

Implementasi Kompresi Citra Dengan Metode Adaptive Huffman Coding Pada Sistem Penjualan Ardawalika Event Organizer

Muhammad Hafidh^{1*}, Achmad Solichin²

^{1,2}Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia

Email: ^{1*}muhhafiedh@gmail.com, ²achmad.solichin@budiluhur.ac.id

(* : corresponding author)

Abstrak- *Event Organizer* merupakan salah satu penyedia jasa profesional dalam penyelenggara berbagai acara baik secara formal maupun non formal yang bertugas untuk membantu *client* dalam menyelenggarakan acara yang diinginkan. Aplikasi ardawalika merupakan aplikasi berbasis *web* yang bergerak dibidang jasa *event organizer*. Aplikasi ardawalika menyimpan informasi *client* di dalam *database* untuk kebutuhan bisnis. Namun semakin banyak data yang dikirimkan ke *server* mempengaruhi peningkatan memori pada *database server* terutama pada *file* citra. Sehingga dibutuhkan algoritma kompresi citra untuk mengatasinya tanpa menghilangkan atau merusak makna informasi didalamnya. Citra merupakan representasi digital dari objek gambar. Kompresi adalah proses pengubahan dari sekumpulan data menjadi suatu bentuk kode sehingga menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Proses dalam pengkodean informasi menggunakan bit atau informasi unit lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan sistem encoding tertentu. Pada tahap kompresi, Citra yang telah dijadikan *buffer* kemudian ditransformasikan menjadi bitmap RGBA (*red, green, blue, alpha*) untuk dilakukan proses pengompresan guna mengurangi kualitas dan ukuran citra sesuai dengan yang dibutuhkan aplikasi ardawalika. Tahap terakhir sebelum citra disimpan adalah dekompresi, Pada tahap ini *buffer* dari data citra yang telah melalui proses kompresi kemudian dijadikan citra kembali dari dan ditransformasikan menjadi *stringbase 64* agar dapat dibaca oleh *server*. Mengitung rasio hasil kompresi dengan melakukan pengujian terhadap 3 citra didapati hasil rasio kompresi yang didapatkan dengan menggunakan metode adaptive Huffman coding adalah 96%, 98,63%, 98,23% sedangkan untuk *library opensource sharp* adalah 97%, 98,8%, 98,4%. Implementasi Kompresi Citra dengan Metode Adaptive Huffman Coding pada sistem penjualan Ardawalika *Event Organizer* berjalan dengan baik, semakin banyak *value* yang memiliki nilai sama kompresi akan semakin efektif, Algoritma *Adaptive Huffman Coding* melakukan kompresi dengan bentuk pohon Huffman, dan pohon akan terus berkembang dan terupdate sampai *value* terakhir.

Kata Kunci: kompresi, adaptive, Huffman, citra

Implementation Of Image Compression Using The Adaptive Huffman Coding Method In The Ardawalika Event Organizer Sales System

Abstract- *Event Organizer* is one of the professional service providers in organizing various formal and non-formal events that assist clients in organizing the desired event. The ardawalika application is a web-based application engaged in event organizer services. The ardawalika application stores client information in the database for business needs. However, the more data sent to the server affects the memory increase on the database server, especially the image files. So it takes an image compression algorithm to overcome it without eliminating or destroying the meaning of the information in it. Image is a digital representation of the image object. Compression is the process of converting data into a form of code so as to save the need for storage space and time for data transmission. The process of encoding information uses bits or other units of information that are lower than the data representation that is not encoded with a certain coding system. At the compression stage, the image that has been used as a buffer is then transformed into an RGBA (*red, green, blue, alpha*) bitmap for processing. compressing to reduce the quality and size of the image as required by the ardawalika application, the last stage before the image is stored is decompression, at this stage the buffer from the image data that has gone through the compression process is then converted into an image back from and transformed into *stringbase 64* so that it can be read by the server. calculating the compression ratio by testing 3 images, it was found that the compression results obtained using the adaptive Huffman coding method were 96%, 98.63%, 98.23% while for the Sharp open source library were 97%, 98.8%, 98.4%. Implementation of Image Compression with Adaptive Huffman Coding Method in the system in the sale of Ardawalika Event Organizer is going well, the more values that have the same value the more effective it will be, the Adaptive Huffman Coding Algorithm compresses the Huffman tree shape, and the tree will continue to grow and update until the last value.

