KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)
Volume 6, Nomor 1, November 2022
DOI: 10.30865/komik.v6i1.5718
ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

Analisis Perbandingan Algoritma Discrete Cosine Transform (DCT) Dan Run Length Encoding Pada Kompresi File Mp4

Ananda Mulia

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: nanda.mulia20@gmail.com

Abstrak-Perkembangan teknologi salah satu aspek penting dalam dunia teknologi informasi. Banyak algoritma telah dikembangkan untuk mengkompres file, termasuk algoritma Discrete Cosine Transform dan algoritma Run Length Encoding. Pelajaran ini menyajikan analisis komparatif dari kedua algoritma berdasarkan file vidio terkompresi dalam format file mp4. Analisis penelitian ini bertujuan untuk memberikan pengetahuan tentang algoritma mana yang lebih efektif diantara kedua algoritma tersebut agar ruang penyimpanan dan waktu pengiriman file menjadi lebih cepat . Diantaranya Discrete Cosine Transform dan Run Length Encoding Dengan terdapatnya beberapa algoritma tersebut perlu dilakukan pemilihan algoritma yang lebih tepat sehingga perlu dilakukan perbandingan algoritma kompresi. Metode eksponensial adalah yang memungkinkan untuk melakukan analisis perbandingan dari kedua algoritma. Dengan demikian, kita dapat melihat algoritma mana yang lebih efektif dan cepat untuk mengkompresi file vidio mp4.

Kata Kunci: Perbandingan; Discrete Cosine Transform; Run Length Encoding; Kompresi; File Vidio

Abstract-Technological development is one of the important aspects in the world of technology information. Many algorithms have been developed to compress files, including Discrete Cosine Transform algorithm and Run Length algorithm encoding. This lesson presents a comparative analysis of the two algorithms based on compressed video files in mp4 file format. This research analysis aims to provide knowledge about which algorithm is more effective between the two algorithms so that space faster file storage and delivery times. Among them Discrete Cosine Transform and Run Length Encoding With Some of these algorithms need to be chosen a more appropriate algorithm so it is necessary to compare the compression algorithms. The exponential method is the one that allows to do comparative analysis of the two algorithms. Thus, we can see which algorithm is more effective and faster to compress mp4 video files.

Keywords: Comparison; Discrete Cosine Transform; Run Length Encoding; Compression; Video Files

1. PENDAHULUAN

Perkembangan teknologi salah satu aspek penting dalam dunia teknologi informasi. Seiring waktu dengan perkembangan teknologi, data memiliki peranan yang sangat penting, besarnya penyimpanan yang didalam nya berupa memori ataupun hardisk pada media penyimpanan. Dapat kita menyajikan sebuah informasi data dengan ukuran yang lebih besar sebagai contoh pada file vidio. File vidio yang dapat kita ketahui pada media sosial seperti You Tube, Facebook, Tik Tok, Instagram dan masih banyak media sosial lainya. Ukuran file video tersebut tergolong begitu besar dan adanya pengaruh pada ruang penyimpanan dan waktu pengiriman data juga tergolong begitu lama. Maka data file tersebut dapat kita olah melakukan dengan cara mengkompresikan data. Kompresi merupakan sebuah cara untuk memadatkan data sehingga hanya memerlukan penyimpanan yang lebih kecil dan mempersingkat waktu pertukaran data tersebut, sehingga ukuran file tersebut akan memperkecil ruang kosong pada memori media penyimpanan.

Adapun file vidio yang digunakan pada penelitian ini berupa file berekstensi MP4. File MP4 salah satu format video yang mungkin sering dijumpai di internet. MP4 merupakan sebuah alternatif format video selain beberapa format video Digital seperti AVI, MKV, 3GP, RMVB dimana masing-masing memiliki sifat dan kualitas yang berlainan. Maka adanya Rancangan kompresi file mp4 agar dapat mengetahui dalam mengurangi ukuran pemakaian data pada penyimpanan komputer sehingga kecil kemungkinan data akan lebih cepat dalam proses pengiriman data. Pada perancangan penelitian ini dalam mengkompresi file digunakan aplikasi Microsoft Visual basic 2010. Dan menggunakan dua algoritma yang akan di bandingkan.

Pada penelitian ini dilakukan perbandingan algoritma untuk mengetahui algoritma manakah yang efektif dalam kompresi file berformat Mp4. Algoritma 2 yang akan dibandingkan yaitu algoritma Discrete Cosine Transform dan Run Length Encoding. Kedua algoritma tersebut merupakan algoritma yang di proses kompresi. Variabel perbandingan kedua algoritma adalah Kompresi Rasio (CR), Ratio Compression (RC) dan Space Saving (SS). Maka dapat di analisis perbandingan tersebut menggunakan metode perbandingan Eksponensial.

