

PERANCANGAN SISTEM APLIKASI *VOICE CLIENT* UNTUK PENGEMBANGAN *CLIENT VOICE TERMINAL* YANG TERINTEGRASI DENGAN *NAVAL VOICE COMMUNICATION SYSTEM*

DESIGN OF A VOICE COMMUNICATION SYSTEM FOR TERMINAL VOICE CLIENT DEVELOPMENT INTEGRATED WITH NAVAL VOICE COMMUNICATION SYSTEM

Wahyu Putri Riskiyani¹, Aris Hartaman, S.T., M.T.², Wisnu Adji Kharisma, S.T.³

^{1,2}Prodi D3 Teknologi Telekomunikasi, Universitas Telkom

³ Divisi System Engineering, PT Len Industri (Persero)

wahyuputrir13@gmail.com, arishartaman@telkomuniversity.ac.id, wisnu.adji@len.co.id

Abstrak

Indonesia merupakan Negara Maritim, oleh karena itu diperlukan adanya pengawalan di wilayah laut sebagai upaya untuk menjaga kedaulatan Negara Kesatuan Republik Indonesia. Dalam menjalankan tugasnya untuk menjaga kedaulatan Negara Kesatuan Republik Indonesia, Kapal Republik Indonesia (KRI) memerlukan sistem komunikasi yang dapat meningkatkan efisiensi dalam koordinasi personel antar ruang di dalam kapal. Teknologi komunikasi yang digunakan dalam sistem ini adalah teknologi komunikasi VoIP yang berjalan pada protokol SIP dan RTP. Oleh karena itu, proyek ini akan menghasilkan *output* sistem aplikasi *client* untuk pengembangan *client voice terminal* yang terintegrasi dengan *naval voice communication system* berbasis *embedded system* yang memiliki fitur-fitur untuk mempermudah terjadinya komunikasi antar ruang di dalam kapal laut.

Pada Proyek Akhir ini difokuskan pada aplikasi *core* yang akan berjalan pada sistem *embedded linux OS* dan *UI apps development*. Aplikasi *core* ini dibangun menggunakan Linphone SDK dan bahasa pemrograman C yang akan diintegrasikan dengan *user interface* dan *operating system* pada Beaglebone Black dan 4D LCD Cape 7".

Hasil dari perancangan sistem komunikasi adalah dapat menukarkan suara antar pengguna dengan baik. Selain itu, kualitas sistem komunikasi yang dihasilkan dari pengujian menggunakan parameter *Quality of Service* (QoS) dan *Mean Opinion Score* (MOS) dengan panggilan berjarak 7m dan 25m. Kedua pengukuran tersebut menghasilkan kualitas yang bagus dilihat dari parameter QoS. Nilai MOS yang didapat dari perhitungan QoS dengan panggilan berjarak 7m maupun 25m berada pada kategori baik.

Kata Kunci: VoIP, Aplikasi Core, Embedded System, Beaglebone Black, Intranet, QOS

Abstract

Indonesia is a maritime country, therefore it is necessary to have an escort in the sea area as an effort to maintain the sovereignty of the Unitary State of the Republic of Indonesia. In carrying out its duties to maintain the sovereignty of the Unitary State of the Republic of Indonesia, the Republic of Indonesia Ship (KRI) requires a communication system that can increase efficiency in coordinating personnel between spaces on board. The communication technology used in this system is VoIP communication technology that runs on SIP and RTP protocols. Therefore, this project will produce a client application system output for the development of client voice terminals that are integrated with the naval voice communication system based on embedded systems that have features to facilitate communication between spaces in ships.

In this Final Project, the focus is on the core application that will run on the Linux OS embedded system and UI apps development. This core application is built using Linphone SDK and C programming language which will be integrated with the user interface and operating system on Beaglebone Black and 4D LCD Cape 7".

The result of the communication system design is that it can exchange voices between users well. In addition, the quality of the communication system generated from testing using Quality of Service (QoS) and Mean Opinion Score (MOS) parameters with calls 7m and 25m away. Both measurements produce good quality in terms of QoS parameters. The MOS value obtained from the QoS calculation with calls within 7m and 25m is in the good category.

