

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

sutarti86@gmail.com¹, fitraakbar06@gmail.com², andidavinchi@gmail.com³

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 yang 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 *software 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 melalui jaringan yang mengirimkan paket-

paket data *VoIP* dapat digunakan pada *conference room* yang terintegrasi 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 pengamanannya, 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*.

Lestaringati 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 inialisasi 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 aplikasi-aplikasi 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 *VQManager*, karena *VQManager* 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 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 penransmisian 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*:

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

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 berbeda-beda 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*:

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

Packet Loss

Packet loss didefinisikan sebagai kegagalan transmisi paket data mencapai tujuannya. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinan, 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.

Tabel 1. Kategori Standar Packet Loss

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

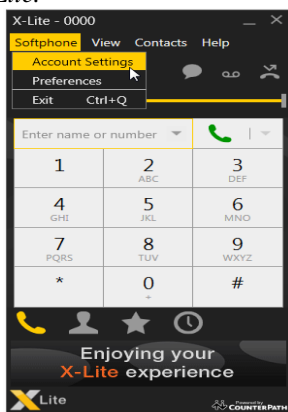
Berikut adalah cara menghitung *packet loss*:

$$Packet\ Loss = \left(\frac{\text{Jumlah paket yang dikirim} - \text{jumlah paket yang hilang}}{\text{Jumlah paket yang dikirim}} \right) \times 100 \%$$

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 bertahun-tahun selama pengembangan industri *VoIP*. Pada masa-masa 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

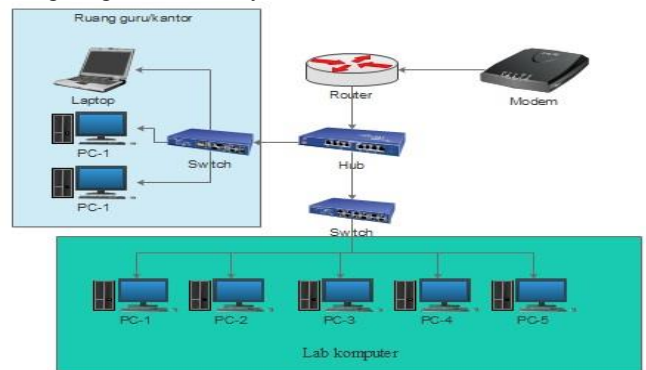
Menurut Rosmeri et al (2013), *Linux* adalah nama dari sebuah sistem operasi yang berbasis *Unix* yang disebarluaskan ke masyarakat secara gratis dan berada di bawah lisensi *GNU General Public License (GPL)*, yang berarti bahwa *Linux* didistribusikan berikut dengan *source code*-nya. Ketersediaan akses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

sumber memungkinkan para pengguna 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 disebar ke *client* dengan *LAN* di lingkungan SMK Karya Mandiri.



Gambar 3. Topologi Jaringan Usulan

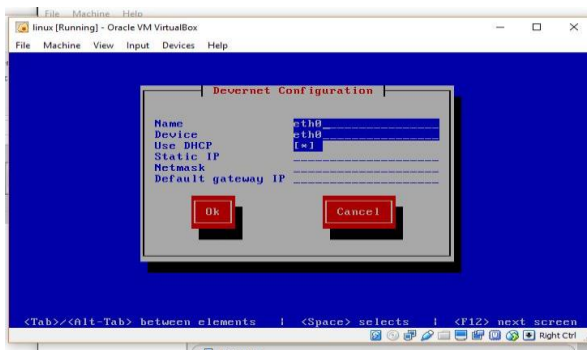
Rancangan Aplikasi

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.

