BIMASATI (Bulletin of Multi-Disciplinary Science and Applied Technology)
ISSN : 2809-6096 (Media Online)
Vol 1, No 2, Februari 2022
Hal : 34-40

Analisa Kombinasi Algoritma Burrows Wheeler Transform dan Adaptive Huffman Coding untuk Kompresi Citra

Gusron Hasibuan

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: gusronhasbu@gmail.com

Abstrak−Citra adalah salah satu alat yang manusia pakai untuk menyampaikan pesan kepada manusia lainnya. Citra telah berkembang seiring dengan perkembangan peradaban manusia. Kombinasi Burrows Wheeler Transform(BWT) dengan Adaptive Huffman Coding(AHC) Proses Algoritma Adaptive Huffman Codingadalah membaca satu per satu dari karakter paling kiri hingga akhir, maka akan dilakukan perubahan data karakter awal menggunakan Burrows Wheeler Transformuntuk melakukan perubahan urutan dari karakter set yang akan dibaca oleh Adaptive Huffman Coding. ini bertujuan untuk mengkompresi citra dengan menggabungkan metode Burrows Wheeler Transformdan Algoritma Adaptive Huffman Coding untuk membuat kapasitas file gambar menjadi kecil sehingga dapat menghemat media penyimpanan dan tidak lambat jika pengiriman citra dari satu tempat ke tempat lain.Hasil kompresi tergantung pada pemilihan kualitas kompresi yang diinginkan. Jika memilih kompresi dengan kualitas standar, maka citra hasil kompresi dengan citra yang asli tidak akan terlihat perbedaannya namun pengurangan ukuran bytes tidak terlalu drastis. Tetapi apabila kita memilih kualitas kompresi rendah, maka ukuran bytes pada citra akan berkurang namun kualitas gambar hasil kompresi akan terlihat perbedaannya dengan citraasli.

Kata Kunci: Citra; Kompresi; BWT; AHC

Abstract−Image is one of the tools that humans use to convey messages to other humans. Image has developed along with the development of human civilization. Combination of Burrows Wheeler Transform (BWT) with Adaptive Huffman Coding (AHC) The process of Adaptive Huffman Coding Algorithm is reading one by one from the leftmost character to the end, it will change the initial character data using Burrows Wheeler Transform to change the order of the character set to be read by Adaptive Huffman Coding. This aims to compress the image by combining the Burrows Wheeler Transform method and the Adaptive Huffman Coding Algorithm to make the image file capacity small so that it can save storage media and is not slow when sending images from one place to another. Compression results depend on the selection of the desired compression quality. If you choose compression with standard quality, then the compressed image with the original image will not see the difference, but the reduction in bytes size is not too drastic. But if we choose a low compression quality, then the size of the bytes in the image will be reduced but the quality of the compressed image will be visible the difference with the original image.

Keywords: Image; Compression; BWT; AHC

1. PENDAHULUAN

Sejak ditemukannya alat untuk menangkap suatu gambar pada bidang dua dimensi (citra) berupa kamera, dengan semakin berkembangnya teknologi pada saat ini sehingga hal tersebut tidak hanya berfokus pada alat-alat yang digunakan untuk menangkap citra tersebut. Dengan menggunakan kamera digital, semuapersyaratan untuk penyimpanan,manipulasi dan transfer gambar digital dapat dilakukan, sehingga teknologi yang dapat mengelola suatu citra yang telah ditangkap juga merupakan hal yang sangat penting,karena citra yang telah ditangkap oleh kamera tersebut tidak dapat dipastikan akan menghasilkan citra yang baik dan sesuai dengan kebutuhan manusia. Sebagian besar data citra terdiri dari data multimedia dan mereka menempati sebagian besar dari *bandwidth* komunikasi untuk mengembangkan komunikasi multimedia.