Keywords: VoIP, Core Application, Embedded System, Beaglebone Balck, Intranet, QOS

1. PENDAHULUAN

Ketika menjalankan tugas dalam menjaga kedaulatan Negara Kesatuan Republik Indonesia, Kapal Republik Indonesia (KRI) memerlukan suatu sistem komunikasi terintegrasi yang dapat dijadikan sebagai sarana

komunikasi antar personel untuk menghindari miskomunikasi. *Naval voice communication system* merupakan sistem komunikasi yang terintegrasi pada kapal angkatan laut dan memiliki fitur-fitur yang telah disesuaikan sehingga mudah digunakan dan menghasilkan komunikasi yang akurat dengan *quality of service* yang baik.

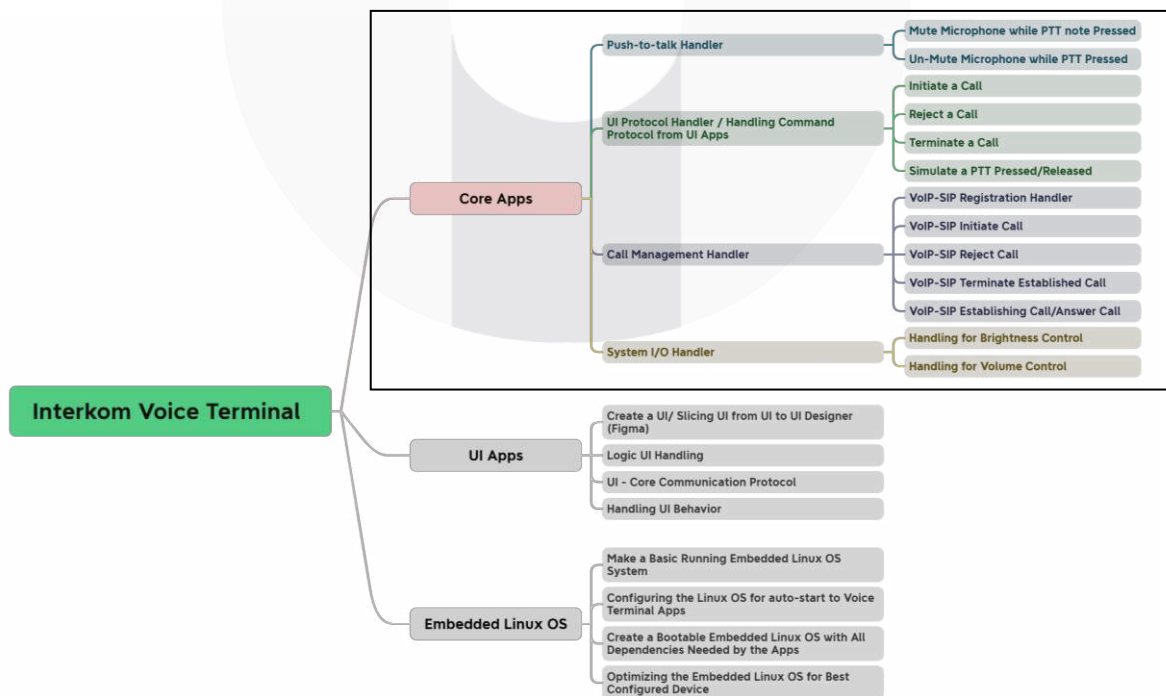
Voice over Internet Protocol (VoIP) merupakan teknologi yang memanfaatkan *Internet Protocol* untuk menyediakan komunikasi *voice* secara elektronik dan *real-time* [1]. Sinyal suara diubah menjadi kode digital yang kemudian akan ditransmisikan dalam bentuk paket-paket data kemudian dikirim melalui sebuah media transmisi berupa kabel. Teknologi VoIP juga digunakan pada kapal laut yang memerlukan banyak *node* dalam berkomunikasi seperti contoh pada ruang kontrol, ruang mesin, dan ruang lainnya. Saat berkomunikasi di dalam kapal laut dibutuhkan hasil komunikasi yang memiliki kelancaran saat digunakan. Sehingga dibutuhkan suatu sistem komunikasi yang mendukung kelancaran saat komunikasi sedang berjalan agar tidak terjadi suatu *miss communication* atau hal lain yang tidak diinginkan. Pada pengembangannya, platform *operating system (OS)* yang banyak digunakan dalam aplikasi sistem VoIP adalah Linux. Linux digunakan karena sifatnya yang *open source* dan ringan saat dijalankan pada komputer. Selain itu, dengan penggunaan Linux sebagai OS juga akan mempermudah dalam pengembangan aplikasi VoIP.

