

Perancangan Aplikasi Voice Chat dengan Socket Programming pada Android untuk Jaringan Lokal

Allyza Nanda Purwari¹, Nugroho Suharto², Abdul Rasyid³

^{1,2} Program Studi Jaringan Telekomunikasi Digital,

Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

³ Program Studi Teknik Telekomunikasi,

Jurusan Teknik Elektro, Politeknik Negeri Malang, Indonesia

allyzananda@gmail.com, nugroho.suharto@polinema.ac.id, abdul.rasyid@polinema.ac.id

Abstract— Voice chat application is an application designed to replace the HT function on an Android smartphone. Besides being able to make voice calls, this application can be used to communicate with text. This application does not require cellular networks to communicate, simply by connecting to a network provided by an existing access point. The applications can be used without using a cellular network, when making an application requires a socket. Socket is an API that exists on Android studio software. While socket programming is a way to use socket components to create an application, one of which is a voice chat application. After being installed in android smartphones, each smartphone is connected to a network provided by an existing access point so that it can communicate by voice or text. Communication is carried out between smartphones with one another point to point. The test results can be concluded that this application can be used to communicate with voice and text, as long as the user is on the same network. This application can be used in all types of Android mobile phones for voice communication but communication with text can only be used on certain mobile brands.

Keywords— Communication, Operating System, Socket Programming

Abstrak— Aplikasi *voice chat* merupakan aplikasi yang dirancang untuk menggantikan fungsi HT pada *smartphone* android. Selain dapat melakukan panggilan dengan suara, aplikasi ini juga dapat digunakan untuk melakukan komunikasi dengan *text*. Aplikasi ini tidak memerlukan jaringan seluler untuk komunikasi, cukup dengan tersambung pada jaringan yang disediakan oleh *access point*. Agar aplikasi dapat digunakan tanpa menggunakan jaringan seluler, maka pada saat pembuatan aplikasi memerlukan adanya *socket*. *Socket* merupakan komponen/API (*Application Programming*) yang ada pada *software* android studio. Sedangkan *socket programming* merupakan cara untuk menggunakan komponen/API *socket* untuk membuat sebuah aplikasi salah satunya aplikasi *voice chat*. Aplikasi yang sudah dibuat nantinya akan diinstall pada beberapa *smartphone* android. Setelah terinstall, masing-masing *smartphone* dihubungkan pada jaringan yang disediakan oleh *access point* yang sudah ada agar dapat melakukan komunikasi dengan suara maupun *text*. Komunikasi yang dilakukan antar *smartphone* satu dengan yang lainnya secara *point to point*. Hasil pengujian dapat disimpulkan bahwa aplikasi ini dapat digunakan untuk melakukan komunikasi dengan suara maupun *text*, selama *user* berada pada jaringan yang sama. Dan juga aplikasi ini dapat digunakan disegala jenis *handphone* Android untuk melakukan komunikasi dengan suara namun pada komunikasi dengan *text* hanya bisa digunakan pada merk *handphone* tertentu saja.

Kata kunci— Komunikasi, Sistem Operasi, *Socket Programming*

I. PENDAHULUAN

Aplikasi *voice chat* merupakan aplikasi yang dirancang untuk menggantikan fungsi HT pada *smartphone* android. Dimana selain dapat melakukan panggilan dengan suara, aplikasi ini juga dapat digunakan untuk melakukan komunikasi dengan *text*. Untuk melakukan komunikasi, aplikasi ini tidak memerlukan jaringan seluler[1], cukup dengan tersambung pada jaringan yang disediakan oleh *access point* yang sudah ada. Karena penggunaan jaringan seluler dibutuhkan adanya server sedangkan untuk penggunaan *access point* tidak memerlukan server[2]. Agar aplikasi dapat digunakan tanpa menggunakan

jaringan seluler[3], maka pada saat pembuatan aplikasi memerlukan adanya *socket*.

Socket merupakan komponen/API (*Application Programming*) yang ada pada *software* android studio[4][5]. Sedangkan *socket programming* merupakan cara untuk menggunakan komponen/API *socket* untuk membuat sebuah aplikasi salah satunya aplikasi *voice chat*. Agar suatu *socket* dapat berkomunikasi dengan *socket* lainnya, maka *socket* butuh diberi suatu alamat unik berupa Alamat IP dan Nomer Port sebagai identifikasi[6][7]. Dengan menggunakan *socket programming* maka dapat dilakukan pertukaran informasi tanpa menggunakan jaringan seluler dan jangkauan untuk

berkomunikasi tidak terbatas selama terhubung pada *access point*.