Algoritma Discrete Cosine Transform adalah metode yang pertama kali di perkenalkan oleh ahmed, Natarajan dan Rao pada tahun 1974 dalam makalahnya yang berjudul "On Image Processing and a Discrete Cosine Transform". Discrete Cosine Transform adalah sebuah teknik untuk mengubah sebuah sinyal kedalam komponen frekuensi dasar. Sifat dari Discrete Cosine Transform adalah mengubah informasi citra yang signifikan di konsentrasikan hanya pada beberapa koefisien DCT. Transformasi kosinus diskrit (discrete cosine transform) sering disingkat DCT mirip dengan transformasi fourier, hanya saja DCT menggunakan komponen sinus saja. DCT telah menjadi pilihan sebagai dasar algoritma kompresi JPEG dan MPEG[1]. Teknik yang memungkinkan untuk mengubah sinyal menjadi komponen frekuensi dasar. Teknik ini biasanya digunakan dalam kompresi gambar dan video.

Algoritma Run Length Encoding adalah teknik pengkodean loseless yang sederhana yang bekerja dengan memanfaatkan urutan data yang sama dalam suatu kelompok data. Dengan RLE, sebuah kumpulan data yang sama

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

cukup disimpan dalam sebuah nilai data tersebut dan sebuah nilai yang menyatakan jumlah data tersebut. Dalam 2omput yang dirancang, RLE digunakan untuk mengompres urutan byte keluaran encoder yang sebagian besar terdiri dari urutan 0 (nol) karena pengaruh interframe coding[2].

Berdasarkan penelitian terdahulu yang dilakukan oleh Hananto Edy Wibowo pada tahun 2006 dengan judul Penelitian "Kompresi Video Menggunakan Discrete Cosine Transform "menyimpulkan bahwa Pengembangan aplikasi gambar digital telah meningkatkan permintaan untuk gambar dan teknik kompresi video komputer dan efektif[3].

Berdasarkan penelitian terdahulu yang dilakukan oleh Ryan Anggara pada tahun 2015 dengan judul "Perancangan Dan Analisis Steganografi Vidio Dengan Menyisipkan Teks Menggunakan Metode DCT" menyimpulkan bahwa DCT digunakan untuk mengubah blok 8x8 piksel berturut-turut dari gambar menjadi 6[4].

Berdasarkan penelitian terdahulu yang dilakukan oleh Dwi Megasari pada tahun 2007 dengan judul "Analisis Dan Parancangan Kompresi Vidio Dengan Algoritma Run Length Encoding" Menyimpulkan bahwa algoritma Run Length Encoding menghasilkan ukuran memori komputer yang dikompresi menjadi lebih kecil karena disimpan dalam bentuk codeword sederhana yang mewakilkan banyak nya urutan warna yang sama, untuk sederhana hasil kompresi yang di hasilkan diantara 20% sampai 50% dari ukuran memori sebelum di kompresi[5].

2. METODOLOGI PENELITIAN

2.1 Kompresi

Kompresi merupakan suatu proses memperkecil ukuran data untuk menghasilkan representasi digital yang padat atau kompak, namun tetap mampu merepresentasikan jumlah informasi yang terkandung dalam data tersebut. Dalam gambar, video dan audio, hasil kompresi dalam meminimalkan jumlah bit rate untuk representasi digital. Dalam beberapa publikasi, istilah kompresi sering disebut sebagai pengkodean sumber, kompresi data, kompresi bandwidth, dan kompresi sinyal[1].

2.2 File Video

File vidio merupakan sebuah format file yang membagi beberapa kategori, tergantung perangkat perekam dan pemutaranya. Yang dapat ditemukan berbagai jenis format file vidio digital seperti AVI, MKV, 3GP, RMVB dan masih banyak lagi dimana file vidio ini masing – masing memiliki kualitas dan sifat yang berlainan[2].

2.3 Algoritma *Discrete Cosine Transform* (DCT)

Transformasi kosinus diskrit (discrete cosine transform) sering disingkat DCT mirip dengan transformasi fourier, hanya saja DCT menggunakan komponen sinus saja. DCT telah menjadi pilihan sebagai dasar algoritma kompresi JPEG dan MPEG. DCT mempunyai dua sifat utama untuk kompresi citra dan video[2], yaitu:

a. Mengkonsentrasikan input nilai ke dalam sejumlah kecil koefisien (input compaction).
b. Meminimalkan saling ketergantungan diantara koefisien-koefisien (decorrelation).

