

Aplikasi Kompresi Citra Dengan Menerapkan Algoritma SPIHT (Set Partitioning In Hierarchical Trees)

Milpadi Yanti

Prodi Studi Teknik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma

Jl. Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia

Email: 1milpadiyanti@gmail.com

Abstrak-Seiring dengan berkembangnya teknologi digital saat ini, dengan banyaknya peralatan digital yang dapat ditemui dalam kehidupan sehari-hari seperti komputer, kamera digital, handphone dan sebagainya. Dibandingkan dengan data teks, citra mempunyai karakteristik tersendiri, yaitu citra merupakan data yang kaya dengan informasi. Bahkan ada sebuah istilah yang cukup populer yaitu "a picture is more than a thousand words" yang berarti bahwa sebuah gambar dapat memberikan informasi yang lebih banyak jika dibandingkan dengan informasi yang disajikan dalam bentuk kata-kata atau tulisan. Tetapi pada sebuah gambar digital mempunyai kapasitas atau ukuran yang cukup besar, jika disimpan akan membutuhkan memori yang besar untuk menyimpan data tersebut, mengingat media penyimpanan sangatlah terbatas, dan jika harus dikirim maka akan dibutuhkan waktu yang cukup lama dalam proses penransmisian. Salah satu solusinya adalah dengan cara mengkompresi data citra tersebut. Dengan kompresi citra jumlah bit yang relatif sedikit sehingga transmisi citra dapat berjalan lebih cepat.

Kata Kunci : Aplikasi, Kompresi, Algoritma SPIHT

Abstract-Along with the development of digital technology today, with many digital equipment that can be found in everyday life such as computers, digital cameras, cellphones and so on. Compared with text data, image has its own characteristics, namely image is data that is rich in information. There is even a term that is quite popular, namely "a picture is more than a thousand words" which means that an image can provide more information than the information presented in words or writing. But on a digital image that has a capacity or size large enough, if it is stored it will require a large memory to store the data, considering the storage media is very limited, and if it has to be sent it will take quite a long time in the transmission process. One solution is to compress the image data. With image compression the number of bits is relatively small so that image transmission can run faster.

Keywords: Application, Compression, SPIHT Algorithm

1. PENDAHULUAN

Perkembangan teknologi digital banyaknya ditemui dalam kehidupan sehari-hari seperti komputer, kamera digital, *handphone* dan sebagainya. Hal ini menyebabkan terjadinya peningkatan yang cukup pesat terhadap permintaan kebutuhan komunikasi data digital, data atau informasi tidak hanya disajikan dalam bentuk teks, tetapi juga dapat berbentuk gambar dan video, atau yang sering disebut dengan sebagai multimedia dan salah satu yang cukup banyak diminati adalah pengolahan citra digital.

Dibandingkan dengan data teks, citra mempunyai karakteristik tersendiri, yaitu citra merupakan data yang kaya dengan informasi. Bahkan ada sebuah istilah yang cukup populer yaitu "a picture is more than a thousand words" yang berarti bahwa sebuah gambar dapat memberikan informasi yang lebih banyak jika dibandingkan dengan informasi yang disajikan dalam bentuk kata-kata atau tulisan. Berdasarkan penelitian terdahulu, menyatakan bahwa citra memiliki ukuran dan kapasitas yang besar, sehingga membutuhkan memori yang sangat besar untuk menyimpannya dan sangat berpengaruh pada waktu transmisi data[1].

Teknik kompresi citra merupakan teknik yang dapat digunakan agar diperoleh kualitas citra yang sesuai dengan yang diinginkan, tetapi dengan jumlah bit yang ditransmisikan kecil. Kompresi citra digital bekerja dengan cara mencari pola-pola perulangan atau biasa disebut dengan redundansi pada data dan menggantinya dengan sebuah penanda tertentu. Proses kompresi bertujuan untuk mereduksi ukuran data dengan mempertimbangkan kualitas data yang masih memadai untuk dapat dinikmati. Berdasarkan penelitian terdahulu, mengatakan bahwa kompresi dilakukan dengan tujuan mengurangi redundansi dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien. Metode kompresi citra mempunyai beberapa algoritma, antara lain adalah PNG, GIF dan JPEG. Beberapa diantaranya menggunakan transformasi seperti DCT (*Discrete Cosine Transform*) maupun DWT (*Discrete Wavelet Transform*). DWT mempunyai kemampuan mengelompokkan energi citra terkonsentrasi pada sekelompok kecil koefisien, mampu memberikan kombinasi informasi frekuensi dan skala, sehingga lebih akurat dalam rekonstruksi citra. Salah satu metode kompresi yang memanfaatkan transformasi wavelet adalah *Set partitioning in Hierarchical Trees* (SPIHT) merupakan salah satu algoritma kompresi citra yang mampu mencapai rasio kompresi yang tinggi[2].

