

IMPLEMENTASI KOMPRESI AUDIO MENGGUNAKAN ALGORITMA SHANNON-FANO PADA APLIKASI PODCAST

Irfan Agus Febriansya¹, Danang Haryo Sulaksono²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknik Elektro dan Teknologi Informasi, Institut Teknologi Adhi Tama Surabaya, Surabaya.

Informasi Artikel

Naskah Diterima: 26 April 2021

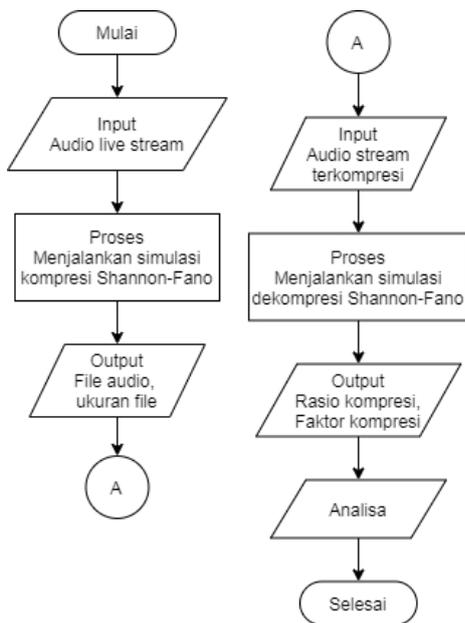
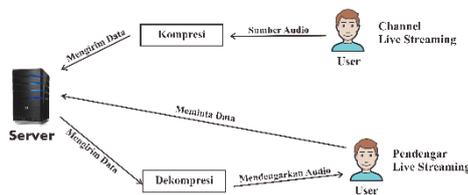
Direvisi : 15 Juni 2021

Disetujui : 16 Juni 2021

doi: 10.36055/setrum.v10i1.11171

***Korespondensi Penulis:**
irfanagus311@gmail.com

Graphical abstract



Abstract

Live Streaming is a live broadcast media that is popular with many people to get the latest information about politics, technology, education, and others. The longer the live stream will result in bigger data. While bandwidth and storage space has a maximum limit. To overcome the problem bandwidth when transmission and limited storage space can be overcome by compression and decompression techniques. Compression functions to shrink or compress the size of the audio data so as to save capacity bandwidth and storage space. While decompression functions to restore lost data, the resulting information can be heard in full without reducing the audio quality. In this test using the algorithm Shannon-fano, Shannon-fano is a compression algorithm lossless that produces binary code based on probability. From this test, it is carried out in 15 minutes each test and produces 10 compressed files that will be used as a reference in the test parameters. The average compression ratio value is 87.62%. The average value of the compression factor is 1.14%. Meanwhile, from testing QoS (Quality of Service) using two parameters Delay and Throughput. The Average throughput resulting value is 119 KBps. The Average delay resulting value is 12 ms, then it refers to the standard TIPPHON that can be categorized as "Very Good".

Keywords: Compression, Shannon-Fano, Live Stream, Audio, QoS

Abstrak

Live Streaming yaitu salah satu media penyiaran secara langsung yang sudah digemari banyak orang untuk memperoleh informasi terkini mengenai politik, teknologi, edukasi dan lain-lain. Semakin lama live streaming akan menghasilkan data yang besar. Sedangkan bandwidth dan ruang penyimpanan mempunyai batas maksimal. Untuk mengatasi masalah bandwidth saat transmisi dan ruang penyimpanan yang terbatas dapat teratasi dengan teknik kompresi dan dekompresi. Kompresi yang berfungsi untuk mengecilkan atau memampatkan ukuran data audio sehingga menghemat kapasitas bandwidth dan ruang penyimpanan. Sedangkan dekompresi berfungsi untuk mengembalikan data yang hilang, maka informasi yang dihasilkan dapat didengar secara utuh tanpa mengurangi kualitas audio. Dalam pengujian ini menggunakan algoritma Shannon-fano, Shannon-fano termasuk salah satu algoritma kompresi lossless yang menghasilkan kode biner berdasarkan probabilitas. Dari pengujian ini dilakukan dalam 15 menit setiap pengujian dan menghasilkan 10 file terkompresi yang akan dijadikan acuan dalam parameter pengujian. Nilai rata-rata rasio kompresi yang dihasilkan sebesar 87,62%. Nilai rata-rata faktor kompresi yang dihasilkan sebesar 1,14%. Sedangkan dari pengujian QoS (Quality of Service) menggunakan 2 parameter Delay dan Throughput. Nilai rata-rata Throughput yang dihasilkan yaitu 119 KBps. Nilai rata-rata Delay yang dihasilkan yaitu 12 ms, maka merujuk pada standar TIPPHON dapat dikategorikan "Sangat Bagus".

Kata kunci: Kompresi, Shannon-Fano, Live Stream, Audio, QoS

© 2021 Penerbit Jurusan Teknik Elektro UNTIRTA Press. All rights reserved

1. PENDAHULUAN

Podcast merupakan media pendistribusian file audio yang melalui internet agar bisa diputar di komputer maupun di pemutar portabel lainnya. Istilah *podcast* berasal dari kata 'pod' yang berarti *Playable on Demand* dan kata 'cast' yang berarti *Broadcasting*. Dalam sebuah *channel* *podcast* memiliki suatu tema dalam pembahasan misalnya tentang edukasi, teknologi, politik dan sebagainya[1]. Disetiap tema tersebut juga memiliki episode-episode dengan judul dan durasi yang berbeda beda, sehingga pendengar bisa memilih tema maupun judul yang mereka suka dan dapat mengabaikan tema maupun judul yang tidak disukai. Jadi kelebihan dari *podcast* itu sendiri dapat diakses dengan cara *streaming*, *live stream* dan kita dapat mendengarkan kapanpun kita suka.

Dikarenakan durasi sebuah *live* maupun konten *podcast* yang tidak di batasi maka, ada masalah yang terjadi pada *podcast* yaitu masalah penyimpanan data audio dan transmisi data. Semisal durasi *live streaming* 1 jam dengan *sample rate* 16000 Hz berarti data stream yang akan disiarkan dan disimpan dengan ukuran file 230400000 *byte* atau 230,4 *Megabyte*, maka akan membutuhkan banyak ruang penyimpanan[2]. Dengan ukuran data stream audio yang relatif besar akan memperlama proses transmisi. Proses ini juga dipengaruhi oleh besarnya *bandwidth* dan kondisi sinyal yang tidak stabil pada masing-masing orang dapat mengakibatkan terjadinya *buffering*. Dari permasalahan tersebut dapat teratasi dengan cara mengkompresi data stream audio untuk mengurangi ukurannya dan mempercepat proses transmisi sehingga mengurangi terjadinya *buffering* dan tanpa mengurangi kualitas audio tersebut.

