



# **Analisis Dan Implementasi Kompresi File Citra Menggunakan Algoritma Burrows Wheeler Transform**

**Asrul Soleh Harahap**

Fakultas Ilmu Komputer Dan Teknologi Informasi, Tenik Informatika, Universitas Budi Darma, Medan, Indonesia

Email: [asrul09hrp@gmail.com](mailto:asrul09hrp@gmail.com)

**Abstrak**—Perkembangan teknologi saat ini juga mempengaruhi besarnya ukuran data dari suatu file, terutama file dalam bentuk citra. Citra atau gambar secara digital memiliki ukuran yang cukup besar sehingga dapat menggunakan ruang memori yang cukup besar. Ukuran data yang besar akan membutuhkan ruang penyimpanan yang besar pula. Maka dari itu diperlukan sebuah solusi seperti membuat sebuah media penyimpanan berkapasitas besar. Akan tetapi solusi tersebut kurang efisien karena kedepannya ukuran file gambar akan semakin besar pula dengan kualitas gambar yang lebih baik. Maka dari itu diperlukan solusi tambahan untuk menghemat pemakaian ruang media penyimpanan, seperti mengurangi ukuran dari data yang akan disimpan. Untuk mengurangi penggunaan memori dan mengurangi ukuran file tanpa mengurangi informasi yang ada di dalam file tersebut dapat diatasi dengan metode kompresi. Kompresi citra adalah suatu aplikasi kompresi data yang dilakukan terhadap citra digital yang bertujuan untuk memperkecil suatu bit yang ada pada file citra tersebut sehingga dapat disimpan atau ditransmisikan secara cepat dan efisien. Metode Burrows Wheeler Transform ini sangat bekerja dengan baik pada gambar dan teks, dan dapat mencapai rasio kompresi yang sangat tinggi.

**Kata Kunci:** Kompresi, Citra, Burrows Wheeler Transform.

**Abstract**— Current technological developments also affect the size of the data from a file, especially files in the form of images. Digital images or images have a size large enough so that they can use a large enough memory space. Large data sizes will require large storage space as well. Therefore we need a solution such as making a large capacity storage media. However, this solution is less efficient because in the future the image file size will be even bigger with better image quality. Therefore, additional solutions are needed to save the use of storage media space, such as reducing the size of the data to be stored. To reduce memory usage and reduce file size without reducing the information contained in the file can be overcome by the compression method. Image compression is a data compression application that is carried out on digital images which aims to reduce a bit in the image file so that it can be stored or transmitted quickly and efficiently. The Burrows Wheeler Transform method works very well with images and text, and can achieve very high compression ratios.

**Keywords:** Compression, Image, Burrows Wheeler Transform

## **1. PENDAHULUAN**

Perkembangan teknologi saat ini juga mempengaruhi besarnya ukuran data dari suatu file, terutama file dalam bentuk citra. Citra atau gambar secara digital memiliki ukuran yang cukup besar sehingga dapat menggunakan ruang memori yang cukup besar. Ukuran data yang besar akan membutuhkan ruang penyimpanan yang besar pula. Perkembangan teknologi dalam media penyimpanan kedepannya akan semakin besar, akan tetapi kedepannya citra atau gambar juga akan semakin besar ukurannya dengan kualitas gambar yang semakin bagus[1]. Maka dari itu diperlukan sebuah solusi untuk menghemat pemakaian ruang media penyimpanan, seperti mengurangi ukuran dari data yang akan disimpan. Proses pengurangan ukuran data disebut juga dengan teknik pemampatan atau kompresi.

