

Penerapan Algoritma Ternary Comma Code Pada Aplikasi Kompresi File Gambar

Nurma Ningsih, Surya Darma Nasution, Taronisokhi Zebua*

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: nurmaningsih@mail.com, *taronisokhi@stmik-budidarma.ac.id

Abstrak– Sejak ditemukannya alat untuk menangkap sebuah gambar (citra) yaitu berupa kamera membuat perkembangan teknologi tidak hanya berfokus pada alat-alatnya saja tapi juga harus fokus pada tempat penyimpanan data tersebut. Kebutuhan akan media penyimpanan merupakan suatu hal yang sangat dibutuhkan dalam setiap alat komunikasi. Sistem komunikasi data dan informasi tidak hanya berupa teks tapi juga dapat berupa gambar, video, dan audio. Ukuran file gambar yang relatif lebih besar dibandingkan dengan file teks merupakan salah satu penyebab dari besarnya memori yang digunakan untuk penyimpanan file gambar. Besarnya ukuran file gambar juga dapat menyebabkan lamanya waktu untuk mengirimkan file gambar tersebut. Proses kompresi file gambar merupakan salah satu cara yang dapat dilakukan untuk mengatasi masalah ukuran file gambar. Kompresi gambar merupakan suatu usaha untuk meminimalkan ukuran file atau byte dari file grafis tanpa mengurangi kualitas gambar yang ada. Penyimpanan dalam jumlah tertentu, seperti ruang disk atau memori yang memungkinkan untuk melakukan pengurangan ukuran file gambar. Banyak algoritma yang dapat digunakan untuk mengkompresi ukuran file gambar di antaranya algoritma ternary comma code yang dapat membantu mengecilkan ukuran file gambar. Algoritma ternary comma code memiliki ruang waktu yang lebih cepat dibandingkan dengan algoritma lain.

Kata Kunci: Kompresi, Algoritma Ternary Comma Code, File Gambar

Abstract– Since the discovery of a tool to capture an image (image) in the form of a camera making technological developments not only focus on the tools but also must focus on where the data is stored. The need for storage media is something that is needed in every communication tool. Data and information communication systems are not only in the form of text but can also be in the form of images, videos and audio. Image file size that is relatively larger compared to text files is one of the causes of the amount of memory used for storing image files. The size of the image file can also cause the length of time to send the image file. Image file compression process is one way that can be done to overcome the problem of image file size. Image compression is an attempt to minimize the file size or bytes of a graphic file without reducing the quality of existing images. A certain amount of storage, such as disk space or memory, makes it possible to reduce the image file size. Many algorithms that can be used to compress image file sizes include the ternary comma code algorithm that can help reduce image file sizes. The ternary comma code algorithm has a faster time space compared to other algorithms.

Keywords: Compression, Ternary Comma Code Algorithm, Image File

1. PENDAHULUAN

Sejak ditemukannya alat untuk menangkap sebuah gambar (citra) yaitu berupa kamera membuat perkembangan teknologi tidak hanya berfokus pada alat-alatnya saja tapi juga harus fokus pada tempat penyimpanan data tersebut. Kebutuhan akan media penyimpanan merupakan suatu hal yang sangat dibutuhkan dalam setiap alat komunikasi. Sistem komunikasi data dan informasi tidak hanya berupa teks tapi juga dapat berupa gambar, video, dan audio.

Gambar merupakan suatu tampilan objek yang ditampilkan dalam bentuk lain yang memberikan representasi visual (gambaran) tentang suatu objek atau pemandangan. Gambar tersimpan dalam bentuk *file* yang memiliki format gambar yang berbeda. Format gambar/*file* gambar merupakan tempat penyimpanan gambar yang sering digunakan untuk berbagai keperluan seperti halaman *web* dan lain-lain. Format gambar juga mampu menyimpan gambar dalam beberapa model yang disediakan *photoshop*. Ada banyak jenis *file* gambar seperti JPG, JPEG, BITMAP, GIF dan lain sebagainya.