Keuntungan DCT antara lain : DCT (Discrete Cosine Transform) menghitung kuantitas bit-bit image dimana pesan tersebut disembunyikan didalamnya. Walaupun image yang dikompresi dengan Lossy Compression akan menimbulkan kecurigaan karena perubahan LSB akan terlihat jelas, pada metode ini hal ini tidak akan terjadi karena metode ini terjadi di domain frekuensi di dalam image.

Kekurangan DCT antara lain : Tidak tahan terhadap perubahan suatu objek dikarenakan pesan mudah dihapus karena lokasi penyisipan data dan pembuatan data dengan metode DCT diketahui.

2.4 Algoritma *Run Length Encoding* (RLE)

Run Length Encoding (RLE) merupakan teknik enkripsi Lossless sederhana yang bekerja menggunakan urutan data yang sama dalam kumpulan data. Dengan RLE, kumpulan data yang serupa hanya disimpan dalam nilai data tersebut dan nilai yang mewakili jumlah data tersebut. Pada yang dirancang, RLE digunakan untuk memampatkan urutan byte keluaran encoder yang sebagian besar terdiri dari 0 (nol) string karena pengaruh pengkodean interframe[4].

Misalkan rangkaian data berikut (17 byte) harus di kompres ABBBBBBCCCCDEEEEF menggunakan kompresi RLE, data dikompresi hanya membutuhkan 10 byte dan terlihat seperti ini A * 6B 4C D * 4E F setelah itu dikompresi string data berulang di gantikan oleh karakter (*) diikuti oleh jumlah karakter yang berulang dan karakter berulang itu sendiri. Karakter tidak tetap, dapat berbeda dari implementasi ke implementasi[11].

Tabel 1. Contoh Nilai Belum Terkompres Run Length Encoding

1	2	1	1	1
1	3	4	4	4
1	1	3	3	5
1	1	1	3	3

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

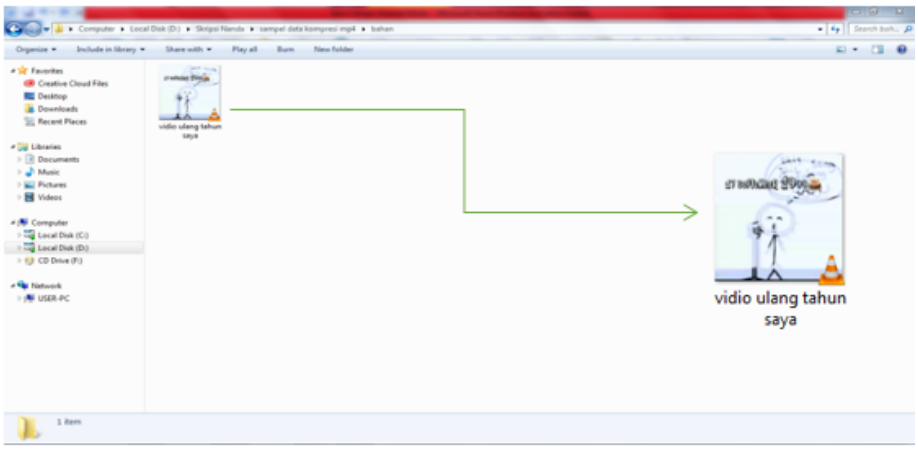
Diubah dalam bentuk sekuensial 1 2 1 1 1 1 1 3 4 4 4 4 1 1 3 3 3 5 1 1 1 1 3 3 = 24 byte Dihitung jumlah kemunculan data (1,1) (2,1) (1,5) (3,1) (4,4) (1,2) (3,3) (5,1) (1,4) (3,2) Data kompresi 1 2 1 1 5 3 1 4 4 1 2 3 3 5 1 1 4 3 2 = 20 byte.