Kompresi citra adalah suatu aplikasi kompresi data yang dilakukan terhadap citra digital yang bertujuan untuk memperkecil suatu bit yang ada pada file citra tersebut sehingga dapat disimpan atau ditransmisikan secara cepat dan efisien. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital itu dengan cara kombinasi dari tiga warna dasar yaitu merah, hijau, dan biru[2][3]. Citra di ubah ke bentuk digital agar bisa disimpan dalam media penyimpanan. Tapi jika insentitas warna berkurang bisa saja ada informasi yang hilang dari citra asli. Hasil penelitian sebelumnya yang dilakukan oleh Irma Rapmayani Lubis pada tahun 2017 dengan judul “Analisa Perbandingan Algoritma Huffman dengan Algoritma Burrows Wheeler Transform pada kompresi citra menggunakan metode Eksponensial” menyimpulkan bahwa algoritma Burrows Wheeler Transform menghasilkan peforma waktu yang kurang baik tetapi algoritma ini memiliki kinerja kompresi yang tinggi pada format citra jpeg yang terlihat pada rasionya[4].

## **2. METODOLOGI PENELITIAN**

### **2.1 Kompresi**

Kompresi berarti memampatkan/mengecilkan ukuran. Sedangkan kompresi data adalah proses mengkodekan informasi menggunakan bit atau *information-bearing unit* yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem encoding tertentu[5].

### **2.2 Dekompresi**

Sebuah data yang sudah dikompres tentunya harus dapat dikembalikan lagi ke bentuk aslinya, prinsip ini dinamakan dekompresi. Untuk dapat merubah data yang terkompres diperlukan cara yang berbeda seperti pada waktu proses kompres dilaksanakan. Jadi pada saat dekompres terdapat catatan *header* yang berupa *byte-byte* yang berisi catatan mengenai isi dari *file* tersebut[6].



**2.3 Citra**

Citra merupakan salah satu media yang paling populer digunakan dalam pertukaran informasi pada saat sekarang. Peningkatan kebutuhan terhadap penggunaan citra harus pula didukung oleh media penyimpanan yang tersedia. Namun biasanya citra yang dihasilkan dari setiap perangkat berkamera memiliki ukuran yang relatif besar. Citra yang memiliki ukuran besar tentunya akan membutuhkan media penyimpanan yang besar pula. Begitu juga dengan keperluan pengiriman, semakin besar citra maka waktu yang dibutuhkan untuk pengiriman juga semakin lama. Sementara keterbatasan media penyimpanan dan kapasitas bandwidth menjadi masalah bagi setiap pengguna[3].

**2.5 Algoritma Burrows-Wheeler Transform (BWT)**

Pada penelitian ini, algoritma yang digunakan dalam membandingkan kompresi citra adalah *Burrows-Wheeler Transform (BWT)*. *Burrows Wheeler Transform (BWT)* merupakan algoritma proses yang melakukan transformasi terhadap blok data teks menjadi suatu bentuk baru yang tetap dan mengandung karakter sama hanya saja urutannya yang berbeda dan algoritma BWT ini bersifat *reversible* (yang dapat dibalik)[7]. Sebelum melakukan proses kompresi dengan metode *Burrows-Wheeler Transform (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 pemindaian zig-zag. Setelah bentuk dari data citra diubah lalu dilanjutkan 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 kiri untuk setiap baris selanjutnya. Tahap pertama dari BWT ditampilkan melalui Gambar 1.

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 1.** 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.

