

# Penerapan Algoritma Transformasi Walsh-Hadamard dalam Pemampatan File BMP

Ainun Mardiah, Berto Nadeak

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia

Email Penulis Korespondensi: [ainun.riga1998@gmail.com](mailto:ainun.riga1998@gmail.com)

Submitted: 01/02/2022; Accepted: 27/02/2022; Published: 28/02/2022

**Abstrak**—Ukuran *file* citra berformat *bmp* relatif besar dimana semakin baik kualitas citra yang dihasilkan, maka ukuran *pixel* yang dibutuhkan untuk merekam citra tersebut semakin besar, proses kompresi adalah salah satu alternatif yang sangat dibutuhkan untuk memperkecil kapasitas suatu *file* citra, agar tidak terlalu boros dalam menggunakan ruang penyimpanan. Dengan ukuran suatu data yang telah dilakukan proses pengecilan atau pengurangan *bit* terhadap suatu *file* citra, pada saat melakukan proses pemindahan atau pengiriman *file* citra yang telah dilakukan pengurangan *bit*, maka proses pemindahan terhadap *file* akan semakin mudah dan cepat. karena media ruang penyimpanan memiliki ruang yang cukup untuk menampung *file* citra yang telah dilakukannya pengurangan *bit* atau proses kompresi terhadap suatu *file* yang telah di pindahkan tersebut. Teknik kompresi pada aplikasi menggunakan algoritma *Transformasi walsh hadamard* menghasilkan dari citra terkompresi serta merubah ukuran size awal, isi karakter citra, dan parameter untuk kompresi *Cr* (*Compression Ratio*) dalam mengurangi ukuran *file*.

**Kata Kunci:** Kompresi; File BMP; Algoritma *Walsh-Hadamard*

**Abstract**—The size of the image file in the *bmp* format is relatively large, where the better the quality of the resulting image, the larger the pixel size required to record the image, the compression process is one of the alternatives that is needed to reduce the capacity of an image file, so that it is not too wasteful in using space storage. With the size of a data that has been reduced or bit reduced to an image file, when moving or sending an image file that has been bit done, the process of moving the file will be easier and faster. because the storage space media has enough space to store an image file that has scattered bits or the compression process of a file that has been moved. The compression technique in the application uses the *Walsh Hadamard Transformation* algorithm to produce a compressed image and change the initial size, image character content, and parameters for compression of *Cr* (*Compression Ratio*) in reducing file size.

**Keywords:** Compression; BMP File; *Walsh-Hadamard* Algorithm

## 1. PENDAHULUAN

Pada umumnya kehidupan dalam bidang teknologi informasi, pemindahan data atau sebuah informasi tersebut sangat sering dilakukan melalui media penyimpanan ataupun melalui suatu media seperti internet. Kecepatan pengiriman suatu data atau informasi dalam suatu bentuk yang bertipe *file* citra akan menjadi bagian utama dalam pertukaran suatu informasi atau data dimasa yang akan datang. Besarnya suatu ukuran data yang akan dipindahkan, sangat dibutuhkan ruang penyimpanan yang cukup untuk suatu proses pemindahan, dan proses pemindahan bisa saja gagal dilakukan jika ukuran *file* yang akan dipindahkan lebih besar dari ruang yang tersedia, dan salah satu teknik yang dibutuhkan untuk masalah tersebut adalah teknik kompresi. kompresi bertujuan untuk memperkecil ukuran data sehingga dalam melakukan proses pemindahan atau dalam penyimpanan suatu *file* akan menjadi lebih mudah atau efisien.

