

Kompresi File Citra Dengan Algoritma Transformasi Walsh-Hadamard

Eferoni Ndruru¹
Universitas Budi Darma
ronindruru@gmail.com

Abstrak

Kompresi citra dikembangkan untuk memudahkan penyimpanan dan pengiriman citra. Teknik kompresi yang ada sekarang memungkinkan citra dikompresi sehingga ukurannya menjadi jauh lebih kecil dari pada ukuran asli. Secara umum, metode kompresi data dapat dibagi ke dalam dua kelompok yaitu lossy dan lossless. Lossy merupakan kompresi citra dimana hasil kompresi dari citra yang terkompresi tidak sama dengan citra aslinya karena ada informasi yang hilang, tetapi masih bisa ditolerir oleh persepsi mata. Mata dapat membedakan perubahan kecil pada gambar. Metode ini menghasilkan rasio kompresi lebih tinggi dari pada metode lossless. Pada penelitian ini algoritma yang digunakan untuk membandingkan kompresi data adalah Transformasi Walsh-Hadamard dan Run Length Encoding (RLE). Transformasi citra merupakan pokok bahasan yang sangat penting dalam pengolahan citra. Citra hasil proses transformasi dapat dianalisis kembali, diinterpretasikan dan dijadikan acuan untuk melakukan pemrosesan selanjutnya. Tujuan diterapkannya transformasi citra adalah untuk memperoleh informasi (feature extraction) yang lebih jelas yang terkandung dalam suatu citra. Transformasi Walsh-Hadamard merupakan suatu transformasi orthogonal yang menjadikan suatu signal menjadi suatu himpunan gelombang yang berbentuk tegak lurus (orthogonal) dan segi empat (rectangular). Algoritma Run Length Encoding (RLE) bekerja berdasarkan sederetan karakter yang berurutan. Algoritma ini bekerja dengan memindahkan pengulangan byte yang sama berturut-turut (secara terus menerus). Metode ini digunakan untuk mengompresi citra yang memiliki kelompok pixel yang berderajat keabuan yang sama.

Kata kunci—3-5 kompresi, Transformasi Walsh-Hadamard, kompleksitas, citra

Abstract

Image compression was developed to facilitate image storage and transmission. Current compression techniques allow the image to be compressed so that its size is much smaller than the original size. In general, data compression methods can be divided into two groups, namely lossy and lossless. Lossy is image compression where the compression result of the compressed image is not the same as the original image because there is missing information, but it can still be tolerated by eye perception. The eye can distinguish small changes to the image. This method produces a higher compression ratio than the lossless method. In this study, the algorithms used to compare data compression are the Walsh-Hadamard Transform and Run Length Encoding (RLE). Image transformation is a very important subject in image processing. The image resulting from the transformation process can be re-analyzed, interpreted and used as a reference for further processing. The purpose of applying image transformation is to obtain clearer information (feature extraction) contained in an image. The Walsh-Hadamard transformation is an orthogonal transformation that turns a signal into a set of waves in the form of perpendicular (orthogonal) and rectangular (rectangular). The Run Length Encoding (RLE) algorithm works based on a sequence of consecutive characters. This algorithm works by moving the repetition of the same byte in a row (continuously). This method is used to compress images that have groups of pixels with the same gray degree.

Keywords—3-5 Compression, Walsh-Hadamard transform, complexity, imagery

1. PENDAHULUAN

Perkembangan teknologi informasi ternyata berdampak pada perkembangan ilmu pengetahuan yang lain. Semuanya merupakan informasi yang sangat penting, detail karakteristik citra tidak boleh ada yang hilang. Sebelum ada teknologi informasi, citra-citra tersebut disimpan dalam bentuk film sehingga disimpan dalam jumlah yang besar dan tentu saja data citra tersebut memerlukan tempat penyimpanan yang besar. Dengan hadirnya teknologi informasi, citra-citra tersebut disimpan dalam bentuk file. Sayangnya, file-file jenis citra ini berukuran relative besar sehingga file-file tersebut belum dapat disimpan. Salah satu solusi memperkecil ukuran file adalah dengan melakukan kompresi citra. Dalam ilmu komputer kompresi citra bertujuan untuk meminimalkan jumlah bit yang diperlukan untuk merepresentasikan citra. Kompresi citra adalah waktu pengiriman data pada saluran komunikasi lebih singkat dan membutuhkan ruang memori dalam storage yang lebih sedikit.

