah dikendalikan, dalam hal ini dapat dikompresi, dan dapat diubah ke format yang lebih baik dan data digital lebih tahan terhadap *noise* daripada analog.

Wireshark

Wireshark merupakan salah satu tools atau aplikasi "Network Analyzer" atau penganalisis jaringan. Penganalisisan kinerja jaringan itu dapat melingkupi berbagai hal, mulai dari proses menangkap paket-paket data atau informasi yang berlalu-lalang dalam jaringan, sampai pada digunakan pula untuk sniffing (memperoleh informasi penting seperti password email). Wireshark sendiri merupakan free tools untuk Network Analyzer yang ada saat ini.

Wireshark mempunyai sekian banyak *feature* termasuk juga *display filter language* yang banyak dan kapabilitas mereka dalam satu aliran pada sesi *TCP*. Paket *sniffer* sendiri diambil kesimpulan sebagai satu

buah *tool* yang berkemampuan menahan dan melakukan pencatatan pada *traffic* data dalam jaringan. Pada saat data dalam jaringan, packet *sniffer* bisa menangkap *protocol data unit (PDU)*, lakukan *decoding* juga analisis pada isi paket. *Wireshark* sebagai satu di antara packet *sniffer* yang diprogram sedemikian rupa agar mengetahui berbagai macam bentuk protokol jaringan.

Wireshark memiliki *feature* yang lengkap, berikut ini adalah fitur-fitur yang dimiliki *wireshark*:

- 1. *Multiplatform* Bisa dipakai untuk beberapa basis sistem operasi (*Unix, Mac, Windows, serta Linux*).
- 2. Bisa lakukan *capture* paket data jaringan secara *real time*.
- 3. Bisa menampilkan informasi protokol jaringan dari paket data secara komplit.
- 4. Paket data bisa disimpan jadi *file* serta nantinya bisa dibuka kembali untuk analisis lebih lanjut.
- 5. *Filtering* paket data jaringan.
- 6. Pencarian paket data dengan persyaratan spesifik.
- 7. Pewarnaan penampilan paket data untuk memudahkan analisis paket data.
- 8. Menampilkan data statistik.
- 9. Untuk lakukan *capture* paket data yang keluar maupun masuk pada jaringan, *wireshark* membutuhkan piranti fisik *NIC* (*Network Interface Card*).

Quality of Service (QoS)

Menurut Ramadhan (2015), *Quality of Service* (*QoS*) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan bandwith, mengatasi delay dan jitter. Berikut ini adalah beberapa contoh parameter QoS seperti delay, jitter, dan packet loss. QoS sangat ditentukan oleh kualitas jaringan yang digunakan. Terdapat beberapa factor yang dapat menurunkan nilai QoS, seperti: Redaman, Distorsi, dan Noise.

Kinerja jaringan komputer dapat bervariasi akibat beberapa masalah, seperti halnya masalah *bandwidth*, *latency* dan *jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Sebagai contoh, komunikasi suara (seperti *VoIP* atau *IP Telephony*) serta video *streaming* dapat membuat pengguna frustrasi ketika paket data aplikasi tersebut dialirkan di atas jaringan dengan *bandwidth* yang tidak cukup, dengan *latency* yang tidak dapat diprediksi, atau *jitter* yang berlebih. Fitur *QoS* ini dapat menjadikan *bandwidth*, *latency*, dan *jitter* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan di dalam jaringan tersebut.

Delay

Delay adalah waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. Oleh karenanya delay dalam suatu jaringan juga merupakan unjuk kerja yang dapat dijadikan acuan dalam menilai kemampuan dan kualitas pentransmisian data. Akibat dari delay, data

yang diterima akan mengalami keterlambatan waktu datang sehingga hal ini menyebabkan menunggu sejenak data tersebut sampai pada tujuan. *Delay* akan sangat dirasakan ketika melakukan transmisi paket data yang bersifat *UDP* atau secara *realtime*. Sebagai contoh ketika menghubungi seseorang dari Surabaya yang ada di tempat sangat jauh jaraknya, di luar negri melalui VoIP misalkan, akan sering dijumpai *delay* suara yang cukup terlambat datang untuk merespon suara dari tempat lain.

Berikut adalah cara menghitung *delay*:

$$Delay = \frac{Total \ delay}{Jumlah \ total \ paket}$$

Jitter

Jitter dapat definisikan sebagai variasi delay antara blok-blok yang berurutan. Besarnya nilai jitter akan sangat dipengaruhi oleh variasi beban traffic dan besarnya tumbukan antar paket (congestion) yang ada dalam jaringan. Semakin besar beban traffic di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya congestion dengan demikian nilai jitter-nya akan semakin besar. Semakin besar nilai jitter akan mengakibatkan nilai QoS akan semakin turun. Untuk mendapatkan nilai QoS jaringan yang baik, nilai jitter harus dijaga seminimum mungkin. Jitter dapat diketahui nilainya dengan mengukur nilai peak-nya, yang nantinya dijadikan patokan dalam menentukan kualitas jaringan.

Perbedaan *delay* dengan *jitter* terletak dari waktu keterlambatan, *jitter* memiliki perbedaan yang tidak menentu terhadap keterlambatan pada tiap waktunya. Hal ini dikarenakan kemampuan alat yang berbedabeda dalam merespon suatu data tiap waktu. Perbedaan ini menyebabkan data ketika melintasi jaringan, jarak antar blok informasi menjadi tidak seragam lagi. Hal inilah yang mungkin berbeda dengan *delay*, yang cenderung memiliki keterlambatan yang konstan pada tiap waktunya.