Voice Communication System (VCS) client terdapat tiga luaran berupa *software* komunikasi *VCS client* atau aplikasi *core*, *software UI VCS client*, *operating system VCS client*. Proyek ini berfokus pada pengembangan *software* aplikasi *core* untuk meningkatkan performa dan efisiensi dari aplikasi tersebut. Aplikasi ini dibuat ulang dengan peningkatan performa dan penggunaan *resource* yang lebih efisien. Pengembangan ini bersifat berkelanjutan sehingga akan terus ada pembaruan-pembaruan terhadap sistem dan aplikasi.

Dalam penelitian proyek akhir ini, dirancang aplikasi *core* dari sistem komunikasi VoIP *client* dengan fitur komunikasi yang dapat mempermudah komunikasi antar matra TNI dan bertujuan untuk mengetahui kualitas layanan pada aplikasi VoIP *client*. Untuk mengetahui kualitas dari sistem komunikasi, dilakukan pengujian kualitas layanan dengan dua metode, yaitu kualitatif dan kuantitatif menurut parameter QoS. Metode kualitatif digunakan untuk mengetahui kualitas panggilan yang terjadi antar *voice terminal unit*, sedangkan metode kuantitatif merupakan metode yang digunakan untuk mengetahui kualitas layanan melalui perbandingan dua kabel ethernet dengan panjang yang berbeda. Kualitas layanan yang digunakan sebagai parameter kedua metode ini adalah *delay*, *jitter*, *troughput*, dan *packetloss*.

2. PERANCANGAN APLIKASI CORE VOIP CLIENT

Pada proyek akhir ini, dilakukan perancangan sistem komunikasi *voice client* untuk pengembangan *client voice terminal* yang terintegrasi dengan *naval voice communication system*. Sistem yang dirancang merupakan sebuah aplikasi *core* yang dapat berkomunikasi antar VoIP *client* menggunakan Linphone SDK dengan bahasa pemrograman C. Sistem ini merupakan sistem pada bagian aplikasi *core* yang merupakan pengembangan lanjutan dari *Embedded Linux OS*. Aplikasi *core* terpasang pada Beaglebone Black yang terintegrasi dengan 4D LCD CAPE 7”, yang selanjutnya aplikasi *core* ini dapat merespon perintah dari UI. Pembagian kerja terdapat pada Gambar 2.1. Penulis mengerjakan bagian *Core Apps*.