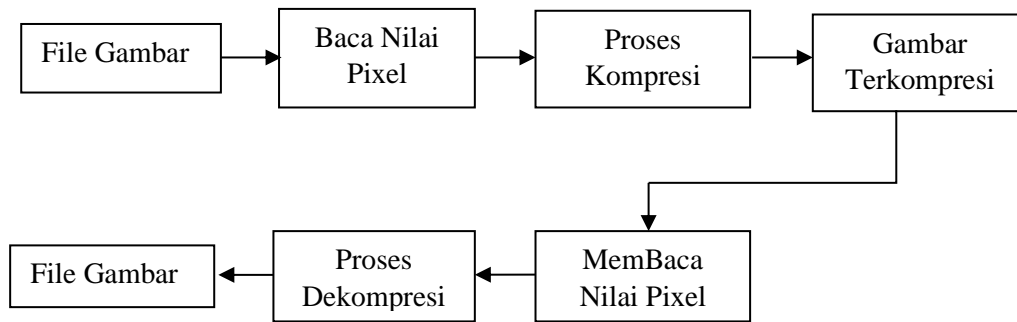
index	Step 2	index	Step3
0	1 4 2 6 3 2 5 3	0	3
1	2 5 3 1 4 2 6 3	1	3
2	2 6 3 2 5 3 1 4	2	4
3	3 1 4 2 6 3 2 5	3	5
4	3 2 5 3 1 4 2 6	4	6
5	4 2 6 3 2 5 2 1	5	1
6	5 3 1 4 2 6 3 2	6	2
7	6 3 2 5 3 1 4 2	7	2

**Gambar 2.** Tahap kedua dan ketiga *Forward Transform* dari BWT

Vektor p asli muncul di baris kelima pada Gambar 2.2, 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.

**3. HASIL DAN PEMBAHASAN**

Pada penelitian ini, bertujuan untuk menganalisa dari cara kerja dan efisiensi pada perancangan perangkat lunak kompresi file citra dengan menggunakan algoritma *Burrows wheeler transform*. Tahap analisa ini terhadap suatu sistem dilakukan sebelum tahapan perancangan yang dilakukan. Adapun tujuan dilakukan analisa terhadap suatu sistem adalah untuk mengetahui alasan mengapa sistem tersebut diperlukan, yaitu dengan merumuskan kebutuhan-kebutuhan dari sistem tersebut untuk meminimalisir sumber daya yang berlebih serta membantu merencanakan penjadwalan pembentukan. Berikut merupakan prosedur kompresi dan dekompresi file gambar.



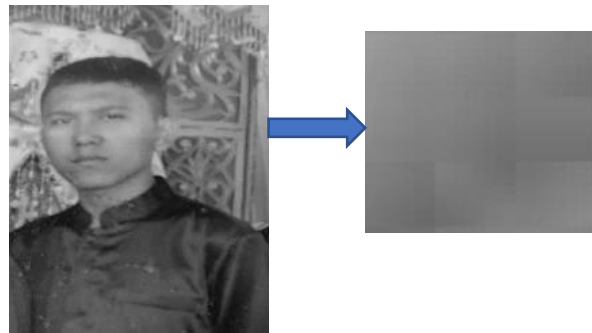
**Gambar 3.** Prosedur Kompresi Dekompresi *File Gambar*

### 3.1 Kompresi Dengan *Burrows Wheeler Transform*

Dalam melakukan kompresi *file* citra sebelumnya harus dilakukan analisa terhadap *file* citra yang akan dikompresi. Dalam menganalisa *file* citra harus dilakukan mengambil *sample file* citra untuk mendapatkan nilai dari *pixel*-nya. Berikut adalah langkah untuk mengkompresi dan mendekompresi *file* citra.

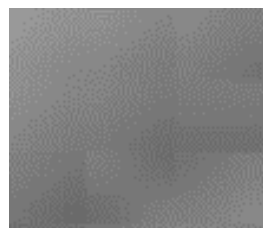
- a. Melakukan analisa proses kompresi *file* citra menggunakan algoritma *Burrows Wheeler Transform*

Dalam melakukan kompresi pada *file* citra adalah terlebih dahulu harus memasukan *file* citra. *File* citra yang digunakan untuk dikompresi dalam penelitian ini adalah berukuran 784 x 1280 *pixel*.