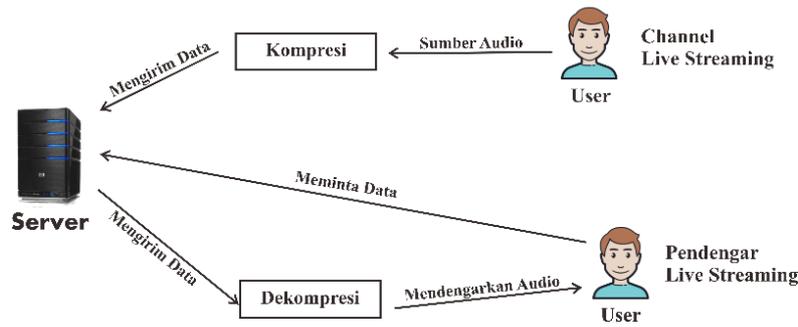
Kompresi data adalah proses mengubah aliran data input (sumber data asli) menjadi aliran data lain (output atau terkompresi) yang memiliki ukuran lebih kecil[3]. Ada dua tipe yang digunakan untuk memampatkan ukuran file, yaitu kompresi *lossless* dan kompresi *lossy*. Kompresi *lossless* merupakan suatu teknik kompresi data tanpa menghilangkan satupun informasi saat sebelum dikompresi[4]. Sedangkan kompresi *lossy* merupakan kompresi data yang menghilangkan beberapa informasi sehingga tidak seperti data aslinya[5][6].

Pada penelitian sebelumnya, teknik kompresi file gambar homogen menggunakan algoritma *Shannon-Fano*. Berdasarkan penelitian tersebut, peneliti menguji rasio kompresi rata-rata di sekitar 52%. Dan hasil yang diperoleh dari peneliti tersebut yaitu, gambar yang dikembalikan dalam proses dekompresi tidak kehilangan sedikitpun informasi[7].

Pada penelitian kali ini penulis menekankan penerapan kompresi data stream audio menggunakan algoritma *Shannon-Fano* ke dalam media *streaming* atau *podcast* berbasis android. Audio akan diencode dan dilakukan pencarian simbol berdasarkan probabilitas yang terbanyak kemudian membagi dua bagian dan bagian tersebut akan diberi label 0 dan 1[4]. Pada saat dekompresi algoritma *Shannon-Fano* bertujuan untuk mengembalikan data audio yang sudah dikompresi, sehingga pada saat audio di transmisikan ke pendengar diharapkan informasi pada audio dapat dikembalikan seperti halnya sebelum proses kompresi. Untuk

2. METODE PENELITIAN

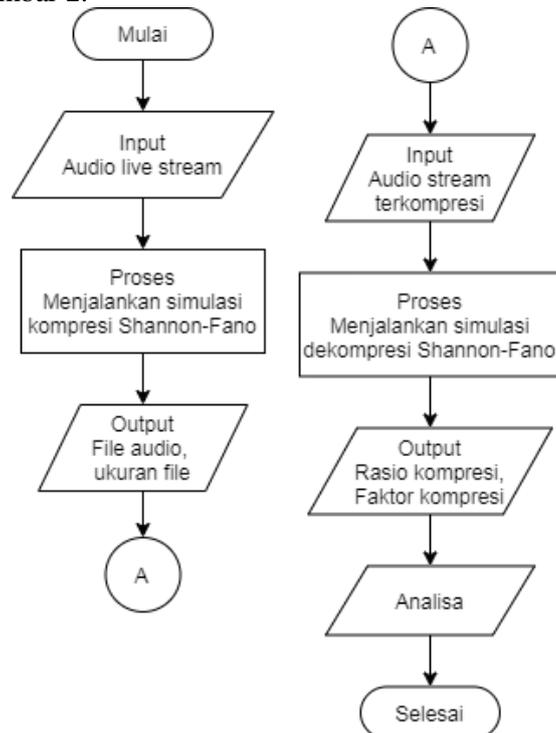
Gambaran umum sistem ini akan menjelaskan bagaimana skema rancangan aplikasi kompresi dan dekompresi algoritma *Shannon-Fano* yang berbasis Android bisa dapat dilihat pada Gambar 1. Dimana pengguna mempunyai *channel* untuk melakukan *Live Stream*, kemudian audio data stream dikompresi yang akan dikirim di Server. Pendengar akan meminta ke server sesuai yang diinginkan, setelah itu server merespon dan mengirimkan audio data stream yang terkompres dan akan didekompresi diperangkat pendengar.



Gambar 1. Gambaran Umum

2.1 Perancangan Sistem

Perancangan sistem ini menjelaskan mengenai alur proses pengerjaan sistem guna memudahkan implementasi. Data masukan, proses dan keluaran dari implementasi algoritma *Shannon-Fano* dapat dilihat pada gambar 2.



Gambar 2. Flowchart Sistem

Bedasarkan flowchart sistem yang ada pada Gambar 2 dapat dijelaskan sebagai berikut:

- Tahap pertama yaitu, data masukan yang berupa data audio live stream.
- Tahap kedua yaitu, data masukan tersebut akan diolah atau dikompresi dengan menggunakan algoritma *Shannon-Fano*.
- Tahap ketiga, keluaran yang dihasilkan yaitu data stream yang terkompresi.
- Tahap keempat yaitu, masukan yang berupa data stream audio yang telah terkompresi.
- Tahap kelima yaitu, data masukan akan diolah atau didekompresi dengan menggunakan algoritma *Shannon-Fano*.
- Tahap keenam, keluaran yang dihasilkan yaitu file audio, *Compression Ratio* dan *Compression Factor*.
- Tahap terakhir yaitu, Analisa hasil keluaran dari algoritma *Shannon-Fano*. Keluaran yang dihasilkan akan berupa tabel untuk mengetahui apakah faktor kompresi dari algoritma *Shannon-Fano* lebih baik.

2.2 Algoritma *Shannon-Fano*

Algoritma *Shannon-Fano* ditemukan oleh *Claude Shannon* dan *Robert Fano* dalam publikasi buku yang berbeda pada tahun 1949. Algoritma ini merupakan algoritma pertama yang ia perkenalkan untuk kompresi sinyal digital dalam bukunya yang berjudul "*The Mathematical Theory of Communication*". Dalam proses pengembangan algoritma ini, *Claude Shannon* dan *Robert Fano* memperoleh hasil reduksi minimum pada data[5].

Algoritma *Shannon-Fano* berdasarkan *variable length code*, yang berarti bahwa setiap simbol pada data diwakili oleh kode simbol yang lebih pendek dari simbol pada data. Jika semakin tinggi probabilitas suatu simbol, maka semakin pendek kode simbol. Algoritma ini menghasilkan kode simbol yang panjangnya tidak sama dalam memperkirakan panjang setiap kode simbol, sehingga kode tersebut unik dan dapat didekode. Algoritma *Shannon-Fano* bisa diterapkan dengan dua cara, yaitu dengan membangun tabel berdasarkan probabilitas pada setiap simbol dan dengan membangun pohon biner.

2.2.1 Algoritma Kompresi *Shannon-Fano*

Langkah-langkah kompresi algoritma *Shannon-Fano* sebagai berikut :

- 1) Membuat daftar frekuensi atau peluang kemunculan setiap simbol atau karakter dari data yang diencode.
- 2) Mengurutkan daftar tersebut berdasarkan frekuensi atau peluang kemunculan secara menurun (dari frekuensi simbol terbesar hingga yang terkecil).
- 3) Membagi dua bagian daftar tersebut dengan jumlah peluang kemunculan simbol yang sama atau hampir mendekati sama antara bagian yang satu dengan bagian yang lainnya.
- 4) Memberi label pada setiap bagian, bagian atas diberi label 0, sedangkan bagian bawah diberi label 1.
- 5) Mengulangi dari langkah ketiga sampai bagian tidak bisa dibagi lagi.

