

NASKAH PUBLIKASI

**KOMPRESI CITRA DENGAN METODE *ARITHMETIC CODING*
DALAM KAWASAN *ENTROPY CODING***



Disusun Sebagai Salah Satu Syarat Menyelesaikan Program Studi Strata 1
Jurusan Elektro Fakultas Teknik
Universitas Muhammadiyah Surakarta

Disusun Oleh:

CAHYO HENDI PRASETYO

D 400 080 012

**FAKULTAS TEKNIK JURUSAN TEKNIK ELEKTRO
UNIVERSITAS MUHAMMADIYAH SURAKARTA**

2013

LEMBAR PERSETUJUAN

Naskah publikasi dengan judul “Kompresi Citra Dengan Metode *Arithmetic Coding* Dalam Kawasan *Entropy Coding*” ini diajukan oleh:

NAMA : CAHYO HENDI PRASETYO

NIM : D 400 080 012

Guna memenuhi salah satu syarat untuk menyelesaikan program Sarjana jenjang pendidikan Strata-Satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Muhammadiyah Surakarta, telah diperiksa dan disetujui pada:

Hari :

Tanggal :

Mengetahui,

Pembimbing 1



(Ratnasari Nur Rohmah, S.T., M.T.)

Pembimbing 2



(Muhammad Kusban, S.T., M.T.)

**SURAT PERNYATAAN PUBLIKASI
KARYA ILMIAH**

Bismillahirrahmanirrohim

Yang bertanda tangan di bawah ini, saya

Nama : Cahyo Hendi Prasetyo
NIM : D400080012
Fajultas / Jurusan : Teknik / Elektro
Jenis : Skripsi
Judul : Kompresi Citra Dengan Metode Arithmetic
Coding Dalam Kawasan Entropy Coding

Dengan ini menyatakan bahwa saya menyetujui untuk

1. Memberikan hak bebas royalty kepada perpustakaan UMS atas penulisan karya ilmiah saya, demi pengembangan ilmu pengetahuan.
2. Memberikan hak menyimpan, mengalih mediakan / mengalih formatkan, mengelola dalam bentuk pangkalan data (*database*), mendistribusikannya, serta menampilkannya dalam bentuk *sofcopy* untuk kepentingan akademis pada perpustakaan UMS, tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis / pencipta.
3. Bersedia dan menjamin untuk menanggung secara pribadi tanpa melibatkan pihak perpustakaan UMS, dari semua bentuk tuntutan hukum yang timbul atas pelanggaran hak cipta dalam karya ilmiah ini.

Demikian pernyataan ini saya buat dengan sesungguhnya dan semoga dapat digunakan sebagaimana semestinya.

Surakarta, Februari 2013
Yang menyatakan,



(Cahyo Hendi Prasetyo)

KOMPRESI CITRA DENGAN METODE ARITHMETIC CODING DALAM KAWASAN ENTROPY CODING

**CAHYO HENDI PRASETYO
FAKULTAS TEKNIK JURUSAN TEKNIK ELEKTRO
UNIVERSITAS MUHMMADIYAH SURAKARTA
E-mail : cahyohendiprasetyo@yahoo.co.id**

ABSTRAKSI

Perkembangan media penyimpan berkapasitas besar mengakibatkan orang tidak lagi menemui masalah jika mempunyai file dengan ukuran yang besar. Lebih-lebih jika file yang kita punya merupakan file citra. Walaupun demikian, adakalanya ukuran file yang besar tersebut terasa mengganggu jika kita harus manage media penyimpan yang kita punya untuk bermacam-macam data. Kompresi data menjadi cara yang efisien untuk mengatasi permasalahan di atas. Ada dua tipe kompresi data, yaitu kompresi lossy dan kompresi lossles. Kompresi citra bersifat lossy, citra mengalami penurunan akibat proses yang terjadi didalamnya. Sedangkan pada kompresi citra yang bersifat lossles, citra tidak mengalami penurunan.