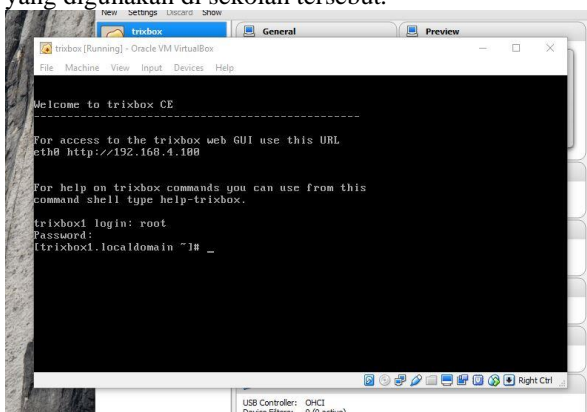


Gambar 4. Instalasi Trixbox Server



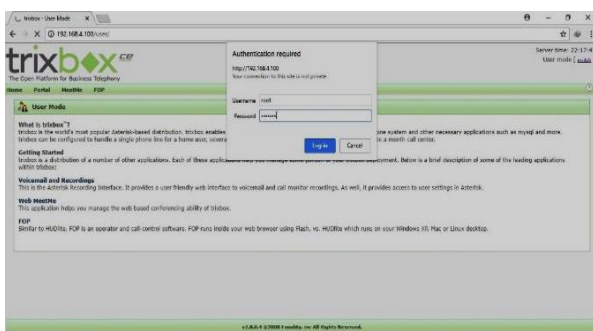
Gambar 5. Konfigurasi Network pada Trixbox

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.



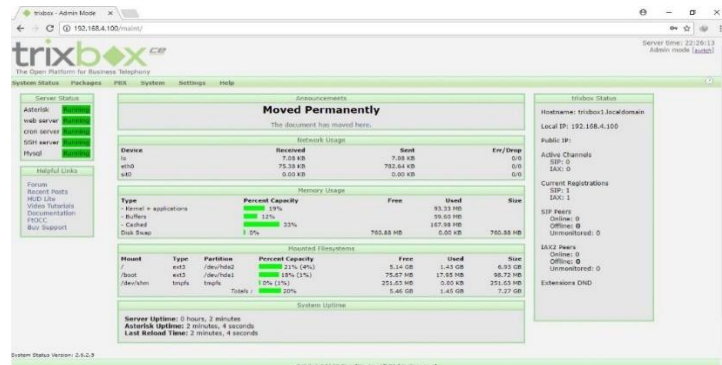
Gambar 6. Tampilan Server VoIP Menggunakan Trixbox

Konfigurasi Voip PBX



Gambar 7. Konfigurasi VoIP PBX

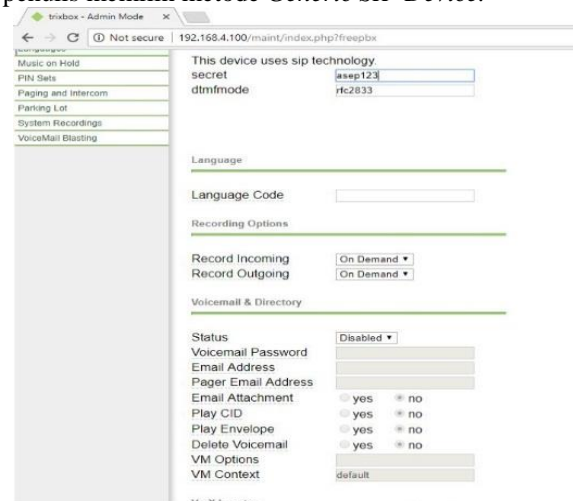
Pada konfigurasi VoIP tahap awal untuk menggunakan VoIP masuk ke dalam administrator dengan memasukkan username dan password (maint, password). Setelah berhasil login halaman administrator akan terlihat dan konfigurasi selanjutnya adalah dengan menambahkan account telepon yang akan digunakan.



Gambar 8. Tampilan Server VoIP Administrator

Add SIP Extension

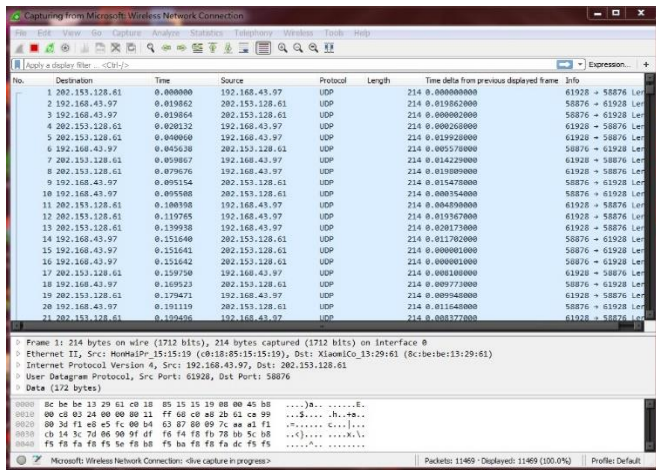
Penambahan extension pada server VoIP dilakukan dengan memilih menu PBX, lalu pilih extension maka langkah selanjutnya tinggal memilih metode apa yang akan digunakan. Pada penelitian ini penulis memilih metode Generic SIP Device.



