PENGEMBANGAN DAN ANALISIS KOMBINASI *RUN LENGTH ENCODING* DAN *RELATIVE ENCODING* UNTUK KOMPRESI CITRA

Yosia Adi Jaya[1]
yosia.adi@ti.ukdw.ac.id

Lukas Chrisantyo[2]
lukaschris@ti.ukdw.ac.id

Willy Sudiarto Raharjo[3]
willysr@ti.ukdw.ac.id

Abstract

Data Compression can save some storage space and accelerate data transfer. Among many compression algorithm, Run Length Encoding (RLE) is a simple and fast algorithm. RLE can be used to compress many types of data. However, RLE is not very effective for image lossless compression because there are many little differences between neighboring pixels. This research proposes a new lossless compression algorithm called YRL that improve RLE using the idea of Relative Encoding. YRL can treat the value of neighboring pixels as the same value by saving those little differences / relative value separately. The test done by using various standard image test shows that YRL have an average compression ratio of 75.805% for 24-bit bitmap and 82.237% for 8-bit bitmap while RLE have an average compression ratio of 100.847% for 24-bit bitmap and 97.713% for 8-bit bitmap.

Keywords: *compression*, *lossless*, *RLE*, *relative*, YRL

1. Pendahuluan

Kompresi data sudah banyak diterapkan dalam teknologi informasi. Dalam hal penyimpanan, kompresi data dapat mengurangi penggunaan kapasitas penyimpanan. Dalam proses pertukaran data, kompresi data dapat mempercepat proses tukar data dan mengurangi penggunaan sumber daya (jika melalui jaringan Internet). Manfaat-manfaat tersebut akan menghemat biaya penyimpanan dan pertukaran data. Data yang semakin besar dan kebutuhan bertukar data yang semakin sering membutuhkan kompresi data yang cepat dan efektif. Salah satu kompresi data yang cepat adalah *Run Length Encoding (RLE)*.

RLE mengubah nilai yang sama berturut-turut menjadi 2 nilai saja, yaitu jumlah nilai yang sama dan nilai itu sendiri. Konsep *RLE* yang sederhana dan umum memungkinkan *RLE* untuk mengompresi banyak jenis data. Citra merupakan salah satu jenis data yang dapat dikompres menggunakan *RLE*. Hasil kompresi *lossy* citra dengan menggunakan *RLE* cukup baik. Akan tetapi, hasil kompresi *lossless* citra dengan menggunakan *RLE* kurang baik karena piksel-piksel yang berdekatan pada citra memiliki nilai warna yang hampir sama (tidak sama persis).

Di sisi lain, ada *Relative Encoding* yang cara kerjanya adalah menyimpan nilai relatif dari nilai sebelumnya. Dari cara kerja *RLE* dan *Relative Encoding*, penulis mendapat gagasan mengombinasikan kedua teknik kompresi tersebut untuk melakukan kompresi *lossless* citra sederhana. Dalam penelitian ini, citra yang akan digunakan adalah citra *bitmap* yang tidak menggunakan kompresi agar dapat mengetahui efektivitas dari teknik kompresi pada penelitian ini saja.

[1] Universitas Kristen Duta Wacana , Fakultas Teknologi Informasi, Program Studi Teknik Infomatika.

[2] Universitas Kristen Duta Wacana , Fakultas Teknologi Informasi, Program Studi Teknik Infomatika.

[3] Universitas Kristen Duta Wacana , Fakultas Teknologi Informasi, Program Studi Teknik Infomatika.

2. Landasan Teori

2.1. Tinjauan Pustaka

Ada beberapa penelitian yang berusaha mengembangkan *RLE* dengan menambahkan metode-metode tertentu. Metode yang pernah dilakukan antara lain adalah melakukan preproses pada input citra, melakukan kompresi yang berbeda berdasarkan keadaan tertentu, dan melakukan perubahan struktur data input. Berikut adalah contoh penelitian-penelitian tersebut.

