**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

# Penerapan Algoritma *Rice Codes* Pada Aplikasi Kompresi *File* Gambar

**Putri Fitria**

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: putrifit@gmail.com

**Abstrak**−Seiring dengan berkembangnya teknoogi yang semakin canggih seperti saat ini maka semakin banyakpula informasi yang di perlukan, bermacam ragam informasi yang di perlukan seperti file gambar. file gambar adalah file yang mempunyai banyak karakter shingga selalu menyebabkan masalah pada penyimpanan memory. tingginya aktivitas penyimpanan dapat menimbulkan suatu masalah dimana diperlukannya tempat penyimpanan yang besar. Ukuran data yang semakin besar dapat mengakibatkan pemborosan ruang penyimpananalah satu untuk mengatasi permasalahan tersebut yaitu dengan cara mengkompresi mengunakan metode alagoritma rice codes dengan adanya metode tersebut maka dapat menyimpan lebih banyak informasi.Hasil yang diharapkan merupakan sebuah aplikasi yang bisa melakukan proses kompresi dari ukuran besar menjadi ukuran lebih kecil dan dapat di dekompresi untuk mengembalikan file yang telah dikompresi menjadi file asli atau file semula.

**Kata Kunci:** Kompresi, File gambar, Rice Codes

**Abstract**−Along with the development of technology that is increasingly sophisticated as it is now, the more information is needed, various kinds of information needed such as image files. Image files are files that have a lot of characters so that they always cause memory storage problems. high storage activity can cause a problem where a large storage space is needed. Increasing data size can result in a waste of storage space to overcome one of these problems, namely by compressing using the rice codes algorithm method with the existence of these methods, it can store more information. The expected result is an application that can perform the compression process from large size to smaller size and can be decompressed to restore files that have been compressed into original files or original files.

**Keywords**: Compression, Image File, Rice Codes

## 1. PENDAHULUAN

Secara sederhana Kompresi merupakan suatu proses untuk mengecilkan *file* dari ukuran aslinya. Pada saat ini aplikasi Kompresi yang sering digunakan adalah *WinZip* (menghasilkan format *zip*), *WinRar* (menghasilkan format *Ra*r) dan *7zip* (menghasilkan format archiver) yaitu dengan tujuan memampatkan dokumen dan menghemat ruang pada memori. Data yang dikompresi bisa berupa gambar, audio, video dan teks. *file* gambar merupakan suatu kumpulan dari karakter-karakter atau *string* yang menjadi satu kesatuan.di dalam *file* gambar yang mempunyai banyak karakter maka akan selalu menyebabkan masalah pada penyimpanan *memory*.

Ruang penyimpanan yang sedikit tidak dapat menyimpan *file* yang lebih banyak, seperti suatu media penyimpanan yang memiliki *space* kosong sebesar 10MB dan setiap gambar memiliki ukuran 2MB, *space* yang kosong tersebut penuh hanya dengan menyimpan 5 gambar saja, ukuran gambar yang besar membuat penyimpanan cepat penuh sehingga dapat menimbulkan masalah pada media penyimpanan, untuk mengatasi permasalahan tersebut yaitu dengan cara mengKompresi kan *file* gambar yang asli menjadi *file* gambar yang lebih kecil ukurannya dari ukuran semula, dan dapat menempati banyak ruang didalam *memory* sehingga *space* yang 10MB bisa menampung *file* gambar lebih banyak lagi.

Kompresi *file* gambar adalah proses untuk memperkecilkan suatu ukuran sehingga ukuran *file* gambar tersebut menjadi lebih kecil. Sebagai contoh pada *file* gambar citra dimana semakin bagus kualitas gambar yang telah dihasilkan, maka ukuran *pixel* yang dibutuhkan untuk merekam gambar tersebut semakin besar, sehingga berimbas pada ukuran *file* yang harus disimpan pada media penyimpanan. Ada dua tipe utama Kompresi , yaitu Kompresi *lossless* dan *lossy*. Kompresi *lossy* yaitu Kompresi yang akan hilang datanya jika sedang dalam proses Kompresi . sedangkan Kompresi *lossless* jika sedang di Kompresi maka data atau informasi tersebut tidak hilang.dan hasil kualitas citra yang telah di Kompresi tidak berkurang.