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.



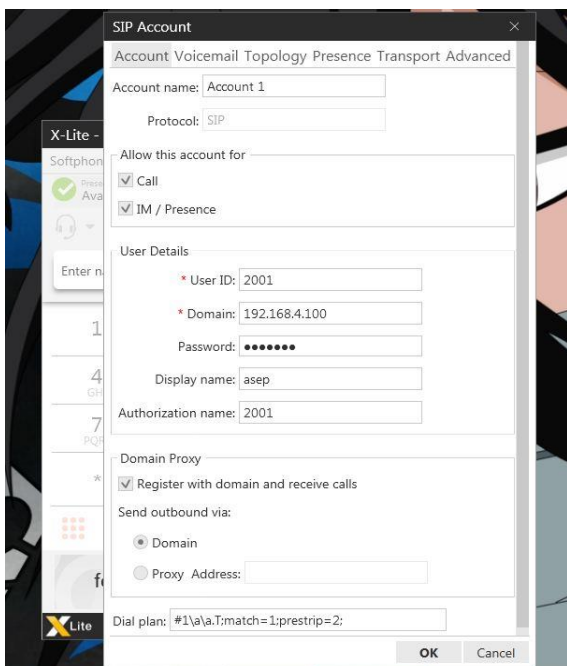
Gambar 10. Capture Wireshark Ketika Panggilan Berlangsung

Instalasi X-lite

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

Konfigurasi X-Lite

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



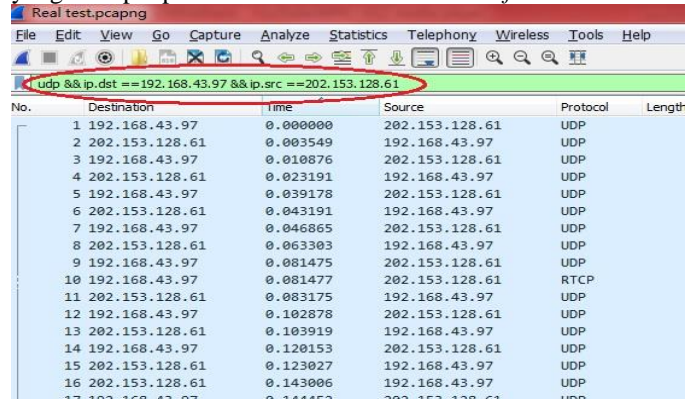
Gambar 11. Konfigurasi X-Lite

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 analyzer.

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



Gambar 12. Capture saat Melakukan Filterisasi

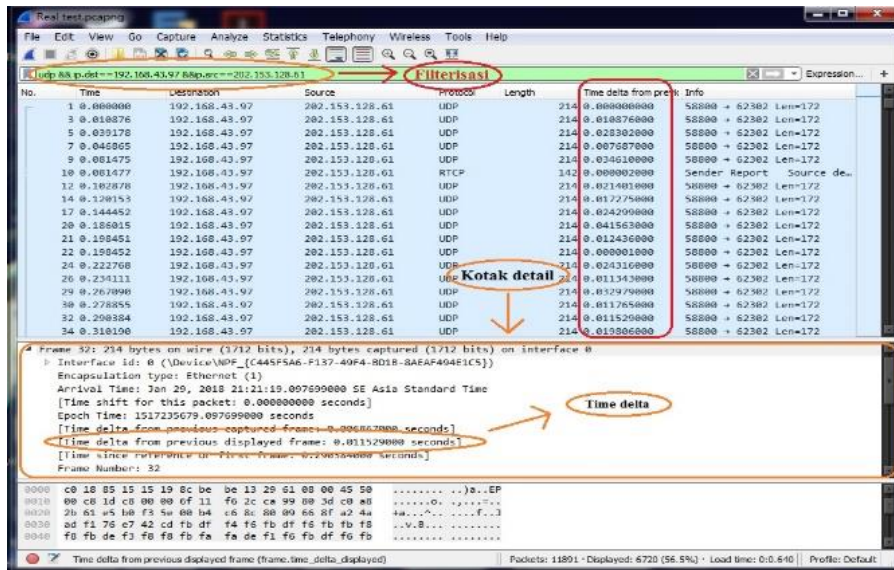
Yang dilingkari merah adalah alamat destination dan source yang akan dianalisis IP dan protokolnya. Perhitungan delay diambil dengan mem-filter paket yang lewat dari IP sumber ke IP tujuan dengan menggunakan TCP/UDP. Ini berguna agar terlihat apa yang di-filter.

