



Analisis Perbandingan Kompresi Citra Menggunakan Algoritma Run Length Encoding dan Algoritma Fixed Length Binary Encoding

Umi Hani Lestari*, Yusuf Ramadhan Nasution, Ibnu Rusydi

Fakultas Sains dan Teknologi, Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia

Email: ^{1,*}umihanilestari01@gmail.com, ²ramadhannst@uinsu.ac.id, ³ibnurussydi@uinsu.ac.id

Email Penulis Korespondensi: umihanilestari01@gmail.com

Abstrak—Teknologi berkembang dengan sangat cepat dan akan terus meningkat, sehingga menjadi peran penting dalam proses pengiriman informasi atau data dari satu perangkat ke perangkat lain. Kecepatan pengiriman tergantung pada ukuran dari data yang akan dikirim. Data dengan ukuran yang lebih besar memerlukan waktu pengiriman yang lebih lama. Jumlah ruang penyimpanan yang dibutuhkan meningkat jika semakin banyak file yang disimpan. Hal tersebut mendorong pengembangan teknik pengecilan file yang juga disebut sebagai teknik kompresi data, dengan tujuan meminimalisir berkurangnya kualitas data setelah melakukan pengiriman dan mengurangi jumlah ruang penyimpanan yang diperlukan. Teknik kompresi mempunyai beberapa algoritma yang dapat digunakan untuk mengecilkan ukuran file. Seperti pada penelitian ini, proses kompresi dilakukan dengan algoritma run length encoding dan algoritma fixed length binary encoding. Kedua algoritma tersebut memiliki hasil kompresi yang berbeda, sehingga perlu melakukan perbandingan. Untuk melakukan perbandingan digunakan 6 file citra grayscale berekstensi *.jpg dengan resolusi yang berbeda-beda dan membandingkan kinerjanya sesuai dengan parameter yang telah ditetapkan. Hasil perbandingan kompresi dari salah satu data citra yang beresolusi 300 x 300 pada algoritma Run Length Encoding memiliki Ratio of Compression (RC) 1.038792, Compression Ratio (CR) 96.266%, Redundancy (Rd) 3.734%, Waktu kompresi 399ms, dan Waktu dekompresi 297ms. Sedangkan pada algoritma Fixed Length Binary Encoding memiliki Ratio of Compression (RC) 1.37, Compression Ratio (CR) 73.248%, Redundancy (Rd) 26.752%, Waktu kompresi 3258ms, dan Waktu dekompresi 1047ms. Sehingga dari hasil tersebut dapat dikatakan kinerja yang lebih baik dalam melakukan kompresi citra adalah algoritma Fixed Length Binary Encoding dibandingkan dengan Run Length Encoding.

Kata Kunci: Kompresi; Citra; Run Length Encoding; Fixed Length Binary Encoding

Abstract—Technology is developing very quickly and will continue to increase, so it plays an important role in the process of sending information or data from one device to another. The speed of transmission depends on the size of the data to be sent. Data with a larger size requires a longer delivery time. The amount of storage space required increases as more files are stored. This has led to the development of file shrinking techniques, also known as data compression techniques, with the aim of minimizing the loss of data quality after transmission and reducing the amount of storage space required. Compression techniques have several algorithms that can be used to reduce file size. As in this research, the compression process is done with the run length encoding algorithm and the fixed length binary encoding algorithm. Both algorithms have different compression results, so it is necessary to make a comparison. To make the comparison, 6 grayscale image files with *.jpg extension are used with different resolutions and compare their performance according to predetermined parameters. The compression comparison results of one image data resolution of 300 x 300 in the Run Length Encoding algorithm has a Ratio of Compression (RC) 1.038792, Compression Ratio (CR) 96.266%, Redundancy (Rd) 3.734%, Compression time 399ms, and Decompression time 297ms. While the Fixed Length Binary Encoding algorithm has a Ratio of Compression (RC) of 1.37, Compression Ratio (CR) of 73.248%, Redundancy (Rd) of 26.752%, Compression time of 3258ms, and Decompression time of 1047ms. So from these results it can be said that the better performance in compressing images is the Fixed Length Binary Encoding algorithm compared to Run Length Encoding.

Keywords: Compression; Image; Run Length Encoding; Fixed Length Binary Encoding

1. PENDAHULUAN

Teknologi informasi telah berkembang dengan sangat cepat dan akan terus meningkat sehingga memberikan kemudahan dalam kehidupan manusia di beberapa bidang. Salah satu kemudahan yang ditawarkan adalah pengembangan teknik pengecilan file yang biasa dikenal dengan teknik kompresi data (Prabiantissa et al., 2023). Teknik kompresi saat ini memungkinkan data dimampatkan sehingga menjadi ukuran yang jauh lebih kecil dari ukuran aslinya. Teknik kompresi dikembangkan dengan tujuan untuk mengurangi ukuran file sehingga tidak mengurangi kualitas pada data saat melakukan pengiriman data (Saragih & Utomo, 2020). Teknik kompresi citra yaitu suatu teknik merubah data berbentuk kode ASCII atau angka menjadi sebuah kode atau sandi, kemudian memanipulasi kode ini agar dapat dikembalikan ke bentuk data asli (Sunardi et al., 2021). Teknik kompresi terbagi menjadi 2 yakni kompresi *lossy* dan kompresi *lossless*. Kompresi *Lossless* didefinisikan sebagai suatu metode kompresi data dimana selama proses kompresi ini tidak terdapat informasi yang berkurang maupun hilang. Kompresi *Lossy* merupakan metode untuk kompresi data, dimana proses kompresinya dengan cara menghapus atau menghilangkan beberapa informasi dari data asli, namun tidak menghilangkan banyak informasi yang terkandung pada data (Pangesti et al., 2020). Dalam kompresi *lossy*, kesalahan tidak dapat ditoleransi sehingga data sebelum dikompresi harus sama dengan yang dihasilkan dari kompresi. Namun, pada kompresi *lossless*, kesalahan masih diperbolehkan selama tidak mengubah pola data (Prayoga & Suryaningrum, 2018). Kualitas gambar yang dihasilkan dengan metode ini akan bergantung pada seberapa besar nilai data gambar yang dikurangi. Semakin besar nilai data gambar yang dikurangi, semakin rendah kualitas gambar (Sanjaya & Aria, 2018).

Perbandingan kompresi citra merupakan pembuktian untuk mengetahui teknik kompresi mana yang lebih baik untuk digunakan. Teknik kompresi mempunyai tujuan serupa, akan tetapi karena ada beberapa teknik kompresi, maka

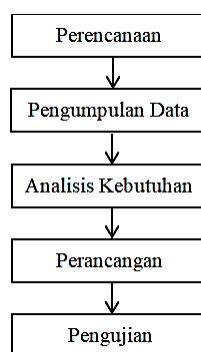
hasil kompresinya akan berbeda. Maka dari hal tersebut, dibutuhkan perbandingan teknik kompresi supaya teknik kompresi yang digunakan sesuai dengan keinginan. Perbandingan kinerja dari masing-masing algoritma dapat dilihat berdasarkan parameternya. Parameter penelitian ini adalah *Compression Ratio* (Cr), *Ratio of Compression* (RC), *Redudancy* (Rd), serta waktu yang diperlukan untuk melakukan kompresi dan dekompresi *file* citra. Algoritma yang digunakan dalam rangka membandingkan kinerja kompresi citra penelitian ini adalah Algoritma *Fixed Length Binary Encoding* dan Algoritma *Run Length Encoding*.