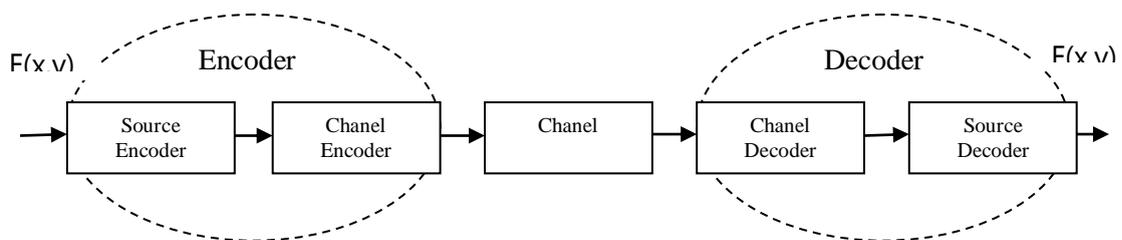
Pada penelitian ini algoritma yang digunakan untuk membandingkan kompresi data adalah Transformasi Walsh-Hadamard dan Run Length Encoding (RLE). Transformasi citra merupakan pokok bahasan yang sangat penting dalam pengolahan citra. Citra hasil proses transformasi dapat dianalisis kembali, diinterpretasikan dan dijadikan acuan untuk melakukan pemrosesan selanjutnya. Tujuan diterapkannya transformasi citra adalah untuk memperoleh informasi (feature extraction) yang lebih jelas yang terkandung dalam suatu citra. Transformasi Walsh-Hadamard merupakan suatu transformasi orthogonal yang menjadikan suatu signal menjadi suatu himpunan gelombang yang berbentuk tegak lurus (orthogonal) dan segi empat (rectangluar). Transformasi Walsh-Hadamard dapat didefinisikan dengan dua cara yaitu representasi rekursif dan biner. Kedua cara tersebut akan menghasilkan matriks yang sama. Apabila dari matriks tersebut diambil dua baris secara acak, maka dua baris matriks tersebut akan bersifat saling orthogonal.

Algoritma Run Length Encoding (RLE) bekerja berdasarkan sederetan karakter yang berurutan. Algoritma ini bekerja dengan memindahkan pengulangan byte yang sama berturut-turut (secara terus menerus). Metode ini digunakan untuk mengompresi citra yang memiliki kelompok pixel yang berderajat keabuan yang sama.

2. METODE PENELITIAN

2.1 Kompresi

Kompresi citra dapat di modelkan dan dibuat menjadi dua bagian utama yaitu bagian *encoder* dan *decoder*. *Encoder* berfungsi membuat representasi simbol-simbol (kode) dari citra *input* $f(x,y)$. Dengan kata lain *encoder* berfungsi membuat citra kompresi dari citra *input*. Setelah melalui transmisi pada kanal (*channel*), citra terkompresi tersebut masuk kesistem *decoder*. Pada sistem ini citra terkompresi akan di rekonstruksi kembali untuk menghasilkan citra *output* $f(x,y)$ [4].