Berdasarkan penelitian terdahulu, menyimpulkan bahwa kelebihan dari algoritma SPIHT ini adalah algoritma citra yang mampu mencapai rasio kompresi yang tinggi dengan cara mengkodekan koefisien hasil transformasi *wavelet* memberikan perbaikan pada kualitas gambar[2]. Umumnya sebuah citra memiliki duplikasi data didalamnya, yaitu suatu piksel mempunyai intensitas warna yang sama dengan piksel-piksel tetangganya. Semakin besar ukuran citra maka kemungkinan terjadinya duplikasi ini menjadi semakin besar pula. Prinsip umum yang digunakan pada proses kompresi citra adalah mengurangi duplikasi data pada citra sehingga memori yang dibutuhkan untuk merepresentasikan citra menjadi lebih sedikit dari pada representasi citra semula atau yang tidak terkompresi.

Penelitian ini membahas tentang kompresi citra, sehingga diharapkan proses kompresi citra dapat mengurangi redundansi dari data-data yang terdapat dalam citra sehingga dapat di simpan atau ditransmisikan secara efisien. Proses

pengompresan citra digunakan teknik algoritma SPIHT (*Set Partitioning In Hierarchical Trees*). Algoritma SPIHT (*Set Partitioning In Hierarchical Trees*) bekerja dengan cara mengolah kesamaan turunan antar *sub-band* dalam kompresi wavelet pada citra, sehingga mampu mencapai kompresi yang maksimal.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Metodologi penelitian dimulai dari pengumpulan data, melakukan penelitian, mendapatkan keterangan informasi untuk penelitian, penulis menggunakan bantuan berupa metode, adapun metodologi penelitian yang dilakukan oleh penulis yaitu:

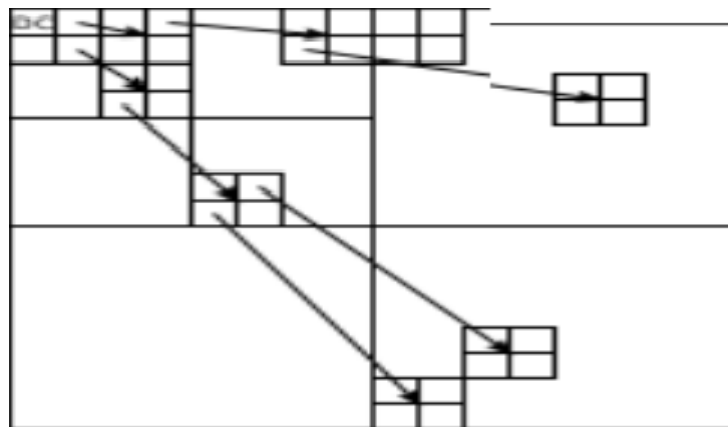
- StudiKepustkaan(*library Research*)
kegiatan untuk menghimpun informasi yang relevan dengan topik atau masalah yang menjadi obyek penelitian. Informasi tersebut dapat diperoleh dari buku-buku, karya ilmiah, tesis, disertasi, ensiklopedia, internet, dan sumber-sumber lain
- Analisis
Tahapan ini akan dilakukan kompresi citra dengan menerapkan algoritma SPIHT dan perancangan antar muka sistem.
- Perancangan Aplikasi
Melakukan perancangan aplikasi kompresi sesuai dari hasil pendeklarasian pada bagian analisa
- Uji Coba
Tahap ini dilakukan pengimplementasian aplikasi kompresi pada citra digital berdasarkan Algoritma SPIHT dan diuji dengan menggunakan aplikasi Matlab 2015.
- Dokumentasi
Pembuatan laporan penelitian dalam bentuk skripsi yang menjadi tugas akhir selama perkuliahan.