Keywords: compression, adaptive, Huffman, image

1. PENDAHULUAN

Aplikasi yang berbasis *web* dan dengan menggunakan kompresi, telepon seluler saat ini sedang berkembang karena informasi menjadi lebih mudah didapatkan dan menjadi semakin terjangkau. Banyak perusahaan konvensional yang melakukan digitalisasi. Akibatnya, berbagai jenis aplikasi dikembangkan dari bahasa

pemrograman yang berbeda dan dapat diunduh dari ponsel cerdas atau dilihat dari situs web.

Kompresi merupakan proses pengubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data[1]. Kompresi gambar adalah proses penyederhanaan data gambar menjadi kode sederhana tanpa membuang bagian-bagian yang ada didalamnya. Dengan kompresi, gambar yang memiliki ukuran besar dapat diubah menjadi gambar berukuran kecil sebelum dilakukan pengiriman[2]. Bitmap adalah sebuah bahan matriks yang menjelaskan individualitas dari titik-titik individual dalam gambar yang biasanya disebut dengan *pixel*, yang membentuk sebuah gambar. Bitmap digunakan untuk gambar-gambar seperti foto realistis dan gambar yang memerlukan detail yang rumit[3]. Citra dalam format BMP ada tiga macam, yaitu citra biner, citra berwarna, dan citra hitam putih (*grayscale*). Citra biner hanya mempunyai dua nilai keabuan, yaitu 0 dan 1.[4]. Dengan bertambahnya *user* dan makin banyaknya data yang dikirim ke *server* mempengaruhi peningkatan memori pada *database server* terutama pada *file* citra. Sehingga dibutuhkan algoritma kompresi citra untuk mengatasinya tanpa menghilangkan atau merusak makna informasi didalamnya.

Pengkodean Huffman dinamis, atau disebut juga *adaptive Huffman coding*, pertama kali disusun oleh Faller dan Gallager. Kemudian, D. E. Knuth melakukan peningkatan terhadap algoritma tersebut dan menghasilkan algoritma baru yang disebut algoritma FGK. Versi terbaru dari pengkodean Huffman dinamis dideskripsikan oleh Vitter pada tahun 1987[5]. Awalnya kompresi dimulai dengan pohon Huffman yang kosong. Tidak ada *value* yang terisi sebelumnya. *Value* pertama dimasukkan pada iterasi dalam bentuk terkompresi. *value* ini kemudian ditambahkan ke dalam pohon, jika *value* ini ditemukan lagi didalam pohon maka frekuensi bertambah satu, dan akan memodifikasi pohon.[6]. Dalam proses *encoding* dan *decoding* pada *adaptive Huffman coding*, *server* hanya memproses citra tanpa menyimpan citra terlebih dahulu. Maka digunakan *buffer* sebagai penyimpanan sementara. *buffer* adalah sepotong memori yang menyimpan paket sementara ketika ada terlalu banyak data yang dikirim ke antarmuka jaringan (baik yang dihasilkan oleh host, atau dalam kasus *switch* atau *router*, diterima pada antarmuka yang berbeda) dari apa yang dapat ditransmisikan pada media fisik antarmuka jaringan[7]. Karena *database server* memiliki penyimpanan yang terbatas. Dibutuhkan tempat penyimpanan yang terpisah untuk citra. CDN (*Content Delivery Networks*) merupakan sebuah solusi untuk mengatasi *bottleneck* dalam pengiriman konten ke pengguna, terutama dari sisi skalabilitas, ketersediaan, dan performa. Pada CDN, Konten direplikasi dari server asal ke *server-server* pengganti dalam jaringan layanan CDN di seluruh dunia[8]. Data yang dapat dibaca oleh *server* berbentuk JSON. Struktur JSON, seperti associative array atau *object* dengan properties di PHP, tanda kurung kurawal ({ }) digunakan untuk mengindikasikan level kedalaman yang baru di data struktur. Sedangkan key dan value-nya dipisahkan oleh tanda titik dua (:) dan setiap record pada level tertentu akan dipisahkan dengan tanda koma (,) [9]. Penelitian yang dilakukan oleh Putra dan Solichin yaitu melakukan kompresi pada data teks untuk mengurangi penyimpanan pada *database server*[10].