Gambar 2. 1 Pembagian Kerja *Intercom Voice Terminal*

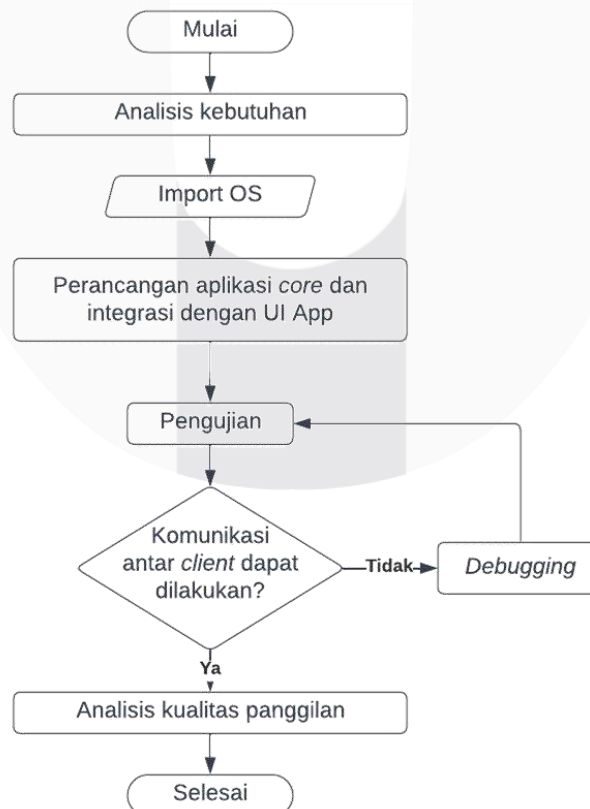
2.1 Tahapan Perancangan

Aplikasi *core* yang dirancang, akan diimplementasikan dengan UI dan *operating system* di dalam Beaglebone Black dan 4D LCD CAPE 7" seperti pada Gambar 2.2. Beaglebone Black merupakan *platform* pengembangan berbasis ARM yang dikembangkan oleh BeagleboneBoard.org. Beaglebone Black biasa digunakan untuk pengembangan proyek elektronik, *Internet of Things* (IoT), dan aplikasi lain karena kemampuan dan fleksibilitas yang tinggi. Sedangkan 4D LCD CAPE 7" merupakan perangkat keras berupa papan tambahan yang dirancang khusus untuk Beaglebone Black dengan ukuran layar tujuh inch. 4D LCD CAPE 7" memiliki fitur tujuh tombol dan layar sentuh atau *capacitive touch controller* dengan resolusi 480x272 [2].



Gambar 2. 2 Perangkat Voice Terminal Unit

Pada proyek akhir ini, dilakukan perancangan aplikasi *core* VoIP *client* dengan beberapa tahapan yang dapat dilihat pada Gambar 2.3 berikut:



Gambar 2. 3 Diagram Alir Pengerjaan Proyek Akhir

Berdasarkan Gambar 2.3, akan dijelaskan tahapan dan alur proses perancangan aplikasi *core VoIP client* yang dilakukan melalui beberapa tahap seperti berikut:

1. Analisis Kebutuhan

Tahap pertama, analisis kebutuhan yang diperlukan pada perancangan aplikasi *core*. Pada analisis kebutuhan ini, dilakukan diskusi dengan PT Len Industri (Persero) mengenai hal yang diperlukan dalam perancangan aplikasi. Dari hasil diskusi yang dilakukan, kebutuhan yang akan digunakan untuk pengerjaan proyek ini adalah kebutuhan perangkat lunak dan perangkat keras. Kebutuhan perangkat lunak untuk perancangan aplikasi adalah Liphone SDK, *operating system*, *text editor*, Putty, bahasa pemrograman C, GNU *compiler collection*, SIP *server*, *user interfaces*, Oracle VM Virtualbox, dan Wireshark untuk pengujian. Sedangkan kebutuhan perangkat keras untuk pengerjaan proyek akhir adalah *server*, Laptop/PC, Beaglebone Black, 4D LCD CAPE 7", kabel UTP 2m, 5m, dan 20m, *soundcard*, *earphone/headset*, VTU untuk pengujian.

2. Import Operating System

Tahap kedua, melakukan *import* sistem operasi yang telah terbangun untuk perancangan aplikasi *core*. *Operating system* merupakan tugas dari *embedded Linux OS develop* yang bertanggung jawab untuk menyiapkan OS dengan *dependencies* sesuai dengan kebutuhan agar dapat digunakan untuk pengerjaan aplikasi *core* dan UI.

3. Perancangan Aplikasi Core

Tahap ketiga, melakukan perancangan aplikasi *core* dan integrasi aplikasi *core* dengan UI *app*. Pada tahap ini terdiri dari konfigurasi *network* atau alamat IP, *setup* untuk *cross compile*, membuat fungsi C untuk registrasi, melakukan dan menerima panggilan, mengakhiri panggilan, *push-to-talk*, serta mengatur *volume* dan *brightness*. Setelah fungsi berhasil dirancang, selanjutnya adalah melakukan penggabungan dengan UI *app* agar aplikasi *core* dapat berjalan sesuai dengan perintah yang diterima dari UI *app*.

4. Pengujian Fungsionalitas

Tahap keempat, pengujian pertama yang bertujuan untuk mengetahui apakah aplikasi yang telah dirancang dapat melakukan komunikasi dengan baik atau tidak. Pengujian ini dilakukan dengan melakukan panggilan antar perangkat VTU yang telah terbangun dengan jarak yang berbeda. Apabila hasil pengujian berhasil, maka akan dilanjutkan ke tahap berikutnya, jika tidak sesuai dengan yang diharapkan, maka akan dilakukan *debugging*.