2.2 Kompresi Data

Proses kompresi citra bertujuan untuk mereduksi ukuran data dengan mempertimbangkan kualitas data yang masih memadai untuk dapat dinikmati dan mengurangi duplikasi data pada citra sehingga memori yang dibutuhkan untuk mempresentasikan citra menjadi lebih sedikit dari pada representasi citra semula atau yang tidak terkompresi [1]. Kompresi data bertujuan untuk meminimalkan kebutuhan memori dalam mempresentasikan citra digital dengan mengurangi duplikasi data di semula. Kompresi data berarti suatu teknik untuk memampatkan dalam citra sehingga memori yang dibutuhkan menjadi lebih sedikit dari pada representasi citra data agar diperoleh data [2].

2.3 Algoritma SPIHT (*Set Partitioning In Hierarchical Trees*)

Algoritma SPIHT (*Set Partitioning In Hierarchical Trees*) secara intensif menggunakan struktur data dinamis dari koefisien *wavelet*, untuk mengeksplotasi *selfsimilarities*. Hubungan *parentchild* pada koefisien *wavelet* ditunjukkan pada Gambar 1.



Gambar 1. Ilustrasi Hubungan *Parent-Child* dari Koefisien SPIHT

Hubungan *Parent-Child* dari Koefisien SPIHT pada Algoritma SPIHT koefisien-koefisien diklasifikasikan kedalam tiga set, yaitu[5]:

- LIP (*list of insignificant pixel*) merupakan koordinat dari koefisien yang tidak signifikan berdasarkan threshold saat ini.
- LSP (*list of significant pixel*) merupakan koordinat dari koefisien yang signifikan berdasarkan threshold saat ini.
- LIS (*list of insignificant sets*) merupakan koordinat dari akar dengan subpohon yang tidak signifikan.

Selama proses kompresi, set dari koefisien pada LIS diperbaharui dan jika koefisien menjadi signifikan dipindahkan dari LIP ke LSP. Dengan demikian *bitstream* dapat diorganisasi secara progressif. Dengan cara yang sama set secara berurutan

dievaluasi sesuai LIS, dan saat set yang ditemukan signifikan ia dihilangkan dari daftar dan dipartisi. Subset baru dengan lebih dari satu elemen ditambahkan kembali ke LIS, dengan set koordinat tunggal ditambahkan ke akhir LIP atau LSP, tergantung apakah mereka signifikan atau tidak. Algoritma pendekodean SPIHT menggunakan metode yang sama pengkodeannya, sehingga citra dapat direkonstruksi. Karena pada proses kompresi terjadi proses pemfilteran, maka citra hasil. Rekonstruksi akan mengalami distorsi. Distorsi dinyatakan dengan *Mean Square Error* (MSE):

a. Langkah-langkah Kompresi

Kompresi data bertujuan untuk meminimalkan kebutuhan memori dalam mempresentasikan citra digital dengan mengurangi duplikasi data di semula. Kompresi data berarti suatu teknik untuk memampatkan dalam citra sehingga memori yang dibutuhkan menjadi lebih sedikit dari pada representasi citra data agar diperoleh data. Proses kompresi adalah sebagai berikut[6]:

1. Tahap pertama citra akan direpresentasikan dalam bentuk matriks.
2. Kemudian berdasarkan representasi citra dalam bentuk matriks, maka akan disusun suatu senarai berantai dari baris pertama sampai baris terakhir. Setiap nomor warna merupakan satu simpul dalam senarai.

b. Langkah-langkah Dekompresi

Dekompresi citra adalah proses untuk meminimalkan jumlah bit yang merepresentasikan suatu citra sehingga ukuran citra menjadi lebih kecil. Proses dekompresi adalah sebagai berikut[6]:

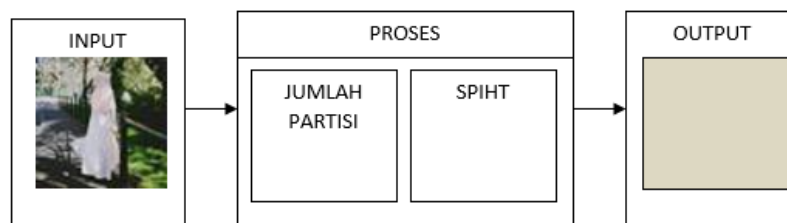
1. Ambil dua *byte* pertama dari sederetan karakter yang berurutan dalam suatu senarai berantai citra.
2. Lakukan pengulangan untuk perintah selanjutnya.
3. Mengecek apakah kedua *bytes* yang diambil apakah sama.
4. Jika sama maka kedua *bytes* tersebut di keluarkan, dan dihitung pengulangan yang terjadi untuk dicatat pada *counter*.
5. Keluaran nilai pencatat *counter* dengan karakter khusus.
6. Lakukan *update* terhadap pointer yang menunjuk pada deretan data yang terakhir.
7. Ulangi langkah nomor 2.
8. Jika tidak sama *byte* pertama di keluarkan, *byte* yang kedua dijadikan *byte* pertama, dan ambil *byte* berikutnya sebagai *byte* kedua kemudian lakukan *update* terhadap *pointer* yang menunjuk data *byte* yang terakhir dan ulangi langkah nomor 2.