Penelitian ini diimplementasikan dengan menggunakan algoritma Adaptive Huffman Coding sebagai kompresi citra pada sistem penjualan Ardawalika *Event Organizer*. Tujuan dari implementasi ini adalah untuk mengurangi ukuran gambar yang ditransfer antara *server* dan klien. Di sisi *server*, Citra yang disimpan akan disesuaikan dengan kebutuhan klien. *Server* melakukan kompresi dan dekompresi kemudian mengirimkannya ke klien untuk ditampilkan dan mendapatkan informasi yang sebenarnya. Semua proses kompresi dan dekompresi terjadi di *server*.

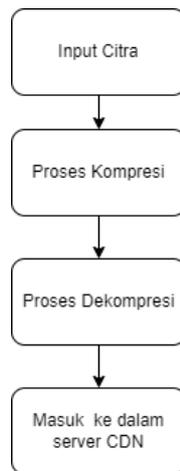
2. METODE PENELITIAN

2.1 Data Penelitian

Data pada penelitian ini adalah dengan menyediakan paket-paket milik Ardawalika yang dimana nantinya akan digunakan oleh para user baik saat ingin memilih paket yang mereka inginkan melalui aplikasi tersebut. Paket yang disediakan oleh Ardawalika sangat bervariasi. Dimulai dari paket untuk acara perusahaan atau korporasi dengan varian paket paling hemat sampai dengan paket spesial. Untuk paket pernikahan varian paketnya dimulai dari standar sampai pernikahan mewah, dan untuk paket keluarga dimulai dari paket reuni keluarga hemat sampai paket keluarga spesial. Dan pada aplikasi ini pun menyediakan paket custom, dimana nantinya para user yang akan memesan mereka bisa mengkustom paket sesuai dengan budget yang ardawalika miliki.

2.2 Penerapan Metode

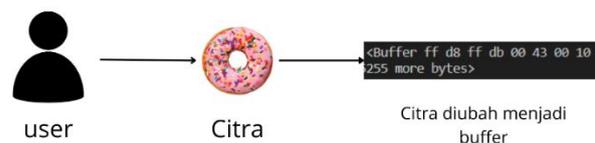
Proses kompresi citra pada penelitian ini dengan menggunakan metode *adaptive Huffman coding* terdapat 4 tahapan yang menjadi rancangan utama. Rancangan ini sebagai gambaran proses kompresi dari awal sampai akhir. Proses dapat dilihat pada gambar 1.



