JIKTEKS : Jurnal Ilmu Komputer dan Teknologi Informasi
Vol. 02, No. 01, Desember 2023, hlm. 23 - 34
ISSN: 2986-5417
Laman Jurnal: jurnal.faatuatua.com/index.php/JIKTEKS

Implementasi Algoritma Dynamic Markov Compression (DMC) Untuk Kompresi Templates Desain Grafis

Amri Munthe
Teknik Informatika, Universitas Budi Darma, Indonesia
e-mail: *munteamri7@gmail.com*

Keywords:

Algorithm,
Graphic Design,
DMC,
Implementation,
Compression,
Teamplate.

ABSTRACT

Nowadays, everything is digital and many applications have been created to make work easier or as a means of entertainment. Some applications that exist today provide templates that can be used easily. Currently, the application that is still often used to produce graphic design templates is Canva. However, the templates produced have a large size such as templates with png format, which causes a long delivery process, especially when used online. The design of this system consists of several main processes, namely the compression process and the decompression process as well as the process of calculating the performance of Compression Ratio (CR), Ratio of Compression (RC), Rendudancy (RD), and Space Saving (SS). In this study the authors used the Dynamic Markov Compression algorithm where this compression technique was introduced by Gordon Cormak and Nigel Horspol which is one of the lossless compression techniques, where this algorithm is able to compress large capacity data into smaller capacity data and the original data can be reconstructed after decompression. The performance results of the Dynamic Markov Compression algorithm aim to determine its performance on graphic design templates. In this algorithm there are two stages, namely compression and decompression, the compression stage with the aim of compressing the size of graphic design templates and the decompression stage aims to restore the size of graphic design templates to their original size. The results of the application that the author did manually with the Dyanamic Markov Compression algorithm are Ratio of Compression (RC) = 1.76, Compression Ratio (CR) = 56.6%, Rendudancy (RD) = 43.3%, Space Saving (SS) = 43.4% with the conclusion that the percentage of graphic design templates compression results using the Dynamic Markov Compression algorithm is declared successful.

Kata Kunci

Algoritma,
Desain Grafis,
DMC,
Implementasi,
Kompresi,
Teamplate.

ABSTRAK

Saat ini segala sesuatunya sudah berbasis digital dan sudah banyak aplikasi yang dibuat untuk mempermudah segala pekerjaan ataupun dijadikan sebagai sarana hiburan. Beberapa aplikasi yang ada saat ini menyediakan templates yang dapat digunakan dengan mudah. Saat ini aplikasi yang masih sering digunakan untuk menghasilkan templates desain grafis adalah canva. Namun templates yang dihasilkan memiliki ukuran besar seperti templates dengan format png, yang menyebabkan proses pengiriman lama terutama ketika digunakan secara daring. Perancangan sistem ini terdiri dari beberapa proses utama yaitu proses kompresi dan proses dekompresi serta proses perhitungan kinerja Compression Ratio (CR), Ratio of Compression (RC), Rendudancy (RD), dan Space Saving (SS). Pada penelitian ini penulis menggunakan algoritma Dynamic Markov Compression yang dimana teknik kompresi ini diperkenalkan oleh Gordon Cormak dan Nigel Horspol yang merupakan salah satu teknik kompresi yang bersifat lossless, dimana algoritma ini mampu melakukan kompresi data yang berkapasitas besar menjadi data berkapasitas lebih kecil dan data asli dapat direkontruksi kembali setelah dekompresi. Hasil kinerja dari algoritma Dynamic Markov Compression bertujuan untuk mengetahui performansinya terhadap templates desain grafis. Pada algoritma ini terdapat dua tahap yaitu kompresi dan dekompresi, tahap kompresi dengan tujuan memampatkan ukuran templates desain grafis dan tahap dekompresi bertujuan untuk mengembalikan ukuran templates desain grafis ke ukuran semula. Hasil penerapan yang penulis lakukan secara manual dengan algoritma Dyanamic Markov Compression yaitu Ratio of Compression (RC) = 1,76, Compression Ratio (CR) = 56,6%, Rendudancy (RD) = 43,3%, Space Saving (SS) = 43,4% dengan kesimpulan bahwa persentase hasil kompresi templates desain grafis menggunakan algoritma Dynamic Markov Compression dinyatakan berhasil.

Korespondensi Penulis *):

Amri Munthe
Universitas Budi Darma
Jl. Sisingamangaraja No. 338 Simpang Limun, Kota Medan

Diajukan: 05-12-2023 | Diterima: 12-12-2023 | Diterbitkan: 30-12-2023

1. PENDAHULUAN

Templates desain grafis merupakan kerangka atau pola yang telah dibuat sebelumnya untuk membantu dalam pembuatan desain grafis. Templates tersebut digunakan sebagai kerangka kerja yang dapat digunakan secara berulang dalam proses desain, karena memuat elemen-elemen yang telah ditentukan sebelumnya seperti *layout*, tata letak, elemen grafis dan gaya visual. Templates desain grafis dapat mempercepat proses desain, sehingga menghemat waktu dan produksi yang lebih cepat. Templates desain grafis membantu konsistensi visual dalam suatu proyek desain, dengan menggunakan templates yang telah tersedia seperti logo, warna, dan gaya visual yang baik.

Templates desain grafis juga dapat digunakan untuk menghemat biaya produksi, karena menggunakan template yang sudah ada. Penggunaan templates desain grafis semakin sering digunakan dalam bidang desain dan kreatif, karena templates desain grafis tersebut dapat membantu dan mempermudah proses pembuatan desain dan meningkatkan efesiensi dalam pengembangan desain grafis yang semakin banyak diminati sekarang ini, sehingga membutuhkan ruang penyimpanan yang cukup besar, terutama ketika digunakan untuk transfer data secara daring atau melalui jaringan.