File JPEG merupakan salah satu format yang sering digunakan untuk penyimpanan sebuah gambar yang memiliki beberapa kelebihan diantaranya mampu menampilkan gambar dengan kedalaman 24 bit. Format ini juga umum digunakan untuk menyimpan gambar hasil foto pada handphone. Banyaknya foto yang disimpan dari hasil kamera membuat memori/ruang penyimpanan menjadi penuh. Ukuran *file* gambar yang relatif lebih besar dibandingkan dengan *file* teks merupakan salah satu penyebab dari besarnya memori yang digunakan untuk penyimpanan *file* gambar. Besarnya ukuran *file* gambar juga dapat menyebabkan lamanya waktu untuk mengirimkan *file* gambar tersebut. Proses kompresi *file* gambar merupakan salah satu cara yang dapat dilakukan untuk mengatasi masalah ukuran *file* gambar.

Kompresi itu sendiri merupakan suatu proses mengecilkan/memampatkan ukuran *file* gambar yang memiliki kapasitas besar ke dalam kapasitas yang lebih kecil sehingga ruang memori yang dibutuhkan lebih sedikit. Mengkompresi sebuah gambar merupakan suatu usaha untuk meminimalkan ukuran *file* tanpa mengurangi kualitas gambar yang ada sehingga memori yang digunakan dalam penyimpanan *file* gambar tersebut menjadi lebih kecil. File gambar dengan ukuran besar mengakibatkan banyaknya ruang memori atau ruang disk menjadi penuh sehingga di butuhkan kompresi untuk meminimalkan ukuran *file* gambar.

Menurut penelitian sebelumnya yang dilakukan oleh Melita, Yuliana pada tahun 2011 kompresi gambar merupakan proses meminimalkan ukuran *file* dalam *byte* dari *file* grafis tanpa mengurangi kualitas gambar yang akan diterima[1].

Dapat disimpulkan kompresi merupakan usaha untuk memperkecil ukuran *file* gambar sehingga memudahkan kita untuk mengalokasikan ruang penyimpanan. File yang telah dialokasikan akan berdampak baik pada tempat penyimpanan (memori) kerna ruang penyimpanannya menjadi sedikit/lebih kecil.

Banyak algoritma yang dapat di gunakan untuk mengkompresi ukuran *file* dari antaranya algoritma *Ternary Comma Code* yang dapat membantu mengecilkan ukuran *file* gambar. Algoritma *Ternary Comma Code* memiliki ruang waktu yang lebih cepat dibanding dengan algoritma yang lain. Algoritma *Ternary Comma Code* merupakan jenis kompresi *lossless* yang dapat di gunakan dalam mengkompresikan file gambar tanpa mengurangi kualitas gambar.

2. METODOLOGI PENELITIAN

2.1 Kompresi

Proses kompresi merupakan proses mereduksi ukuran suatu data untuk menghasilkan representasi digital yang padat atau mampat (*compact*) namun tetap dapat mewakili kualitas informasi yang terkandung pada data tersebut. Sebuah *file* citra, *video*, dan *audio*, kompresi mengarah pada minimisasi jumlah bit *rate* untuk representasi digital. Beberapa *literature*, istilah kompresi sering disebut juga *source coding*, *data compression*, *bandwidth compression*, dan *signal compression*.

Data dan informasi adalah dua hal yang berbeda. Setiap data terkandung suatu informasi. Namun tidak semua bagian data terkait dengan informasi atau pada suatu data terdapat bagian-bagian data yang berulang untuk mewakili informasi yang sama. Bagian data yang tidak terkait atau bagian data yang berulang tersebut disebut dengan data berlebihan (*redundancy data*). Tujuan kompresi data adalah untuk mengurangi data berlebihan sehingga ukuran data menjadi lebih kecil dan lebih ringan dalam proses transmisi[3].