Masalah pada citra adalah besarnya ruang penyimpanan yang diperlukan karena file-file gambar yang didapat sangatlah besar dan dapat menempati banyak ruang dalam media penyimpanan. Selain itu, data citra berukuran besar jika dikirim melalui jaringan akan membuat pengiriman citra dari satu tempat ke tempat lain menjadi lambat. Oleh karena itu dibutuhkan teknik yang efisien untuk dapat melakukan kompresi citra. Kompresi citra digital merupakan upaya untuk melakukan transformasi terhadap data atau simbol penyusunan citra digital menjadi data atau simbol lain, tanpa menimbulkan perubahanyang terlihat signifikan atas citra digital tersebut bagi mata manusia yang mengamatinya. Tujuannya adalah untuk mengurangi redudansi dari data-data yang terdapat dalam citra sehingga dapat di simpulkan atau ditransmisikan secara efisien.[1]

Pada algoritma kompresi data sangatlah banyak jenisnya salah satunya adalah Algoritma *Burrows Wheeler Transform* dan *Adaptive Huffman Coding*. Penggunaan data digital yang semakin merambah di berbagai aspek sering menimbulkan masalah dalam hal ruang penyimpanan dan transmisi data .Oleh karena itu, pengembangan teknik kompresi perlu dilakukan.Pengembangan algoritma kompresi seringkali didasari algoritma yang sudah ada atau berusaha mengkombinasikan dua atau lebih algoritma.

Burrows Wheeler Transform merupakan algoritma proses melakukan transformasi terhadap blok data citra menjadi suatu bentuk baru yang tetap mengandung karakter sama hanya saja urutannya yang berbeda dan algoritma BWT ini bersifat reversible (yang dapat dibalik). Beberapa algoritma dasar yang sering dikembangkan diantaranya adalah *BurrowsWheeler Transform* (BWT) dan *Adaptive Huffman coding* (AHC).[2]

Metode iniumum digunakan untuk kompresi data.Ini berfungsi sebagai dasar untuk beberapa program popular yang digunakan pada komputer pribadi.Adapun Adaptive Huffman Coding disebut juga Dynamic Huffman Coding adalah teknik pengkodean adaptif berdasarkan pengkodean Huffman.Ini memungkinkan pembuatan kode karena simbol

simbol ditrasmisikan tidak memiliki pengetahuan awal tentang distribusi sumber yang memungkinkan pengkodean one-pass dan adaptasi terhadap perubahan kondisi data. Manfaat prosedur one-pass adalah bahwa sumbernya dapat dikodekan secara real time, meskipun menjadi lebih sensitif terhadap kesalahan transmisi, karena hanya satu kerugian yang menghancurkan keseluruhan kode.[3]

2. METODOLOGI PENELITIAN

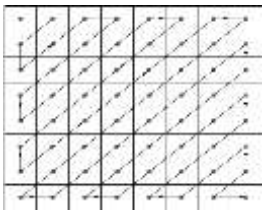
2.1 Kompresi

Kompresi citra digital merupakan upaya untuk melakukan transformasi terhadap data atau simbol, tanpa menimbulkan perubahan yang signifikan atascitra digital tersebut bagi mata manusia yang menikmatinya. Kompresi harusdilakukan secara efektif sehingga citra digital yang dihasilkan setelah proses kompresi mempunyai ukuran lebih kecil dibandingkan sebelum proses kompresi. Informasi tersebut dapat berupa data teks dan data citra, dimana data teks merupakan kumpulan dari karakter-karakter atau string yang menjadi suatu kesatuan. Citra atau gambar sangat membantu dalam aktifitas dan kecepatan pengiriman informasi dalam bentuk citra akan menjadi bagian utama dalam pertukaran informasi saat ini dan masa yang akan datang.[5]

2.2 Algoritma Burrows Wheeler Transform

Metode Burrows-Wheeler Transform (BWT) adalah algoritma kompresi data, yang pertama kali diperkenalkan pada tahun 1994 oleh Michael Burrows danDavid Wheeler. Gagasan utamanya adalah memperoleh rasio kompresi data yang lebih baik untuk menghemat ruang penyimpanan dan memberikan transmisi data yang lebih cepat melalui jaringan-jaringan yang berbeda. Kompresi berdasarkan BWT adalah salah satu algoritma terkemuka yang mendekati paling baik untuk data teks pada saat sekarang, namun BWT juga dapat digunakan untuk meningkatkan kinerja kompresi untuk citra.[7]

