KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)
Volume 6, Nomor 1, November 2022
DOI: 10.30865/komik.v6i1.5764
ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

Perancangan Website Video Streaming Dengan Memanfaatkan Algoritma Rice Codes Dalam Memaksimalkan Space Penyimpanan Server

Wiyuda Pratama Mahardika

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universtias Budi Darma Medan, Indonesia
Email: wiyudapratama310@gmail.com

Abstrak−Website saat ini sudah berkembang pesat, semakin banyak pula kini yang bisa dilakukan oleh pengguna di website, tidak hanya untuk mencari sebuah informasi atau bertukar informasi. Saat ini fungsi dari website sudah berkembang jauh lebih baik, bahkan saat ini website bisa dijadikan sebagai layanan video streaming dimana pengguna dapat melihat video yang mereka sukai di website. Namun, ukuran video yang cukup besar bisa menjadi penghalang atau penyebab dari membengkaknya biaya penggunaan penyimpanan pada server. Karena semakin besar ukuran video yang di upload maka akan semakin banyak pula penyimpanan yang di perlukan oleh server untuk menyimpan video tersebut. Kompresi merupakan salah satu teknik yang bisa digunakan untuk mengurangi ukuran dari suatu data dengan cara menghilangkan sejumlah bit yang terdapat pada suatu file. Untuk itu, dalam mengatasi ukuran file video yang relative cukup besar solusi ini dapat digunakan untuk mengatasi permasalahan tersebut. Algoritma rice codes merupakan salah satu dari algoritma kompresi yang dapat digunakan untuk mengurangi ukuran dari suatu data dan cukup efektif dalam mengkompresi file video. Dalam pengimplementasiannya, memanfaatkan teknik kompresi dengan menggunakan algoritma rice codes dalam mengompresi file video dinilai sangat efektif karena dapat menghasilkan nilai dari compression ratio (CR) sebesar 56,875%. Tentunya dengan hasil yang didapat tersebut file video yang ukurannya relative cukup besar dapat berkurang setengahnya dari ukuran aslinya.

Kata Kunci: Website; Video; Kompresi; Rice Codes

Abstract−The website is currently growing rapidly, now there is more that users can do on the website, not only to search for information or exchange information. Currently the function of the website has developed much better, even now the website can be used as a video streaming service where users can view the videos they like on the website. However, the size of the video that is large enough can be a barrier or a cause of increasing the cost of using storage on the server. Because the larger the size of the uploaded video, the more storage the server will need to store the video. Compression is a technique that can be used to reduce the size of a data by removing the number of bits contained in a file. For this reason, in dealing with relatively large video file sizes, this solution can be used to overcome this problem. The rice codes algorithm is one of the compression algorithms that can be used to reduce the size of data and is quite effective in compressing video files. In its implementation, utilizing compression techniques using the rice codes algorithm in compressing video files is considered very effective because it can produce a compression ratio (CR) value of 56.875%. Of course, with the results obtained, video files that are relatively large in size can be reduced by half from their original size.

***Keywords**:* Website; Video; Compression; Rice Codes

1. PENDAHULUAN

Saat ini website merupakan teknologi yang sudah sangat populer dalam menyebarluaskan informasi. Banyak informasi yang bisa didapatkan dengan menggunakan website, seperti berita, ilmu pengetahuan, *video streaming* dan lain sebagainya[1]. *Video streaming* merupakan suatu website yang digunakan untuk menonton film kesukaan masing-masing, mulai dari drama korea, anime, *box office* dan lain sebagainya. Hampir dari semua orang selalu menonton *video* setiap harinya, karena selain dapat menghibur, *video* juga dapat menghilangkan rasa bosan bagi penikmatnya.

Semakin populernya website, maka perlu dikembangkan juga teknologi pendukung untuk bisa mendapatkan performa yang baik. Server yang baik tentunya akan memberikan performa yang baik juga bagi pengguna dalam mengakses sebuah website. Tidak hanya itu, pada sisi *developer* juga harus memperhatikan setiap rincian program yang di buat, agar saat program dari website tersebut di *apply* pada sisi *server* tidak terlalu memakan banyak *space* penyimpanan, karena hal ini dapat memperngaruhi dalam performa website itu sendiri.