Penelitian yang dilakukan oleh Sartika Siahaan, menyimpulkan bahwa uji coba yang dilakukan untuk membandingkan kompresi file audio antara algoritma Fibonacci dan Invert Elias Delta menunjukkan hasil yang berbeda. Algoritma Fibonacci mendapatkan hasil 11,074, sedangkan algoritma Invert Elias Delta mendapatkan hasil 10,217. Sehingga jumlah usaha yang dilakukan oleh algoritma semakin besar ketika nilai totalnya lebih besar (Siahaan, 2024). Penelitian yang dilakukan Pujiyanto, Mujito, Prasetyo, dan Prabowo menyimpulkan bahwa pada kompresi metode *Huffman* ukuran file mengalami penurunan setelah proses kompresi seluruh *dokument*. Sedangkan pada metode *Run Length Encoding* terdapat 1 file berjenis PDF yang mengalami kenaikan ukuran file awalnya 402237 Bytes menjadi 402813 Bytes. Sehingga metode *Huffman* lebih baik untuk melakukan proses kompresi dibandingkan dengan metode *Run Length Encoding* (Pujiyanto et al., 2020). Penelitian yang dilakukan Sari Magdalena Simanjuntak, menyimpulkan bahwa proses kompresi dan dekompresi menggunakan algoritma Shannon Fano Code dan Fibonacci Code dilakukan dengan mengubah sampel data menjadi bilangan heksadesimal menggunakan aplikasi Hxd. Kemudian, proses kompresi menggunakan algoritma Shannon Fano Code dan Fibonacci Code menghasilkan rasio kompresi, yang dihasilkan dengan menggunakan algoritma ini. Dengan menggunakan 48 sampel yang sama, hasil rasio kompresi yang dihasilkan algoritma Shannon Fano adalah 69% perubahan, sedangkan hasil rasio kompresi yang dihasilkan Fibonacci adalah 66% perubahan (S. M. Simanjuntak, 2023). Penelitian yang dilakukan Ledi Varia Simanjuntak, menyimpulkan bahwa uji coba yang dilakukan dengan algoritma Elias Delta Code dan Levenstein menunjukkan bahwa program hanya dapat mengompresi file dengan ekstensi *.txt. Program kompresi file ini digunakan dalam bahasa pemrograman Visual Basic.Net 2008 sangat baik dalam kompresi dan dekompresi file. Nilai rasio kompresi dipengaruhi oleh isi file. Semakin banyak pengulangan karakter yang ada dalam file yang dikompresi, semakin besar rasio kompresi file tersebut (L. V. Simanjuntak, 2020).

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini bertujuan melihat perbandingan kinerja dari Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* dalam mengompresi gambar. *Research and Development* (R&D) atau metode penelitian dan pengembangan digunakan untuk metode penelitian ini. Metode ini adalah sebuah metode yang dipergunakan dalam rangka menghasilkan suatu produk serta untuk menguji efektivitas produk itu sendiri untuk membandingkan kinerja algoritma dalam mengompresi gambar. Adapun tahapan pada metode R&D ini yaitu pada Tahap pertama, mengumpulkan informasi tentang masalah dan kebutuhan penelitian. Tahap kedua, menyusun rencana penelitian, menentukan tugas dan tujuan penelitian. Tahap ketiga, menyiapkan komponen serta data pendukung untuk pembuatan aplikasi. Tahap keempat, menguji coba program dalam skala yang terbatas. Tahap kelima, melakukan perbaikan aplikasi yang didasarkan pada uji coba sebelumnya. Tahap keenam, merupakan tahap utama uji coba yang dilakukan berdasarkan hasil perbaikan dari tahap keempat, hasil dari tahap ini mengevaluasi pencapaian hasil uji coba yaitu hasil aplikasi (Rumetna et al., 2020). Tahapan pelaksanaan penelitian dan membangun sistem bisa dilihat pada gambar 1 di bawah ini:



Gambar 1. Tahapan Penelitian

2.1.1 Perencanaan

Perencanaan penelitian ini dimulai dengan mengidentifikasi permasalahan yang ditemukan. Selanjutnya merencanakan proses-proses yang akan terjadi pada aplikasi. Sistem yang dibangun pada penelitian ini digunakan untuk



membandingkan kinerja Algoritma *Fixed Length Binary Encoding* dan Algoritma *Run Length Encoding* dalam kompresi citra.

2.1.2 Pengumpulan Data

Data penelitian ini dikumpulkan melalui beberapa tahapan atau metode yaitu dokumentasi serta studi literatur.

a. Studi Literatur

Tahap ini mengumpulkan referensi dari jurnal, buku (*e-book*), tugas akhir, serta sumber lain yang menyangkut Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* dalam mengkompresi gambar. Tujuannya adalah mendapatkan kebutuhan data dan informasi untuk menyusun tugas akhir.

b. Dokumentasi

Ini adalah tahapan dimana semua aktivitas pembuatan sistem didokumentasikan secara tertulis berbentuk skripsi.

2.1.3 Analisis Kebutuhan

Dalam pembuatan sistem diperlukan analisis kebutuhan. Terdapat dua bagian dari analisis ini, yaitu kebutuhan fungsional dan non-fungsional.

a. Kebutuhan fungsional yaitu fungsi atau proses yang perlu untuk dipenuhi sistem yang dirancang. Kebutuhan ini, mencakup:

1. Sistem harus dapat membaca informasi pada *file* gambar keabuan (*grayscale*) berekstensi *.jpg.
2. Sistem harus dapat melakukan kompresi pada *file* gambar dengan algoritma *Fixed Length Binary Encoding* dan algoritma *Run Length Encoding*, dan dapat melakukan dekompresi yang berupaya mengembalikan gambar asli sebelum dikompresi.
3. Sistem harus dapat menghitung dan menampilkan hasil dari beberapa parameter perbandingan setelah dikompresi yaitu *Compression Ratio* (Cr), *Ratio of Compression* (RC), waktu, serta *Redudancy* (Rd).

b. Kebutuhan non-fungsional merupakan kebutuhan yang perlu untuk dipenuhi oleh sistem agar dapat menerima masukan yang sesuai dari pengguna. Kebutuhan ini mencakup:

1. Efisiensi
Sistem dirancang dengan tampilan sederhana dan menarik untuk mempermudah pengguna menggunakan aplikasi.
2. Performa
Sistem dalam hal ini diharuskan bisa menjalankan tugas dengan baik dan sistem harus dapat menampilkan hasil kompresi dan hasil dekompresi.
3. Dokumentasi
Sistem yang dirancang diharuskan untuk bisa menyimpan *file* yang diperoleh dari proses kompresi dan dekompresi.
4. Kontrol
Sistem yang dirancang harus tetap terkendali agar kinerja sistem tetap beroperasi dengan baik sesuai keinginan pengguna.
5. Informasi
Sistem yang dirancang harus dapat memberikan informasi tentang berbagai data yang digunakan pada sistem.

2.1.4 Perancangan

Tahap perancangan adalah tahapan di mana sistem akan dirancang sesuai analisis kebutuhan. Tujuan perancangan adalah untuk menunjukkan gambaran yang jelas dan membangun rancangan sesuai dengan yang telah ditentukan.

2.1.5 Pengujian

Pelaksanaan pengujian sistem ditujukan dalam rangka mengetahui kinerja sistem serta melihat apakah dalam hal ini sistem beroperasi secara baik sesuai fungsionalitas yang diharapkan. Pelaksanaan pengujian ini juga untuk mengetahui bagaimana perbandingan kedua metode dalam mengkompresi citra. Pengujian ini dilakukan dengan menginput gambar, gambar yang dimasukkan harus sesuai persyaratan untuk dapat diproses sistem, selanjutnya sistem memprosesnya menurut algoritma yang dipilih, selanjutnya sistem akan menghitung nilai parameter pada gambar, setelah itu akan terlihat hasil dari citra yang telah dikompresi. Sistem ini diuji dengan menggunakan *Black Box Testing*.