Templates desain grafis yang sangat berkembang dan kompleks, serta kebutuhan akan peyimpanan data yang efesien menjadi semakin penting seiring dengan pertumbuhan jumlah data yang dihasilkan oleh para desainer grafis. Dalam templates desain grafis sering digunakan berbagai jenis templates atau pola desain. Penggunaan templates desain grafis semakin banyak digunakan dalam bidang desain dan kreatif, karena templates desain grafis tersebut dapat membantu dan mempermudah proses pembuatan desain yang cukup besar, sehingga membutuhkan ruang peyimpanan yang cukup besar. Maka dengan adanya metode kompresi, maka permasalahan tersebut dapat diatasi.

Berdasarkan penelitian terdahulu, yang mengatakan bahwa ukuran file templates yang besar memerlukan memori ruang penyimpanan yang dibutuhkan juga besar dan waktu untuk transfer file akan lebih lama. Ukuran file templates yang besar dapat menjadi kendala dalam waktu pengiriman file templates sehingga memperpanjang waktu transfer [1].

Berdasarkan hal itu dibutuhkan suatu proses untuk menghemat ruang penyimpanan tersebut, sehingga dapat membantu proses pengiriman maupun penggunaan templates yang akan digunakan. Dengan adanya metode kompresi, maka permasalahan tersebut dapat diatasi dengan baik.

Kompresi atau pemampatan merupakan salah satu ilmu pengetahuan di bidang komputer. Pemampatan dilakukan karena adanya keterbatasan dalam ruang memori, keterbatasan media penyimpanan data dan kebutuhan waktu transfer data yang terbatas. Dengan adanya permasalahan tersebut maka dilakukan suatu teknik pengkompresian data agar dapat mengoptimalkan ruang penyimpanan dan kinerja komputer.

Berdasarkan penelitian terdahulu, mengatakan bahwa salah satu manfaat dari kompresi adalah untuk mengurangi kapasitas kosong dalam memori media peyimpanan, sehingga pengguna tidak perlu menggunakan terlalu banyak media penyimpanan. Menggunakan teknik kompresi, data dapat dikecilkan sehingga lebih efisien dalam penyimpanan dan pertukaran data [2].

Meskipun ada banyak algoritma kompresi data yang tersedia, tidak semua algoritma tersebut cocok untuk digunakan dalam mengkompresi data grafis yang kompleks dan membutuhkan penggunaan pola desain. Algoritma *dynamic markov compression* memiliki keunggulan dalam mengkompresi data yang memiliki pola yang teratur seperti data teks dan data citra, sehingga algoritma dynamic markov compression ini cocok dalam mengkompresi templates desain grafis.

Algoritma *dynamic markov compression* pertama kali diperkenalkan oleh Gordon Cormack dan Nigel Horspol. Beliau mengembangkan algoritma *dynamic markov compression* ini sebagai salah satu cara untuk mengkompresi data teks yang memiliki pola teratur daripada algoritma kompresi yang tersedia saat itu. Algoritma *dynamic markoc compression* adalah algoritma kompresi data yang berbasis pada model *finite-state*.

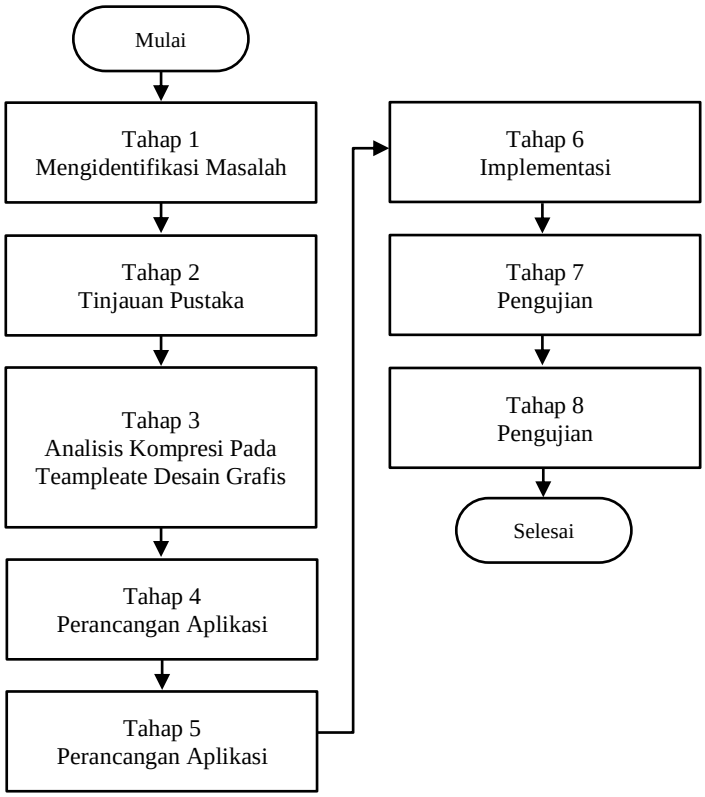
Berdasarkan penelitian terdahulu, mengatakan bahwa metode *dynamic markov compression* merupakan teknik kompresi adaptif yang memanfaatkan finite-state machine untuk merepresentasikan karakteristik file input. Metode ini didasarkan pada model *finite-state* sehingga struktur mesin berubah saat file diproses [3].