Arota Eka Setiawan menerapkan *Burrows-Wheeler Transform (BWT)* terhadap input citra lalu mengompresi citra dengan menggunakan *RLE*. Kemudian hasil kompresi citra dengan didahului BWT dibandingkan dengan kompresi citra tanpa BWT. Hasil penelitian menunjukkan bahwa rasio kompresi citra dengan didahului BWT lebih baik dari rasio kompresi citra tanpa BWT. (Setiawan, 2014)

Dalam penelitian berjudul "*A New Method Which Combines Arithmetic Coding with RLE for Lossless Image Compression*" dilakukan kompresi yang berbeda berdasarkan kemiripan nilai dengan nilai sebelumnya. Suatu piksel dikompres dengan *RLE* jika nilainya sama dengan piksel di kiri dan atasnya, jika tidak maka akan dikompres dengan *Arithmetic Coding*. Algoritma gabungan tersebut memiliki rasio kompresi yang lebih baik dari *Arithmetic Coding* biasa. (Abdmouleh, Masmoudi, & Bouhlel, 2012)

I Made Agus Dwi Suarjaya mengusulkan suatu algoritma baru yang dinamai *j-bit encoding (JBE)* untuk mengoptimasi kompresi data. Input data yang telah diubah dengan JBE dikompres dengan berbagai macam algoritma kompresi. Hasil penelitian menunjukkan perbaikan rasio kompresi yang sangat signifikan untuk input citra. (Suarjaya, 2012)

2.2. Kompresi

Kompresi data adalah proses konversi data input ke data lain yang memiliki ukuran lebih kecil (Salomon, 2004). Kompresi dibagi menjadi 2, yaitu:

- Kompresi *Lossless* (tidak menghilangkan informasi apapun, sehingga hasil dekompresi sama persis dengan bentuk asli)
- Kompresi *Lossy* (menghilangkan detail yang kurang penting, tetapi hasil dekompresi masih mewakili bentuk asli)

Ada beberapa nilai yang digunakan dalam pengukuran kompresi, yaitu:

- *Compression ratio* (ukuran hasil kompresi / ukuran asli)
- *Compression factor* (ukuran asli / ukuran hasil kompresi)
- *Bit per pixel* (pada kompresi citra)
- *The speed of compression* / kecepatan kompresi (diukur dengan *cycles per byte (CPB)*)

Pada kompresi citra, ada beberapa urutan pembacaan data yang sering digunakan, yaitu: *Non-interlaced vertical*, *Non-interlaced horizontal*, *Non-interlaced zigzag*, dan *Interlaced*

2.3. *RLE*

Ide dibalik *RLE* adalah: jika data item D terjadi N kali berturut-turut dalam input, ganti N kejadian menjadi pasangan ND. Langkah-langkah kompresi *lossless RLE* pada citra adalah sebagai berikut:

[1] Tentukan jumlah N maksimum dalam konstanta max.
[2] Input nilai, masukkan dalam variabel D.
[3] Set variabel N = 0.
[4] Input nilai selanjutnya, masukkan dalam variabel temp.
    [4.1] Jika temp sama dengan D, maka N ditambah 1.
        [4.1.1] Jika N < max, kembali ke langkah 4.
        [4.1.2] Jika N >= max, ouput N D. Kembali ke langkah 2.
    [4.2] Jika temp tidak sama dengan D, output N D.
    [4.3] Masukkan temp ke D, set N = 0, kembali ke langkah 4.

Misal : 12, 12, 12, 12, 12, 12, 12, 87, 87, 87, 5, 5, 5, 5, 5, 5 dapat dikompres menjadi (7) 12, (3) 87, (6) 5

2.4. Relative Encoding

Relative Encoding digunakan dalam kasus dimana input berupa rentetan angka yang nilainya tidak berbeda jauh. Angka pertama disimpan apa adanya lalu angka selajutnya merupakan selisih dengan nilai sebelumnya. Langkah-langkah kompresi *Relative Encoding* pada citra adalah sebagai berikut:

[1] Tentukan selisih maksimum dalam konstanta maxdiff.
[2] Input nilai, masukkan dalam variabel prev.
[3] Output prev.
[4] Input nilai, masukkan dalam variabel value.
[5] Hitung value – prev, masukkan dalam variabel diff.
    [5.1] Jika | diff | <= maxdiff, output diff.
    [5.2] Jika | diff | > maxdiff, output value.
[6] Masukkan value ke dalam prev. Kembali ke langkah 4.