Penggunaan *video* dengan ukuran yang cukup besar dalam website tentunya juga akan berpengaruh besar dalam performa website. Semakin besar *file video* yang di *upload* pada website maka akan membuat *load* website dari *server* semakin lama dan akan memakan banyak *space* penyimpanan pada *server*, maka dari itu *developer* juga harus memperhatikan hal ini. Untuk bisa mengatasi hal tersebut, sebelum mengupload *video* pada website, terlebih dahulu sebaiknya *video* tersebut di kompresi agar ukuran file *video* yang digunakan bisa lebih kecil dari pada yang aslinya.

Kompresi merupakan proses mengubah ukuran dari sebuah data untuk menghasilkan representasi digital yang padat namun tetap dapat mewakili kualitas informasi yang terdapat pada data tersebut. Algoritma *Rice Codes* merupakah salah satu algoritma dari kompresi yang bisa digunakan untuk melakukan kompresi terhadap *file video*[2]–[4]. Algoritma ini menggunakan sistem yang didapat dari teknik *Golomb Coding*, dimana dapat menghasilkan *prefix* yang lebih mudah. Saat ini, penelitian terkait algoritma *rice codes* dalam proses kompresi *file video* masih belum terlalu banyak, hal ini lah yang membuat penulis tertarik untuk mengangkat algoritma tersebut untuk diteliti dan melihat seberapa efektif algoritma ini dalam mengkompresi *file video*.

Kiki Ramayani pada tahun 2021 pernah melakukan penelitian terkait kompresi terhadap *file video* dengan menggunakan algoritma *rice codes*, dalam penelitiannya tersebut dia menyimpulkan bahwa dengan menggunakan

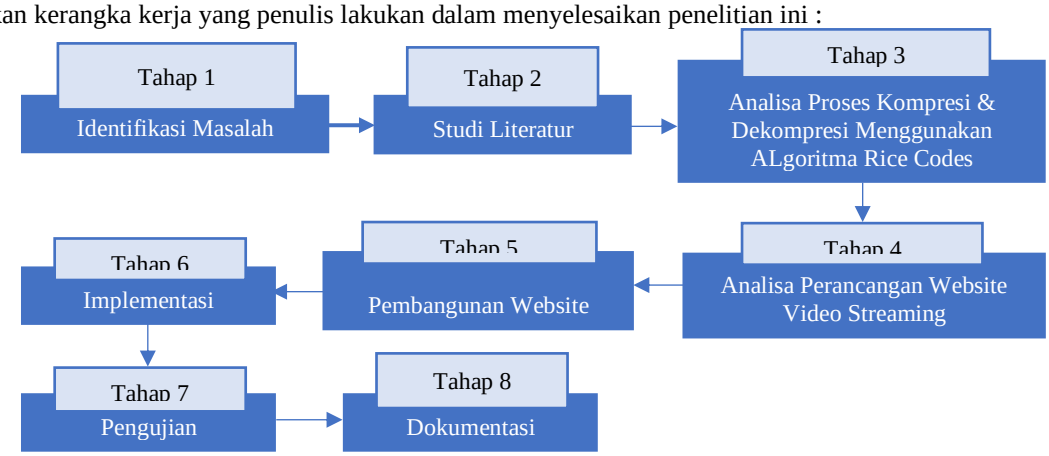
ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

algoritma *rice codes* mampu mengkompresi *file video* dengan baik dengan mendapatkan nilai dari *compression ratio* sebesar 64%, yang dimana tidak dapat menggembalikan isi *file video* aslinya Ketika dilakukan proses dekompresi[2]. Penelitian selanjutnya dilakukan oleh Putri Fitria pada tahun 2020 pernah melakukan penelitian terkait kompresi terhadap *file* gambar dengan menggunakan algoritma *rice codes*, dalam penelitiannya tersebut dia menyimpulkan bahwa dengan menerapkan algoritma *rice codes* mampu memkompresi *file* gambar yang bereksensi jpeg dengan sangat bagus dan tanpa kekurangan isi file aslinya. Pada penelitiannya juga menghasilkan *Ratio Compresion* yaitu sebesar 1,97 dan *Compresion Ratio* sebesar 50,9%[5]. Irwansyah dkk pada tahun 2018 pernah melakukan penelitian terkait kompresi terhadap *file* teks dengan menggunakan algoritma *rice codes*, dalam penelitiannya tersebut dia menyimpulkan bahwa tidak hanya dapat mengompresi file gambar algoritma *rice codes* ternyata juga dapat digunakan untuk melakukan kompresi terhadap *file* teks[6]. Selanjutnya penelitian yang dilakukan oleh Vity Ernanda dkk pada tahun 2020 pernah melakukan penelitian terkait kompresi terhadap teks sekaligus pengamanan pada aplikasi *chatting* dengan menggunakan algoritma *rice codes* sebagai algoritma kompresi dan algoritm *Rivest Shamir Adleman* (RSA), dalam penelitiannya tersebut dia menyimpulkan bahwa dengan menerapakan proses kompresi dan pengamanan pesan teks pada aplikasi *chatting* dapan membuat *space* penyimpanan lebih sedikit dan akan lebih aman[7].