Berikut adalah cara menghitung jitter:

$$Jitter = \frac{Total \ jitter}{Total \ paket \ yang \ di \ terima}$$

Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinkan, diantaranya yaitu:

- 1. Terjadinya overload traffic di dalam jaringan.
- 2. Tabrakan (congestion) dalam jaringan.
- 3. Error yang terjadi pada media fisik.
- 4. Kegagalan yang terjadi pada sisi penerima antara lain bisa disebabkan karena *overflow* yang terjadi pada *buffer*.

Di dalam implementasi jaringan *IP*, nilai *packet loss* ini diharapkan mempunyai nilai yang minimum. Secara umum biasanya terdapat pengkategorian performansi jaringan berdasarkan nilai *packet loss* yaitu sangat bagus, bagus, jelek, dan sedang.

Kategori Performasi	Packet Loss
Sangat bagus	0
Bagus	3 %
Sedang	15 %
Jelek	25 %

Tabel 1.	Kategori	Standar	Packet	Loss
----------	----------	---------	--------	------

Berikut adalah cara menghitung *packet loss*: *Packet Loss* =

(Jumlah paket yang dikirim–jumlah paket yang hilang) X Jumlah paket yang dikirim

100 %

Softphone

Softphone adalah perangkat lunak yang mensimulasikan aksi telepon dan memungkinkan penulis membuat, menerima dan mengelola panggilan suara melalui internet. *Softphone* biasanya berjalan di komputer, *tablet, PC*, dan *smartphone* yang diperlukan untuk melakukan panggilan *VoIP* dan panggilan video.

Softphones telah berkembang selama bertahuntahun selama pengembangan industri VoIP. Pada masamasa awal VoIP, softphone adalah replikasi telepon tradisional di atas layar. Saat ini, softphone digabungkan sebagai antarmuka dasar untuk aplikasi komunikasi. **Softphones** berbeda berdasarkan fungsionalitas mereka, sesuai tujuan penggunaannya, dengan kecanggihan dan kompleksitas protokol yang digunakan, dan pada fitur yang ditawarkan. Misalnya, Softphone yang dirancang untuk keperluan bisnis cenderung memiliki antarmuka yang besar dan banyak fitur dengan menu dan pilihan kaya. Di sisi lain, aplikasi smartphone dan chat memiliki antarmuka softphone sederhana dan sederhana yang hanya membutuhkan satu atau dua sentuhan jari untuk memulai panggilan. Softphone yang digunakan adalah Softphone X Lite.



Gambar 2. Softphone X-lite

Linux

Menurut Rosmeri et all (2013), *Linux* adalah nama dari sebuah sistem operasi yang berbasis *Unix* yang disebarluaskan ke masyarakat secara gratis dan berada di bawah lisensi *GNU General Public License (GPL)*, yang berarti bahwa Linux didistribusikan berikut dengan *source code*-nya. Ketersediaan akses kode

memungkinkan para pengguna sumber untuk memodifikasi sistem operasi ini yang kemudian diperbolehkan juga untuk digunakan dan didistribusikan kembali secara bebas. Nama "Linux" berasal dari nama pembuatnya, yang diperkenalkan tahun 1991 oleh Linus Torvalds. Sistemnya, peralatan sistem dan pustakanya umumnya berasal dari sistem operasi GNU, yang diumumkan tahun 1983 oleh Richard Stallman. Kontribusi GNU adalah dasar dari munculnya nama alternatif GNU/Linux, yang menjadi perbedaan paling utama antara sistem operasi Linux dengan sistem operasi populer lainnya adalah terletak pada kernel Linux dan komponen-komponen yang menyusunnya yang bila diakses secara bebas dan terbuka.

III. METODE PENELITIAN

Topologi Jaringan Usulan

Topologi jaringan untuk VoIP di SMK Karya Mandiri adalah topologi jaringan yang umum digunakan yaitu topologi star untuk access jaringan nirkabel. Modem atau ISP yang digunakan terhubung langsung ke router, kemudian terhubung ke hub dan switch untuk disebarkan kepada client dengan LAN di lingkungan SMK Karya Mandiri.



Gambar 3. Topologi Jaringan Usulan

Rancangan Aplikasi Instalasi *Trixbox Server*

Trixbox di-install pada komputer server fungsinya adalah untuk membuat VoIP server dan mengkonfigurasi account pengguna telepon agar dapat saling terhubung. Trixbox merupakan distro linux yang berbasis CentOS. Interface nya menggunakan Free PBX (GUI) atau tampilan grafis.



Gambar 4. Instalasi Trixbox Server



Gambar 5. Konfigurasi Network pada Trixbox

Konfigurasi jaringan server menggunakan DHCP IP address yang akan digunakan pada server VoIP, IP address pada server VoIP akan berubah-ubah sesuai dengan jaringan internet yang diterima. Saat ini penulis menggunakan akses internet dari modem wifi IndiHome yang digunakan di sekolah tersebut.



Gambar 6. Tampilan Server VoIP Menggunakan Trixbox

Konfigurasi Voip PBX



Gambar 7. Konfigurasi VoIP PBX

Pada konfigurasi VoIP tahap awal untuk menggunakan VoIP masuk ke dalam administrator dengan memasukkan username dan password (maint, password). Setelah berhasil login halaman administrator akan terlihat dan konfigurasi selanjutnya adalah dengan menambahkan account telepon yang akan digunakan.



Gambar 8. Tampilan Server VoIP Administrator

Add SIP Extension

Penambahan *extension* pada *server VoIP* dilakukan dengan memilih menu PBX, lalu pilih *extension* maka langkah selanjutnya tinggal memilih metode apa yang akan digunakan. Pada penelitian ini penulis memilih metode *Generic SIP Device*.