Aplikasi yang sudah dibuat nantinya akan diinstall pada beberapa *smartphone* android. Setelah terinstall, masing-masing *smartphone* dihubungkan pada jaringan yang disediakan oleh *access point* yang sudah ada agar dapat melakukan komunikasi dengan suara maupun teks. Komunikasi yang dilakukan antar *smartphone* satu dengan yang lainnya secara *point to point*. Dilakukan pula pengukuran kualitas jaringan, yaitu QoS (Quality of Service).

Quality of Service (QoS) adalah kemampuan suatu elemen jaringan, seperti aplikasi jaringan, host, atau router untuk memiliki tingkatan jaminan bahwa elemen jaringan tersebut dapat memenuhi kebutuhan suatu layanan[8][9]. Terdapat beberapa parameter Quality of Service (QoS), yaitu sebagai berikut:

1. *Bandwidth*

Bandwidth adalah luas atau lebar cakupan frekuensi yang digunakan oleh sinyal dalam medium transmisi.

2. *Throughput*

Throughput adalah kemampuan sebenarnya suatu jaringan dalam melakukan pengiriman data. Biasanya throughput selalu dikaitkan dengan bandwidth dalam kondisi yang sebenarnya. Throughput dirumuskan pada persamaan (1).

$$Throughput = \frac{Packet\ received\ (kb)}{Time\ transmitted\ (s)} \quad (1)$$

Adapun standar Throughput menurut TIPHON ditunjukkan pada Tabel I.

TABEL I
STANDAR THROUGHPUT MENURUT TIPHON

Kategori Throughput	Throughput	Indeks
Bad	0-338 kbps	0
Poor	338-700 kbps	1
Fair	700-1200 kbps	2
Good	1200 kbps – 2,1 Mbps	3
Excelent	>2,1 Mbps	4

3. *Jitter*

Jitter adalah variasi atau perubahan latency dari *delay* atau variasi waktu kedatangan paket. Jitter juga didefinisikan sebagai gangguan pada komunikasi digital maupun analog yang disebabkan oleh perubahan sinyal karena referensi posisi waktu. Adapun standar jitter menurut TIPHON ditunjukkan pada Tabel II.

TABEL II
STANDAR JITTER MENURUT TIPHON

Kategori Jitter	Jitter	Indeks
Poor	125-225 ms	1
Medium	75-125 ms	2
Good	0-75 ms	3
Perfect	0 ms	4

4. *Packet Loss*

Packet loss adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Paket yang hilang ini dapat terjadi karena collision dan

congestion pada jaringan. Packet Loss dihitung berdasarkan persentase paket yang berhasil dikirim, dirumuskan pada persamaan (2).

$$Packet\ Loss = \frac{Packet\ transmitted - Packet\ Received}{Time\ transmitted} \times 100\% \quad (1)$$

Adapun standar *packet loss* menurut TIPHON ditunjukkan pada Tabel III.

TABEL III
STANDAR PACKET LOSS MENURUT TIPHON

Kategori Jitter	Jitter	Indeks
Poor	>25%	1
Medium	12-24%	2
Good	3-14%	3
Perfect	0-2%	4

5. *Latency*

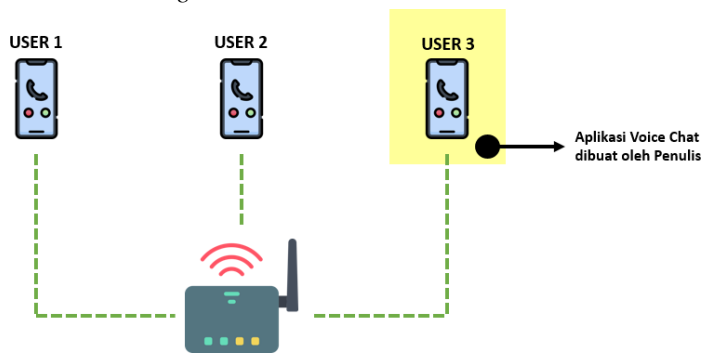
Latency adalah total waktu tunda suatu paket yang diakibatkan oleh proses transmisi dari satu titik ke titik lain yang menjadi tujuannya. *Delay* di dalam jaringan terdiri dari *delay processing*, *delay packetization*, *delay serialization*, *delay jitter buffer* dan *delay network*. Adapun standar *latency* menurut TIPHON ditunjukkan pada Tabel IV.