2.4 Citra

Citra atau gambar adalah alat yang digunakan manusia untuk menyampaikan pesan kepada manusia lainnya. Citra merupakan kombinasi antara titik, garis, bidang, dan warna untuk menciptakan suatu imitasi dan suatu obyek, seperti obyek fisik dan obyek manusia[8].

3. HASIL DAN PEMBAHASAN

Algoritma *Set Partitioning in Hierarchical Trees* merupakan sebuah teknik kompresi yang akan merubah ukuran suatu gambar menjadi ukuran lebih kecil dari ukuran sebelumnya. Perubahan pada ukuran gambar yang dikompresi akan sedikit mempengaruhi kualitas dari gambar tersebut. Dalam penelitian ini, gambar yang akan dikompresi adalah sebuah gambar berwarna RGB. Gambar ini akan dikompresi dengan menggunakan algoritma *Set Partitioning in Hierarchical Trees* dengan mengambil nilai *pixel* gambar tersebut melalui media aplikasi matlab.



Gambar 2. Bagan kompresi SPIHT

Untuk melakukan analisa dengan tepat disediakan satu *file* sampel citra, sampel *file* citra yang disediakan yaitu milfa.jpg yang berukuran 1,90 KB (1.953 *bytes*) atau 2 KB resolusi 64x64 . Berikut merupakan sampel *file* citra yang akan dikompresi.



Gambar 3. Sampel citra input milfa.jpg 64x64

Berdasarkan sampel citra yang ada di atas, masukan pertama adalah *file* gambar yang belum dilakukannya proses kompresi.

3.1 Proses Set Partitioning In Hierarchical Trees (SPIHT) Algorithm

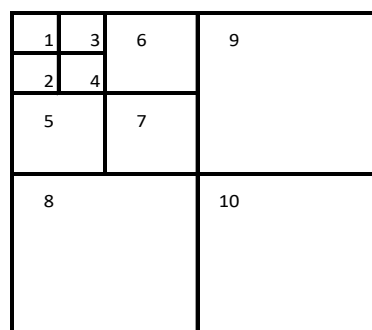
Proses *Set Partitioning in Hierarchical Trees* (SPIHT) dari beberapa sumber disampaikan dengan cara yang berbeda, akan tetapi tujuan dan maksud penyampain algoritma SPIHT adalah sama, yaitu menerapkan urutan partisi, dan menempatkan *pixel* untuk *output* dengan operasi ambang batas (*threshold*) pada setiap *pixel* kunjungannya. Ambang batas juga menggunakan parameter *List Insignificant Pixel* (LIP), *List Insignificant Set* (LIS), dan *List Set Pixel* (LSP). Berikut adalah tabel LIP, LIS, LSP yang digunakan sebagai acuan untuk kompresi sesuai partisinya.

Tabel 1. Partisi 10 matrik dengan parameter LIP, LIS, dan LSP

Partisi	LIP	LIS	LSP
0	0	240	256
1	24	216	232
2	48	192	208
3	72	168	184
4	96	144	160
5	120	120	136
6	144	96	112
7	168	72	88
8	192	48	64
9	216	24	40
10	240	0	16

Pada tabel di atas, dapat dilihat secara kolom tabel yang pertama adalah partisi, LIP, LIS, dan LSP. Partisi adalah jumlah matrik yang akan dibuat secara urutan sebanyak 9 matrik, sedangkan matrik partisi 0 adalah citra input jadi tidak digunakan. Untuk LIP, LIS dan LSP karena nilai tertinggi warna RGB adalah 255 dan partisi sebanyak 10 maka untuk mencapai ambang batas ke matrik 10 dilakukan penjumlahan matrik kelipatan 24 sehingga pada partisi 1 LIP ambang batasnya 24, berarti dimulai nilai 1 hingga 24 adalah bagian partisi 1. Partisi 2 dimulai dari nilai 25 hingga 48 adalah bagiannya, begitu seterusnya untuk partisi yang lain.