Ada banyak sekali metode kompresi data yang ada saat ini, namun pada tugas akhir ini akan dibahas tentang salah satu teknik kompresi data Arithmetic Coding pada kompresi citra. Metode Arithmetic Coding ini menggantikan satu deretan simbol input dengan sebuah bilangan floating point. Semakin panjang dan semakin kompleks pesan yang dikodekan, semakin banyak bit yang diperlukan untuk keperluan tersebut. Output dari arithmetic coding ini adalah satu angka yang lebih kecil dari 1 dan lebih besar atau sama dengan 0. Angka ini secara unik dapat di-decode sehingga menghasilkan deretan simbol yang dipakai untuk menghasilkan angka tersebut. Untuk menghasilkan angka output tersebut, tiap simbol yang akan diencode diberi satu set nilai probabilitas.

Dari penelitian yang penulis lakukan, dapat disimpulkan bahwa kompresi menggunakan algoritma Arithmetic Coding dapat menghasilkan citra dengan ukuran file yang lebih kecil. Namun apabila citra uji yang digunakan mempunyai ukuran file yang besar atau di atas 151 x 151 pixel, waktu yang dibutuhkan untuk melakukan proses kompresi maupun dekompresi membutuhkan waktu yang cukup lama. Dari hasil pengamatan, tingkat rasio kompresi tidak dipengaruhi oleh besarnya ukuran file, melainkan tergantung komponen warna gambar yang bersangkutan. Dan setelah dilakukan perhitungan rasio kompresi didapatkan berkisar antara 1,887-5,715, dimana gambar dengan nama bawahlaut yang mempunyai ukuran file 50400 Byte rasionya paling kecil, yaitu 1,887. Sedangkan gambar dengan nama MU yang ukuran filenya 60700 Byte mempunyai rasio yang paling besar, yaitu 5,715. Dari hasil penelitian yang penulis amati, citra uji dengan citra hasil rekonstruksi memiliki kualitas yang sama.

Kata Kunci : Citra, Arithmetic Coding, Kompresi, dekompresi, Lossles

1. PENDAHULUAN

Kebutuhan manusia terhadap suatu gambar atau citra (*image*) sejak dahulu merupakan suatu kebutuhan yang tidak bisa diabaikan begitu saja. Sejak ditemukannya alat untuk menangkap gambar seperti kamera, perkembangan teknologi semakin maju. Namun tidak hanya terfokus pada alat-alat untuk menangkap gambar itu saja, tapi juga pada teknologi untuk mengolah gambar itu sendiri. Pada perkembangan saat ini, gambar tidak hanya digunakan untuk suatu dokumentasi kejadian yang dialami. Saat ini gambar sudah dapat menjadi alat bantu mulai dari keperluan sehari-hari seperti pemetaan hutan, identifikasi forensik maupun sidik jari di kepolisian dan dunia kedokteran.

Hampir semua citra memerlukan media penyimpanan yang cukup besar. Hal ini dapat menimbulkan masalah yang cukup serius ketika citra disimpan dalam database dengan keterbatasan media penyimpanan yang ada. Masalah lain adalah ketika diinginkan untuk mengirimkan citra digital dengan menggunakan jalur komunikasi atau internet. Dengan ukuran file yang besar maka menimbulkan masalah pada pengiriman gambar yaitu waktu pengiriman yang lama. Sehingga diupayakan suatu solusi yang dapat mereduksi besarnya ukuran file gambar. Salah satu solusi untuk mengatasi masalah diatas adalah dengan melakukan kompresi.

Sesuai dengan latar belakang yang telah dipaparkan di atas, penulis akan membuat sebuah aplikasi sederhana berupa aplikasi Pengkompresi Citra Dengan Metode *Arithmetic Coding* dalam kawasan *Entropy Coding*.

a. Citra Digital

Citra digital merupakan hasil

Penangkapan suatu objek fisik menggunakan peralatan pencitraan digital, dimana setiap bagian dari gambar tersebut direpresentasikan dalam bentuk *pixel* (*picture elements*). Suatu citra digital dapat diedit, dimanipulasi, dikirim, dihapus, di-copy atau dimasukkan ke berkas komputer lainnya atau ke halaman web.

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan dalam pita magnetik. Komputer hanya dapat bekerja dengan bilangan numerik yang berhingga, sehingga gambar harus diubah ke dalam bentuk bilangan numerik berhingga (gambar digital) sebelum diproses dalam suatu komputer. Untuk mengubah gambar yang bersifat kontinu menjadi gambar digital diperlukan proses pembuatan kisi-kisi arah horizontal dan vertikal, sehingga diperoleh gambar dalam bentuk array dua dimensi. Proses tersebut dikenal sebagai proses digitalisasi atau *sampling*.