Ukuran *file* citra berformat *bmp* relatif besar dimana semakin baik kualitas citra yang dihasilkan, maka ukuran *pixel* yang dibutuhkan untuk merekam citra tersebut semakin besar, proses kompresi adalah salah satu alternatif yang sangat dibutuhkan untuk memperkecil kapasitas suatu *file* citra, agar tidak terlalu boros dalam menggunakan ruang penyimpanan. Dengan ukuran suatu data yang telah dilakukan proses pengecilan atau pengurangan *bit* terhadap suatu *file* citra, pada saat melakukan proses pemindahan atau pengiriman *file* citra yang telah dilakukan pengurangan *bit*, maka proses pemindahan terhadap *file* akan semakin mudah dan cepat. karena media ruang penyimpanan memiliki ruang yang cukup untuk menampung *file* citra yang telah dilakukannya pengurangan *bit* atau proses kompresi terhadap suatu *file* yang telah di pindahkan tersebut.

Penelitian sebelumnya yang telah dilakukan oleh Hendri pada tahun 2014 telah berhasil membuktikan bahwa mengkompresi citra dari format BMP ke format PNG menggunakan metode *Lossless* dan dapat digunakan pada format PNG dimana citra yang dikompresi dengan format PNG dapat dikembalikan sama persis dengan data citra asli sebelum dikompresi dengan kedalaman warna pada format maksimum 48 bit per *pixel* memberikan kemampuan untuk menyimpan citra secara *lossless* dengan rasio kompresi yang tinggi. [1]

Sedangkan penelitian sebelumnya yang dilakukan Santi Ika Murpratiwi, Oka Widyantara pada tahun 2018 telah berhasil membuktikan bahwa pemilihan algoritma kompresi optimal untuk citra digital *Bitmap* yang dilakukan dengan membandingkan beberapa algoritma kompresi *lossy* untuk menemukan satu metode optimal yang cocok untuk citra digital *bitmap*. Algoritma kompresi yang dibandingkan *Absolute moment Block Truncation Coding* (AMBTC), proses perbandingan dilakukan dengan menghitung rasio kompresi, *Mean Square Error* (MSE) dan PSNR untuk mengetahui ukuran hasil kompresi dan kualitas citra hasil.[2]

Dengan hal tersebut penulis menggunakan algoritma *Transformasi Walsh-Hadamard* untuk mengetahui kinerja kompresi apabila dilakukan terhadap *file* BMP (citra berformat *\*bmp*), sehingga *file* citra yang berukuran

besar akan dikompresi menjadi ukuran yang lebih kecil, dimana *file* citra akan membutuhkan waktu pemindahan yang lebih singkat dibandingkan dengan *file* citra yang tidak terkompresi serta memperkecil ruang alokasi penyimpanan data.

## 2. METODOLOGI PENELITIAN

### 2.1 Kompresi Data

*File* kompresi dimaksudkan sebagai *file-file* yang digabungkan menjadi satu dengan tujuan untuk memperoleh ukuran *file* yang lebih kecil dibandingkan dengan *file* aslinya. *File* yang dikompresi memungkinkan *file* lebih cepat ketika di-download dan lebih banyak data yang tersimpan dalam media penyimpanan eksternal.[2]

Kompresi adalah proses mengubah aliran data *input* (sumber aliran atau data mentah asli) ke aliran data lain (*output*, *bitstream*, atau aliran terkompresi) yang memiliki ukuran lebih kecil. *Stream* dapat berupa *file*, *buffer memori*, atau *bit* individu yang dikirim pada saluran komunikasi[3].

### 2.2 Algoritma Transformasi Walsh-Hadamard

Transformasi walsh hadamard merupakan transformasi yang hanya memiliki fungsi basis dalam dua jenis nilai yaitu -1 dan 1. Transformasi walsh hadamard dapat digunakan untuk melakukan kompresi citra dengan sifat kompresi yaitu *lossy* dimana diperbolehkan terjadinya kehilangan beberapa data maupun sebagian besar data pada citra. Transformasi hadamard dapat menggunakan rumus sebagai berikut.