2.2 Kompresi

Kompresi data adalah proses mengubah kumpulan data menjadi kode untuk menghemat ruang penyimpanan dan mempercepat proses transmisi sehingga lebih mudah untuk mengirimkan data. Proses ini dilakukan dengan menemukan pola pengulangan dalam data dan menggantinya dengan suatu tanda (Fitriya et al., 2017). Kompresi bisa dikatakan sebagai proses memperkecil ukuran asli dan jumlah ukuran data, sehingga dengan hal ini tidak memerlukan ruang penyimpanan yang besar (Nas et al., 2019). Tujuan dari kompresi citra digital adalah menekan redundansi informasi pada suatu citra, oleh karena itu bisa dilakukan penyimpanan dan bisa dikirimkan secara efisien (Putra & Muhammad Trisna Aryun, 2021). Proses kompresi ini memiliki kebalikan yang dinamakan proses dekompresi. Ini merupakan suatu proses dimana data baru dari proses kompresi dikembalikan ke data asli sebelum dikompresi (Utomo et al., 2021).



2.3 Konsep Kompresi

Proses kompresi dilakukan dengan membaca bit data setiap delapan bit, oleh karena itu akan membentuk satu karakter yang bisa dibaca komputer. Metode kompresi data menghasilkan bit data baru yang sifatnya lebih sedikit daripada bit data sebelum proses kompresi. Secara umum bit yang lebih sedikit tidak dapat dibaca komputer sebelum proses dekompresi dijalankan. Karakter pada komputer diwakili 8 bit bilangan biner dalam ASCII. Jika jumlah bit data bukan kelipatan 8, maka dibuat variabel baru yang menambahkan bit data sehingga bit data habis dibagi 8. Variabel tersebut disebut dengan *padding* dan *flag* (Ginting, 2017).

a. *Padding*

Padding adalah penambahan bit 0 pada hasil proses kompresi sebanyak jumlah bit data yang kurang sehingga bit data dapat dibagi delapan. Misalnya 1001011, dimana terdapat 7 bit data dalam bilangan biner. Kemudian, bit 0 ditambahkan 1 kali agar jumlah bit data habis dibagi delapan. Jadi, setelah *padding* bit bit data menjadi 10010110.

b. *Flag*

Flag adalah penambahan bilangan biner sebanyak delapan bit setelah proses *Padding*, yang menunjukkan bahwa terdapat *n* buah *padding* dalam bit data yang dihasilkan dari kompresi. Tujuan penambahan *flag* ini adalah untuk membuat bit data hasil kompresi yang dihasilkan selama proses dekompresi lebih mudah dibaca. Misalkan bit data yang diberi *padding* adalah 10010110. Karena terdapat 1 bit penambahan *padding* maka *flag* nya adalah bilangan biner dari 1 dengan panjang 8 bit, yaitu 00000001, sehingga bit data menjadi 1001011000000001 (Sandra, 2023).

2.4 Parameter Analisis Kompresi

Parameter yang digunakan untuk mengukur kualitas dari suatu teknik kompresi data adalah sebagai berikut:

- a. *Ratio of Compression (RC)* sendiri merupakan suatu perbandingan antara ukuran bit data sebelum kompresi dan sesudah kompresi. Hal ini bisa dituliskan sebagaimana berikut:

$$Rc = \frac{\text{ukuran bit data sebelum dikompresi}}{\text{ukuran bit data setelah dikompresi}} \quad (1)$$

- b. *Compression Ratio (CR)* bisa dikatakan sebagai perbandingan persentase antara bit data terkompresi dengan tidak terkompresi. Secara matematis, dapat dituliskan:

$$Cr = \frac{\text{ukuran bit data setelah dikompresi}}{\text{ukuran bit data sebelum dikompresi}} \times 100 \% \quad (2)$$

- c. *Redudancy (Rd)* adalah persentase perbedaan antara ukuran data sebelum dan ukuran setelah kompresi. Secara matematis, dapat dituliskan:

$$Rd = 100\% - Cr \quad (3)$$

- d. Waktu kompresi dan dekompresi merupakan kebutuhan waktu untuk menyelesaikan kedua proses tersebut. Waktu yang semakin singkat, maka metode dalam proses kompresi dan dekompresi tersebut akan semakin efisien (Hasan et al., 2021).

2.5 Algoritma Run Length Encoding

Algoritma *Run Length Encoding* adalah teknik untuk mengkompresi data yang berisi karakter berulang. Cara kerja Algoritma ini yaitu mencari karakter yang diulang tiga kali atau lebih dalam *file* gambar kemudian mengubahnya menjadi sebuah bit penanda dan bit yang memberi informasi tentang banyaknya karakter berulang serta diakhiri karakter yang dikompres. Deretan 8-bit yang membentuk karakter ASCII adalah bit penanda yang dimaksud dalam hal ini (Mansyuri, 2021).

2.6 Algoritma Fixed Length Binary Encoding

Pengkodean pada Algoritma *Fixed Length Binary Encoding* yaitu pada tiap karakter mempergunakan kode biner dengan karakter yang panjangnya ditentukan menurut banyaknya frekuensi setiap karakter. Frekuensi kode biner pada karakter yang sering muncul cenderung lebih kecil daripada frekuensi karakter yang jarang muncul. FLBE mengambil karakter dalam string untuk dikompresi dan menghitung banyaknya karakter yang muncul dalam string. FLBE dalam menghasilkan string bit baru yaitu dengan merubah bit karakter menjadi *fixed-length code*. Hasil dari kompresi FLBE adalah string bit tersebut (Pratiwi & Zebua, 2019).

3. HASIL DAN PEMBAHASAN

3.1 Pembahasan

Pembahasan pada penelitian ini terdapat beberapa tahapan yaitu, analisis data, representasi data, hasil analisis data, perancangan sistem, dan hasil pengujian.



3.1.1 Analisis Data

Data yang digunakan dalam penelitian ini adalah citra *grayscale* berekstensi *.jpg yang dikompresi menggunakan metode Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding*. Selama proses analisis data citra dipilih sesuai dengan ketentuan yang telah ditentukan. Setelah itu, metode Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* diterapkan pada sistem aplikasi. Berikut adalah langkah-langkah yang diperlukan untuk menjalankan sistem aplikasi:

- Memilih metode yang akan digunakan untuk melakukan kompresi.
- Menginput citra *grayscale* yang berformat *.jpg.
- Melakukan perhitungan parameter analisis kompresi yaitu *ratio of compression*, *ratio compression*, *Redudancy*, dan waktu.
- Tampilan berupa hasil hasil perhitungan dan hasil kompresi berbentuk *file text* yang telah dikompresi.

3.1.2 Representasi Data

Citra *grayscale* berekstensi *.jpg akan digunakan untuk melakukan pengujian, tetapi penulis hanya mengambil *sample* data pengujian berukuran 5 x 5 pixel untuk pengujian. *Sample* nilai intensitas citra *grayscale* berukuran 5 x 5 pixel ditunjukkan pada tabel 1.

Tabel 1. *Sample* Citra Berukuran 5 x 5

232	232	184	184	184
98	98	255	255	255
175	175	175	117	117
110	240	240	240	240
234	234	234	54	54

Sample citra ini terdiri dari lima baris dan lima kolom, dan memiliki nilai citra *grayscale* dengan rentan waktu antara 0-255. *Sample* tersebut akan digunakan untuk menerapkan Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding*.

3.1.3 Hasil Analisis Data

Pada proses implementasi penggunaan Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* untuk mengkompresi citra digital, digunakan *sample* citra *grayscale* 8 bit berekstensi *.jpg yang berukuran 5 x 5 *pixel*. Dari *sample* pada tabel 1, maka dilakukan proses kompresi dengan Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* sebagai berikut:

- Algoritma *Run Length Encoding*

Berikut adalah proses kompresi dengan Algoritma *Run Length Encoding* menggunakan data citra yang ada pada tabel 2.