Untuk LIS dan LSP dilakukan pengurangan sebanyak 24. Tetapi pada LIS nilai awal citra inputnya adalah 240, ini dimaksudkan nilai warna ambang batas LIP sebelumnya dipatokan disini jadi nilai 255 hingga 240 adalah bagian citra input. Sedangkan untuk partisi 1 nilai $240 - 24 = 216$, sehingga nilai 240 hingga 216 adalah bagian partisi 1, dan begitu seterusnya dilakukan pengurangan untuk ambang batas partisi lainnya. Untuk LSP nilai awal partisi 0 atau citra input digunakan adalah 256, ini ditujukan untuk membatasi jarak antara LIS dengan LSP, seperti pada partisi 1 LIS 216 maka LSP 232. Matrik citra keabuan merah akan dikompresi berdasarkan kunjungan setiap pixelnya dengan tujuan penyesuaian kerangka partisi seperti pada gambar 4.



Gambar 4. Struktur partisi matrik SPIHT

Untuk mendapatkan posisi part 1 pada gambar di atas gambar 3.8 dilakukan dengan cara melihat nilai pembanding ambang batas (*threshold*) setiap partisi. Nilai banding part1 dengan LIP = 24 , LIS = 216, dan LSP = 232. Nilai tersebut dibandingkan dengan nilai citra matrik milpa.jpg keabuan merah = x yang diawali pada posisi 0,0 yaitu 115 (nilai 0,0 = 115)

Part1(0,0) :

$$\text{Jika nilai } x_0 > \text{LIP} \begin{cases} 1 & x_0 - \text{LIS} \\ 0 & x_0 - \text{LIP} \end{cases}$$

$$\text{True} = 115 > 24, \text{ maka } x_0 - \text{LIS}$$

$$= 115 - 216$$

$$= -101, \text{ nilai warna terkecil adalah 0, maka } -101 \text{ dianggap } 0$$

$$= 0$$

Part1(0,1) :

$$\text{Jika nilai } x_1 > \text{LIP} \begin{cases} 1 & x_1 - \text{LIS} \\ 0 & x_1 - \text{LIP} \end{cases}$$

$$\text{True} = 128 > 24, \text{ maka } x_1 - \text{LIS}$$

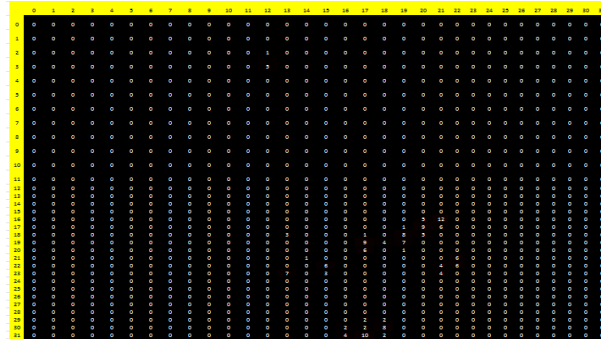
$$= 128 - 216$$

$$= -88, \text{ nilai warna terkecil adalah } 0, \text{ maka } -38 \text{ dianggap } 0$$

$$= 0$$

Lakukan hal yang sama sampai dengan langkah ke Part1 (0,7)

Dan begitu seterusnya untuk pengerjaan semua *pixel* part1 keabuan merah dan hasil dapat dilihat pada gambar 5 di bawah ini. Penilaian kuantitasi setiap *pixel* dengan melakukan *threshold* untuk Low Low (LL) pada R1 disebut part1.



Gambar 5. Matrik 32x32 merah *encoding* 1 LL disebut part1

Berikutnya nilai banding part2 dengan LIP = 48, LIS = 192, dan LSP = 208. Nilai tersebut dibandingkan dengan nilai citra matrik milpa.jpg keabuan merah = x yang diawali pada posisi 0,0 yaitu 115 (nilai 0,0 = 115)