Dalam penelitian sebelumnya yang dilakukan oleh Erwin Dwika Putra untuk menganalisis perbandingan Kompresi gambar dan audio menggunakan algoritma Lempel Ziv Welch (LZW) disimpulkan bahwa *file* audio(*.wav) yang telah dikompresi menghasilkan 11,06% dan pada *file* gambar menunjukkan hasil yang berbeda-beda berdasarkan tingkatan dominan warna yang terdapat pada gambar tersebut, untuk dominan hijau dan campur menghasilkan *ratio* penyimpanan cukup rendah dibawah 50%, sedangkan dominan biru dan merah mencapai diatas 50% [1].

Dan pada penelitian sebelumnya oleh Rahmad syah, yang berjudul analisis perbandingan pemampatan data menggunakan algoritma *Rice coding* dan *Lemple Ziv Storer Symanski* (LZSS) pada jaringan client-server, disimpulkan bahwa hasil *file* citra yang telah diuji yaitu 80-90% dan rasionya yaitu 0.18%-6.08% dengan waktu yang digunakan 0-3 detik dan hasil yang telah di Kompresi pada algoritma *rice code* dan algoritma LZSS cukup bagus di bandingkan dengan *file* yang belum di Kompresi[2].


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

Dari sekian banyaknya metode Kompresi yang ada penulis hanya memilih metode *rice code*, metode *rice code* adalah suatu algoritma yang bisa kita gunakan untuk mengKompresi data yang berukuran besar sehingga data yang telah dikompresi menjadi kecil dari ukuran sebelumnya.algoritma *rice code* yaitu algoritma yang menggunakan dari sistem golom *coding*. Dimana teknik sistem tersebut akan menghasilkan kode-kode *prefix*. biasanya algoritma *rice code* di gunakan pada proses *encoding entropi* untuk mengKompresi tipe *file* gambar dan *file* audio. Oleh sebab itu pada kasus Kompresi data *string* hasilnya tidak berbeda dengan data awal yang digunakan.

Penggunaan algoritma *Rice codes* dalam Kompresi *file* gambar yaitu untuk memberikan manfaat yang sangat besar dalam penyimpanan serta membutuhkan ruang memori yang lebih besar untuk *file* yang lebih banyak dibandingkan dengan *file* gambar yang belum dikompresi .

# 2. METODOLOGI PENELITIAN

## 2.1 Kompresi

Kompresi adalah proses mengkonversikan sebuah aliran input data (sumber aliran data, atau asli) menjadi aliran data lainnya (aliran data dalam bentuk *bit*, atau *output* data yang telah dipadatkan) yang memiliki ukuran lebih kecil[3].

Sedangkan Kompresi data adalah proses mengkodekan informasi menggunakan *bit* atau *information* yang lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan suatu sistem enkoding tertentu. Contoh Kompresi sederhana yang biasa dilakukan misalnya adalah menyingkat kata-kata yang sering digunakan tapi sudah memiliki konvensi umum, misalnya kata "mama" dikompres menjadi kata "ma". Pengiriman data hasil Kompresi dapat dilakukan jika pihak pengirim yang melakukan Kompresi dan pihak penerima memiliki aturan yang sama dalam hal Kompresi data. Kompresi data menjadi sangat penting karena memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, memperkecil kebutuhan *bandwidth*. Teknik Kompresi bisa dilakukan terhadap data teks/*biner*, gambar (JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263)[4].

## 2.2 File Gambar

*File* adalah kumpulan berbagai informasi yang berhubungan dan juga tersimpan di dalam *secondary storage*, secara konsep *file* memiliki beberapa tipe ada yang bertipe data terdiri dari *numeric*, *character* dan *binary* atau *texs*, gambar, vidio, *slide* dan lain lain. Lalu ada juga *file* yang bertipe program. Atau definisi *file* adalah arsip ataupun data yang tersimpan di dalam komputer.