This device uses sip te	chnology.	
secret	asep123	
dtmfmode	rfc2833	
-		
Language		
Language Code Recording Options		
Record Incoming Record Outgoing	On Demand • On Demand •	
Status Voicemail Password Email Address Email Atdacess Email Attachment Play CID Play Envelope Delete Voicemail VM Options	Disabled • yes * no yes * no yes * no yes * no yes * no	
	192:168.4.100/maint/index.pl This device uses sip te secret dtmfmode Language Language Language Code Record Incoming Record Incoming Record Outgoing Voicemail & Directory Status Vokemail Password Email Address Pager Enail Address Pager Envelope Delete Voicemail VM Octions	

Gambar 9. Tampilan Add Extension

Konfigurasi Wireshark

Pada konfigurasi *Wireshark*, tahap awal untuk meng-*capture* data pilih pada menu *capture* kemudian pilih *option* kemudian klik *interface* yang sedang aktif. Data *capture* yang akan dianalisis yaitu paket RTP. Karena *VoIP* adalah komunikasi yang dilakukan secara *Real Time*, maka pada *wireshark* perlu dilakukan konfigurasi untuk menyaring RTP. Data *capture* komunikasi yang sedang berlangsung telah didapat, kemudian data *capture* yang didapat perlu disaring untuk mendapatkan protokol RTP, karena protokol RTP yang diamati terdapat dalam protokol UDP.



Gambar 10. Capture Wireshark Ketika Panggilan Berlangsung

Instalasi X-lite

Pada tahap penginstalan *X-Lite*, sama seperti instalasi aplikasi pada umunya. *X-Lite* sebagai perangkat lunak yang digunakan penelpon untuk saling berkomunikasi pada komputer atau laptop.

Konfigurasi X-Lite

Setelah aplikasi X-Lite terinstal pada masingmasing komputer penelpon maka diperlukan konfigurasi pada masing-masing X-Lite yang sudah terinstal pada komputer, X-Lite dikonfigurasikan untuk menentukan sebuah identitas atau extension yang sudah didaftarkan dan terintegrasi pada server VoIP. Setelah X-Lite dikonfigurasikan dan berhasil maka akan ada pemberitahuan display atau layar bahwa X-Lite telah ready dan siap untuk digunakan.



Gambar 11. Konfigurasi X-Lite

IV. HASIL DAN PEMBAHASAN

Pengujian VoIP

Metode Pengukuran

Pada tahap ini akan melakukan analisis jaringan VoIP, pengujian dilakukan dengan cara melakukan pengukuran parameter *delay*, *jitter*, dan *packet loss* dengan menggunakan *wireshark* sebagai *network analizer*.

Perhitungan *delay* dilakukan *capture* pada *interface* yang terdapat pada *client1* atau *client2* lalu di-*filter*.

Eile	Edit View Go Capture	<u>A</u> nalyze <u>S</u> tati	stics Telephony <u>W</u> irel	ess <u>T</u> ools <u>H</u> elp
	🔳 🥂 🗿 <u>🎼 🕋 🗙 🖻</u>	9.000	Ŧ 🛓 🚍 🔲 🔍 🔍	. 🔍 🎹
	udp && ip.dst ==192.168.43.97 &	& ip.src ==202.153.	128.61	
No.	Destination	Time	Source	Protocol Length
E .	1 192.168.43.97	0.000000	202.153.128.61	UDP
	2 202.153.128.61	0.003549	192.168.43.97	UDP
	3 192.168.43.97	0.010876	202.153.128.61	UDP
	4 202.153.128.61	0.023191	192.168.43.97	UDP
	5 192.168.43.97	0.039178	202.153.128.61	UDP
	6 202.153.128.61	0.043191	192.168.43.97	UDP
	7 192.168.43.97	0.046865	202.153.128.61	UDP
	8 202.153.128.61	0.063303	192.168.43.97	UDP
	9 192.168.43.97	0.081475	202.153.128.61	UDP
ł	10 192.168.43.97	0.081477	202.153.128.61	RTCP
1	11 202.153.128.61	0.083175	192.168.43.97	UDP
	12 192.168.43.97	0.102878	202.153.128.61	UDP
	13 202.153.128.61	0.103919	192.168.43.97	UDP
	14 192.168.43.97	0.120153	202.153.128.61	UDP
	15 202.153.128.61	0.123027	192.168.43.97	UDP
	16 202.153.128.61	0.143006	192.168.43.97	UDP
	17 102 168 /3 07	A 144452	202 153 128 61	IIDP

Gambar 12. Capture saat Melakukan Filterisasi

Yang dilingkari merah adalah alamat *destination* dan *source* yang akan dianalisis IP dan protokolnya. Perhitungan *delay* diambil dengan mem-*filter* paket yang lewat dari IP sumber ke IP tujuan dengan menggunakan *TCP/UDP*. Ini berguna agar terlihat apa yang di-*filter*.

Begitu pula dengan *jitter*, yang didapatkan dengan rata-rata *jitter* saat panggilan. data tersebut kemudian di-*save* dengan format .csv agar dapat dibuka di *microsoft excel. Packet loss* juga didapatkan yaitu presentase dari paket yang dikirim dengan paket yang diterima.

Delay

Delay adalah waktu tunda antar paket di sisi penerima, satuan yang dipakai adalah milidetik (*ms*). Tujuan analisis *delay* adalah untuk mengetahui seberapa cepat jaringan yang dipakai paket dari pengirim ke penerima. Selain itu juga untuk mengetahui besarnya pengaruh Jaringan yang diguinakan terhadap kualitas VoIP yang dihasilkan.