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

Dengan  $u = 0, 1, 2, \dots, N-1$  dan  $x = 0, 1, 2, \dots, N-1$ , sedangkan bilai  $n$  mengikuti aturan  $N=2^n$ . Dari rumus diatas  $N$  merupakan banyaknya data,  $bi(x)$  menyatakan bit ke- $i$  dari representasi biner  $x$ . Begitu juga berlaku untuk  $bi(u)$  dimana nilai bit-bitnya tergantung pada nilai  $u$ . Setelah diperoleh matriks walsh hadamard kemudian matriks tersebut dikalikan dengan nilai pixel pada citra kemudian dilakukan proses perhitungan walsh hadamard. Dalam metode ini rumus yang digunakan adalah rumus hadamard. Rumus hadamard didalam pembentukan matriks transformasinya adalah sebagai berikut.

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

Persamaan diatas digunakan untuk melakukan proses kompresi citra dimana bit merupakan posisi nilai biner dari  $x$  maupun  $u$ . Dimana  $f(x)$  merupakan nilai fungsi citra yang asli sebelum dilakukan proses kompres. Sedangkan nilai -1 merupakan nilai ketetapan dari metode *walsh-hadamard*. Untuk fungsi balik menggunakan metode hadamard dapat menggunakan rumus sebagai berikut.

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

Persamaan diatas menunjukkan rumus yang digunakan untuk melakukan proses dekomposisi terhadap citra yang telah dikompres. Yang membedakan fungsi kompresi dan dekomposisi hanya terletak dari nilai yang diproses, dimana pada proses dekomposisi ini nilai yang diproses merupakan nilai hasil kompresi citra yang mana nilai tersebut disimbolkan dengan  $H(u)$ . Nilai  $H(u)$  inilah yang digunakan untuk mendapatkan nilai balik yang terjadi menjadi hasil dari proses dekomposisi menggunakan metode *walsh-hadamard*.

## 3. HASIL DAN PEMBAHASAN

Citra merupakan salah satu media yang sering digunakan dalam berkomunikasi dan digunakan sebagai alat pengungkapan, ilustrasi, ingatan (*memorize*) dan lain sebagainya. Masalah yang sering terjadi adalah terkadang ukuran citra yang dihasilkan semakin besar, maka jumlah *pixel* yang dibutuhkan untuk proses pemindahan semakin lama, dan membutuhkan alokasi penyimpanan yang besar, sehingga sebagian banyak orang merasa terganggu dalam proses penyimpanan karena banyak memakan ruang dari penyimpanan yang disediakan penelitian ini bertujuan untuk melakukan analisa dari cara kerja dalam perancangan perangkat lunak pengkompresian file citra dengan menggunakan algoritma *Transformasi walsh hadamard*.

Alasan dilakukan kompresi file citra berbentuk bmp dengan algoritma *transformasi walsh hadamard* karena merupakan suatu transformasi yang sederhana dan diaplikasikan dalam kompresi data terkait proses transmisi citra dan penyimpanan. *Transformasi walsh hadamard* merepresentasi rekursif berbentuk biner dengan memanfaatkan pengkodean matriks yang sama apabila dari matriks tersebut diambil dua baris secara acak, maka dua baris matriks tersebut akan merubah ukuran suatu citra menjadi ukuran lebih kecil dari ukuran sebelumnya, perubahan pada ukuran citra yang dikompresi akan sedikit mempengaruhi kualitas dari citra tersebut. *file* citra yang tersimpan berupa *file bmp* dalam hal ini *file* yang akan dikompresi adalah berupa *file* berformat (\*.BMP) dengan maksimal 10 MB.

Dalam penelitian ini citra yang akan dilakukannya proses kompresi adalah sebuah citra bertipe *grayscale*. Citra yang bertipe *grayscale* akan dikompresi menggunakan algoritma *Transformasi walsh hadamard* dengan mengambil nilai *pixel* citra tersebut melalui media aplikasi matlab. Proses tersebut mengkompresi citra berdasarkan dari nilai *pixel* RGB berbentuk biner dengan memanfaatkan bilangan *Transformasi walsh hadamard* kemudian menggunakan parameter untuk kompresi Cr (*Compression Ratio*), Rc (*Ratio of Compression*), dan SS (*Space Saving*) akan mengurangi ukuran file berdasarkan algoritma *Transformasi walsh hadamard*. Adapun tujuan dari analisa terhadap sitem yang akan dibangun yaitu untuk mengurangi sumber daya yang berlebih.

