T-21

PERBANDINGAN RASIO KOMPRESI PADA KOMPRESI CITRA DIGITAL BITMAP MENGGUNAKAN KOMBINASI METODE DISCRETE COSINE TRANSFORM DAN ARITHMETIC CODING DENGAN BERBAGAI DIMENSI CITRA SUMBER

Dimas Aryo Prakoso[1], Kuswari Hernawati[2]

[1,2]Jurusan Pendidikan Matematika FMIPA UNY
[1]mr.dimasaryo@gmail.com, [2]kuswari@uny.ac.id

Abstrak

Pada umumnya citra digital bitmap memiliki ukuran yang relatif besar. Hal ini mempengaruhi kecepatan transmisi data dan membutuhkan tempat penyimpanan memori yang besar sehingga diperlukan kompresi untuk mereduksi ukuran datanya. Beberapa metode kompresi data yang dapat digunakan adalah Discrete Cosine Transform (DCT) dan arithmetic coding. Dalam tulisan ini akan dibandingkan rasio kompresi yang mampu dihasilkan oleh proses kompresi-dekompresi citra digital bitmap menggunakan kombinasi metode DCT dan arithmetic coding dengan berbagai dimensi citra sumber.

Berdasarkan hasil penelitian yang dilakukan disimpulkan bahwa rasio kompresi dan kualitas citra rekonstruksi semakin meningkat seiring dengan meningkatnya dimensi citra sumber. Citra sumber berdimensi 64 x 64 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 6, PSNR rata-rata 27 dB, dan MSE rata-rata 122,32. Citra sumber berdimensi 128 x 128 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 7, PNSR rata-rata 29 dB, dan MSE rata-rata 86,12. Citra sumber berdimensi 256 x 256 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 11, PNSR rata-rata 31 dB, dan MSE rata-rata 54,19. Citra sumber berdimensi 512 x 512 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 14, PNSR rata-rata 33, dan MSE rata-rata 33,28. Citra sumber berdimensi 1024 x 1024 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 19, PSNR rata-rata 37 dB, dan MSE rata-rata 13,92.

Kata kunci: kompresi, citra digital bitmap, discrete cosine transform, arithmetic coding

PENDAHULUAN

1. PENDAHULUAN

Pada umumnya citra digital ini memiliki ukuran yang relatif besar. Misalnya citra digital *bitmap* 24-*bit* berdimensi 1024 x 768 piksel membutuhkan memori penyimpanan sebesar 18.874.368 *bits* atau sekitar 2 Mb. Semakin besar ukuran suatu data maka kecepatan transmisi datanya juga semakin lambat dan membutuhkan tempat penyimpanan (*memori*) yang semakin besar. Oleh karena itu dikembangkan berbagai teknik untuk mereduksi ukuran data citra digital tersebut dengan cara mengurangi tingkat redundansi datanya. Teknik seperti ini dinamakan teknik kompresi (*encoding*) citra

digital. Sedangkan teknik untuk merekonstruksi kembali citra digital tersebut dinamakan dekompresi (*decoding*) citra digital. Sistem kompresi-dekompresi data dinamakan *codec* (*encoder-decoder*).

Terdapat dua jenis kompresi citra digital, yaitu kompresi *lossless* dan kompresi *lossy*. Jika citra digital hasil rekonstruksi identik dengan citra digital sumber, maka disebut sebagai kompresi *lossless*. Jika citra digital hasil rekonstruksi tidak identik dengan citra digital sumber karena ada informasi yang hilang pada saat proses kompresi-dekompresi, maka disebut sebagai kompresi *lossy*. Rasio kompresi yang dapat dicapai oleh kompresi *lossy* lebih baik daripada kompresi *lossless* (Pu, 2005: 204). Ukuran yang digunakan untuk mengetahui kualitas citra rekonstruksi pada kompresi *lossy* antara lain *Mean Square Error* (MSE) dan *Peak Signal to Noise Ratio* (PSNR).