Gambar 1. Penerapan Metode Adaptive Huffman Coding

2.2.1 Input Citra

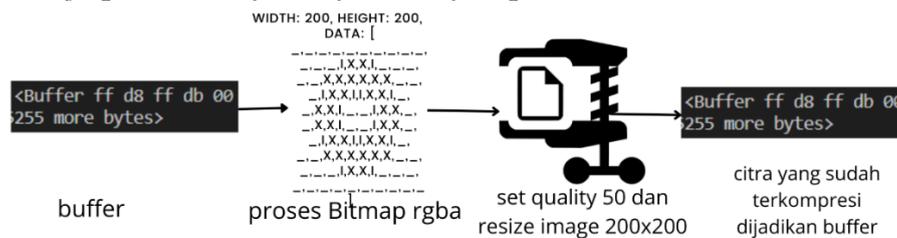
Pada tahap ini *user* menginputkan *file* citra yang akan dilakukan proses kompresi. Citra yang dimasukan harus berektensi JPEG atau PNG. Hasil dari inputan di ubah menjadi *buffer* dan dikirim untuk dilakukan kompresi. Gambar 2 merupakan tahapan yang dilakukan sebelum proses kompresi terjadi. Pengubahan citra menjadi menjadi buffer tergantung dengan ukuran citra yang dimasukan.



Gambar 2. Input Citra

2.2.2 Kompresi

Pada tahap ini, inputan yang diterima kemudian di transformasikan menjadi bitmap RGBA. Selanjutnya bitmap dilakukan operasi untuk menurunkan kualitas gambar menjadi 50, dan mengubah ukuran gambar menjadi 200x200 untuk lebar dan Panjang. Proses kompresi dapat dilihat pada gambar 3.



Gambar 3. Penerapan Kompresi Metode Huffman Coding

2.2.3 Dekompresi

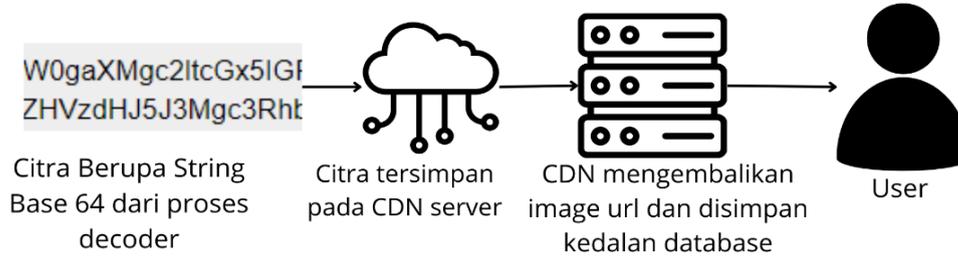
Pada tahap ini hasil dari proses Kompresi dikembalikan menjadi citra yang sudah terkompresi dan menjadikan string base64 agar terbaca oleh *sever* CDN. Dapat dilihat pada gambar 4.



Gambar 4. Penerapan Dekompresi Adaptive Huffman Coding

2.2.4 Server CDN

Server CDN *imagekit* menerima *image* dengan format Base64 string dan mengembalikan *image url*. Selanjutnya *image url* disimpan kedalam database. Gambar 5 merupakan



Gambar 5. Penerapan Penyimpanan Citra Kedalam CDN server

2.3 Spesifikasi Database

Spesifikasi basis data menggambarkan struktur data fisik dari sebuah sistem atau aplikasi. Dalam spesifikasi ini dijelaskan data yang akan disimpan kedalam *server*. Spesifikasi basis data dapat dilihat pada tabel 1.

Tabel 1. Spesifikasi tabel *users*

Nama Field	Type Data	Ukuran	Keterangan
<i>id</i>	<i>Serial</i>	4	<i>Id User</i>
<i>email</i>	<i>Varchar</i>	255	<i>Email User</i>
<i>password</i>	<i>varchar</i>	255	<i>Password user</i>
<i>username</i>	<i>Varchar</i>	255	<i>Username user</i>
<i>fullname</i>	<i>Varchar</i>	255	<i>Nama user</i>
<i>phoneNumber</i>	<i>Varchar</i>	255	<i>Telepon user</i>
<i>imgUrl</i>	<i>Varchar</i>	255	<i>Foto user</i>
<i>address</i>	<i>Varchar</i>	255	<i>Alamat user</i>
<i>role</i>	<i>Enum Users role</i>	255	<i>Hak akses user</i>
<i>status</i>	<i>Varchar</i>	255	<i>Status aktif user</i>
<i>createdAt</i>	<i>Timestamp</i>	4	Waktu membuat data pribadi
<i>updatedAt</i>	<i>Timestamp</i>	4	ketika <i>user</i> mengganti data pribadi