Part2(0,0) :

$$\text{Jika nilai } x_0 > \text{LIP} \begin{cases} 1 & x_0 - \text{LIS} \\ 0 & x_0 - \text{LIP} \end{cases}$$

$$\text{True} = 115 > 48, \text{ maka } x_0 - \text{LIS}$$

$$= 115 - 192$$

$$= -77, \text{ nilai warna terkecil adalah } 0, \text{ maka } -60 \text{ dianggap } 0$$

$$= 0$$

Part2(0,1) :

$$\text{Jika nilai } x_1 > \text{LIP} \begin{cases} 1 & x_1 - \text{LIS} \\ 0 & x_1 - \text{LIP} \end{cases}$$

$$\text{True} = 128 > 48, \text{ maka } x_1 - \text{LIS}$$

$$= 128 - 192$$

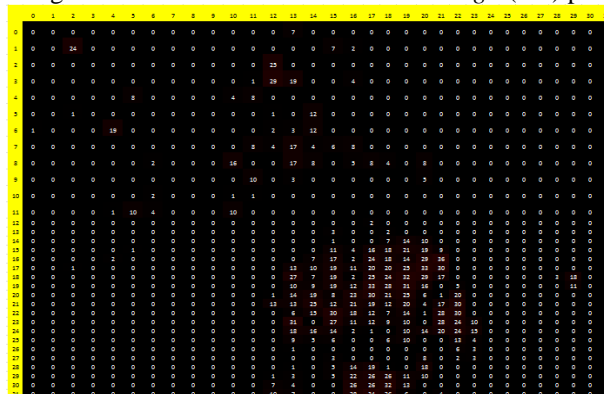
$$= -64, \text{ nilai warna terkecil adalah } 0, \text{ maka } -64 \text{ dianggap } 0$$

$$= 0$$

Lakukan hal yang sama sampai dengan langkah ke Part2 (0,7)

Begitu seterusnya untuk pengerjaan semua *pixel* part2 keabuan merah dan hasil dapat dilihat pada gambar 6 di bawah ini.

Penilaian kuantitasi setiap *pixel* dengan melakukan *threshold* untuk Low High (LH) pada R1 disebut part2.



Gambar 6. Matrik 32x32 merah *encoding* 1 LH3 disebut part2

Cara yang sama dilakukan untuk partisi lainnya hingga partisi terakhir. Dengan parameter tabel 1 sebelumnya. Dan berikut ini untuk hasil *encoding* part lainnya.

Gambar 7. Matrik 32x32 merah *encoding* 1 HH3 disebut part3

Gambar 9. Matrik 32x32 merah *encoding* 1 HL2 disebut part5.

Gambar 11. Matrik 32x32 merah *encoding* 1 LH1 disebut part7.

Gambar 13. Matrik 32x32 merah *encoding* 1 HH1 disebut part9.

Gambar 8. Matrik 32x32 merah *encoding* 1 LH2 disebut part4.

Gambar 10. Matrik 32x32 merah *encoding* 1 HH2 disebut part6.

Gambar 12. Matrik 32x32 merah *encoding* 1 HL1 disebut part8.

Gambar 14. Matrik 32x32 merah *encoding* 1 HH1 disebut part10.

Hasil kompresi SPIHT proses perhitungan rasio, dengan ukuran sampel $32 \times 32 = 16.384$ bit, 1 *Kbyte* (KiloByte) = 1024, 1Byte = 8 bit sehingga menghasilkan nilai prameter:

a. Rasio Kompresi

$$\text{Rasio} = 100\% - \left[\frac{\text{hasil kompresi}}{\text{data asli}} \times 100\% \right] \dots \dots \dots (1)$$

b. Redudancy

$$\text{Compression Ratio} = \frac{\text{Compressed Size}}{\text{Uncompressed Size}} \dots \dots \dots (2)$$

c. Save storage/save space.

$$\text{Space Saving} = 1 - \frac{\text{Compressed Size}}{\text{Uncompressed Size}} \dots \dots \dots (3)$$

Tabel 2. Ratio hasil kompresi.

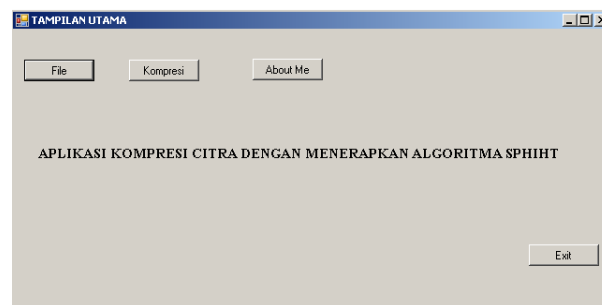
Kompresi	Ukuran	Ratio	Kompresi ratio	Save space
Citra Asli	3 KB	-1,49	0,66	-1,49
10	2,88 KB	-1,43	1,44	-1,43
9	2,75 KB	-1,37	1,37	-1,37
8	2,63 KB	-1,31	1,31	-1,31
7	2,50 KB	-1,24	1,25	-1,24
6	2,38 KB	-1,18	1,19	-1,18
5	2,25 KB	-1,12	1,12	-1,12
4	2,13 KB	-1,06	1,06	-1,06
3	2,00 KB	-0,99	1	-0,99
2	1,88 KB	-0,93	0,94	-0,93
1	1,75 KB	-0,87	0,87	-0,87