Gambar adalah sebuah perpaduan antara titik, garis, bidang dan warna yang berguna untuk mencitrakan sesuatu Citra / gambar *(image*) merupakan hal yang vital dan menjadi bagian integral dari kehidupan sehari-hari. Pada kepentingan tertentu, citra (gambar) digunakan sebagai alat untuk mengungkapkan pertimbangan (*reason)*, interpretasi, ilustrasi, penggambaran (*represent*), ingatan *(memorise)*, pendidikan, komunikasi, *evaluasi*, *navigasi*, survai, hiburan, dan lain sebagainya[6].

## 2.3 Algoritma *Rice Code*

*Rice code* adalah salah satu algoritma Kompresi yang dapat memperkecil ukuran data yang lebih kecil dari ukuran sebelumnya. Sebelum adanya *Rice code*, terdapat algoritma yang disebut sebagai *Golomb codes* yang merupakan *family*dengan *Rice code* karena memiliki persamaan yang bergantung pada pemilihan parameter m, dimana m adalah himpunan dari 2 (m = 2k). *Rice code*disebut juga sebagai *Golomb-Rice code*, yang diberi nama sesuai dengan penciptanya yaitu Robert F. Rice pada tahun 1979[3].

*Rice code* merupakan *special case* dari *Golomb codes* yang mana nilai x dikodekan k pertama digeser kekanan untuk mendapatkan nilai *unary coded*. Kemudian urutan terendah dari k nilai asli dari x dilanjutkan sebagai k yang bernilai *biner* (Moffat & Turpin 2002). *Rice code* menunjukkan penggunaan sebuah subset dari turunan *Golomb Code* untuk menghasilkan sebuah kode yang sederhana dan mudah diimplementasikan dalam kasus aritmatika *biner* secara efisien, dan keduanyadigunakan pada beberapa metode untuk Kompresi audio *lossless.*

Di dalam algortima *Rice*, ada sebuah nilai k yang artinya adalah banyaknya angka 1 pada *suffix* dari kode terKompresi . Dalam proses *encode*, dilakukan pemisahan pada *prefix* dan *suffix*. Ketika proses *decode*, *decoder* membaca *sign bit* dan lompat ke angka 0 pertama dari sebelah kiri, yang mana akan berlanjut kembali untuk penambahan *bit* pada k selanjutnya. Untuk nilai k dalam proses Kompresi menggunakan *Rice code*, dapat dilihat pada tabel 2.1.

1. Langkah – langkah kompresi *file* gambar pada algoritma *Rice Codes* :
    a. Tentukan terlebih dahulu *file* gambar yang ingin dikompresi.
    b. Hitung jumlah biner keseluruhan *file* gambar tersebut.
    c. Kelompokkan masing – masing biner yang sama kedalam tabel untuk menghitung jumlah frekuensi kemunculan.
    d. Nilai biner diurutkan ke dalam tabel dari frekuensi yang terbesar sampai frekuensi yang terkecil.


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

e. Masukkan nilai k=2 pada tabel yg frekuensi kemunculannya telah di urutkan.
f. Susun kode – kode yang telah dibuat pada tabel tersebut dengan posisi karakter pada *biner.*

2. Langkah – langkah dekompresi *file* gambar pada algoritma *Rice Codes* :
   a. Tentukan panjang *string bit* yang harus dibaca.
   b. Baca*string bit* dan lompat ke angka 0 pertama dari sebelah kiri, yang mana akan berlanjut kembali untuk penambahan bit selanjutnya.
   c. Pembacaan *string bit* dilakukan dari indeks terkecil sampai indeks terakhir dengan terus menambahkan nilai pada indeks sebelumnya

# 3. HASIL DAN PEMBAHASAN

## 3.1 Analisa Masalah

Analisa yang dilakukan dalam *file* gambar yaitu proses cara kerja kompresi file gambar. dan hanya mengarah pada kompresi *file* gambar yang berekstensi jpeg. Secara garis besar penulis melakukan analisa kompresi gambar untuk cara kerja dan hasil kompresi. Kompresi merupakan berkurangnya ukuran suatu *file* menjadi lebih kecil dari ukuran aslinya. Manfaat kompresi memang sangat menguntugkan untuk sebuah penyimpanan, *file* gambar yang mempunyai warna yang memiliki beberapa *pixel* dan disetiap *pixel* tersebut diwakili oleh sejumlah bit yang dapat menyebabkan banyak data yang berlebihan, hal ini selalu menjadi masalah pada penyimpanan memori. Dan dengan adanya kompresi, memori yang sedikit bisa lebih banyak memuat *file* gambar.