2.2 Algoritma Ternary Comma Code

Angka binary (basis 2) adalah dua bit yaitu 0 dan 1. Sama dengan angka *Ternary* (basis 3) di dasarkan pada 3 digit angka yaitu 0, 1, dan 2. Setiap angka bisa dikodekan dalam 2 bit tetapi 2 bit tersebut dapat memiliki 4 nilai. Hal tersebut tampak masuk akal untuk bekerja dengan menggunakan sistem angka *Ternary* di mana setiap angka diwakili oleh 2 bit dan dalam sebuah penjumlahan menjadi tiga angka terdapat simbol keempat yaitu *Comma* (c). Dengan memasukkan simbol c memudahkan membangun *Ternary Comma Code* untuk bilangan bulat. *Comma Code* untuk n adalah bentuk sederhana dari perwakilan *Ternary* dari n – 1 yang di ikuti oleh c. Seperti *Comma Code* dari 8 adalah 21c karna $7=2*3+1$ dan *Comma Code* dari 18 adalah 122c karna $17=1*9+2*3+2$ [5].

Langkah-langkah algoritma *Ternary Comma Code* :

1. Setiap bilangan n (bilangan bulat) dikurang satu.
2. Hasil nilai pengurangan n-1 di bagi 3 (karna *Ternary* merupakan basis 3).
3. Masukkan c (*Comma Code*) untuk n dari perwakilan *Ternary* dari n-1.

2.3 Citra

Citra merupakan kemiripan dari suatu representasi (gambaran) dari suatu objek. Sebagai keluaran dari suatu perekaman data citra dapat bersifat optik berupa foto, analog berupa sinyal video seperti gambar pada monitor televisi. Terdapat tiga jenis citra yaitu citra biner, citra warna, dan citra hitam putih [8].

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

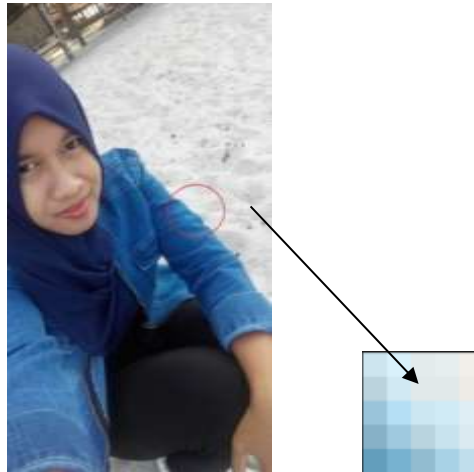
Analisa merupakan proses identifikasi mengenai hal-hal yang dibutuhkan dan harus ada dalam menyelesaikan suatu masalah, baik analisa sistem maupun analisa masalah yang akan diselesaikan. Analisa sistem itu sendiri merupakan proses identifikasi dari sistem yang akan dibangun untuk menentukan proses *input*, pengolahan data, dan *output*. Analisa masalah merupakan proses identifikasi masalah yang terjadi seperti sebab dan akibat dari sistem yang akan dibangun.

Permasalahan dari penelitian ini yaitu bagaimana memperkecil ukuran *file* gambar dengan menguraikan prosedur pemampatan ukuran *file* gambar berdasarkan teknik kompresi sehingga menghasilkan ukuran *file* gambar yang lebih kecil. Pengompresian ukuran *file* menggunakan algoritma *Ternary Comma Code*. *file* gambar yang dikompresi sebagai contoh dalam penelitian ini yaitu *file* gambar yang berformat JPG.