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Pada metodologi penelitian akan jabarkan mengenai tahapan-tahapan atau kerangka kerja penelitian yang akan dilakukan oleh penulis dalam menyelesaikan penelitian. Kerangka kerja penelitian terdiri dari beberapa tahapan yang terhubung secara sistematis, yang bertujuan untuk membantu penulis dalam menyelesaikan penelitian. Berikut ini merupakan kerangka kerja yang penulis lakukan dalam menyelesaikan penelitian ini :



Gambar 1. Kerangka Kerja Penelitian

Dari diagram di atas, dapat penulis uraikan satu persatu dari tiap tahapan yang terdapat dalam kerangka kerja penelitian di atas, sebagai berikut:

1. Identefikasi Masalah
   Pada tahap ini dimana penulis mencari sebuah masalah disekitar penulis, yang mana masalah tersebut sering merepotkan bagi penulis dan orang lain, serta menguraikan inti dari sumber masalah tersebut, sehingga bisa penulis teliti dan mencari solusi terbaik dalam menyelesaikan masalah tersebut.
2. Studi Literatur
   Pada tahap ini penulis melakukan pemahaman lebih terkait space penyimpanan server website, serta kompresi guna untuk memperkecil suatu file video, dengan melalui sumber seperti buku, dan juga jurnal.
3. Analisa Proses Kompresi Dan Dekompresi Menggunakan Algoritma Rice Codes
   Pada tahap ini penulis mencoba menganalisa terkait proses kompresi dan dekompresi pada file video dengan menggunakan algoritma rice codes. Yang mana bertujuan untuk menentukan langkah demi langkah untuk menyelesaikan proses kompresi maupun proses dekompresi dengan menggunakan algoritma rice codes.
4. Analisa Pembangunan Website video streaming
   Pada tahap ini penulis menganalisa kebutuhan dari pembangunan website video streaming, seperti bahasa pemrograman yang digunakan, maupun user interface (UI) dari website video streaming yang akan dibangun.
5. Perancangan Website
   Pada tahap ini penulis melakukan perancangan user interface (UI) yang sudah penulis analisa dan penulis melakukan slicing dari ui tersebut dengan menggunakan bahasa pemrograman yang sudah penulis tentukan.
6. Implementasi
   Pada tahap ini penulis melakukan implementasi pada website video streaming yang sudah penulis rancang dengan mengimplementasikan kompresi file video pada website tersebut agar masalah diawal bisa teratasi.
7. Pengujian

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

Pada tahap ini penulis melakukan pengujian pada website video streaming yang sudah dibangun, serta sudah diimplementasikan kompresi file video pada website tersebut sehingga file video yang digunkan bisa lebih kecil ukurannya.

8. Dokumentasi
   Pada tahap ini merupakan proses akhir yang penulis lakukan pada penelitian kali ini yang dibuat dalam bentuk laporan skirpsi sehingga mudah dibaca dan dipahami, serta dikembangkan kembali.

2.2 Kompresi

Kompresi merupakan proses mengubah ukuran dari sebuah data untuk menghasilkan representasi digital yang padat namun tetap dapat mewakili kualitas informasi yang terdapat pada data tersebut[8]–[10]. Berdasarkan kandungan informasi sifat kompresi dapat dibedakan menjadi dua kelompok yaiut :

1. *Lossless Compression*
   Pada jenis kompresi ini informasi yang terdapat pada *file* hasil kompresi sama dengan informasi pada *file* aslinya, sehingga *file* hasil kompresi dapat dikembalikan secara sempurna seperti *file* awalnya. Oleh karena itu metode ini sering juga disebut sebagai sebagai *error free compression.*
2. *Lossy Compression*
   Pada jenis kompresi ini mengijinkan terjadinya kehilangan data tertentu dari file yang dikompresi, sehingga dapat menghasilkan rasio kompresi yang lebih tinggi, dan apabila *file* hasil kompresi direkonstruksi kembali atau didekompresi kembali maka hasilnya tidak akan sama dari *file* aslinya, tetapi informasi yang terkandung didalamnya tidak sampai berubah atau hilang[11].