Misal : 70, 71, 72.5, 73.1 dapat dikompres menjadi 70, 1, 1.5, 0.6

2.5. Struktur file bitmap

Struktur file bitmap sangat sederhana dan terdiri dari *header* file bitmap, *header* informasi bitmap, tabel warna (jika ada), dan *array* byte yang menyusun gambar bitmap (Whitrow, 2008).

Header file bitmap berisi informasi tentang tipe, ukuran, dan *layout* dari file bitmap. dua byte pertama adalah karakter B dan M yang menunjukkan tipe file. Empat byte selanjutnya menampung ukuran file dalam byte. Empat byte selanjutnya tidak dipakai dan diisi 0. Empat byte yang terakhir menampung jarak antara *header* dengan titik awal data bitmap dalam byte. Secara formal, dapat ditulis sebagai berikut:

BITMAPFILEHEADER {
    uint 2 bytes file type
    dword 4 bytes file size in bytes
    uint 2 bytes reserved
    uint 2 bytes reserved
    dword 4 bytes offset to data in bytes
} **BITMAPFILEHEADER;**

Header informasi bitmap berisi dimensi, tipe kompresi, dan format warna dari bitmap. Secara formal, *header* informasi bitmap adalah sebagai berikut:

BITMAPINFOHEADER {
    dword 4 bytes needed for BITMAPINFOHEADER structuresize
    long 4 bytes bitmap width in pixels
    long 4 bytes bitmap height in pixel
    word 2 bytes 1
    word 2 bytes bits/pixel (1 = monochrome)
    dword 4 bytes compression 0, 8, 4
    dword 4 bytes image size in bytes (may be 0 for monochrome)
    long 4 bytes pixels/meter
    long 4 bytes pixels/meter
    dword 4 bytes number of colour indexes used by bitmap in colour table
    dword 4 bytes number of colour indexes considered important
} **BITMAPINFOHEADER;**

Tabel warna digunakan untuk merepresentasikan warna yang lebih dari 8-bit. Tabel warna tidak terdapat pada bitmap 24-bit karena tiap piksel masih dapat direpresentasikan dengan 8-bit *blue-green-red (BGR)*.

Data bitmap terdiri dari bytes yang mewakili piksel-piksel dari kiri ke kanan dalam tiap baris (*scan lines*). *Scan lines* dalam bitmap disimpan dari bawah ke atas. Berarti nilai pertama dalam *array* mewakili piksel di pojok kiri bawah gambar.

3. Metodologi Penelitian

3.1. Persiapan

- Mempelajari dan mendalami teori dan implementasi *RLE*, *Relative Encoding*, dan struktur file bitmap.
- Membuat *encoder* dan *decoder* dengan metode *RLE* untuk memperkuat pemahaman.