**Gambar 4.** grayscale asli

Berdasarkan gambar diatas maka dapat dianalisa bahwa dalam proses kompresi *file* gambar dapat diambil *sample* gambar dengan ukuran 3 x 3 *pixel*, dengan tujuan untuk mengetahui cara kerja kompresi menggunakan algoritma *Burrows Wheeler Transform* dalam kompresi *file* gambar, berikut ini adalah *sample* dari gambar yang akan dikompresi:



**Gambar 5.** *sample* gambar 3 x 3 yang akan dikompresi

- b. Langkah selanjutnya adalah dengan melakukan pembacaan isi *file* gambar. Setiap Isi *file* gambar terdapat nilai *pixel*. Nilai *pixel* tersebut diproses dalam bentuk 3 x 3 sehingga menghasilkan 9 *pixel* sebagai *sample* gambar yang akan dianalisa. Adapun nilai 9 *pixel* gambar tersebut dapat dilihat pada gambar di bawah ini. Berikut ini adalah hasil pengambilan *sample* gambar yang akan dikompresi berdasarkan dari nilai 9 *pixel* akan ditunjukkan pada Tabel 3.1 berikut ini :

**Tabel 1.** Nilai 9 titik *pixel*

166	120	141
135	113	113
63	71	45

Berdasarkan pada Tabel di atas, terdapat nilai *pixel file* gambar *sample*. Untuk keperluan hitungan manual hanya diambil *sample* nilai sebanyak 25 karakter dari nilai *pixel file* gambar *sample*. Adapun nilai *pixel file* gambar *sample*

tersebut adalah 166, 120, 141, 135, 113, 113, 163, 71, 45. Selanjutnya adalah akan dilakukan pengelompokan nilai *pixel* berdasarkan nilai *frekuensi*. Urutan karakter nilai *pixel* dapat dilihat pada tabel berikut ini:

**Tabel 2.** Pembacaan Nilai *Pixel* dan *frekuensi*

Pixel	Frekuensi
113	2
45	1
71	1
120	1
135	1
141	1
63	1
166	1
Total Pixel	9

Berdasarkan pada tabel di atas, didapatkan beberapa nilai *pixel* yang sama. Sebelum proses kompresi citra, langkah awal adalah membaca nilai *pixel* citra kemudian membuat tabel nilai *pixel* yang diurutkan dari nilai frekuensi terbesar (nilai *pixel* yang sama) ke terkecil. Urutan nilai *pixel* dapat dilihat pada tabel di bawah ini :

**Tabel 3.** Nilai *pixel* awal

pixel	Nilai	Bit	Frek	Bit x Frek
	Biner			
113	01110001	8	2	16
45	00101101	8	1	8
71	01000111	8	1	8
120	01111000	8	1	8
135	10000111	8	1	8
141	10001101	8	1	8
63	00111111	8	1	8
166	10100011	8	1	8
Jumlah Bit				72

Berdasarkan tabel di atas, satu nilai *pixel desimal* (karakter) bernilai delapan *bit* bilangan *biner*. Sehingga 9 titik *pixel* mempunyai nilai *biner* sebanyak 72 *bit*. 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 ke dua dan ke tiga dari BWT ditampilkan melalui table 3.4.

**Tabel 4.** Tahap ke dua dan ke tiga *Forward Transform* dari BWT

Index	Step II	Step III
0	45 166 120 135 63 113 141 113 <b>71</b>	71
1	63 113 141 113 71 45 166 120 <b>135</b>	135
2	71 45 166 120 135 63 113 141 <b>113</b>	113
3	113 71 45 166 120 135 63 113 <b>141</b>	141
4	113 141 113 71 45 166 120 135 <b>63</b>	63
5	120 135 63 113 141 113 71 45 <b>166</b>	166
6	135 63 113 141 113 71 45 166 <b>120</b>	120
7	141 113 71 45 166 120 135 63 <b>113</b>	113
8	166 120 135 63 113 141 113 71 <b>45</b>	45

Vektor *p* asli muncul di baris ke-9 pada table 3.4, dan output dari BWT berada pada kolom terakhir, dinyatakan dengan:

$p = [ 71 \ 113 \ 63 \ 141 \ 166 \ 120 \ 113 \ 135 \ 45 ]^T$  dengan indeks = 8. Hasil dapat ditulis sebagai  $BWT = [Index, L]$ , dimana *L* adalah output dari *Burrows-Wheeler Transform* dan *index* menggambarkan lokasi asli dari urutan leksikografi. Untuk mendapatkan rasio kompresi metode Transformasi *Burrows Wheller* memanfaatkan metode *Huffman*.