Untuk menentukan kualitas dari hasil kompresi, terdapat beberapa factor yang dapat digunakan sebagai acuan penilaian, diantaraya :

1. *Ratio Compression* (RC)
   Ratio Compression (RC) merupakan nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi. Untuk menghitung *ratio compression* dapat menggunakan rumus seperti dibawah
   $$RC = \frac{Ukuran\ Data\ Sebelum\ Dikompresi}{Ukuran\ Data\ Setelah\ Dikompresi} \quad (1)$$
2. *Compression Ratio* (CR)
   Compression Ratio (CR) merupakan presentasi perbandingan antara data yang sudah dikompresi dengan data sebelum dikompresi. Untuk menghitung *compression ratio* dapat menggunakan rumus seperti dibawah :
   $$CR = \frac{Ukuran\ Data\ Setelah\ Dikompresi}{Ukuran\ Data\ Sebelum\ Dikompresi} \quad (2)$$

Setelah suatu *file* telah berhasil dikompresi, maka *file* hasil kompresi tersebut perlu dikembalikan seperti *file* aslinya, untuk kemudian dapat dibaca kembali. Proses ini biasa disebut sebagai proses dekompresi, yang mana pada proses ini akan dilakukan proses mengembalikan sebuah data yang telah berhasil dikompresi. Sama seperti pada saat melakukan kompresi, pada proses dekompresi juga terbagi menjadi dua jenis yaitu, *lossy* dan *lossless*[12].

2.3 File Video

Video merupakan suatu objek yang merepresentasikan data yang telah dioleh, yang merupakan gabungan antara gambar serta suara. *Video* juga bisa diartikan sebagai serangkaian gambar yang ditampilkan secara berutan. Menurut Cecep Kustandi mengunkapkan bahwa *video* adalah alat yang dapat menyajikan informasi, memaparkan proses, menjelaskan berbagai proses yang rumit, mengajarkan keahlian serta menyingkat ataupun memperlambat waktu dan mempengaruhi sikap[13].

2.4 Algoritma Rice Codes

Algoritma *rice codes* atau biasa disebut juga *Golomb-Rice Codes* ditemukan oleh Robert F.Rice yang menggunakan *subset* dari turunan *Golomb Codes* untuk menghasilkan sebuah perhitungan yang lebih sederhana tetapi mungkin *suboptional* dari kode awalan itu sendiri[14],[12]. Bisa dilihat dari gambar di atas bahwa algoritma *rice codes* ini dapat dianggap sebagai kode yang menunjukan posisi (q) dan *offset* (r). Sehingga posisi q dan r *offset* untuk pengkodean *integer N* pada *rice codes* yang menggunakan parameter *M*.

$$q = \left[\frac{(x-1}{M}\right]. \quad (3)$$

Dan nilai *r* dengan bit *b*.

$$r = x - qM - 1. \quad (4)$$

Hasilnya

$$(q + 1)r. \quad (5)$$

Perlu diperhatikan bahwa *r* didapat dari berbagai jumlah bit, dan secara khusus bit *b* hanya untuk *rice codes*, dan menghubungkan antara b-1 dan b bit untuk kode (M yaitu bukan kelipatan 2).