Adapun teknik pada kompresi *file* gambar ini adalah dengan mengganti nilai pixel yang berulang-ulang dengan suatu pola tertentu sehingga pixel yang berlebihan tersebut dapat memperkecil ukurannya. Algoritma dimulai dengan memberikan rangkaian string sebagai masukan, bagaimana menghasilkan keluaran algoritma berupa biner atau kode yang menterjemahkan setiap *pixel* masukan agar *pixel* tersebut mempunyai jumlah bit yang sedikit dibandingkan dengan *pixel* yang tidak dimampatkan. Dengan demikian, masalahnya adalah bagaimana memperoleh kode tersebut dengan frekuensi yang telah diurutkan dan tabel frekuensinya sebagai masukan dan kode biner yang lebih pendek sebagai keluaran.

Dalam menganalisa cara kerja algoritma *rice code* maka perlu adanya sebuah aplikasi yang akan mengetahui bagaimana proses yang dihasilkan dari algoritma dalam melakukan kompresi. Aplikasi yang dirancang ini yaitu aplikasi yang berbasis dektop. *Tools* yang digunakan untuk merancang aplikasi analisa penerapan yaitu dengan menggunakan *Microsoft Visual Basic.Net* 2008 sebagai alat bantu dalam melakukan *design* untuk kompresi dan dekompresi.

Kompresi *file* gambar dapat memperkecilkan ukuran. dari ukuran besar menjadi ukuran yang lebih kecil. cara kerja kompresi file gambar yaitu dimulai dari penginputan *file* gambar lalu di kompresi. file gambar yang telah di kompresi ukurannya akan semakin kecil.

Proses dekompresi yaitu mengembalikan file gambar terkompresi menjadi file gambar asli. Proses yang dilakukan pada dekompresi memulai dengan mengnputkan file gambar terkomresi kemudian di dekompresi setelah melakukan proses dekompresi maka file akan kembali semula file gambar jpeg.

## 3.2 Penerapan Algoritma *Rice Code*

Algoritma *rice code* memiliki sebuah nilai k yang artinya adalah banyaknya angka 1 pada *suffix* dari kode terkompresi. Dalam proses *encode*, dilakukan pemisahan pada prefix dan *suffix*. Ketika proses *decode*, decoder membaca *sign bit* dan lompat ke angka 0 pertama dari sebelah kiri, yang mana akan berlanjut kembali untuk penambahan bit pada k selanjutnya.

1. Proses Kompresi
   Proses kompresi dilakukan pada Gambar asli yang berukuran 1003x258 dan gambar tersebut di potong dengan ukuran 5x5. Pada gambar tersebut yang akan dikompresi hanya sampelnya saja. Sampel gambar yang akan di kompresi dapat dilihat sebagai berikut:

**Gambar 1.** Sampel yang akan di kompresi

Gambar yang di kompresi sebelumnya akan di masukkan ke aplikasi matlap yaitu untuk mendapat nilai pixel pada suatau Gambar.Untuk mendapat nilai pixel terlebih dahulu download aplikasi matlap, aplikasi matlap


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

ini di guanakan untuk mencari nilai pixel yang ada dalam sebuah gambar. Tampilan nilai pixel gambar pada aplikasi matlap dapat dilihat pada gambar berikut.

**Gambar 2.** Nilai Gambar pada aplikasi Aplikasi Matlab

a. Sebelum di kompresi
Setealah menetukan jumlah keseluruhan *pixel* dan menetukan jumlah *pixel* yang tidak berulang, kemudian *pixel-pixel* tersebut di susun kedalam tabel 1. seperti di bawah ini.

