

# Perancangan Aplikasi Kompresi File Gambar Dengan Menggunakan Algoritma Stout Code

Ismail Marzuki Sianturi

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma,  
Medan, Indonesia

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

**Abstrak**– Dalam kompresi *file* gambar, diperoleh salah satu hal terpenting dan umum terjadi bahwa proses dekompresi selalu diperlukan agar kompresi dapat bekerja. Proses kompresi digunakan untuk memperkecil ukuran file, Sementara Proses dekompresi merupakan suatu cara untuk mengembalikan data yang terkompresi kembali menjadi data asli. Kompresi data dapat membantu mengurangi ukuran data yang dikirimkan melalui jaringan, seperti internet, yang dapat mempercepat waktu transfer data. Adapun solusi untuk memecahkan masalah yaitu melakukan kompresi. Algoritma yang digunakan untuk mengkompresi adalah Algoritma *Stout code*. Dengan menggunakan Algoritma ini maka hasil kompresi akan lebih baik dan bertujuan untuk mengurangi penyimpanan *file* gambar. dengan ukuran yang lebih kecil daripada ukuran aslinya sehingga lebih efisien dalam menyimpannya serta mempersingkat waktu pertukaran data tersebut. Dengan adanya kompresi diharapkan dapat menghemat biaya serta waktu yang dikeluarkan menambah fasilitas media penyimpanan data pada komputer serta mempercepat proses. Penelitian ini diharapkan bisa dimanfaatkan untuk pengurangan ukuran data suatu *file* gambar. Pada tahap awal yang akan dianalisa penulis ialah dengan mencari *file* gambar yang akan dikompres kemudian diubah menjadi nilai *hexadecimal* melalui aplikasi HxD. Setelah didapat nilai *hexadecimal file* gambar maka akan dilakukan proses kompresi dan diperoleh *file* hasil kompresi.

**Kata Kunci:** Kompresi, File, Gambar, Algoritma Stout Code

**Abstract**– In compressing image files, one of the most important and common things that happens is that the decompression process is always required for compression to work. The compression process is used to reduce the file size, while the decompression process is a way to return compressed data back to original data. Data compression can help reduce the size of data sent over a network, such as the internet, which can speed up data transfer time. The solution to solving the problem is compression. The algorithm used for compression is the Stout code algorithm. By using this algorithm, the compression results will be better and aim to reduce image file storage. with a smaller size than the original size so that it is more efficient in storing it and shortens the time for exchanging the data. With compression, it is hoped that it can save costs and time spent adding data storage media facilities on computers and speeding up processing. This research is expected to be used to reduce the data size of an image file. In the initial stage, the author will analyze it by looking for image files that will be compressed and then converted to hexadecimal values through the HxD application. After obtaining the hexadecimal value of the image file, the compression process will be carried out and the compressed file will be obtained.

**Keywords:** Compression, File, Picture, Algorithm Stout Code

## 1. PENDAHULUAN

Pada era saat ini kebutuhan akan informasi merupakan hal yang penting bagi masyarakat umum karena semakin banyak informasi yang disimpan secara digital dan ada peningkatan kebutuhan akan media penyimpanan data yang lebih besar. Oleh karena itu, diperlukan suatu mekanisme penyimpanan data alternatif agar dapat menyimpan data sebanyak-banyaknya dengan menggunakan media penyimpanan yang ada.

Dalam kompresi *file* gambar, diperoleh salah satu hal terpenting dan umum terjadi bahwa proses dekompresi selalu diperlukan agar kompresi dapat bekerja. Proses kompresi digunakan untuk memperkecil ukuran file, Sementara Proses dekompresi merupakan suatu cara untuk mengembalikan data yang terkompresi kembali menjadi data asli. Pemampatan data atau Kompresi merupakan metode untuk mengurangi jumlah data yang disimpan pada media penyimpanan, sehingga lebih mudah untuk dikelola. Kompresi data dapat membantu mengurangi ukuran data yang dikirimkan melalui jaringan, seperti internet, yang dapat mempercepat waktu transfer data.

Pada dasarnya teknik kompresi citra digunakan untuk mengetahui cara transmisi data dan penyimpanan data. Kompresi citra saat ini banyak diaplikasikan pada penginderaan jauh, penyiaran televisi. Ada beberapa cara untuk mengelompokkan data dan mengompresi file menggunakan teknik penyimpanan data yang berbeda. Salah satunya adalah LZ78 yaitu teknik kompresi file yang bisa digunakan melakukan kompresi pada sebuah citra[1]

File citra yang ukurannya lebih besar bisa menghabiskan waktu yang lebih lama pada proses pengiriman data misalnya kalau file gambar yang digunakan yaitu file yang berformat Bitmap (.bmp) yang ukurannya bisa membuat proses pengiriman menjadi lama, dan menggunakan ruang memori yang sangat dalam penyimpanannya. Maka untuk mengatasi masalah tersebut digunakanlah kompresi yang dilakukan untuk mempercepat proses penyimpanan dan pengambilan file gambar. Pada teknik kompresi ada dua metode utama yang dilakukan dalam kompresi data yaitu *Lossless compression* dan *Lossy Compression*. *Lossless compression* merupakan proses kompresi data dimana proses tersebut tidak menghilangkan data dan hasil kompresi tidak menurun dan data tersebut sama dengan aslinya. Sedangkan *Lossy compression* yaitu proses selama kompresi data terdapat kompresi yang mengakibatkan hasil dari data yang terkompresi tidak sama dengan data aslinya dan ada informasi yang hilang, namun masih bisa ditoleransi[2].