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.

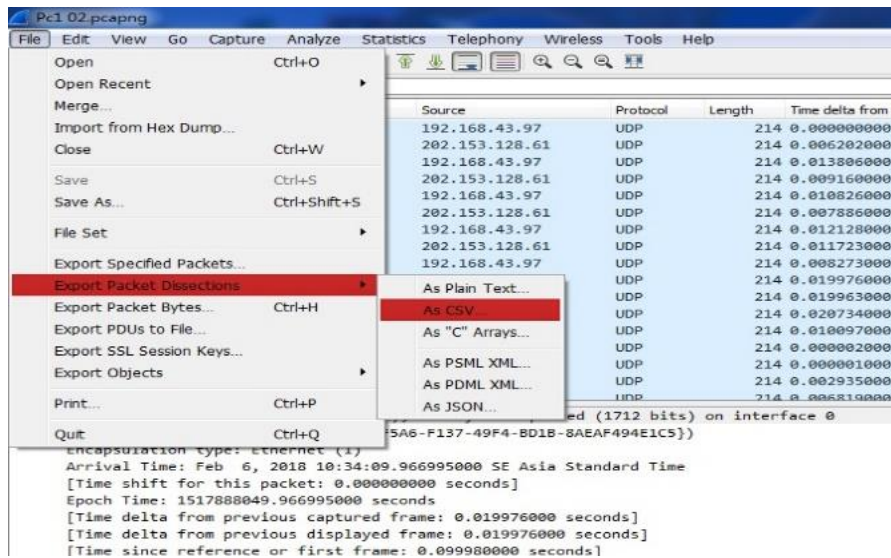
Delay

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 digunakan terhadap kualitas VoIP yang dihasilkan.

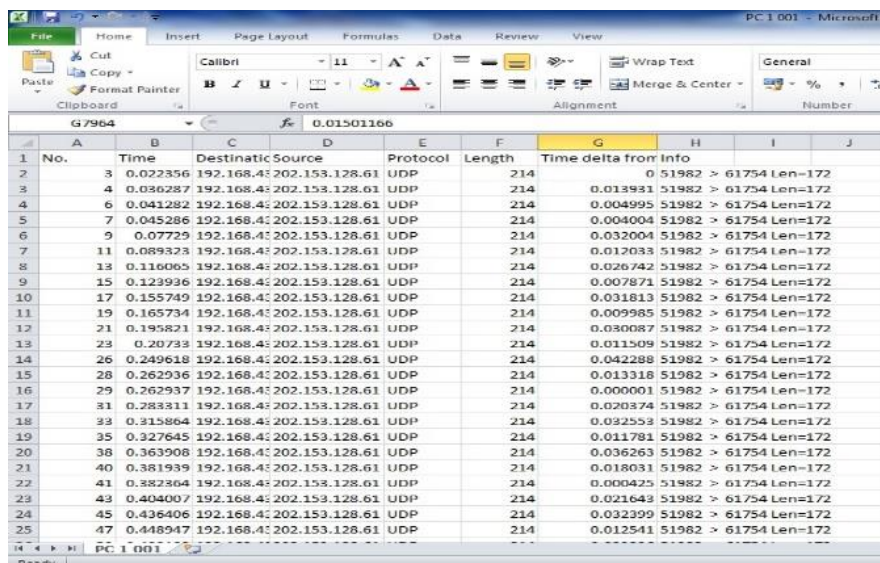
Pengukuran dilakukan dengan melakukan komunikasi VoIP melalui jaringan ISP IndiHome dengan lama waktu bervariasi, kemudian dilihat dari waktu berangkat suatu paket di client1 dengan waktu tiba paket di client2 untuk kemudian dicari selisihnya. Dalam pengukuran delay, terlebih dahulu melakukan filterisasi, 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.



