Kompresi File Citra Digital Dengan Arithmetic Coding

Fricles Ariwisanto Sianturi
Program Studi S1 Teknik Informatika STMIK Pelita Nusantara Medan
Jl. Iskandar Muda No. 1 Medan,
e-mail : sianturifricles@gmail.com

Abstrak

Pada dasarnya ada tiga bidang yang menangani pengolahan data berbentuk citra, yaitu: grafika komputer, pengolahan citra, dan visi komputer. Pada bidang grafika komputer banyak dilakukan proses yang bersifat sintesis yang mempunyai ciri data masukan berbentuk deskriptif dengan keluaran hasil proses yang berbentuk citra. Sedangkan proses di dalam bidang visi komputer merupakan kebalikan dari proses grafika komputer. Terakhir, bidang pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual, yakni data masukan maupun data keluarannya berbentuk citra. Pada pengolahan citra terdapat enam jenis operasi pengolahan, yaitu peningkatan kualitas citra, restorasi citra, kompresi citra, segmentasi citra, analisis citra, dan rekonstruksi citra. Pada umumnya informasi yang ada dalam suatu citra terletak pada strukturnya. Agar mudah memahami suatu citra dapat dilakukan dengan menyederhanakan struktur citra tersebut. Pada pengolahan citra terdapat enam jenis operasi pengolahan, yaitu peningkatan kualitas citra, restorasi citra, kompresi citra, segmentasi citra, analisis citra, dan rekonstruksi citra. Pada umumnya informasi yang ada dalam suatu citra terletak pada strukturnya. Agar mudah memahami suatu citra dapat dilakukan dengan menyederhanakan struktur citra tersebut.

Kata kunci : *Grafika Komputer*, Pengolahan Citra, *Arithmetic Coding*

Abstract

Basically there are three fields that handle image-shaped data processing, namely: computer graphics, image processing, and computer vision. In the field of computer graphics many synthetic processes are done that have the characteristics of input data in the form of descriptive output of the process in the form of images. While the process in the field of computer vision is the opposite of the computer graphics process. Finally, the field of image processing is the processing and image analysis that involves a lot of visual perception, ie input data and output data in the form of images. In the image processing there are six types of processing operations, namely image quality improvement, image restoration, image compression, image segmentation, image analysis, and image reconstruction. In general, the information contained in an image lies in its structure. In order to easily understand an image can be done by simplifying the image structure. In the image processing there are six types of processing operations, namely image quality improvement, image restoration, image compression, image segmentation, image analysis, and image reconstruction. In general, the information contained in an image lies in its structure. In order to easily understand an image can be done by simplifying the image structure.

Keywords : Computer Graphics, Image Processing, Arithmetic Coding

1. PENDAHULUAN

Kompresi data merupakan suatu hal yang esensial. Teknik kompresi ini esensial karena ukuran dari data semakin lama semakin besar, tetapi belum optimal karena tidak didukung oleh perkembangan dari teknologi penyimpanan data dan *bandwidth* (untuk kecepatan *download* data

dari internet) yang seimbang. Sementara orang orang pun menginginkan data dengan kualitas terbaik dan kuantitas (ukuran) yang minimum. Melihat masalah-masalah tadi, maka pemecahannya adalah maksimalisasi kompresi, yaitu mengurangi tempat yang digunakan oleh data yang dimampatkan. Pemampatan data (*data compression*) merupakan salah satu kajian di dalam ilmu komputer yang bertujuan untuk mengurangi ukuran file sebelum menyimpan atau memindahkan data tersebut ke dalam media penyimpanan (*storage device*). Media penyimpanan seperti *floppy disk, hard disk* dan CD (*Compact Disc*) mempunyai kapasitas yang terbatas. Jika data yang akan disimpan pada media penyimpanan semakin bertambah dan berukuran besar, maka media penyimpanan tidak dapat menyimpan data tersebut karena melebihi kapasitas. Oleh karena itu, untuk mengatasi masalah ini digunakanlah pemampatan data.