Hasil kompresi: (232,2) (184,3) (98,2) (255,3) (175,3) (117,2) (110,1) (240,4) (234,3) (54,2)

Setelah selesai melakukan kompresi menggunakan metode *Run Length Encoding*, selanjutnya dilakukan pengkodean dari data tersebut.

Hasil pengkodean: 232 2 184 3 98 2 255 3 175 3 117 2 110 1 240 4 234 3 54 2

Total = 20 *pixel*

Berdasarkan data dari tabel 1 diatas, maka diperoleh:

Ukuran data sebelum dikompresi = 5 x 5 x 8 = 200 bit. Hasil tersebut didapatkan dari ukuran *sample* citra dikali dengan jumlah bit pada *sample* citra.

Ukuran data setelah dikompresi = 20 x 8 = 160 bit. Hasil tersebut didapatkan dari total hasil pengkodean dikali dengan jumlah bit pada *sample* citra.

Adapun perhitungan dari kinerja *Run Length Encoding* dari data pada tabel 1 setelah dikompresi adalah sebagai berikut:

- Ratio of Compression*

$$RC = \frac{\text{ukuran bit data sebelum dikompresi}}{\text{ukuran bit data setelah dikompresi}} = \frac{200}{160} = 1,25$$

Dapat disimpulkan bahwa nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi adalah 1,25 dari hasil perhitungan RC.

- Compression Ratio*

$$CR = \frac{\text{ukuran bit data setelah dikompresi}}{\text{ukuran bit data sebelum dikompresi}} \times 100\% = \frac{160}{200} \times 100\% = 80\%$$

Dapat disimpulkan bahwa persentase perbandingan antara bit data yang sudah dikompresi dengan bit data yang belum dikompresi adalah 80% dari hasil perhitungan CR.

- Redudancy (Rd)*



$$Rd = 100\% - Cr = 100\% - 80\% = 20\%$$

Dapat disimpulkan bahwa selisih antara ukuran data sebelum dan setelah dikompresi adalah 20% dari hasil perhitungan Rd.

b. Algoritma *Fixed Length Binary Encoding*

Berikut adalah proses kompresi dengan Algoritma *Fixed Length Binary Encoding* menggunakan data citra yang ada pada tabel 1. Ukuran string dari citra tersebut dapat dilihat pada tabel 2 berikut:

Tabel 2. Data Sebelum Dikompresi

Indeks Warna	Frekuensi	ASCII Binary	Bit	Bit x Frekuensi
54	2	00110110	8	16
98	2	01100010	8	16
110	1	01101110	8	8
117	2	01110101	8	16
175	3	10101111	8	24
184	3	10111000	8	24
232	2	11101000	8	16
234	3	11101010	8	24
240	4	11110000	8	32
255	3	11111111	8	24
Jumlah bit x frekuensi			200 bit	

Dari tabel 2 diatas, nilai Frekuensi didapatkan dari jumlah kemunculan nilai Indeks Warna yang terdapat pada tabel 1. ASCII Binary merupakan nilai biner dari tiap Indeks Warna dan nilai Bit didapatkan dari jumlah bit pada biner.

Tabel 3. Data Setelah Dikompresi Menggunakan Algoritma FLBE

Indeks Warna	Frekuensi	Kode FLBE	Bit	Bit x Frekuensi
240	4	00000	5	20
175	3	00001	5	15
184	3	00010	5	15
234	3	00011	5	15
255	3	00100	5	15
54	2	00101	5	10
98	2	00110	5	10
117	2	00111	5	10
232	2	01000	5	10
110	1	01001	5	5
Jumlah bit x frekuensi			125 bit	

Indeks Warna yang terdapat pada tabel 3 diurutkan dari jumlah frekuensi yang paling banyak muncul hingga yang paling sedikit. Kemudian Kode FLBE didapatkan berdasarkan ketentuan pada FLBE dan nilai Bit didapatkan dari jumlah bit pada Kode FLBE. Dari tabel 3 diatas dihasilkan string bit sebanyak 125 bit.

Sebelum didapatkan hasil string bit baru, dilakukan penambahan bit *padding* dan *flag* terlebih dahulu. Penambahan bit *padding* dan *flag* dilakukan berdasarkan sisa jumlah bit dibagi bilangan 8, sehingga menghasilkan sisa bagi sebanyak lima bit. Maka diperoleh *padding* 000 atau jika diubah ke bit menghasilkan 3 bit. Dengan penambahan *padding* sebanyak tiga bit, maka diperoleh *flag* 00000011 atau jika diubah ke bit menghasilkan 8 bit.

Jadi, total panjang bit keseluruhan setelah dilakukan penambahan bit adalah 125 bit + 3 bit + 8 bit = 136 bit. Sehingga perhitungan dari kinerja *Fixed Length Binary Encoding* adalah:

1. *Ratio of Compression*

$$RC = \frac{\text{ukuran bit data sebelum dikompresi}}{\text{ukuran bit data setelah dikompresi}} = \frac{200}{136} = 1,47$$

Dapat disimpulkan bahwa nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi adalah 1,47 dari hasil perhitungan RC.

2. *Compression Ratio*

$$CR = \frac{\text{ukuran bit data setelah dikompresi}}{\text{ukuran bit data sebelum dikompresi}} \times 100\% = \frac{136}{200} \times 100\% = 68\%$$

Dapat disimpulkan bahwa nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi adalah 68% dari hasil perhitungan RC.

3. *Redudancy (Rd)*

$$Rd = 100\% - Cr = 100\% - 68\% = 32\%$$

Dapat disimpulkan bahwa selisih antara ukuran data sebelum dan setelah dikompresi adalah 32% dari hasil perhitungan Rd.

Dari hasil perhitungan parameter pada Algoritma *Run Length Encoding* dan Algoritma *Fixed Length Binary Encoding* diatas, dapat dilihat perbandingannya pada tabel 4 berikut:

Tabel 4. Perbandingan Hasil Perhitungan Parameter

Algoritma	Parameter		
	RC	CR	Rd
<i>Run Length Encoding</i>	1,25	80%	20%
<i>Fixed Length Binary Encoding</i>	1,47	68%	32%

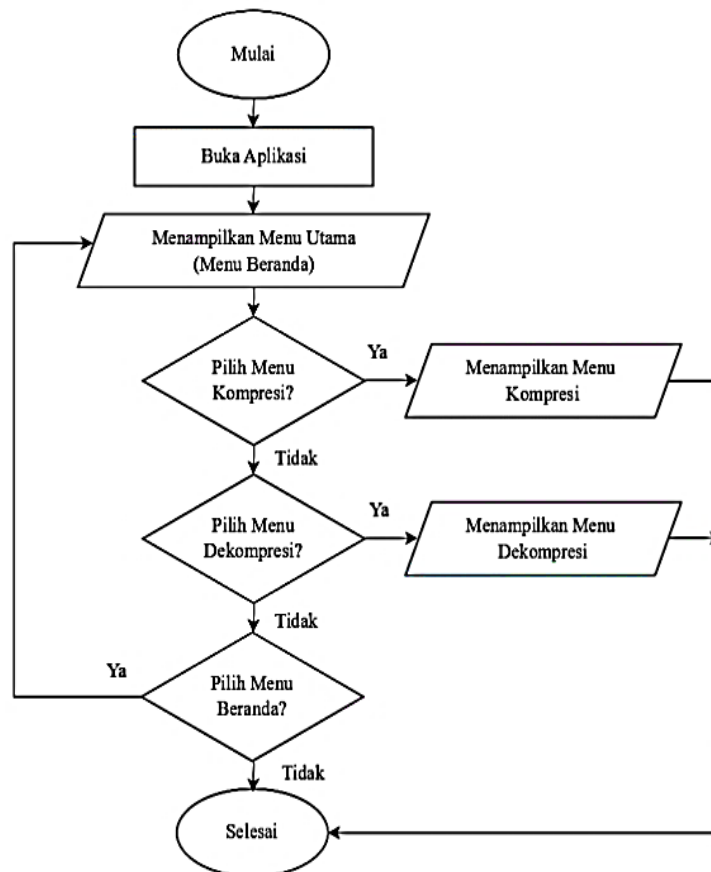
Dari tabel 4 dapat disimpulkan bahwa algoritma *Fixed Length Binary Encoding* lebih baik dibandingkan dengan algoritma *Run Length Encoding*. Karena nilai RC dan Rd lebih efektif jika hasilnya lebih besar, sedangkan CR lebih efektif jika hasilnya lebih kecil.