5. Analisis Kualitas Panggilan

Tahap kelima, analisis kualitas panggilan. Pada tahap ini yang dilakukan adalah pengujian menggunakan aplikasi Wireshark untuk mengetahui kualitas layanan yang dihasilkan dari komunikasi *client* dengan aplikasi VoIP, selanjutnya hasil dari pengukuran akan dibandingkan dengan parameter QoS. Perbandingan ini dilakukan untuk mengetahui kualitas layanan yang ada pada aplikasi VoIP dengan parameter QoS. Selain itu, dilakukan perhitungan nilai MOS berdasarkan nilai QoS untuk mengetahui nilai keseluruhan kualitas suara saat panggilan.

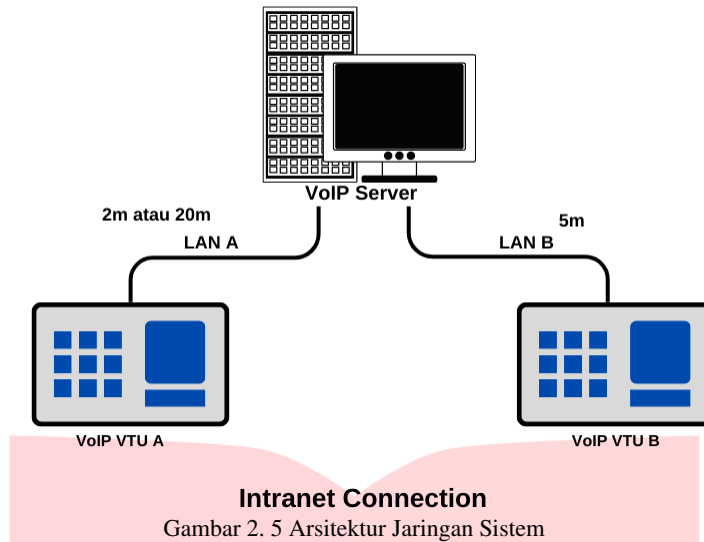
Hasil perancangan aplikasi *core* dapat menerima respon yang diperintahkan oleh *user interfaces* dari 4D LCD CAPE 7". Salah satu tampilan melakukan panggilan dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Tampilan Hasil Pengerjaan Proyek

2.2 Arsitektur Jaringan

Pada proyek akhir ini terdapat arsitektur jaringan yang sesuai untuk mendukung skenario percobaan fungsionalitas sistem aplikasi VoIP antar *Voice Terminal Unit* (VTU) agar komunikasi dapat berjalan dengan baik, maka dari itu akan dibangun arsitektur jaringan lokal dengan beberapa perangkat yang dijelaskan pada Gambar 2.4.



Skenario jaringan yang digunakan pada VoIP VTU ini adalah seperti yang telah digambarkan di atas, yaitu VTU A dan VTU B terhubung dengan VoIP server menggunakan kabel ethernet (UTP). VTU A merupakan perangkat yang akan diberikan kabel dengan panjang 2m atau 20m untuk pengujian, sedangkan VTU B akan diberikan kabel dengan panjang tetap yaitu 5m.