$$b = [\log_2(M)] \quad (6)$$

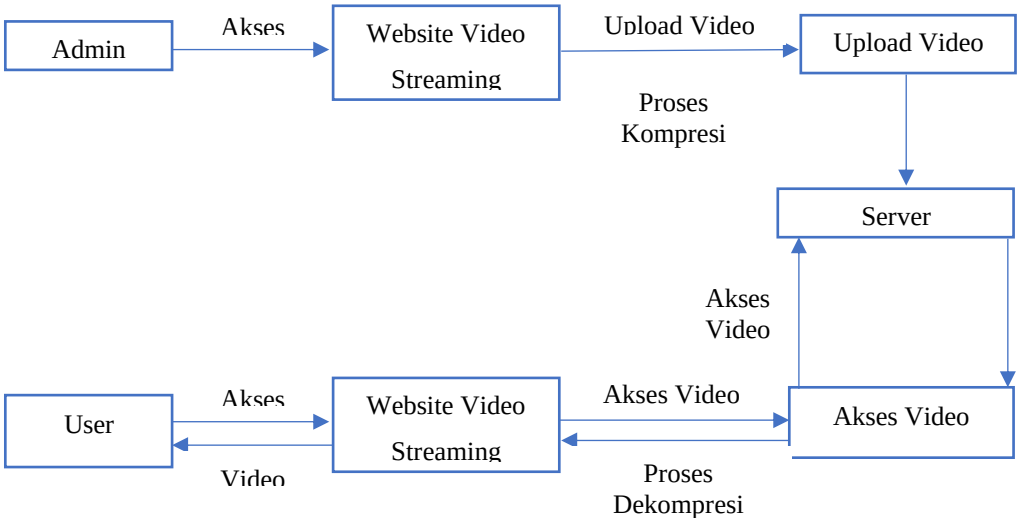
Jika $0 \leq r < 2^b - M$ gunakan b – 1 bit untuk mengkodekan *r*
Jika $2^b - M \leq r < M$ gunakan b bit untuk mengkodekan *r*
$b = \log 2\ (M)$ jika *M* adalah hasil kedua dan dapat dikodekan semua

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

3. HASIL DAN PEMBAHASAN

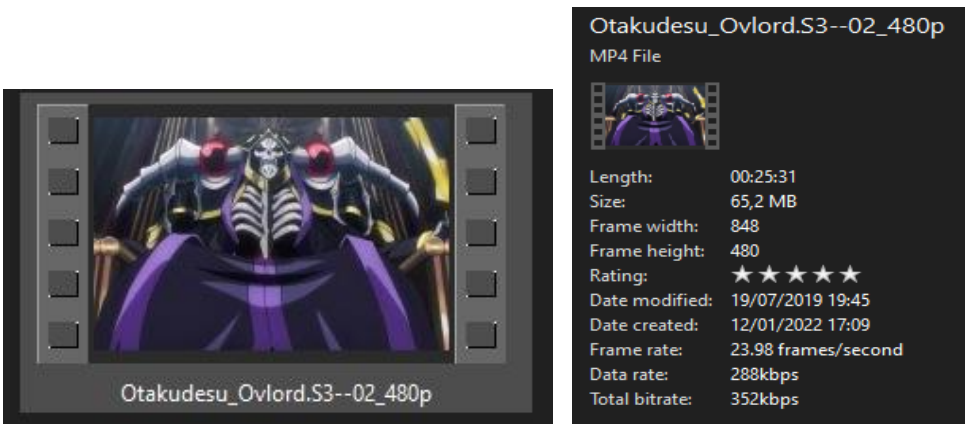
Pada bagian ini analisa merupakan perhitungan sekaligus perancangan website video streaming yang didalamnya menerapkan proses kompresi terhadap file video dengan menggunakan algoritma rice codes. Dengan menerapakan algoritma rice codes yang merupakan tipe kompresi lossless pada website video streaming, yang dimana diharapkan setelah diterapakannya algoritma tersebut dapat mengurangi borosnya penggunakan space penyimpanan server dari website video streaming, sehingga dapat menghemat biaya penyimpanan server. Bagaimana penerapan kompresi terhadap file video pada website video streaming?. Untuk penerapannya sendiri bagaimana proses kompresi itu akan terjadi pada website video streaming adalah dimulai dari seorang admin yang akan mengupload video baru pada website. Sebelum video tersebut akan terupload dan tersimpan pada server terlebih dahulu video tersebut akan dilakukan proses kompresi agar ukuran dari file video menjadi lebih kecil, dan setelah berhasil terkompresi barulah video tersebut akan terupload ke server.

Sedangkan saat seorang user akan mengakses video yang dicarinya dan kemudian user tersebut menonton video yang diinginkannya atau mendownload video tersebut. Hal yang akan terjadi adalah saat terjadinya permintaan dari sisi user pada website video streaming agar menampilkan video yang diinginkannya, maka alurnya adalah user mencari video yang diinginkannya pada website video streaming dan website video streaming akan melakukan permintaan kepada server untuk memberikan video yang dicari oleh user tersebut, tetapi sebelum video yang tersimpan pada server di tampilkan pada website video streaming terlebih dahulu video tersebut akan terdekompresi dan setelah berhasil barulah video tersebut akan ditampilkan pada website video streaming untuk di lihat oleh user. Berikut gambaran bagaimana video akan terkompresi sebelum terupload pada server dan bagaimana video akan terdekompresi sebelum ditampilkan pada website video streaming :