3.1.4 Perancangan Sistem

Berdasarkan analisis yang telah dilakukan terhadap algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding* untuk mengkompresi citra digital, maka perancangan yang dibangun adalah *Flowchart*. Untuk membuat algoritma yang terstruktur dan mudah dipahami oleh orang lain, dibutuhkan alat bantu dalam bentuk diagram alir.

a. *Flowchart* Beranda

Flowchart Beranda adalah alur proses saat pengguna berada pada menu beranda. Berikut adalah rancangan dari *Flowchart* beranda yang dapat dilihat pada gambar 2:



Gambar 2. *Flowchart* Beranda Pada Sistem

Berdasarkan gambar *flowchart* beranda, dapat dijelaskan bahwa langka pertama bagi pengguna yang membuka aplikasi akan dihadapkan dengan menu beranda yang berisi informasi tentang judul dan penulis dan terdapat dua menu lainnya yaitu menu kompresi dan menu dekompresi.

b. *Flowchart* Kompresi

Flowchart kompresi adalah alur proses saat pengguna melakukan kompresi terhadap citra *grayscale* yang berektensi *.jpg. Berikut adalah rancangan dari *Flowchart* kompresi yang dapat dilihat pada gambar 3:

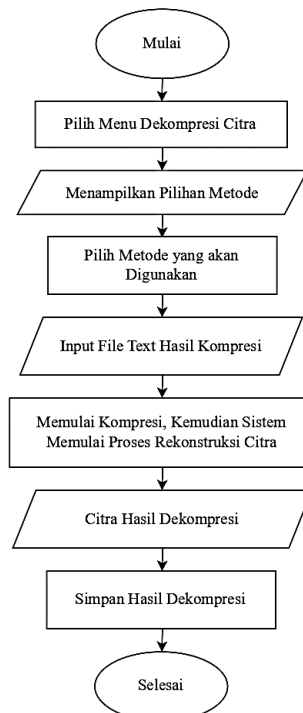


Gambar 3. Flowchart Kompresi Pada Sistem

Berdasarkan gambar *flowchart* kompresi, dapat dijelaskan bahwa pengguna akan melakukan pemilihan citra *grayscale* yang berekstensi *.jpg yang akan dikompresi kedalam sistem. Setelah itu, sistem akan membaca nilai *pixel* citra. Setelah nilai *pixel* dibaca, karakter dihitung, diuraikan, diurutkan berdasarkan frekuensi kemunculan. Kemudian sistem akan melakukan kompresi berdasarkan algoritma yang dipilih dan menghitung nilai parameternya. Setelah proses kompresi selesai, sistem akan menampilkan hasil kompresi.

c. *Flowchart* Dekompresi

Flowchart dekompresi adalah alur proses saat pengguna melakukan dekompresi terhadap citra telah dikompresi. Berikut adalah rancangan dari *Flowchart* dekompresi yang dapat dilihat pada gambar 4:

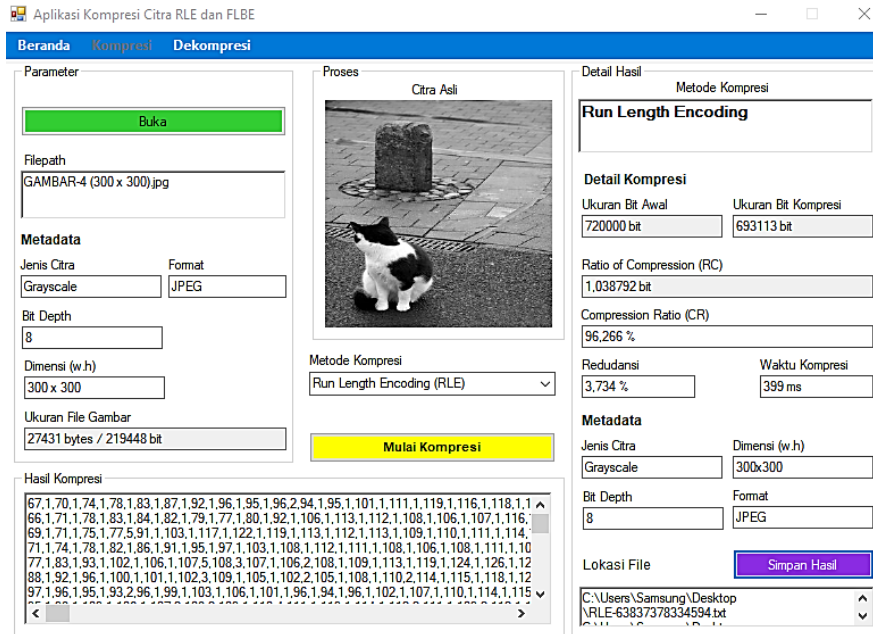


Gambar 4. Flowchart Dekompresi Pada Sistem

Berdasarkan gambar *flowchart* dekompresi, dapat dijelaskan bahwa pengguna akan melakukan pemilihan citra yang telah dikompresi. Setelah itu, sistem akan melakukan dekompresi sesuai dengan algoritma yang telah dipilih. Kemudian sistem akan menghitung waktu dekompresi. Setelah proses dekompresi selesai, sistem akan menampilkan hasil dekompresi.

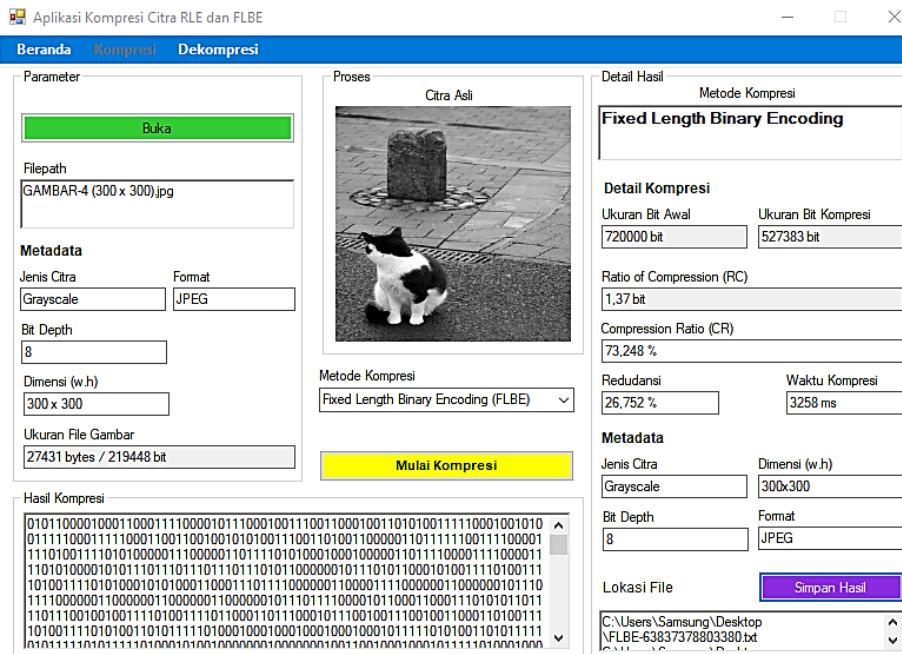
3.2 Hasil Pengujian

Uji coba dilakukan pada gambar berekstensi *.jpg dengan dimensi 50 x 50, 100 x 100, 200 x 200, 300 x 300, 400 x 400, 500 x 500 *pixel*. *Sample* gambar uji bisa dilihat dalam gambar 5, 6, 7, 8 dibawah.