Gambar dapat direpresentasikan dengan array dua dimensi, dimana setiap elemen array dikenal sebagai elemen gambar (*pixel*). Pembagian suatu gambar menjadi sejumlah *pixel* dengan ukuran tertentu ini akan menentukan resolusi *spatial* (derajat kehalusan) yang diperoleh. Semakin kecil ukuran *pixel*-nya, maka akan semakin halus gambar yang diperoleh, karena informasi yang hilang akibat pengelompokan tingkat keabuan pada proses pembuatan kisi-kisi akan semakin kecil. Angka pada gambar digital menggambarkan bagaimana setiap *pixel* menggambarkan kecerahan (*brightness*) gambar tersebut pada titik yang bersangkutan. Proses lain yang dilakukan dalam suatu gambar

digital adalah proses kuantisasi. Dalam proses ini tingkat keabuan setiap *pixel* dinyatakan sebagai suatu harga *integer*. Batas-batas harga *integer* atau besarnya daerah tingkat keabuan yang digunakan untuk menyatakan tingkat keabuan *pixel* akan menentukan resolusi kecerahan dari gambar yang diperoleh. Kalau digunakan 3 bit untuk menyimpan harga *integer* tersebut, maka akan diperoleh sebanyak 8 tingkat keabuan.

Seluruh tahapan konversi di atas dikenal sebagai konversi analog ke digital, yang biasanya akan menyimpan hasil suatu proses pada memori gambar. Sebaliknya, sebagai hasil suatu proses pengolahan gambar digital, kadang-kadang perlu mengeluarkan gambar dari memori gambar ke bentuk peragaan pada monitor televisi atau ke bentuk cetakan foto. Proses konversi balikan ini dikenal sebagai konversi digital ke analog.

b. Kompresi Data

Kompresi data dalam ilmu komputer, ilmu pengetahuan dan seni adalah sebuah penyajian informasi ke dalam bentuk yang lebih sederhana (Pu, 2006). Kompresi data atau *source coding* dalam ilmu komputer dan teori informasi adalah proses meng-*encode* informasi dengan menggunakan lebih sedikit bit dari suatu sumber yang belum di-*encode* melalui penggunaan skema pengkodean yang spesifik (Data Compression, 2009). Kompresi data dapat diartikan juga sebagai proses yang dapat mengubah suatu aliran data masukan (sumber atau data asli) ke dalam aliran data yang lain (keluaran atau data yang dimampatkan) yang memiliki ukuran yang lebih kecil (Salomon, 2007)

Proses kompresi data didasarkan pada kenyataan bahwa pada hampir semua jenis data selalu terdapat pengulangan pada komponen data yang

dimilikinya, misalnya didalam suatu teks kalimat akan terdapat pengulangan penggunaan huruf alphabet dari huruf a sampai dengan huruf z. Kompresi data melalui proses encoding berusaha untuk menghilangkan unsur pengulangan ini dengan mengubahnya sedemikian rupa sehingga ukuran data menjadi lebih kecil.

Kompresi data dapat dibagi ke dalam dua teknik yaitu *lossless compression* dan *lossy compression*.

1. Lossless compression

Pada teknik ini tidak ada kehilangan informasi. Jika data dimampatkan secara *lossless*, data asli dapat direkonstruksi kembali sama persis dari data yang telah dimampatkan, dengan kata lain data asli tetap sama sebelum dan sesudah pemampatan. secara umum teknik *lossless* digunakan untuk penerapan yang tidak bisa mentoleransi setiap perbedaan antara data asli dan data yang telah direkonstruksi. *Lossless compression* disebut juga dengan *reversible compression* karena data asli bisa dikembalikan dengan sempurna. Akan tetapi rasio kompresinya sangat rendah, misalnya pada data teks, gambar seperti GIF dan PNG. Contoh metode ini adalah *Shannon-Fano Coding*, *Huffman Coding*, *Arithmetic Coding* dan lain sebagainya.