Pengukuran dilakukan dengan melakukan komunikasi VoIP melalui jaringan ISP IndiHome dengan lama waktu bervariasi, kemudain dilihat dari waktu berangkat suatu paket di *client1* dengan waktu tiba paket di *client2* untuk kemudian dicari selisihnya. Dalam pengukuran *delay*, terlebih dahulu melakukan filterisasi, kemudian mencari *time delta* yang berada di kotak detail setelah itu save file tersebut dengan format .csv agar bisa dibuka di *microsoft office excel* untuk kemudian dihitung jumlah rata-rata *delay* tersebut.

Tme DESTRIGON Source 10.enemode 102.158.43.07 202.153.128.61 30.enemode 102.158.43.07 202.153.128.61 50.enemode 102.158.43.07 202.153.128.61 70.enemode 102.158.43.07 202.153.128.61 90.enemode 102.158.43.07 202.153.128.61 10.enemode 102.158.43.07 202.153.128.61 10.enemode 102.158.43.07 202.153.128.61 10.enemode 102.158.43.07 202.153.128.61 11.enemode 102.158.43.07 202.153.128.61 12.enemode 102.158.43.97 202.153.128.61 12.enemode 102.156.43.97 202.153.128.61 12.enemode 102.156.43.97 202	UDP UDP UDP UDP UDP UDP UDP UDP UDP UDP	Length The delta from pr 4	Prime Frid SR800 6.2380 Len=172 S8800 4.2382 Len=172 S8800 6.2380 Len=172 S8800 6.2892 Len=172 S8800 6.2882 Len=172 S8800 6.2802 Len=172 S8800 6.2802 Len=172 S8800 6.2802 Len=172 S8800 6.2802 Len=172 S8800 6.2902 Len=172
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	UDP UDP UDP UDP UDP UDP UDP UDP UDP UDP	214 0.6663066600 214 0.612376600 214 0.02332600 214 0.07567600 214 0.07567600 214 0.021461060 214 0.6127451600 214 0.6127451600 214 0.6127451600 214 0.6124209000 214 0.024209000 214 0.024316600 214 0.024316600 214 0.024316600 214 0.024316600 214 0.024316600 214 0.024316400 214 0.024316400000000000000000000000000000000000	SSB00 - 6.320 Lum-122 SSB00 - 6.320 Lum-127 SSB00 - 6.320 Lum-127 SSB00 - 6.220 Lum-122 SSB00 - 6.320 LUM-122
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		214 0.6187/600 214 0.62330200 214 0.6756700 214 0.6756700 214 0.63451000 214 0.6127/500 214 0.61127/500 214 0.61127/50000000000000000000000000	58800 + 62302 Len-172 58800 + 62302 Len-172
$ \begin{array}{llllllllllllllllllllllllllllllllllll$		214 0.02302000 214 0.02757000 214 0.034507000 214 0.03450002003 214 0.021401000 214 0.021401000 214 0.021401000 214 0.024200000 214 0.02420000 214 0.02430000 214 0.024316000 214 0.024316000000000000000000000000000000000000	\$5800 + 62302 Len-172 58000 + 62302 Len-172
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		214 0.00750700 214 0.075700 214 0.03000 214 0.02420980 214 0.01225080 214 0.012420980 214 0.012420980 214 0.01243000 214 0.0000100 214 0.0000100 214 0.0000100	50000 + 62302 Len-172 50000 + 62302 Len-172
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		214 0.034610000 142 0.050002003 214 0.021401000 214 0.03725000 214 0.0415230000 214 0.041563800 214 0.000001000 214 0.000001000 214 0.000001000 214 0.000001000 214 0.02310000 214 0.023100000 214 0.0231000000 214 0.023100000 214 0.0231000000 214 0.0231000000 214 0.0231000000 214 0.023100000000000000 214 0.020000000000000000000000000000000000	58000 + 02302 Len-172 Sender Report Source dem 580000 + 02302 Len-172 S88000 + 02302 Len-172
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		142 0.000002000 214 0.021001900 214 0.01275600 214 0.01275600 214 0.012436000 214 0.012436000 214 0.0012001000 214 0.00001000 214 0.00001000	Sender Report Source de. 56000 + 62302 Len-172 58800 + 62302 Len-172
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		214 8.021403000 214 8.017275000 214 6.017275000 214 8.041552000 214 8.041552000 214 8.021236000 214 8.021316000 214 9.02316000	58800 + 02302 Len-172 58800 + 62302 Len-172
14 0.120153 1902.168.43.97 2002.153.128.63 17 0.14453 1902.168.43.07 2002.153.128.63 20 0.1690.128.63 2002.153.128.63 2002.153.128.63 21 0.196451 1902.168.43.07 2002.153.128.63 22 0.196452 1902.166.43.07 2002.153.128.63 24 0.2227 192.166.43.97 202.153.128.61 24 0.2227 192.166.43.97 202.153.128.61 26 0.221.101.43.97 202.153.128.61 29 0.261991 192.166.43.97 202.153.128.61 29 0.279111 192.166.43.97 202.153.128.61 29 0.261991 192.166.43.97 202.153.128.61 29 0.279111 192.166.43.97 202.153.128.61		214 8.617275666 214 8.624209666 214 8.641553866 214 8.012436686 214 8.000001000 214 8.024316898 ak detail 24 8.024316898	S8800 → 62302 Len=172 S8800 → 62302 Len=172
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	UDP UDP UDP UDP UDP UDP UDP	214 0.024299600 214 0.041563000 214 0.012436000 214 0.000001000 214 0.020316000 214 0.02316000	58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172
20 0.185045 102.168.43.07 202.153.128.64 21 0.196452 102.168.43.07 202.153.128.64 22 0.196452 192.160.43.07 202.153.128.64 24 0.2227 192.160.43.07 202.153.128.61 24 0.2227 192.160.43.97 202.153.128.61 29 0.23111 192.106.43.97 202.153.128.61 29 0.25111 192.106.43.97 202.153.128.61 29 0.261911 192.106.43.97 202.153.128.61 29 0.261911 192.106.43.97 202.153.128.61		214 0.041563000 214 0.012436000 214 0.000001000 214 0.024316000 ak detail 24 0.024316000	58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172
21 0.108451 192.168.43.97 202.153.128.61 22 0.196452 192.168.43.97 202.153.128.61 24 0.222768 192.168.43.97 202.153.128.61 26 0.594.57 202.153.128.61 202.153.128.61 26 0.224.61 192.168.43.97 202.153.128.61 27 0.267.9411 192.168.43.97 202.153.128.61 29 0.267.9494 192.168.43.97 202.153.128.61		214 0.012436000 214 0.000001000 214 0.024316000 214 0.024316000	58800 → 62302 Len=172 58800 → 62302 Len=172 58800 → 62302 Len=172
22 0.159452 192.168.43.97 202.153.128.61 24 0.222768 192.168.43.97 202.153.128.61 26 0.23411 192.108.43.97 202.153.128.61 29 0.267996 192.168.43.97 202.153.128.61	UDP UDP UDP Kot	214 0.000001000 214 0.024316000 ak detail 214 0.011343000	58800 + 62302 Len=172 58800 + 62302 Len=172
24 0.222768 192.168.43.97 202.153.128.61 26 0.234111 192.168.43.97 202.153.128.61 29 0.267896 192.168.43.97 202.153.128.61	UDE KOT	ak detail 214 0.024316000	58888 + 62382 Len:172
26 0.254111 192.168.43.97 202.153.128.61 29 0.267090 192.168.43.97 202.153.128.61	UBP KOC	ak detail 2 4 9 911 14 1990	DOUDD - DEPER SEIT-STR
29 0.267090 192.168.43.97 202.153.128.61	LIDE	010TT343000	58800 + 62302 Len=172
	UNP	214 0.032979000	58800 + 62302 Len=172
50 0.278855 192.168.45.97 202.153.128.61	UDP	214 0.011765000	58880 + 62302 Len=172
32 0.290384 192.168.43.97 202.153.128.61	UDP	214 0.011529800	58800 + 62302 Len=172
34 0.310190 192.168.43.97 202.153.128.61	UDP	214 0.0193866880	58800 + 62302 Len=172
Interface Los 0 (LOUIDAVANG) (Comparation of the Comparation of the Co	econds]	Time deita	>
Frame Number: 32 c0 18 85 15 15 19 8c be be 13 29 61 08 00 45 50)a	EP	
00 c8 1d c8 00 00 6f 11 f6 2c ca 99 80 3d c0 a8			
2b 61 #5 b0 f3 5# 00 b4 c6 8c 80 09 66 8f a2 4a +a.	^f.		