Saat ingin mengkompresi suatu *file* gambar maka harus dilakukan pencarian *file* gambar. Setelah mendapatkan *file* gambar yang ingin dikompres maka yang akan di lakukan selanjutnya yaitu melakukan pembacaan nilai *pixel*. Setelah nilai dari *pixel* terbaca maka lakukan kompresi dengan menggunakan algoritma *ternary comma code* dan dapatlah *file* gambar yang telah terkompresi. Begitu juga pada saat ingin melakukan dekompresi. Hal pertama yang harus dilakukan yaitu membaca nilai *pixel* yang telah dikompresi. Setelah nilai *pixel* terbaca bentuk kembali ke nilai *pixel* awal sehingga kembali ke bentuk *file* gambar yang asli.

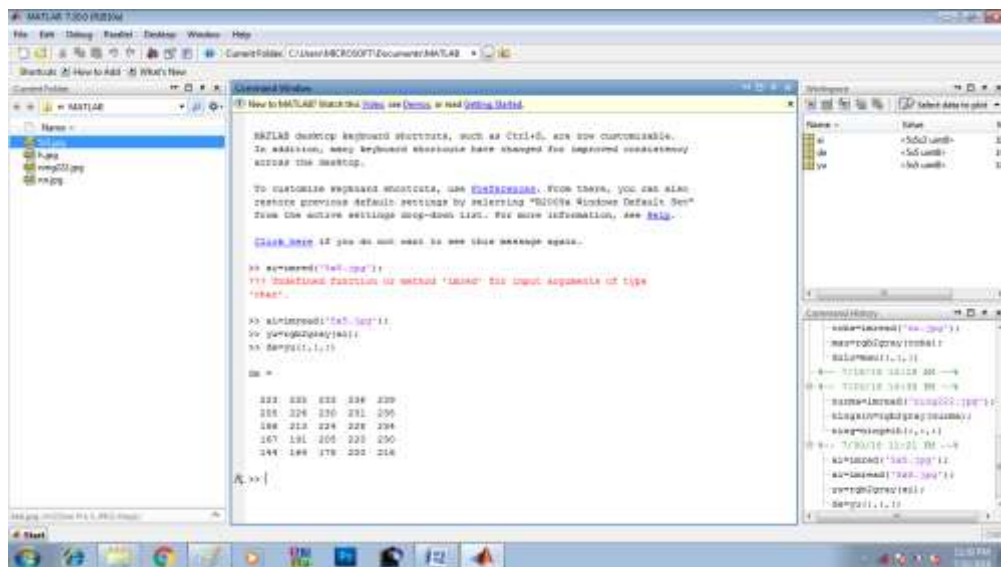
3.2 Penerapan Algoritma Camellia

File yang digunakan dalam contoh kasus ini yaitu file gambar berformat JPG yang memiliki ukuran 2160x3840 pixel. Sebagai contoh analisa yang dilakukan di ambil citra sebesar 5 x 5 pixel dengan kedalaman warna 24 bit yang akan dikompresi dengan menggunakan algoritma Ternary Comma Code.



Gambar 1. File Gambar

Setiap isi dari file gambar memiliki nilai pixel. Pengambilan setiap nilai pixel dari contoh gambar 5 x 5 pixel yaitu dengan menggunakan aplikasi matlab. Nilai dari masing-masing pixel dapat di lihat pada gambar 3.3.



Gambar 2. Nilai dari 5 x 5 pixel

Citra warna dengan ukuran 5x5 pixel menghasilkan nilai matriks dengan 25 pixel seperti di bawah ini :

Tabel 1. Nilai matriks 5x5 pixel

223	235	233	236	239
205	226	230	231	235
186	213	224	228	234
167	191	205	220	230
144	164	178	200	216

Berdasarkan tabel matriks di atas sebelum dikompresi nilai setiap pixel harus dikelompokkan seperti tabel di bawah ini :

Tabel 2. Nilai matriks yang belum dikompresi

No.	Nilai Pixel	ASCII Binary	Frek	Bit	Frek x Bit
1.	144	10010000	1	8	8
2.	164	10100100	1	8	8
3.	167	10100111	1	8	8