3.2 Implementasi

Kebutuhan sistem merupakan program yang mencakup spesifikasi perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*) yang dibutuhkan oleh aplikasi. Implementasi adalah pengerjaan suatu kegiatan yang bermuara pada aktivitas yang merupakan suatu proses kegiatan yang terencana dengan baik yang dilakukan dengan bersungguh-sungguh berdasarkan prosedur yang tertentu untuk mencapai suatu tujuan.

a. Tampilan kompresi

Pada form ini akan ditampilkan proses kompresi gambar berekstensi JPEG dengan menggunakan algoritma SPIHT. Dimana dalam prosesnya terlebih dahulu *user* menginputkan *file* yang akan dikompresi dalam bentuk file gambar dan memilih alokasi penyimpanan *file* yang akan dikompresi. Adapun gambar tampilan tersebut dapat dilihat pada gambar dibawah ini.



Gambar 15. tampilan awal bagian kompresi.

b. Tampilan dekompresi

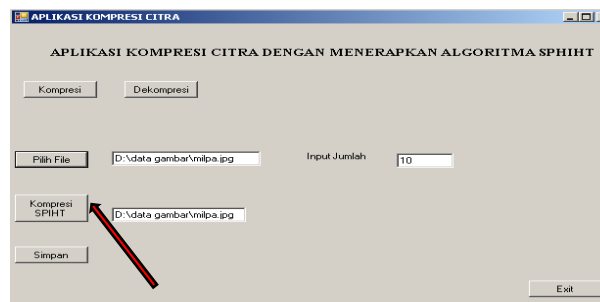
Pada form ini akan ditampilkan proses dekompresi gambar berekstensi JPEG dengan menggunakan algoritma SPIHT. Dimana dalam prosesnya terlebih dahulu *user* menginputkan *file* yang akan dikompresi dalam bentuk *file* gambar dan memilih alokasi penyimpanan *file* yang akan didekompresi. Adapun gambar tampilan tersebut dapat dilihat pada gambar dibawah ini.



Gambar 16. tampilan awal bagian dekompresi.

c. Proses kompresi

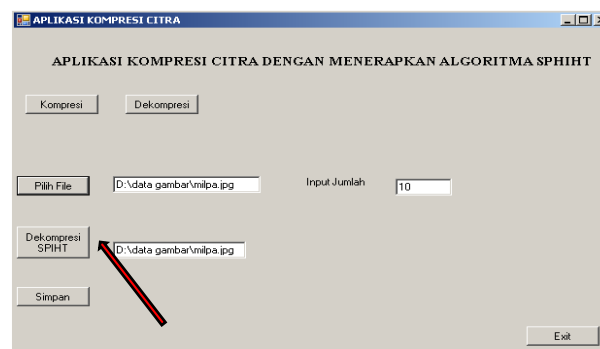
Pada proses kompresi maka yang pertama sekali dilakukan adalah memilih *file* gambar yang akan dikompresi dan memilih lokasi penyimpanan yang akan dikompresi. selanjutnya diikuti dengan proses kompresi pada *file* tersebut. Adapun tampilannya dapat dilihat pada gambar dibawah ini.



Gambar 17. Kompresi SPIHT

d. Proses Dekompresi

Pada proses dekompresi ini yang pertama sekali dilakukan adalah memilih *file* gambar yang akan di dekompresi dan memilih lokasi penyimpanan yang di kompresi atau di dekompresi. Selanjutnya diikuti dengan melakukan proses kompresi pada *file* tersebut. Untuk tampilan gambar *file* tersebut dapat dilihat pada gambar dibawah ini.



Gambar 18. Dekompresi SPIHT.