Open	Ctrl+O	🖲 🖗 📑 🗐 🗹 🗗	Q. 11			
Open Recent	•					
Merge		Source	Protocol	Length	Time delta from	
Import from Hex Dump		192.168.43.97	UDP	214	0.000000000	
Close	Chrluit	202.153.128.61	UDP	214	0.006202000	
Close	CONTRA	192.168.43.97	UDP	214	0.013806000	
Save	Ctrl+S	202.153.128.61	UDP	214	0.009160000	
Save Ac	Ctrl+Shift+S	192.168.43.97	UDP	214	0.010826000	
30VC (3	Gurranicra	202.153.128.61	UDP	214	0.007886000	
File Set		192.168.43.97	UDP	214	0.012128000	
Export Specified Packets		202.153.128.61	UDP	214	0.011723000	
		192.168.43.97	UDP	214	0.008273000	
		As Phin Text	UDP	214	0.019976000	
Expect Dacket Buter	Christia	AS FIAIT TEAL	UDP	214	0.019963000	
Export Packet Bytes	Cui+n	AS CSV	UDP	214	0.020734000	
Export PDUs to File		As "C" Arrays	UDP	214	0.010097000	
Export SSL Session Keys.			UDP	214	0.000002000	
Export Objects		As PSML XML	UDP	214	0.000001000	
Expert objects	70	As PDML XML	UDP	214	0.002935000	
	Ctrl+P	As ISON	UNP	214	9 995819999	
Print	rines.		(1712 bits	1712 bits) on interface 0		
Print	L	Fre , eu	(areas pres	, on anear		