Gambar 13. Mencari Rata-rata Delay



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



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

Pengujian ke-	Rata-rata <i>delay</i> (ms)	
	Panggilan dari <i>Client 1</i> ke <i>Client 2</i>	Panggilan dari <i>Client 1</i> ke <i>Client 1</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:

$$delay = \frac{\text{total delay}}{\text{jumlah total paket}} = \frac{119,5078}{7961} = 0.01501166$$

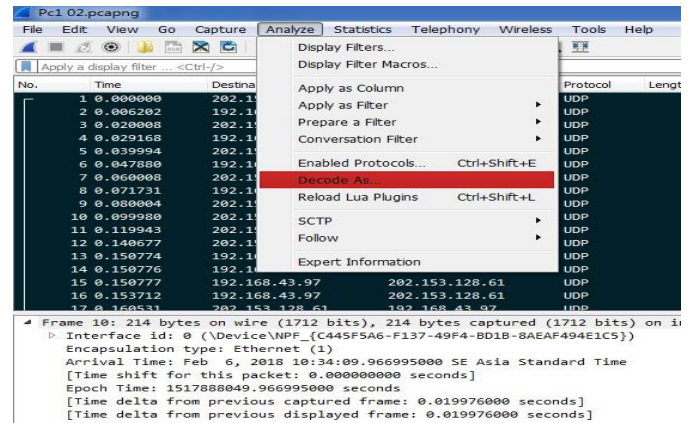
Melihat hasil dari tabel 2, terlihat bahwa *delay* yang didapat dari *client1* maupun *client2* pada pengujian *delay*. Hasil yang didapat dari *client1 delay* yang terbesar adalah 0,019934736 ms dan *delay* yang terkecil adalah 0,012908102 ms. Sedangkan hasil yang didapat dari *client2 delay* yang terbesar adalah 0,019987068 ms dan *delay* yang terkecil adalah 0,011158369 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

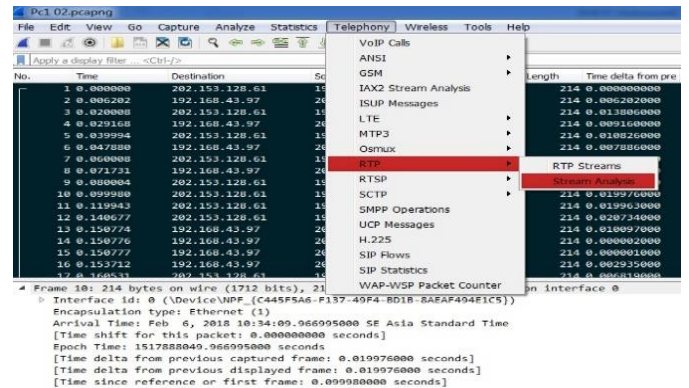
Jitter adalah variasi *delay* yang terjadi karena ketidak-stabilan kondisi jaringan sehingga waktu penerimaan paket dan penerima berbeda-beda. Tidak hanya parameter *delay* yang diamati namun juga parameter *Jitter*. Tujuan pengukuran *Jitter* adalah untuk mengetahui besarnya *Jitter* yang didapat dari 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 kemudian dilihat dari waktu memulai suatu paket di *client1* dengan waktu tiba paket di *client2* untuk kemudian dicari selisihnya.

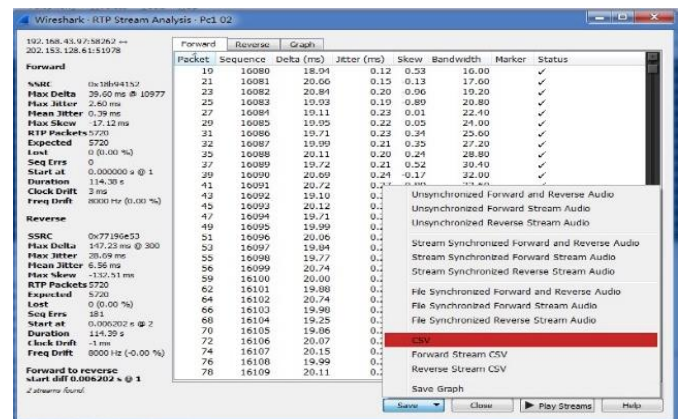
Pengukuran *jitter* dilakukan dengan cara melakukan filterisasi kemudian ke menu *analyze* untuk mengubah protokol dari *UDP* menjadi *RTP* pilih *decode as*, setelah melakukan *decode as* dan protokol telah diubah, berikutnya ke menu *telephony* untuk selanjutnya menganalisis *RTP Stream analysis*. 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



Gambar 17. RTP Stream Analisis Mencari Jitter



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.