Algoritma ini bekerja dengan membangun model markov dari data input untuk memprediksi kemunculan karakter atau piksel selanjutnya. Kemudian hasil prediksi akan dicocokkan dengan data input dan memperbarui model markov berdasarkan hasil cocokan tersebut. Proses ini diulang secara iteratif, sehingga model markov terus diperbarui dan hasil prediksi akan dikodekan menjadi kode biner yang lebih pendek, sehingga menghasilkan kompresi data.

2. METODE PENELITIAN

2.1 Kerangka Kerja Penelitian

Kerangka kerja penelitian menjabarkan tentang beberapa tahapan proses yang saling terkait satu sama lain secara sistematis.



Gambar 1. Kerangka Kerja Penelitian

Diagram kerangka kerja penelitian di atas, maka dapat diuraikan pembahasan dari masing-masing tahapannya sebagai berikut :

1. Mengidentifikasi masalah
   Tahapan ini penulis menguraikan sumber masalah dalam proses penelitian yang sedang dilakukan dengan membuat latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian dan manfaat penelitian, sehingga penulis dapat memperbaiki kelemahan dan kekurangan untuk menyempurnakan hasil uji coba penelitian.
2. Tinjauan pustaka
   Dengan mencari, membaca dan mengumpulkan berbagai sumber referensi seperti buku, jurnal dan artikel bacaan lainnya untuk dapat lebih memahami tentang pembahasan pokok penelitian yang akan dilakukan.
3. Analisa proses kompresi *templates* desain grafis.
   Tahapan ini dilakukan untuk mengetahui proses kompresi pada *templates* desain grafis dengan menggunakan algoritma *Dynamic Markov Compression* (DMC) yang bertujuan mempercepat waktu pengiriman dan memperkecil ruang penyimpanan. Demikian juga untuk Analisa proses dekompresi *templates* desain grafis.
   Mengembalikan *templates* desain grafis yang telah dikompresi menjadi *templates* asli seperti semula sebelum dilakukan pengkompresian.
4. Perancangan aplikasi
   Di tahapan ini memberikan gambaran tentang penggunaan sistem aplikasi kompresi pada *templates* desain grafis. Tahapan ini menggunakan *templates* yang telah diubah kedalam bentuk sederhana yang mudah dimengerti oleh pemakai.
5. Implementasi
   Tahapan ini dilakukan proses penerapan algoritma *Dynamic Markov Compression* (DMC) pada sistem yang telah dirancang menggunakan aplikasi Visual Studio.Net 2008.
6. Pengujian
   Tahapan ini bertujuan untuk mengetahui hasil akhir penelitian yang dilakukan pada kompresi *templates* desain grafis dan untuk mengetahui apakah sistem yang telah dirancang dapat berjalan dan digunakan dengan baik sesuai kebutuhan atau masih memerlukan pengembangan dan perbaikan pada sistem tersebut.
7. Dokumentasi

Tahapan ini merupakan akhir penelitian yang dibuat kedalam bentuk laporan penelitian. Tahapan ini akan menjelaskan terhadap aplikasi yang telah dibuat agar memudahkan pembaca untuk dapat mengembangkan lebih luas lagi terhadap aplikasi yang telah dirancang.

2.2 Kompresi

Kompresi dalam ilmu komputer diartikan sebagai teknik pemampatan data dengan tujuan untuk meminimalisir penyimpanan data dan mempercepat pengiriman data sehingga ruang penyimpanan menjadi lebih efesien dan mempersingkat waktu *transfer* data [4]. Data yang besar akan berpengaruh pada penyimpanan perangkat sehingga perlu dilakukan sebuah cara yaitu kompresi.

Proses kompresi data pada suatu algoritma memiliki beberapa parameter yang digunakan untuk mengukur frekuensi tersebut [4], yaitu:

1. *Ratio of Compression* (*RC*), merupakan nilai perbandingan antara ukuran bit data sebelum dikompresi dengan ukuran bit data yang telah dikompresi. Dalam sistem matematis dapat dituliskan sebagai berikut:
$$R_C = \frac{\text{Ukuran data sebelum dikompresi}}{\text{Ukuran data setelah dikompresi}} \quad \text{..... (1)}$$
2. *Compression Ratio* (*CR*), merupakan persentase perbandingan antara data yang sudah dikompresi dengan data yang belum dikompresi.
$$C_R = \frac{\text{Ukuran data setelah dikompresi}}{\text{Ukuran data sebelum dikompresi}} \times 100\% \quad \text{..... (2)}$$
3. *Redudancy* (Rd), adalah bagian tambahan yang berisi data-data sebelum dikompresi. Setelah melakukan kompresi data, dapat dihitung redudansi data tersebut, yaitu persentase dari selisih antara ukuran suatu data sebelum kompresi dengan setelah kompresi.
$$R_d = \frac{\text{File sebelum dikompresi} - \text{File setelah dikompresi}}{\text{Ukuran file sebelum dikompresi}} \text{x} 100\% \quad \text{..... (3)}$$
4. *Space Saving* (SS), adalah persentase dari selisih antara data yang belum terkompresi dengan yang telah dikompres.
$$SS = 100\% - C_R \quad \text{..... (4)}$$

Dilihat dari *output* yang dihasilkan, teknik kompresi terbagi menjadi dua jenis [4], yaitu:

1. Kompresi *Lossless*
   Kompresi data algoritma *lossless* digunakan untuk mengurangi jumlah informasi sumber yang akan ditransmisikan dalam sedemikian rupa sehingga ketika informasi didekompresi, tidak ada kehilangan informasi. Kompresi *lossless* ini dimungkinkan karena memiliki statistik perulangan dan hasil perulangan tersebut merepresentasikan data lebih ringkas tanpa kehilangan keaslian data atau informasi yang dikompresi tersebut.
2. Kompresi *Lossy*
   Kompresi data *lossy* kontras dengan kompresi data *lossless*. Algoritma kompresi data *lossy* tidak menghasilkan tepat salinan informasi setelah dekompresi. Dalam skema ini, beberapa informasi mengalami kehilangan. Kompresi *lossy* mengurangi ukuran *file* dengan menghilangkan beberapa data berlebihan setelah *decoding*.