Salah satu metode kompresi *lossy* yang sangat populer dalam kompresi citra digital adalah metode *discrete cosine transform* (DCT). Transformasi ini memanfaatkan korelasi antar elemen dalam himpunan data masukan. Jika data masukan terdiri dari data-data yang berkorelasi secara kuantitas, maka kebanyakan koefisien DCT yang dihasilkan adalah 0 atau bilangan yang mendekati 0. Koefisien hasil transformasi tersebut kemudian dikuantisasi dengan bilangan tertentu sehingga menghasilkan banyak redundansi data. Metode ini tidak mereduksi ukuran data tetapi menghasilkan data dengan tingkat redundansi yang tinggi. Oleh karena itu metode ini perlu dikombinasikan dengan metode lain untuk mereduksi ukuran datanya, antara lain *Run Length Encoding*, *Huffman*, atau *Arithmetic Coding*.

Metode *arithmetic coding* merupakan salah satu metode kompresi *lossless* yang memakai teknik *statistical modeling* dengan cara mengkodekan suatu barisan karakter/pesan menjadi suatu bilangan tunggal. Dasar dari metode *arithmetic coding* adalah kenyataan bahwa peluang kumulatif suatu rangkaian simbol merupakan *subinterval* unik di dalam interval [0; 1). Jika data masukan tidak mengalami perubahan pada saat proses kompresi, maka metode tersebut dinamakan *static arithmetic coding.* Jika data masukan mengalami perubahan saat proses kompresi, maka metode tersebut dinamakan *dinamic arithmetic coding*.

Menurut Bodden (2007: 42), metode *arithmetic coding* sangat cocok untuk data yang memiliki nilai ragam kecil. Hal ini sangat sesuai dengan karakteristik koefisien DCT yang memiliki ragam kecil dan tingkat redundansi data yang tinggi karena kebanyakan koefisiennya akan dikuantisasi menjadi bilangan 0. Oleh karena itu diperlukan penelitian tentang proses kompresi citra digital *bitmap* menggunakan kombinasi metode DCT dan *arithmetic coding* untuk mengetahui rasio kompresi dan kualitas citra rekonstruksi yang mampu dihasilkan oleh proses tersebut.

2. CITRA DIGITAL

Citra didefinisikan sebagai fungsi kontinyu dua dimensi *f(x,y)*, *x* dan *y* merupakan koordinat spasial, dan setiap nilai *f(x,y)* merupakan intensitas atau derajat keabuan (*gray level*) citra pada koordinat *(x,y)*. Jika f(x,y) diskrit, maka dinamakan citra digital. Citra digital tersusun atas sejumlah elemen yang disebut piksel/ *pixel* (*Picture Element*). Satu piksel berarti satu titik pada citra. Nilai setiap piksel merupakan kuantisasi nilai intensitas cahaya atau derajat keabuan pada setiap titik. Dengan demikian, suatu citra digital dapat dipandang sebagai sebuah matriks 2 dimensi yang elemen-elemennya menunjukkan intensitas cahaya terkuantisasi (Miano, 1999: 1). Sebuah citra digital berdimensi M x N dapat direpresentasikan dalam matriks seperti berikut (Gonzales. 2001: 55):

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix}$$

Ada 2 jenis citra digital yang sering digunakan, yaitu citra digital *bitmap* dan citra digital *vector*. Citra digital *bitmap* disimpan sebagai *array* yang berisi nilai piksel, sedangkan citra digital *vector* disimpan sebagai deskripsi matematis komponen penyusunnya seperti titik, garis, kurva, dan bidang. Salah satu citra berformat *bitmap* adalah *Windows* BMP.

3. MODEL WARNA

Model warna (*color model)* merupakan sistem untuk merepresentasikan warna secara numerik.Ada dua jenis model warna yang sering digunakan, yaitu:

a. Model warna RGB

Model warna RGB berasal dari ide tentang pembentukan warna dari tiga warna dasar atau yang disebut sebagai *additive primary colours of light*. Ketiga warna dasar tersebut adalah *red* (R), *green* (G), dan *blue* (B). Setiap warna merupakan perpaduan dari ketiga warna dasar tersebut.