Sebelum melakukan proses kompresi dengan metode *Burrows-WheelerTransform* (BWT), terdapat proses yang harus dilewati terlebih dahulu. Proses tersebut adalah data citra yang akan melalui proses kompresi harus diubah dari bentuk piksel dua dimensi ke bentuk piksel satu dimensi berurutan dengan cara pemindain zig-zag



Gambar 1. Proses Zig-Zag

Setelah bentuk dari data citra diubah lalu dilansjutkan dengan metode BWT. Misalkan sebuah vektor 1 dimensi berurutan *p* telah didapat seperti berikut:

$$P = [\ 3\ 2\ 5\ 3\ 1\ 4\ 2\ 6\ ]$$

Vektor *p* disalin ke baris pertama, atau disebut sebagai indeks 0. Urutan selanjutnya adalah mengurutkan dengan cara merubah susunan perputaran ke kiriuntuk setiap baris selanjutnya. Tahap pertama dari BWT ditampilkan melalui Gambar 2.

index	Step 1							
0	3	2	5	3	1	4	2	6
1	2	5	3	1	4	2	6	3
2	5	3	1	4	2	6	3	2
3	3	1	4	2	6	3	2	5
4	1	4	2	6	3	2	5	3
5	4	2	6	3	2	5	3	1
6	2	6	3	2	5	3	1	4
7	6	3	2	5	3	1	4	2

Gambar 2. Tahap Pertama *Forward Transform* dari BWT

Tahap selanjutnya adalah setiap baris disusun secara leksikografi (berdasarkan kamus). Tahap terakhir dari proses BWT adalah output yang terdiri dari indeks terakhir pada langkah sebelumnya. Berikut tahap kedua dan ketiga dari BWT ditampilkan melalui Gambar 2.

index	Step 2							
0	1	4	2	6	3	2	5	3
1	2	5	3	1	4	2	6	3
2	2	6	3	2	5	3	1	4
3	3	1	4	2	6	3	2	5
4	3	2	5	3	1	4	2	6
5	4	2	6	3	2	5	2	1
6	5	3	1	4	2	6	3	2
7	6	3	2	5	3	1	4	2

index	Step3
0	3
1	3
2	4
3	5
4	6
5	1
6	2
7	2

Gambar 3. Tahap Kedua dan Ketiga *Forward Transform* dari BWT

Vektor *p* asli muncul di baris kelima, dan output dari BWT berada pada kolom terakhir, dinyatakan dengan:

L = [3 3 4 5 6 1 2 2]

dengan indeks = 4. Hasil dapat ditulis sebagai BWT = [*index,L*] dimana *L* adalah output dari *Burrows-Wheeler Transform* dan *index* menggambarkan lokasi asli dari urutan leksikografi.

2.3 Algoritma Adaptive Huffman Coding

Adaptive Huffman coding pertama kali diperkenalkan oleh Faller dan Gallager (Faller 1973, Gallaber 1978).Knuth memberikan kontribusi dengan peningkatan pada algoritmanya (Knuth 1985) dan menghasilkan algoritma yang dikenal dengan algoritma FGK.Versi terbaru dari *Adaptive Huffman Coding* diperkenalkan oleh Vitter (Vitter 1987).Semua metode yang ditemukan merupakan skema *define-word* menentukan *mapping* dari pesan sumber menjadi *codeword* di dasari pada perkiraan probabilitas pesan sumber.Kode bersifat adaptif, berganti sesuai dengan perkiraan optimalnya pada saat itu.Dalam hal ini, *Adaptive Huffman Code* merespon lokalitas.Dalam pengertian, *encoder*mempelajari karakteristik dari sumber.*Decoder* harus mempelajari kesamaan dengan *encoder* dengan memperbaharui pohon Huffman sehingga sinkron dengan *encode.*

Keuntungan lain dari sistem ini adalah kebutuhan transmisi data, data akan lewat hanya sekali (tanpa *statistic table*). Tentu saja, metode *one-pass* tidak akan menarik apabila jumlah bit yang ditransmisikan lebih besar dari metode *two-pass*. Namun, performa dari metode ini, dalam ruang lingkup jumlah bit yang ditransmisikan, dapat lebih baik daripada *static huffman coding*. Permasalahan ini tidak kontradiktif dengan optimalisasi pada metode statis, karena metode ini optimal berdasarkan *time-variant*.Kinerja dari metode adaptif dapat lebih buruk daripada metode statik.Metode *adaptive Faller*, Gallager dan Knuth merupakan dasar dari UNIX *utility compact*.Kinerja *compact* ini termasuk bagus, karenafaktor kompresinya mencapai 30-40%.[8]