3.2. Pengembangan

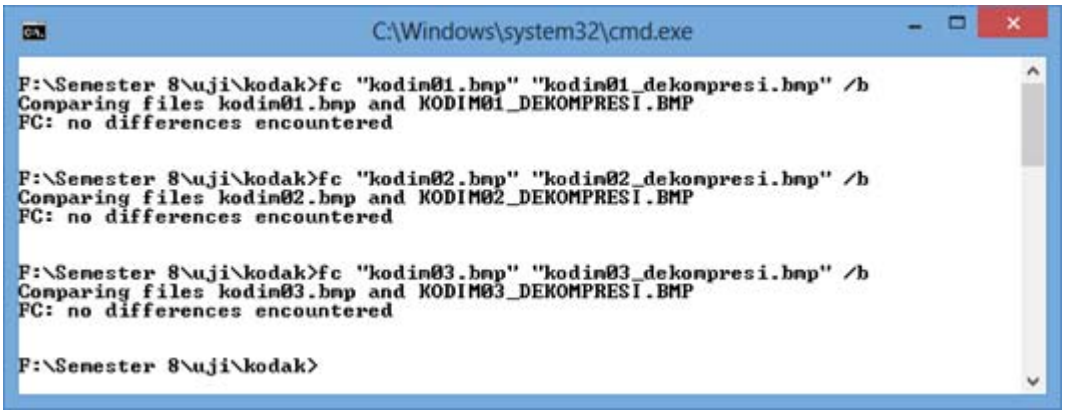
- Merancang algoritma untuk mengombinasikan *RLE* dan *Relative Encoding* (kombinasi akan berupa satu algoritma yang memiliki karakteristik seperti *RLE* dan *Relative Encoding*, tetapi bukan keduanya dan bukan *RLE* yang dilanjutkan dengan *Relative Encoding* ataupun sebaliknya).
- Membuat *encoder* dan *decoder* yang mengimplementasikan algoritma yang telah dirancang.

3.3. Pengujian

- Mengumpulkan *standard image test* (citra digital yang sering digunakan untuk menguji algoritma kompresi citra). *Standard image test* yang digunakan berasal dari http://r0k.us/graphics/kodak/, http://imagecompression.info/test_images/, dan http://sipi.usc.edu/database/database.php?volume=misc.
- Mengompresi citra digital yang telah dikumpulkan dengan algoritma yang dikembangkan dan dengan *RLE* murni.
- Membandingkan rasio kompresi dari kedua hasil kompresi tersebut.

4. Hasil Penelitian dan Analisis

Sistem yang sudah dibangun diuji dengan cara mengompresi 55 bitmap 24-bit dan 22 bitmap 8-bit lalu hasil kompresi didekompres dan dibandingkan dengan file bitmap awal. Hasil perbandingan menunjukkan bahwa kedua file sama persis untuk semua gambar. Hal ini menunjukkan sistem telah berhasil diimplementasikan dan berfungsi dengan benar. Beberapa contoh hasil perbandingan dapat dilihat pada Gambar 1.



Gambar 1. Perbandingan antara file awal dan file hasil dekompresi

Gambar-gambar yang digunakan untuk menguji sistem, juga dikompres dengan *RLE* sebagai pembanding rasio kompresi sistem. Data-data ukuran file bitmap, ukuran file *RLE*, ukuran file yrl, serta rasio kompresi *RLE* dan rasio kompresi yrl dapat dilihat pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1.
Data Hasil Pengujian Bitmap 24-bit