2.3 Teamplate Desain Grafis

Templates desain grafis adalah salah satu pola atau kerangka dasar berisi format atau tata letak yang digunakan untuk memudahkan pembuatan desain grafis [5]. Templates tersebut menyediakan struktur dan layout yang sudah ditentukan sebelumnya, sehingga memudahkan desainer ataupun pengguna dalam menempatkan setiap elemen elemen desain seperti gambar, ikon, teks dan elemen visual lainnya. Templates desain grafis sering digunakan dalam berbagai bidang proyek kreatif seperti poster, brosur dan kartu nama. Salah satu perangkat penyedia templates desain grafis yang populer saat ini dan mudah digunakan adalah canva, yang menyediakan berbagai templates desain grafis yang siap untuk digunakan.

Canva juga menyediakan fitur dan alat untuk memudahkan desainer dan pengguna untuk menyesuaikan desain yang diinginkan. Selain itu canva juga menyediakan elemen-elemen desain yang dapat di custom oleh desainer dan pengguna seperti gambar, icon dan font. Canva juga menyediakan format peyimpanan file templates desain [5],yaitu:

1. *Joint Photographic Group* (JPG).
   Format ini digunakan untuk meyimpan foto dengan ukuran file yang kecil. JPG menggunakan kompresi dengan kehilangan data, sehingga dapat mengurangi kualitas gambar.
2. *Portable Network Graphics* (PNG).
   Format ini digunakan untuk menyimpan gambar dengan latar belakang transparan atau gambar dengan teks dan grafik yang kompleks. PNG tidak kehilangan data ketika dilakukan kompresi, sehingga menjaga kualitas gambar yang baik.
3. *Portable Document Format* (PDF).
   Format ini digunakan untuk menyimpan desain yang komplek, seperti presentasi, brosur dan dokumen dengan tata letak yang rumit.
4. MPEG-4 Part 14 (MP4).

Format ini digunakan untuk menyimpan video yang umum dan kompatibel dengan banyak platform dan perangkat yang dihasilkan oleh canva.

5. *Graphics Interchange Format* (GIF).
   Format ini digunakan untuk meyimpan animasi atau gambar dengan sedikit warna. GIF memiliki ukuran file yang kecil, tetapi terbatas dengan warna.

Portable Network Grafik (PNG) digunakan dalam desain grafis dan web, karena format png ideal untuk menyimpan logo, grafik dan elemen-elemen desain yang menggunakan latar belakang transparan. *Portable Network Grafik* (PNG) mendukung warna 8-bit dan 24-bit. Versi 8-bit untuk gambar dengan warna terbatas, seperti grafik ikon dan ilustrasi sederhana. Sementara itu, versi 24-bit mendukung jutaan warna dan lebih digunakan untuk logo, foto atau gambar yang kompleks [5].

Templates desain grafis sering kali memiliki pola atau urutan data yang saling tergantung, seperti pengunaan warna atau bentuk yang serupa, sehingga ukuran file tersebut menjadi besar dan menjadi kendala ketika hendak mengirim atau menyimpannya.

2.4 Algoritma Dynamic Markov Compression (DMC)

Algoritma *Dynamic Markov Compression* atau yang disingkat dengan DMC merupakan salah satu metode kompresi yang bersifat *lossless* yaitu tidak menghilangkan keaslian data atau informasi sedikitpun. Metode ini dikembangkan oleh Gordon Cormak dan Nigel Horspol. Algoritma *Dynamic Markov Compression* (DMC) adalah metode kompresi data yang menggunakan model *Markov* untuk memprediksi simbol berikutnya dalam urutan data. Model *markov* adalah model statistik yang dapat digunakan untuk memodelkan urutan data yang saling tergantung, seperti urutan karakter dalam sebuah teks atau urutan piksel dalam sebuah gambar.

Penerapan algoritma *Dynamic Markov Compression* (DMC) pada kompresi *templates* desain grafis dapat membantu mengurangi ukuran *file template* tanpa mengorbankan kualitas desain. Dengan menggunakan model *markov* untuk memprediksi simbol berikutnya dalam urutan data, maka algoritma *Dynamic Markov Compresion* (DMC) dapat memampatkan urutan data tersebut dengan lebih baik (efektif) [8].

Kelebihan algoritma *dynamic markov compression* merupakan teknik kompresi yang adaptif, karena struktur mesin *finite-state* berubah seiring dengan pemrosesan file. Kemampuan kompresinya tergolong amat baik, meskipun waktu perhitungan yang dibutuhkan lebih besar dibandingkan metode lain. Kecepatan kompresi dari *dynamic markov compression* lebih unggul dibandingkan dengan kompresi model *huffman* [6].