3. HASIL DAN PEMBAHASAN

3.1 Lingkungan Percobaan

Agar aplikasi sistem penjualan dengan menggunakan metode kompresi citra ini dapat berjalan dengan baik, maka spesifikasi yang dipakai untuk implementasi aplikasi ini juga harus mendukung. Spesifikasi berikut ini adalah agar dapat mendukung sistem ini berjalan, diantaranya adalah:

a. Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) yang digunakan untuk implementasi aplikasi adalah sebagai berikut:

1. Sistem Operasi Microsoft Windows 11 64-bit
2. Bahasa Pemrograman JavaScript
3. Database Postgres
4. Visual Studio Code (VSC)
5. Google Chrome (Official Build) (64-bit)
6. Framework nodejs
7. CDN Imagekit

b. Perangkat Keras (*Hardware*)

Perangkat Keras (*Hardware*) yang akan digunakan untuk implementasi aplikasi adalah sebagai berikut:

1. Processor Intel i7-Gen 8th
2. RAM / Memory 16GB
3. Laptop MSI GL638RCS
4. SSD 256GB

3.2 Implementasi Metode Adaptive Huffman Coding

3.2.1 Input Citra

Tahapan ini menjadi langkah awal yang harus dilalui agar kompresi dapat dilakukan, pada tahapan ini *user* memasukan *file* citra dan juga data diri. Data pribadi dimasukan ke database sedangkan citra di proses untuk dilakukan kompresi. Pada gambar 6 merupakan contoh inputan citra yang akan dilakukan proses kompresi.



Gambar 6. Sample Citra

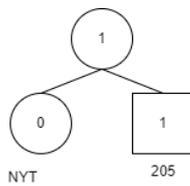
3.2.2 Kompresi

Pada tahap ini citra dijadikan *bitmap* untuk dilakukan operasi kompresi. Tabel 2 merupakan contoh sample citra yang telah dijadikan *bitmap*.

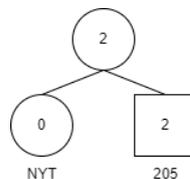
Tabel 2. Sample value 3x3

205	205	206
204	206	203
204	204	205

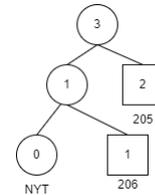
Selanjutnya dilakukan perhitungan frekuensi untuk setiap nilai pada *sample value*. Dengan menggunakan pohon Huffman yang berfungsi untuk mengkompres citra. Pada gambar 7 merupakan proses iterasi untuk setiap *value* dari *bitmap*.