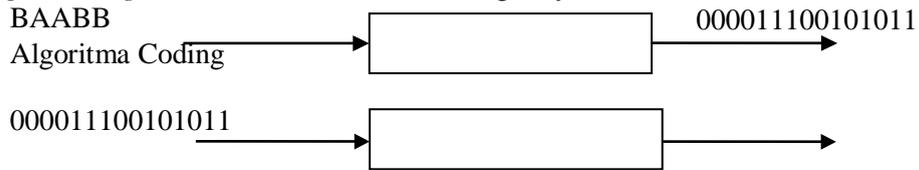
Gambar1 Model umum kompresi citra

Dari penjelasan diatas, dapat bagi 2 (dua) jenis kompres yaitu sebagai berikut :

1. Kompresi *Lossless*

Pada kompresi *lossless*, karena harus mempertahankan kesempurnaan informasi,

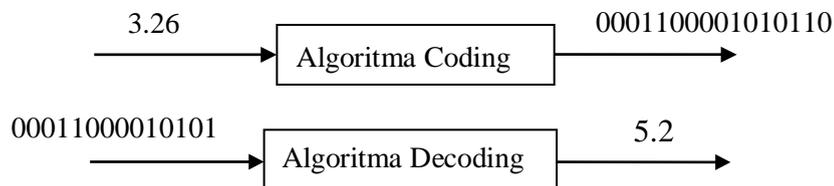
sehingga hanya terdapat proses *coding* dan *decoding*, tidak terdapat proses kuantisasi. Kompresi tipe ini cocok diterapkan pada berkas basis data, *spread sheet*, berkas *word processing*, citra bio medis, dan lain sebagainya.



Gambar 2 Ilustrasi kompresi lossless

2. Kompresi *Lossy*

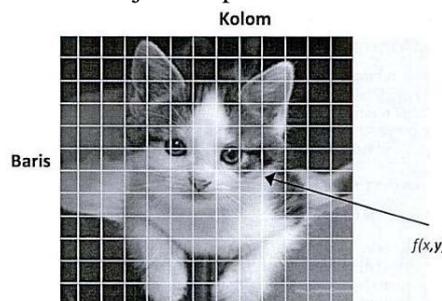
Kompresi data yang bersifat *lossy* mengizinkan terjadinya kehilangan sebagian data tertentu dari pesan tersebut, sehingga dapat menghasilkan rasio kompresi yang tinggi. Kompresi data *lossy* ini efektif jika di terapkan pada penyimpanan data analog yang Didigitalisasi seperti gambar, video dan suara[5].



Gambar 3 Ilustrasi kompresi *Lossy*

2.2 Pengertian Citra

Secara harfiah citra (*image*) adalah gambar pada bidang dwi matra atau dua dimensi. Citra juga dapat di artikan sebagai kumpulan titik-titik dengan intensitas warna tertentu yang membentuk suatu kesatuan dan mempunyai pengertian artistik. Citra sebagai salah satu komponen multimedia yang memegang peranan sangat penting sebagai salah satu bentuk informasi visual. Pada umumnya representasi citra digital membutuhkan memori yang besar. Semakin besar ukuran citra tentu semakin besar pula memori yang dibutuhkannya. Pada sisi lain, kebanyakan citra mengandung duplikasi data. Duplikasi data pada citra dapat berarti dua hal. Pertama, besar kemungkinan suatu *pixel* dengan *pixel* tetangganya memiliki intensitas yang sama, sehingga penyimpanan setiap *pixel* memboroskan tempat. Kedua, citra banyak mengandung bagian (*region*) yang sama, sehingga bagian yang sama ini tidak perlu di kodekan berulang kali karena redundan [2]. Suatu citra dapat di definisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitude f di titik koordinat (x,y) di namakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y, dan nilai amplitude f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Gambar 2.1 menunjukkan posisi koordinat citra digital [6].



Gambar 4 Ilustrasi Digitalisasi citra

Citra dapat ditulis dalam bentuk matrik sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0.0) & f(0.1) & \dots & f(0.N-1) \\ f(1.0) & f(1.1) & \dots & f(1.N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1.0) & f(M-1.1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 5 Posisi Koordinat Citra Digital

2.3 Walsh Hadamard Transform

Transformasi *Walsh Hadamard* dapat di gunakan untuk melakukan kompresi citra dengan sifat kompresi yaitu *lossy*

Untuk fungsi basis (kernel) dari transformasi *hadamard* dapat menggunakan rumus sebagai berikut.

$$g(x,u) = \frac{1}{N} (-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)} \dots \dots \dots (1)$$

Dengan $u = 0,1,2,\dots,N-1$, dan $x=0,1,2,\dots,N-1$, sedangkan nilai n mengikuti aturan $N=2^n$.