Secara umum transisi ditandai dengan 0/p atau 1/q dimana p dan q menunjukkan jumlah transisi dari *state* dengan *input* 0 atau 1. Nilai probabilitas bahwa input selanjutnya bernilai 0 adalah p/(p+q) dan input selanjutnya bernilai 1 adalah q/(p+q). Lalu bila bit sesudahnya ternyata bernilai 0, jumlah bit 0 ditransisi sekarang ditambah 1 menjadi p+1. Begitu pula bila bit sesudahnya ternyata bernilai 1, jumlah bit 1 di transisi sekarang ditambah 1 menjadi q+1.
Algoritma kompresi *dynamic markov compression* :
1. s = 1 (jumlah *state* sekarang)
2. t = 1 (state sekarang)
3. T[1][0] = T[1][1] = 1 (model inisialisasi)
4. C[1][0] = C[1][1] = 1 (inisialisasi untuk menghindari masalah frekuensi nol)
5. Untuk setiap *input bit* e :

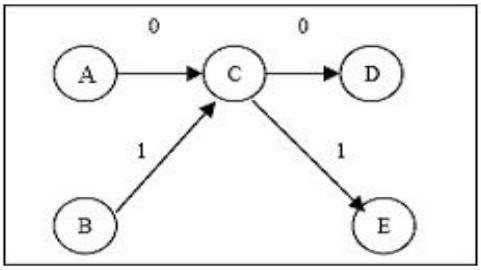
Dimana :

u = t
t = T[u][e] (ikuti transisi)
Kodekan e dengan probabilitas : C[u][e] / (C[u][0] + C[u][1])
C[u][e] = C[u][e]+1

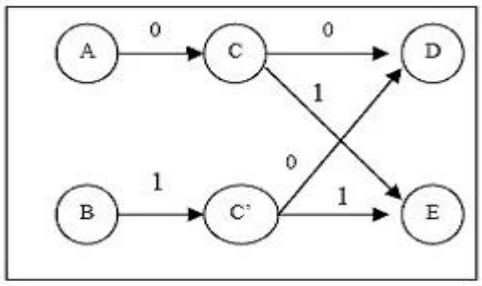
Jika ambang batas *cloning* tercapai, maka :
1. s = s + 1 (*state* baru t‘)
2. T[u][e] = s ; T[s][0] = T[t][0] ; T[s][1] = T[t][1]
3. Pindahkan beberapa dari C[t] ke C[s]

Masalah tidak terdapatnya kemunculan suatu bit pada *state* dapat diatasi dengan menginisialisasi model awal *state* dengan satu. Probabilitas dihitung menggunakan frekuensi relatif dari dua transisi yang keluar dari state yang baru. Jika frekuensi transisi dari suatu *state t* ke *state* sebelumnya, yaitu *state u*, sangat tinggi, maka *state* t dapat di-*cloning*. Ambang batas nilai *cloning* harus disetujui oleh *encoder* dan *decoder*. *State* yang di-*cloning* diberi simbol t‘ (lihat gambar 2 dan 3).

Aturan *cloning* adalah sebagai berikut :
1. Semua transisi dari *state* u dikirim ke *state* t‘. Semua transisi dari *state* lain ke *state* t tidak berubah.
2. Jumlah transisi yang keluar dari t‘ harus mempunyai rasio yang sama (antara 0 dan 1) dengan jumlah transisi yang keluar dari t.
3. Jumlah transisi yang keluar dari t dan t‘ diatur supaya mempunyai nilai yang sama dengan jumlah transisi yang masuk.



Gambar 2. Model DMC Sebelum Cloning



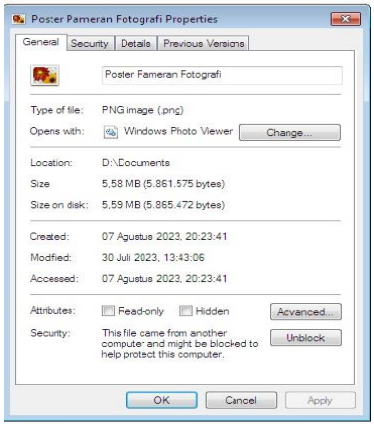
Gambar 3. Model DMC Sesudah Cloning

Dari skema yang telah ada akan diterapkan untuk kompresi dan dekompresi *file templates* desain menggunakan algoritma *dynamic markov compression*, akan dilihat kecepatan dan besaran kompresi yang dapat dihasilkan algoritma *dynamic markov compression* dalam kompresi *file templates*.

3. HASIL DAN ANALISIS

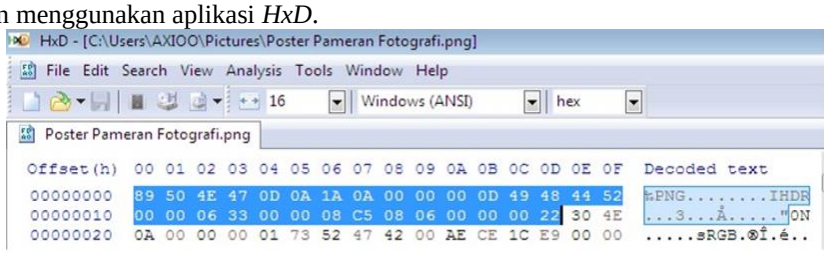
3.1 Proses Kompresi Teamplate Desain Grafis

Mengkompresi *templates* desain grafis menggunakan algoritma *dynamic markov compression* yang merupakan salah satu teknik kompresi *lossless*. Berdasarkan gambar 3 dibawah ini dengan sampel *templates* yang digunakan untuk proses analisa, pilih *file templates* desain grafis berformat.png terlebih dahulu untuk dilakukan kompresi. Berikut merupakan sampel *file templates* yang akan dikompresi:



Gambar 3. Sampel File Templates

Berdasarkan sampel yang tertera, maka *file templates* dengan format .png tersebut diubah kedalam bentuk nilai *hexadecimal* dengan menggunakan aplikasi *HxD*.



Gambar 4. Nilai Hexadecimal Sampel Data

Melakukan kompresi *file templates*, dapat diambil 30 nilai *hexadecimal* dengan total 240 bit (30 *byte*) untuk mewakili *file templates* dengan *format* .png yang telah dirubah dengan aplikasi *HxD*.