Gambar 14. Penyimpanan File yang Di-export ke Format CSV

IF	Cut		Calibri	- 111 -	A* A*		≫~ ≕ wa	ap Text	Genera	1	
-	ste	· -							544		1 -
100	Forn	at Painter	B 7 0				THE THE ME	rge & Center •		70 .	.0
	Clipboard			Font	T.m.		Alignment		1	Number	1
	G7964	-	e	fr 0.0150116	96						_
All	A	B	С	D	E	F	G	н	1	23	
1	No.	Time	Destinatic	Source	Protocol	Length	Time delta fron	rInfo			
2	3	0.022356	192.168.43	202.153.128.61	UDP	214	0	51982 > 617	754 Len-	172	
3	4	0.036287	192.168.43	202.153.128.61	UDP	214	0.013931	51982 > 617	54 Len=	172	
4	6	0.041282	192.168.49	202.153.128.61	UDP	214	0.004995	51982 > 617	/54 Len=	172	
5	7	0.045286	192.168.43	202.153.128.61	UDP	214	0.004004	51982 > 617	754 Len=	172	
6	9	0.07729	192.168.45	202.153.128.61	UDP	214	0.032004	51982 > 617	754 Len=	172	
7	11	0.089323	192.168.43	202.153.128.61	UDP	214	0.012033	51982 > 617	754 Len-	172	
8	13	0.116065	192.168.43	202.153.128.61	UDP	214	0.026742	51982 > 617	54 Len=	172	
9	15	0.123936	192.168.43	202.153.128.61	UDP	214	0.007871	51982 > 617	754 Len=	172	
10	17	0.155749	192.168.40	202.153.128.61	UDP	214	0.031813	51982 > 617	754 Len=	172	
11	19	0.165734	192.168.43	202.153.128.61	UDP	214	0.009985	51982 > 617	754 Len=	172	
17	21	0.195821	192.168.43	202.153.128.61	UDP	214	0.030087	51982 > 617	54 Len-	172	
13	23	0.20733	192.168.43	202.153.128.61	UDP	214	0.011509	51982 > 612	/54 Len=	172	
14	26	0.249618	192.168.43	202.153.128.61	UDP	214	0.042288	51982 > 617	754 Len=	172	
15	28	0.262936	192.168.43	202.153.128.61	UDP	214	0.013318	51982 > 617	754 Len=	172	
16	29	0.262937	192.168.43	202.153.128.61	UDP	214	0.000001	51982 > 617	754 Len-	172	
17	31	0.283311	192.168.43	202.153.128.61	UDP	214	0.020374	51982 > 613	54 Len=	172	
18	33	0.315864	192.168.45	202.153.128.61	UDP	214	0.032553	51982 > 61	754 Len=	172	
19	35	0.327645	192.168.43	202.153.128.61	UDP	214	0.011781	51982 > 617	754 Len=	172	
20	38	0.363908	192.168.43	202.153.128.61	UDP	214	0.036263	51982 > 617	754 Len=	172	
21	40	0.381939	192.168.43	202.153.128.61	UDP	214	0.018031	51982 > 617	754 Len-	172	
22	41	0.382364	192.168.43	202.153.128.61	UDP	214	0.000425	51982 > 617	54 Len=	172	
23	43	0.404007	192.168.45	202.153.128.61	UDP	214	0.021643	51982 > 61	754 Len=	172	
24	45	0.436406	192.168.43	202.153.128.61	UDP	214	0.032399	51982 > 617	754 Len=	172	
25	47	0.448947	192.168.45	202.153.128.61	UDP	214	0.012541	51982 > 617	754 Len=	172	
14	E E E PC	1 001									

Gambar 15. File CSV yang Ditampilkan pada Ms Excel

Hasil Pengukuran Delay

Di bawah ini dapat dilihat hasil pengukuran nilai delay dari layanan VoIP yang merupakan rata-rata nilai yang diperoleh ketika pengujian jaringan. **Tabel 2. Rata-rata** Delay

Penguijan	Rata-rata delay (ms)				
ke-	Panggilan dari <i>Client 1</i> ke <i>Client 2</i>	Panggilan dari <i>Client 1</i> ke <i>Client 2</i>			
1	0,013257151	0,011158369			
2	0.01501166	0,015675784			
3	0,019934736	0,019987068			
4	0,012908102	0,015667169			
5	0,01895287	0,014700861			

Dalam perhitungan manualnya sebagai berikut:

 $delay = \frac{1}{\text{jumlah total paket}}$

$$= \frac{119,5078}{7961} = 0.01501166$$

Melihat hasil dari tabel 2, terlihat bahwa *delay* yang didapat dari *client1* maupun *client2* pada pengujian *delay*. Hasil yang didapat dari *client1 delay* yang terbesar adalah 0,019934736 *ms* dan *delay* yang terkecil adalah 0,012908102 *ms*. Sedangkan hasil yang didapat dari *client2 delay* yang terbesar adalah 0,019987068 *ms* dan *delay* yang terkecil adalah 0,01158369 *ms*, Dari pengujian sebanyak 5 kali dengan waktu bervariasi didapat hasil *delay* tertinggi *client1* 0,019934736 *ms* dan *client2* 0,019987068 *ms* sedangkan *delay* terendah *client1* 0,012908102 *ms* dan *client2* 0,011158369 *ms*.

Jitter

Jitter adalah variasi *delay* yang terjadi karena ketidak-stabilan kondisi jaringan sehingga waktu penerimaan paket dan penerima berbeda-beda. Tidak hanya parameter *delay* yang diamati namun juga parameter *Jitter*. Tujuan pengukuran *Jitter* adalah untuk mengetahui besarnya *Jitter* yang didapat dari tiap-tiap jaringan yang digunakan ketika melakukan panggilan.

Pengukuran dilakukan dengan melakukan komunikasi *VoIP* melalui jaringan *ISP IndiHome* di SMK Karya Mandiri dengan waktu bervariasi kemudain dilihat dari waktu memulai suatu paket di *client1* dengan waktu tiba paket di *client2* untuk kemudian dicari selisihnya.

Pengukuran *jitter* dilakukan dengan cara melakukan filterisasi kemudian ke menu *analyze* untuk mengubah protokol dari *UDP* menjadi *RTP* pilih *decode as*, setelah melakukan *decode as* dan protokol telah diubah, berikutnya ke menu *telephony* untuk selanjutnya menganalisis *RTP Stream analysis*. Kemudian *save file* tersebut dengan format .csv agar bisa dibuka di *microsoft office excel* untuk kemudian dihitung jumlah rata-rata *jitter* tersebut.