Ide utama dari kompresi dan dekomperi dimulai dengan pohon huffman yang kosong dan memodifikasi simbol yang sedang dibaca dan diproses.*compressor* dan *decomprossor* memodifikasi pohon dengan cara yang sama, sehingga pada setiap titik dalam proses akan menggunakan kode yangsama. meskipun kode-kode dapat berubah dalam setiap langkah. dimana*decoder* mencerminkan operasi *encoder*.

Awalnya kompresi dimulai dengan pohon huffman yang kosong.Tidak ada simbol-simbol yang telah ada sebelumnya.simbol pertama dimasukkkan pada aliran dalam bentuk terkompresi. simbol ini kemudian ditambahkan ke dalam pohon, jika simbol ini ditemukan lagi didalam aliran (pohon) maka frekuensi bertambah satu, dan akan memodifikasi pohon, pohon diperiksa lagi untuk melihat apakah itu masih pohon huffman (kode terbaik). jika tidak susun kembali pohon, dan mengakibatkan perubahan kode.

Pada Gambar 2.4 menunjukkan apa pohon biner yang akan menjadi pohon Huffman. Pohon di dalam gambar berisi lima simbol: A, B, C, D, dan E. Hal ini ditunjukkan dengan adanya simbol dan frekuensin (dalam tanda kurung), setelah 16 simbol telah dimasukkan dan diproses. Properti yang akan menjadikannyapohon Huffman adalah jika kita memindai tingkat demi tingkatnya, memindai setiap tingkat dari kiri ke kanan, dan pergi dari bawah (daun) ke atas (akar), frekuensi akan diurutkan atau tidak turun. Dengan demikian, simpul kiri bawah

(A) memiliki frekuensi terendah, dan simpul kanan atas (akar) memiliki frekuensi tertinggi. Ini disebut milik *sibling property*.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Masalah

Citra merupakan data yang berbentuk gambar biasanya citra memiliki ukuran yang besar, semakin bagus atau baik citra yang dihasilkan maka semakin besar pula kapasitas citra yang dimiliki.Di zaman yang serba teknologi seperti ini banyak orang menggunakan gambar sebagai sarana informasi karena informasi dengan menggunakan gambar jauh lebih efektif, mudah dimengerti dan terkesan jauh lebih menarik. Berdasarkan analisa permasalahan tesebut penulis akan melakukan suatu teknik atau cara bagaimana menghasilkan citra dengan kualitas yang bagus dan kapasitas yang kecil tanpa mengurangi kualitas dari citra tersebut.

Teknik yang dilakukan yaitu dengan mengkombinasikan dua algoritma pada sebuah citra. Algoritma yang pertama yaitu algoritma *Burrows WheelerTransform* (BWT) yaitu algoritma yang poses *Encoding* menghasilkan sebuah barisan *L* dan sebuah indeks s. Alogitma yang kedua yaitu *Adaptive Huffman Coding* (AHC) yaitu algorritma yang prinsip kerjanya menggunakan kode dimana tiap karakter *(symbol)* di kodekan dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek, dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang.

3.1.1. Penerapan Algoritma Burrows Wheeler Transform

Citra akan dikompresi sesuai dengan langkah-langkah yang telah ditetapkan pada algoitma *Burrows Wheeler Transform*. Dari hasil yang didapat pada algoitmaini akan dilakukan kombinasi terhadap algoritma *Adaptive Huffman Coding.* Adapun langkah-langkah yang dilakukan dalam mengkompesi cita dengan menggunakan algoitma *Burrows Wheeler Transform* yaitu sebagai beikut :

