

Pemampatan Citra Keabuan dengan Algoritma *Adaptive Huffman* dan *LZW*

Djuned Fernando Djusdek, Tohari Ahmad, dan Hudan Studiawan
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: tohari@if.its.ac.id, hudan@if.its.ac.id

Abstrak—Perkembangan data digital cukuplah pesat, mulai dari ukuran hingga jenis berkas. Untuk mengimbangi hal tersebut maka perlu adanya algoritma pemampatan guna mengurangi ukuran penggunaan media penyimpanan. Salah satu algoritma yang cukup lama ada adalah *Adaptive Huffman*, dan serta mengkombinasikannya dengan *LZW*, maka hasil pemampatan yang didapat akan lebih hemat. Pada paper ini penulis mencoba membuktikan hal tersebut dan juga dengan menambahkan sebuah algoritma pengolahan awal untuk berkas citra guna bersifat *lossy*. Algoritma pengolahan awal yang akan penulis jabarkan menggunakan metode *mean*, median dan modus untuk menghitung dan menentukan nilai *LSB* baru secara adaptif. Dari hasil percobaan didapatkan rerata nilai *PSNR* sebesar 36,26 dB dan rerata rasio pemampatan sebesar 44,77 % pada metode median.

Kata Kunci—*Adaptive Huffman*, *LZW*, *LSB*, pemampatan citra, *lossy*.

I. PENDAHULUAN

DEWASA ini perkembangan berkas digital semakin pesat, mulai dari format hingga ukuran berkas yang sangat besar. Semakin besarnya ukuran berkas semakin besar pula daya tampung media penyimpanan yang dibutuhkan. Namun media penyimpanan juga memiliki keterbatasan daya tampung, sehingga tidak semua berkas dapat disimpan dalam sebuah media penyimpanan saja.

Solusinya adalah dengan melakukan pemampatan berkas, sehingga berkas tetap dapat disimpan dan memakan ruang penyimpanan secukupnya. Pemampatan berkas sendiri sudah cukup lama berkembang. Diawali dengan adanya *Huffman Coding* hingga munculnya teknik-teknik pemampatan semisal teknik *dictionary* yang merupakan basis dari Algoritma *LZ* dan turunannya.

Berkas sendiri bermacam-macam adanya, ada yang berupa berkas citra yang didalamnya menyimpan informasi warna, piksel dan intensitas. Saat ini sudah banyak algoritma pemampatan citra yang umum digunakan, seperti *JPEG*, *PNG*, *GIF*, dan lain sebagainya. Namun dari semua pemampatan citra tersebut hanya sedikit yang menggunakan teknik *dictionary*. Contohnya *PNG* dan *GIF* [1] serta beberapa riset lainnya yang detailnya dapat dilihat pada artikel [2] dan [3]. Teknik *dictionary* sendiri memiliki keunggulan dalam ukuran rasio berkas serta mudah dalam penggunaannya.

Artikel ini menggunakan artikel [4] sebagai acuan, dan mengimplementasikan algoritma pemampatan citra secara *lossy* dan berbasis *dictionary* dengan menggunakan *Adaptive*

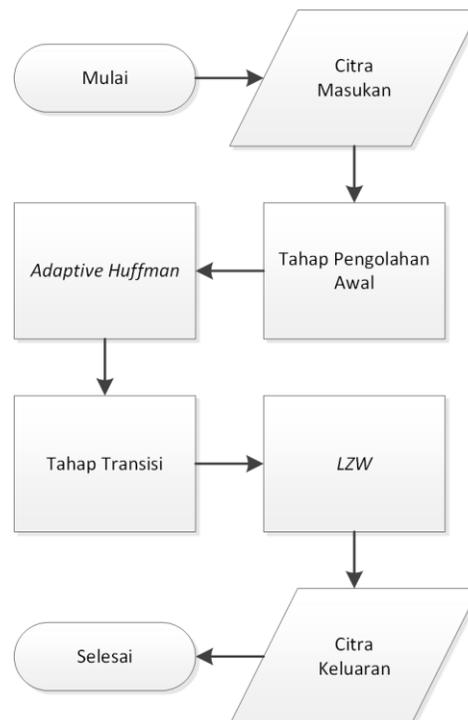
Huffman dan *LZW*, serta dengan mengubah *Last Significant Bit (LSB)* agar berkas citra dapat dikemas menjadi sebuah berkas baru yang memiliki ukuran lebih hemat dibandingkan berkas citra awal. Susunan artikel ini adalah sebagai berikut: 1 Pendahuluan, 2 Desain Algoritma, 3 Pengujian, dan 4 Kesimpulan.