4. KESIMPULAN

Berdasarkan hasil analisa yang telah dilakukan maka penulis mengambil kesimpulan yaitu, Berdasarkan prosedur kompresi dengan menggunakan algoritma SPIHT telah melakukan proses *file* gambar, yang menggunakan sampel berekstensi *JPEG sehingga aplikasi yang diharapkan dapat berjalan sesuai dengan teknik kompresi. Berdasarkan penelitian ini yang telah menggunakan algoritma SPIHT, membuktikan *file* gambar yang semula memiliki ukuran *file* besar dapat dikompresi menjadi ukuran yang kecil, dan akan memberikan manfaat ruang memori yang lebih sedikit. *Tools* yang akan dirancang dan dibangun menggunakan bahasa pemrograman *Microsoft Visual Studio 2008* dan *Matlab 2015*.

REFERENCES

- [1] V. Lusiana, "Teknik Kompresi Citra Digital Untuk Penyimpanan File Menggunakan Format XML," *Teknol. Infomasi Din.*, vol.

- 19, p. 2, 2014.
- [2] C. T. Utari, "IMPLEMENTASI ALGORITMA RUN LENGTH ENCODING UNTUK PERANCANGAN APLIKASI KOMPRESI DAN DEKOMPRESI FILE CITRA," *J. TIMES*, vol. 5, no. 2337–3601, p. 2, 2016.
 - [3] Nia Maulidia, "Analisa Pengiriman Citra Terkompresi SPIHT Dengan Teknik Spread Spectrum Direct Sequence (DS-SS)," *J. MULTINETICS*, vol. 1, p. 1, 2015.
 - [4] and T. I. A. Satyapratama, M. Yunus, P. Studi, "KOMPRESI FILE GAMBAR BMP DAN PNG," *Ejurnal Stimata*, vol. 6, no. 69–81, p. 2, 2015.
 - [5] dan M. P. Rema, NR, Binnu Ani Oommen, "Image Compression Using SPIHT with Modified Spatial Orientation Trees," *Div. Electron. Eng. Sch. Eng. CUSAT Kochi-682022*, vol. 1732 – 173, 2014.
 - [6] C. T. Utari, "IMPLEMENTASI ALGORITMA RUN LENGHT ENCODING UNTUK PERANCANGAN APLIKASI KOMPRESI DAN DEKOMPRESI FILE CITRA," *TIMES*, vol. 2, no. 2337–3601, 2016.
 - [7] E. Iswandy, "SISTEM PENUNJANG KEPUTUSAN UNTUK MENENTUKAN PENERIMAAN DAN SANTUNAN SOSIAL ANAK NAGARI DAN PENYALURANNYA BAGI MAHASISWA DAN PELAJAR KURANG MAMPUDI KENAGARIAN BARUNG–BARUNG BALANTAI TIMUR," *TEKNOIF*, vol. 3, p. 2, 2015.
 - [8] PRISMAHARDI AJI RIYANTOKO, *RESTORASI CITRA PADA KOMPRESI SPIHT (SET PARTITIONING IN HIERARCHICAL TREES) MENGGUNAKAN METODE ITERATIF LANCZOS-HYBRID REGULARIZATION*. Surabaya, 2016.
 - [9] T. S. D. M. Pulung Nurtantio Andono, *Pengolahan Citra Digital*. Yogyakarta: ANDI, 2018.
 - [10] S. Dharwiyanti, Sri dan Wahono, Romi, (*UML*), *Pengantar Unified Modelling Language*. 2003.
 - [11] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen dengan Database Oracle (Studi Kasus : UIN Sumatra Utara Medan).," *J. Ilmu Komput. dan Inform.*, vol. 3, p. 1, 2018.
 - [12] A. . R. S. M, *Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Bandung: Modula, 2011.
 - [13] P. . Drs. Suarga, M.Sc., M.Math., *Algoritma dan Pemrograman*. Yogyakarta, 2012.
 - [14] R. Priyanto, *Langsung Bisa Visual Basic.NET 2008*. Yogyakarta: ANDI, 2009.
 - [15] B. Cahyono, "PENGUNAAN SOFTWARE MATRIX LABORATORY (MATLAB) DALAM PEMBELAJARAN ALJABAR LINIER," *PHENOMENON*, vol. 1, p. 1, 2013.
 - [16] N. Ahmadl and Arifyanto Hadinegoro2, "METODE HISTOGRAM EQUALIZATION UNTUK PERBAIKAN CITRA DIGITAL," *Semin. Nas. Teknol. Inf. Komun. Terap. 2012*, no. 979-26-0255–0, 2012.