2.2.2 Algoritma Dekompresi *Shannon-Fano*

Langkah-langkah dekompresi algoritma *Shannon-Fano* sebagai berikut :

- 1) Membaca bit data pertama dari data yang dikompresi.
- 2) Jika bit tersebut tersedia pada daftar kode *Shannon-Fano*, maka bit tersebut disubstitusi menjadi simbol yang sesuai dengan kode *Shannon-fano*. Jika bit tersebut tidak tersedia pada kode *Shannon-Fano*, maka bit tersebut akan digabungkan dengan bit selanjutnya.
- 3) Ulangi langkah dua sampai ada bit yang sama dengan kode *Shannon-Fano*, substitusi kode tersebut menjadi simbol yang sesuai dan sampai bit terakhir dari data yang terkompresi.

2.3 Skenario Pengujian

Pada poin ini akan membahas skenario pengujian hasil kompresi algoritma *Shannon-Fano*. Yang pertama, metode yang digunakan untuk menguji kinerja dari kompresi menggunakan *Compression Ratio* dan *Compression Factor*. Saat dilakukan simulasi dimana pengguna melakukan live streaming yang akan menghasilkan ukuran data audio asli dan sudah terkompresi. Ukuran data audio sebelum dan sesudah terkompresi akan dijadikan parameter. Dalam pengujian ini menghasilkan nilai *Compression Ratio* dan *Compression Factor* yang kemudian dilakukan perhitungan rata-rata dari metode tersebut dan disajikan dalam bentuk tabel.

Yang kedua, metode yang digunakan dalam pengujian yaitu *Quality Of Service(QOS)*. Dalam tahap ini, pengujian menggunakan aplikasi wireshark untuk mendapatkan nilai besaran paket perdetik yang akan dijadikan parameter. Kemudian menghitung rata-rata nilai *Throughput* dan *Delay* dari semua transmisi audio live streaming yang disimulasikan. Penyajian ditahap ini menggunakan tabel yang nantinya akan menghasilkan kesimpulan.

2.3.1. *Compression Ratio*

Rasio kompresi(*Compression Ratio*) merupakan perbandingan antara ukuran data asli dengan data yang sudah terkompresi[8]. Cara mendapatkan nilai dari rasio kompresi dengan rumus:

$$\text{Compression Ratio} = \frac{\text{size of the output stream}}{\text{size of the input stream}}$$

1

2.3.2. *Compression Factor*

Faktor kompresi(*Compression Factor*) merupakan cara untuk mengetahui seberapa besar faktor yang dihasilkan dari proses kompresi[9]. Cara mendapatkan nilai dari faktor kompresi dengan rumus:

$$\text{Compression Factor} = \frac{\text{size of the input stream}}{\text{size of the output stream}} \quad \boxed{2}$$

2.3.3. *Throughput*

Throughput yaitu jumlah bit yang sukses diterima dalam perdetik [10]. Cara mendapatkan nilai dari *throughput* dengan rumus:

$$\text{Throughput} = \frac{\text{Paket yang diterima(Byte)}}{\text{Lama pengiriman paket}} \quad \boxed{3}$$

2.3.4. *Delay*

Delay adalah waktu yang dibutuhkan oleh sebuah paket data yang terhitung dari saat pengiriman sampai diterima.[11] Cara mendapatkan nilai dari *Delay* dengan rumus:

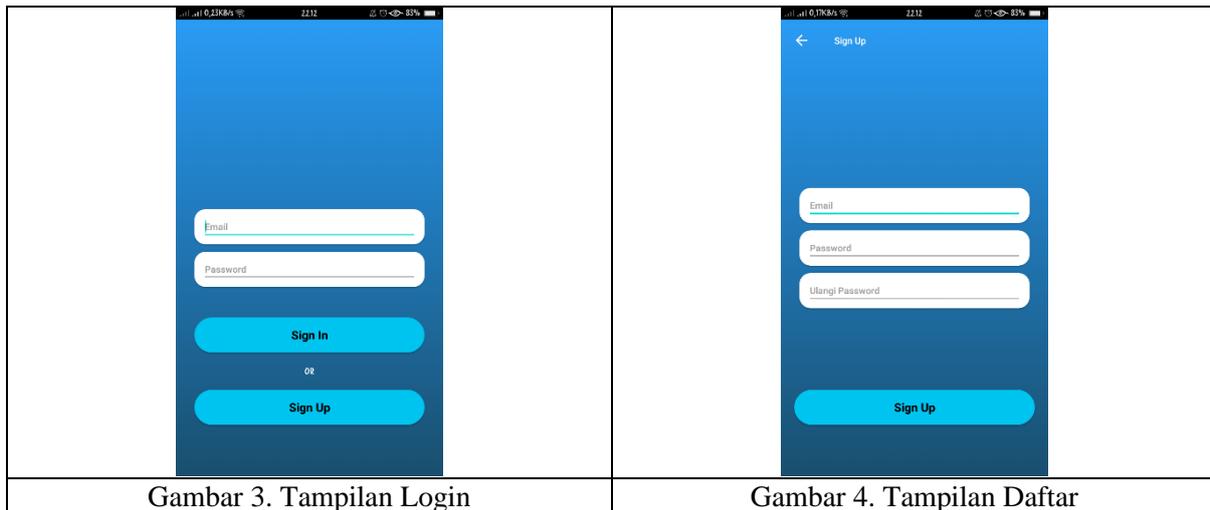
$$\text{Delay} = \frac{\text{Total delay}}{\text{Total paket yang diterima}} \quad \boxed{4}$$

3. HASIL DAN PEMBAHASAN

3.1 Tampilan Login dan Daftar

Tampilan login yang ditampilkan pertama kali jika pengguna belum mempunyai akun maupun pengguna sudah melakukan logout sebelumnya yang ditunjukkan pada Gambar 3. Untuk melakukan login menggunakan email dan password yang telah terdaftar pada aplikasi. Saat login dengan email yang belum terdaftar atau password yang di masukkan tidak sesuai maka akan muncul pemberitahuan.

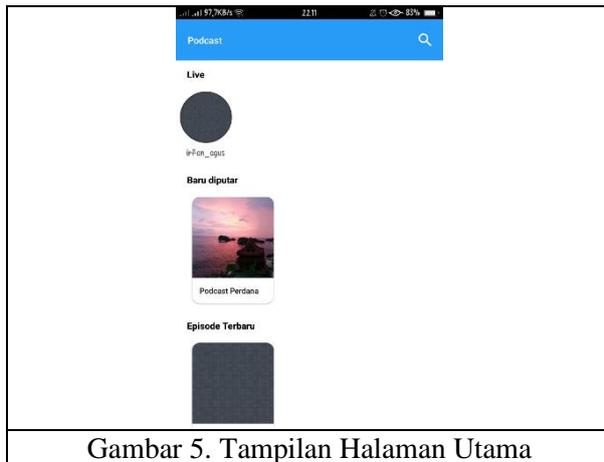
Tampilan daftar akun baru bagi yang belum mempunyai akun yang ditunjukkan pada Gambar 4. Pengguna membuat akun dengan memasukkan email, password dan konfirmasi password.