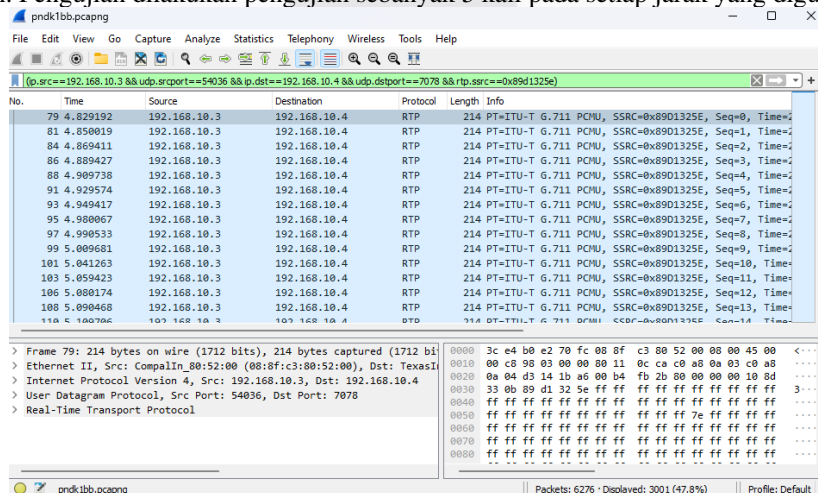
3.1 Skenario Pengujian

Terdapat dua pengujian yang akan dilakukan untuk mengetahui keberhasilan sistem yang telah dibangun. Pengujian pertama adalah pengujian fungsionalitas sistem dengan cara melakukan aktivitas panggilan dari VTU A ke VTU B dengan jarak 7m dan 25m. Parameter keberhasilan dari sistem ini adalah berfungsinya fitur-fitur aplikasi yang telah ditentukan pada bab sebelumnya, seperti melakukan panggilan, mengakhiri panggilan, *volume control*, dan *brightness control*.

Pengujian kedua adalah pengujian kualitas layanan menggunakan parameter QoS pada sistem aplikasi. Pengujian ini dilakukan menggunakan dua VTU, yaitu VTU A dan VTU B dengan memperhatikan jarak kabel yang digunakan. Pengukuran dilakukan saat kedua VTU telah terhubung dengan server melalui jaringan *intranet*. Setelah itu, melakukan panggilan suara selama 60 detik. Nilai parameter dalam pengujian dapat diperoleh dengan pengambilan data trafik paket pada sisi VTU A dengan menjalankan aplikasi Wireshark pada saat kedua VTU telah terhubung. Parameter pengujiannya adalah rata-rata *delay*, *jitter*, *packetloss*, dan *throughput*. Model skenario pengujian dapat dilihat pada Gambar 2.5.

3.3 Pengambilan Data

Data yang diambil merupakan data paket untuk perhitungan rata-rata *delay*, *packetloss*, *throughput*, dan *jitter*. Pengujian dilakukan dengan dua panjang kabel yang berbeda, yaitu 2m dan 20m. Kedua panjang kabel ini digunakan untuk menghubungkan VTU A ke server, sedangkan pada kabel yang digunakan untuk menghubungkan VTU B ke server memiliki panjang yang tetap yaitu 5m. Sehingga total jarak yang digunakan adalah 7m dan 25m. Pengujian dilakukan pengujian sebanyak 5 kali pada setiap jarak yang digunakan.



Gambar 3. 1 Proses Pengambilan Data

Pengambilan data menggunakan Wireshark dapat dilakukan dengan *capturing packets* seperti pada Gambar 3.1. Buka aplikasi Wireshark, memilih *interface* Ethernet dan klik *start* untuk memulai pengambilan data. Proses pengambilan data dimulai dengan *capturing packets* lalu melakukan panggilan dari VTU A ke VTU B selama 60 detik. Setelah pengambilan data paket telah selesai, maka dilakukan perhitungan kualitas panggilan berdasarkan parameter QoS yang selanjutnya digunakan untuk perhitungan nilai MOS dengan formula seperti di bawah ini:

1. Throughput

$$Throughput = \frac{\text{Data yang dikirim (bytes)}}{\text{Waktu pengiriman data (s)}} \tag{1} [3]$$

Tabel 3. 1 Parameter *Throughput*

Kategori	<i>Throughput</i> (kbps)
Sangat Bagus	65 s/d 100
Bagus	40 s/d 65
Sedang	5 s/d 10
Buruk	<5

2. Packetloss

$$Packetloss = \left(\frac{\text{Data yang dikirim} - \text{Paket data yang diterima}}{\text{Paket data yang dikirim}} \right) \times 100 \tag{2} [4]$$

Tabel 3. 2 Parameter *Packetloss*

Kategori	<i>Packetloss</i> (%)
Sangat Bagus	$0 \leq pl < 3$
Bagus	$3 \leq pl < 15$
Sedang	15
Buruk	≥ 25

3. Delay

$$\text{Rata - rata delay} = \frac{\text{Total delay}}{\text{Total paket yang diterima}} \tag{3} [4]$$

Tabel 3. 3 Parameter *Delay*

Kategori	<i>Delay</i> (ms)
Sangat bagus	<150
Bagus	150 ms s/d 300 ms
Sedang	300 ms s/d 450 ms
Buruk	>450 ms

4. Jitter

$$Jitter = \frac{\Sigma \text{Variasi delay}}{(\Sigma \text{Paket yang diterima} - 1)} \tag{4}[4]$$