No	Nama file	Ukuran file (byte)			Rasio kompresi	
		bmp	*RLE*	yrl	*RLE*	yrl
1	kodim01	1,179,702	1,284,912	1,100,299	108.918%	93.269%
2	kodim02	1,179,702	1,255,117	851,654	106.393%	72.192%
3	kodim03	1,179,702	1,179,836	782,551	100.011%	66.335%
4	kodim04	1,179,702	1,272,402	951,179	107.858%	80.629%
5	kodim05	1,179,702	1,283,554	1,093,319	108.803%	92.678%
6	kodim06	1,179,702	1,210,747	937,363	102.632%	79.458%
7	kodim07	1,179,702	1,212,656	841,723	102.793%	71.350%
8	kodim08	1,179,702	1,283,728	1,100,941	108.818%	93.324%
9	kodim09	1,179,702	1,266,408	836,605	107.350%	70.917%
10	kodim10	1,179,702	1,264,248	874,203	107.167%	74.104%
11	kodim11	1,179,702	1,248,152	924,237	105.802%	78.345%
12	kodim12	1,179,702	1,219,450	797,241	103.369%	67.580%
13	kodim13	1,179,702	1,289,209	1,186,333	109.283%	100.562%
14	kodim14	1,179,702	1,281,070	1,026,561	108.593%	87.019%
15	kodim15	1,179,702	1,189,956	850,063	100.869%	72.057%
16	kodim16	1,179,702	1,231,413	862,461	104.383%	73.108%
17	kodim17	1,179,702	1,266,282	914,529	107.339%	77.522%
18	kodim18	1,179,702	1,306,803	1,063,211	110.774%	90.125%
19	kodim19	1,179,702	1,286,997	977,031	109.095%	82.820%
20	kodim20	1,179,702	862,876	634,617	73.144%	53.795%
21	kodim21	1,179,702	1,272,554	904,891	107.871%	76.705%
22	kodim22	1,179,702	1,284,118	982,821	108.851%	83.311%
23	kodim23	1,179,702	1,235,164	842,319	104.701%	71.401%
24	kodim24	1,179,702	1,215,057	993,662	102.997%	84.230%
25	artificial	978,654	713,141	587,625	72.870%	60.044%
26	big_building	1,102,554	1,193,695	1,018,030	108.266%	92.334%
27	big_tree	1,098,354	1,159,849	958,080	105.599%	87.229%
28	bridge	2,165,154	2,231,471	1,664,958	103.063%	76.898%
29	cathedral	2,209,254	2,328,398	1,735,207	105.393%	78.543%
30	deer	959,754	829,612	550,769	86.440%	57.386%
31	fireworks	1,102,554	445,874	387,187	40.440%	35.117%
32	flower_foveon	978,654	831,363	605,065	84.950%	61.826%
33	hdr	978,654	876,884	628,470	89.601%	64.218%
34	leaves_iso_200	976,554	1,082,990	973,482	110.899%	99.685%
35	leaves_iso_1600	976,554	1,085,704	981,604	111.177%	100.517%
36	nightshot_iso_100	1,102,554	761,116	597,752	69.032%	54.215%
37	nightshot_iso_1600	1,102,554	1,102,022	712,778	99.952%	64.648%

No	Nama file	Ukuran file (byte)			Rasio kompresi	
		bmp	*RLE*	yrl	*RLE*	yrl
38	spider_web	982,854	930,910	661,040	94.715%	67.257%
39	zone_plate	978,654	593,822	597,185	60.677%	61.021%
40	4.1.01	196,662	211,392	161,257	107.490%	81.997%
41	4.1.02	196,662	200,521	151,201	101.962%	76.884%
42	4.1.03	196,662	207,193	113,074	105.355%	57.497%
43	4.1.04	196,662	213,853	159,245	108.741%	80.974%
44	4.1.05	196,662	211,958	142,545	107.778%	72.482%
45	4.1.06	196,662	212,178	168,320	107.890%	85.588%
46	4.1.07	196,662	185,636	105,704	94.393%	53.749%
47	4.1.08	196,662	190,376	119,602	96.804%	60.816%
48	4.2.01	786,486	824,146	508,579	104.788%	64.665%
49	4.2.02	786,486	738,727	558,127	93.928%	70.965%
50	4.2.03	786,486	880,195	812,562	111.915%	103.316%
51	4.2.04	786,486	871,670	661,361	110.831%	84.091%
52	4.2.05	786,486	850,702	559,457	108.165%	71.134%
53	4.2.06	786,486	873,930	703,041	111.118%	89.390%
54	4.2.07	786,486	854,309	654,178	108.624%	83.177%
55	house	786,486	833,126	619,764	105.930%	78.802%

Tabel 2.
Data Hasil Pengujian Bitmap 8-bit

No	Nama file	Ukuran file (byte)			Rasio kompresi	
		bmp	*RLE*	yrl	*RLE*	yrl
56	5.1.09	66,616	68,128	61,672	102.270%	92.578%
57	5.1.10	66,616	68,124	66,369	102.264%	99.629%
58	5.1.11	66,616	67,616	41,549	101.501%	62.371%
59	5.1.12	66,616	67,752	44,935	101.705%	67.454%
60	5.1.13	66,616	16,884	15,317	25.345%	22.993%
61	5.1.14	66,616	67,984	62,107	102.054%	93.231%
62	5.2.08	263,224	267,332	215,930	101.561%	82.033%
63	5.2.09	263,224	266,676	235,264	101.311%	89.378%
64	5.2.10	263,224	260,576	255,831	98.994%	97.191%
65	5.3.01	1,049,656	1,059,436	898,586	100.932%	85.608%
66	5.3.02	1,049,656	1,065,680	967,806	101.527%	92.202%
67	7.1.01	263,224	266,724	221,559	101.330%	84.171%