Gambar 16. Mengubah Protokol dengan Cara Decode as





192.168.43.9	7:58262 ↔	Forward	Reverse	Graph					
202.153.128.	61:519/8	Packet	Sequence	Deta (ms)	litter (ms)	Skew	Bandwalth	Marker	Status
Forward		10	16080	18.94	0.12	0.53	16.00		1
	manufacture of	21	16081	20.66	0.15	0.13	17.60		1
Hay Delta	39.60 ms @ 10977	23	16082	20,84	0.20	0.96	19.20		1
Hay litter	2.60 ms	25	16083	19.93	0.19	0.89	20.80		1
Mean Jitter	0.39 ms	27	16084	19.11	0.23	0.01	22.40		1
Max Skew	-17.12 ms	29	16085	19.95	0.22	0.05	24.00		1
RTP Packet	\$ 5720	31	16086	19.71	0.23	0.34	25.60		1
Expected	5720	32	16087	19.99	0.21	0.35	27.20		1
Lost	0 (0.00 %)	35	16088	20.11	0.20	0.24	28.80		1
Seq Errs	0	37	16089	19.72	0.21	0.52	30.40		1
Start at	0.000000 s @ 1	30	16090	20.69	0.24	-0.17	32.00		1
Duration	114.30 s	41	16091	20.72	0.22		11.40		
Clock Drift	3 ms	43	16092	19.10	0.3	Uns	ynchronized F	orward a	and Reverse Audio
req Drift	8000 Hz (0.00 %)	45	16093	20.12	0.1	11			- Branchard Brandler
		47	16094	19.71	0.3	Ons	ynchronizeu P	orward 3	scream notice
everse		49	16095	19.99	0.3	Uns	ynchronized R	teverse 5	Stream Audio
SSRC	0x77196e53	51	16096	20.06	0.3				
Max Delta	147.23 ms @ 300	53	16097	19.84	0.3	Str	earn Synchron	ized Forv	ward and Reverse Audio
Hax Jitter	28.09 ms	55	16098	19.77	0.1	Str	am Synchron	ized For	vard Stream Audio
Hean Jitter	6.56 ms	56	16099	20.74	0.3				
Max Skew	-132.51 ms	59	16100	20.00	0.1	500	am Synchron	Red Kev	erse Stream Audio
RTP Packet	s 5720	62	16101	19.88	0.3	*14.0	Construction of	Francisco	and therease Andre
Expected	5720	64	16102	20.74	0.3	File	synchronized	Porward	and Reverse Audio
Lost	0 (0.00 %)	66	16103	19,98	0.3	Fle	Synchronized	Forward	Stream Audio
Seq trrs	181	68	16104	19.25	0.1	File	Synchronized	Reverse	Stream Audio
start at	0.005202.8 @ 2	70	16105	19.86	0.3	1.00			
Duration	119.395	72	16106	20.07	0.1				
Free Duitt	0000 Liz (-0.00 04)	74	16107	20.15	0.3				
ried Dunc	0000 mz (*0.00 %s)	76	16108	10.00	0.1	For	waro stream o	SV	
orward to a	reverse 06202 s @ 1	78	16109	20.11	0.:	Rev	erse Stream (SV	
2 streams found						Sav	e Graph		

Gambar 18. Penyimpanan *File* Rata-rata *Jitter* dengan Format *CSV*

Hasil pengukuran *Jitter*

Pada tabel 3 dapat dilihat hasil pengukuran nilai *Jitter* dari layanan *VoIP* yang merupakan rata-rata yang telah dianalisis.

	Rata-rata	jitter (ms)
Pengujian ke-	Panggilan dari <i>Client</i> 1 ke <i>Client</i> 2	Panggilan dari <i>Client</i> 2 ke <i>Client</i> 1
1	0,42 ms	11,56 ms
2	0.39 ms	0,44 ms
3	0,32 ms	11,96 ms
4	8,40 ms	0,54 <i>ms</i>
5	0,42 ms	11,34 <i>ms</i>

Tabel 3. Rata-rata Jitter

Hitungan manualnya adalah sebagai berikut:

 $Jitter = \frac{Total variasi delay}{Total paket yang diterima} = \frac{2522,268}{5978} = 0,422066 ms$

Terlihat bahwa hasil yang didapat dari *client1* maupun *client2* pada pengujian kestabilan, hasil yang didapat dari *client2* cenderung menampilkan hasil yang lebih besar, dibandingkan dengan hasil *jitter* pada *client1*. Dari pengujian sebanyak 5 kali dengan waktu bervariasi didapat hasil *jitter* tertinggi masing-masing, pada *client1* sebesar 8,40 *ms* dan 11,96 *ms* pada *client2*, sedangkan *jitter* terendah pada *client1* adalah 0,32 *ms* dan 0,44 *ms* pada *client2*.

Packet Loss

Packet Loss adalah banyaknya paket yang hilang atau rusak pada saat pengiriman paket dibandingkan dengan jumlah total pengiriman paket yang di terima pada sisi *client*. Satuan yang dipakai adalah persen (%). Pengukuran disini bertujuan untuk mendapatkan nilai perbandingan jumlah paket yang hilang atau rusak terhadap total paket yang berhasil diterima oleh *client*.

Pengukuran ini dikhususkan pada pengukuran paket *error ratio*, selama proses dilakukan, Perekaman *traffic* paket data oleh *software wireshark* di sisi *client* kemudian dilanjutkan dengan proses filterisasi paket, proses tersebut dilakukan dengan pengambilan data sebanyak 5 kali.