Pada pengolahan citra terdapat enam jenis operasi pengolahan, yaitu peningkatan kualitas citra, restorasi citra, kompresi citra, segmentasi citra, analisis citra, dan rekonstruksi citra. Pada umumnya informasi yang ada dalam suatu citra terletak pada strukturnya. Agar mudah memahami suatu citra dapat dilakukan dengan menyederhanakan struktur citra tersebut. Ada banyak sekali metode kompresi data yang ada saat ini. Sebagian besar metode tersebut bisa dikelompokkan ke dalam salah satu dari dua kelompok besar, *Statistical Based* dan *Dictionary Based*. Contoh dari *Dictionary Based Coding* adalah *Lempel Ziv Welch* dan contoh dari *Statistical Based Coding* adalah *Huffman Coding* dan *Arithmetic Coding* yang merupakan algoritma terbaru.

Pemrosesan citra adalah ilmu untuk memanipulasi gambar, yang melingkupi teknik-teknik untuk memperbaiki atau mengurangi kualitas gambar, menampilkan bagian tertentu dari gambar, membuat sebuah gambar yang baru dari beberapa bagian gambar yang sudah ada, dan beberapa teknik manipulasi gambar lainnya. Suatu citra yang mempunyai kontras rendah dapat dihasilkan dari sumber citra dengan proses pencahayaan atau penerangan yang rendah atau karena adanya kesalahan *setting* pada saat pengambilan citra berlangsung.

2. METODOLOGI PENELITIAN

II.1. Pengertian Kompresi Citra

Kompresi citra adalah proses pemanfaatan citra yang bertujuan untuk mengurangi duplikasi data pada citra sehingga memory yang digunakan untuk merepresentasikan citra menjadi lebih sedikit daripada representasi citra semula. *Lossy Compression* merupakan kompresi citra dimana hasil dekompresi dari citra yang terkompresi tidak sama dengan citra aslinya, artinya bahwa ada informasi yang hilang, tetapi masih bisa ditolerir oleh persepsi mata. Metode ini menghasilkan rasio kompresi yang lebih tinggi dari pada metode *lossless*.Untuk kompresi *Loseless,* berdasarkan cara mereduksi data yang akan dikompresi, terbagi lagi menjadi 2 kelompok besar algoritma:

a. Algoritma berbasis entropi

Algoritma berbasis entropi, atau disebut juga berbasis statistik, menggunakan model statistik dan probabilitas untuk memodelkan data, keefisienan kompresi bergantung pada berapa banyak karakter yang akan digunakan dan seberapa besar distribusi probabilitas pada data asli. Contoh algoritma yang berbasis entropi adalah: *Huffman Coding, Adaptive Huffman, dan Shannon Fano.*

b. Algoritma berbasis *Dictionary*

Algoritma berbasis *Dictionary*, bekerja dengan cara menyimpan pola masukan sebelumnya, dan menggunakan index dalam mengakses pola tersebut jika terdapat perulangan. Contoh algoritma yang berbasis dictionary adalah: LZ77, LZ78, LZW, DEFLATE, dan LZMA.

Proses kompresi melibatkan 2 proses, yaitu proses kompresi *(Compression),* dan dekompresi *(Decompression).* Pada proses *file* asli yang dibaca, lalu dilakukan pengkodean untuk membuat

file hasil kompresi. Proses ini disebut juga dengan proses *encoding* untuk membentuk *file* asli dari *file* yang sudah dimampatkan tersebut. Proses pengkodean kembali dilakukan. Proses ini disebut juga *decoding.*

Ada faktor mengapa citra sangat tepat dilakukan proses kompresi agar tidak terjadi korelasi yang signifikan antara *pixel* dengan *pixel* tetangga. Dan biasanya korelasi ini disebut dengan korelasi spasial. Dari faktor inilah muncul data berlebihan *(Redudancy).* Kompresi data dicapai dengan mengurangi *redudancy* (kelebihan data) tapi ini juga membuat data kurang dapat diandalkan, lebih rentan terhadap kesalahan.

Membuat data yang lebih handal, disisi lain dilakukan dengan menambahkan bit cek dan bit paritas, sebuah proses yang meningkatkan ukuran kode.