3. HASIL DAN PEMBAHASAN

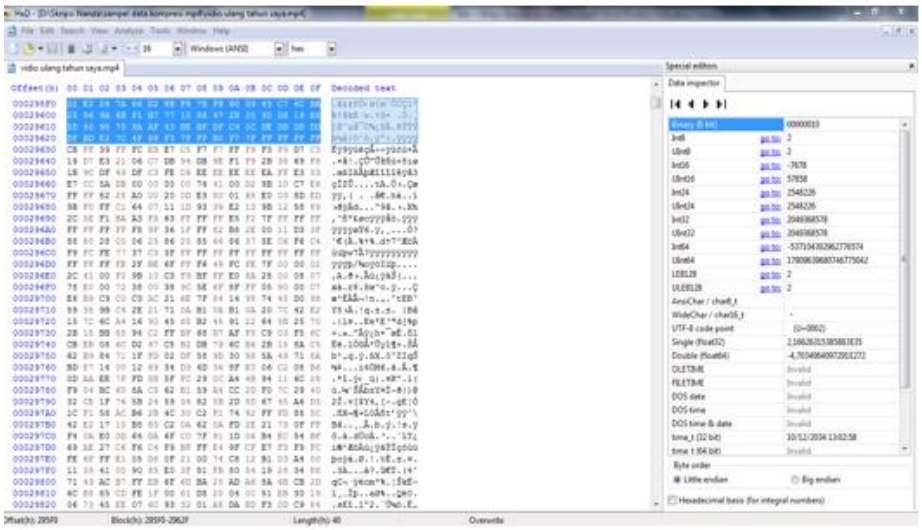
Berdasarkan analisa, kompresi file vidio dengan format ekstensi file MP4 memiliki kapasitas yang cukup besar. Dengan mengkompresi file MP4, file berukuran lebih besar akan di kompresi ke ukuran yang lebih kecil dan menghemat ruang penyimpanan dan algoritma yang lebih tepat dikenal untuk mengkompresi file MP4 dengan algoritma Discrete Cosine Transform dan Run Length Encoding. Dalam penelitian ini akan membahas 2 proses utama yaitu kompresi, dan peneliti akan melakukan kompresi dengan menggunakan dua algoritma yaitu algoritma Discrete Cosine Transform dan Run Length Encoding. Sebelum file MP4 dikompres, lebih dulu dilakukan pembacaan file MP4 agar didapatkan nilai hexadesimal menggunakan aplikasi HxD. Berikut contoh file MP4 yang dikompres serta . Pada penelitian ini hal yang paling utama ialah membandingkan kompresi dari algoritma *Discrete Cosine Transform* (DCT) dan *Run Lenght Encoding*.

Tabel 2. Sampel Data File Mp4

		Keterangan
Nama File	:	Video Ulang Tahun Saya.mp4
Jenis Item	:	File MP4 (mp4)
Ukuran	:	9,92 MB



Gambar 1. File Mp4 Yang Akan Di Kompresi



Gambar 2. Isi File Dari Hasil Hxd 8x8

Dari gambar diatas dapat diketahui nilai heksadesimal *file* MP4 sampel. Untuk keperluan perhitungan manual, maka hanya akan diambil nilai sebanyak 64 karakter nilai heksadesimal file MP4 sampel. Nilai Heksadesimal diambil

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

dari sisi kiri ke kanan. Nilai hexadesimal *file* MP4 ini ialah 02, E2, 26, 7A, 66, D2, 8B, F8, 7B, F8, 60, D3, 43, C7, 6C, BA, 25, 86, 9A, 6B, F1, B7, 77, 15, 86, 47, 2B, 20, 90, D6, 19, B8, 5D, 40, 98, 75, 9A, AF, 43, BE, BF, DF, C4, 0C, DE, DD, DD, DD, DF, BD, E2, 7C, 4F, 88, F3, 7F, FF, B0, F7, 7F, FF, FF, FF, FF. Nilai data ini sebelum dimasukan dalam tabel dan akan dilakukan pembacaan nilai dengan mengubah nilai hexadesimal menjadi nilai desimal untuk pembacaan nilai tersebut dapat memudahkan dalam proses perhitungan manual dengan algoritma DCT dan RLE. Nilai yang telah diubah ke desimal akan ditampilkan pada tabel dibawah ini:

Tabel 3. Sampel Data Nilai File Mp4

2	226	38	122	102	210	139	248
123	248	96	211	67	199	108	186
37	134	154	107	241	183	119	21
134	71	43	32	144	214	25	184
93	64	152	117	154	175	67	190
191	223	196	12	222	221	221	221
223	189	226	124	79	136	243	127
255	176	247	127	255	255	255	255

3.1 Proses Kompresi Algoritma DCT

Dalam proses kompresi menggunakan algoritma DCT pada vidio format ekstensi file mp4. Pada contoh vidio yang berukuran 9,92 mb yang akan digunakan untuk pengambilan nilai menggunakan software HXD dan Setiap nilai file vidio akan diambil sedikit dibagikan ke dalam bentuk makroblok 8x8 sesuai dengan nilai sampel data, maka akan terbentuk jumlah 64 nilai desimal yang akan dihitung dalam proses manual sesuai dengan rumus algoritma yang digunakan[10]. Dapat dilihat Pada tabel di bawah ini:

Tabel 4. Matriks 8x8

2	226	38	122	102	210	139	248
123	248	96	211	67	199	108	186
37	134	154	107	241	183	119	21
134	71	43	32	144	214	25	184
93	64	152	117	154	175	67	190
191	223	196	12	222	221	221	221
223	189	226	124	79	136	243	127
255	176	247	127	255	255	255	255