Gambar 2. Analisa Cara Kerja Kompresi Dan Dekompresi Pada Website Video Streaming

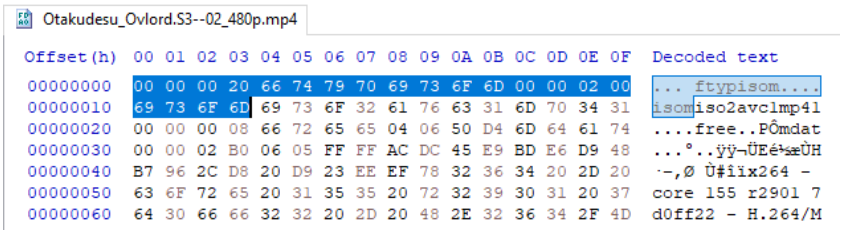
3.1 Penerapan Algoritma Rice Codes



Gambar 3. Sampel File Video

Dari sampel *file video* di atas, berikutnya penulis akan mencari nilai *hexadecimal* dari *file video* tersebut dengan menggunakan *software HxD*. Berikut adalah nilai *hexadecimal* dari *file video* di atas :

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410



Gambar 4. Nilai Hexadecimal dari Fole Video

Dari gambar di atas maka penulis akan mengambil 20 nilai *hexadecimal* yang akan penulis jadikan sebagai sampel perhitungan kompresi dan dekompresi menggunakan algoritma *rice codes*. Berikut adalah tabel nila *hexadecimal* yang akan penulis jadikan sebagai sampel perhitungan :

Tabel 1. Sampel Data Nilai *Hexadecimal*

00	00	00	20	66	74	79	70	69	73
6F	6D	00	00	02	00	69	73	6F	6D

Berikutnya setelah didapat sampel nilai *hexadecimal* yang akan diproses untuk perhitungan kompresi secara manual, selanjutnya adalah mengurutkan nilai *hexadecimal* tersebut berdasarkan frekuensi yang paling banyak muncul. Dibawah ini adalah pengurutan nilai *hexadecimal* berdasarkan frekuensinya :

Tabel 2. Pengurutan Nilai *Hexadecimal* Berdasarkan Frekuensi

N	*Hexadecimal*	Frekuensi	Biner	Bit	Bit x Frek
0	00	6	0000 0000	8	48
1	6D	2	0110 1101	8	16
2	6F	2	0110 1111	8	16
3	69	2	0110 1001	8	16
4	73	2	0111 0011	8	16
5	20	1	0010 0000	8	8
6	66	1	0110 0110	8	8
7	74	1	0111 0100	8	8
8	79	1	0111 1001	8	8
9	70	1	0111 0000	8	8
10	02	1	0000 0010	8	8
	Total	20			160

Berikutnya, setelah nilai *hexadecimal* telah diurutkan berdasarkan frekuensinya dan juga telah didapatkan nilai *biner* nya, selanjutnya adalah mengganti nilai *biner* dengan *codeword* yang sudah dicari sebelumnya sesuai dengan ketentuan dari algoritma *rice codes*. Dibawah ini adalah hasil dari penggantian nilai *biner* dengan *codeword* nya :

Tabel 3. Penggantian Nilai *Biner* Dengan *Codeword*

N	*Hexadecimal*	Frekuensi	*Codeword*	Bit	Bit x Frek
0	00	6	0000	4	24
1	6D	2	0001	4	8
2	6F	2	0010	4	8
3	69	2	0011	4	8
4	73	2	01000	5	10
5	20	1	01001	5	5
6	66	1	01010	5	5
7	74	1	01011	5	5
8	79	1	011000	6	6
9	70	1	011001	6	6
10	02	1	011010	6	6
	Total	20			91

Setelah nilai biner diganti dengan *codeword* dari algoritma *rice codes*, maka selanjutnya adalah menyusun kembali seluruh sampel dari nilai *hexadecimal* dan juga sesuaikan dengan *codeword* nya. Berikut ini adalah hasil dari penyusunan kembali nilai *hexadecimal* dan *codeword* nya :

Tabel 4. Penyusunan Kembali Nilai *Hexadecimal* dan *Codeword*

00	00	00	20	66
0000	0000	0000	01001	01010

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

74	79	70	69	73
01011	011000	011001	0011	01000
6F	6D	00	00	02
0010	0001	0000	0000	011010
00	69	73	6F	6D
0000	0011	01000	0010	0001