TABEL IV
STANDAR LATENCY MENURUT TIPHON

Kategori Latency	Latency	Indeks
Poor	>450 s	1
Medium	300-450 s	2
Good	150-300 s	3
Perfect	<150 s	4

II. METODE PENELITIAN

A. Blok Diagram Sistem



Gambar 1. Diagram blok sistem

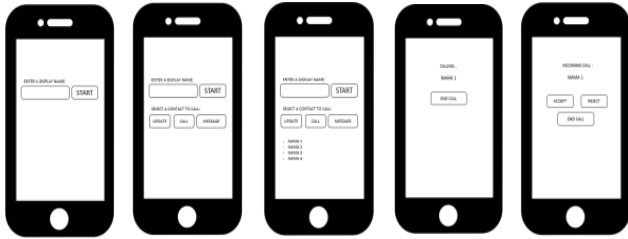
Pada Gambar 1 ditunjukkan blok diagram perancangan sistem diatas akan dijelaskan mengenai alur yang dilakukan selama penelitian, diantaranya sebagai berikut:

Melakukan instalasi aplikasi *voice chat* pada masing – masing *handphone user*. Selanjutnya *user* menghubungkan *smartphone* ke jaringan lokal yang sudah tersedia (*user* terhubung melalui *Access point*) agar dapat menjalankan aplikasi *voice chat*. *Handphone* yang sudah sudah terhubung ke jaringan lokal maka dapat menjalankan aplikasi yang sudah dibuat untuk melakukan komunikasi. *User* mengirimkan data ke *user* lain berupa *voice chat* dan pesan

text secara *point-to-point* untuk mendapatkan nilai data *QoS*.

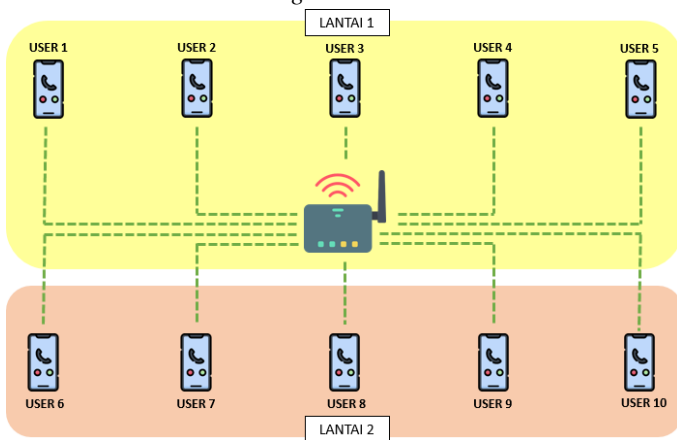
B. Perancangan Aplikasi

Aplikasi dirancang menggunakan *software* Android Studio. Dimana pada Android Studio terdapat API *socket* yang digunakan untuk merancang aplikasi. Layout perancangan aplikasi ditunjukkan pada Gambar 2.



Gambar 2. Layout perancangan aplikasi

C. Flowchart Alur Pengambilan Data



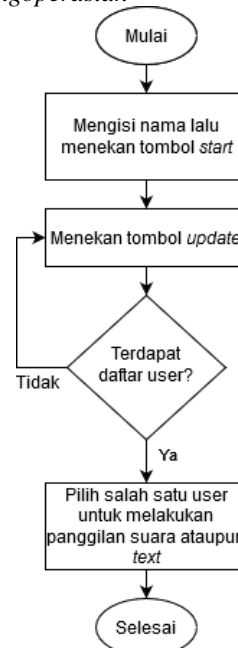
Gambar 3. Flowchart alur pengambilan data

Pada blok diagram alur pengambilan data akan dijelaskan mengenai alur yang dilakukan selama penelitian, diantaranya sebagai berikut:

Prosedur yang pertama yaitu meng-*install* aplikasi *Voice Chat* pada handphone masing – masing *user*. Proses instalasi selesai, selanjutnya masing-masing *user* menyambungkan handphone pada jaringan yang sudah ada (router). Setelah tersambung pada jaringan yang sama, maka masing-masing *user* dapat melakukan panggilan dengan suara maupun dengan *text*. *User* 1 akan melakukan panggilan pertama ke *User* 2, selanjutnya *User* 1 akan melakukan panggilan kedua pada *User*

3. *User* 1 akan melakukan panggilan pada setiap *user* secara bergantian. Begitu pula dengan *User* 2, *User* 3, *User* 4, *User* 5, *User* 6, *User* 7, *User* 8, *User* 9, dan *User* 10 akan melakukan panggilan pada setiap *user* lainnya. Pada setiap panggilan akan dihitung kualitas jaringannya yakni *delay*, *throughput*, dan *packet loss*.