### 3.1 Penerapan Algoritma Transformasi Walsh Hadamard

Dalam *Transformasi walsh hadamard* yang hanya memiliki fungsi basis bilangan pecahan dengan interval -1 sampai 1 (*sin* dan *cos*). *Transformasi walsh hadamard* dapat digunakan untuk melakukan kompresi citra dengan sifat kompresi yaitu *lossy* dimana diperbolehkan terjadinya kehilangan beberapa data maupun sebagian besar data pada citra. Proses kompresi dan dekomposisi hanya terletak dari nilai yang diproses, dimana pada proses dekomposisi ini nilai yang diproses merupakan nilai kompresi citra yang mana nilai tersebut disimbolkan dengan H(u). Nilai H(u) inilah yang digunakan untuk mendapatkan nilai balik yang menjadi hasil dari proses dekomposisi menggunakan algoritma *walsh hadamard*.

*File* citra terkadang memiliki ukuran yang besar dibandingkan dengan ukuran *file* lainnya. Dengan melakukan proses kompresi terhadap *file* yang berukuran besar, *file* tersebut akan dikompresi menjadi ukuran kecil, dan akan mengurangi alokasi penyimpanan. Dalam melakukan analisa terhadap *file* citra harus dilakukan proses pengambilan sampel *file*, untuk mengambil nilai *pixel* dari sempal *file* tersebut. Berikut adalah proses mengkompresi dan dekomposisi *file* citra.

1. Analisa terhadap proses kompresi file citra dengan menggunakan *Transformasi walsh hadamard*, langkah pertama dalam melakukan kompresi file citra adalah dengan memasukan file citra yang akan dikompresi yaitu file citra berformat BMP yang berukuran 720 x 720 pixel.



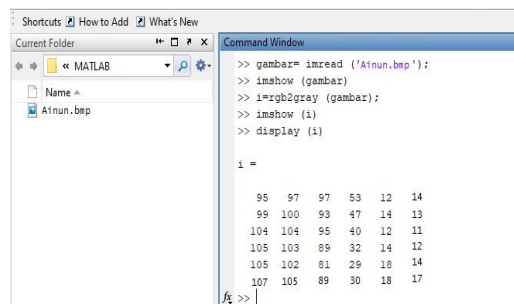
Gambar 1. File Citra.bmp asli

Berdasarkan citra diatas akan diambil sempel citra untuk keperluan hitung manual menggunakan algoritma *Transformasi walsh hadamard* resolusi yang akan diambil adalah 6 x 6 piksel. Citra sampel tersebut akan diambil nilai desimal dengan cara mengekstraksi elemen nilai setiap pixel.



Gambar 2. Sampel Citra 6 x 6 Pixel

Langkah selanjutnya adalah dengan melakukan pembacaan isi *file* citra. Didalam *file* citra terdapat nilai *pixel*. Nilai *pixel* diambil dalam bentuk 6 x 6 sehingga menghasilkan 36*pixel* sebagai sampel citra yang akan dianalisa. Nilai *pixel* yang akan dianalisa dapat dilihat di citra



Gambar 3. Matlab Citra.bmp 6x6 Pixel

Berdasarkan dari gambar 3 maka didapatkan nilai *pixel* karakter memperlihatkan matriks pada nilai-nilai piksel setiap baris dan kolom sebagai berikut:

Tabel 1. Nilai 36 titik pixel

95	97	97	53	12	14
99	100	93	47	14	13
104	104	95	40	12	11
105	103	89	32	14	12
105	102	81	29	18	14
107	105	89	30	18	17

Langkah selanjutnya yang harus dilakukan adalah nilai-nilai citra tersebut menjadi deretan satu baris sebagai berikut :

Fixel (x) = 95 97 53 12 **14 14** 99 100 93 47 **14** 13 104 104 95 40 12 11 105 103 89 32 **14** 12 105 102 81 29  
18 **14** 107 105 89 30 18 17