b. Model warna LC

Model warna ini juga sering disebut sebagai model warna YCbCr. *Luminance* (Y) merupakan komponen yang merepresentasikan intensitas keabuan dari citra digital. *Chrominance Blue* (Cb) merepresentasikan intensitas kebiruan pada citra digital, sedangkan *Chrominance Red* (Cr) merepresentasikan intensitas kemerahan pada citra digital. Hubungan antara model warna RGB dan model warna LC ditunjukkan dalam persamaan berikut ini (Miano, 1999: 6):

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = 0.167R - 0.3313G + 128$$
$$Cr = 0.5R - 0.4187G - 0.0813B + 128$$
$$R = Y + 1.402Cr$$
$$G = Y - 0.4414(Cb - 128) - 0.71414(Cr - 128)$$
$$B = Y + 1.722(Cb - 128)$$

Model warna LC ini lebih unggul di bidang kompresi citra digital karena komponen *luminance* dapat dikompresi dengan tingkat keakuratan yang lebih tinggi (rasio kompresi yang lebih rendah), sedangkan komponen *chrominance* dikompresi dengan tingkat keakuratan yang lebih rendah (rasio kompresi yang lebih tinggi). Hal ini memanfaatkan fakta bahwa mata manusia lebih sensitif terhadap perubahan tingkat kecerahan (*brightness*) daripada perubahan warna (Pu, 2005: 196).

4. METODE DISCRETE COSINE TRANSFORM

Dalam kompresi citra digital, DCT yang digunakan adalah DCT 2 dimensi karena citra digital merupakan data dua dimensi. Jika data sumber adalah himpunan m x n data p_{xy} (piksel, *audio samples*, dan sebagainya), maka koefisien DCT satu dimensi ke-*ij* ,

dengan *i* = 0, 1, ..., m-1 dan j = 0, 1, ..., n-1, dapat dihitung dengan persamaan (Salomon, 2004: 293):

$$G_{ij} = \frac{2}{\sqrt{mn}} C_i C_j \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} P_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2m}\right)$$

Koefisien pertama G_{00} disebut sebagai koefisien DC, sedangkan sisanya adalah koefisien AC. Sedangkan dekompresi citra digital dilakukan dengan mentransformasikan koefisien DCT yang sudah terkuantisasi menggunakan persamaan IDCT berikut ini (Salomon, 2004: 293):

$$P_{xy} = \frac{2}{\sqrt{mn}} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{2m}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right)$$

Tahapan kompresi citra digital dengan metode DCT adalah sebagai berikut (Salomon, 2004: 293):

a. Citra digital dibagi menjadi sejumlah k blok berdimensi 8 x 8 piksel. Piksel dinotasikan dengan P_{xy}. Jika lebar atau tinggi citra digital tidak habis dibagi 8, maka baris paling bawah atau kolom paling kanan ditambahkan dengan data bernilai 0 sehingga lebar atau tinggi citra habis dibagi 8.
b. DCT 2 dimensi diterapkan untuk setiap blok B_i. Hasilnya kita sebut sebagai vektor $W^{(i)}$ dari 64 koefisien transformasi $w_j^{(i)}$ dimana j = 0, 1, . . . , 63. k vektor $W^{(i)}$ menjadi baris dalam matriks W.

$$W = \begin{bmatrix} w_0^{(1)} & w_1^{(1)} & \dots & w_{63}^{(1)} \\ w_0^{(2)} & w_1^{(2)} & \dots & w_{63}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ w_0^{(k)} & w_1^{(k)} & \dots & w_{63}^{(k)} \end{bmatrix}$$

c. Enam puluh empat kolom dari W dinotasikan dengan $C^{(0)}, C^{(1)}, \dots, C^{(63)}$. k elemen dari $C^{(j)}$ adalah $w_j^{(1)}, w_j^{(2)}, \dots, w_j^{(k)}$. Koefisien pertama vektor $C^{(0)}$ merupakan koefisien DC.
d. Setiap vektor $C^{(j)}$ dikuantisasi secara terpisah untuk menghasilkan vektor $Q^{(j)}$. Selanjutnya elemen $Q^{(j)}$ dapat ditulis ke dalam aliran data terkompresi.

Decoder membaca 64 koefisien terkuantisasi vektor $Q^{(j)}$ dari setiap k elemen, dan kemudian menyimpan hasilnya sebagai kolom sebuah matriks. Setiap elemen tersebut didekuantisasi. Kemudian setiap elemen ditransformasikan menggunakan IDCT untuk merekonstruksi data B_i.