2. Lossy compression

Pada teknik ini akan terjadi kehilangan sebagian informasi. Data yang telah dimampatkan dengan teknik ini secara umum tidak bisa direkonstruksi sama persis dari data aslinya. Di dalam banyak penerapan, rekonstruksi yang tepat bukan suatu masalah. Sebagai contoh, ketika sebuah sample suara ditransmisikan, nilai eksak dari setiap sample suara belum tentu diperlukan. Tergantung pada yang memerlukan kualitas suara yang

direkonstruksi, sehingga banyaknya jumlah informasi yang hilang di sekitar nilai dari setiap sample dapat ditoleransi.

c. Algoritma Arithmetic Coding

Pada umumnya algoritma kompresi data melakukan penggantian satu atau lebih simbol input dengan kode tertentu. Berbeda dengan cara tersebut, *arithmetic coding* menggantikan satu deretan simbol input dengan sebuah bilangan *floating point*. Semakin panjang dan semakin kompleks pesan yang dikodekan, semakin banyak bit yang diperlukan untuk keperluan tersebut. Output dari *arithmetic coding* ini adalah satu angka yang lebih kecil dari 1 dan lebih besar atau sama dengan 0. Angka ini secara unik dapat di-*decode* sehingga menghasilkan deretan simbol yang dipakai untuk menghasilkan angka tersebut. Untuk menghasilkan angka output tersebut, tiap simbol yang akan di-*encode* diberi satu set nilai probabilitas.

Gagasan utama pada *arithmetic coding* adalah memberikan setiap simbol sebuah *range* atau interval. Dimulai dengan range $[0..1)$, setiap *range* dibagi dalam beberapa *subrange*, dimana ukurannya sebanding dengan probabilitas (kemungkinan yang muncul dari simbol-simbol yang sama). Semakin tinggi probabilitas yang dimiliki oleh suatu simbol, semakin tinggi pula *range* yang diberikan terhadap simbol tersebut. Tanda “[..)” berarti bahwa angka yang didalamnya diikutsertakan kecuali angka terakhir.

Setelah menentukan *range* dan probabilitasnya, mulai dapat melakukan proses *encoding* simbol-simbol, setiap simbol menentukan dimana output angka *floating point* itu berada. Sebagai contoh, misalnya terdapat citra digital dalam bentuk matriks seperti pada Gambar 1. berikut ini.

255	50	50	255	25	25
25	50	25	120	180	50
25	75	25	50	75	120
120	180	180	255	255	25
255	255	120	25	255	255
255	255	50	25	255	255

Gambar 1. potongan pixel citra digital

Kemudian dapat diperoleh probabilitas untuk setiap simbol. Setelah probabilitas tiap simbol telah diketahui, tiap simbol akan diberikan *range* tertentu yang nilainya diantara 0 dan 1, sesuai dengan probabilitas yang ada. Hal ini dapat dilihat pada Tabel 1. di bawah ini.

Selanjutnya dilakukan proses *encoding* menggunakan algoritma *encoding*. Karena gambar 1. mempunyai 36 simbol, maka kita misalkan 4 simbol saja seperti pada Gambar 2. di bawah ini.

Tabel 1. probabilitas untuk Gambar 1

Simbol	Frekuensi	Probabilitas	Range
255	12	1/3	0.00 – 0.33
50	6	1/6	0.33 – 0.5
25	9	1/4	0.5 – 0.75
120	4	1/9	0.75 – 0.86
180	3	1/12	0.86 – 0.94
75	2	1/18	0.94 – 1.00
Total	36	1	0.00 – 1.00

255	50	50	255	25	25
25	50	25	120	180	50
25	75	25	50	75	120
120	180	180	255	255	25
255	255	120	25	255	255
255	255	50	25	255	255