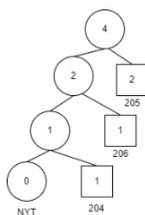
Gambar 7 (a). Iterasi ke-1



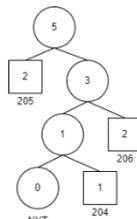
Gambar 7 (b). Iterasi ke-2



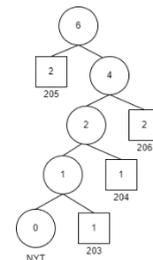
Gambar 7 (c). Iterasi ke-3



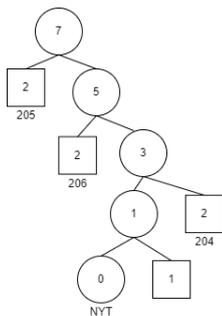
Gambar 7 (d). Iterasi ke-4



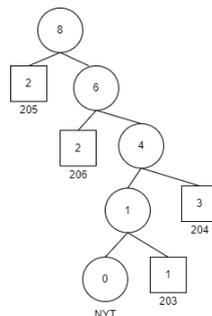
Gambar 7 (e). Iterasi ke-5



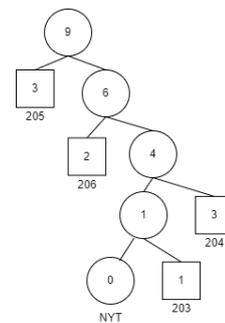
Gambar 7 (f). Iterasi ke-6



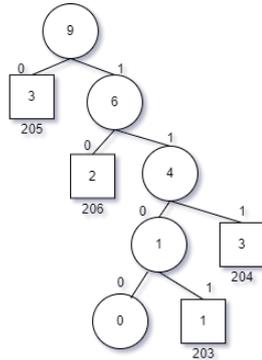
Gambar 7 (g). Iterasi ke-7



Gambar 7 (h). Iterasi ke-8



Gambar 7 (i). Iterasi ke-9



Gambar 7 (j). Penambahan *Codeword*

Codeword merupakan untaian biner hasil akhir dari iterasi *Huffman tree* diberikan code dengan 0 dan 1 sebagai *codeword*. Hasil *codeword* dapat dilihat pada tabel 3.

Tabel 3. *codeword*

value	frekuensi	<i>CodeWord</i>
203	1	1101
204	3	111
205	3	0
206	2	10

3.2.3 Dekompresi

Pada tahap ini *codeword* yang sudah didapatkan pada proses kompresi akan dikembalikan menjadi citra yang sudah terkompresi. Dapat dilihat pada tabel 4.

Tabel 4. Tabel *Decoder*

<i>CodeWord</i>	value
1101	203
111	204
0	205
10	206

3.2.4 Server CDN

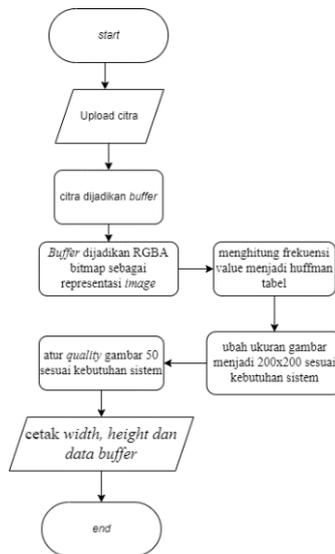
Server CDN imagekit akan menyimpan citra hasil kompresi dan mengembalikan *image url* kemudian disimpan ke dalam database sistem.

3.3 Flowchart

Flowchart adalah jenis diagram yang mewakili alur kerja atau proses. *Flowchart* juga dapat didefinisikan sebagai representasi diagram dari suatu algoritma, pendekatan langkah demi langkah untuk menyelesaikan tugas. *Flowchart* menunjukkan langkah-langkah sebagai kotak dari berbagai jenis, dan urutannya dengan menghubungkan kotak dengan panah.

3.3.1 Flowchart Kompresi

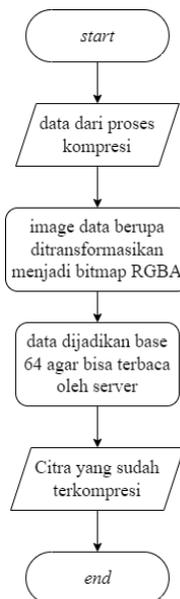
Flowchart pada gambar 8 menggambarkan alur proses adaptive Huffman Coding Untuk proses *encoder*. Diawali dengan memilih *folder* citra yang akan dikompresi. Hasil akhir dari proses kompresi adalah *buffer image* yang akan diproses selanjutnya.



Gambar 8. Flowchart proses kompresi

3.3.2 Flowchart Dekompresi

Flowchart pada gambar 9 menggambarkan alur proses *adaptive Huffman Coding* Untuk proses dekompresi. Data yang didapatkan dari proses kompresi selanjutnya dikembalikan menjadi citra untuk disimpan kedalam server.



Gambar 9. Flowchart proses dekompresi

3.4 Pengujian

Pada tahap ini dilakukan pengujian dengan menggunakan 3 foto untuk mengetahui ratio hasil kompresi dengan metode *adaptive Huffman coding*. Serta mengetahui perbedaan citra sebelum dan sesudah dilakukan kompresi. Hasil kompresi dapat dilihat pada tabel 5 dan 6.