5. METODE ARITHMETIC CODING

Metode *arithmetic coding* dikembangkan pertama kali oleh Abramson dan Peter Elias di awal tahun 1960 . Metode ini dikembangkan dari hasil observasi Shannon dan Fano pada tahun 1948 tentang pengkodean N simbol menggunakan peluang kumulatifnya (Pu, 2005: 101). Dasar dari metode *arithmetic coding* adalah kenyataan bahwa peluang kumulatif suatu rangkaian simbol sama dengan subinterval unik di dalam

interval [0; 1). Metode ini memproses keseluruhan rangkaian simbol menjadi sebuah bilangan pecahan (*floating-point*) kurang dari 1 atau lebih dari sama dengan 0.

a. Metode *arithmetic coding* menggunakan bilangan pecahan (*floating point*)

Langkah-langkah utama dalam proses kompresi menggunakan metode *arithmetic coding* adalah sebagai berikut (Salomon, 2004: 109):

1) Menyatakan interval awal [0; 1).
2) Mengulangi 2 langkah berikut ini untuk setiap simbol X dalam rangkaian data masukan:
   a) Bagi interval menjadi beberapa subinterval dengan ukuran yang sesuai dengan peluang simbol X tersebut.
   b) Pilih subinterval untuk X dan definisikan sebagai interval baru. Hal ini dilakukan dengan cara memperbarui nilai [*low*; *high*). Jika *Rng* adalah panjang interval lama, *Hrg*(X) adalah batas atas simbol X, *Lrg*(X) adalah batas bawah simbol X, maka nilai [*newlow*; *newhigh*) dapat dihitung dengan persamaan:

$$newhigh = low + Rng \times Hrg(X)$$
$$newlow = low + Rng \times Lrg(X)$$

3) Hasil akhirnya adalah sebuah bilangan tunggal yang berada di dalam interval akhir.

Proses dekompresi dimulai dengan mencari simbol dengan interval yang memuat kode hasil kompresi. Selanjutnya nilai kode (*Code*) tersebut diperbarui dengan persamaan (Salomon, 2004: 11):

$$Code = \frac{(Code - Lrg(X))}{Rng}$$

b. Metode *arithmetic coding* menggunakan bilangan bulat (*integer*)

Kelemahan metode *arithmetic coding* menggunakan bilangan pecahan (*floating-point)* adalah lambat dan dapat kehilangan ketepatannya (*loss precision*). Oleh karena itu implementasi metode *arithmetic coding* lebih baik menggunakan bilangan bulat (*integer*) 16 bit atau 32 bit (Nelson. 2000: 76).

Jika menggunakan bilangan bulat, Kita tidak dapat menyatakan peluang sebuah simbol yang berupa pecahan dari bilangan. Oleh karena itu batas bawah dan batas atas interval pada setiap simbol perlu dinormalisasi menggunakan persamaan (Bodden, 2007: 23):

$$LowCount = \sum_{i=0}^{symbol\ -1} CumCount(i)$$

$$HighCount = LowCount + CumCount(Symbol)$$

CumCount merupakan jumlah kumulatif frekuensi simbol. Jika menggunakan *integer* 32 *bits*, maka batas atas awal yang dapat digunakan adalah 0x7FFFFFFF. Bilangan tersebut merupakan nilai maksimum dari *integer* 31 bits. Kita menyisakan 1 *bits* untuk menghindari *overflows*. Batas bawah awal adalah 0. Selanjutnya batas bawah dan batas atas diperbarui dengan persamaan (Bodden, 2007: 24):

$High = Low + Step * HighCount - 1$
$Low = Low + Step * LowCount$
dengan

$$Step = \frac{High - Low + 1}{total}$$

Jika menggunakan integer 32 bits, maka proses dekompresi dimulai dengan menyatakan interval awal [0, 0x7FFFFFFF). Setelah itu menghitung nilai dari kode (*code*) hasil kompresi dengan persamaan (Bodden, 2007: 26):

$$Value = \frac{Code - Low}{Step}$$

dengan

$$Step = \frac{High - Low + 1}{total}$$

Selanjutnya adalah mencari simbol dengan interval yang memuat *value*.