Rumus *hadamard* didalam pembentukan matriks transformasinya adalah sebagai berikut.

$$H(u) = \frac{1}{N} \sum_{x=0}^{n-1} f(x) (-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)} \dots \dots \dots (2)$$

Untuk fungsi balik menggunakan metode *hadamard* dapat menggunakan rumus berikut.

$$f(x) = \frac{1}{N} \sum_{u=0}^{n-1} H(u)(-1)^{\sum_{i=0}^{n-1} bi(x)bi(u)} \dots \dots \dots (3)$$

3. HASIL DAN PEMBAHASAN

3.1 Proses Kompres

Di dalam melakukan kompresi terhadap citra digital terdapat proses-proses perhitungan yang perlu diketahui, terlebih dahulu diambil sebuah citra dengan ukuran 3x2. Proses perhitungan ini digunakan untuk mempermudah di dalam menganalisa kemampuan dari setiap metode sejauh mana metode-metode yang digunakan bekerja dari segi kemampuan mengkompres citra, dan seberapa cepat metode tersebut dapat mengkompres citra input.

Kemudian masih dengan citra masukan yang sama dilakukan kompresi citra menggunakan metode *Walsh-Hadamard* dengan kernel *Hadamard*. Langkah pertama gabungkan seluruh data pada citra seperti sebelumnya. Kemudian hitung nilai kernel dari citra tersebut. Pertama sekali cari nilai b, untuk lebih jelasnya perhitungan kernel dari citra tersebut dapat dilihat pada perhitungan berikut ini.

f(x)=97 98 98 99 99 99
N=6
n=3
b(0) = 000
b(1) = 001
b(2) = 010
b(3) = 011

Setelah nilai b diperoleh langkah selanjutnya cari nilai g (x,u), nilai g (x,u) ini lah yang nanti nya akan digunakan untuk membentuk kernel dari metode *Walsh-Hadamard*. Untuk lebih jelasnya lihat perhitungan berikut ini:

Setelah nilai b diperoleh langkah selanjutnya cari nilai g (x,u), nilai g (x,u) ini lah yang nanti nya akan digunakan untuk membentuk kernel dari metode *Walsh-Hadamard*. Untuk lebih jelasnya lihat perhitungan berikut ini :

$$\begin{aligned} g(0,0) &= (-1)^{(0)(0)+(0)(0)+(0)(0)} = -1^0 = 1 \\ g(1,0) &= (-1)^{(1)(0)+(0)(0)+(0)(0)} = -1^0 = 1 \\ g(2,0) &= (-1)^{(0)(0)+(1)(0)+(0)(0)} = -1^0 = 1 \\ g(3,0) &= (-1)^{(1)(0)+(1)(0)+(0)(0)} = -1^0 = 1 \\ g(4,0) &= (-1)^{(0)(0)+(0)(0)+(1)(0)} = -1^0 = 1 \\ g(5,0) &= (-1)^{(1)(0)+(0)(0)+(1)(0)} = -1^0 = 1 \\ g(0,1) &= (-1)^{(0)(1)+(0)(0)+(0)(0)} = -1^0 = 1 \\ g(0,2) &= (-1)^{(0)(0)+(1)(0)+(0)(0)} = -1^0 = 1 \\ g(0,3) &= (-1)^{(0)(1)+(0)(1)+(0)(0)} = -1^0 = 1 \\ g(0,4) &= (-1)^{(0)(0)+(0)(0)+(0)(1)} = -1^0 = 1 \\ g(1,1) &= (-1)^{(1)(1)+(0)(0)+(0)(0)} = -1^0 = 1 \\ g(1,2) &= (-1)^{(1)(0)+(0)(1)+(0)(0)} = -1^0 = 1 \\ g(1,3) &= (-1)^{(1)(1)+(0)(1)+(0)(0)} = -1^1 = -1 \\ g(1,4) &= (-1)^{(1)(0)+(0)(0)+(0)(1)} = -1^0 = -1 \end{aligned}$$