**Tabel 1.** Sebelum Kompresi

| No | Nilai Pixel | Binary | Bit (ASCII) | Frekuensi | Frekuensi x Bit |
|---|---|---|---|---|---|
| 1 | 87 | 01010111 | 8 bit | 3 | 24 |
| 2 | 89 | 01011001 | 8 bit | 2 | 16 |
| 3 | 85 | 01010101 | 8 bit | 2 | 16 |
| 4 | 86 | 01010110 | 8 bit | 1 | 8 |
| 5 | 84 | 01010100 | 8 bit | 1 | 8 |
| 6 | 88 | 01011000 | 8 bit | 1 | 8 |
| 7 | 156 | 10011100 | 8 bit | 1 | 16 |
| 8 | 161 | 10100001 | 8 bit | 3 | 24 |
| 9 | 165 | 10100101 | 8bit | 1 | 8 |
| 10 | 189 | 10111101 | 8 bit | 1 | 8 |
| 11 | 187 | 10111011 | 8 bit | 2 | 16 |
| 12 | 186 | 10111010 | 8 bit | 2 | 16 |
| 13 | 185 | 10111001 | 8 bit | 1 | 8 |
| 14 | 184 | 10111000 | 8 bit | 1 | 8 |
| 15 | 183 | 10110111 | 8 bit | 3 | 24 |
| | | Jumlah | | | 208 |

b. Setelah di kompresi
Pembuatan tabel data *pixe*l ang telah diurutkan berdasarkan frekuensi terbanyak dan kolom decimal diubah dengan nilai k=2 pada*rice code*, Maka hasilnya seperti pada tabel 2. :

**Tabel 2.** Sesudah di kompresi

| No | Nilai Pixel | Rce Code | Bit (ASCII) | Frekuensi | Frekuensi x Bit |
|---|---|---|---|---|---|
| 1 | 87 | 000 | 3 bit | 3 | 9 |
| 2 | 161 | 001 | 3 bit | 3 | 9 |
| 3 | 183 | 010 | 3 bit | 3 | 9 |
| 4 | 89 | 011 | 3 bit | 2 | 6 |
| 5 | 85 | 1000 | 4 bit | 2 | 8 |
| 6 | 156 | 1001 | 4 bit | 2 | 8 |
| 7 | 165 | 1010 | 6 bit | 1 | 6 |
| 8 | 187 | 1011 | 4 bit | 2 | 8 |


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

| No | Nilai Pixel | Rce Code | Bit (ASCII) | Frekuensi | Frekuensi x Bit |
|---|---|---|---|---|---|
| 9 | 186 | 10000 | 4 bit | 2 | 8 |
| 10 | 86 | 110001 | 5 bit | 1 | 5 |
| 11 | 84 | 110010 | 6 bit | 1 | 6 |
| 12 | 88 | 110011 | 6 bit | 1 | 6 |
| 13 | 189 | 110100 | 6 bit | 1 | 6 |
| 14 | 185 | 110101 | 6 bit | 1 | 6 |
| 15 | 184 | 110110 | 6 bit | 1 | 6 |
| | | Jumlah | | | 106 |

Dari nilai pixel Pixel 87. 89. 89, 87. 87, 85 85, 86 , 84 , 88, 156. 161. 161 161 . 165 . 189. 187 186 187 186 185 184 183 183 183 maka dapat bitnya. 000 011 011 000 000 1000 1000 110001 110010 110011 1001 001 001 001 1010 110100 1011 10000 1011 10000 110101 110110 010 010 010.

Dengan demikian, untuk menyimpan *file* gambar yang terdiri dari 25 pixel dengan menggunakan algoritma *rice codes* dibutuhkan 106*bit*.

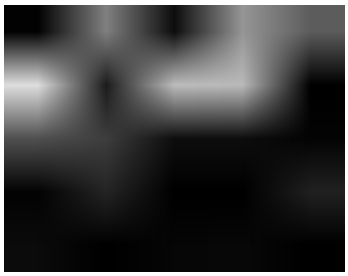
Sebelum mendapatkan hasil gambar terkompresi maka terlebih dahulu nilai binernya di susun dalam bentuk 8 digit biner tersebut adalah sebagai berikut:

0000110110000001000100011000111001011001110010010010011010110100101110000101110000110101110110010010010

| | | |
|---|---|---|
| 00001101=13 | 10000001= 129 | 00010001=17 |
| 10001110=142 | 01011001=89 | 11001001=201 |
| 00100110=38 | 10110100=180 | 10111000=184 |
| 00111000=56 | 01101011=107 | 10110010=178 |
| 01001000=72 | | |