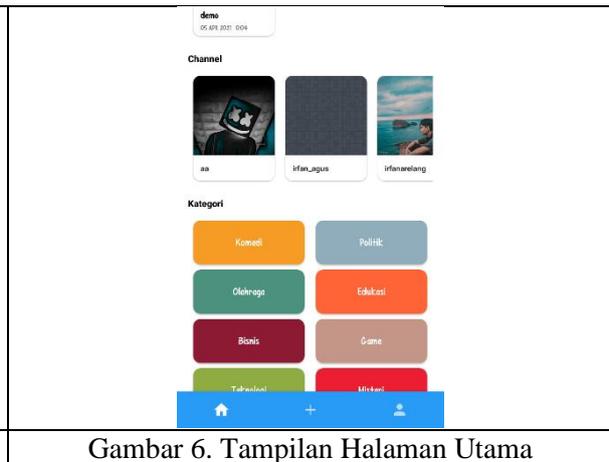
3.2 Tampilan Halaman utama

Halaman utama yang menampilkan daftar live, pemutaran terbaru, episode terbaru, channel dan kategori yang ditunjukkan pada Gambar 5 dan Gambar 6. Tampilan list live akan tersedia jika ada

yang melakukan live stream. Sedangkan yang pemutaran terbaru akan terlihat jika pengguna baru memutar episodenya. Untuk episode terbaru akan terlihat jika channel yang telah diikuti baru mengupload episode. Channel pada gambar diatas ditampilkan berjumlah 20 dengan secara acak.



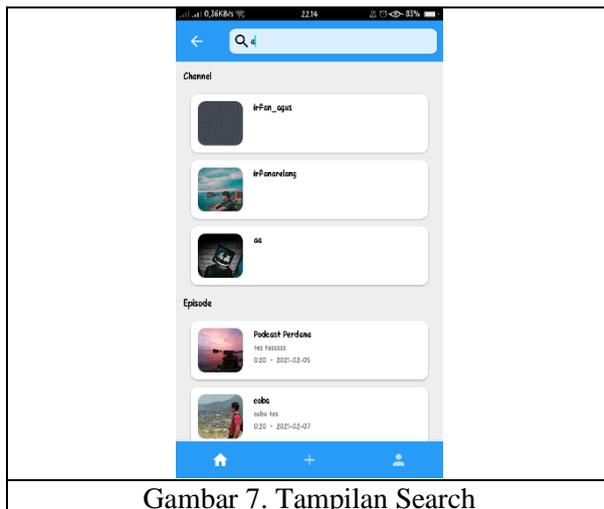
Gambar 5. Tampilan Halaman Utama



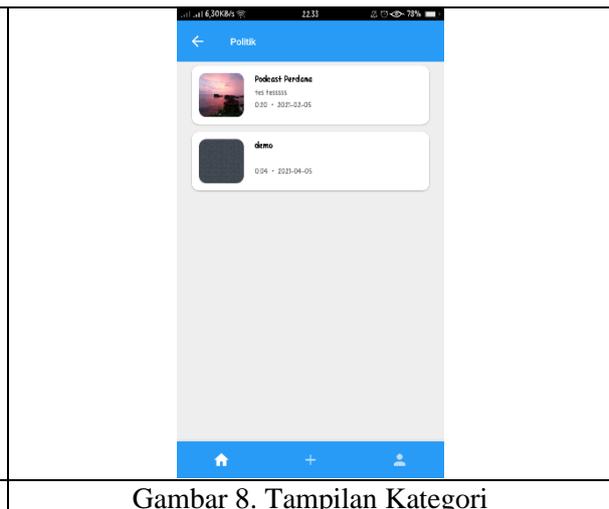
Gambar 6. Tampilan Halaman Utama

3.3 Tampilan Search dan Kategori

Tampilan pencarian nama channel dan episode yang berdasarkan keyword yang diinginkan oleh pengguna yang ditunjukkan pada Gambar 7. Sedangkan tampilan daftar sebuah kategori yang sudah di pilih sebelumnya pada tampilan Halaman Utama yang ditunjukkan pada Gambar 8.



Gambar 7. Tampilan Search

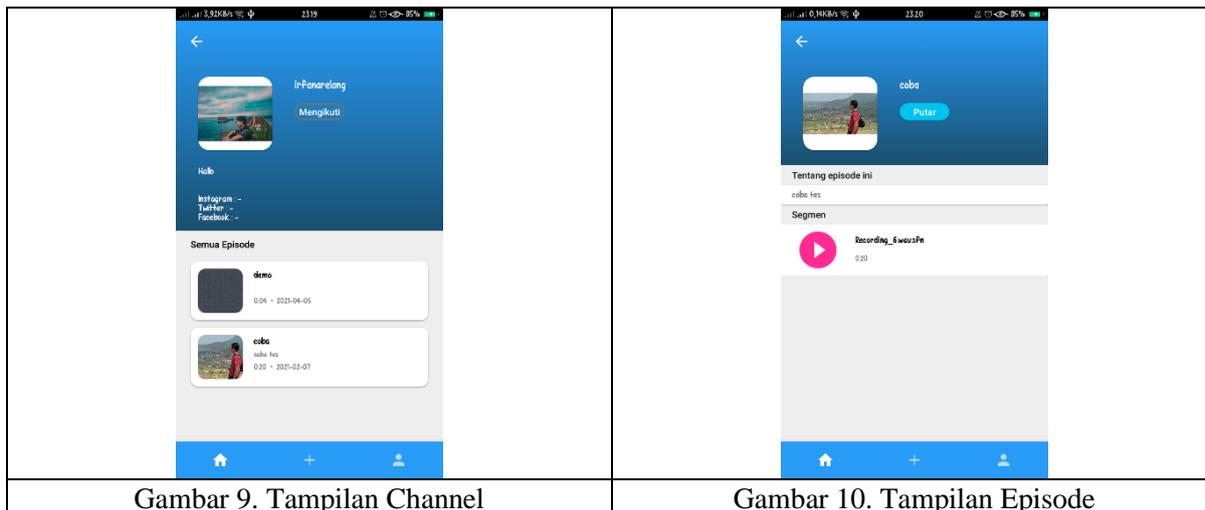


Gambar 8. Tampilan Kategori

3.4 Tampilan Channel dan Episode

Tampilan informasi dari channel yang telah di pilih di halaman utama yang ditunjukkan pada Gambar 9. Informasi yang disampaikan yaitu identitas dari channel tersebut misalkan nama, foto, deskripsi dan semua episode dari channel tersebut. Di halaman ini pengguna juga bisa mengikuti channel dengan menekan tombol ikuti dan tidak mengikuti.