No.	Nilai Pixel	ASCII Binary	Frek	Bit	Frek x Bit
4.	178	10110010	1	8	8
5.	186	10111010	1	8	8
6.	191	10111111	1	8	8
7.	200	11001000	1	8	8
8.	205	11001101	2	8	16
9.	213	11010101	1	8	8
10.	216	11011000	1	8	8
11.	220	11011100	1	8	8
12.	223	11011111	1	8	8
13.	224	11100000	1	8	8
14.	226	11100010	1	8	8
15.	228	11100100	1	8	8
16.	230	11100110	2	8	16
17.	231	11100111	1	8	8
18.	233	11101001	1	8	8
19.	234	11101010	1	8	8
20.	235	11101011	2	8	16
21.	236	11101100	1	8	8
22.	239	11101111	1	8	8
Jumlah					200

1. Proses Kompresi

Proses kompresi untuk algoritma Ternary Comma Code dapat dilihat pada tabel di bawah :

Tabel 3. Nilai matriks yang telah diurutkan secara *descending* dan tabel *sorted* dan telah di kompresi dengan kode Ternary Comma Code

No.	Nilai Pixel	Ternary Comma Code	Biner (Nol)	Frek	Bit	Frek x Bit
1	205	0c	1	2	1	2
2	230	1c	10	2	2	4
3	235	2c	11	2	2	4
4	144	10c	100	1	3	3
5	164	11c	101	1	3	3
6	167	12c	110	1	3	3
7	178	20c	111	1	3	3
8	186	21c	1000	1	4	4
9	191	22c	1001	1	4	4
10	200	100c	1010	1	4	4
11	213	101c	1011	1	4	4
12	216	102c	1100	1	4	4
13	220	110c	1101	1	4	4
14	223	111c	1110	1	4	4
15	224	112c	1111	1	4	4
16	226	120c	10000	1	5	5
17	228	121c	10001	1	5	5
18	231	122c	10010	1	5	5
19	233	200c	10011	1	5	5
20	234	201c	10100	1	5	5
21	236	202c	10101	1	5	5
22	239	210c	10110	1	5	5
Jumlah						89

Proses untuk mendapatkan kode ternary yaitu bilangan bulat dari n-1 yang di ikuti oleh c. Sebagai contoh di ambil dari tabel 3.4 dengan nilai n=13. Jika n-1, maka 13-1=12 lalu lakukan pembagian 3 dan di hasilkan nilai ternary 1101 dan masukkan simbol comma code (c). Sehingga nilai ternary dari n=13=1101c.

Berdasarkan nilai pixel sebelum dikompresi yaitu “223 235 233 236 239 205 226 230 231 235 186 213 224 228 234 167 191 205 220 230 144 164 178 200 216” dihasilkan nilai pixel yang telah dikompresi dengan algoritma Ternary Comma Code “0c1c2c10c11c12c20c21c22c100c101c102c110c111c112c120c121c122c200c201c202c210c”.

Dalam sebuah perhitungan komputer tidak mengenal basis 3 (0, 1, dan 2), maka untuk mencari nilai bit di atas digunakan rumus $n-1$ dan di ubah ke basis 2 (0 dan 1). Misalnya nilai $n=13-1=12$ maka nilai bit dari $13=1101$. Setelah mendapat nilai bit masukkan nilai bit dari *comma code* (c). Nilai c didapat dari jumlah n terakhir dari tabel 3.4. Diketahui nilai terakhir dari tabel 3.4 ($n=22$) maka bit dari c ($n=23$) sehingga nilai bit dari $c=10111$. Gabungkan nilai bit dari setiap *pixel* tambahkan nilai bit dari c. Dari penggabungan bit dihasilkan bit seperti berikut “110111 1010111 1110111 10010111 10110111 11010111 11110111 100010111 100110111 101010111 101110111 110010111 110110111 11110111 1000010111 1000110111 1001010111 1001110111 1010010111 1010110111 1011010111”