Selanjutnya untuk mendapatkan hasil gambar terkompresi maka bit yang telah di susun 8 digit diubah ke decimal dan disusun kembali dalam bentuk matriks maka dapat dilihat gambar sebagai berikut:

$$\begin{bmatrix} 13 & 129 & 17 & 142 & 89 \\ 201 & 38 & 180 & 184 & 56 \\ 107 & 178 & 72 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



**Gambar 3.** Hasil gambar yang telah di kompresi

Sehingga total bit yang diperoleh adalah sebanyak 100 bit. Dan dari hasil kompresi tersebut dapat diukur kinerja algoritma *Rice Codes* sebagai berikut :

a. *Ratio of Compression* ($R_C$)

Ratio of Compression ($R_C$) adalah nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi.

$$R_C = \frac{\text{Jumlah bit sebelum dikompresi}}{\text{Jumlah bit sesudah dikompresi}}$$

$$R_C = \frac{208 \text{ bit}}{106 \text{ bit}} = 1.97$$

b. *Compression Ratio* ($C_R$)

*Compression Ratio*($C_R$) adalah persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.

$$C_R = \frac{\text{Jumlah bit sesudah dikompresi}}{\text{Jumlah bit sebelum dikompresi}} \times 100\ \%$$

$$C_R = \frac{106 \text{ bit}}{208 \text{ bit}} \times 100\% = 50.9\%$$


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

2. Proses Dekompresi
   Setelah *file* dikompresi, maka akan terbentuk sebuah *file* baru yang berekstensi jpeg yang lebih kecil bitnya dimana *file* tersebut yang nantinya akan digunakan untuk melakukan proses dekompresi. Proses mendekompresi adalah sebagai berikut:

a. Dilakukan dengan terlebih dahulu menentukan banyak pixel yang harus dibaca.Kemudian pada proses pembentukankode - kode *biner* hasil kompresi menggunakan algoritma *Rice Codes* disusun berdasarkan posisi *pixel* awal maka diperoleh susunan biner seperti dibawah ini :

Selanjutnya adalah dengan mengembalikan *binary* menjadi *pixel bit* semula. Untuk mengembalikan *binary* menjadi *string bit* semula dapat dilakukan melalui langkah-langkah berikut ini:
000011011000000100010001100011100101100111001001001001101011010010111000010111000011010111
0110010010010

Pembacaan *pixel bit*, kemudianlompat ke angka 0 pertama dari sebelah kiri, yang mana akan berlanjut kembali untuk penambahan *bit* pada *bit* selanjutnya. Pembacaan *pixel bit* dilakukan dari indeks terkecil sampai indeks terakhir dengan terus menambahkan nilai pada indeks sebelumnya. Indeks ke 0 adalah 0, tidak terdapat dalam tabel, indeks ke 1 adalah 00, tidak terdapat pada tabel, indeks ke 2 adalah 000, terdapat pada tabel, indeks ke 3 adalah 011, terdapat pada tabel bahwa 000 dan 011 adalah *nilai pixel* dari 87 dan 89, maka dituliskan kedalam *file*, begitu seterusnya sehingga hasil pixel dekompresi menjadi " 87 89 89 87 87 85 85 86 84 88 156 161 161 161 165 189 187 186 187 186 185 184 183 183 183

**Gambar 4.** Hasil gambar yang telah di dekomresi

### 3.3 Pengujian Aplikasi

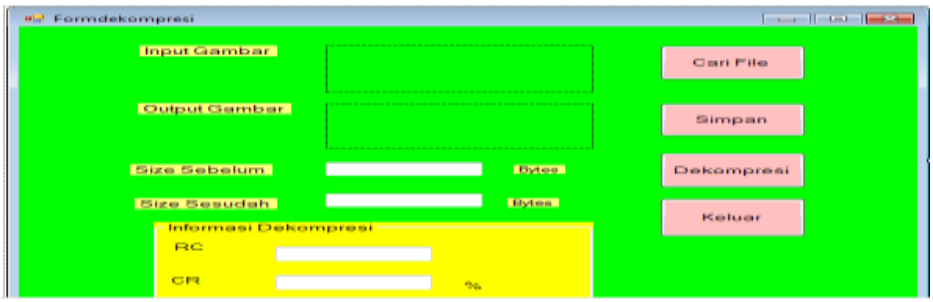
Pada *Form* ini akan ditampilkan proses kompresi *File* gambar berekstensi (jpeg) dengan menggunakan algoritma *Rice Codes.* Dimana dalam prosesnya terlebih dahulu *user* menginputkan *File* yang akan dikompresi dalam bentuk *File* gambar dan memilih lokasi penyimpanan *File* yang di kompresi. Selanjutnya diikuti dengan melakukan proses kompresi terhadap *File* tersebut. Adapun gambar proses tersebut dapat dilihat pada gambar dibawah ini :