1. Data matriks tersebut di ubah menjadi vektor, sehingga didapat vektor = [1 2 6 6 8 6 4 7 7].
2. Tentukan nilai ferkuensi kemunculan. Hasilnya adalah:

**Tabel 5.** Tabel Frekuensi

Nilai Output/Keabuan	Frekuensi
1	1
2	1

4	1
6	3
7	2
8	1

3. Urutkan frekuensi dari yang terkecil ke frekuensi yang terbesar. Data setelah diurutkan : [ 1 2 4 8 7 6].  
 Proses penghitungan hasil kompresi dengan cara mengalikan panjang kode dengan jumlah frekuensi pada setiap data, kemudian hasil perkalian dijumlahkan seperti tabel di bawah.

**Tabel 6.** Perhitungan Kompresi BWT

Nilai Output/keabuhan	Frekuensi	Kode huffman	Bit	Frekuensi x bit	Jumlah
6	3	11	2	3 x 2	6
7	2	10	2	2 x 2	4
1	1	010	3	1 x 3	3
2	1	011	3	1 x 3	3
4	1	000	3	1 x 3	3
8	1	001	3	1 x 3	3
Total					22

Rasio kompresi adalah =  $100\% - (\text{ukuran citra hasil kompresi} / \text{ukuran citra semula}) \times 100\%$

$$\text{Rasio} = 100\% - (22/72) \times 100\%$$

$$\text{Rasio} = 100\% - (0.305) \times 100\%$$

$$\text{Rasio} = 100\% - 30.5\%$$

$$\text{Rasio} = 100\% - 30.5\%$$

$$\text{Rasio} = 69.5\%$$

Bila dilihat perbandingan sebelum dikompresi, file citra terdiri dari 72 bit. Tapi setelah dilakukan kompresi menjadi 22 bit sehingga didapat besar rasio sebesar 69.5 % yang artinya 69.5 % dari citra semula berhasil dimampatkan. Ukuran citra yang berkurang adalah  $72 - 22 \text{ bit} = 50 \text{ bit}$ . Berikut gambar hasil kompresi sebagai berikut :



**Gambar 6.** Hasil Kompresi

### 3.2 Dekompresi dengan Burrows Wheeler Transform

Berdasarkan proses dekompresi hal dilakukan adalah menganalisa keseluruhan nilai output/pixel dari kompresi sebelumnya. Dalam algoritma *Burrows Wheeler Transform*, *reverse transform* yang dapat dibalik sehingga vektor asli dapat dikembalikan dari output yang dihasilkan oleh *burrows wheeler transform*. Dibentuk *reverse transform* hanya output *L* dan index yang dibutuhkan untuk mengembalikan vektor urutan aslinya. Tahap *reverse transform* seperti tabel berikut

**Tabel 7.** Tahap Reverse Transform

Index	(i = 1)			(i = 2)		
	a	B	C	A	B	C
0	71	45	45...71	7145	45166	45166...71
1	135	63	63...135	13563	63113	63113...135
2	113	71	71...113	11371	7145	7145...113
3	141	113	113...141	141113	11371	11371...141
4	63	113	113...63	63113	113141	113141...63
5	166	120	120...166	166120	120135	120135...166
6	120	135	135...120	120135	13563	13563...120
7	113	141	141...113	113141	141113	141113...113

8	45	166	166...45	45166	166120	166120...45
---	----	-----	----------	-------	--------	-------------