Tabel 1. Nilai Hexa *File* Sampel

Nilai Hexadecimal	Frekuensi
89	1
50	1
4E	1
47	1
0D	2
0A	2
1A	1
00	10
49	1
48	1
44	1
52	1
06	2
33	1
08	2
C5	1
22	1

Selanjutnya lakukan pengurutkan nilai *hexadecimal* berdasarkan frekuensi terbesar hingga frekuensi terkecil. Pada pembahasan kali ini nilai biner diawal berjumlah 8 *bit*, seperti yang dapat dilihat pada tabel dibawah ini ukuran *string* bit sebelum dilakukan proses kompresi yaitu 240 bit.

Tabel 2. Nilai Hexadecimal Kompresi

No	Nilai hexa	Freq	Nilai biner	Bit	Freq * Bit
1	00	10	00000000	8	80
2	0D	2	00001101	8	16
3	0A	2	00001010	8	16
4	06	2	00000110	8	16
5	08	2	00001000	8	16
6	89	1	10001001	8	8
7	50	1	01010000	8	8
8	4E	1	01001110	8	8
9	47	1	01000111	8	8
10	1A	1	00011010	8	8
11	49	1	01001001	8	8
12	48	1	01001000	8	8
13	44	1	01000100	8	8
14	52	1	01010010	8	8
15	33	1	00110011	8	8
16	C5	1	11000101	8	8
17	22	1	00100010	8	8
TOTAL	**30**	-	-	**240**	

Berikut ini merupakan tabel yang berisi nilai *codeword dynamic markov compresion* dimulai untuk nilai N dari 1 sampai 17. Pembentukan dilakukan berdasarkan formula atau rumus 2*(n+3), sehingga untuk nilai n = 1 adalah 2*(2+3)=10. Lalu dibentuk dari hitungan bilangan prima yang dimulai dengan nilai 3, 5, 7, 11, 13,.., dstnya. Sehingga diperoleh codeword dari n=1 adalah 11.

Tabel 3. Codeword Dynamic Markov Compression

N	Codeword
1	11
2	101
3	011
4	1001
5	0101
6	0011
7	00101
8	010001
9	00011
10	0010001
11	000101
12	000011
13	0000101
14	00010001
15	0000011
16	0010000001
17	00000101

Kemudian nilai *hexadecimal* diurutkan berdasarkan frekuensi nilai tertinggi hingga frekuensi nilai terendah, lalu tambahkan dengan *codeword* dmc.

Tabel 4. Nilai *Hexadecimal* yang telah dikompresi dengan dmc.

N	Nilai *hexa*	Freq	*Codeword*	Bit	Freq * Bit
1	00	10	11	2	20
2	0D	2	101	3	6
3	0A	2	011	3	6
4	06	2	1001	4	8
5	08	2	0101	4	8
6	89	1	0011	4	4
7	50	1	00101	5	5
8	4E	1	010001	6	6
9	47	1	00011	5	5
10	1A	1	0010001	7	7
11	49	1	000101	6	6
12	48	1	000011	6	6
13	44	1	0000101	7	7
14	52	1	00010001	8	8
15	33	1	0000011	7	7
16	C5	1	0010000001	10	10
17	22	1	00000101	8	8
Total Bit	127				

Berdasarkan tabel tersebut, maka ukuran akhir *string* bit yang dihasilkan dari kompresi dengan menggunakan algoritma *dynamic markov compression* ialah 127 bit. Lalu rubah dan sesuaikan *codeword dynamic markov compresion* dengan masing-masing nilai *hexadecimal* data sebelum dikompresi (diawal).

Tabel 5. Codeword Dynamic Markov Compression

89	**50**	**4E**	**47**	**0D**	**0A**
0011	00101	010001	00011	101	011
1A	**0A**	**00**	**00**	**00**	**0D**

0010001	011	11	11	11	101
49	**48**	**44**	**52**	**00**	**00**
000101	000011	0000101	00010001	11	11
06	**33**	**00**	**00**	**08**	**C5**
1001	0000011	11	11	0101	0010000001
08	**06**	**00**	**00**	**00**	**22**
0101	1001	11	11	11	00000101

String bit yang dihasilkan dari *codeword* berjumlah 127 bit.
"00110010 10100010 00111010 11001000 10111111 11101000 10100001 10000101 00010001 11111001 00000111 11101010 01000000 10101100 11111110 0000101"
Kemudian mencari nilai *padding* dan *flagging.*

1. Jika sisa bagi panjang *string* bit terhadap 8 ialah 0, maka tidak perlu ditambahkan *padding* hanya penambahan *flagging*.
2. Jika sisa panjang *string* bit terhadap 8 ialah n (1,2,3,4,5,6,7) maka tambahkan *padding* dan *flagging* di akhir string bit. Dengan rumus *padding* 7 – n + "1", sedangkan *flagging* dengan rumus 9 – n (sisa hasil bagi).
   Karena 127 tidak habis dibagi 8, dilakukan penambahan *padding* dan *flaging.*

Tabel 6. Penambahan padding dan flaging

Padding	Flagging
127 *mod* 8 = 7 = n	9 – n
7 - 7 + "1" = **1**	9 – 7 = 2 = **00000010**

Tambahkan hasil *padding* = **1** dan hasil *flagging* = **00000010** di akhir *string* bit:
"00110010 10100010 00111010 11001000 10111111 11101000 10100001 10000101 00010001 11111001 00000111 11101010 01000000 10101100 11111110 00001010**1 00000010**"
Setelah ditambahkan *padding* dan *flagging* total bit menjadi **136** bit. Lalu pisahkan per 8 bit, kemudian rubah ke dalam bentuk *hexadecimal* serta karakter seperti tabel dibawah :