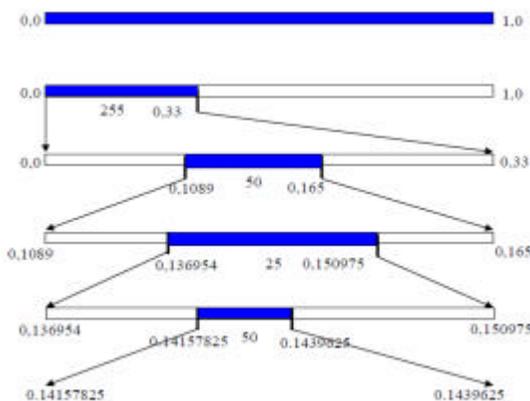
Gambar 2. potongan symbol citra digital

Pertama diambil simbol '255'.
 Nilai *Code Range* (kondisi awal) adalah $1.0 - 0.0 = 1$.
 $high_range(S) = 0.33$,
 $low_range(S) = 0.00$.
 Kemudian diperoleh nilai :
 $High = low + CR * high_range$
 $= 0.00 + 1.0 * 0.33 = 0.33$
 $Low = low + CR * low_range$
 $= 0.00 + 1.0 * 0.00 = 0.00$
 Kemudian diambil simbol '50'. Nilai *CodeRange* adalah $0.33 - 0.00 = 0.33$.
 $high_range(A) = 0.5$,
 $low_range(A) = 0.33$.
 Kemudian diperoleh nilai :
 $High = low + CR * high_range$
 $= 0.00 + 0.33 * 0.5 = 0.165$
 $Low = low + CR * low_range$
 $= 0.00 + 0.33 * 0.33 = 0.1089$
 Dan seterusnya yang dapat diringkas pada Tabel 2 di bawah.

Tabel 2. proses encoding untuk Gambar 2

Karakter	Low	High	CodeRange
	0.00	1.00	
255	0.00	0.33	1.00
50	0.1089	0.165	0.33
25	0.13695	0.150975	0.0561
50	0.14157825	0.1439625	0.014025

Perulangan untuk setiap interval dapat kita lihat pada Gambar 3. berikut



Gambar 3. perulangan setiap interval untuk gambar 2.

Dari proses tersebut didapat nilai $low = 0.14157825$. Nilai inilah yang direpresentasikan untuk membawa citra pada gambar 2.

Selanjutnya apabila gambar tersebut akan di-*decode* maka digunakan algoritma *decoding* yang telah disebutkan diatas. Misalkan untuk gambar 2 akan dilakukan *decoding*. Pertama sekali diperoleh *encoded number* 0.14157825 dengan simbol awal '255'. Maka proses *decoding* untuk simbol '255' sebagai berikut.

$$Low = 0.00$$

$$High = 0.33$$

$$Code\ Range = 0.33 - 0.00 = 0.33$$

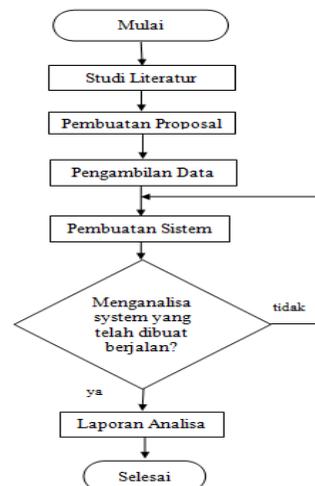
$$Encoded_number = 0.14157825 - 0.00 = 0.14157825$$

$$Encoded_number = 0.14157825 / 0.33 = 0.429025$$

Proses *decoding* dilakukan seterusnya sampai tidak ada lagi simbol. Proses *decoding* diringkas pada Tabel 3 berikut.

Tabel 3. proses encoding pada Gambar 2

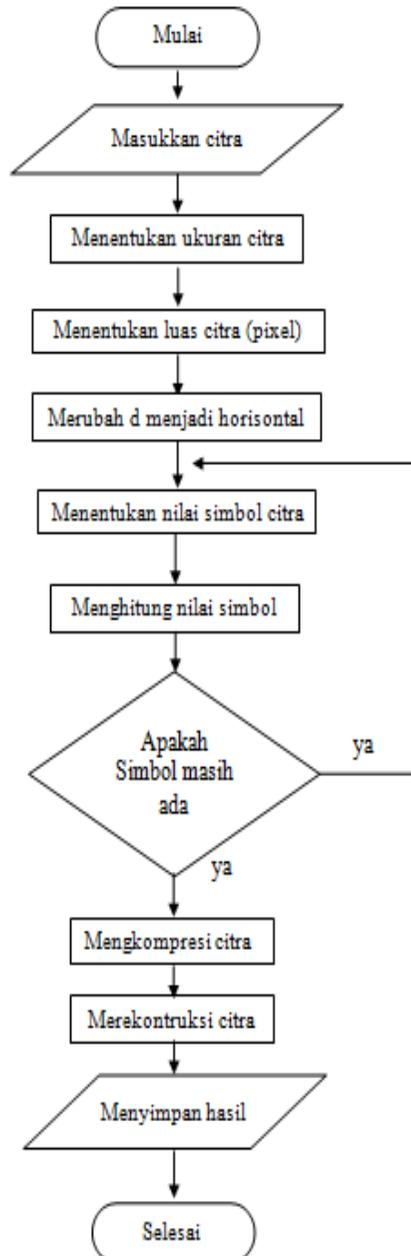
encoded number	simbol	low	high	code range
0.429025	255	0.00	0.33	0.33
0.5825	50	0.33	0.5	0.17
0.33	25	0.5	0.75	0.25
0	50	0.33	0.5	0.17