Jumlah nilai piksel : N = 36

Ortogonal : n = 5

$b_i(x)$  menyatakan bit ke-i dari representasi biner x, bila x = 14 dengan representasi biner 1110, maka :

$$b_0(x) = 0, \quad b_1(x) = 1, \quad b_2(x) = 1 \quad b_3(x) = 1 \quad b_4(x) = 1$$

$$b(0) = 000$$

$$b(1) = 001$$

$$b(2) = 010$$

$$b(3) = 011$$

$$b(4) = 100$$

$$b(5) = 101$$

Setelah nilai b diperoleh langkah selanjutnya cari nilai  $g(x,u)$ , nilai  $g(x,u)$  inilah yang nantinya akan digunakan untuk membentuk kernel dari metode yang digunakan ini. Lakukan perhitungan sampai  $b(5,5)$ , Untuk lebih jelasnya lihat perhitungan berikut ini.

- $g(0,0) = (-1)^{(0)(0) + (0)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(1,0) = (-1)^{(1)(0) + (0)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(2,0) = (-1)^{(0)(0) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(3,0) = (-1)^{(1)(0) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(4,0) = (-1)^{(0)(0) + (0)(0) + (1)(0)} = -1^\circ \sin \cos = 1$
- $g(5,0) = (-1)^{(1)(0) + (0)(0) + (1)(0)} = -1^\circ \sin \cos = 1$
- $g(0,1) = (-1)^{(0)(1) + (0)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(0,2) = (-1)^{(0)(0) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(0,3) = (-1)^{(0)(1) + (0)(1) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(0,4) = (-1)^{(0)(0) + (0)(0) + (0)(1)} = -1^\circ \sin \cos = 1$
- $g(1,1) = (-1)^{(1)(1) + (0)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(1,2) = (-1)^{(1)(0) + (0)(1) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(1,3) = (-1)^{(1)(1) + (0)(1) + (0)(0)} = -1^\circ \sin \cos = -1$
- $g(1,4) = (-1)^{(1)(0) + (0)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(1,5) = (-1)^{(1)(1) + (0)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(2,1) = (-1)^{(0)(1) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = -1$
- $g(2,2) = (-1)^{(0)(0) + (1)(1) + (0)(0)} = -1^\circ \sin \cos = -1$
- $g(2,3) = (-1)^{(0)(0) + (1)(0) + (1)(0)} = -1^\circ \sin \cos = -1$
- $g(2,4) = (-1)^{(0)(0) + (1)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(2,5) = (-1)^{(0)(0) + (1)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(3,1) = (-1)^{(1)(0) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(3,2) = (-1)^{(0)(1) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = -1$
- $g(3,3) = (-1)^{(0)(0) + (1)(0) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(3,4) = (-1)^{(0)(0) + (1)(0) + (1)(0)} = -1^\circ \sin \cos = -1$
- $g(3,5) = (-1)^{(0)(0) + (1)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(4,1) = (-1)^{(1)(0) + (0)(1) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(4,2) = (-1)^{(0)(1) + (0)(1) + (0)(0)} = -1^\circ \sin \cos = 1$
- $g(4,3) = (-1)^{(0)(0) + (0)(1) + (1)(0)} = -1^\circ \sin \cos = -1$
- $g(4,4) = (-1)^{(0)(0) + (1)(1) + (0)(0)} = -1^\circ \sin \cos = -1$
- $g(4,5) = (-1)^{(0)(0) + (0)(0) + (1)(1)} = -1^\circ \sin \cos = -1$
- $g(5,1) = (-1)^{(1)(0) + (0)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(5,2) = (-1)^{(0)(1) + (0)(0) + (0)(1)} = -1^\circ \sin \cos = 1$
- $g(5,3) = (-1)^{(0)(0) + (1)(0) + (0)(1)} = -1^\circ \sin \cos = -1$
- $g(5,4) = (-1)^{(0)(0) + (0)(1) + (0)(1)} = -1^\circ \sin \cos = 1$
- $g(5,5) = (-1)^{(1)(0) + (0)(0) + (1)(1)} = -1^\circ \sin \cos = 1$