Tabel 7. Hasil Kompresi *Dynamic Markov Compression*

Biner	**Hexadesimal**	**Karakter**
00110010	32	2
10100010	A2	¢
00111010	3A	:
11001000	C8	È
10111111	BF	¿
11101000	E8	È
10100001	A1	¡
10000101	85	…
00010001	11	
11111001	F9	Ù
00000111	7	
11101010	EA	Ê
01000000	40	@
10101100	AC	¬
11111110	FE	Þ
00001010	B	
00000010	2	

Setelah *file templates* dikompresi dengan algoritma *dynamic markov compression*, maka diperoleh hasil karakter baru dan kemudian disimpan dengan *file* kompresi *dynamic markov compression*, tabel karakter-karakter hasil kompresi disusun menjadi *file.text* dalam bentuk *file notepad.*

Mengetahui kinerja algoritma *dynamic marov compression* maka perlu dilakukan perhitungan dengan parameter yang telah ditentukan sebagai berikut :
Ukuran data sebelum dikompresi = 240/8 = 30 *byte*

Ukuran data sesudah dikompresi = 136/8 = 17 *byte*

Ratio of Compression (RC) $= \frac{\text{ukuran sebelum dikompresi}}{\text{ukuran setelah dikompresi}}$

$= \frac{240}{136} = 1{,}76$

Compression of Ratio (CR) $= \frac{\text{ukuran setelah dikompresi}}{\text{ukuran sebelum dikompresi}} \text{ X } 100\%$

$= \frac{136}{240}\ x\ 100\%$

$= 56{,}6\%$

Redudancy (Rd) $= \frac{\text{sebelum dikompresi} - \text{setelah dikompresi}}{\text{ukuran sebelum dikompresi}} \text{ X } 100\%$

$= \frac{240-136}{240}\ x\ 100\%$

$= 43{,}3\%$

Space Saving (Ss) = 100% – *Compression of Ratio (Cr)*

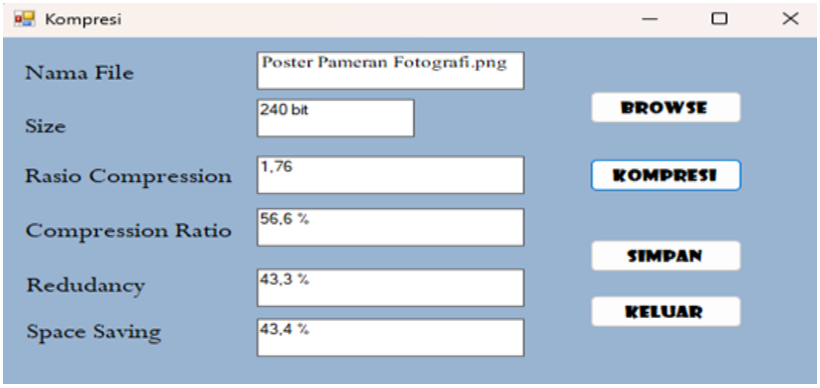
= 100% - 56,6%

= 43,4%

3.2 Pengujian Sistem

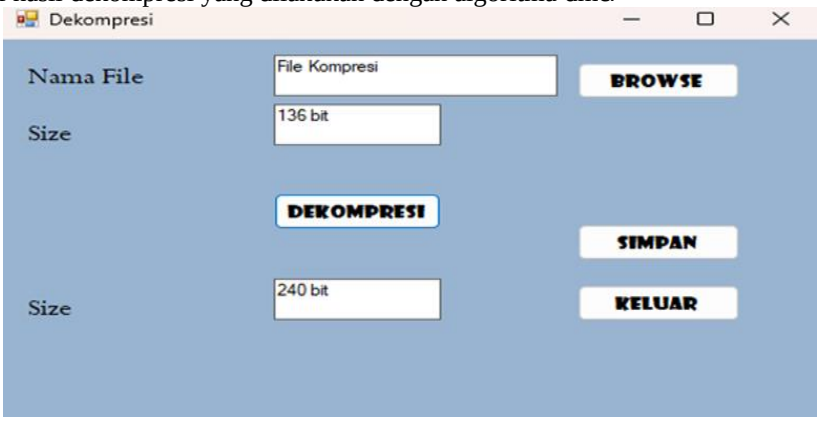
Adanya pengujian sistem ini bertujuan agar dapat mengetahui bagaimana sistem yang sudah dibangun dan akan dijalankan. Pada sistem ini dirancang dengan sesederhana mungkin agar dapat diakses dengan mudah. Berikut hasil dari *printout* saat sistem sedang berjalan.

1. *Form* Kompresi.
   Form ini akan menampilkan proses kompresi pada *templates* desain grafis dengan menggunakan algoritma *dynamic markov compression*. Dimana tahapan awalnya ialah *input templates* desain grafis yang akan dikompresi.



Gambar 5. Tampilan Output Form Kompresi

2. *Form* Dekompresi.
   Form ini akan menampilkan proses dekompresi pada *templates* desain grafis dengan algoritma *dynamic markov compression* dengan cara memasukkan *templates* desain grafis yang telah disimpan dari hasil kompresi. Berikut ini gambar dari hasil dekompresi yang dilakukan dengan algoritma dmc.



Gambar 6. Tampilan Output Form Dekompresi

Dari hasil pengujian di atas, setelah mendapatkan nilai dari algoritma dynamic markov compression, maka diperoleh *templates* desain grafis terkompresi memiliki ukuran yang lebih kecil dibandingkan dengan *templates* desain grafis sebelum dikompresi.