Berdasarkan tabel di atas, selanjutnya adalah penyusun kembali *string bit* dari hasil *codeword* pada proses kompresi dan *string bit* tersebut dibagi menjadi perdelapan (8) *bit* yang menjadi sebagai berikut :
"00000000 00000100 10101001 01101100 00110010 01101000 00100001 00000000 01101000 00001101 00000100 001"

Setelah terbentuk *string bit* seperti seperti di atas dan telah dibagi menjadi perdelapan (8) *bit*, dengan total jumlah bit yaitu 91 (sembilan puluh satu). Selanjutnya jika dilihat bahwa 91 (sembilan puluh satu) tidak habis dibagi dengan 8 (delapan) dimana terdapat 3 bit yang tersisa, maka pada kasus ini kita perlu menambahkan yang namanya *padding* dan *flagging*, agar nantinya dengan ditambahkan *padding* dan *flagging* tersebut jumlah bit akan bertambah dan jika dibagi 8 (delapan) maka total *bit* akan habis. Berikut ini adalah cara menambahkan *padding* dan *flagging* pada *string bit* :

1. *Padding*
   Untuk menambahkan *padding* pada suatu *string bit* dapat menggunakan rumus sebagai berikut ;
   $(7 - n) + "1"$
   Dimana, n disini adalah hasil dari sisa bagi *string bit* sebelumnya yaitu 86 % 8 : 3. Dan 1 disini adalah nilai *biner*. Sehingga jika dimasukkan kedalam rumus yaitu :
   $(7 - n) + "1" = (7 - 3) + "1" = 4 + "1" = 00001$
   Jadi *padding* yang akan kita tambahkan pada *string bit* adalah "00001" *bit*
2. *Flagging*
   Untuk menambahkan *flagging* pada suatu *string bit* dapat menggunakan rumus sebagai berikut :
   $9 - n$
   Sama seperti *padding*, n merupakan sisa dari hasil bagi *string bit* sebelumnya yaitu 3. Sehingga jika dimasukkan kedalam rumus adalah sebagai berikut ;
   $9 - n = 9 - 3 = 6 =$ "00000110"
   Jadi *flagging* yang akan kita tambahkan pada *string bit* adalah *biner* dari 6 yaitu "00000110"

Selanjutnya setelah nilai *padding* dan *flagging* telah di dapat, berikutnya tambahkan nilai tersebut pada *string bit* sebelumnya. Sehingga setelah ditambahkan *padding* dan *flagging* maka *string bit* nya akan menjadi seperti berikut :
"00000000 00000100 10101001 01101100 00110010 01101000 00100001 00000000 01101000 00001101 00000100 001**00001 00000110**"

Langkah terakhir dalam proses kompresi adalah mengubah setiap blok pada *string bit* kedalam nilai *hexadecimal* dan setelah itu mengubahnya kembali menjadi suatu karakter tertentu sesuai dengan tabel ASCII. Adapun karakter yang dihasilkan setelah proses kompresi selesai adalah sebagai berikut :

Tabel 5. Karakter Hasil Kompresi

No	*Biner*	*Hexadecimal*	Karakter
1	00000000	0	0
2	00000100	4	EOT
3	10101001	A9	®
4	01101100	6C	L
5	00110010	32	2
6	01101000	68	I
7	00100001	21	!
8	00000000	0	0
9	01101000	68	L
10	00001101	D	CR
11	00000100	4	EOT
12	001**00001**	21	!
13	**00000110**	6	6

Berikut adalah parameter yang akan penulis gunakan untuk mengecek seberapa efektif algoritma *rice codes* ini:

1. *Ratio Compression* (RC)
   $RC = \frac{ukuran\ data\ sebelum\ dikompresi}{ukuran\ data\ setelah\ dikompresi}$
   $RC = \frac{160}{91}$
   $RC = 1.7582$
2. *Compression Ratio* (CR)

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 404-410

$$CR = \frac{ukuran\ data\ setelah\ dikompresi}{ukuran\ data\ sebelum\ di\ kompresi} \times 100\%$$
$$CR = \frac{91}{160} \times 100\%$$
$$CR = 56.875\%$$

Dari perhitungan di atas bisa disimpulkan bahwa algoritma *rice codes* cukup efeketif untuk mengkompresi *file video* dengan menghasilkan persentasi hasil kompresi sebesar 56.875% .