1. Siapkan citra untuk diubah telebih dahulu dari bentuk dua dimensi menjadibentuk satu dimensi.

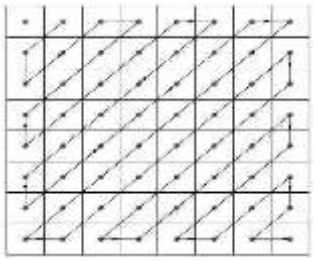
2. Lakukan pembentukkan matriks dimana nilai baris matriks sama dengan panjang data citra satu dimensi.
3. Pada baris pertama matriks lakukan pengisian data citra asli untuk baris berikutnya diisi dengan data citra sebelumnya dengan cara melakukanperputaran data citra dari kiri.
4. Semua baris telah terisi langkah selanjutnya urutkan setiap baris padamatriks secara *leksikografi* (berdasarkan kamus).
5. Hasil dari kolom terakhir sebgai *output* dari BWT lakukan penyimpanan baris nilai data citra awal dengan simbol I *(indeks)* guna untuk melakukan proses *invers* jika ingin kembali ke bentuk semula. Hitung rasio dan waktu kompesi pada algoitma *output* BWT.

Ada proses yang harus dilakukan terlebih dahulu yaitu matiks yang sebulumnya bebentuk piksel dua dimensi diubah terlebih dahulu kedalam piksel bebentuk satu dimensi beurutan dengan cara pemindaan zig-zag. Setelah bentuk dari data citra diubah lalu dilanjutkan dengan metode BWT. Misalkan sebuah citra dengan matriks seperti berikut:

Tabel 1. nilai awal

i\j	0	1	2
0	122	122	122
1	124	124	124
2	127	127	127

Hasil dari nilai kuantitasi yang telah didapat selanjutnya dilintaskan dalam bentukzig-zag, untuk lebih jelas dapat dilihat dibawah ini.



Gambar 4. Proses Zig-zag

Setelah dilakukan peubahan bentuk citra berdasarkan proses zig-zag dengan Algoitma BWT. Misalkan sebuah vektor 1 dimensi berurutan Q telah didapat seperti dibawah ini : Q = [122 122 124 127 124 122 127 127]

Vektor Q disalin pada baris petama, atau disebut sebagai indeks 0. Langkah selanjutnya adalah melakukan pengurutan dengan cara melakukan perubahan susunan dengan perputaran ke kiri untuk setiap baris selanjutnya. Adapun tahap pertama pada algoritma BWT dapat dilihat pada Tabel 2 di bawah ini:

Tabel 2. TahapPertama*Forward Transform*dari BWT

Index	Nilai Citra Tahap 1
0	122 122 124 127 124 122 124 127 127
1	122 124 127 124 122 124 127 127 122
2	124 127 124 122 124 127 127 122 122
3	127 124 122 124 127 127 122 122 124
4	124 122 124 127 127 122 122 124 127
5	122 124 127 127 122 122 124 127 124
6	124 127 127 122 122 124 127 124 122
7	127 127 122 122 124 127 124 122 124
8	127 122 122 124 127 124 122 124 127

Tahap selanjutnya yaitu melakukan penyusunan tiap baris *leksikografi*(berdasarkankamus). Adapun tahap terakhir dari langkah-langkah algoritma BWT yaitu hasil *output* yang terdiri dari index terakhir pada langkah yang dilakukan sebelumnya. Berikut adalah tahap kedua dan ketiga pada Tabel 3. di bawah :

Tabel 3. Tahapkedua dan ketiga *Forward Transform* dari BWT

Index	Nilai pixel Tahap 2	Nilai TahapPixel 3
0	122 122 124 127 124 122 124 127 **127**	127
1	122 124 127 124 122 124 127 127 **122**	122
2	122 124 127 127 122 122 124 127 **124**	124
3	124 127 124 122 124 127 127 122 **122**	122
4	124 122 124 127 127 122 122 124 **127**	127