Berdasarkan penelitian yang dilakukan Surya Darma Nasution pada tahun 2016 tentang “Perancangan Kompresi File Teks Dengan Menerapkan Algoritma “GOLDBACH CODES” Disimpulkan dalam penelitian tersebut bahwa hasil file kompresi teks sebelum dan sesudah di kompresi setelah dibandingkan mencapai 50% rasio perbandingannya. Algoritma “GOLDBACH CODES” yaitu algoritma yang diasumsikan menggunakan teori Goldbach Conjecture yaitu Semua bilangan genap positif yang lebih besar dari 2 merupakan penjumlahan dari dua bilangan prima[3].

Berdasarkan penelitian yang dilakukan Agus Adi Pramadani dkk pada tahun 2019 menyimpulkan bahwa dengan teknik kompresi, data awal berukuran besar, sehingga dapat memperkecil ukuran data dan menghemat alokasi memori. Lebih mudah dan lebih cepat. Dengan menggunakan algoritma Even Rodeh Code, tersebut hasil kompresi sangat baik dalam melakukan transmisi data dan pemindahan alokasi file gambar lebih mudah dan cepat[4].

Berdasarkan penelitian yang dilakukan oleh Ali Sabandy Sinaga dan Efori Buulolo pada tahun 2020 tentang “Perancangan Aplikasi File Animasi Flash FLA dengan menggunakan Algoritma *Stout code*” disimpulkan bahwa hasil kompresi dari nilai k mempunyai hasil yang berbeda-beda dari setiap nilainya dan hasil kompresi akan menguntungkan dalam melakukan pengiriman dan pemindahan file animasi FLA akan semakin mudah[5].

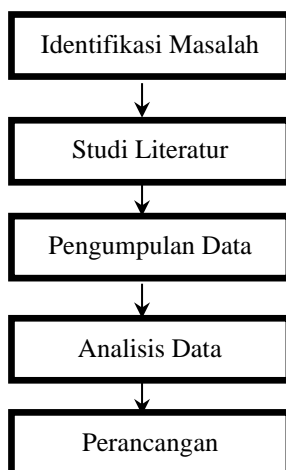
Berdasarkan penelitian yang dilakukan oleh Elvia Hariska pada tahun 2021 tentang “Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma *Additive Code*” menyimpulkan kompresi data dapat mengurangi ukuran penyimpanan data pada memori internal maupun eksternal, mempercepat pengiriman data dikarenakan ukuran penyimpanan menjadi lebih kecil[2].

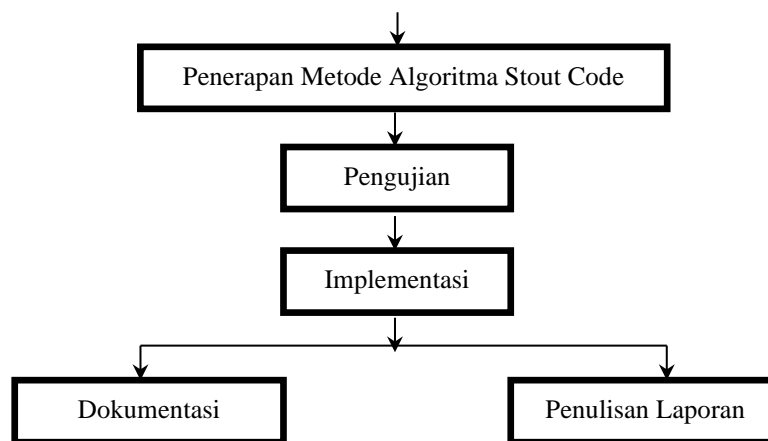
Dalam pembahasan ini proses hasil sebuah file gambar yang sudah terkompres, kemudian dalam pengembalian file gambar semula akan melalui proses dekompresi. Berdasarkan hal tersebut di atas maka penulis tertarik untuk merancang aplikasi file gambar. Untuk itulah penulis tertarik untuk mengajukan judul “Perancangan Aplikasi Kompresi File Gambar Dengan Menggunakan Algoritma *Stout Code*”

## 2. METODE PENELITIAN

### 2.1 Kerangka Kerja Penelitian

Pada penelitian Penilaian Kinerja Aparatur Desa merupakan suatu bentuk kerangka kerja yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Adapun kerangka kerja yang penulis lakukan dapat dilihat pada gambar 1, yaitu kerangka kerja penelitian.