Tabel 8. Hasil Pengujian

No	Sebelum Dikompresi		Sesudah Dikompresi		Kinerja	
	Nama templates desain grafis	Ukuran awal	Nama templates desain grafis	Ukuran akhir		
1	Poster pameran.png	240 bit	Poster terkompresi	136 bit	RC	1,76
					CR	56,6%
					Rd	43,3%
					Ss	43,4%
2	Hitam merah kolase.png	224 bit	Kolase terkompresi	120 bit	RC	1,86
					CR	53,7%
					Rd	46,4%
					Ss	46,3%

4. KESIMPULAN

Pembahasan atas penelitian yang telah dilakukan, maka didapatkan hasil akhir dari penelitian ini yang dapat disimpulkan menjadi beberapa kesimpulan, sebagai berikut:

1. Setelah mengikuti prosedur kompresi dan dekompresi dengan menggunakan algoritma *dynamic markov compression* bahwa *templates* desain grafis berekstensi .png yang memiliki ukuran *templates* yang cukup besar, dapat dikompres menjadi *templates* dengan ukuran yang lebih kecil.
2. Setelah menerapkan algoritma *dynamic markov compression*, maka didapatkan hasil parameter *Ratio of Compression* (Rc) = 1,76, *Compression Rati* (Cr) = 56,6%, *Redudancy* (Rd) = 43,3%, dan *Space Saving* (Ss) = 43,4%
3. Adanya sistem aplikasi yang dibuat dapat membantu mengkompresi *templates* desain grafis.

REFERENSI

[1] Nainggolan, S., Pendahuluan, I., & Codes, A. A. G. (2019). Analisa Perbandingan Algoritma Goldbach Codes Dengan Algoritma Dynamic Markov Compression (DMC) Pada Kompresi File Teks Menggunakan. 6 (Dmc), 395–399. [Available]

[2] Jaya, I. K., & Perangin-angin, R. (2018). Analisa Perbandingan Rasio Kecepatan Kompresi Algoritma Dynamic Markov Compression Dan Huffman. Sinkron : Jurnal Dan Penelitian Teknik Informatika, 2(2), 78–85. [Available]

[3] Saragih, N. E., & Harahap, F. (2019). Perancangan Aplikasi Kompresi SMS dengan Algoritma Dynamic Markov Compression pada Android. Jurnal Ilmiah Core IT : Community Research Information Technology, 7(1), 1–6. [Available]

[4] Pradana, A. (2022). Analisa Perbandingan Algoritma Elias Gamma Code Dan Algoritma Goldbach Code Pada Kompresi File Dokumen. 6(November), 345–356. [Available]

[5] Alphi, "Pengertian, Apa itu Templating, Fungsi, Kelebihan, Kekurangan, Jenis, Contoh, Jasa Pembuatan Website-Metafora Indonesia Tehnology," idmetafora.com, Apr. 19, 2014. [Available]

[6] G. Yuniadi, "Pengembangan Aplikasi Kompresi Teks Menggunakan Algoritma Dynamic Markov Compression (DMC) Pada Layanan SMS," dspace.uii.ac.id, 2011, Accessed: Feb. 27, 2023. [Available]

[7] Lubis, A. D., & Ginting, G. (2020). Design of Steganographic Applications in A Processed Image using Algorithm Dynamic Markov Compression. 4(2), 50–56. [Available]

[8] Salomon, D., & Motta, G. (2019). Handbook of Data Compression 5th. In Journal of Chemical Information and Modeling (Vol. 53, Issue 9) [Available]

[9] M. C. Aruan and W. Rahayu, "Analisis Performa Algoritma Kompresi Data dalam Penyimpanan dan Transfer Data," LANCAH: Jurnal Inovasi dan Tren, vol. 1, no. 2, pp. 228–232, Nov. 2023. [Available]

[10] K. M. R. Hutahaean, "Penerapan Algoritma Inverted Elias Delta Untuk Kompresi Konten Pada Aplikasi Psikologi Berbasis Android," Jurnal Riset Teknik Informatika dan Data Sains, vol. 1, no. 1, pp. 26–37, Dec. 2022, Accessed: Mar. 03, 2024. [Available]

[11] M. Iqbal, R. A. Prayogi, and D. Yunitasari, "Analisis Kompresi File Teks Menggunakan Algoritma Lempel Ziv Welch (LZW)," Unnes Journal of Mathematics, vol. 11, no. 2, pp. 112–119, 2022. [Available]

[12] C. Imam and M. F. Siregar, "Implementation Of Huffman And LZ78 Algorithm For Character Compression," INFOKUM, vol. 10, no. 5, pp. 207–211, Dec. 2022, Accessed: Mar. 03, 2024. [Available]

[13] U. Jayasankar, V. Thirumal, and D. Ponnurangam, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 2, pp. 119–140, Feb. 2021. [Available]

[14] C. K. Jha and M. H. Kolekar, "Empirical Mode Decomposition and Wavelet Transform based ECG Data Compression Scheme," IRBM, May 2020. [Available]

[15] "Hiding Data Using Efficient Combination of RSA Cryptography, and Compression Steganography Techniques," ieeexplore.ieee.org, 2021. [Available]

[16] "Optimizing Error-Bounded Lossy Compression for Scientific Data by Dynamic Spline Interpolation | IEEE Conference Publication | IEEE Xplore," ieeexplore.ieee.org. (accessed Mar. 03, 2023). [Available]

[17] H. Issah and E. Martin, "Wavelet decomposition for passive data compression and processing." Accessed: Mar. 03, 2024. [Available]