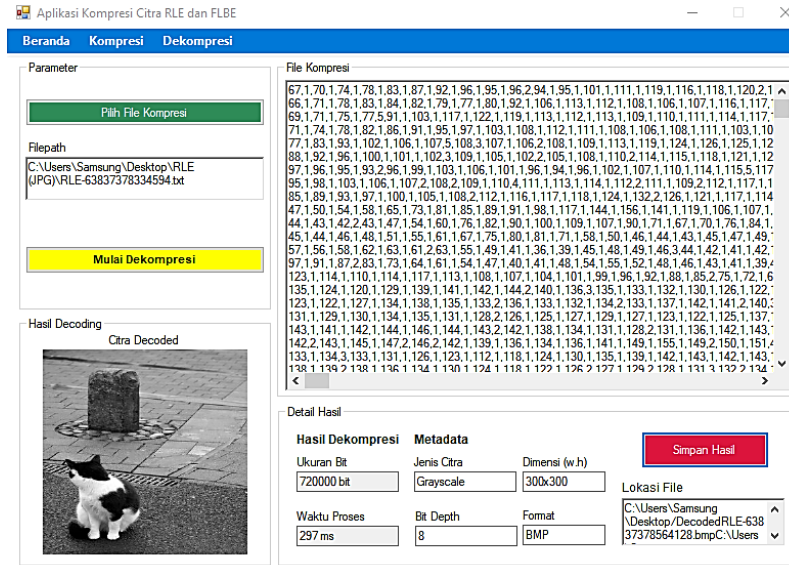
Gambar 5. Kompresi Citra RLE

Gambar 5 diatas merupakan salah satu hasil pengujian kompresi citra yang berdimensi 300 x 300 dengan algoritma *Run Length Encoding* yang dilakukan pada aplikasi, dari gambar tersebut terdapat hasil perhitungan dari parameter kompresi.



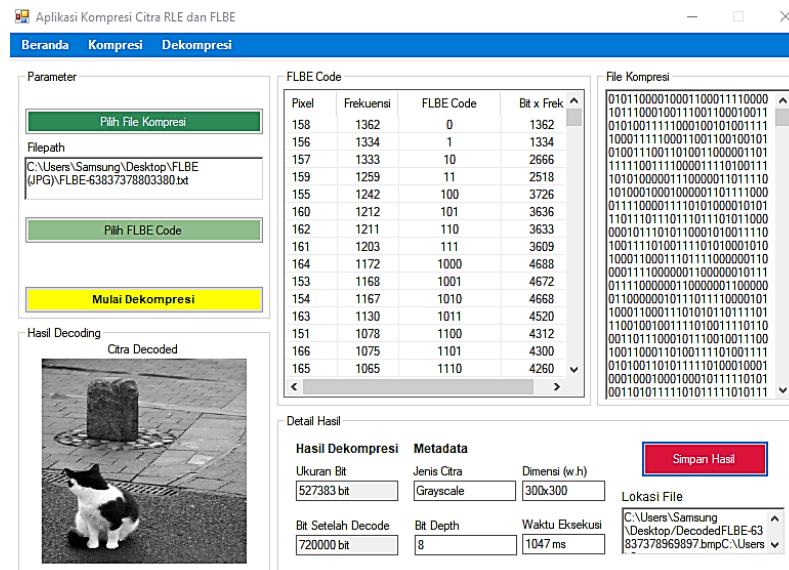
Gambar 6. Kompresi Citra FLBE

Gambar 6 diatas merupakan salah satu hasil pengujian kompresi citra yang berdimensi 300 x 300 dengan algoritma *Fixed Length Binary Encoding* yang dilakukan pada aplikasi, dari gambar tersebut terdapat hasil perhitungan dari parameter kompresi.



Gambar 7. Dekomposisi Citra RLE

Gambar 7 diatas merupakan salah satu hasil pengujian dekomposisi citra yang berdimensi 300 x 300 dengan algoritma *Run Length Encoding* yang dilakukan pada aplikasi.



Gambar 8. Dekomposisi Citra FLBE

Gambar 8 diatas merupakan salah satu hasil pengujian dekomposisi citra yang berdimensi 300 x 300 dengan algoritma *Fixed Length Binary Encoding* yang dilakukan pada aplikasi. Dibawah ini dipaparkan tabel hasil pengujian setiap citra uji yang dilakukan pada gambar berekstensi *.jpg dengan dimensi 50 x 50, 100 x 100, 200 x 200, 300 x 300, 400 x 400, 500 x 500 pixel.

Tabel 5. Hasil Pengujian Gambar Berdasarkan Rasio Kompresi dan Kompresi Rasio

Resolusi	Rc (<i>Ratio of Compression</i>)		Cr (<i>Compression Ratio</i>)	
	RLE	FLBE	RLE	FLBE
50 x 50	1.018071	1.47	98.225 %	68.005 %
100 x 100	1.015615	1.41	98.462 %	70.921 %
200 x 200	1.026411	1.37	97.427 %	72.738 %
300 x 300	1.038792	1.37	96.266 %	73.248 %
400 x 400	1.050791	1.36	95.166 %	73.6 %
500 x 500	1.061815	1.36	94.178 %	73.781 %

Tabel 5 diatas, merupakan hasil pengujian dari 6 citra *grayscale* berekstensi *.jpg yang memiliki resolusi berbeda-beda berdasarkan Rc (*Ratio of Compression*) dan Cr (*Compression Ratio*) pada algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding*.

Tabel 6. Hasil Pengujian Gambar Berdasarkan Ukuran

Resolusi	Ukuran Awal (bit)		Ukuran Kompresi (bit)	
	RLE	FLBE	RLE	FLBE
50 x 50	20000	20000	19645	13601
100 x 100	80000	80000	78770	56737
200 x 200	320000	320000	311766	232761
300 x 300	720000	720000	693113	527383
400 x 400	1280000	1280000	1218130	942076
500 x 500	2000000	2000000	1883568	1475625

Tabel 6 diatas, merupakan hasil pengujian dari 6 citra *grayscale* berekstensi *.jpg yang memiliki resolusi berbeda-beda berdasarkan Ukuran awal sebelum dikompresi dan Ukuran setelah dikompresi pada algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding*.

Tabel 7. Hasil Pengujian Gambar Berdasarkan Waktu Kompresi dan Dekompresi

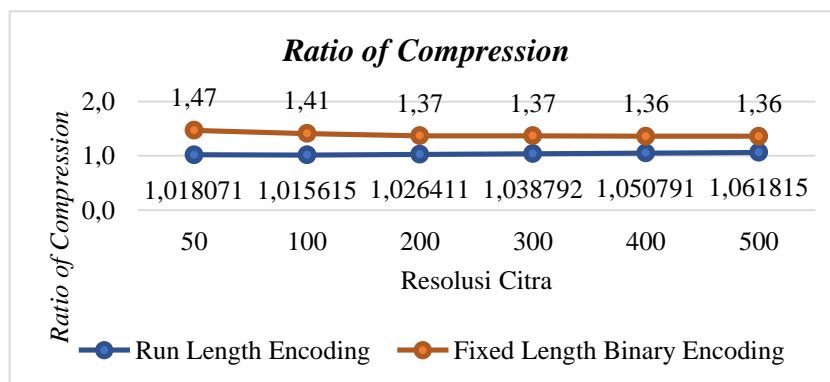
Resolusi	Waktu Kompresi (ms)		Waktu Dekompresi (ms)	
	RLE	FLBE	RLE	FLBE
50 x 50	9	85	27	26
100 x 100	45	339	34	117
200 x 200	179	1419	129	451
300 x 300	399	3258	297	1047
400 x 400	724	5850	533	1841
500 x 500	1108	8946	829	2876

Tabel 7 diatas, merupakan hasil pengujian dari 6 citra *grayscale* berekstensi *.jpg yang memiliki resolusi berbeda-beda berdasarkan Waktu kompresi dan Waktu Dekompresi pada algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding*.