Gambar 1. Kerangka Kerja Penelitian

Keterangan:

- a. Mengidentifikasi Masalah  
Pada tahap penelitian ini, yang dilakukan adalah mengidentifikasi masalah yang terjadi mengenai kompresi *file* gambar. Masalah yang ada terkait kapasitas untuk keamanan data.
- b. Studi Literatur  
Studi Literatur dilakukan untuk mencari data dan informasi yang berkaitan dengan kompresi *file* gambar dan metode stout code pada buku, jurnal dan lain-lain yang berkaitan dengan permasalahan yang ingin diselesaikan pada penelitian ini.
- c. Pengumpulan Data  
Pada tahapan ini dilakukan pengumpulan data yang akan digunakan dalam melakukan penelitian dengan cara wawancara. Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan merangkai, mencapai tujuan penelitian untuk mengkompresikan *file* gambar.
- d. Analisa Data  
Setelah dilakukan pengumpulan data, maka pada tahapan ini data tersebut akan dianalisa untuk mengetahui apakah data tersebut bisa digunakan untuk mengkompresikan *file* gambar.
- e. Penerapan Metode *Stout Code*  
Rancangan mengkompresikan *file* gambar untuk mengecilkan ukuran kapasitas *file* gambar yang akan di kompresikan dengan menerapkan metode *StoutCode*.
- f. Perancangan  
Setelah melakukan analisa dan penerapan metode, tahapan selanjutnya yaitu perancangan aplikasi menggunakan tools programman *Microsoft Visual BasicNET*.
- g. Pengujian  
Tahapan selanjutnya adalah melakukan pengujian aplikasi yang sudah selesai. Pengujian dilakukan untuk melihat kesesuaian hasil yang diperoleh aplikasi perancangan yang telah dibuat..
- h. Implementasi  
Pada tahapan ini, setelah dilakukan proses pengujian maka hasil dari pengujian tersebut kembali dianalisa untuk memastikan apakah hasil tersebut sesuai dengan tujuan dari penelitian ini dilakukan.
- i. Dokumentasi  
Tahapan selanjutnya adalah dokumentasi. Pada tujuan ini akan ditampilkan hasil dari penerapan metode stout code untuk mengkompresikan *file* pdf dan juga laporan penelitian dari penelitian yang dibuat.
- j. Penulisan Laporan  
Tahapan terakhir pada penelitian ini yaitu penulisan laporan penelitian. Pada tahapan ini akan dijabarkan proses dan masalah-masalah yang dihadapi dalam penelitian yang dibuat kedalam bentuk laporan. Penulisan laporan penelitian dilakukan sebagai bentuk tanggung jawab dari hasil penelitian yang dilakukan untuk mendokumentasikan seluruh kegiatan penelitian dalam bentuk skripsi yang nantinya juga dibuat dalam bentuk artikel ilmiah yang akan dipublikasikan.

## 2.2 Perancangan

Perancangan adalah langkah awal pada fase pengembangan rekayasa *system*. Perancangan yaitu proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan secara rinci perangkat, proses atau system yang memungkinkan realisasi fisiknya, satu proses atau satu system secara detail yang membolehkan dilakukan realisasi fisik. Fase ini adalah inti teknis dari proses rekayasa perangkat lunak. Pada fase ini elemen-elemen dari model analisa dikonversikan. Dengan menggunakan satu dari sejumlah metode perancangan data, perancangan antarmuka, perancangan arsitektur dan perancangan prosedur.

Banyak langkah yang perlu dilakukan dalam perancangan perangkat lunak. Langkah-langkah tersebut menggambarkan struktur data, struktur program, karakteristik antarmuka dan detail prosedur yang merupakan sintesa dari keperluan-keperluan informasi. Perancangan data adalah langkah pertama dari empat kegiatan perancang andalan rekayasa perangkat lunak. Aktivitas utamadalam perancangan data adalah memilih gambaran logik dari stuktur data yang dikenali selama fase spesifikasi dan pendefinisian keperluan. Pemilihan ini melibatkan analisis algoritma dari alternative struktur dalam rangka menentukan perancangan yang paling efisien[6].

### 2.3 Aplikasi

Aplikasi yaitu sekumpulan elemen yang saling berinteraksi dan salingberhubungan untuk akyivitas bersama-sama utuk mencapai tujuan tertentu. Aplikasi pada dasarnya adalah hasil suatu proses pengembangan yang dilakukan dengan teknologi terkeni tersedia perangkat lunak untuk mempermudah proses pengelolaan maupun pemantauan suatu kegiatan. Perangkat lunak sering dibagi menjadi perangkat lunak aplikasi dan perangkat lunak sistem. Perangkat lunak yang dirancang untuk membantu pengguna melakukan pegerjaan tugas sehari- hari dinamakan perangkat lunak aplikasi[7].

### 2.4 File Gambar

*File Gambar* adalah deskripsi, masuk akal, atau tiruan suatu objek. Gambarsebagai keluaran atau sistem perekaman data dapat berupa optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat disimpan pada suatu media penyimpanan. *File Gambar* dapat digabungkan menjadi dua bagian yaitu citra diam dan citra bergerak. Citra dapat diartikan sebagai citra tunggal yang tidak bergerak sedangkan citra bergerak merupakan suatu rangkaian citra diam yang ditampilkan dengan beruntun sehingga memberi kesan kepada mata seagai gambar yang bergerak. Dalam setiap citra terdapat rangkaian yaitu disebut *Frame*. Citra dalam format *BMP* ada tiga macam: citra biner, citra berwarna, dan citra hitam-putih (*grayscale*). [8]

### 2.5 Kompresi File Gambar

Kompresi adalah mengubah data dalam bentuk kumpulan karakter ke dalam format yang dikodekan untuk menghemat kebutuhan penyimpanan dan waktu transmisi data. Kompresi data sangat penting karena mengurangi kebutuhan penyimpanan data., mempercepat pengiriman data, memperkecil kebutuhan *bandwith*. Terdapat banyak metode untuk mengkompresi data. Metode-metode tersebut lahir dari ide yang berbeda-beda. Namun, prinsip dasar yang menjadi dasar tiap metode adalah sama, yaitu mengkompresi data dengan menghilangkan *redundancy* dari data asli.