**Gambar 5.** Tampilan *Form* Kompresi

Pada *Form* ini akan ditampilkan proses dekompresi *File* gambar.Dimana dalam prosesnya terlebih dahulu *user* menginputkan *File* yang akan di dekompresi dalam bentuk *File* gambar dan memilih lokasi penyimpanan*File* yang didekompresi. Selanjutnya diikuti dengan melakukan proses dekompresi terhadap *File* tersebut. Adapun gambar proses tersebut dapat dilihat pada gambar dibawah ini :


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

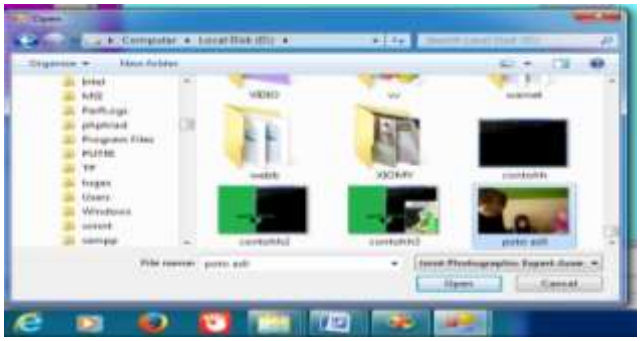
**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)




**Gambar 6.** Tampilan *Form* Dekompresi

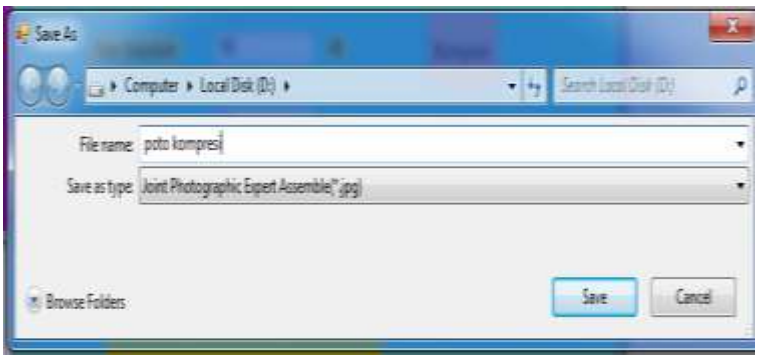
Hasil Pengujian ini memuat tentang hasil perangkat lunak yang di bangun berupa *print screen*-nya, tampilannya memuat tentang proses kompresi dan dekompresi pada algoritma *Rice Codes* serta menu keluar. untuk tampilan hasil dapat di lihat dibawah ini.

Pada proses kompresi maka yang pertama sekali dilakukan adalah memilih *File* gambar yang akan dikompresi dan memilih lokasi penyimpanan *File* gambar yang di kompresi. lokasi penyimpan tesebut dapat dilihat pada gambar.