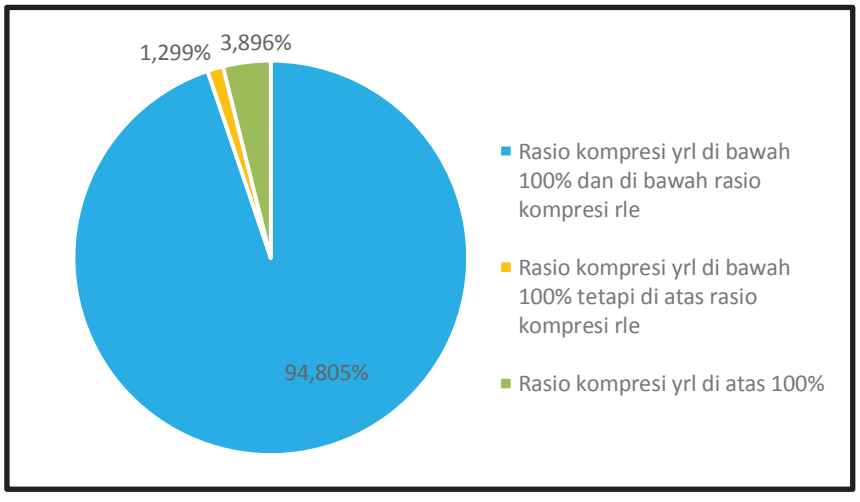
Tabel 4.
Data Hasil Pengujian Bitmap 8-Bit (lanjutan)

No	Nama file	Ukuran file (byte)			Rasio kompresi	
		bmp	*RLE*	yrl	*RLE*	yrl
68	7.1.02	263,224	249,540	152,326	94.801%	57.869%
69	7.1.03	263,224	267,152	223,946	101.492%	85.078%
70	7.1.04	263,224	266,620	221,971	101.290%	84.328%
71	7.1.05	263,224	267,764	249,714	101.725%	94.867%
72	7.1.06	263,224	267,944	251,785	101.793%	95.654%
73	7.1.07	263,224	268,000	257,245	101.814%	97.729%
74	7.1.08	263,224	266,720	202,595	101.328%	76.967%
75	7.1.09	263,224	267,396	234,296	101.585%	89.010%
76	7.1.10	263,224	267,436	227,448	101.600%	86.409%
77	7.2.01	1,049,656	1,065,048	760,630	101.466%	72.465%

Dari data hasil pengujian, secara keseluruhan dapat dilihat bahwa sistem berhasil memperkecil 96.104% gambar uji (74 dari 77 gambar), sedangkan *RLE* hanya berhasil memperkecil 20.779% gambar uji (16 dari 77 gambar). Selain itu, hanya ada 1 file dimana rasio kompresi *RLE* lebih kecil dari pada rasio kompresi sistem. Hal ini menunjukkan bahwa secara umum sistem yang mengimplementasikan YRL lebih baik dari pada sistem yang mengimplementasikan *RLE* dalam mengompresi file bitmap.

4.1. Analisis Pengujian Bitmap 24-bit

Untuk menganalisis sistem, hasil pengujian bitmap 24-bit dapat dibagi menjadi 3 kelompok. Ketiga kelompok hasil tersebut adalah rasio kompresi sistem di bawah 100% dan di bawah rasio kompresi *RLE*, rasio kompresi sistem di bawah 100% tetapi di atas rasio kompresi *RLE*, dan rasio kompresi sistem di atas 100%. Persentase ketiga kelompok hasil tersebut dapat dilihat pada Gambar 2