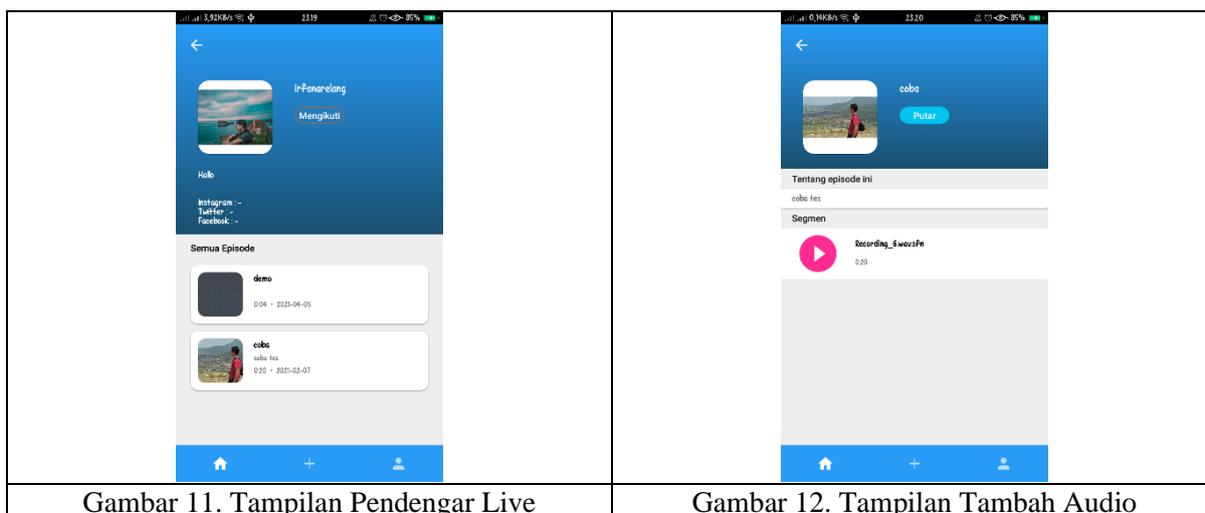
Pada Gambar 10 menampilkan informasi dari episode. Informasi yang ditampilkan yaitu judul episode, foto episode, deskripsi, durasi dari episode tersebut. Untuk memulai podcast bisa di tekan tombol putar. Jika pengguna membuka halaman episodenya sendiri maka pengguna bisa melakukan edit episode dan menghapusnya.



3.5 Tampilan Pendengar Live dan Tambah Audio

Informasi yang ditampilkan berupa judul live dan waktu yang ditunjukkan pada Gambar 11. Audio podcast yang didengarkan sudah dilakukan proses dekompresi. Jika pengguna mengakhiri live, maka ada informasi yang muncul bahwa live sudah berakhir.

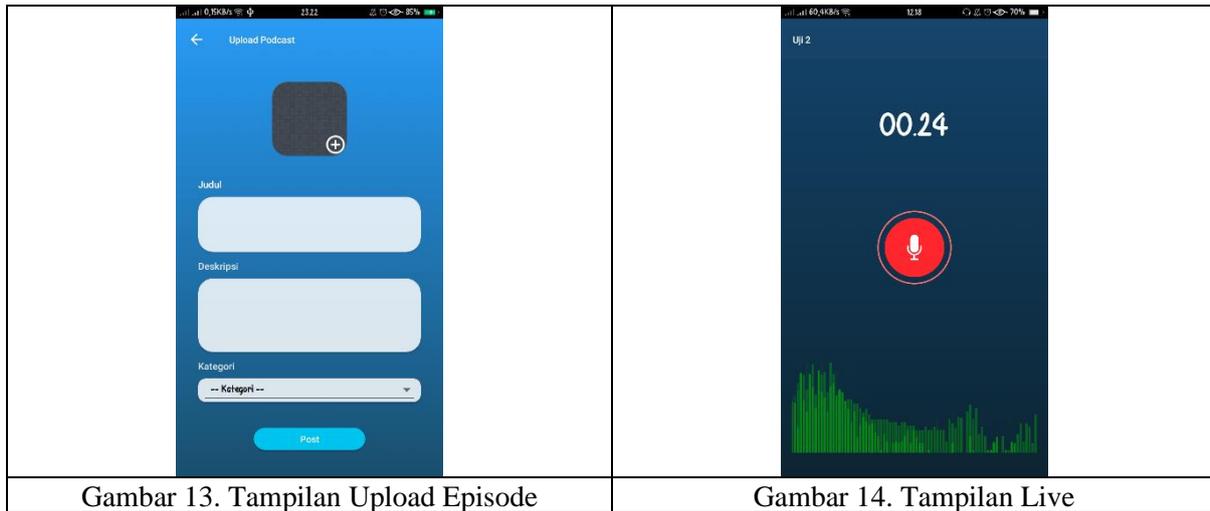
Tampilan halaman untuk menambahkan audio yang akan di upload ke server bisa ditunjukkan pada Gambar 12 . Audio yang bisa di tambahkan yaitu berformat wav kemudian akan di upload ke server tetapi belum menjadi sebagai episode. Jika ingin mengupload audio sebagai episode bisa dengan menekan tombol kirim di pojok kanan atas yang akan di alihkan ke halaman upload untuk mengisi form yang sudah ditentukan.



3.6 Tampilan Upload Episode dan Tampilan Live

Tampilan upload episode berfungsi untuk memasukkan foto, judul, deskripsi dan kategori episode yang ditunjukkan pada Gambar 13. Jika foto tidak dimasukkan maka akan otomatis sesuai yang telah ditentukan.

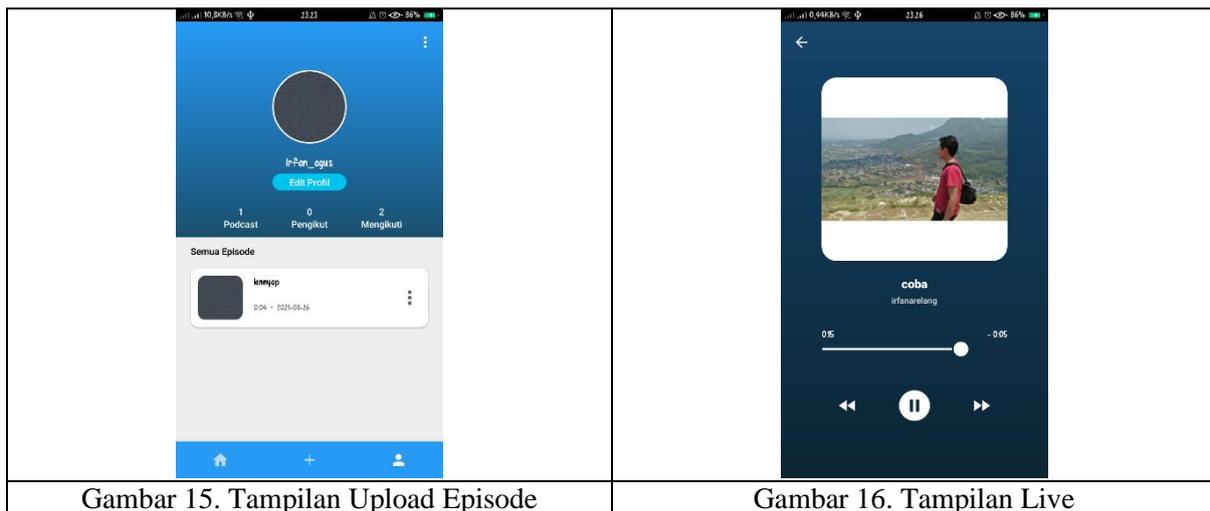
Tampilan halaman live yang sebelumnya pengguna diharuskan untuk mengisi judul dan kategori. Untuk memulai live bisa menekan tombol mikrofon yang berwarna hijau bisa, jika sudah mulai tombol akan berganti warna menjadi merah yang ditunjukkan pada Gambar 14. Jadi tombol warna merah menandakan pengguna masih melakukan live. Di halaman ini stream audio podcast di kompresi terlebih dahulu sebelum di transmisikan ke server. Setelah pengguna mengakhiri live audio akan di upload ke database sebagai episode baru.