Menurut D, Putra (2010:269) proses kompresi bisa dapat mengakibatkan terjadinya kehilangan informasi pada citra hasil. Oleh karena itu dibutuhkan suatu kriteria untuk mengukur kebenaran hasil kompresi yang sering disebut *fidelity criteria*. Ada dua jenis kriteria yang digunakan untuk mengukur kebenaran orang tentang kualitas hasil kompresi.

II.2. Arithmetic Coding

Arithmetic Coding memiliki sejarah yang sangat penting karena pada saat itu algoritma ini sukses menggantikan *Huffman Coding* selama 25 tahun. *Arithmetic Coding* memiliki kelebihan terutama ketika memproses kumpulan abjad yang relatif sedikit. Awalnya *Arithmetic Coding* diperkenalkan oleh Shannon, Fano dan Elias. Kemudian dikembangkan oleh Pasco (1976), Rissanene (1976, 1984) dan Langdon (1984). Tujuannya memberikan ide alternatif yang pada saat itu setiap proses pengkodean dilakukan dengan menggantikan setiap simbol masukan dengan suatu *codeword* (Wijaya, Marvin., A. 2012. *Pengolahan Citra Digital menggunakan Matlab*. Bandung: Informatika). Sebagai gantinya, aliran simbol masukan digantikan dengan sebuah angka *single floating point.*

Pada umumnya, algoritma kompresi data didasarkan pada pemilihan cara melakukan penggantian satu atau lebih elemen-elemen yang sama dengan kode tertentu. Berbeda dengan cara tersebut, *Arithmetic Coding* menggantikan suatu deret simbol input dalam suatu *file* data dengan sebuah bilangan menggunakan proses aritmatika. Semakin panjang dan semakin kompleks pesan yang dikodekan, semakin banyak bit yang diperlukan untuk proses kompresi dan dekompresi data.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 didekompresikan sehingga menghasilkan deretan simbol yang dipakai untuk menghasilkan angka tersebut. Dapat didefinisikan *Arithmetic Coding* adalah suatu bagian dari *entropy encoding* yang mengkonversi suatu data ke dalam bentuk data yang lain dengan lebih sering menggunakan sedikit bit dan jarang menggunakan lebih banyak bit karakter. Teknik pengkodean ini memisahkan pesan masukan ke dalam simbol dan menukar masing -masing

Implementasi *Arithmetic Coding* harus memperhatikan kemampuan encoder dan decoder, yang umumnya mempunyai keterbatasan jumlah *mantissa*. Hal ini dapat menyebabkan kesalahan atau *error* apabila suatu *Arithmetic Coding* mempunyai kode dengan *floating point* yang sangat panjang. Sehingga diberikan solusi berupa modifikasi algoritma *Arithmetic Coding* dengan menggunakan bilangan *integer*. Modifikasi ini mampu mengatasi keterbatasan pengolahan *floating point* dalam melakukan kompresi dan dekompresi data. Modifikasi dengan bilang integer juga dipakai karena jumlah bit kodenya lebih sedikit dan mempercepat proses kompresi dan dekompresi data karena perhitungan *integer* jauh lebih cepat dari perhitungan *floating point* serta dapat diimplementasikan dalam program (Sutoyo, T , 2012. *Teori Pengolahan Citra Digital*. Semarang ,Andi)

Misalkan $S = \{s_1, s_2, \ldots, s_n\}$ adalah kumpulan simbol sumber dengan distribusi probabilitas kemunculan $P = \{p_1, p_2, \ldots, p_n\}$. Subinterval untuk masing-masing perulangan dapat berasal dari setiap simbol menurut probabilitas tersebut. Sebagai contoh, perulangan awal dari [0,1) dapat dibagi ke dalam n interval sebagai berikut :

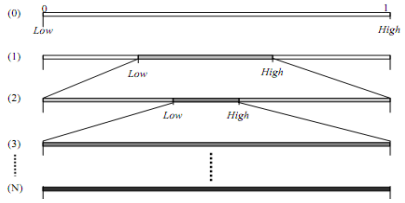
$(0,p_1)$
$(0_1,p_1+p_2)$
$(p_1+p_2,p_1+p_2+p_3)$