Pada suatu teknik yang digunakan dalam proses kompresi data terdapat beberapa faktor atau variabel yang biasa yang digunakan untuk menganalisa kuliatas dari suatu teknik kompresi data tersebut yaitu:

1. *Ratio of Compression (Rc)*

*Ratio of Compression (Rc)* adalah nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi

Rumus :

$$Rc = \frac{\text{Jumlah bit sebelum dikompresi}}{\text{Jumlah bit sesudah dikompresi}} \times 100\% \quad \dots\dots\dots(1)$$

2. *Compression of Ratio (Cr)*

*Compression of Ratio (Cr)* adalah persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.

Rumus :

$$Cr = \frac{\text{Jumlah bit sudah dikompresi}}{\text{Jumlah bit sebelum dikompresi}} \times 100\% \quad \dots\dots\dots(2)$$

3. *Space saving (Ss)*

*Space saving (Ss)* adalah selisih antara data yang belum dikompresi dengan besar data yang dikompresi

Rumus :  
 $S_s = 100\% - C_r \dots\dots\dots(3)$

4. *Time Compression*

*Time Compression* adalah waktu yang dibutuhkan untuk melakukan proses kompresi dan dekompresi. Semakin kecil waktu yang diperoleh maka semakin efisien metode yang digunakan dalam proses kompresi dan dekompresi itu.

**2.6 Algoritma Stout Code**

Dalam *Stout Code*, panjang variabel untuk kode bilangan bulat mirip dengan kode *Elias Omega* dan *Even-Rodeh Code*. Codeword yang dihasilkan oleh algoritma *Stout Code* bergantung pada pilihan parameter  $l$  yang lebih besar atau sama dengan 2. Dalam artikel singkatnya *Quentin Stout* memperkenalkan dua keluarga rekursi RI dan SI, kode panjang variable integer. Stout code bersifat *universal* dan asimtotik sebelumnya membaca pembaca panjang  $L$  Bilangan bulat  $n$  diberikan oleh  $L = 1 + \log_2 n$  [9]. Dan dilambangkan dengan representasi biner  $B(n, l)$  dari  $l$  - bit (kode beta) dari bilangan bulat  $n$ . Contohnya,  $B(12, 5) = 01100$ . Untuk  $l \geq 2$ , awalan dapat diartikan dengan.

$$R_l(n) = \begin{cases} B(n, l), & \text{for } 0 \leq n \leq 2^l - 1, \\ R_l(L)B(n, L), & \text{for } n \geq 2^l. \end{cases} \dots\dots\dots(4)$$

Kedua dari *Stout Code* memiliki metode yang serupa, tetapi dengan awalan berbeda yang dilambangkan dengan  $SI(n)$ . Untuk nilai kecil  $l$ , kelompok ini menawarkan beberapa peningkatan dibandingkan dengan kode RI. Khususnyaini menghilangkan sedikit redundansi dalam kode RI karena panjang kelompok tidak boleh 0. Awalan SI mirip dengan awalan RI dengan perbedaannya bahwa panjang kelompok untuk  $L$  mengkodekan  $L - 1 - l$ . Awalan  $SI(n)$  didefinisikan secara rekursif oleh :

$$S_l(n) = \begin{cases} B(n, l), & \text{for } 0 \leq n \leq 2^l - 1, \\ R_l(L - 1 - l)B(n, L), & \text{for } n \geq 2^l. \end{cases} \dots\dots\dots(5)$$

Dalam membangun codeword dari algoritma Stout Code pertama kali menentukan nilai  $l$ , pada penelitian ini menggunakan nilai  $l = 2$ . Kemudian setelah menentukan nilai  $l$ , maka masukan rumus sebagai berikut :