Tabel 8. Hasil Pengujian Gambar Berdasarkan Redudansi Data

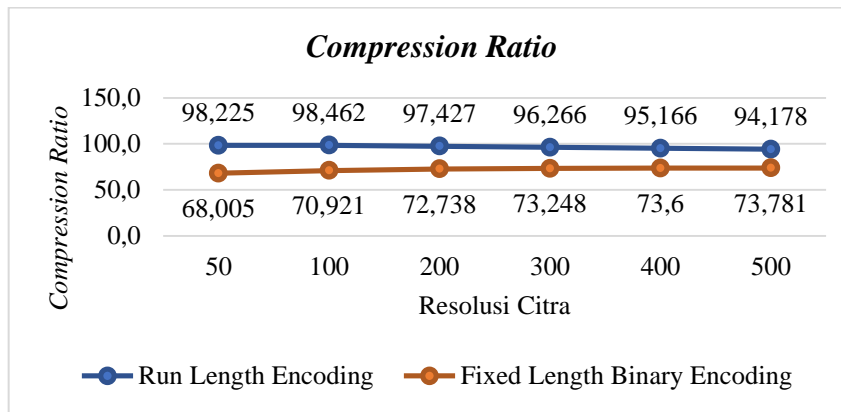
Resolusi	Redudansi Data	
	RLE	FLBE
50 x 50	1.775 %	31.995 %
100 x 100	1.538 %	29.079 %
200 x 200	2.573 %	27.262 %
300 x 300	3.734 %	26.752 %
400 x 400	4.834 %	26.4 %
500 x 500	5.822 %	26.219 %

Tabel 8 diatas, merupakan hasil pengujian dari 6 citra *grayscale* berekstensi *.jpg yang memiliki resolusi berbeda-beda berdasarkan Redudansi data pada algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding*. Dari tabel 5, 6, 7, 8 diatas dapat dibentuk grafik perbandingan hasil pengujian citra antara Algoritma *Run Length Encoding* atau Algoritma *Fixed Length Binary Encoding* dengan berdasar pada variabel *Compression Ratio* (CR), *Ratio of Compression* (RC), *Redudancy* (Rd), dan waktu kompresi dan dekompresi sebagaimana yang terlihat pada gambar 9, 10, 11, 12, 13 berikut.



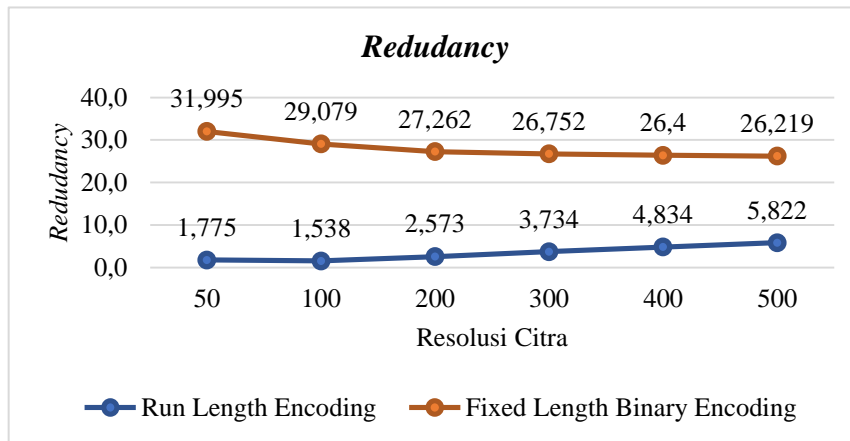
Gambar 9. Grafik *Ratio of Compression*

Berdasarkan grafik pada gambar 9, menunjukkan bahwa rasio kompresi pada algoritma *Fixed Length Binary Encoding* lebih besar dibanding algoritma *Run Length Encoding*. Semakin besar *Ratio of Compression* (RC) algoritma maka semakin efektif algoritma tersebut dalam melakukan kompresi citra.



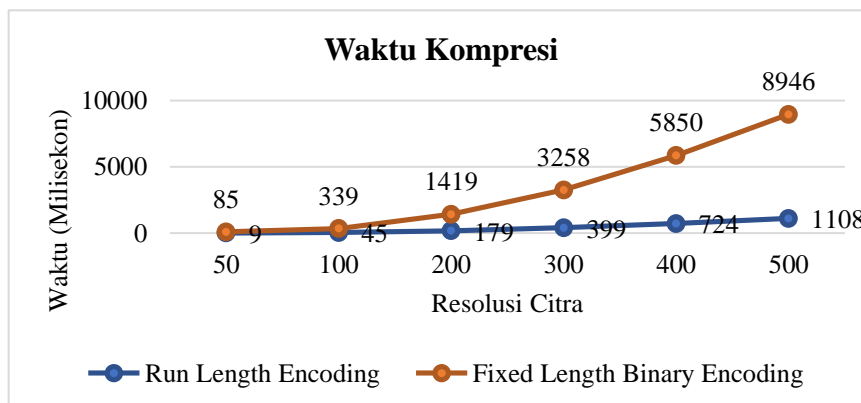
Gambar 10. Grafik *Compression Ratio*

Berdasarkan grafik pada gambar 10, bisa diketahui bahwa lebih efektif algoritma *Fixed Length Binary Encoding* dibandingkan algoritma *Run Length Encoding* karena memiliki nilai *Compression Ratio* (CR) yang lebih kecil sehingga algoritma tersebut lebih efektif untuk digunakan dalam mengompresi *file* citra dengan *pixel* yang besar.



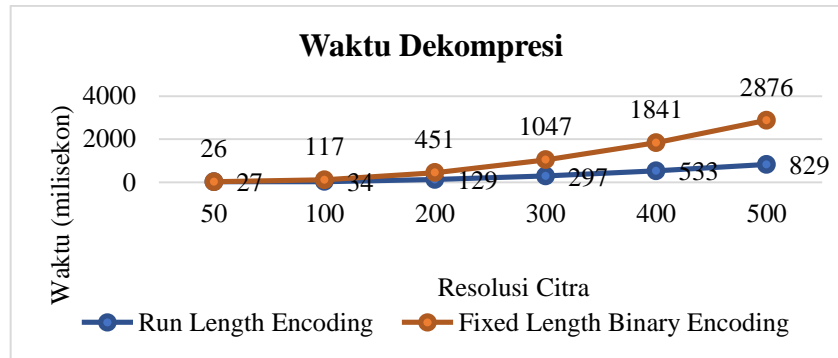
Gambar 11. Grafik *Redudancy*

Berdasarkan grafik pada gambar 11, bisa dilihat bahwa lebih efektif algoritma *Fixed Length Binary Encoding* dibandingkan algoritma *Run Length Encoding* karena memiliki nilai *Redudancy* (Rd) lebih besar sehingga algoritma tersebut lebih efektif untuk digunakan dalam mengompresi *file* citra dengan *pixel* yang besar.



Gambar 12. Grafik *Waktu Kompresi*

Berdasarkan grafik pada gambar 12, bisa dilihat bahwa waktu yang dibutuhkan pada algoritma *Run Length Encoding* relatif sedikit dalam mengompresi citra dibandingkan algoritma *Fixed Length Binary Encoding* yang memerlukan waktu lebih banyak. Waktu yang dibutuhkan untuk mengompresi citra bergantung pada besarnya ukuran resolusi pada citra. Semakin besar resolusi citra, maka semakin lama pula waktu diperlukan kedua algoritma untuk mengompresi citra.