D. Prosedur Pengoperasian



Gambar 4. Diagram Alir Prosedur Pengoperasian

Berdasarkan diagram alur prosedur pengoperasian dapat dijelaskan sebagai berikut: Pertama *user* harus masuk ke aplikasi yang telah dirancang, kemudian mengisi kolom nama yang sudah tersedia. Setelah *user* mengisi nama selanjutnya *user* menekan tombol *start* agar nama yang dimasukkan dapat tersimpan. Selanjutnya *user* menekan tombol *update* agar dapat melihat daftar *user* lain yang sudah tersimpan. Nama yang sudah disimpan akan dibroadcast setiap satu menit sekali. Jika daftar *user* sudah tampil maka *user* akan memilih salah satu nama *user* lainnya untuk melakukan komunikasi. Untuk melakukan komunikasi dengan suara maka *user* menekan tombol *call* sedangkan untuk melakukan komunikasi dengan *text* maka *user* menekan tombol *message*.



E. Parameter Penelitian

Parameter penelitian ditentukan berdasarkan sistem yang telah dibuat dan digunakan sebagai parameter pengujian dalam sistem. Berikut parameter penelitian dalam perancangan yaitu: Keberhasilan instalasi aplikasi pada beberapa handphone dengan merk dan tipe yang berbeda; Kemampuan perangkat komunikasi untuk terhubung pada server; dan Kualitas jaringan saat melakukan komunikasi.

III. HASIL DAN PEMBAHASAN

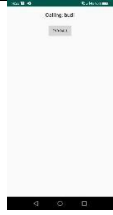
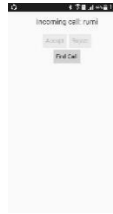


A. Pengujian Tampilan Sebelum Melakukan Komunikasi

TABEL V
HALAMAN UTAMA DAN DAFTAR USER

No.	Perencanaan	Pengujian Aplikasi	Hasil Program
1.	Tampilan awal sebelum <i>user</i> mengisikan nama dan menekan tombol start.		Berhasil
2.	Tampilan setelah <i>user</i> mengisikan nama dan menekan tombol <i>start</i> , lalu menekan tombol <i>update</i> untuk mengetahui daftar <i>user</i> lainnya yang terhubung pada jaringan yang sama.		Berhasil

B. Pengujian Tampilan Saat Melakukan Komunikasi

TABEL VI
MELAKUKAN PANGGILAN DAN CHATTING

No.	Perencanaan	Pengujian Aplikasi	Hasil Program
1.	Tampilan saat <i>user</i> melakukan panggilan, dimana terdapat tombol <i>end call</i> untuk mengakhiri panggilan yang sedang berlangsung.		Berhasil
2.	Tampilan saat <i>user</i> sedang menerima panggilan masuk, dimana terdapat beberapa tombol yakni tombol <i>accept</i> untuk menerima panggilan, tombol <i>reject</i> untuk menolak panggilan, dan <i>end call</i> untuk mengakhiri panggilan yang sedang berlangsung.		Berhasil
3.	Tampilan daftar pesan yang masuk dari berbagai <i>user</i> yang saling terhubung.		Berhasil
4.	Tampilan saat <i>user</i> satu dengan <i>user</i> lainnya saling mengirim pesan <i>text</i> .		Berhasil

C. Pengujian Komunikasi Dengan Suara

TABEL VII
DAFTAR TIPE HANDPHONE DAN TITIK LOKASI

No.	Nama	Tipe Handphone	Lokasi
1.	Rumi	Advan	Lt. 1
2.	Reza	Asus	Lt. 1
3.	Bono	Lenovo	Lt. 1
4.	Adit	Samsung Prime	Lt. 1
5.	Jojo	Redmi	Lt. 1
6.	Rinda	Samsung	Lt. 2
7.	Tata	Samsung	Lt. 2
8.	Budi	Oppo	Lt. 2
9.	Mukidi	Realme	Lt. 2
10.	Tule	Samsung A8+	Lt. 2

Setelah melakukan pengujian didapatkan hasil dari 10 *user* yang menggunakan aplikasi yang telah dirancang diketahui bahwa tipe handphone juga mempengaruhi kecepatan penggunaan aplikasi pada saat menerima panggilan, karena setelah pengujian dari *user* Adit dan *user* Tule terdapat perbedaan kecepatan dimana mereka menggunakan merk handphone yang sama namun tipe handphone yang berbeda dan hasilnya aplikasi pada *user* Tule lebih cepat saat ada panggilan masuk daripada *user* Adit.