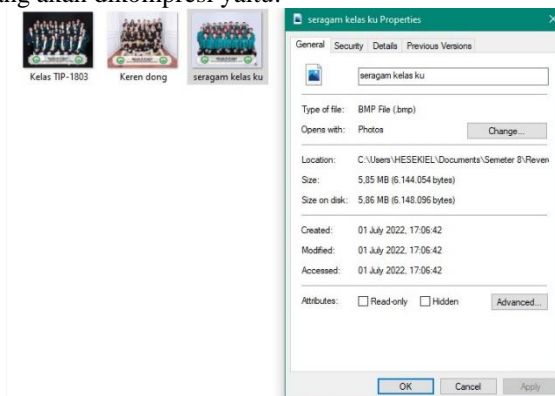
1. Untuk nilai  $0 = n = 2^l - 1$   
 Karena nilai  $l = 2$ , maka  $0 = n = 2^2 - 1 \square 0 = n = 3$ 
  - a.  $n = 1$ , maka codeword yang dihasilkan merupakan nilai biner dari  $n$  adalah 01 dan diambil sebanyak nilai  $l$  adalah 2 sehingga codeword yang dihasilkan adalah : 01
  - b.  $n = 2$ , maka codeword yang dihasilkan merupakan nilai biner dari  $n$  adalah 10 dan diambil sebanyak nilai  $l$  adalah 2 sehingga codeword yang dihasilkan adalah : 10
  - c.  $n = 3$ , maka codeword yang dihasilkan merupakan nilai biner dari  $n$  adalah 11 dan diambil sebanyak nilai  $l$  adalah 2 sehingga codeword yang dihasilkan adalah : 11
2. Untuk nilai  $n = 2^l$   
 Karena nilai  $l = 2$ , maka  $n = 2^2 \square n = 4$  :
  - a.  $n = 4$ , dengan nilai biner 100, maka nilai  $L = 3$  diambil dari panjang bit dari nilai biner  $n$  dan langkah berikutnya mencari nilai R.  
 $R_2(3-1-2) = 0$ , nilai binernya adalah 0 dan diambil sebanyak nilai  $l$ , yaitu 2 sehingga nilai R = 00  
 $R_1(L-1-l) B(n, L) = 00 100$
  - b.  $n = 5$ , dengan nilai biner 101, maka nilai  $L = 3$   
 $R_2(3-1-2) = 0$ , nilai binernya adalah 0 dan diambil sebanyak nilai  $l$ , yaitu 2 sehingga nilai R = 00  
 $R_1(L-1-l) B(n, L) = 00 101$
  - c.  $n = 6$ , dengan nilai biner 110, maka nilai  $L = 3$   
 $R_2(3-1-2) = 0$ , nilai binernya adalah 0 dan diambil sebanyak nilai  $l$ , yaitu 2 sehingga nilai R = 00  
 $R_1(L-1-l) B(n, L) = 00 110$
  - d.  $n = 7$ , dengan nilai biner 111, maka nilai  $L = 3$   
 $R_2(3-1-2) = 0$ , nilai binernya adalah 0 dan diambil sebanyak nilai  $l$ , yaitu 2 sehingga nilai R = 00  
 $R_1(L-1-l) B(n, L) = 00 111$

- e.  $n = 8$ , dengan nilai biner 1000, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1000$
- f.  $n = 9$ , dengan nilai biner 1001, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1001$
- g.  $n = 10$ , dengan nilai biner 1010, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1010$
- h.  $n = 11$ , dengan nilai biner 1011, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1011$
- i.  $n = 12$ , dengan nilai biner 1100, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1100$
- j.  $n = 13$ , dengan nilai biner 1101, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1101$
- k.  $n = 14$ , dengan nilai biner 1110, maka nilai  $L = 4$   
 $R_2(4-1-2) = 1$ , nilai binernya adalah 01 dan diambil sebanyak nilai 1, yaitu2 sehingga nilai  $R = 01$   
 $R_1(L-1-l) B(n, L) = 01\ 1110$

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Analisa Dalam Penetapan Algoritma Stout Code

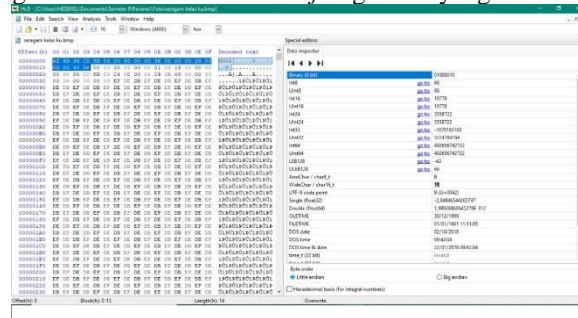
Dalam analisa perhitungan ini merupakan komputasi dan perancangan perangkat lunak kompresi *file* gambar yang penulis gunakan dengan algoritma stout code. Algoritma stout code adalah metode kompresi *lossless*, yang dapat memperkecil ukuran volume suatu data berdasarkan karakter yang dikandungnya. Penelitian ini dilakukan pengkompresian dengan mengkompresi file berformat .bmp yang biasanya relative besar Oleh karena itu hasil penelitian ini bisa diharapkan membantu mengurangi ukuran data suatu *file* gambar. Contoh *file* gambar yang akan terkompresi adalah file yang berformat .bmp melalui proses kompresi. Pada proses awal analisa yang dilakukan penulis adalah dengan mencari *file* gambar yang akan dikompres kemudian diubah menjadi nilai *hexadesimal* melalui aplikasi *HxD*, setelah didapat nilai *hexadesimal file* gambar akan di lakukan proses kompresi dan diperoleh *file* hasil kompresi. Berikut merupakan sampel data yang akan dikompresi yaitu:



Gambar 2. Sampel Data

**3.2 Penerapan Algoritma Stout Code**

Berdasarkan analisa, hasil kompresi file gambar dalam ukuran yang lebih daripada yag lain. Mengkompresi File gambar memanfaatkan file besar menjadi ukuran yang lebih kecil dan mengurangi alokasi memori. Penelitian ini,menjelaskan dua proses utama kompresi dan dekompresi, dan penelitian ini penulis menggunakan Algoritma Stout code untuk mengkompresi file gambar.Sebelum mengkompresi file, terlebih dahulu dilakukan memindai gambar untuk mendapatkan nilai *hexadesimal*. Mencari*hexadesimal* yang terdapat pada.bmp menggunakan aplikasi HxD untuk menemukan nilai *hexadesimal* pada aplikasi *file* gambar.Sebelum file gambar di kompres, langkah pertama yaitu mencari *file* gambar untukdi kompresi dengan menggunakan Algoritma stout code. Berikut *file* gambar yang akan dikompresi.