Sebelum ditulis ke sebuah *file* hasil kompresi dilakukan penambahan *string* bit itu sendiri apakah habis dibagi 8 dan berapa sisanya jika dibagi 8. Jumlah string bit setelah dikompresi adalah 89 ditambah string bit dari c. String bit dari c yaitu 5 lalu dikali dengan jumlah *pixel* sebanyak 22 maka didapatlah string bit dari c sebanyak 110. Jadi total keseluruhan bit $89+110=199$ di bagi 8 dihasilkan 24bit dan memiliki sisa 2. Lakukan pemisahan bit menjadi beberapa kelompok yang terdiri dari 8 bit seperti gambar di bawah ini :

11011110	10111111	01111001	01111011	01111101
01111111	01111000	10111100	11011110	10101111
01110111	11001011	11101101	11111010	11111111
01111000	01011110	00110111	10010101	11100111
01111010	01011110	10110111	10110101	11

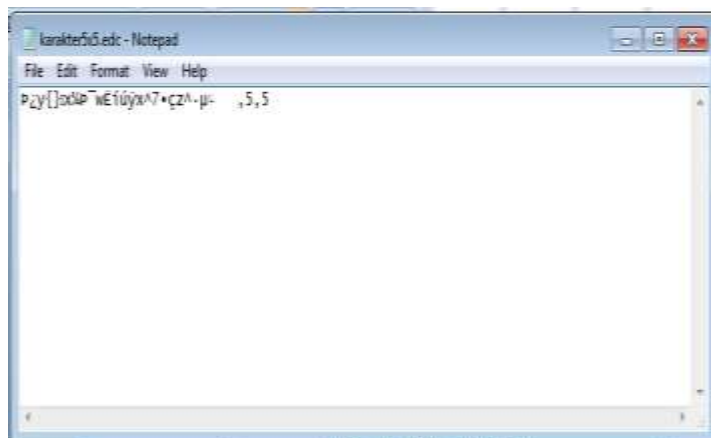
Gambar 3. Pembagian bit

Berdasarkan pembagian nilai biner di hasilkan 25 kelompok nilai biner baru yang sudah terkompresi. Nilai biner yang telah di kelompokkan dirubah kedalam suatu karakter. Sebelum dirubah terlebih dahulu mencari nilai decimal dari string bit tersebut dengan menggunakan kode ASCII. Karakter dari hasil kompresi di rubah dengan menggunakan excel agar lebih memudahkan untuk mencari karakter tersebut. Lalu karakter tersimpan dalam sebuah file dengan ekstensi .edc . Jika file dibuka dengan aplikasi notepad, maka karekter yang akan di tampilkan seperti berikut.



Gambar 4. Karakter Hasil Kompresi

Agar pada saat proses dekompresi nantinya dapat diketahui bahwa dimensi dari citra awal adalah 5x5 pixel maka akan ditambahkan string “,5,5” pada karakter hasil kompresi yang terdapat pada gambar 3.5 diatas. Karakter koma (,) yang pertama digunakan untuk memisahkan karakter hasil kompresi dengan karakter dimensi. Karakter “5” yang pertama digunakan untuk mengetahui nilai dari sumbu x yang terdapat pada citra awal. Karakter koma (,) yang kedua digunakan untuk memisahkan nilai sumbu x dengan nilai sumbu y pada citra awal. Karakter “5” yang kedua digunakan untuk mengetahui nilai dari sumbu y yang terdapat pada citra awal.