Tabel 5. Pengujian Kompresi Citra

No.	Size File Base 64 Sebelum Dikompresi (Bytes)	Size File Base 64 Sesudah Dikompresi (Bytes)	Ratio Kompresi	Gambar Sebelum Dikompresi	Gambar Sesudah Dikompresi
1	275512	11160	96%		
2	1427032	19432	98,63%		
3	1103008	19488	98,23%		

Tabel 6. Pengujian Dengan *Library Sharp*

No.	Size File Base 64 Sebelum Dikompresi (Bytes)	Size File Base 64 Sesudah Dikompresi (Bytes)	Ratio Kompresi	Gambar Sebelum Dikompresi	Gambar Sesudah Dikompresi
1	275512	8536	97%		
2	1427032	17704	98,8%		

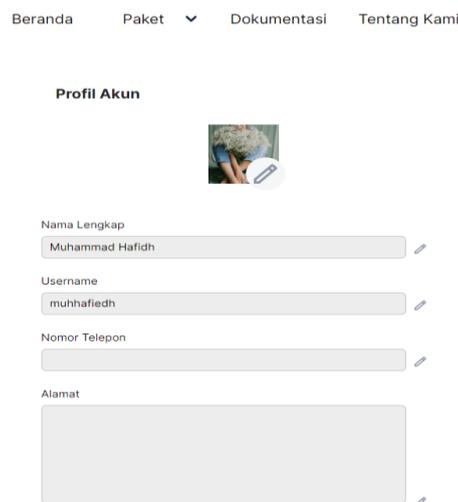
No.	Size File Base 64 Sebelum Dikompresi (Bytes)	Size File Base 64 Sesudah Dikompresi (Bytes)	Ratio Kompresi	Gambar Sebelum Dikompresi	Gambar Sesudah Dikompresi
3	1103008	17200	98,4%		

3.5 Tampilan Layar

Pada bagian ini diuraikan mengenai tampilan layar implementasi kompresi citra dengan menggunakan algoritma *Adaptive Huffman Coding* pada sistem penjualan Ardawalika *Event Organizer* mulai dari pertama kali dijalankan sampai dengan selesai. Berikut ini akan diberikan penjelasan dari gambar masing-masing tampilan yang ada pada aplikasi ini.

3.6 Tampilan Layar *User Profile*

Pada tampilan *user profile* berisikan *field* untuk mengganti data pribadi atau mengubah password. Kompresi terdapat pada tampilan layar *user profile*. tampilan *user profile* dapat dilihat pada gambar 10.



Gambar 10. Tampilan Layar *User Profile*

3.7 Evaluasi Program

Evaluasi program merupakan salah satu hal yang perlu dilakukan dalam pengembangan aplikasi untuk menganalisa dan mengetahui hasil yang telah dicapai oleh aplikasi yang telah dibuat dalam penelitian ini, dilakukan evaluasi program untuk menggunakan hasil yang dicapai pada aplikasi ini. Dalam aplikasi ini ditemukan beberapa kelebihan dan kekurangan antara lain :

Kelebihan:

- 1) Memiliki rasio kompresi yang cukup besar.
- 2) Dibandingkan dengan *library open source sharp* citra tidak terpotong.