Setelah matriks original disusun maka matriks original akan dikurangi dengan 128 karena algoritma DCT bekerja pada rentang -128 sampai 127. Hasil setelah dikurang -128, dapat dilihat pada tabel sebagai berikut:

Tabel 5. Hasil Matriks M

-126	98	-90	-6	-26	82	11	120
-5	120	-32	83	-61	71	-20	58
-91	6	26	-21	113	55	-9	-107
6	-57	-85	-96	16	86	-103	56
-35	-64	24	-11	26	47	-61	62
63	95	68	-116	94	93	93	93
95	61	98	-4	-49	8	115	-1
127	48	119	-1	127	127	127	127

Menghitung Matriks T rumus ketetapan dari algoritma DCT dari T(0,0) sampai T(7,7). Dapat dilihat pada rumus sebagai berikut:

$T(i,j) = \frac{1}{\sqrt{N}} \; JIKA \; i = 0$

$= \frac{2}{\sqrt{N}} \; COS \; \frac{(2j+1)in}{2N} \; JIKA \; i \neq 0$

N = ordo matriks

Berdasarkan bentuk T(0,0) sampai T(7,7) akan diasumsikan menjadi matriks T, dapat dilihat pada tabel dibawah ini:

Tabel 6. Matriks T

0,3536	0,3536	0,3536	0,3536	0,3536	0,3536	0,3536	0,3536

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

0,4904	0,4158	0,278	0,0979	-0,0971	-0,2773	-0,4154	-0,4902
0,462	0,1916	-0,1909	-0,4617	-0,4623	-0,1924	0,1901	0,4614
0,4158	-0,0971	-0,4902	-0,2787	0,2767	0,4907	0,0994	-0,4145
0,3537	-0,3531	-0,3543	0,3526	0,3548	-0,352	-0,3554	0,3514
0,278	-0,4902	0,0963	0,4167	-0,4145	-0,1002	0,491	-0,2747
0,1916	-0,4623	0,4614	-0,1894	-0,1938	0,4632	-0,4604	0,1872
0,0979	-0,2787	0,4167	-0,4909	0,4898	-0,4136	0,274	-0,0924

Setelah itu matriks T tersebut akan dibuat matriks baru yang merupakan matriks transpose(Tt), dapat dilihat pada tabel dibawah ini:

Tabel 7. Matriks Transpose T

0,3536	0,4904	0,462	0,4158	0,3537	0,278	0,1916	0,0979
0,3536	0,4158	0,1916	-0,0971	-0,3531	-0,4902	-0,4623	-0,2787
0,3536	0,278	-0,1909	-0,4902	-0,3543	0,0963	0,4614	0,4167
0,3536	0,0979	-0,4617	-0,2787	0,3526	0,4167	-0,1894	-0,4909
0,3536	-0,0971	-0,4623	0,2767	0,3548	-0,4145	-0,1938	0,4898
0,3536	-0,2773	-0,1924	0,4907	-0,352	-0,1002	0,4632	-0,4136
0,3536	-0,4154	0,1901	0,0994	-0,3554	0,491	-0,4604	0,274
0,3536	-0,4902	0,4614	-0,4145	0,3514	-0,2747	0,1872	-0,0924

Menghitung Matriks D hasil dari proses ini nantinya matriks D akan terisi dengan Koefisiensi DCT. tapi sebelumnya cari koefisien DCT (*Discrete Cosine Transform*) dengan rumus yang pada di bawah ini sebagai berikut:

$$D = (T * M) * T^t$$

Maka berikut perkalian matriks D dapat dilihat hasil pada tabel dibawah ini:

Tabel 8. Hasil Matriks D

-15,7542	16,99373	-14,7027	-0,88216	-3,25178	8,060666	0,745247	4,154093
-0,86703	20,74676	-1,70447	-0,78901	-2,09145	9,651205	-3,84079	7,923887
-14,8661	0,319589	0,947513	-4,75283	18,5086	-1,01905	-0,78941	-20,5724
0,882161	0,541847	-19,2377	-7,45667	1,561031	17,58482	1,939115	11,39477
-4,37739	-2,1943	3,931029	-1,07321	3,272959	6,857488	-4,20147	10,67117
6,19295	12,91358	-1,25991	-23,7191	13,71498	0,933724	21,1511	10,56628
6,436227	11,7144	8,59579	0,075305	-3,37495	1,81945	24,37634	-0,05129
4,396415	6,5577	22,87958	-0,20348	21,8587	14,42922	6,514186	1,084296

a. Melakukan Kuantisasi
Proses ini menggunakan aturan matriks kuantisasi. Aturan matriks kuantisasi Diasumsikan sebagai matriks Q, dijelaskan pada tabel dibawah ini:

Tabel 9. Matriks Kuantisasi Diasumsikan Sebagai Matriks Q

16	11	10	16	24	51	51	61
12	12	14	19	26	60	60	55
14	13	16	24	40	69	69	56
14	17	22	29	51	80	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Maka persamaan matriks kuantisasi adalah sebagai berikut, dimana round berarti mendekatkan nilai hasil pembagian ke pembulatan nilai integer terdekat yaitu:

$$C_{ij} = round\ \frac{D_{ij}}{Q_{ij}}$$

Dimana i dan j merupakan baris dan kolom dari matriks, persamaan ini merupakan pembagian elemen dari matriks, bukan pembagian matriks. Maka Berdasarkan hasil dari kuantisasi yang di dapat yaitu dapat dilihat pada tabel dibawah ini sebagai berikut:

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

Tabel 10. Hasil Dari Matriks Kuantisasi Diasumsikan Sebagai Matriks C

-1	2	-1	0	0	0	0	0
0	2	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

b. Menentukan Rasio Kompresi DCT

Berdasarkan data yang diperoleh untuk mencari hasil dari rasio kompresi DCT yang akan di gunakan pada langkah kuantisasi matriks yang sudah di dapatkan pada algoritma DCT di atas, selanjutnya mengelompokan urutan yang sama pada dibawah ini[9]. Urutan kuantisasi kompresi DCT sebagai berikut:

(-1, 2, -1, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

Maka hasil pengkodean urutan kemunculan yang sama

-1 2 -1 0,6 2 0,6 -1 0,9 -1 0,37 Semuanya ada = 14

Maka dari data ini dapat dilakukan perhitungan kinerja kompresinya yaitu:

1. Ukuran sebelum dikompresi matriks asli 8 x 8 = 64
2. Ukuran setelah di kompresi matriks kuantisasi dengan pengkodean yang sama = 14

Compression Ratio (CR) $= \frac{Ukuran\ data\ sesudah\ dikompresi}{Ukuran\ data\ sebelum\ dikompresi}$ x 100%

$= \frac{14}{64}$ x 100%

= 21,87%

Ratio of Compression (RC) $= \frac{Ukuran\ data\ sebelum\ dikompresi}{Ukuran\ data\ sesudah\ dikompresi}$

$= \frac{64}{14}$

= 4,57 Byte

Space Saving (SS) $= 100\% - CR$

= 100% - 21,87

= 78,13%

3.2 Proses Kompresi Algoritma *Run Length Encoding* (RLE)

Kompresi dengan algoritma Run Length Encoding ialah menyusun urutan nilai atau karakter yang sama Sebagai berikut[12]:

a. Berdasarkan sampel data file mp4 yang sudah didapatkan sebelumnya yang diubah bentuk nya ke nilai heksadesimal ke desimal yang akan dihitung manual sebagai berikut:

Table 11. Sampel Data Nilai File Mp4

2	226	38	122	102	210	139	248
123	248	96	211	67	199	108	186
37	134	154	107	241	183	119	21
134	71	43	32	144	214	25	184
93	64	152	117	154	175	67	190
191	223	196	12	222	221	221	221
223	189	226	124	79	136	243	127
255	176	247	127	255	255	255	255

b. Ubah Data dalam bentuk Sekunsial dari kiri ke kanan pada dibawah ini:
2, 226, 38, 122, 102, 210, 139, 248, 123, 248, 96, 211, 67, 199, 108, 186, 37, 134, 154, 107, 241, 183, 119, 21, 134, 71, 43, 32, 144, 214, 25, 184, 93, 64, 152, 117, 154, 175, 67, 190, 191, 223, 196, 12, 222, 221, 221, 221, 223, 189, 226, 124, 79, 136, 243, 127, 255, 176, 247, 127, 255, 255, 255, 255.
Semuanya ada = 64.

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

Setelah itu:

c. Hitung jumlah kemunculan nilai yang sama pada dibawah ini:
Jika nilainya tidak sama maka di anggap 1 byte jika sama maka dianggap 2 byte dibaca dari kiri ke kanan maka hasil kompresinya:
(2) (226) (38) (122) (102) (210) (139) (248) (123) (248) (96) (211) (67) (199) (108) (186) (37) (134) (154) (107) (241) (183) (119) (21) (134) (71) (43) (32) (144) (214) (25) (184) (93) (64) (152) (117) (154) (175) (67) (190) (191) (223) (196) (12) (222) (221,3) (223) (189) (226) (124) (79) (136) (243) (127) (255,4).
Setelah disusun hasil kompresinya, selanjutnya:

d. Hasil Pengkodean maka disusun pada dibawah ini:
2, 226, 38, 122, 102, 210, 139, 248, 123, 248, 96, 211, 67, 199, 108, 186, 37, 134, 154, 107, 241, 183, 119, 21, 134, 71, 43, 32, 144, 214, 25, 184, 93, 64, 152, 117, 154, 175, 67, 190, 191, 223, 196, 12, 222, 221, 3, 223, 189, 226, 124, 79, 136, 243, 127, 255, 4.
Maka semuanya adalah = 57.