ISSN : 2809-6096 (Media Online)
Vol 1, No 2, Februari 2022
Hal : 34-40

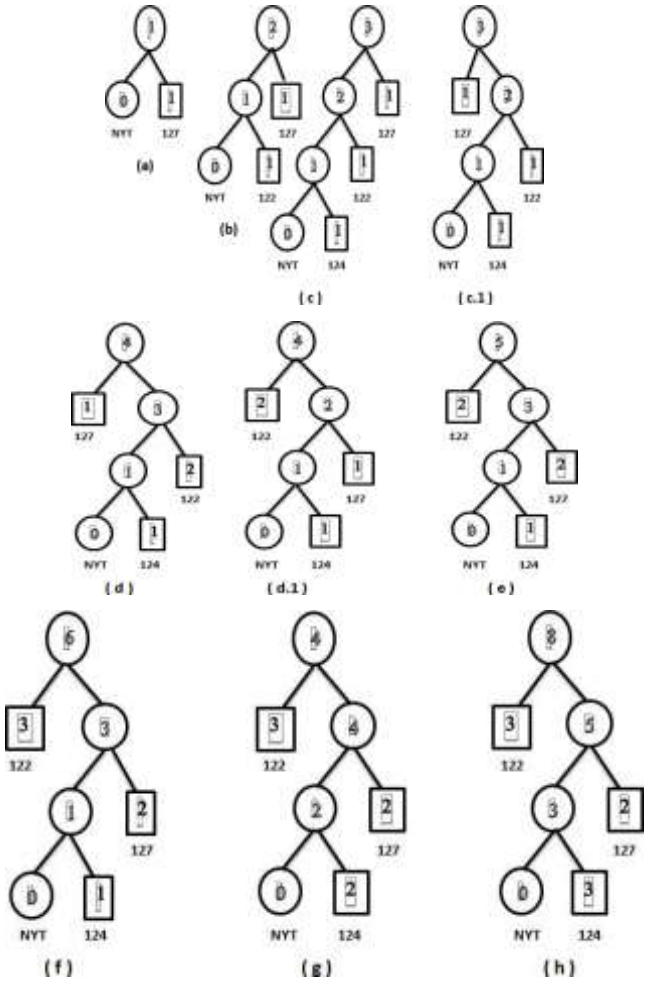
Index	Nilai pixel Tahap 2	Nilai TahapPixel 3
0	122 122 124 127 124 122 124 127 **127**	127
1	122 124 127 124 122 124 127 127 **122**	122
5	124 127 127 122 122 124 127 124 **122**	122
6	127 124 122 124 124 127 122 122 **124**	124
7	127 127 122 122 124 127 124 122 **124**	124
8	127 122 122 124 127 124 122 124 **127**	127

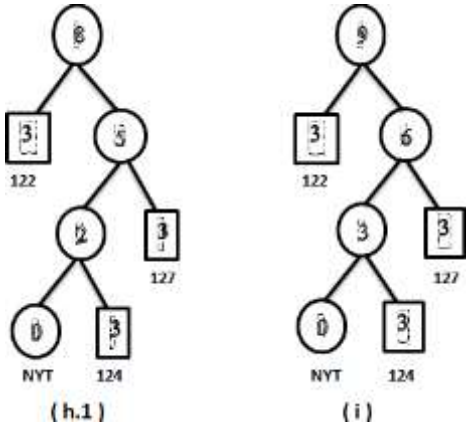
3.1.2. Penerapan Algoritma *Adaptive Huffman Coding* (AHC)

Setelah melakukan kompresi dengan algoritma *Burrows Wheeler Transform* langkah selanjutnya dalam mengkombinasi citra yaitu lakukan kompresi dari hasil*output* yang didapat pada algoritma BWT dengan menggunakan algoritma yang kedua yaitu algoritma *Adaptive Huffman Coding*.

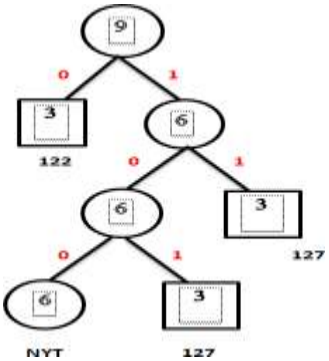
1. Kompresi Gambar dengan Algoritma *Adaptive Huffman Coding*

Langkah yang dilakukan pada algoritma AHC ini yaitu melakukan pembacaan *codingstring* yangdimasukkan kedalam pohon dengan ketentuan frekuensi yang memiliki jumlah tertinggi berada diatas. Misalkan ada sebuahvektor Q = [127 122 124 122 127 122 124 124 127]. Adapun caranya dapatdilihat sebagai berikut.