Setelah lakukan perhitungan sampai  $f(5,5)$ , dilakukan perhitungan pencarian kernel *walsh hadamard* dari  $g(0,0), g(0,1), g(0,2), g(0,3), g(0,4), g(0,5), g(1,1), g(1,2), g(1,3), g(1,4), g(1,5), g(2,1), g(2,2), g(2,3), g(2,4), g(2,5), g(3,1), g(3,2), g(3,3), g(3,4), g(3,5), g(4,1), g(4,2), g(4,3), g(4,4), g(4,5), g(5,1), g(5,2), g(5,3), g(5,4), g(5,5)$  diperoleh hasilnya pada tabel 3.1 berikut ini :

**Tabel 2.** Hasil kernel *walsh hadamard*

$x \backslash u$	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 diikuti dengan nilai pada kernel.

Fixel (x) = 95 97 53 12 14 14 99 100 93 47 14 13 104 104 95 40 12 11 105 103 89 32 14 12 105 102 81 29 18 14 107 105 89 30 18 17

Transformasi hadamard dari citra  $f(x)$  tersebut dapat dihitung dengan cara berikut.

$$\begin{aligned}
 H(u=0) &= (95 + 97 + 53 + 12 + 14 + 14 + 99 + 100 + 93 + 47 + 14 + 13 + 104 + 104 + 95 + 40 + 12 + 11 + 105 + 103 + 89 + 32 + 14 + 12 + 105 + 102 + 81 + 29 + 18 + 14 + 107 + 105 + 89 + 30 + 18 + 17) \\
 &= 2087/36 \\
 &= 57,9
 \end{aligned}$$

$$\begin{aligned}
 H(u=1) &= (95 - 97 + 53 - 12 + 14 - 14 + 99 - 100 + 93 - 47 + 14 - 13 + 104 - 104 + 95 - 40 + 12 - 11 + 105 - 103 + 89 - 32 + 14 - 12 + 105 - 102 + 81 - 29 + 18 - 14 + 107 - 105 + 89 - 30 + 18 - 17) \\
 &= 323 \\
 &= 323/36 \\
 &= 5,87
 \end{aligned}$$

$$\begin{aligned}
 Hs(u=2) &= (95 + 97 - 53 - 12 + 14 + 14 - 99 - 100 + 93 - 47 - 14 + 13 + 104 - 104 - 95 + 40 - 12 - 11 + 105 + 103 - 89 - 32 + 14 - 12 - 105 + 102 + 81 - 29 - 18 + 14 - 107 - 105 + 89 - 30 - 18 + 17) \\
 &= -97 \\
 &= -97/36 \\
 &= -2,69
 \end{aligned}$$

$$\begin{aligned}
 H(u=3) &= (95 - 97 - 53 + 12 + 14 - 14 - 99 + 100 + 93 - 47 - 14 + 13 + 104 - 104 - 95 + 40 + 12 - 11 - 105 + 103 + 89 - 32 - 14 + 12 + 105 - 102 - 81 + 29 + 18 - 14 - 107 + 105 + 89 - 30 - 18 + 17) \\
 &= 13 \\
 &= 13/36 \\
 &= 0,36
 \end{aligned}$$

$$\begin{aligned}
 H(u=4) &= (95 + 97 + 53 + 12 - 14 - 14 - 99 + 100 + 93 + 47 - 14 - 13 - 104 + 104 + 95 + 40 - 12 - 11 - 105 + 103 + 89 + 32 - 14 - 12 - 105 + 102 + 81 + 29 - 18 - 14 - 107 + 105 + 89 + 30 - 18 - 17) \\
 &= 705 \\
 &= 705/36 \\
 &= 19,5
 \end{aligned}$$

$$\begin{aligned}
 H(u=5) &= (95 - 97 + 53 - 12 - 14 + 14 - 99 + 100 - 93 - 47 + 14 - 13 + 104 - 104 - 95 + 40 - 12 + 11 - 105 - 103 + 89 - 32 + 14 - 12 - 105 + 102 - 81 + 29 - 18 - 14 + 107 - 105 + 89 - 30 - 18 + 17) \\
 &= -331 \\
 &= -331/36 \\
 &= -9,19
 \end{aligned}$$