$$g(1,5)=(-1)^{(1)(1)+(0)(0)+(0)(1)} = -1^1 = -1$$

$$g(2,1)=(-1)^{(0)(1)+(1)(0)+(0)(0)} = -1^0 = -1$$

$$g(2,2)=(-1)^{(0)(0)+(1)(1)+(0)(0)} = -1^1 = -1$$

Lakukan perhitungan sampai $g(5,5)$, setelah dilakukan perhitungan pencarian kernel *Walsh-Hadamard* diperoleh hasilnya pada tabel 3.1 berikut ini.

Tabel 1 Hasil kernel *Walsh-Hadamard*

x \ u	0	1	2	3	4	5
0	1	1	1	1	1	1
1	1	-1	1	-1	1	-1
2	1	1	-1	-1	1	1
3	1	-1	-1	1	1	-1
4	1	1	1	1	-1	-1
5	1	-1	1	-1	-1	1

Setelah kernel terbentuk langkah selanjutnya yaitu memproses nilai data di ikuti dengan nilai pada kernel $f(x) = (979898999999)$. Transformasi *Hadamard* dari citra $f(x)$ tersebut dapat di hitung dengan cara berikut.

$$H(u=0) = (97 + 98 + 98 + 99 + 99 + 99) / 6 = 98$$

$$H(u=1) = (97 - 98 + 98 - 99 + 99 - 99) / 6 = -0,33$$

$$H(u=2) = (97 + 98 - 98 - 99 + 99 + 99) / 6 = 33$$

$$H(u=3) = (97 - 98 - 98 + 99 + 99 - 99) / 6 = 0$$

$$H(u=4) = (97 + 98 + 98 + 99 - 99 - 99) / 6 = 32$$

$$H(u=5) = (97 - 98 + 98 - 99 - 99 + 99) / 6 = -0,33$$

Dari perhitungan di atas terdapat nilai 0, nilai 0 ini nantinya tidak akan di simpan. Hal ini di karenakan citra akan dilakukan kompresi. Untuk lebih jelas nya perhitungan rasio dari citra input menggunakan metode *Walsh-Hadamard* adalah sebagai berikut:

$$R = \frac{\text{Ukuran Asli} - \text{Ukuran Kompersi}}{\text{Ukuran Asli}} \times 100\%$$

$$R = \frac{6 - 3}{6} \times 100\% = 50\%$$

Jadi dari hasil perhitungan ke dua metode menggunakan citra masukan yang sama diperoleh bahwa citra lebih baik di kompres menggunakan metode *Walsh-Hadamard* ini. Metode ini cocok digunakan untuk contoh kasus seperti diatas, hal ini di sebabkan rasio kompres dari metode ini jauh lebih tinggi yaitu sebesar 50% di banding kan dengan nilai rasio percobaan sebelum nya yang mencapai nilai 16.67%

3.2 Proses Hasil Dekompresi

Dalam melakukan proses kompresi pastilah setelah itu dilakukan proses dekomposisi ulang, proses dekomposisi dilakukan untuk mengembalikan *file* citra ke bentuk aslinya dengan menggunakan metode yang sama pada saat melakukan kompresi terhadap *file* citra tersebut. Kemudian dilakukan proses dekomposisi pada *file* citra hasil kompresi menggunakan metode *Walsh-Hadamard* dan masih menggunakan kernel *Hadamard* seperti sebelumnya. Data dari proses kompresi sebelumnya adalah sebagai berikut.