II. DESAIN ALGORITMA

Alur algoritma pemampatan ini terdiri atas empat tahap sesuai dengan diagram alur yang tertera pada Gambar 1.

A. Pengolahan Awal

Menggunakan metode *mean*, median dan modus untuk menghitung nilai distribusi. Nilai distribusi digunakan untuk



Gambar 1. Diagram Alur Algoritma Pemampatan

menentukan nilai *LSB* pengganti. Dengan ketentuan sesuai dengan (1) dan (2).

$$dist = F(metode) * 100 \% \div ukuran(citra) \quad (1)$$

$$F(dist) = (dist \mid 5 < [dist] \leq 15) \quad (2)$$

Dimana $F(dist)$ adalah nilai distribusi yang didapatkan setelah melakukan penghitungan dengan metode *mean*, median maupun modus. Tidak dengan ketiganya sekaligus.

B. Adaptive Huffman

Adaptive Huffman yang digunakan mengacu pada artikel [5]. Dengan menggunakan jumlah simbol = 2^8 atau 256 karakter. Pada tahap ini berkas citra akan dimampatkan secara linear atau dengan mengubah bentuk dari dua dimensi menjadi satu dimensi.

C. Tahap Transisi

Pada tahap ini, aliran dantum hasil pemampatan dengan *Adaptive Huffman* akan dipotong-potong sesuai dengan nilai modus kedalaman Pohon *Huffman* yang terbentuk. Dengan melakukan pemotongan ini maka hasil pemampatan dengan *Adaptive Huffman* dapat dimampatkan kembali secara maksimal oleh LZW. Pemotongan ini sendiri memiliki batasan sesuai dengan (3).

$$F(s) = (s \mid 2 \leq s \leq 8) \tag{3}$$

Dimana s adalah ukuran pemotongan sepanjang s bit, dan $F(s)$ adalah ukuran terpilih.

D. LZW

LZW yang digunakan mengacu pada artikel [6], dan struktur data yang digunakan mengacu pada artikel [7]. Dengan menggunakan ukuran kamus data sesuai (4).

$$F(l) = (l \mid 8 \leq l \leq 16) \tag{4}$$

Dimana l adalah ukuran kamus data sepanjang l bit, dan $F(l)$ adalah ukuran terpilih.

III. PENGUJIAN

A. Skenario Pengujian

Pengujian yang dilakukan dibagi kedalam tiga bagian, yakni:

1. Pengujian *mean*, median dan modus;
2. Pengujian alur transisi; dan
3. Pengujian penentuan nilai *LSB*.

Parameter pengujian yang digunakan adalah rasio pemampatan dan *PSNR*. Dimana rasio pemampatan yang digunakan sesuai dengan (5).

$$rasio = \frac{ukuran(citra\ akhir)}{ukuran(citra\ awal)} \tag{5}$$

Dengan ketentuan bahwa semakin kecil rasio semakin baik dan semakin besar nilai *PSNR* semakin baik pula. Sebuah citra *lossy* dikatakan baik jika nilai *PSNR* yang dimiliki lebih besar dari 40 *dB* dan dikatakan cukup jika dibawah 30 *dB* [8].

B. Citra Pengujian

Citra yang digunakan menggunakan tipe berkas *PGM P5*. Citra-citra tersebut dapat diunduh atau diakses pada laman <http://www.ece.rice.edu/~wakin/images/> dan http://www.csd.uwo.ca/courses/CS4487a/image_samples.html.