Jika data yang diproses semakin banyak maka batas bawah (*low*) dan batas atas (*high*) akan saling mendekati terus menerus hingga keduanya bernilai sama. Untuk menghindari hal ini maka dikembangkan aturan penskalaan yang dinamakan E1, E2, E3 *scaling* (Bodden, 2007: 28).

1) E1 dan E2 *scaling*

Jika batas bawah dan batas atas sama-sama kurang dari atau lebih dari setengah rentang suatu bilangan, maka *most significant bits* (MSB) dari kedua variabel tersebut tidak akan berubah. Oleh karena itu MSB tersebut dapat disimpan pada file hasil kompresi. Selanjutnya perlu penskalaan terhadap kedua variabel tersebut dengan aturan berikut ini:

a) Jika batas atas kurang dari setengah rentang suatu bilangan maka dinamakan E1 *scaling*. Bit 0 akan disimpan pada aliran data keluar. Selanjutnya nilai *low* dan *high* diperbarui dengan persamaan (Bodden, 2007: 28):

   $low = low \times 2$

   $high = high \times 2 - 1$

b) Jika batas bawah lebih dari setengah rentang suatu bilangan, maka dinamakan E2 *scaling*. Bit 1 akan disimpan pada aliran data keluar. Selanjutnya nilai *low* dan *high* diperbarui dengan persamaan (Bodden, 2007: 28):

   $low = 2 \times (low - half)$

   $high = 2 \times (high - half) + 1$

2) E3 *scaling*

Jika batas bawah lebih besar atau sama dengan nilai maksimum kuarter pertama (*first quarter*) suatu interval dan batas atasnya kurang dari nilai maksimum kuarter ketiga (*third quarter*) pada suatu rentang, maka tidak ada MSB sampai proses kompresi selesai sehingga *encoder* tidak dapat menyimpan hasil kompresi. Untuk menyelesaikan

persoalan ini batas bawah dan batas atas perlu diperbarui dengan persamaan (Bodden, 2007: 29):

$$low = 2 \times (low - first\ quarter)$$

$$high = 2 \times (high - first\ quarter) + 1$$

E3 *scaling* diulangi hingga batas bawah dan batas atas memenuhi persyaratan pada E1 dan E2 *scaling*. Banyaknya perulangan disimpan dalam suatu *counter*. Jika nilai batas bawah dan batas atas sudah memenuhi syarat untuk E1 dan E2 *scaling*, *encoder* akan menyimpan *bits* pada aliran data keluar sebanyak nilai pada *counter*. Jika E1 scaling, maka *bits* yang disimpan adalah 1. Jika E2 scaling, maka *bits* yang disimpan adalah 0.

6. RASIO KOMPRESI

Pengukuran rasio kompresi dilakukan dengan membandingkan ukuran data hasil kompresi dan ukuran data citra sumber. Jika LD' adalah ukuran data hasil kompresi, dan LD adalah ukuran data citra sumber, maka rasio kompresi Rs dapat dihitung dengan persamaan (Pu, 2005: 11):

$$Rs = \frac{LD'}{LD}$$

7. KUALITAS CITRA REKONSTRUKSI

a. *Mean Square Error* (MSE)

MSE menunjukkan kuadrat rata-rata selisih nilai piksel citra rekonstruksi dengan citra sumber. Semakin kecil nilai MSE maka kualitas citra tersebut semakin baik. Misalkan P_i merupakan piksel dari citra digital rekonstruksi dan Q_i merupakan piksel dari citra digital sumber, dengan i = 1, 2, 3, ..., N, maka nilai MSE σ^2 dapat dihitung dengan persamaan (Pu, 2005: 204):

$$\sigma^2 = \frac{1}{N}\sum_{t=1}^{N}(P_i - Q_i)^2$$

b. *Peak Signal to Noise Ratio* (PSNR)