Gambar 2. Persentase 3 kelompok hasil pengujian bitmap 24-bit

Gambar yang memiliki rasio kompresi sistem di bawah 100% dan di bawah rasio kompresi *RLE* berjumlah 51 gambar. Diantara gambar tersebut, fireworks.bmp dan kodim20.bmp memiliki rasio kompresi yang paling rendah, yaitu sebesar 35.117% dan 53.795%. Kedua gambar tersebut memiliki ciri yang sama, yaitu memiliki banyak warna yang

hampir sama. Pada fireworks.bmp, hampir seluruh gambar berwarna hitam dan hanya ada sedikit warna putih gambar kilatan cahaya kembang api. Pada kodim20.bmp, ada banyak warna putih pada gambar langit dan banyak warna hijau pada gambar rumput. Hal inilah yang menyebabkan kedua gambar tersebut memiliki rasio kompresi yang rendah.

Gambar yang memiliki rasio kompresi sistem di bawah 100% tetapi di atas rasio kompresi *RLE* berjumlah 1 gambar. Gambar tersebut adalah zone_plate.bmp. Gambar zone_plate.bmp memiliki warna latar yang *solid* (tidak ada gradasi warna) dan sedikit motif. Dengan warna latar yang *solid*, baik *RLE* maupun sistem dapat mengompresnya. Akan tetapi, jumlah data berturut-turut yang dapat disimpan *RLE* (256) lebih banyak dari sistem (64). Oleh sebab itu, jika ada 256 data yang sama berturut-turut, sistem akan membutuhkan ruang 4 kali lebih banyak dari yang dibutuhkan *RLE*. Hal inilah yang menyebabkan rasio kompresi sistem lebih besar dari pada rasio kompresi *RLE*.

Gambar yang memiliki rasio kompresi sistem di atas 100% berjumlah 3 gambar. Gambar tersebut adalah kodim13.bmp, leaves_iso_1600.bmp, dan 4.2.03.bmp. Ketiganya memiliki ciri yang sama, yaitu adanya banyak kombinasi warna yang kontras (memiliki perbedaan warna yang tajam). Pada kodim13.bmp terdapat banyak kombinasi warna kontras pada gambar salju di gunung dan riak air di sungai. Pada leaves_iso_1600.bmp terdapat banyak kombinasi warna kontras pada gambar sinar matahari yang menerobos melalui dedaunan. Pada 4.2.03.bmp terdapat banyak warna kontras pada gambar bulu binatang. Warna kontras berarti memiliki selisih RGB yang jauh, sehingga sistem tidak dapat menampung nilai selisih tersebut dan terpaksa menyimpannya secara terpisah. Dalam sistem, terdapat penanda yang membedakan byte yang dikompresi dan yang tidak dikompresi. Jadi jika ada banyak byte yang tidak bisa dikompresi, ukuran file justru akan bertambah karena adanya penanda tersebut.

4.2. Analisis Pengujian Bitmap 8-bit

Hasil pengujian bitmap 8-bit menunjukkan bahwa rasio kompresi sistem di bawah 100% dan di bawah rasio kompresi *RLE* untuk semua data uji. Akan tetapi, rata-rata rasio kompresi sistem pada bitmap 8-bit (82.237%) lebih tinggi dari rata-rata rasio kompresi sistem pada bitmap 24-bit (75.805%). Hal ini disebabkan karena bitmap 8-bit hanya memiliki 1 channel warna sehingga jika ada perbedaan warna yang tajam, pasti ada perbedaan nilai yang besar pada channel tersebut. Sedangkan pada bitmap 24-bit terdapat 3 channel warna sehingga jika ada perbedaan warna yang tajam, kemungkinan perbedaan nilai yang besar hanya terdapat pada salah satu channel warna.