3.7 Tampilan Profil dan Pemutar Podcast

Halaman ini menampilkan semua episode pengguna tersebut. Pengguna bisa mengedit profil dengan menekan tombol edit profil akan ditunjukkan ke halaman edit profil. Selain edit profil pengguna juga bisa edit dan hapus episode yang ditunjukkan pada Gambar 15.

Pada Gambar 16 merupakan tampilan pemutaran podcast secara streaming. Selain play dan pause bisa juga melakukan maju dan mundur beberapa menit sesuai keinginan.



3.8 Hasil Pengujian

Hasil Pengujian ini akan dilakukan dengan cara melakukan *Live Streaming* dimana perangkat yang digunakan yaitu 1 Laptop sebagai server, kemudian 2 Smartphone sebagai pengguna *Live Streaming* dan sebagai pendengar. Pengujian ini dilakukan sebanyak 10 kali dengan waktu kurang lebih 15 menit yang akan menghasilkan file kompresi kemudian menjadi parameter pengujian. Untuk pengujian *Quality of Service(QOS)* yaitu berasal dari mencapture lalu lintas data dengan menggunakan aplikasi *Wireshark*.

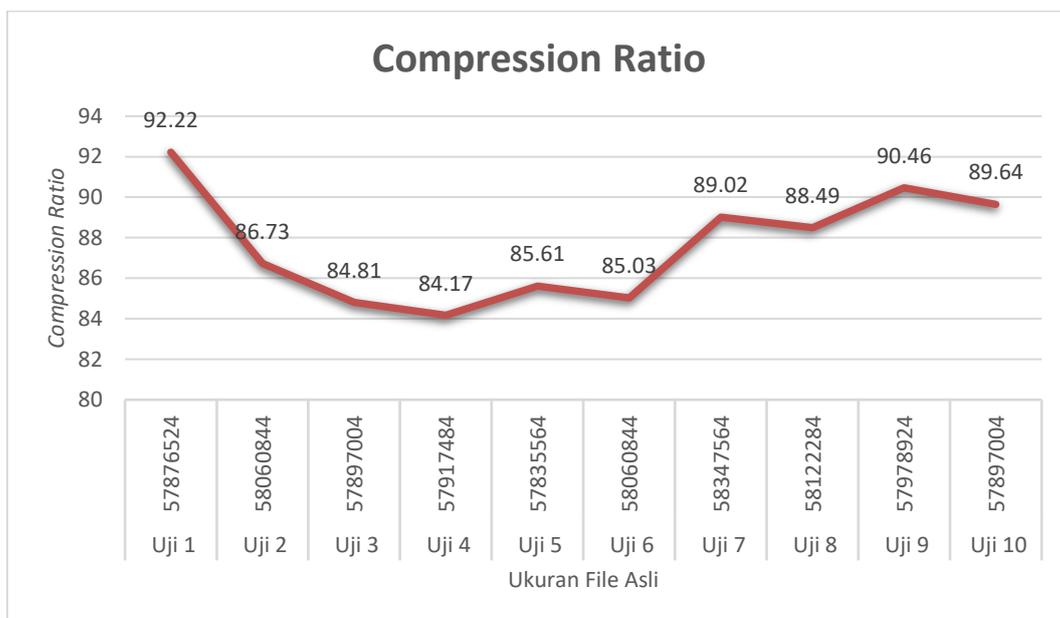
3.8.1 Pengujian *Compression Ratio*

Pengujian ini akan membandingkan ukuran file asli dengan file yang sudah terkompresi yang menghasilkan nilai Rasio Kompresi. File kedua ini dihasilkan dari simulasi live streaming yang dilakukan sebanyak 10 kali dengan ukuran file yang berbeda-beda. Berikut adalah tabel hasil dari nilai *Compression Ratio*.

Tabel 1. Tabel Hasil *Compression Ratio*

No	Nama File	Ukuran File Asli (Byte)	Ukuran File Kompresi (Byte)	Compression Ratio (%)
1	15_11075507042021_audio.wav	57876524	53373650	92,22
2	15_12392507042021_audio.wav	58060844	50357045	86,73
3	15_13075207042021_audio.wav	57897004	53373650	84,81
4	15_13352407042021_audio.wav	57917484	50357045	84,17
5	15_14212207042021_audio.wav	57835564	53373650	85,61
6	15_14522907042021_audio.wav	58060844	50357045	85,03
7	15_15294107042021_audio.wav	58347564	53373650	89,02
8	15_15575807042021_audio.wav	58122284	50357045	88,49
9	15_16302707042021_audio.wav	57978924	53373650	90,46
10	15_16593307042021_audio.wav	57897004	50357045	89,64
Rata-Rata Compression Ratio				87,62

Pada Tabel 1 menghasilkan nilai rata-rata sebesar 87,62 dari pengujian 10 file asli dan sesudah terkompresi. Nilai rasio terkecil yang dihasilkan yaitu 84,17 pada file 15_13352407042021_audio.wav dan nilai rasio terbesar yaitu 92,22 pada file 15_11075507042021_audio.wav. Jadi bisa diambil kesimpulan bahwa semakin kecil nilai rasio kompresi menandakan bahwa banyak data yang hilang sehingga menghemat ruang penyimpanan. Besarnya kecilnya rasio kompresi juga tidak dipengaruhi oleh ukuran file yang ditunjukkan pada Gambar 17.



Gambar 17. Grafik Hasil Compression

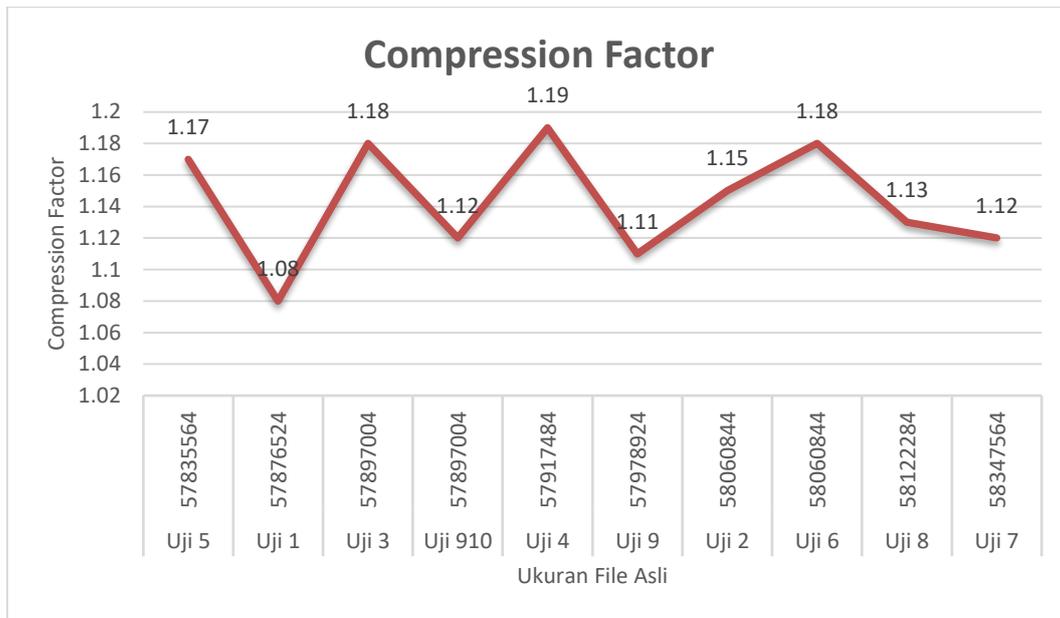
3.8.2 Pengujian *Compression Factor*

Pada pengujian ini dilakukan dengan rumus (2.2), berikut adalah 44able hasil dari nilai *Compression Factor*.