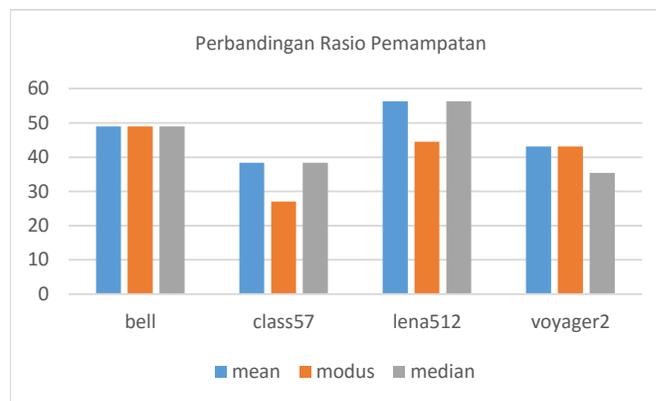
Citra-citra tersebut antara lain:

- *lena512.pgm*
- *bell.pgm*
- *class57.pgm*
- *voyager2.pgm*

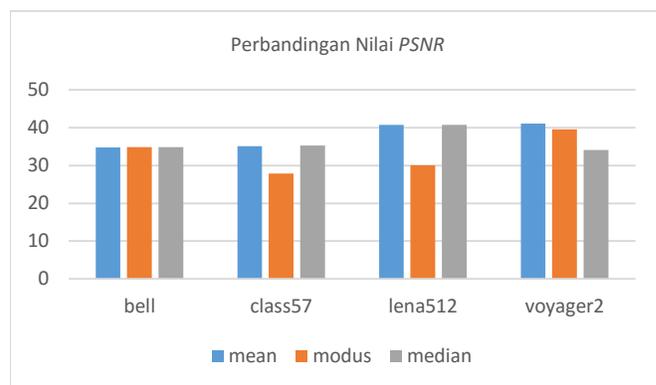
C. Hasil Pengujian

1) Pengujian Metode Mean, Median dan Modus

Hasil pengujian menunjukkan *trade-off* muncul pada metode median, dimana rerata rasio pemampatan yang diperoleh adalah 44,77 % dan rerata *PSNR* sebesar 36,26 *dB*. Grafik hasil pengujian dapat dilihat pada Gambar 2 dan 3.



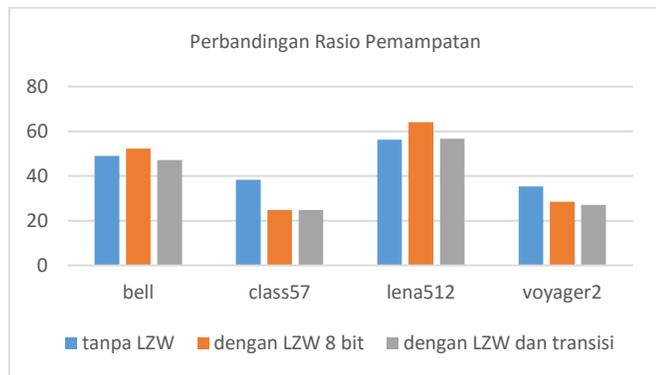
Gambar. 2. Diagram Perbandingan Rasio Pemampatan dengan Menggunakan Metode *Mean*, Median dan Modus



Gambar. 3. Diagram Perbandingan Nilai *PSNR* dengan Menggunakan Metode *Mean*, Median dan Modus

2) Pengujian Alur Transisi

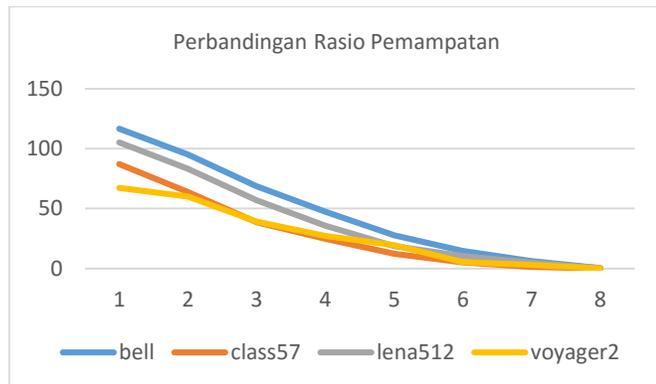
Hasil pengujian menunjukkan bahwa dengan menambahkan alur transisi *LZW* secara umum membuat hasil pemampatan menjadi lebih mangkus, dimana rasio pemampatan menjadi lebih padat dari awalnya memiliki rerata 44,77 % menjadi 38,99 %. Grafik hasil pengujian dapat dilihat pada Gambar 4. Hasil juga menunjukkan bila menggunakan tahap transisi atau pemotongan aliran dantum adalah langkah tepat.