Tabel 3. 4 Parameter *Jitter*

Kategori	<i>Jitter</i> (ms)
Sangat bagus	0 ms
Bagus	0 ms s/d 75 ms
Sedang	75 ms s/d 125ms
Buruk	125 ms s/d 225 ms

5. Perhitungan Nilai Mean Opinion Score (MOS)

MOS digunakan untuk mengetahui kualitas suara dalam jaringan VoIP. Pada proyek akhir ini, MOS dihitung menggunakan hasil perhitungan *E-model/ R Factor* berdasarkan standar ITU-T karena lebih objektif. Adapun hubungan antara *R Factor* dengan MOS dapat dilihat pada rumus dibawah ini [5]:

$$R = 94,2 - I_e - I_d \tag{5}$$

$$I_e = 0,024d + 0,11(d - 177,3)H(d - 177,3) \tag{6}$$

$$I_d = 7 + 30 \ln(1 + 15e) \tag{7}$$

Hasil dari *E-model / R Factor* memiliki korelasi dengan MOS sebagai berikut:

$$MOS = 1 + (0.035 R) + ((7 \times 10^{-6})R(R - 60)(100 - R)) \tag{8}$$

dimana,
 $d = \text{delay (ms)}$
 $e = \text{packetloss}$
 $R = R \text{ Factor}$
 $H =$

- $H(x) = 0, \text{ if } x < 0$
- $H(x) = 1, \text{ if } x > 0$

Tabel 3. 5 Nilai MOS

Nilai MOS	Kategori
4,3 s/d 5,0	Sangat Baik
4,0 s/d 4,3	Baik
3,6 s/d 4,0	Cukup Baik
3,1 s/d 3,6	Buruk
2,6 s/d 3,1	Cukup Buruk
1,0 s/d 2,6	Sangat Buruk

Data diambil menggunakan Wireshark seperti pada Gambar 3.1 yang dilakukan sebanyak 5 kali setiap setiap jaraknya dengan durasi 60 detik. Hasil pengambilan data yang didapat untuk perhitungan parameter QoS seperti *jitter*, *delay*, *packetloss*, dan *throughput* dapat dilihat pada Table 3.6.

Tabel 3. 6 Trafik Panggilan Berjarak 7m dan 25m

Pengujian ke-	Panggilan Berjarak 7m				Panggilan Berjarak 25m			
	Paket yang di capture	Paket yang dikirim (bytes)	Paket yang diterima (bytes)	Waktu (s)	Paket yang di capture	Paket yang dikirim (bytes)	Paket yang diterima (bytes)	Waktu (s)
1	3.001	642.242	642.242	59,991	2.966	634.724	634.724	59,300
2	2.996	641.144	641.144	59,901	3.024	647.164	647.164	60,441
3	3.021	646.522	646.522	60,380	3.023	646.922	646.922	60,439
4	3.023	646.950	646.950	60,430	3.012	644.568	644.568	60,221
5	3.022	646.764	646.764	60,381	3.022	646.764	646.764	60,381

3.3 Hasil Pengolahan Data

Pengambilan data yang telah dilakukan sebelumnya akan digunakan untuk perhitungan parameter QoS menggunakan formula *throughput* pada (1), *packetloss* pada (2), *delay* (3), dan *jitter* (4). Sehingga, didapatkan hasil perhitungan menurut parameter QoS setiap jarak yang digunakan dengan durasi panggilan 60 detik seperti pada Table 3.7.

Tabel 3. 7 Hasil Perhitungan Kualitas Panggilan

Pengujian ke-	Panggilan Berjarak 7m				Panggilan Berjarak 25m			
	<i>Throughput</i> (kbps)	<i>Packetloss</i> (%)	<i>Delay</i> (ms)	<i>Jitter</i> (ms)	<i>Throughput</i> (kbps)	<i>Packetloss</i> (%)	<i>Delay</i> (ms)	<i>Jitter</i> (ms)
1	85,645	0	19,99	10,68	85,628	0	19,9	1,13
2	85,627	0	19,99	0,99	85,658	0	19,8	5,7
3	85,660	0	19,98	1,09	85,629	0	19,9	4,9
4	85,646	0	19,99	9,59	85,627	0	19,9	7,5
5	85,691	0	19,98	1,13	85,672	0	19,9	0,8

Dari hasil perhitungan kualitas panggilan dengan parameter QoS di atas, maka selanjutnya didapatkan nilai *Mean Opinion Score* (MOS) berdasarkan standar ITU-T. Nilai MOS ini lebih objektif dengan perhitungan *R Factor* menggunakan parameter QoS daripada penilaian dari pendengaran manusia yang dianggap subjektif. Nilai MOS dengan panggilan berjarak 7m dan 25m dapat dilihat pada tabel 3.8.