Gambar 5. Hasil Kompresi Dengan Penambahan Informasi Dimensi Citra Awal

Berdasarkan hasil kompresi dengan algoritma *Ternary Comma Code* dan penambahan nilai string sebesar 32 bit, maka perhitung kinerja kompresi sesuai parameter yang sudah ditentukan yaitu sebagai berikut :

a. *Ratio Of Compression (Rc)*

$$Rc = \frac{\text{Ukuran data sebelum dikompresi}}{\text{Ukuran data sesudah dikompresi}} = \frac{200}{121} = 1,65$$

b. *Compression Ratio (Cr)*

$$Cr = \frac{\text{Ukuran data sesudah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100\% = \frac{121}{200} \times 100\% = 60,5\%$$

2. Proses dekomposisi

Proses dekomposisi pada *file* gambar hal yang pertama dilakukan mencari informasi dimensi dari citra awal dengan cara mengambil karakter yang terdapat pada file hasil kompresi dimulai dari karakter yang diakhir sampai ketemu karakter koma (,) sebanyak 2 (dua) kali. Karakter pertama yang terdapat diakhir isi file hasil kompresi yaitu “5” menandakan itu adalah sumbu y pada citra awal. Karakter kedua yang terdapat diakhir isi file hasil kompresi yaitu koma (,) yang pertama dimana karakter tersebut merupakan karakter pemisah antara sumbu x dengan sumbu y pada citra awal. Karakter ketiga yang terdapat diakhir isi file hasil kompresi yaitu “5” yang menandakan itu adalah sumbu x pada citra awal. Dan Karakter yang keempat adalah karakter koma (,) yang kedua dimana karakter tersebut merupakan pemisah antara dimensi citra awal dengan *string* hasil kompresi. Karena sudah terdapat 2 (dua) karakter koma (,) maka informasi mengenai ukuran citra awal sudah diketahui yaitu *5x5 pixel*.



Gambar 6. Hasil Kompresi Yang Akan Didekomposisi

Proses dekomposisi selanjutnya yaitu mengembalikan karakter menjadi menjadi nilai decimal dan di rubah menjadi nilai biner(bit). Dalam proses pengembalian nilai biner di gunakan aplikasi excel.

11011110	10111111	11110011	11110111	11110111
01111001	01111011	01111101	01111111	01111000
11101101	11111010	11111111	01111000	01011110
10110111	10110101	11100111	10010101	11100111
01111010	01011110	00110111	10010101	11100111
01011110	10110101	1111		

Gambar 7. Pengembalian Karakter Ke Biner

Berdasarkan pada tabel di atas maka diambil seluruh nilai biner seperti berikut “11011110 10111111 01111001 01111011 01111101 01111111 01111000 10111100 11011110 10101111 01110111 11001011 11101101 11111010 11111111 01111000 01011110 00110111 10010101 11100111 01111010 01011110 10110111 10110101 11”. Proses pengambilan bit akhir yaitu dengan cara menyatukan seluruh biner dan menisahkan biner pixel dengan biner *ternary* yaitu seperti berikut “1 10111 10 10111 11 10111 100 1 0111 101 10111 110 10111 111 10111 1000 10111 1001 10111 1010 10111 1011 10111 1100 10111 1101 10111 1110 10111 1111 10111 10000 10111 10001 10111 10010 10111 10011 10111 10100 10111 10101 10111

10110 **10111** ". Nilai biner yang di hitam kan merupakan nilai dari *ternary* sedangkan yang tidak di hitamkan merupakan nilai dari biner pixel.

Berdasarkan hasil string bit di atas nilai biner dari "10111" di rubah dengan simbol *comma code* (c) dan masing-masing nilai biner di rubah menjadi nilai *ternary*. Maka dari perubahan itu akan dihasilkan nilai dari *ternary* "0c1c2c10c11c12c20c21c22c100c101c102c110c111c112c120c121c122c200c201c202c210c". Susun kembali nilai *ternary* dalam bentuk tabel.