$(p_1+p_2+p_3+\ldots+p_{n-1}+p_2+p_3+\ldots+p_{n-1}+p_n)$

dimana $p_1 + p_2 + ... + p_n = 1$ untuk n simbol dalam abjad $s_1, s_2, ..., s_n$ secara berturut – turut. Jika simbol pertama yang dibaca adalah ke – i dari simbol s_i dalam abjad, maka batas kiri dari subinterval *low* adalah probabilitas kumulatif dari $P_i = p_1 + p_2 + p_3 + ...+ p_{n-1}$ dan batas kanan *high* adalah *low* + P_i. Panjang dari interval *high* – *low* kemudian dapat digunakan untuk menghitung satu dari beberapa interval pada perulangan berikutnya :

[*low*, *low* + (*high* - *low*) P_1)
[*low* + (*high* - *low*) P_1, *low* + (*high* - *low*) P_2)
.
.
.
[*low* + (*high* - *low*) P_{n-1}, *low* + (*high* - *low*) P_n)

dimana probabilitas kumulatif seluruhnya adalah $P_i = p_1 + p_2 + ... + p_n = 1$. Dengan menganggap suatu pesan sumber dengan kumpulan abjad ($s_1, s_2, ..., s_n$) dan distribusi probabilitasnya ($p_l, p_2, ..., p_n$) serta diumpamakan panjang sumber itu adalah N maka dapat dilihat setiap perulangan dari beberapa subinterval pada gambar 1 :



Gambar 1. Perulangan Setiap Interval [*low*, *high*) Sebanyak N

Berikut ini algoritma untuk *encoding* dan algoritma 2 untuk *decoding*. Akan digunakan dua variabel *low* dan *high* untuk mendefinisikan interval [*low*, *high*)

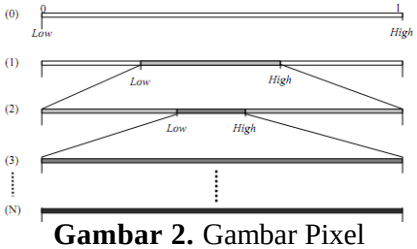
Algoritma Encoding
1: *low* ← *0.0*
2: *high* ← 1.0
3: while (simbol input masih ada)do
4: ambil simbol input
5: *CodeRange* ← *high - low*
6: *high* ← *low + CodeRange x high_range(s)*
7: *low* ← *low + CodeRange x low_range(s)*
8: end while 9: output *low*

Algoritma Decoding
1: ambil encoded symbol (ES)
2: repeat
3: cari range dari simbol yang melingkupi encoded symbol (ES)
4: cetak simbol
5: Code*Range* ← *high_range – low_range*
6: ES = ES – low_range
7: ES = ES/CodeRange
8: until simbol habis

3. HASIL DAN PEMBAHASAN

III.1. Hasil

Pada dasarnya contoh penggunaan metode *Arithmetic Coding* pada citra digital tidak berbeda jauh dengan *file file* lainya. Misalnya terdapat data citra digital kemudian citra tersebut dirubah dalam bentuk matriks seperti gambar 2 dibawah ini :



Gambar 2. Gambar Pixel

III.2. Pembahasan

Keterangan :

1. File photo diatas berukuran 670 Kb dengan nama file citra.bmp dengan resolusi 500 x 500 pixel
2. Setelah dilakukan proses kompresi maka ukuran file menjadi 510 Kb