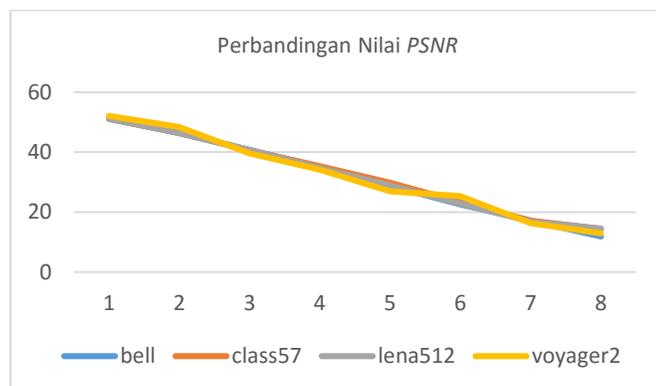
Gambar. 4. Diagram Perbandingan Rasio Pemampatan Dengan dan Tanpa LZW serta dengan Tahap transisi

3) Pengujian Penentuan Nilai LSB

Hasil pengujian menemukan nilai *trade-off* dari nilai *PSNR* dan ukuran rasio dengan berlandaskan pada nilai distribusi diantara 5 % hingga 15 % dan dipilih yang paling kecil. Batasan tersebut didapat dengan menguji setiap citra dengan nilai panjang *LSB* mulai dari 1 hingga 8 bit. Grafik hasil pengujian dapat dilihat pada Gambar 5 dan 6.



Gambar. 5. Diagram Perbandingan Rasio Pemampatan untuk Setiap LSB



Gambar. 6. Diagram Perbandingan Nilai PSNR untuk Setiap LSB

IV. KESIMPULAN

Citra keabuan dapat dimampatkan dengan menggunakan penggabungan *Adaptive Huffman* dan *LZW*, serta akan bersifat *lossy* dengan melalui tahap pengolahan awal. Dengan adanya tahap transisi, pemampatan oleh *LZW* dapat dimaksimalkan. Selain itu dengan membatasi ukuran kamus data dan melakukan *flushing* kamus data, maka ledakan kamus data pada *LZW* dapat

dicegah.

DAFTAR PUSTAKA

- [1] K. Sayood, *Introduction to Data Compression Third Edition*, San Francisco: Elsevier, 2006.
- [2] V. Crnojevic, V. Senk, Z. Trpovski, "Lossy Lempel-Ziv algorithm for image compression," *Telecommunications in Modern Satellite, Cable and Broadcasting Service*, vol. 2, 2003.
- [3] Gabriela Dudek, Przemysław Borys, Zbigniew J. Grzywna, "Lossy dictionary-based image compression method," *Image and Vision Computing*, vol. 25, no. 6, pp. 883-889, 2007.
- [4] Dalvir Kaur and Kamaljit Kaur, "Huffman Based LZW Lossless Image Compression Using Retinex Algorithm," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 8, pp. 3145-3151, 2013.
- [5] J. S. Vitter, "Design and Analysis of Dynamic Huffman Codes," *Journal of the ACM*, vol. 34, no. 4, pp. 825-845, 1987.
- [6] T. A. Welch, "A Technique for High-Performance Data Compression," *IEEE Computer*, vol. 17, no. 6, pp. 8-19, 1984.
- [7] E. Fredkin, "Trie memory," *Communications of the ACM*, 1960.
- [8] Cheddad, A., Condell, J., Curran, K., Kevitt, P.Mc, "Digital Image Steganography : Survey and Analysis of Current Methods," *Signal Processing*, vol. 90, no. 3, pp. 727-752, 2010.