Gambar 13. Grafik Waktu Dekompresi

Berdasarkan grafik gambar 13, bisa dilihat bahwa waktu yang dibutuhkan pada algoritma *Run Length Encoding* relatif sedikit dalam dekompresi citra dibanding algoritma *Fixed Length Binary Encoding* yang memerlukan waktu lebih banyak. Waktu yang dibutuhkan untuk melakukan dekompresi citra bergantung pada besarnya ukuran resolusi pada citra. Semakin besar resolusi citra, maka semakin lama pula waktu yang diperlukan kedua algoritma untuk melakukan proses dekompresi citra.

4. KESIMPULAN

Berdasarkan analisis yang telah dilakukan pada data citra *grayscale* berkstensi *.jpg untuk membandingkan hasil kinerja dari algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding* dalam melakukan kompresi dan dekompresi, dimana proses dalam melakukan kompresi dan dekompresi dengan menggunakan algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding* dilakukan dengan menggunakan 6 citra *grayscale* yang memiliki resolusi berbeda-beda. Dari hasil ujicoba perbandingan kompresi file citra antara algoritma *Run Length Encoding* dan algoritma *Fixed Length Binary Encoding* mempunyai hasil berbeda dalam membandingkan kinerjanya dengan menggunakan 4 kriteria yaitu *Ratio of Compression (RC)*, *Compression Ratio (CR)*, *Redundancy (Rd)* dan Waktu. Hasil perbandingan kompresi dari salah satu data citra yang beresolusi 300 x 300 pada algoritma *Run Length Encoding* memiliki *Ratio of Compression (RC)* 1.038792, *Compression Ratio (CR)* 96.266%, *Redundancy (Rd)* 3.734%, Waktu kompresi 399ms, dan Waktu dekompresi 297ms. Sedangkan pada algoritma *Fixed Length Binary Encoding* memiliki *Ratio of Compression (RC)* 1.37, *Compression Ratio (CR)* 73.248%, *Redundancy (Rd)* 26.752%, Waktu kompresi 3258ms, dan Waktu dekompresi 1047ms. Sehingga dari hasil tersebut dapat disimpulkan bahwa kinerja yang lebih baik dalam melakukan kompresi citra adalah algoritma *Fixed Length Binary Encoding* dibandingkan dengan *Run Length Encoding*. Karena jika nilai *RC* dan *Rd* pada algoritma lebih besar maka semakin efektif algoritma tersebut dalam mengkompresi citra, sedangkan untuk nilai *CR* dan waktu pada algoritma lebih efektif jika nilainya lebih kecil.

REFERENCES

- Fitriya, L. A., Purboyo, T. W., & Prasasti, A. L. (2017). *A Review of Data Compression Techniques*. 12(19), 8956–8963.
- Ginting, Z. A. (2017). *Implementasi Algoritma Rabin Dan Fixed Length Binary Encoding Dalam Pengamanan Dan Kompresi File Rtf*. Universitas Sumatera Utara.
- Hasan, I., Tommy, & Syahputri, N. I. (2021). *Analisis Parameter Kompresi Algoritma Elias Omega Code dan Fibonacci Code Pada File Digital*. 6341(April), 8–22.
- Mansyuri, U. (2021). *KOMPRESI DATA TEKS DENGAN METODE RUN LENGTH mengurangi jumlah data dalam teks . Contoh kompresi sederhana misalnya kata*. 1(2), 102–109.
- Nas, C., Ilham, W., & Syafrinal, I. (2019). *Analisis Algoritma Shannon-Fano Dalam Kompresi Data Pengajuan Proposal Skripsi Mahasiswa STMIK CIC Cirebon*. 8(2), 91–100.
- Pangesti, W. E., Widagdo, G., Riana, D., & Hadianti, S. (2020). *Implementasi Kompresi Citra Digital Dengan Membandingkan Metode Lossy Dan Lossless Compression Menggunakan Matlab*. *Jurnal Khatulistiwa Informatika*, 8(1), 53–58. <https://doi.org/10.31294/jki.v8i1.7759>
- Prabiantissa, C. N., Sulaksono, D. H., Yulastuti, G. E., & Nugroho, A. P. (2023). *Implementasi Algoritma Kompresi Lempel-Ziv-Welch pada Data Citra*. 321–325.
- Pratiwi, D., & Zebua, T. (2019). *ANALISIS PERBANDINGAN KINERJA ALGORITMA FIXED LENGTH BINARY ENCODING DAN ALGORITMA ELIAS GAMMA CODE DALAM*. 3, 424–430. <https://doi.org/10.30865/komik.v3i1.1623>
- Prayoga, E., & Suryaningrum, K. M. (2018). *IMPLEMENTASI ALGORITMA HUFFMAN DAN RUN LENGTH ENCODING PADA APLIKASI KOMPRESI BERBASIS WEB*. IV(2), 92–101.



- Pujianto, Mujito, Prasetyo, B. H., & Prabowo, D. (2020). Perbandingan Metode Huffman dan Run Length Encoding Pada Kompresi Document. *InfoTekJar: Jurnal Nasional Informatika Dan Teknologi Jaringan*, 5(1), 216–223.
- Putra, A. B. W., & Muhammad Trisna Aryun. (2021). Kompresi Citra Digital Dengan Basis Komponen Warna RGB Menggunakan Metode K-Means Clustering. *Jurnal Komputer Terapan*, 7(1), 14–23.
- Rumetna, M. S., Lina, T. N., & Santoso, A. B. (2020). Rancang Bangun Aplikasi Koperasi Simpan Pinjam Menggunakan Metode Research and Development. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 11(1), 119–128. <https://doi.org/10.24176/simet.v11i1.3731>
- Sandra, R. A. (2023). Implementasi Kombinasi Algoritma Tunstall Code Dan Boldi-Vigna Untuk Kompresi File Pdf. *Jurnal Ilmu Komputer, Teknologi Dan Informasi*, 1(2), 34–42.
- Sanjaya, A., & Aria, M. (2018). *Teknik Kompresi pada Transmisi Data Citra Payload KOMURINDO*. 7(2), 103–111. <https://doi.org/10.34010/komputika.v7i2.1512>
- Saragih, S. R., & Utomo, D. P. (2020). Penarapan Algoritma Prefix Code Dalam Kompresi Data Teks. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, 4(1), 249–252. <https://doi.org/10.30865/komik.v4i1.2691>
- Siahaan, S. (2024). Perbandingan Kompresi File Audio Menggunakan Algoritma Fibonacci Dan Algoritma Invert Elias Delta. *Jurnal Sistem Informasi, Teknik Informatika Dan Teknologi Pendidikan*, 3(2), 82–95. <https://doi.org/10.55338/justikpen.v3i2.89>
- Simanjuntak, L. V. (2020). Perbandingan Algoritma Elias Delta Code dengan Levenstein Untuk Kompresi File Teks. *Journal of Computer System and ...*, 1(3), 184–190.
- Simanjuntak, S. M. (2023). Analisis Perbandingan Kompresi File Audio Menggunakan Algoritma Shannon Fano Dengan Algoritma Fibonacci Code. *Jurnal Kajian Ilmiah Teknologi Informasi Dan Komputer*, 2(1), 1–10. <https://doi.org/10.62866/jutik.v2i1.110>
- Sunardi, H., Zulkifli, & Antony, F. (2021). Transformasi Geometri Rotasi Citra Digital Untuk Mendapatkan Kompresi Optimal Menggunakan Metode Lossless Dan Lossy. *Jurnal Ilmiah Informatika Global*, 12(1), 15–22. <https://doi.org/10.36982/jiig.v12i1.1540>
- Utomo, D. A., Kenedi, I., & Jumadi, J. (2021). *Perancangan Aplikasi Kompresi Menggunakan Metode Deflate*. 1(1), 212–219.