**Gambar 3.** Nilai *Hexadesimal* Sampel File Gambar

Berdasarkan gambar nilai *hexadecimal* diatas, maka penulis megambil 20sampel nilai *hexadesimal* untuk proses pengompresian *Hexadesimal*

**Tabel 1.** Tampilan Nilai Bilangan Hexadesimal

42	4D	36	C0	5D	00	00	00	00	00
36	00	00	00	28	00	00	00	40	06

Dari tabel diatas, Hasil bilangan *Hexadeimal* file gambar untuk di hitung secara manual. Berikut nilai *hexadesimal* diurutkan berdasarkan nilai frekuensi yang paling banyak maka berada di urutan pertama.

**Tabel 2.** Nilai *Hexadesimal* Yang Belum Dikompresi

n	Nilai Hexadesimal	Nilai Biner	Bit	Freq	Bit x Freq
1	00	00000000	8	11	88
2	36	01000010	8	2	8
3	42	01001101	8	1	8
4	4D	00110110	8	1	16
5	C0	11000000	8	1	8
6	5D	01011101	8	1	8
7	28	00101000	8	1	8
8	40	01000000	8	1	8
9	06	00000110	8	1	8
<b>Total Bit</b>					<b>160</b>

Setelah megurutkan bilangan *hexadesimal* dan mendapatkan nilai biner berdasarkan frekuensi kemunculannya, langkah selanjutnya menghitung *bit* dan megurutkan kode kode stout code untuk mendapatkan bit file terkompresi. Proses kompresi *file* gambar dapat ditunjukkan pada tabel 4.3 dibawah ini.

**Tabel 3.** Pengurutan kode Stout

N	Hexadesimal	Codeword		Freq	Bit x Freq
		S2 (n)	Bit		

1	00	01	2	11	22
2	36	10	2	2	4
3	42	11	2	1	2
4	4D	00100	5	1	5
5	C0	00101	5	1	5
6	5D	00110	5	1	5
7	28	00111	5	1	5
8	40	011000	6	1	6
9	06	011001	6	1	6
Total Bit x Freq					60

Setelah kode-kode diurutkan berdasarkan perhitungan stout code, langkah selanjutnya adalah menyusun kembali urutan *bit* yang dihasilkan dari proses kompresi sesuai dengan posisi *Hexadecimal*.

**Tabel 4.** String Bit Hasil Kompresi

42	4D	36	C0	5D
11	00100	10	00101	00110
<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>00</b>
01	01	01	01	01
<b>36</b>	<b>00</b>	<b>00</b>	<b>00</b>	<b>28</b>
10	01	01	01	00111
<b>00</b>	<b>00</b>	<b>00</b>	<b>40</b>	<b>06</b>
01	01	01	011000	011001

**Tabel 5.** String Bit Hasil Kompresi

11	00100	10	00101	00110
01	01	01	01	01
10	01	01	01	00111
01	01	01	011000	011001

Kemudian dari hasil total *bit* x frekuensi yang bernilai 60, perlu dilakukan penambaha jumlah *bit* karena 60 tidak habis dibagi 8. Dimana didalam komputer, suatu karakter direpresentasikan oleh bilangan ASCII (*American Standard Code For Information Interchange*), dimana sebanyak 8 *bit* dari bilangan biner. Apabila *bit* suatu data lebih pendek dari 8 *bit* maka tidak akan bisa dibaca oleh komputer sebelum dilakukannya proses *encoding*. Proses *encoding*, *bit-bit* data tersebut di-*encode* setiap *bit* nya sehingga membentuk suatu karakter yang dapat dibaca oleh komputer. Dikarenakan 60 tidak habis di bagi 8 atau bukan kelipatan 8 maka dibentuk variabel yang disebut dengan *padding* dan *flagging* untuk penambahan *bit* data. *Padding* yaitu penambahan *bit* data yang telah dikompres agar seluruh jumlah *bit* data tersebut kelipatan 8. Rumus *padding* ialah  $7 - n + "1"$  sedangkan *flagging* merupakan penambahan 8 *bit* bilangan biner setelah *padding* yang dimana untuk mempercepat dalam membaca *bit-bit* hasil kompresi saat proses dekompresi Rumus:

*flagging* ialah  $9 - n$ . Jika jumlah *bit* habis dibagi 8 atau kelipatan 8, tidak perlu dilakukan *padding*, tapi tetap harus menambahkan *flagging*.