**Gambar 7.** Lokasi *File* Gambar Yang Akan Dikompres


**Gambar 8.** Tampilan Proses *File* Gambar Yang Telah Dikompres

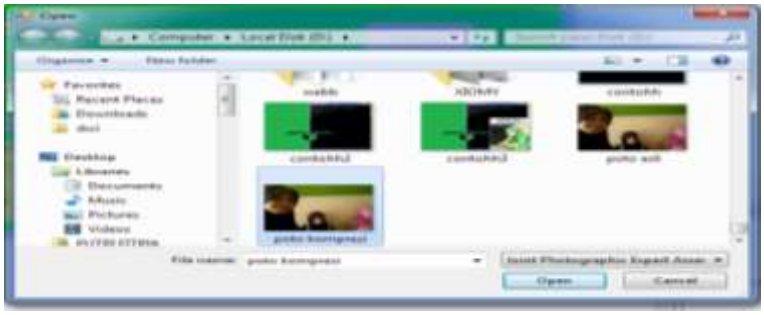



**Gambar 9.** Lokasi *File* Gambar Yang Telah Di Kompresi


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**

**Journal of Computer System and Informatics (JoSYC)**
Volume 1, No. 3, May 2020, Page 158-165
ISSN 2714-8912 (media online)
ISSN 2714-7150 (media cetak)

Pada proses dekompresi maka yang pertama sekali dilakukan adalah memilih *File* gambar yang akan didekompresi dan memilih lokasi penyimpanan *File* gambar yang telah di kompresi seperti pada gambar.



**Gambar 10.** Lokasi *File* Gambar Kompresi Yang akan Di Dekompresi


**Gambar 11.** Proses *File* Gambar Yang Telah Di Dekompresi

## 4. KESIMPULAN

Berdasarkan pembahasan dan evaluasi dari bab – bab sebelumnya dan analisa terhadap pengujian kompresi pada *file* teks maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Pada proses Kompresi *file* gambar yang berektensi jpeg mampu mengkompresi *file* gambar dan mendekompresi hasil kompresi tersebut dengan sangat baik.
2. Dalam penerapkan algoritma *Rice Codes* pada kompresi *file* gambar sangat bagus, hasilnya pun tidak berkurang dari *file* aslinya atau tidak ada pengurangan (isi *file* nya).
3. Aplikasi yang dirancang telah mampu melakukan kompresi *file* gambar dengan menggunakan algoritma *Rice Codes* dan melakukan dekompresi terhadap hasil kompresi dengan algoritma *Rice Codes* menjadi *file* asli.

## REFERENCES

[1] D. A. Erwin Dwika Putra, "Analisis Perbandingan Kompresi Gambar (*. bmp ) dan Audio (*. wav )," no. March, 2017.
[2] M. E. Rahmad Syah, "Analisis Perbandingan Pemampatan Data Menggunakan Algoritma Rice Coding Dan Lemple Ziv Storer Symanski (LZSS) Pada Jaringan Client-Server," pp. 1–6, 2013.
[3] David Salomon Giovanni motta, *Handbook Of Data Compression*. 2010.
[4] H. K. Iwan Fitrianto, "Kompresi file citra bitmap menggunakan algoritma rle dan lz78," vol. 3, no. 2, pp. 81–92, 2011.
[5] D. Putra, *Pengolahan Citra Digital*. C.V Andi OFFEST, 2010.
[6] J. Emitor *et al.*, "Membangun Aplikasi Kompresi Image Menggunakan Metode DPCM ( Differensial Pulse Code Modulation)," vol. 13, no. 2, pp. 49–56, 2017.
[7] C, Antonius, and Rachmat, *Algoritma dan Pemograman dengan Bahasa C*. C.V Andi, 2010.
[8] A. Kadir, *Pengenalan Algoritma Pendekatan Secara Visual dan Interaktif Menggunakan RAPTOR*. C.V Andi OFFEST, 2013.
[9] Heri, Sisworo, *Pengantar Logika Informasi, Algoritma Dan Pemrograman Komputer*. Yogyakarta: Andi, 2009.
[10] R.-A. S and M.Shalahuddin, *Rekayasa Perangkat Lunak*. 2011.
[11] R. Tantra, *Manajemen Proyek Sistem Informasi*. C.V Andi, 2012.
[12] Y. Sugiarti, *Analisis dan Perancangan UML [Unified Modeling Lnguage] General VB.6*. Yogyakarta: Graha Ilmu, 2013.
[13] W. KOMPUTER, *membagun Aplikasi Toko dengan Visual Basic 2008*. C.V Andi OFFEST, 2009.
[14] M. Ichwan, *Pemograman Aplikasi Database dengan Microsoft Visual Basic.net 2008*. Informatika, 2008.


Submitted: **18/04/2020**; Accepted: **17/05/2020**; Published: **26/05/2020**