e. Rumus perhitungan kompresi pada dibawah ini sebagai berikut:
Berdasarkan hasil data susunan diatas maka diproleh:
Ukuran sebelum di kompresi = 64.
Ukuran setelah dikompresi = 57.
Maka dari data ini dapat dilakukan perhitungan kinerja kompresinya yaitu:

Compression Ratio (CR) $= \frac{Ukuran\ data\ sesudah\ dikompresi}{Ukuran\ data\ sebelum\ dikompresi}$ x 100%

$= \frac{57}{64}$ x 100%

= 89,6%

Ratio Of Compression (RC) $= \frac{Ukuran\ data\ sebelum\ dikompresi}{Ukuran\ data\ sesudah\ dikompresi}$

$= \frac{64}{57}$

= 1,12 Byte

Space Saving (SS) = 100% - *CR*

= 100% - 89,6%

= 10,4%

Maka berdasarkan perhitungan diatas maka hasil rasio file kompresi didapatkan sebesar 89,6%.

3.3 Hasil Kinerja Algoritma DCT Dan RLE

Hasil algortima Discrete Cosine Transform dan Run Length Encoding ialah membandingan rasio kompresi dan space saving dari kedua algoritma. Dimana setelah dilakukan perhitungan dengan algoritma Discrete Cosine Transform dan Run Length Encoding maka hasil dari rasio kompresi dan space saving didapatkan, berdasarkan hasil kinerja kompresi tersebut algoritma RLE yang memiliki rasio terbesar dibandingkan algoritma DCT. Adapun tabel perbandingan dapat di lihat pada tabel di bawah ini:

Table 12. Hasil Kinerja Kompresi File Mp4

No	Algoritma	*Kompresi Rasio (CR)*	*Ratio Compression (RC)*	*Space Saving*
1	*Discrete Cosine Transform (DCT)*	21,87%	4,57 Byte	78,13
2	*Run Length Encoding*	89,6%	1,12 Byte	10,4

3.4 Analisa Perbandingan Algoritma Menggunakan Metode Perbandingan Eksponensial

Maka selanjutnya adalah analisa perbandingan tersebut menggunakan *Metode Perbandingan Eksponensial* yang akan di hitung dari kedua hasil kinerja kompresi tersebut pada dibawah ini:
Langkah – langkah Perhitungan Perbandingannya sebagai berikut:

Nilai Perbandingan Algoritma DCT $= (21,87)^{0,3} + (4,57)^{0,4} + (78,13)^{0,3}$

= 2,523 + 1,836 + 3,696

= 8,055

Nilai Perbandingan Algoritma RLE $= (89,6)^{0,3} + (1,12)^{0,4} + (10,4)^{0,3}$

= 3,852 + 1,047 + 2,018

ISSN 2597-4645 (media online)
ISSN 2597-4610 (media cetak)
Page: 271-278

= 6,917

Table 13. Analisa Perbandingan Algoritma Menggunakan Metode Perbandingan Eksponensial

Kriteria	Bobot	Algoritma DCT	Algoritma RLE
CR	0,3	21,87 %	89,6 %
RC	0,4	4,57 Byte	1,12 Byte
SS	0,3	78,13	10,4
Total Nilai $\sum(N)^B$	1	8,055	6,917

Berdasarkan perhitungan yang sudah diperoleh dari nilai akhir atau total nilai perbandingan maka masing – masing algoritma dilakukan kesimpulan bedasarkan analisa tersebut algoritma RLE (Run Length Encoding) yang menjadi algoritma yang bagus dan yang lebih efektif dalam melakukan proses kompresi file vidio MP4. hal ini di karenakan semakin kecil total nilai Perbandingan yang di dapat maka semakin sedikit usaha yang dilakukan oleh algoritma tersebut dalam melakukan proses kompresi.