Packet Loss diukur dengan cara melakukan filterisasi kemudian ke menu analyze untuk mengubah protokol dari UDP menjadi RTP pilih decode as, Setelah melakukan decode as dan protokol telah diubah, berikutnya ke menu telephony untuk selanjutnya menganalisis RTP Stream analysis. Kemudian save file tersebut dengan format .csv agar bisa dibuka di microsoft office excel untuk kemudian dihitung jumlah tersebut.



e-ISSN: 2597-9922, p-ISSN: 2406-7733



Gambar 20. *RTP Streams* Analisis Mencari *Packet Loss*

192.168.43.9	7:61754 ↔	Forward	Reverse	Graph
202.153.128.	61:51982	Packet	Sequence	Delta (ms)
Forward		1	17517	0.00
SSRC	0x4244442d	2	17518	19.62
Max Delta	79.38 ms @ 6757	5	17519	19.36
Max Jitter	5.23 ms	8	17520	22.66
Mean Jitter	0.42 ms	10	17521	19.66
Max Skew	-56.62 ms	12	17522	20.07
RTP Packet	s 5977	14	17523	17.56
Expected	5977	16	17524	20.05
Lost	0 (0.00 %)	18	17525	20.68
Seq Errs	0	20	17526	20.37
Start at	0.000000 s @ 1	22	17527	19.00
Duration	119.55 s	24	17528	20.00
Clock Drift	-42 ms	25	17529	20.67
Freq Drift	/99/Hz (-0.04 %)	27	17530	20.07

Gambar 21. Pengukuran Packet Loss

Hasil Pengukuran Packet Loss

Di bawah ini dapat dilihat hasil pengukuran nilai *Packet Loss* dari layanan *VoIP* yang merupakan rata-rata yang telah dianalisis.

Tabel 4. Packet Loss					
	Packet Loss				
Pengujian ke-	Panggilan dari <i>Client</i> 1 ke <i>Client</i> 2	Panggilan dari <i>Client</i> 2 ke <i>Client</i> 1			
1	0,00 %	1,11 %			
2	0,00 %	0,00%			
3	0,00 %	1,42 %			
4	0,00 %	0,00 %			
5	0,00 %	1,71 %			

Dapat dilihat pada tabel 4, selama panggilan berlangsung *Packet Loss* yang didapat dari *client1* dan *client2* tertinggi sebesar 1,71%. Dari 5 kali pengujian dengan waktu bervariasi dari *client1* maupun *client2* hanya 3 *packet loss* yang didapat, yang berarti terdapat paket yang hilang ketika melakukan panggilan. Berikut adalah kategori standarisasi packet loss:

Kategori Performasi	Packet Loss
Sangat bagus	0
Bagus	3 %
Sedang	15 %
Jelek	25 %

Tabel 5. Kategori Standar Packet Loss

Menurut standarisasi *packet loss* di maka *packet loss* cukup bagus karena *packet loss* yang paling besar hanya 1,71 %.

Tabel 6 merupakan rangkuman dari pengukuran *delay, jitter* dan *packet loss* dengan waktu panggilan bervariasi.

Rata-rata	Terbesar	Terkecil
Delay	(0,019987068 ms)	(0,011158369 ms)
Jitter	(11,96 ms)	(0,32 <i>ms</i>)
Packet Loss	(1,71%.)	(0,00 %)

Tabel 6. Hasil Analisis QoS

V PENUTUP

Kesimpulan

Dari analisis jaringan VoIP yang telah dilakukan oleh penulis pada SMK Karya Mandiri berbasis Linux, maka dapat diambil kesimpulan bahwa:

- 1. Kualitas jaringan terbaik ketika pengujian *VoIP* dengan waktu bervariasi adalah ketika pagi hari atau saat pengguna *ISP* di SMK Karya Mandiri tidak terlalu banyak.
- 2. Rata-rata *delay*, *jitter* dan *packet loss* dengan waktu panggilan bervariasi dengan *delay* terbesar 0,019987068 dan terkecil 0,011158369, *jitter* terbesar 11,96 dan terkecil 0,32, *packet loss* terbesar 1,71% dan terkecil 0,00%.

Saran

Pengujian yang sudah dilakukan dapat ditingkatkan lagi skala jaringannya yang awalnya dari jaringan Local

Area Network (LAN) dapat ditingkatkan skala yang lebih besar lagi yaitu Wide Area Network (WAN).

DAFTAR PUSTAKA

- Eki, S. dkk. (2014). Analisa dan perancangan VoIP menggunaka teknologi Open Source pada pusat Teknologi informasi dan pangkalan data. Vol. 12 No.(1). 45-60
- Eko, B. dkk. (2012). Analisa quality of services (qos) voice over internet protocol (voip) dengan protokol h.323 dan session initial protocol (sip). Vol. 1 No.(2). 125-135
- Iwan, W. (2009). Implementasi VoIP computer to computer berbasis freeware menggunakan session initial protocol. Vol. 3 No.(1). 15-30
- Lestariningati, dkk (2013). Analisa kualitas layanan aplikasi voip dan implementasi penggunaan *linux zentyal* sebagai penyedia layanan *VoIP*. Vol. 2 No.(1). 29-35
- Prahasta, W. (2001). Konsep-konsep Dasar Sistem Informasi Geografis. Penerbit Informatika, Bandung.
- Prakoso, S. (2005). Jaringan Komputer Linux: Konsep Dasar, Instalasi, Aplikasi, Keamanan, dan Penerapan. Penerbit Andi, Yogyakarta: x + 298 hlm.
- Purbo, W. (2007). VoIP Cikal Bakal "Telkom Rakyat". Jakarta Penerbit Infokomputer.
- Rina, A. (2012). Sistem teknologi VoIP (Voice Over Internet Protocol. Vol. 1 No.(30). 1-15
- Taufiq, M. (2008). Membuat SIP *Extensions* pada *Linux Trixbox* untuk *Server VOIP*. Yogyakarta: Penerbit Andi.
- Wahyuddin, (2013). Implementasi VoIP computer to computer berbasisi freeware menggunakan session initial protocol. Vol. 3 No. (1) 50-59
- Yetti, Y. et all (2014). Analisa perancangan *server* voip (voice internet protocol) dengan opensource asterisk dan vpn (virtual private network) sebagai pengaman jaringan antar client. Vol. 12. No (1) 112-121