Gambar 4. flowchart penelitian

Alur penelitian yang digunakan dimulai dari pembuatan studi literatur, pembuatan proposal, pengambilan data, pembuatan sistem, menganalisa sistem yang telah dibuat serta pembuatan laporan. Keseluruhan alur penelitian dapat dilihat dalam *flowchart* Gambar 4. Di atas.

2. METODE PENELITIAN

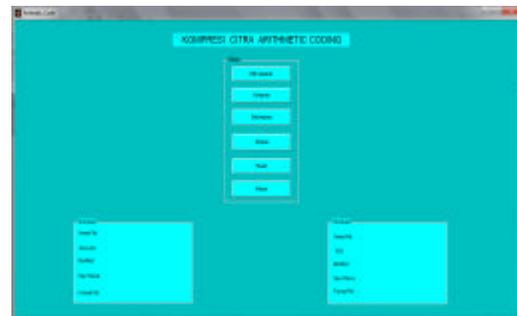


Gambar 5. *flowchart arithmetic coding*

Ada beberapa proses yang dilakukan dalam sistem yang dibuat ini. Langkah pertama adalah memasukkan citra, menentukan ukuran citra, menentukan luas citra, merubah d menjadi horizontal, menentukan nilai simbol citra, menghitung nilai simbol, mengkompresi citra, merekonstruksi citra dan yang terakhir adalah menyimpan. Inti dari sistem ini dapat berfungsi untuk mengkompresi citra digital dan mengembalikan kebentuk semula. Keseluruhan proses *arithmetic coding* dapat dilihat pada *flowchart* Gambar 5. di atas.

3. PENGUJIAN PROGRAM DAN ANALISA HASIL

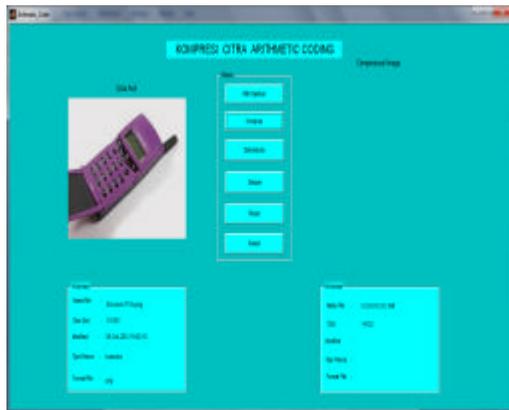
Gambar di bawah merupakan *screenshot* pengujian program dengan *Arithmetic Coding* yang melalui 2 tahap yaitu proses kompresi dan dekompresi yang ditunjukkan pada Gambar 6., 7. dan 8.



Gambar 6. Tampilan program kompresi



Gambar 7. Tampilan setelah memilih citra



Gambar 7. Tampilan hasil kompresi



Gambar 8. Tampilan hasil dekompresi

Analisa hasil merupakan analisa dari perbandingan ukuran citra asli dengan ukuran citra hasil dekompresi. Berikut beberapa tahap dalam menganalisa hasil.

a. Data Citra Uji

Program yang dibuat merupakan kompresi citra yang menggunakan metode *Arithmetic Coding*, proses yang dilakukan pada program kompresi citra ini terdapat dua, yaitu proses kompresi dan dekompresi.

Citra uji yang dipakai dalam Tugas Akhir ini menggunakan 8 buah citra dengan ukuran yang berbeda-beda yaitu bawahlaut, elektro, ericsson, jokowi, katak, linux, MU dan PPA. Kedelapan citra yang digunakan pada tugas akhir ini dapat dilihat pada Tabel 3. dibawah ini.