**Tabel 6.** Penambahan Padding dan Flagging

Padding	Flagging
$60 \text{ mod } 8 = 4$	$9 - n$
$7 - n + 1$	$9 - 4 = 5 = 00000101$
$7 - 4 = + "1"$	

Adapun nilai *hexadecimal* yang sudah terkompresi dapat dilihat dibawah ini.

**Tabel 7.** Hasil Karakter

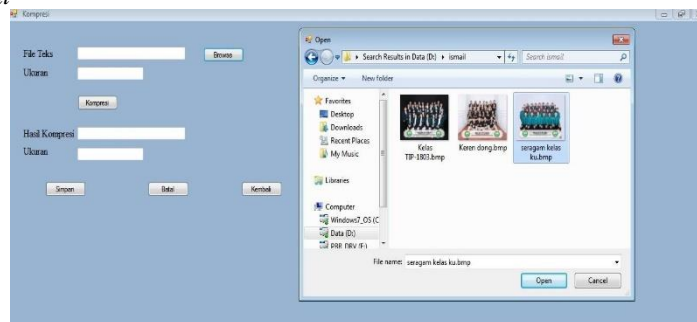


No	Codeword Hasil Kompresi	Decimal	Hexadecimal	Karakter	Keterangan
1	11001001	201	C9	É	Huruf Kapital Latin Dengan Akut
2	00010100	40	28	(	Ornate TANDA Kurung Kiri
3	11001010	202	CA	Ê	Hufur Besar Latin Dengan Lingkaran
4	10101100	172	AC	¬	Tidak Tanda-Tanda
5	10101001	169	A9	©	Tanda HakCipta
6	11010101	213	D5	Õ	Hufuf kecil Latino degan Tilde
7	01100001	97	61	a	Surat Kecil Lati a
8	10010001	145	91	´	Menggabungkan Koma diatas
9	00000101	5	5		ENQ ( <i>Enquiry</i> = tidak terlihat)

### 3.2 Hasil Pengujian

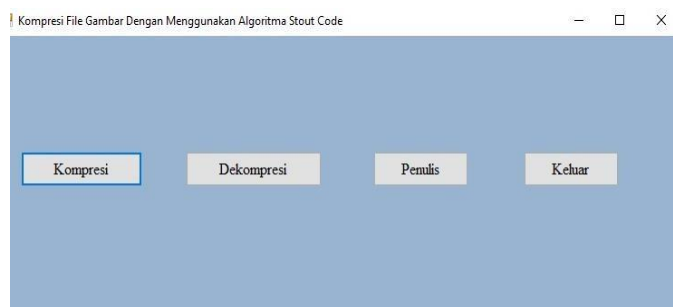
Pengujian sistem dilakukan mengetahui bagaimana sistem yang sudah dibangun berjalan. Sistem ini dirancang menjadi sesederhana mungkin agar pengguna dapat dengan mudah megaksesnya. Dalam hal ini, penulis memberikan *printout* saat sistem sedang berjalan.

#### a. Form Input



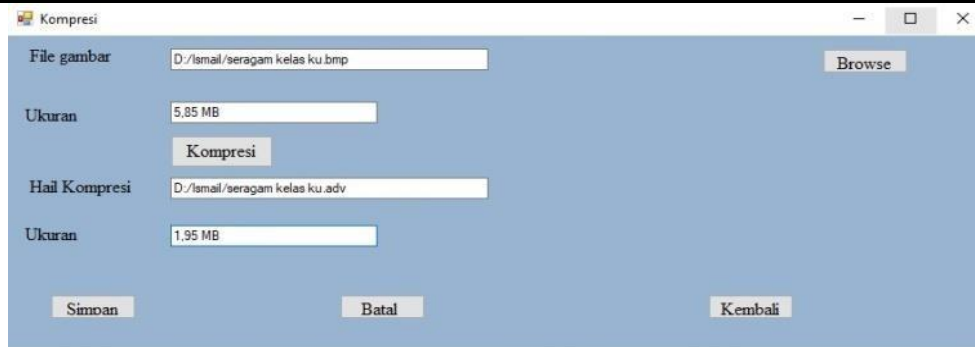
**Gambar 4.** Halaman *input*

#### b. Halaman Menu Utama



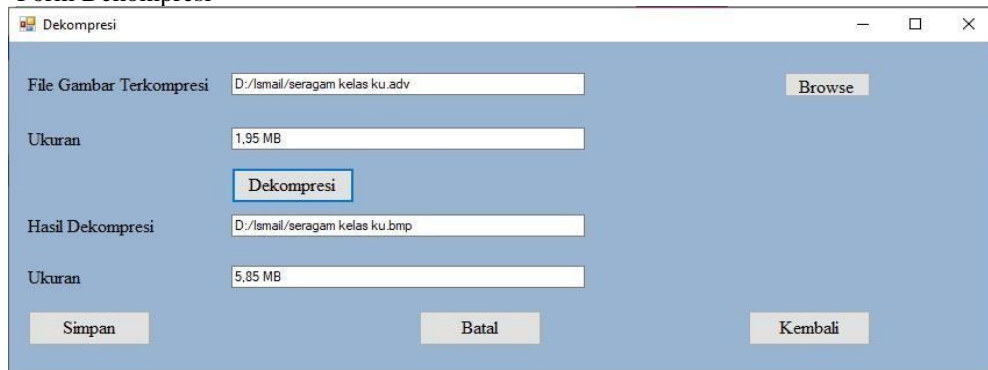
**Gambar 5.** Halaman *Home*

#### c. Halaman Form Kompresi



Gambar 6. Halaman Form Kompresi

d. Form Dekompresi



Gambar 7. Form Dekompresi

## 4. KESIMPULAN

Berdasarkan penelitian yang dilakukan penulis mendapatkan hasil akhir yang disimpulkan menjadi beberapa kesimpulan yaitu sebagai berikut:

1. Setelah melakukan prosedur kompresi menggunakan Algoritma Stout code menghasilkan bahwa *file* gambar yang memiliki ukuran yang cukup besar bisa dikompres menjadi *file* gambar yang baru dengan ukuran yang lebih kecil.
2. Sesudah menerapkan Algoritma Stout code untuk mengompresi *file* gambar telah membuktikan bahwa *file* gambar tersebut berhasil dikompresi, menghasilkan *Ration of Compression* (RC) =2,666 *Compression Ration* (CR) =37,5% dan *Redudancy* ( Rd) =62,5%
3. Aplikasi kompresi *file* gambar dengan menerapkan Algoritma Stout code yaitu menggunakan Microsoft Visual Studio 2008 dapat berjalan dengan baik.

## REFERENCES

- [1] A. F. Siregar, "Perancangan Aplikasi Kompresi File Citra Usg Menggunakan Algoritma Lz78," *J. Pelita Inform.*, vol. 17, no. April, pp. 164–167, 2018.
- [2] E. Hariska *et al.*, "Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma Additive Code," vol. 5, pp. 193–202, 2021, doi: 10.30865/komik.v5i1.3671.
- [3] S. D. Nasution, "PERANCANGAN APLIKASI KOMPRESI FILE TEKSDENGAN MENERAPKAN," no. May, 2017.
- [4] A. A. Pramadi, S. D. Nasution, and B. Purba, "Penerapan Algoritma Even-Rodeh Pada Aplikasi Kompresi File Gambar," *KOMIK (Konferensi Nas. Teknol. Inf. dan Komputer)*, vol. 3, no. 1, pp. 73–84, 2019, doi: 10.30865/komik.v3i1.1570.
- [5] A. S. Sinaga and E. Buulolo, "Perancangan Aplikasi Kompresi File Animasi FlashFLA Dengan Menggunakan Algoritma Stout Codes," *KOMIK (Konferensi Nas....)*, vol. 4, pp. 116–120, 2020, doi: 10.30865/komik.v4i1.2650.
- [6] J. Abdillah, "Pengembangan Sistem Simpan Pinjam Pada Koperasi Guru Sekolah ( Kgs ) Pengembangan Sistem Simpan Pinjam Pada Koperasi Guru Sekolah ( Kgs)," 2011.
- [7] D. A. Megawaty and M. E. Putra, "Aplikasi Monitoring Aktivitas Akademik Mahasiswa Program Studi Informatika Universitas Xyz Berbasis Android," *J.Inform. dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp.

- 65–74, 2020, doi: 10.33365/jatika.v1i1.177.
- [8] C. T. Utari, “Implementasi Algoritma Run Length Encoding Untuk Perancangan Aplikasi Kompresi Dan Dekompresi File Citra,” *J. TIMES*, vol. V, no. 2, pp. 24–31, 2016.
- [9] A. Apijuddin Kompresi and A. S. Codes, “Perancangan Aplikasi Kompresi FileTeks Menggunakan Algoritma Stout Codes,” vol. 9, pp. 183–188, 2021.
- [10] H. S. Purba, “Implementasi Algoritma Stout Code untuk Kompresi Record Database pada Website Dinas Pariwisata dan Kebudayaan Kabupaten Dairi,” vol.10, no. April, pp. 129–133, 2022.
- [11] M. Yogie, “Penerapan Algoritma Goldbach Codes pada Kompresi File Gambar Terenkripsi Vigenere Cihper,” *Pelita Inform. Budi Darma*, vol. 17, no. April, pp.316–320, 2018.
- [12] A. M. Silitonga, S. D. Nasution, and P. Ramadhani, “Implementasi Algoritma Subexponential Code Untuk Kompresi File Gambar,” *KOMIK (Konferensi Nas.Tekno. Inf. dan Komputer)*, vol. 3, no. 1, pp. 340–348, 2019, doi:10.30865/komik.v3i1.1611.
- [13] M. Simangunsong, “Perbandingan Algoritma Elias Delta Code Dan Unary Coding Dalam Kompresi Citra Forensik,” *Resolusi Rekayasa Tek. Inform. dan ...*, vol. 1, no. 1, pp. 18–26, 2020, [Online]. Available: <http://djournals.com/resolusi/article/view/9>
- [14] T. P. Sari, S. D. Nasution, and R. K. Hondro, “Penerapan Algoritma Levenstein Pada Aplikasi Kompresi File Mp3,” *KOMIK (Konferensi Nas. Tekno. Inf. dan Komputer)*, vol. 2, no. 1, pp. 439–443, 2018, doi: 10.30865/komik.v2i1.946.