Tabel 3. Nilai *Ternary*

No.	Nilai <i>Pixel</i>	<i>Ternary Comma Code</i>	Biner (Nol)
1	205	0c	1
2	230	1c	10
3	235	2c	11
4	144	10c	100
5	164	11c	101
6	167	12c	110
7	178	20c	111
8	186	21c	1000
9	191	22c	1001
10	200	100c	1010
11	213	101c	1011
12	216	102c	1100
13	220	110c	1101
14	223	111c	1110
15	224	112c	1111
16	226	120c	10000
17	228	121c	10001
18	231	122c	10010
19	233	200c	10011
20	234	201c	10100
21	236	202c	10101
22	239	210c	10110

Kembalikan nilai bit tersebut kebentuk semula berdasarkan tabel angka dan tabel kode yang telah di dapat dari hasil proses kompresi dan bentuk menjadi matriks dengan perkalian 5×5 *pixel* seperti pada tabel matriks dibawah ini, kemudian rubah menjadi citra :

Tabel 5. Matriks 5×5 *pixel*

223	235	233	236	239
205	226	230	231	235
186	213	224	228	234
167	191	205	220	230
144	164	178	200	216

4. KESIMPULAN

Sesuai dengan apa yang telah dibahas pada bab-bab sebelumnya, maka penulis mengambil kesimpulan sebagai berikut :

1. Mengetahui cara menguraikan prosedur pemampatan ukuran *file* gambar berdasarkan teknik kompresi.
2. Setelah diterapkan algoritma *ternary comma code* pada aplikasi kompresi *file* gambar dihasilkan ukuran *file* gambar yang lebih kecil sehingga memudahkan penggunaan untuk mengalokasikan ruang penyimpanan.
3. Dihasilkan aplikasi kompresi ukuran *file* gambar dengan menggunakan algoritma *ternary comma code* yang dapat mengatasi masalah tentang ukuran *file* gambar.

REFERENCES

- [1] C. T. Utari, "Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra", *J. TIMES*, vol. V, no. 2, pp. 24-31, 2016.
- [2] Iwan Fitrianto Rahmad, Helmi Kurniawan, "Kompresi File Citra Bitmap Menggunakan Algoritma Rle Dan Lz78", *CSRID Journal*, Vol.3 No.2 Juni 2011, Hal. 81 – 92.
- [3] D. Salomon, "Handbook of Data Compression", London. Springer-Verlag London Limited, 2010.
- [4] Hermawati, Fajar Astuti, "Pengolahan Citra Digital Konsep dan Teori", Yogyakarta, CV. Andi Offset, 2013.
- [5] T. Sutoyo, S.SI., Dkk, "Teori Pengolahan Citra Digital", Yogyakarta, CV. Andi Offset, 2009.
- [6] Putra, Darma, "Pengolahan Citra Digital", Yogyakarta, CV. Andi Offset, 2010.

- [7] Dr.Suarga, M.Sc., M.Math., Ph.D, “*Algoritma Dan Pemograman*”, Yogyakarta, CV. Andi Offset, 2012.
- [8] Sutabri, Tata, “*Konsep Sistem Informasi*”, Yogyakarta, CV. Andi Offset, 2012.
- [9] Derwin Suhartono, Afan Galih Salman, Christian Octavianus, “*Aplikasi Penyembunyian Pesan Pada Citra Jpeg Dengan Algoritma F5 Dalam Perangkat Mobile Berbasis Android*”, Seminar Nasional Aplikasi Teknologi Informasi 2012 (SNATI 2012), Yogyakarta, 15-16 Juni 2012.
- [10] S, Rosa A, Shalahuddin, M, “*Rekayasa Perangkat Lunak Terstruktur Dan Berorientasi Objek*”, Bandung, Informatika bandung, 2016.
- [11] Irwanto, Djon, “*Perancangan Object Oriented Software Dengan UML*”, Yogyakarta, CV. Andi Offset, 2006.
- [12] Nugroho, Adi, “*Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP (Unified Software Development Process)*”, Yogyakarta, CV. Andi Offset, 2010.
- [13] Komputer, Wahana, “*Membuat Aplikasi Client Server dengan Visual Basic 2008*”, Yogyakarta, CV. Andi Offset, 2010.