Tabel 3. data citra uji

No	Nama Citra	Ukuran citra Asli (Byte)	Ukuran Citra Digital (pixel)
1.	PPA	45000	150 x 100
2.	bawahlaut	50400	150 x 112
3.	ericsson	51189	151 x 113
4.	elektro	57150	127 x 150
5.	jokowi	63873	151 x 141
6.	linux	67500	151 x 151
7.	MU	67500	150 x 150
8.	katak	68403	151 x 151

b. Data Hasil Pengujian Citra

Tabel 4. Data hasil pengujian citra

No	Nama File	Ukuran citra Asli (Byte)	Ukuran Data Hasil Kompresi (Byte)	Rasio Kompresi	Kualitas citra kembali seperti semula
1.	bawahlaut	50400	26695	1,887	ya
2.	PPA	45000	20765	2,167	ya
3.	jokowi	63873	29294	2,180	ya
4.	linux	67500	19605	3,442	ya
5.	katak	68403	19233	3,556	ya
6.	ericsson	51189	14133	3,621	ya
7.	elektro	57150	14865	3,844	ya
8.	MU	67500	11809	5,715	ya

Dari tabel di atas, dapat disimpulkan bahwa tingkat rasio kompresi sebuah gambar tidak dipengaruhi oleh besarnya ukuran file, melainkan tergantung pada komponen warna gambar yang bersangkutan. Dan setelah proses kompresi dilakukan, didapatkan beberapa hasil perbedaan terhadap besaran rasio kompresi masing-masing citra. Rasio kompresi untuk citra digital berkisar antara 1,887-5,715. Gambar dengan nama bawahlaut yang mempunyai ukuran 50400 *Byte* rasionya paling kecil, sedangkan gambar dengan nama MU yang ukuran filenya 60700 *Byte* mempunyai rasionya yang paling besar.

Apabila ukuran *file* citra uji yang digunakan terlalu besar atau di atas 151 x 151 *pixel*, maka waktu yang

dibutuhkan untuk proses kompresi maupun dekompresi membutuhkan waktu yang lama. Akan tetapi hasil kualitas citra yang sudah dikompresi dapat dikembalikan lagi tepat seperti aslinya.

4. KESIMPULAN

- a. Algoritma *Arithmetic Coding* apabila diimplementasikan pada citra hasilnya cukup baik.
- b. Tingkat rasio kompresi sebuah gambar tidak dipengaruhi oleh besarnya ukuran file, melainkan tergantung komponen warna gambar yang bersangkutan.
- c. Hasil rasio kompresi yang didapatkan dari masing-masing gambar berbeda-beda hasilnya. Gambar dengan nama bawahlaut yang mempunyai ukuran 50400 *Byte* rasionya paling kecil, yaitu 1,887. Sedangkan gambar dengan nama MU yang ukuran filenya 60700 *Byte* mempunyai rasio yang paling besar, yaitu 5,715.
- d. Ukuran *file* citra yang dihasilkan setelah proses dekompresi tepat seperti semula dan kualitasnya pun sama tepat seperti data aslinya.

5. DAFTAR PUSTAKA

- Cleve Moler. 2004. "the creator of MATLAB The Origins of MATLAB".
http://www.mathworks.com/company/newsletters/news_notes/clevescornr/dec04.html
- Edy Sujatmiko. 2007. "Pemilihan Algoritma Optimal untuk Kompresi Data Citra Iris Mata Manusia".
Semarang: Universitas Diponegoro
- Krisnawati. 2007. "Kompresi Citra RGB Dengan Metode Kuantisasi"
Yogyakarta: Sekolah Tinggi

Manajemen Informatika dan Komputer AMIKOM

Paul E. Black 2012.
http://en.wikipedia.org/wiki/Arithmetic_coding

Petrus, Santoso. 2001. "Studi kompresi data dengan metode Aritmathic Coding".
Semarang: Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra

Sriwiyanto, Agus. 2012. "Implementasi Metode *Run Length Encoding* Dalam Kompresi Citra Dengan Citra Hitam Putih".
Surakarta : Universitas Muhammadiyah Surakarta

Widakdo, Ari. 2012. "Implementasi Algoritma Metode *Huffman* Pada Kompresi Citra".
Surakarta : Universitas Muhammadiyah Surakarta

Widiarsono, Teguh . 2005 . *Tutorial Praktis Belajar Matlab*.
Jakarta