Kekurangan:

- 1) Jika dibandingkan dengan kompresi yang dilakukan dari *opensource library sharp*, metode adaptive Huffman coding pada aplikasi ardawalika memiliki rata-rata kompresi yang lebih kecil
- 2) Aplikasi ini tidak bisa meng-cancel untuk orderan yang telah disetujui oleh *user*
- 3) Aplikasi belum sepenuhnya *compatible* untuk *mobile device*

4. KESIMPULAN

a) Kesimpulan

Dari pengujian yang dilakukan pada bab sebelumnya dapat disimpulkan bahwa dari kompresi citra yang dilakukan berhasil menjawab tujuan dari penelitian dan dapat ditarik kesimpulan :

- 1) Implementasi Kompresi Citra dengan Metode *Adaptive Huffman Coding* pada sistem penjualan Ardawalika *Event Organizer* dapat berjalan dengan baik.
- 2) Kompresi rata-rata yang dicapai dengan Metode *Adaptive Huffman Coding* pada citra yang berektensi JPEG dan PNG adalah 97,62 %
- 3) Kompresi rata-rata yang dicapai oleh library *open source sharp* 98.1%. karena *open library sharp* memotong gambar dan tidak memperkecil ukuran gambar.
- 4) Kompresi citra semakin efektif apabila citra memiliki banyak *value* yang sama dan lebih berwarna.

b) Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut pada aplikasi ini, agar menjadi lebih baik adalah sebagai berikut:

- 1) Peneliti selanjutnya diharapkan dapat memvariasikan format ekstensi selain JPEG atau PNG
- 2) Peneliti selanjutnya diharapkan dapat membuat aplikasi yang compatible untuk *mobile device* dan *desktop*

DAFTAR PUSTAKA

- [1] W. E. Pangesti, G. Widagdo, D. Riana, And S. Hadianti, "Implementasi Kompresi Citra Digital Dengan Membandingkan Metode Lossy Dan Lossless Compression Menggunakan Matlab," *J. Khatulistiwa Inform.*, Vol. 8, No. 1, Pp. 53–58, 2020.
- [2] K. Y. Dewi And H. Nurwasito, "Implementasi Algoritma Huffman Untuk Kompresi Gambar Pada Jaringan 6lowpan," Vol. 6, No. 4, Pp. 2548–964, 2022.
- [3] G. Y. Swara, "Pemanfaatan Visualisasi 3d Pada Multimedia Interaktif Dalam Pengenalan Penyakit Demam Berdarah," *J. Teknoif Tek. Inform. Inst. Teknol. Padang*, Vol. 8, No. 1, Pp. 19–24, 2021.
- [4] E. Haryanto, "Penerapan Metode Huffman Dalam Pemampatan Citra Digital," *Semin. Nas. Inform.*, Vol. 1, No. 1, Pp. 283–287, 2015.
- [5] T. J. Pattiasina, "Analisa Kode Huffman Untuk Kompresi Data Teks," *Teknika*, Vol. 1, No. 1, Pp. 1–12, 2012.
- [6] S. Sepandi, "Penerapan Algoritma Adaptive Huffman Coding Pada Aplikasi Kumpulan Komik Berbasis Android," *Pelita Inform. Inf. Dan Inform.*, Vol. 9, Pp. 136–141, 2020.
- [7] A. Fauzan And Fathiah, "Traffic Counting Studi Kasus Di Jalan Teuku Nyak Arief," *J. Informatics Comput. Sci.*, Vol. 5, No. 1, Pp. 75–83, 2019.
- [8] D. Laksmiati, "Implementasi Content Delivery Network (Cdn) Untuk Optimasi Kecepatan Akses Website," *Akrab Juara*, Vol. 5, No. 1, Pp. 49–56, 2020.
- [9] R. Maulana, A. Sulistyanto, And A. S. Rini, "Perancangan Sistem Informasi Pengajuan Dan Pelaporan Pembayaran Tunjangan Kinerja Pada Lembaga Pemasyarakatan Salemba Berbasis Web Menggunakan Skema Rest Api," *J. Manajemen Inform. Jayakarta*, Vol. 1, No. 4, P. 283, 2021.
- [10] M. A. Putra And A. Solichin, "Optimasi Web Service Dengan Penerapan Algoritma Kompresi Lzw: Studi Kasus Aplikasi BluCampus Universitas Budi Luhur," *Skanika*, vol. 1, no. 2, pp. 455–462, 2018.