Tabel 3. 8 Nilai MOS Pada Panggilan Berjarak 7m dan 25m

Pengujian ke-	Panggilan Berjarak 7m				Panggilan Berjarak 25m			
	le	Id	R	Nilai MOS	le	Id	R	Nilai MOS
1	0,47976	7	86,72024	4,2506	0,47760	7	86,72240	4,2506
2	0,47976	7	86,72024	4,2506	0,47520	7	86,72480	4,2507
3	0,47952	7	86,72048	4,2506	0,47760	7	86,72240	4,2506
4	0,47976	7	86,72024	4,2506	0,47760	7	86,72240	4,2506
5	0,47952	7	86,72048	4,2506	0,47760	7	86,72240	4,2506

Hasil perhitungan kualitas panggilan dengan parameter QoS di atas, selanjutnya dibandingkan dengan VTU yang belum dikembangkan. Adapun data VTU yang belum dikembangkan didapatkan dari PT Len Industri (Persero) sebagai berikut:

Tabel 3. 9 Hasil Pengujian VTU Lama

Pengujian ke-	Hasil(ms)
1	230
2	292
3	284
4	242
5	246
Total Average	258,8

4. KESIMPULAN

Berdasarkan hasil perancangan, pengujian, dan analisa yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Dari hasil perancangan, aplikasi ini dapat terintegrasi dengan OS dan aplikasi UI di Beaglebone Black dan LCD CAPE 7”.
2. Berdasarkan hasil pengujian fungsionalitas terhadap fitur-fitur pada sistem aplikasi yang telah dibuat dapat disimpulkan bahwa semua fungsi dapat berjalan dengan baik sebagaimana semestinya.
3. Berdasarkan dari hasil pengujian kualitas menurut parameter QoS, pada pengujian berjarak 7m dan 25m didapatkan nilai *delay* pada setiap pengujian yang masuk kedalam kategori sangat bagus, *jitter* dalam kategori bagus, *throughput* dalam kategori sangat bagus, *packetloss* sangat bagus, dan rata-rata nilai MOS 4,25 yang berarti masuk kedalam kategori bagus.
4. Kualitas perangkat setelah dikembangkan lebih baik terutama pada nilai *delay* yang dibandingkan dengan perangkat lama (VTU B). *Delay* yang dihasilkan perangkat lama lebih dari 100 ms. Sedangkan setelah dikembangkan, nilai *delay* berada dibawah 50 ms.

DAFTAR PUSTAKA

- [1] Berlian, "Membangun Server VoIP Berbasis Asterisk," *Jurnal Media Infotama*, vol. 16, pp. 24-31, 2020.
- [2] 4DSysyems, "4DCAPE-70T," 2023. [Online]. Available: <https://4dsystems.com.au/products/4dcape-70t/>. [Accessed 3 Agustus 2023].
- [3] A. Maydiana, U. K. Usman and N. Andini, "Perbaikan Performansi Layanan Komunikasi VoIP di Tol Layang Jakarta-Cikampek," *e-Proceeding of Engineering*, vol. VIII, no. 2, pp. 1616 - 1627, 2021.
- [4] A. A. Bhuwana, U. A. Ahmad and R. E. Saputra, "Pengukuran Infrastruktur Jaringan Komputer di Kawasan Asrama Universitas Telkom Menggunakan Metode QoS," *e-Proceeding of Engineering*, vol. VIII, no. 2, pp. 1974-1981, 2021.
- [5] N. Mumtaz, D. Perdana and Y. G. Bisono, "Performance Evaluation of VoIP Traffic in 5G Millimeter Wave Network," *IJSSST*, pp. 18.1 -18.6, 2019.