D. Pengujian Komunikasi Dengan Text

Setelah melakukan pengujian didapatkan hasil dari 10 *user* yang menggunakan aplikasi yang telah dirancang diketahui bahwa merk handphone juga mempengaruhi kinerja aplikasi pada saat chatting, karena setelah pengujian dari *user* Adit dan *user* Tule terdapat perbedaan kecepatan dimana mereka menggunakan merk handphone yang sama namun tipe handphone yang berbeda dan hasilnya aplikasi pada *user* Tule lebih cepat saat ada panggilan masuk daripada *user* Adit.

E. Hasil Pengujian Komunikasi Pada Kondisi Berhasil

1) Kualitas Komunikasi Rumi (192.168.12.112) : Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21250 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2.2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

2) Kualitas Komunikasi Reza (192.168.12.105) : Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21650 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek.

Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

3) Kualitas Komunikasi Bono (192.168.12.108) : Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21750 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

4) Kualitas Komunikasi Adit (192.168.12.102): Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21550 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

5) Kualitas Komunikasi Jojo (192.168.12.113) : Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21550 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

6) Kualitas Komunikasi Rinda (192.168.12.106): Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik. Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 12852 Bytes perdetik.

7) Kualitas Komunikasi Rumi (192.168.12.110) : Untuk paket lossnya sama dengan 0 persen dikarenakan banyak data dikirim sama dengan data yang paling akhir pada proses. Sedang untuk throughputnya adalah 21100 Bytes perdetik. Untuk rata rata *delay* 20 proses awal pada hasil monitoring ada 2.1 detik. Bila dilihat pada tabel kategori *delay* termasuk jelek. Hal ini dikarenakan karena membutuhkan waktu untuk mencari jalur untuk sistem jaringan komunikasi dan melewati beberapa tahapan dalam prosesnya sehingga membutuhkan waktu lebih dari 1 detik.

IV. KESIMPULAN

Aplikasi ini dapat digunakan untuk melakukan komunikasi dengan suara maupun *text*, selama *user* berada pada jaringan yang sama. Aplikasi ini dapat digunakan disegala jenis *handphone* Android untuk melakukan komunikasi dengan suara namun pada komunikasi dengan *text* hanya bisa digunakan pada merk *handphone* tertentu saja. *Handphone* yang tidak bisa digunakan untuk komunikasi dengan *text* ialah merk oppo dan realme. Pengujian kualitas jaringan pada saat melakukan komunikasi terbilang buruk pada parameter *delay* dan *throughput* dan unggul pada parameter *packet loss*. Rata-rata *delay* pada *user* sebesar 2.05 detik dan rata-rata *throughput* sebesar 20 kbps. Sedangkan tingkat keberhasilan pengiriman data sebesar 0%.

REFERENSI

- [1] Kurose and Ross, *Computer Networking: A top-down approach featuring the Internet, 3rd edition*, Addison Wesley, 2004.
- [2] Krismawardana, Y., Christyono, Y., & Riyadi, M. A., "Perancangan dan Analisis Antena J-Pole Dualband Dengan Variasi Bentuk "T" Untuk Komunikasi Radio Transceiver Pada Pita VHF dan UHF". Semarang: Universitas Diponegoro., 2015.
- [3] Sofana, Iwan.. *Teori dan Modul Praktikum Jaringan Komputer*. Bandung: Modula, 2011.
- [4] Tua Lesmana, Indra, *Analisis Transfer Data Dengan Pemrograman Socket (Studi Kasus: Kendaraan Bergerak)*. Bandung: Politeknik Negeri Bandung, 2017.
- [5] Selamet, Rachmat, *Pemrograman Socket di Java*. Bandung: Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI, 2017.
- [6] Amin, Miftakul. *Implementasi Socket Programming dalam Pembuatan Sistem Antrian*. Bandar Lampung: Informatics & Business Institute Darmajaya, 2010.
- [7] Fajar, Ahmad Nurul. *Pemrograman Socket Dengan Java Dalam Mengembangkan Software Dengan Arsitektur Client Server*. Jakarta: UIEU, 2006.
- [8] Suhervan, *Analisis Penerapan QOS (Quality Of Service) pada jaringan fram Relay Menggunakan Cisco Router*. Jakarta: Universitas Esa Unggul, 2010.
- [9] Wulandari, Rika, *Analisis Qos (Quality Of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)*. Sukabumi: UPT Loka Uji Teknik Penambangan Jampang Kulon Lembaga Ilmu Pengetahuan Indonesia, 2016.