$$f(x) = (98 - 0,33 \ 33 \ 0 \ 32 \ -0,33)$$

Langkah selanjutnya lakukan proses dekompresi ulang dengan memanfaatkan kernel yang ada. Untuk lebih jelasnya mengenai perhitungan dekompresi *Walsh-Hadamard* ini dapat dilihat pada hasil perhitungan berikut ini.

$$H(u=0) = (98 + (-0,33) + 33 + 0 + 32 + (-0,33)) / 6 = 27$$

$$H(u=1) = (98 - (-0,33) + 33 - 0 + 32 - (-0,33)) / 6 = 27$$

$$H(u=2) = (98 + (-0,33) - 33 - 0 + 32 + (-0,33)) / 6 = 16$$

$$H(u=3) = (98 - (-0,33) - 33 + 0 + 32 - (-0,33)) / 6 = 16$$

$$H(u=4) = (98 + (-0,33) + 33 + 0 - 32 - (-0,33)) / 6 = 16$$

Dari hasil dekompresi diperoleh nilai-nilai pada citra sebagai berikut dan juga disertakan nilai asli dari citra tersebut sebagai perbandingan.

Hasil Dekompresi = 27 27 16 16 16 16

Citra Asli = 97 98 98 99 99 99

Dari hasil dekompresi menggunakan metode *Walsh-Hadamard* diperoleh bahwa citra tersebut mengalami kehilangan data hal ini disebabkan metode kompresi *Walsh-Hadamard* bersifat *lossy* yang artinya metode ini mengizinkan terjadinya kehilangan data pada saat proses kompresi terhadap file citra tersebut. Jadi dapat disimpulkan bahwa metode *Walsh-Hadamard* ini tidak cocok untuk diterapkan pada citra digital, karena metode *Walsh-Hadamard* ini menyebabkan terjadinya kehilangan data terhadap citra yang dikompres.

5. KESIMPULAN

Dari hasil analisa dan uraian diatas, maka ditarik kesimpulan sebagai berikut :

1. Perangkat lunak kompresi/dekompresi citra dengan Algoritma *Walsh-Hadamard* dapat melakukan kompresi file citra dengan format bmp maupun jpg.
2. Kompresi citra dengan Algoritma *Walsh-Hadamard* dapat memperkecil ukuran file, sehingga dapat menghemat ruang tempat penyimpanan (*Storage*).
3. Algoritma *Walsh-Hadamard* dapat digunakan untuk kompresi teks, tidak hanya untuk citra

DAFTAR PUSTAKA

- [1] I. Gede Nengah Bayu Darmawan, G. Made Arya Sasmita, and P. Wira Buana, "Pengembangan Metode Pendeteksi Modifikasi Citra Menggunakan Metode Error Level Analysis," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 7, no. 1, p. 29, 2019, doi: 10.24843/jim.2019.v07.i01.p04.
- [2] E. Ndruru and T. Zebua, "Application of Text Message Held in Image Using Combination of Least Significant Bit Method and One Time Pad," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 13, no. 4, p. 323, 2019, doi: 10.22146/ijccs.46401.
- [3] V. Lusiana, "Teknik Kompresi Citra Digital untuk Penyimpanan File menggunakan Format Data XML," *Teknol. Inf. Din.*, vol. 19, no. 2, pp. 112–119, 2014.
- [4] W. R. Tresnaningsih, E. P. Purwandari, and D. Andreswari, "Deteksi Pemalsuan Citra Copy Move Menggunakan Dyadic Wavelet Dan Scale Invariant Feature Transform," *Pseudocode*, vol. 4, no. 1, pp. 18–28, 2017, doi: 10.33369/pseudocode.4.1.18-28.
- [5] E. Ndruru, "Penerapan Metode Error Level Analysis dan Laplacian of Gaussian Untuk Deteksi Tepi Citra CT Scan Paru-Paru," vol. 1, no. 1, pp. 34–38, 2021.
- [6] E. Ndruru, "Penerapan Algoritma C4 . 5 dalam Memprediksi Dampak Kebakaran Hutan bagi Kesehatan," 2019.