Semakin besar nilai PSNR, maka kualitas citra rekonstruksi juga semakin baik. Jika MSE = 0, maka PNSR = ∞. Satuan yang digunakan adalah *decibel* (dB). Nilai PSNR dapat dihitung dengan persamaan (Pu, 2005: 205):

$$PSNR = 20log_{10}\frac{Max_i|P_i|}{\sigma_d^2}$$

8. HASIL DAN PEMBAHASAN

a. Proses kompresi-dekompresi citra digital bitmap menggunakan kombinsasi metode *discrete cosine transform* dan *arithmetic coding*

1) Citra sumber

Citra digital yang akan digunakan sebagai citra sumber adalah citra digital *bitmap* berformat *windows* BMP 24-*bit*. Citra digital ini memuat *array* yang berisi data piksel

RGB. Setiap piksel berukuran 3 *bytes* (24 *bit*). Setiap piksel merupakan perpaduan dari komponen warna dasar *red*, *green*, dan *blue* yang masing-masing berukuran 1 *byte*. Jadi setiap komponen warna dasar memiliki rentang nilai 0-255 atau memiliki 256 tingkat kecerahan yang berbeda.

2) Proses kompresi

Tahapan proses kompresi citra digital *bitmap* menggunakan kombinasi metode DCT dan *arithmetic coding* adalah sebagai berikut:

a) Membaca data piksel RGB citra sumber
b) Data dipartisi ke dalam sejumlah blok data berukuran 8 x 8 piksel
c) Mengubah model warna RGB menjadi model warna LC pada setiap blok data
d) Transformasi data menggunakan DCT pada setiap blok data.
e) Mereduksi ukuran data koefisien AC menggunakan metode *arithmetic coding*.

3) Berkas hasil kompresi

Berkas hasil kompresi terdiri dari 3 bagian, yaitu *header*, data, dan EOF. Struktur ini merupakan struktur minimal yang berisi data-data yang diperlukan dalam proses dekompresi. *Header* berukuran 31 *bytes* dan EOF berukuran 1 *bytes*. Sedangkan ukuran data menyesuaikan hasil *arithmetic coding*. Struktur berkas hasil kompresi dapat dilihat pada Tabel 1.

Tabel 1. Struktur berkas hasil kompresi.

	Nama	**Ukuran**	**Keterangan**
Header	Tipe	3 *bytes*	Berisi nilai ASCII ‘DND’.
	Height	2 *bytes*	Tinggi citra sumber.
	Width	2 *bytes*	Lebar citra sumber.
	TYOffset	4 *bytes*	Letak tabel frekuensi *luminance* (Y).
	TCbOffset	4 *bytes*	Letak tabel frekuensi *chrominance blue* (Cb).
	TCrOffset	4 *bytes*	Letak tabel frekuensi *chrominance red* (Cr).
	CYOffset	4 *bytes*	Letak kode *luminance* (Y).
	CCbOffset	4 *bytes*	Letak kode *chrominance blue* (Cb).
	CCrOffset	4 *bytes*	Letak kode *chrominance red* (Cr).
Data	DCY	-	DC *luminance* (Y).
	DCCb	-	DC *chrominance blue* (Cb).
	DCCr	-	DC *chrominance red* (Cr).
	TY	-	Tabel frekuensi *luminance* (Y).
	TCb	-	Tabel frekuensi *chrominance blue* (Cb).
	TCr	-	Tabel frekuensi *chrominance red* (Cr).
	CY	-	Kode *luminance* (Y).

	CCb	-	Kode *chrominance blue* (Cb).
	CCr	-	Kode *chrominance red* (Cr).
EOF	EOF	1 *bytes*	Penanda akhir file (*End of File*). Berisi 0x44

4) Proses dekompresi

Tahapan dalam proses dekompresi menggunakan kombinasi metode DCT dan *arithmetic coding* adalah sebagai berikut:

a) Membaca berkas hasil kompresi
b) Merekonstruksi data koefisien AC menggunakan arithmetic coding
c) Transformasi data menggunakan IDCT pada setiap blok data
d) Mengubah model warna LC menjadi model warna RGB pada setiap blok data
e) Menyatukan semua blok data menjadi citra digital bitmap sehingga citra rekonstruksi dapat ditampilkan

b. Hasil pengujian

Pengujian dilakukan dengan cara membandingkan citra hasil rekonstruksi dengan citra sumber. Alat ukur yang digunakan antara lain rasio ukuran citra digital, *mean square error* (MSE), dan *peak signal to noise ratio* (PSNR). Citra yang digunakan sebagai citra sumber adalah 21 citra digital *bitmap* berformat *Windows* BMP 24-*bit* dengan dimensi 64 x 64, 128 x 128, 256 x 256, 512 x 512, dan 1024 x 1024. Tampilan visual citra sumber dapat dilihat pada Tabel 2.