Tabel 2. Hasil *Compression Factor*

No	Nama File	Ukuran File Asli (Byte)	Ukuran File Kompresi (Byte)	<i>Compression Factor</i> (%)
1	15_11075507042021_audio.wav	57876524	53373650	1,08
2	15_12392507042021_audio.wav	58060844	50357045	1,15
3	15_13075207042021_audio.wav	57897004	53373650	1,18
4	15_13352407042021_audio.wav	57917484	50357045	1,19
5	15_14212207042021_audio.wav	57835564	53373650	1,17
6	15_14522907042021_audio.wav	58060844	50357045	1,18
7	15_15294107042021_audio.wav	58347564	53373650	1,12
8	15_15575807042021_audio.wav	58122284	50357045	1,13
9	15_16302707042021_audio.wav	57978924	53373650	1,11
10	15_16593307042021_audio.wav	57897004	50357045	1,12
Rata-Rata <i>Compression Factor</i>				1,14

Pada Tabel 2 menghasilkan nilai rata-rata faktor kompresi sebesar 1,14 dengan nilai tertinggi 1,19 saat pengujian ke-4 dari 10 pengujian. Dalam acuan *Compression Factor* di mana nilai *Compression Factor* semakin tinggi maka semakin baik tingkat kompresinya. Untuk nilai *Compression Factor* dalam pengujian tersebut tidak dipengaruhi oleh ukuran file, dapat dilihat pada Gambar 18 yaitu dalam pengujian ke 1 yang memiliki ukuran 57876524 terkecil dan nilai faktor kompresinya kecil, sedangkan file pengujian ke 4 yang memiliki ukuran 57917484 lebih besar dari ukuran file pengujian ke 1 dan nilai faktor kompresinya besar.



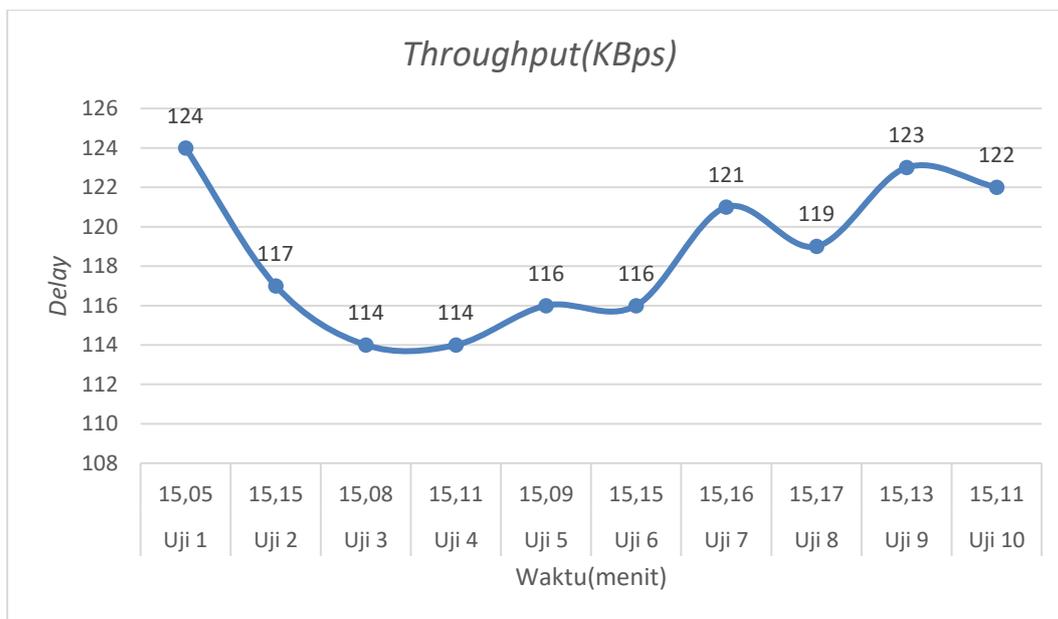
Gambar 18. Grafik Hasil *Compression Factor*

3.8.3 Pengujian *Throughput*

Pada Tabel 3 merupakan hasil dari 10 kali pengujian live stream dalam waktu kurang lebih 15 menit yang memiliki nilai total *throughput* 1186 KBps dan nilai rata-rata 119 KBps. Nilai tertinggi *throughput* saat live streaming pada pengujian ke-1 yaitu memperoleh nilai sebesar 124KBps. Dalam pengujian 1 ke pengujian 4 mengalami penurunan 10 KBps dikarenakan rasio kompresi pada pengujian 3 dan 4 nilainya sangat kecil dibandingkan dengan pengujian lainnya sehingga data yang ditransmisikan lebih kecil yang ditunjukkan pada Gambar 19. Jadi bisa disimpulkan bahwa kompresi data mempengaruhi besar kecilnya *throughput*.

Tabel 3. Hasil *Throughput*

No	Live Streaming ke	Besar Paket (Byte)	Waktu (Menit)	<i>Throughput</i> (KBps)
1	Uji 1	112069506	15,05	124
2	Uji 2	106185304	15,15	117
3	Uji 3	103412287	15,08	114
4	Uji 4	103406551	15,11	114
5	Uji 5	105050833	15,09	116
6	Uji 6	105008294	15,15	116
7	Uji 7	110305962	15,16	121
8	Uji 8	108346641	15,17	119
9	Uji 9	111817937	15,13	123
10	Uji 10	110392022	15,11	122
Total <i>Throughput</i>				1186
Rata-Rata <i>Throughput</i>				119