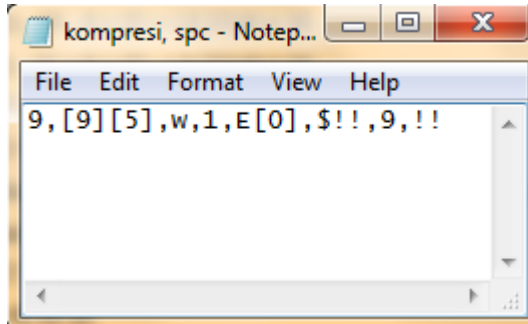
Maka *pixel* yang sudah dibagi kemudian diubah ke nilai kode ASCII untuk mengetahui nilai dari *pixel* yang sudah terkompresi. Adapun nilai *pixel* yang sudah didapat dari nilai yang sudah terkompresi dapat dilihat pada tabel sebagai berikut :

**Tabel 3.** Hasil nilai  $H(u)$  *walsh hadamard*

Nilai $H(u)$ terkompres	ASCII
57,9	9,[9]
5,87	[5],w
-2,69	1,E
0,36	[0],\$

Nilai H(u) terkompres	ASCII
19,5	!!,
-9,19	9,!!

Setelah nilai desimal diketahui, maka nilai akan di ubah menjadi suatu karakter. Karakter hasil dari proses kompresi yang dihasilkan, disimpan dalam suatu *file* dengan *ekstensi* ".spc", apabila *file* tersebut dibuka dengan aplikasi *notepad*, maka akan tampil karakter seperti gambar di bawah ini :



Gambar 4. Hasil Karakter Terkompresi

Dari hasil kompresi dengan nilai H(u) dapat dihitung kinerja dari metode tersebut menggunakan *Compression Ratio* (CR) yaitu :

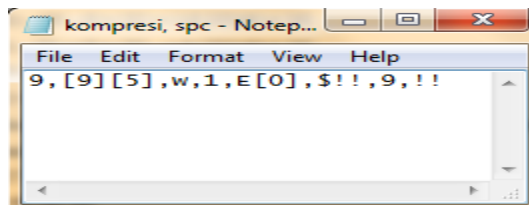
Nilai H(u) : Ukuran file sebelum dikompresi = 36  
 Ukuran file setelah dikompresi = 5

*Ratio of Compression (Rc)* =  $\frac{\text{Ukuran data sebelum di kompresi}}{\text{Ukuran data setelah dikompresi}}$   
 =  $\frac{36}{5}$   
 = 7,2

*Compression Rasio (Cr)* =  $\frac{\text{Ukuran data setelah di kompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100$   
 =  $\frac{5}{36} \times 100\%$   
 = 13,8

*Space Saving* =  $\frac{\text{Ukuran Asli} - \text{Ukuran data setelah di kompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100$   
 =  $\frac{36 - 5}{36} \times 100\%$   
 = 86,1%

2 Analisa proses dekomposisi *file* citra dengan menggunakan *walsh hadamard* Pada proses dekomposisi yang dilakukan adalah menganalisa karakter unik dari nilai H(u) yang terkompresi sebelumnya seperti gambar di bawah ini :



Gambar 5. Hasil Terkompresi H(u)

Proses untuk melakukan dekomposisi pada file citra yang telah dikompresi masih menggunakan karnel *hadamard* seperti sebelumnya. Data proses kompresi sebelumnya adalah sebagai berikut.

$$f(x) = (57,9 \quad 5,87 \quad -2,69 \quad 0,36 \quad 19,5 \quad -9,19)$$

Langkah selanjutnya lakukan proses dekomposisi ulang dengan memanfaatkan kernal yang ada. Untuk lebih jelasnya mengenai perhitungan dekomposisi *walshHadamard* ini dapat dilihat pada hasil perhitungan berikut ini.