Diantara hasil pengujian bitmap 8-bit, 5.1.13.bmp memiliki rasio kompresi paling kecil (22.993%). Gambar 5.1.13.bmp memiliki warna latar putih solid dan ada sejumlah garis pendek yang melintang dan membujur. Oleh karena itu, *RLE* juga dapat mengompresi gambar tersebut dengan baik, dengan rasio kompresi 25.345%. Rasio kompresi gambar tersebut tidak seperti rasio kompresi gambar zone_plate.bmp meskipun memiliki ciri yang sama. Rasio kompresi sistem pada gambar tersebut masih lebih kecil dari rasio kompresi *RLE* karena ukuran gambar kecil (256 x 256 pixel). Jika dalam satu baris memiliki warna yang sama semua, *RLE* membutuhkan 2 byte dan sistem membutuhkan 8 byte, tetapi adanya garis-garis melintang dan membujur pada gambar tersebut menyebabkan kemungkian kejadian di atas (satu baris memiliki warna yang sama semua) menjadi kecil.

4. Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

a. Penambahan ide dasar *Relative Encoding* dapat memperbaiki hasil kompresi *RLE*. Algoritma yang dikembangkan memiliki rata-rata rasio kompresi sebesar 75.805% untuk bitmap 24-bit dan 82.237% untuk bitmap 8-bit. Nilai tersebut lebih baik dari rata-rata rasio kompresi *RLE* murni yang nilainya sebesar 100.847% untuk bitmap 24-bit dan 97.713% untuk bitmap 8-bit.
b. Dalam kondisi tertentu *RLE* dapat mengompresi gambar lebih baik dari algoritma yang dikembangkan karena struktur penyimpanan yang digunakan dalam implementasi

algoritma ini. Rasio kompresi *RLE* lebih rendah ketika banyak warna *solid* pada gambar dan ukuran gambar tidak terlalu kecil, sedangkan rasio kompresi algoritma yang dikembangkan lebih rendah ketika banyak gradasi warna pada gambar.

c. Jika gambar memiliki banyak kemiripan warna, rasio kompresi akan semakin kecil / semakin baik. Jika gambar memiliki banyak warna yang kontras, ada kemungkinan ukuran file hasil kompresi akan membesar / rasio kompresi lebih besar 100%.

5. Saran

Sistem yang dikembangkan dalam penelitian ini masih sangat terbatas. Beberapa hal yang dapat dikembangkan lebih lanjut adalah:

a. Urutan pembacaan bitmap.
b. Sistem menggunakan urutan *non-interlaced horizontal*. Pengujian lebih lanjut
c. dapat membandingkan hasil kompresi jika kompresi dilakukan dengan urutan-urutan pembacaan file yang ada (*horizontal*, *vertical*, dan *zigzag*).
d. Struktur penyimpanan byte terkompresi.
e. Hal ini juga dapat diteliti lebih lanjut untuk menemukan struktur yang lebih efektif dalam menyimpan nilai-nilai hasil kompresi.
f. Mengubah cara menentukan rentetan data dari *Greedy* menjadi *Dynamic Programming* agar kompresi menjadi optimal.

Daftar Pustaka

Abdmouleh, M. K., Masmoudi, A., & Bouhlel, M. S. (2012). A New Method Which Combines Arithmetic Coding with RLE for Lossless Image Compression. *Journal of Software Engineering and Applications*, 41-44.

Franzen, R. (2013, January 27). *True Color Kodak Images.* Retrieved from http://r0k.us/graphics/kodak/

Rawzor - Lossless compression software for camera raw images. (n.d.). *The New Test Images - Image Compression Benchmark.* Retrieved from http://imagecompression.info/test_images/

Salomon, D. (2004). *Data Compression, The Complete Reference, 3rd edition.* New York: Springer.

Setiawan, A. E. (2014). *Implementasi Penggabungan Algoritma Run-Length Encoding dan Metode Burrows-Wheeler Transform Pada Pemampatan Citra BMP 24-Bit.* Yogyakarta: Universitas Kristen Duta Wacana.

Suarjaya, I. M. (2012). A New Algorithm for Data Compression. *IJACSA, 3(8).*

University of Southern California. (n.d.). *SIPI Image Database - Misc*. Retrieved from http://sipi.usc.edu/database/database.php?volume=misc

Whitrow, R. (2008). *OpenGL Graphics Through Applications.* London: Springer-Verlag.