Gambar diatas akan menghasilkan nilai decimal seperti tabel dibawah ini

Tabel 1. Data Pixel

R	G	B
255	75	50
255	25	75
25	50	25
50	75	120
120	180	50

R	G	B
200	80	50
255	25	25
25	80	25
50	75	120
120	180	50

R	G	B
255	50	80
255	25	25
25	75	25
50	75	120
220	120	50

R	G	B
50	75	120
255	25	25
25	75	75
50	75	120
120	180	90

R	G	B
120	180	50
255	25	25
25	75	25
50	75	120
120	180	50

R	G	B
255	50	50
120	180	50
25	75	25
50	75	120
120	180	50

R	G	B
255	50	50
255	25	25
25	80	25
50	75	120
120	180	50

R	G	B
255	50	50
255	25	25
25	75	25
50	75	120
120	180	50

R	G	B
255	50	50
255	25	25
25	75	25
50	75	120
120	90	50

Citra digital *f(x,y)* dapat dibayangkan sebagai sebuah matriks yang indeks baris dan kolomnya mengidentifikasikan sebuah titik pada citra dan nilai dari elemen matriks yang bersangkutan merupakan tingkat warna pada titik tersebut. Elemen tersebut disebut elemen citra, elemen gambar (*picture elements*), *pixels*, atau *pels*, dimana dua kata yang terakhir merupakan singkatan dari "*picture elements*". *Picture elements* atau *pixel* dapat didefinisikan sebagai elemen terkecil dari sebuah citra digital yang menentukan resolusi citra tersebut. Adapun data-data tersebut dapat dilihat pada table dibawah ini

Tabel 2. Potongan piksel data citra digital

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

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 3 seperti berikut.

Tabel 3. Tabel probabilitas dan *range*

Simbol	Frekuensi	Probabilitas	*Range*

255	12	1/3	0.00 – 0.33
50	6	1/6	0.33 – 0.5
25	9	¼	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

Pertama diambil simbol ’255’. Nilai *CodeRange* (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

Proses Decoding

Selanjutnya apabila citra tersebut akan di-*decode* maka digunakan metode *decoding* yang telah disebutkan diatas. Adapun citra yang akan di decoding (kompresi) adalah sebagai berikut:

Tabel 4. Data Decoding

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

Misalkan untuk tabel 4 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
CodeRange = 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 5 berikut.

Tabel 5 Proses *Decoding*

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

Misalkan untuk Tabel 6 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

CodeRange = 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 6 berikut.

Tabel 6 Hasil Proses *Decoding*

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.	0.75	0.25
0	50	0.33	0.5	0.17

4. KESIMPULAN

Berdasarkan pembahasan dan evaluasi dari bab – bab sebelumnya dan analisis terhadap *file* citra uji, maka dapat ditarik kesimpulan sebagai berikut :

1. Hasil kompresi format *.bmp* dapat menghasilkan citra dengan format *.JPG* dan hasil pengkompresan semakin kecil dari file asal
2. Algoritma *Arithmetic Coding* sangat baik di implementasikan untuk *file* citra karena meghasilkan rasio yang cukup besar. Selain itu, waktu kompresi dan dekompresi tidak memerlukan waktu yang lama. Kemudian yang paling penting hasil dekompresi *file* sama seperti semula. Hal ini dapat kita lihat dari hasil pengujian pada bab 4. Oleh karena itu, *Arithmetic Coding* bersifat *lossless.*
3. Algoritma *Arithmetic Coding* juga dapat di implementasikan untuk citra digital karena menghasilkan rasio yang jauh lebih besar dibandingkan *file* teks, akan tetapi pada saat proses dekompresi, kualitas citra rekonstruksi tidak seperti semula.

5. SARAN

Saran yang diusulkan untuk penelitian selanjutnya dapat dikembangkan dengan menambah atau menggunakan beberapa teknik untuk *Arithmetic Coding.*

DAFTAR PUSTAKA

[1] Fricles Ariwisanto Sianturi. Aplikasi Pembelajaran Citra Dengan Menggunakan Metode Computer Assisted Instruction (CAI). *J Ris Komput*. 2016;3(4):1-4. http://e-jurnal.pelitanusantara.ac.id/index.php/JIPN/article/view/292.

[2] Nelson, Mark. *Arithmetic Coding + Statis-tical Modeling = Data Compression*; Dr.Dobb's Journal, February 2012

[3] Fricles Ariwisanto Sianturi. Penerapan Metode Contrast Stretching Untuk Peningkatan Kualitas Citra Bidang Biomedis. J Mantik Penusa. 2015;18(2):70-75. http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/132.

[4] Press, William H. Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, 2016

[5] MacKay, David J.C.Information Theory, Inference and Learning Algorithms, http://wol.ra.phy.cam.ac.uk/mackay, 2012