$$\begin{aligned} H(u=0) &= (57,9 + 5,87 + (-2,69) + 0,36 + 19,5 + (-9,19)) \\ &= 71,75 \\ &= 71,75/36 \\ &= 1,99 \end{aligned}$$

$$\begin{aligned} H(u=1) &= (57,9 - 5,87 + (-2,69) - 0,36 + 19,5 - (-9,19)) \\ &= 64,67 \\ &= 64,67/36 \\ &= 1,79 \end{aligned}$$

$$\begin{aligned} H(u=2) &= (57,9 + 5,87 - (-2,69) - 0,36 + 19,5 + (-9,19)) \\ &= 89,41 \\ &= 89,41/36 \\ &= 2,48 \end{aligned}$$

$$\begin{aligned} H(u=3) &= (57,9 - 5,87 - (-2,69) + 0,36 + 19,5 - (-9,19)) \\ &= 60,01 \\ &= 60,01/36 \\ &= 1,66 \end{aligned}$$

$$\begin{aligned} H(u=4) &= (57,9 + 5,87 + (-2,69) + 0,36 - 19,5 - (-9,19)) \\ &= 38,13 \\ &= 38,13/36 \\ &= 1,05 \end{aligned}$$

$$\begin{aligned} H(u=5) &= (57,9 - 5,87 + (-2,69) - 0,36 - 19,5 + (-9,19)) \\ &= 44,05 \\ &= 44,05/36 \\ &= 1,22 \end{aligned}$$

Dari hasil dekomposisi diperoleh nilai-nilai pada citra sebagai berikut dan juga disertakan nilai asli dari citra tersebut sebagai perbandingan. Dari hasil dekomposisi 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. Dapat dilihat pada citra sebagai berikut ;



**Gambar 6.** Hasil Dekomposisi Sempel *File* citra

Adapun proses dekomposisi nilai  $N=36$  sama persis dengan proses dekomposisi nilai  $n=5$ .

## 4. KESIMPULAN

Berdasarkan hasil analisa yang telah dilakukan, maka penulis mengambil kesimpulan Teknik kompresi pada aplikasi menggunakan algoritma *Transformasi walsh hadamard* menghasilkan dari citra terkompresi serta merubah ukuran size awal, isi karakter citra, dan parameter untuk kompresi  $Cr$  (*Compression Ratio*) dalam mengurangi ukuran file. Algoritma *Transformasi walsh hadamard* membuktikan bahwa *file* citra yang semulanya memiliki ukuran besar dapat di kompresi menjadi ukuran kecil, dan akan memberikan manfaat ruang memori yang lebih sedikit. Aplikasi dibangun menggunakan bahasa pemrograman *Microsoft Visual Studio 2008*, dalam pembuatan *form* perancangan yang terdiri dari *input* dan tampilan *output* dalam pengimplementasian algoritma *Transformasi walsh hadamard*.

## REFERENCES

- [1] D. Salomon, *Data Compression*. USA: Springer, 2010.
- [2] S. S. D. T. Sutoyo, *Teori pengolahan citra digital*. Yogyakarta: Andi, 2009.
- [3] I. W. Pu, *Fundamental Data Compression*. London: Elsevier, 2006.
- [4] I. M. B. Bara, "PERANCANGAN APLIKASI KOMPRESI CITRA MENGGUNAKAN," no. PP, pp. 33–39.
- [5] A. S. A. Kadir, *Teori dan aplikasi pengolahan citra*. Yogyakarta: Andi, 2013.
- [6] S. M. S. Rosa A, *Rekayasa Perangkat Lunak*. Bandung: INFORMATI, 2016.
- [7] Hendrayudi, *Dasar-Dasar Pemrograman Microsoft Visual Basic 2008*. Bandung: PT. Sarana Tutorial Nurani Sejahtera, 2011.
- [8] R. H. S, *Visual Basic 2008 For Pemula Banget*. Jakarta. Jakarta: PT. Elex Media Komputindo, 2009.