Tabel 3. Rata-rata Jitter

Pengujian ke-	Rata-rata jitter (ms)	
	Panggilan dari Client 1 ke Client 2	Panggilan dari Client 2 ke Client 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 ms
5	0,42 ms	11,34 ms

Hitungan manualnya adalah sebagai berikut:

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

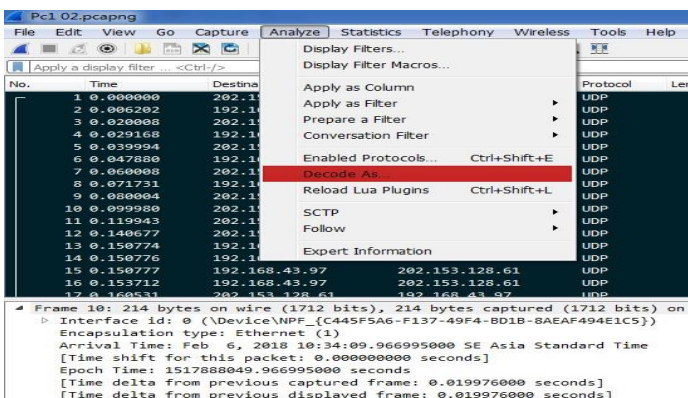
Terlihat bahwa hasil yang didapat dari *client1* maupun *client2* pada pengujian kestabilan, hasil yang didapat dari *client2* cenderung menampilkan hasil yang lebih besar, dibandingkan dengan hasil *jitter* pada *client1*. Dari pengujian sebanyak 5 kali dengan waktu bervariasi didapat hasil *jitter* 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

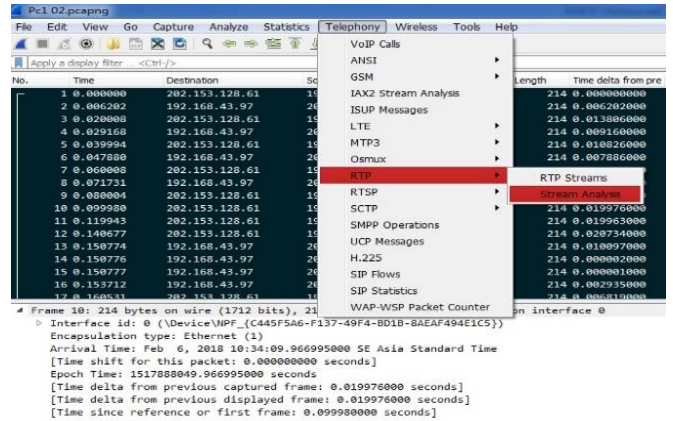
Packet Loss adalah banyaknya paket yang hilang atau rusak pada saat pengiriman paket dibandingkan dengan jumlah total pengiriman paket yang di terima pada sisi *client*. Satuan yang dipakai adalah persen (%). Pengukuran disini bertujuan untuk mendapatkan nilai perbandingan jumlah paket yang hilang atau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



Gambar 19. Mencari Rata-rata Packet Loss



Gambar 20. RTP Streams Analisis Mencari Packet Loss

	Forward	Reverse	Graph
	Packet	Sequence	Delta (ms)
Forward	1	17517	0.00
SSRC	2	17518	19.62
Max Delta	5	17519	19.36
Max Jitter	8	17520	22.66
Mean Jitter	10	17521	19.66
Max Skew	12	17522	20.07
RTP Packets	14	17523	17.56
Expected	16	17524	20.05
Lost	18	17525	20.68
Seq Errs	20	17526	20.37
Start at	22	17527	19.00
Duration	24	17528	20.00
Clock Drift	25	17529	20.67
Freq Drift	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

Pengujian ke-	Packet Loss	
	Panggilan dari Client 1 ke Client 2	Panggilan dari Client 2 ke Client 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*:

Tabel 5. Kategori Standar *Packet Loss*

Kategori Performasi	<i>Packet Loss</i>
Sangat bagus	0
Bagus	3 %
Sedang	15 %
Jelek	25 %

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.

Tabel 6. Hasil Analisis *QoS*

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

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 menggunakan 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
- Lestaringati, 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 berbasis *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