Tabel 2. Tampilan visual citra sumber.

		
Airplane.bmp	Baboon.bmp	Barbara.bmp

 Boats.bmp	 Cablecar.bmp	 Cornfield.bmp
 Flower.bmp	 Flowers.bmp	 Fruits.bmp
 Girl.bmp	 Goldhill.bmp	 Lenna.bmp
 Monarch.bmp	 Moon.bmp	 Pens.bmp
 Pepper.bmp	 Sailboat.bmp	 Soccer.bmp

		
Tiffany.bmp	Yacht.bmp	Zeldacolor.bmp

Rasio kompresi rata-rata, MSE rata-rata, dan PSNR rata-rata yang dihasilkan oleh proses kompresi-dekompresi citra digital bitmap menggunakan kombinasi metode DCT dan *arithmetic coding* dapat dilihat pada Tabel 3.

Tabel 3. Hasil pengujian.

Dimensi	**Rasio**	**Kualitas**	
		PSNR (dB)	**MSE**
64 x 64 piksel	1 : 6	27	122,32
128 x 128 piksel	1 : 7	29	86,12
256 x 256 piksel	1 : 11	31	54,19
512 x 512 piksel	1 : 14	33	33,28
1024 x 1024 piksel	1 : 19	37	13,92

Rasio kompresi dan kualitas citra rekonstruksi semakin meningkat seiring dengan meningkatnya dimensi citra sumber. Misalnya citra sumber berdimensi 64 x 64 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 6, PSNR rata-rata sebesar 27 dB, dan MSE rata-rata sebesar 122,32, sedangkan citra sumber berdimensi 1024 x 1024 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 19, PSNR rata-rata sebesar 37 dB, dan MSE rata-rata sebesar 13,92.

9. KESIMPULAN

Rasio kompresi dan kualitas citra rekonstruksi yang dihasilkan oleh proses kompresi-dekompresi citra digital bitmap menggunakan kombinasi metode *discrete cosine transform* dan *arithmetic coding* terus meningkat seiring meningkatnya dimensi citra sumber. Citra sumber berdimensi 64 x 64 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 6, PSNR rata-rata 27 dB, dan MSE rata-rata 122,32. Citra sumber berdimensi 128 x 128 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 7, PNSR rata-rata 29 dB, dan MSE rata-rata 86,12. Citra sumber berdimensi 256 x 256 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 11, PNSR rata-rata 31 dB, dan MSE rata-rata 54,19. Citra sumber berdimensi 512 x 512 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 14, PNSR rata-rata 33, dan MSE rata-rata 33,28. Citra sumber berdimensi 1024 x 1024 piksel dapat direduksi dengan rasio kompresi rata-rata 1 : 19, PSNR rata-rata 37 dB, dan MSE rata-rata 13,92

10. DAFTAR PUSTAKA

Bodden E, Clasen M, Kneis J. 2007. *Arithmetic Coding Revealed*. Canada: Sable McGill.

Gonzalez RC, Woods RE. 2001. *Digital Image Processing*. New Jersey: Prentice-Hall, Inc.

Miano J. 1999. *Compressed Image File Format*. Massachusetts: Addison Wesley Longman, Inc.

Pu IM. 2005. *Fundamental Data Compression*. Oxford: Butterworth-Heinemann.

Salomon D. 2004. *Data Compression*. New York: Springer-Verlag, Inc.

Singgih Santoso. 2001. *Mengolah data Statistik secara profesional*. Jakarta: PT Elex Media Komputindo Kelompok Gramedia.

Schramm,W. 1984. *Media Besar Media Kecil*. Semarang: IKIP Semarang Press