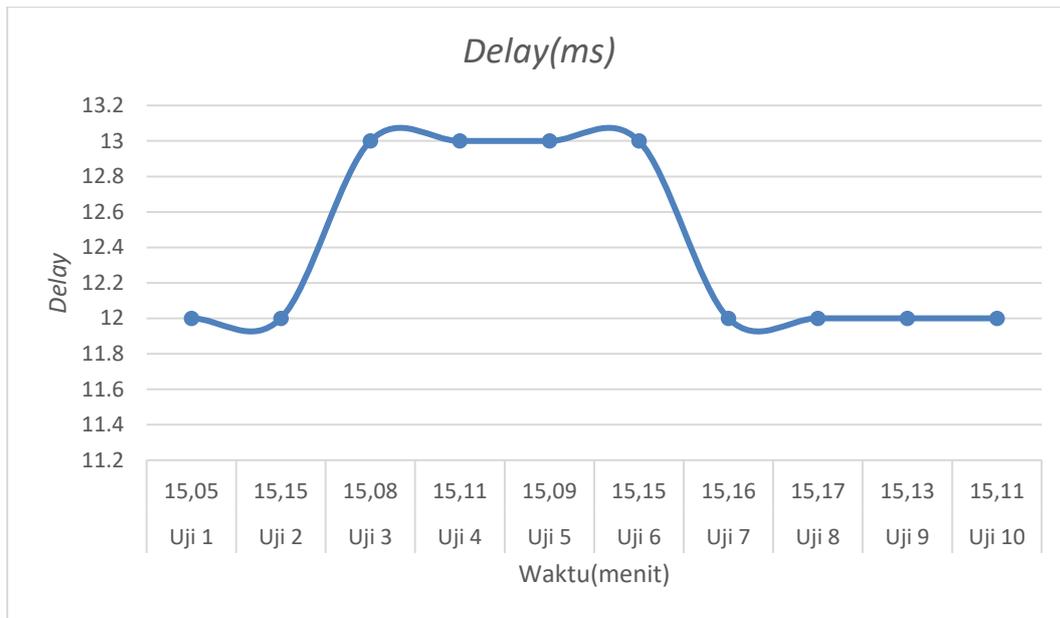
Gambar 19. Grafik Hasil Throughput

3.8.4 Pengujian Delay

Pada Tabel 4 di atas menjelaskan nilai rata-rata delay dari setiap pengujian, dan kemudian menghasilkan total delay dari seluruh pengujian yaitu 123 (ms) dan menghasilkan nilai rata-rata delay dari semua pengujian yaitu 12 (ms). Dari semua pengujian nilai delaynya hampir menyerupai walaupun naik tidak signifikan dengan selisih 1 ms bisa dilihat pada Gambar 20. Nilai rata-rata delay yang dihasilkan menunjukkan index 1 menurut TIPHON[12]. Jadi bisa disimpulkan bahwa kompresi data bisa berpengaruh dalam nilai delay walaupun perubahan yang tidak signifikan dan index yang dihasilkan dari standart TIPHON yaitu “Sangat Bagus”.

Tabel 4. Hasil Delay

No	Live Streaming ke	Jumlah Paket (Byte)	Waktu (Menit)	Delay (ms)
1	Uji 1	76462	15,05	12
2	Uji 2	72990	15,15	12
3	Uji 3	70935	15,08	13
4	Uji 4	70986	15,11	13
5	Uji 5	72020	15,09	13
6	Uji 6	72042	15,15	13
7	Uji 7	75846	15,16	12
8	Uji 8	74291	15,17	12
9	Uji 9	76754	15,13	12
10	Uji 10	75814	15,11	12
Total Delay				123
Rata-Rata Delay				12



Gambar 20. Grafik Hasil Delay

4. KESIMPULAN

4.1 Kesimpulan

Setelah melakukan implementasi program dan pengujian dengan algoritma *Shannon-Fano* dapat ditarik kesimpulan sebagai berikut:

- Hasil *Compression Ratio* pada data audio memiliki rata-rata 87,62% sedangkan *Compression Factor* memiliki rata-rata 1,14% dari 10 pengujian yang telah dilakukan. Dari *Compression Ratio* dan *Compression Factor* didapatkan kesimpulan bahwa tingkat kompresi yang terbaik dari semua pengujian yaitu audio 15_13352407042021_audio.wav dengan hasil *Compression Ratio* 84,17% dan hasil *Compression Factor* sebesar 1,19% dan besarnya file tidak berpengaruh terhadap hasil *Compression Ratio*.
- Hasil *Throughput* pada 10 pengujian menggunakan aplikasi Wireshark dalam waktu kurang lebih 15 menit memiliki rata-rata 119 KBps, sedangkan *Delay* memiliki rata-rata 12 ms. Dari nilai *Throughput* dan *Delay* didapatkan kesimpulan bahwa *delay* berpengaruh terhadap *throughout*, jika nilai *delay* kecil maka hasil nilai *throughput* besar dan jika nilai *delay* besar maka nilai yang dihasilkan *throughput* kecil.

4.2 Saran

Pada hasil penelitian ini penulis menyadari jika masih adanya kekurangan baik penyusunan laporan maupun program yang telah dibuat. Maka penulis mengharapkan kritik dan saran yang membangun dari para pembaca. Adapun saran dari penulis pada implementasi untuk dimasa yang akan mendatang:

- Pada aplikasi podcast ini yang mungkin bisa ditambahkan untuk kompresi data audio yang berbagai macam format.
- Pada fitur aplikasi podcast diharapkan dapat dilakukan pengembangan seperti penambahan fitur notifikasi maupun download file.

REFERENSI

- [1] J. R. Meredith, "The impact of podcasting on perceived learning, classroom

- community, and preferred context for podcast consumption,” *ProQuest Diss. Theses*, no. July, hal. 119, 2013.
- [2] S. Wilson, “Microsoft WAVE soundfile format,” 2003. [Daring]. Tersedia pada: <http://soundfile.sapp.org/doc/WaveFormat/>. [Diakses: 23-Mar-2020].
- [3] D. Salomon, *Data Compression*, 4 ed., vol. 4. London: Springer London, 2007.
- [4] A. K. Bhattacharjee, “Comparison Study of Lossless Data Compression Algorithms for Text Data,” *IOSR J. Comput. Eng.*, vol. 11, no. 6, hal. 15–19, 2013, doi: 10.9790/0661-1161519.
- [5] S. K. Lidya, M. A. Budiman, dan R. F. Rahmat, “Implementasi dan Analisis Kinerja Algoritma Arithmetic Coding dan Shannon-Fano pada Kompresi Citra BMP,” *Snastikom*, 2013.
- [6] M. Syarif, “mengenal-sample-rate-audio,” 2014. [Daring]. Tersedia pada: <http://www.softscients.web.id/2014/10/mengenal-sample-rate-audio.html>. [Diakses: 23-Mar-2020].
- [7] Fathahillah, S. G. Zain, dan Rismawati, “Homogeneous Image Compression Techniques with the Shannon-Fano Algorithm,” *Int. J. Environ. Eng. Educ.*, vol. 1, no. 2, hal. 59–66, 2019, doi: 10.5281/zenodo.3490205.
- [8] K. Sharma dan K. Gupta, “Lossless data compression techniques and their performance,” *Proceeding - IEEE Int. Conf. Comput. Commun. Autom. ICCCA 2017*, vol. 2017-January, hal. 256–261, 2017, doi: 10.1109/CCAA.2017.8229810.
- [9] M. Ricky, F. A. Setyaningsih, dan M. Dipenogoro, “Analisis Kompresi Steganography Pada Citra Digital dengan Menggunakan Metode Least Significant Bit Berbasis Mobile,” *J. Coding*, vol. 06, no. 03, hal. 75–86, 2018.
- [10] Hasanul Fahmi, “Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik,” *J. Teknol. Inf. dan Komun.*, vol. 7, no. 2, hal. 98–105, 2018.
- [11] Y. Prabowo, H. Widiyantara, dan P. Susanto, “Journal of Control and Network Systems,” *JCONES J. Control Netw. Syst.*, vol. 3, no. 2, hal. 9–17, 2014.
- [12] ETSI, “Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS),” *Etsi Tr 101 329 V2.1.1*, vol. 1, hal. 1–37, 1999.