4. KESIMPULAN

Berdasarkan hasil akhir dari penelitian yang penulis lakukan, penulis simpulkan bahwa dengan menggunakan algoritma rice codes dalam mengompresi file video dapat dilakuan dengan baik sesuai dengan prosedurnya, hingga mendapatkan hasil yang jauh lebih baik dalam mengurangi ukuran file video aslinya dengan mendapatkan nilai dari compression ratio sebesar 56,875%. Dalam penerapannya pada website video streaming dengan menggunakan teknik kompresi pada website tersebut dalam mengompresi file video telah berhasil dilakukan. Hasil dari perancangan website video streaming yaitu pada sisi server, digunakan sebagai tempat penyimpanan video yang di upload telah berhasil meminimalisir penggunaanya, karena pada website tersebut telah diterapkan teknik kompresi dengan menggunakan algoritma rice codes.

REFERENCES

[1] W. P. Mahardika, A. Saputra, H. F. W. Halawa, M. Kaja, M. Ndruru, and S. Aripin, "Perancangan Website Blog Anime Dengan Menerapkan Algoritma Rice Codes Untuk Mengkompresi File Gambar," in *Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika (SNESTIK)*, 2022, vol. 1, no. 1, pp. 381–386.

[2] K. Ramayani, "Penerapan Algoritma Rice Codes Untuk Mengkompresi File Video," vol. 5, pp. 186–192, 2021, doi: 10.30865/komik.v5i1.3670.

[3] S. M. Panjaitan, S. D. Nasution, and ..., "Penerapan Algoritma Gopala-Hemachandra Code2 (GH-2 (n)) Pada Kompresi File Audio," *KOMIK (Konferensi …*, vol. 4, pp. 170–177, 2020, doi: 10.30865/komik.v4i1.2676.

[4] C. T. Utari, "Implementasi Algoritma Run Length Encoding Untuk Perancanganaplikasi Kompresi Dan Dekompresi File Citra," *J. Times*, vol. 5, no. 2, pp. 24–31, 2016.

[5] P. Fitria, "Penerapan Algoritma Rice Codes Pada Aplikasi Kompresi File Gambar," vol. 1, no. 3, pp. 158–165, 2020.

[6] I. Irwansyah, S. D. Nasution, and F. Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algortima Rice Codes," *Pelita Inform. Inf. dan Inform.*, vol. 7, no. 2, pp. 219–222, 2018.

[7] V. Ernanda, S. D. Nasution, and F. T. Waruwu, "Penerapan Algoritma Rice Codes dan Algoritma Rivest Shamir Adleman (RSA) Untuk Kompresi dan Pengamanan Teks Pada Aplikasi Chatting," *J. Comput. Syst. Informatics*, vol. 1, no. 3, pp. 92–98, 2020.

[8] R. D. Pratiwi, S. D. Nasution, and F. Fadlina, "Perancangan Aplikasi Kompresi File Teks Dengan Menerapkan Algortima Fixed Length Binary Encoding (Flbe)," *J. Media Inform. Budidarma*, vol. 2, no. 1, 2018.

[9] R. Y. Tanjung and M. Mesran, "Perancangan Aplikasi Kompresi File Dokumen Menggunakan Algoritma Adiitive Code," *JURIKOM (Jurnal Ris. Komputer)*, vol. 8, no. 4, pp. 108–113, 2021.

[10] S. R. Saragih and D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, 2020.

[11] D. Iqbal, "Implementasi Algoritma Levenstein Untuk Kompresi File Video Pada Aplikasi Chatting Berbasis Android," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 266–273, 2019, doi: 10.30865/komik.v3i1.1601.

[12] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: C.V ANDI OFFSET, 2010.

[13] S. Aripin and M. Syahrizal, "Pengaman File Video Menggunakan Algoritma Merkle Hellman Knapsack," *J. Media Inform. Budidarma*, vol. 4, no. 2, p. 461, 2020, doi: 10.30865/mib.v4i2.2039.

[14] O. K. L. Ginting, "IMPLEMENTASI ALGORITMA GOLOMB-RICE CODING UNTUK KOMPRESI FILE CITRA BERBASIS ANDROID," 2017.

[15] M. A. Latif, S. D. Nasution, and P. Pristiwanto, "Analisa Perbandingan Algoritma Rice Codes Dengan Algoritma Goldbach Codes Pada Kompresi File Text Menggunakan Metode Exponential," *Inf. dan Teknol. Ilm.*, vol. 5, no. 2, pp. 112–117, 2018.