Gambar 5. Kompresi *Adaptive Huffman Coding*



Gambar 6. Hasil *codingstring* pohon AHC

Tabel 4. Hasil Kombinasi kompresi algoritma BWT dan AHC

Nilai	Codeward	Bit	Frekuensi	Bit X Frekuensi
127	10	2	3	6
122	0	1	3	3
124	111	3	3	9
Total Bit				18 bit

4. KESIMPULAN

Dari pembahasan pada bab sebelumnya, analisa yang telah dilakukan oleh penulis maka kesimpulan yang diambil adalah sebagai berikut Prosedur yang dilakukan dalam mengompresi *file* citra dimulai dengan memilih *file* citra berformat JPG dan sudah diubah menjadi citra *Grayscale* yang akan dikompresi, dikompresi dengan kombinasi algoritma *burrows wheeler transform* dan *adaptive huffman coding* sehingga menghasilkan ukuran *file* yang lebih kecil dari ukuran *file* sebelum di kompresi. Penerapan kombinasi algoritma *burrows wheeler transform* dan *adaptive huffman coding* untuk mengkompresi *file* citra berformat JPG, dilakukan dengan membaca nilai *pixel* menggunakan aplikasi *Matlab*, lalu diubah menjadi nilai bit baru dan dicocokan dengan tabel kebenaran dari kombinasi algoritma *burrows wheeler transform* dan *adaptive huffman coding* dan diubah menjadi karakter baru.

REFERENCES

[1] R. Krasmala, A. Budimansyah, and U. T. Lenggana, "Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) dan Algoritma Huffman," *J. Online Inform.*, vol. 2, no. 1, p. 1, 2017, doi: 10.15575/join.v2i1.79.
[2] I. Lubis, P. Tarigan, and N. Sitompul, "Analisa Perbandingan Kompresi Citra Menggunakan Metode Discrete Cosine Transform (Dct) Dan Burrows Wheeler Transform (Bwt)," *Pelita Inform.*, vol. 16, pp. 285–287, 2017.
[3] D. Venkatasekhar and P. Aruna, "a Fast Fractal Image Compression Using Huffman Coding," *Asian J. Comput. Sci. Inf. Technol.*, vol. 2, no. 9, pp. 272–275, 2012.
[4] W. W. Kelen and D. Nugraheny, "Analisa Pemrosesan Paralel Untuk Kompresi Dan Dekompresi Data," *Compiler*, vol. 4, no. 1, pp. 65–74, 2015, doi: 10.28989/compiler.v4i1.89.
[5] I. R. Lubis, "Menggunakan Metode Eksponensial," vol. 16, pp. 382–384, 2017.
[6] Y. Darnita, K. Khairunnisyah, and H. Mubarak, "Kompresi Data TeksDengan Menggunakan Algoritma Sequitur," *Sistemasi*, vol. 8, no. 1, p. 104,2019, doi: 10.32520/stmsi.v8i1.429.
[7] Van, V. S. 2009. Image Compression Using Burrows-Wheeler Transform. Tesis. Helsinki University Of Technology.
[8] Marjiyono, "Penerapan Algoritma Ahc Algorithm Dalam Aplikasi," pp. 6– 8, 2015.
[9] T. Zebua and E. Ndruru, "Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 4,p. 275, 2017, doi: 10.25126/jtiik.201744474.

[10] A. Nugroho, *Rekayasa Perangkat Lunak Berorientasi Objek denganMetode USDP*. Yogyakarta: Penerbit ANDI, 2010.
[11] R. Hakim, *Visual Basic 2008 for Pemula Banget*. Jakarta: Elex MediaKomputindo, 2009.
[12] Ihsan and D. P. Utomo, "Analisis Perbandingan Algoritma Even-Rodeh Code Dan Algoritma Subexponential Code Untuk Kompresi File Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
[13] S. R. Saragih and D. P. Utomo, "Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.
[14] Lamsah and D. P. Utomo, "Penerapan Algoritma Stout Codes Untuk Kompresi Record Pada Databade Di Aplikasi Kumpulan Novel," KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer), vol. 4, no. 1, 2020.