4. KESIMPULAN

Ada beberapa kesimpulan yang dapat dilakukan. Kesimpulan tersebut antara lain, kompresi file vidio Mp4 dapat dilakukan dengan mengubah kedalam bentuk heksadesimal dan di ubah menjadi nilai desimal untuk dapat melakukan perhitungan. Bilangan desimal yang didapat diolah menggunakan algoritma *Discrete Cosine Transform* dan *Run Length Encoding* sehingga menghasilkan nilai yang baru yang akan merubah ukuran file vidio *Mp4*. Algoritma *DCT* dan *RLE* dapat dilakukan perbandingan setelah dilakukan perhitungan. Dengan skala perbandingan yaitu rasio kompresi dan space saving. Dengan perbandingan yang dilakukan maka diketahui bahwa persentase rasio kompresi algoritma *RLE* lebih besar daripada algoritma *DCT*. Rasio kompresi dan space dapat diketahui dengan melakukan mengubah nilai awal file vidio Mp4 dengan nilai heksadesmial atau nilai desimal. Didapatkan nilai rasio kompresi algoritma *DCT* sebesar 21,87% dan hasil rasio kompresi dengan algoritma *Run Length Encoding* sebesar 89,6%. Analisa perbandingan digunakan metode perbandingan eksponensial tersebut maka dari total nilai akhir atau nilai perbandingan maka dapat disimpulkan nilai perbandingan algoritma DCT mendapat sebesar 8,055 sedangkan total nilai perbandingan RLE mendapat sebesar 6,917 hal ini di karena kan semakin kecil total nilai tersebut semakin sedikit usaha untuk kompresi maka dapat dinyatakan algoritma RLE (*Run Length Encoding*) yang menjadi efektif pada kompresi file mp4.

REFERENCES

[1] R. Syahputra, "KOMPRESI FILE VIDEO MP4 DENGAN MENGGUNAKAN METODE DISCRETE COSINE TRANSFORM," *J. Ris. Komput.*, vol. 3, no. 1, pp. 2407–389X, 2016.

[2] gelar B. Fairuz Azmi, Budi irawan, "Perancangan Codec Berbasis Algoritma Kompresi H.264 untuk Aplikasi Konferensi Video," vol. 17, no. 1, pp. 1412–0100, 2016.

[3] N. H. Henanto Edy Wibowo, Indra Wijayanto, "Kompresi Video Menggunakan Discrete Cosine Transform," *Makal. STMIK*, 2006.

[4] M. T. Ryan Anggara, Gelar Budiman, S.T., M.T., Ledya Novamizanti S.Si., "PERANCANGAN DAN ANALISIS STEGANOGRAFI VIDEO DENGAN MENYISIPKAN TEKS MENGGUNAKAN METODE DCT," *e-Proceeding Eng.*, vol. 2, no. 2, p. 2615, 2015.

[5] S. S. Dwi Megasari, Bakri Tanet, "Analisis dan Perancangan Kompresi Vidio Dengan Algoritma Run length Encoding," *Jursan Tek. Inform. Skripsi Sarj. Komput.*, 2007.

[6] M. Fatmawaty, "Analisis Perbandingan Kompresi File Wav Menggunakan Metode Huffmandan Run Length Encoding," *J. Teknol. Inf. dan Terap.*, vol. 7, no. 1, pp. 2580–2291, 2020.

[7] S. A. Muhammad Alfarizi, "Penarapan Algoritma Prefix Code Dalam Kompresi File Video," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 4, no. 1, pp. 2597–4645, 2020.

[8] A. A. Tigus Juni Betri, Esti Suryani, "VIDEO WATERMARKING UNTUK PERLINDUNGAN HAK CIPTA DENGAN ALGORITMA KOCH ZHAO," *J. ITSMART V*, vol. 3, no. 2301–7201, p. 2, 2014.

[9] E. Prayoga and K. M. Suryaningrum, "Implementasi Algoritma Huffman Dan Run Length Encoding Pada Aplikasi Kompresi Berbasis Web," *J. Ilm. Teknol. Infomasi Terap.*, vol. 4, no. 2, pp. 92–101, 2018, doi: 10.33197/jitter.vol4.iss2.2018.154.

[10] D. Hadi, "Perancangan Citra Watermaking Pada Citra Digital Menggunakan Metode Discrete Cosine Transform (Dct)," *Pelita Inform. Budi Darma*, vol. 7, pp. 104–107, 2014.

[11] R. Krasmala, A. Budimansyah, and U. T. Lenggana, "Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) dan Algoritma Huffman," *J. Online Inform.*, vol. 2, no. 1, p. 1, 2017, doi: 10.15575/join.v2i1.79.

[12] D. Willfrid, M. Simamora, G. Ginting, and Y. Hasan, "Implementasi Algoritma Run Length Encoding Pada Kompresi File Mp3," *JURIKOM (Jurnal Ris. Komputer)*, vol. 3, no. 4, pp. 5–9, 2016.