**Tabel 8.** Tahap Akhir *Reverse Transform*

Index	(i = 8)		
	A	B	c
0	714516612013563113141	4516612013563113141113	4516612013563113141113..71
1	135631131411137145166	631131411137145166120	631131411137145166120..135
2	113714516612013563113	714516612013563113141	714516612013563113141..113
3	141113714516612013563	113714516612013563113	113714516612013563113..141
4	631131411137145166120	1131411137145166120135	1131411137145166120135..63
5	1661201356311314111371	120135631131411137145.	120135631131411137145...166
6	120135631131411137145.	135631131411137145166	135631131411137145166..120
7	1131411137145166120135	141113714516612013563	141113714516612013563..113
8	4516612013563113141113	1661201356311314111371	1661201356311314111371...45

Output dari *reverse transform* juga berada di index 8, itulah mengapa index dimanfaatkan untuk mempresentasikan output dari forward transform dan urutannya adalah [166 120 135 63 113 141 113 71 45], yang mana memiliki urutan asli dari vektor *p*. Setelah proses dekompresi dengan algoritma *Burrows Wheeler Transform (BWT)* selesai, nilai pixel kembali kesemula dengan cara pemindahan *zig-zag* sehingga citra hasil dekompresi dapat ditampilkan lagi sebagai berikut :



**Gambar 7.** Zig Zag

**Tabel 9.** Nilai awal 9 titik pixel

166	120	141
135	113	113
63	71	45

Adapun gambar hasil dekompresi sebagai berikut:



**Gambar 8.** Hasil Dekompresi

#### 4. KESIMPULAN

Dengan diselesaikannya penelitian ini tentang analisis dan implementasi kompresi file citra menggunakan algoritma burrows wheeler transform yaitu Algoritma Burrows Wheeler Transform tidak bisa mengkompresi, tetapi untuk memproses data masukan secara antrian, dan memproses langsung satu blok data teks sebagai satuan dan dapat menemukan karakter yang sama secara berurutan. Hasil kompresi file citra sebelumnya terdiri dari 72 bit. Tapi setelah dilakukan kompresi menjadi 22 bit sehingga didapat besar rasio sebesar 69.5 %

#### REFERENCES

- [1] I. R. Lubis, "MENGUNAKAN METODE EKSPONENSIAL," vol. 16, pp. 382–384, 2017.
- [2] Suharso and A. Retnoningsih, kamus besar bahasa indonesia. 2005.



- [3] I. Lubis, P. Tarigan, and N. Sitompul, "ANALISA PERBANDINGAN KOMPRESI CITRA MENGGUNAKAN METODE DISCRETE COSINE TRANSFORM ( DCT ) DAN BURROWS WHEELER TRANSFORM ( BWT )," vol. 16, pp. 285–287, 2017.
- [4] I. F. Rahmad and H. Kurniawan, "Kompresi file citra bitmap menggunakan algoritma rle dan lz78," *Csrid*, vol. 3, no. Juni 2011, pp. 81–92, 2011.
- [5] A. F. Siregar, I. Pendahuluan, and A. K. Data, "PERANCANGAN APLIKASI KOMPRESI FILE CITRA USG MENGGUNAKAN ALGORITMA LZ78," vol. 17, no. April, pp. 164–167, 2018.
- [6] C. T. Utari, P. Studi, M. Teknik, U. S. Utara, and K. Citra, "IMPLEMENTASI ALGORITMA RUN LENGTH ENCODING UNTUK PERANCANGAN APLIKASI KOMPRESI DAN DEKOMPRESI," vol. V, no. 2, pp. 24–31, 2016.
- [7] A. N. Latief, L. Haryanto, A. Lawi, A. N. Latief, L. Haryanto, and A. Lawi, "Analisis Transformasi Burrows Wheeler untuk Kompresi Data Analysis of Burrows-Wheeler Transform for Data Compression."
- [8] V. S. Van, *Image Compression Using Burrows Wheeler Transform*. 2009..