

KONVERSI DATA KE FORMAT DNA DAN PERBANDINGAN HASIL KOMPRESINYA MENGGUNAKAN GENCOMPRESS TERHADAP WINRAR

Heilbert Armando Mapaly, Teguh Bharata Adji, Noor Akhmad Setiawan

Jurusan Teknik Elektro dan Teknologi Informasi, Fakultas Teknik
Universitas Gadjah Mada Yogyakarta

hb_1x@yahoo.co.id

Abstrak

Teknik kompresi data saat ini merupakan hal yang penting dalam menyimpan media dalam bentuk digital. Secara umum, teknik kompresi terbagi menjadi lossy compression dan loseless compression. Penelitian ini mencoba untuk menerapkan konsep kompresi DNA yang bersifat loseless. Karena DNA hanya terdiri dari huruf a, c, g, dan t, maka kompresi ini hanya dapat diaplikasikan pada media teks, padahal, media yang ada saat ini tidak hanya berupa teks saja. Oleh sebab itu penelitian ini bertujuan untuk menemukan cara agar kompresi tersebut dapat diaplikasikan pada media lain serta membandingkan hasil kompresi tersebut dengan aplikasi WinRAR. Untuk dapat dikompresi dengan teknik kompresi DNA, semua file tersebut harus dikonversikan ke dalam bentuk DNA. Proses konversi tersebut dilakukan dengan mengubah kode binary yang ada pada file menjadi untaian rantai DNA. 00 diubah menjadi 'a', 01 diubah menjadi 'c', 10 diubah menjadi 'g' dan 11 diubah menjadi 't'. Walaupun pada konversi data ke format DNA ukuran file DNA menjadi lebih besar dari file awal, namun ukuran file hasil kompresi menjadi lebih kecil dari ukuran file awal karena memanfaatkan kelebihan teknik kompresi DNA. Ukuran file hasil kompresi dengan GenCompress menunjukkan hasil yang lebih baik dari WinRAR ketika terdapat pengulangan dengan jumlah yang sama pada isi file.

Kata kunci: konversi, GenCompress, algoritma DNA.

1. Pendahuluan

Hingga saat ini, berbagai penelitian telah dilakukan untuk menghasilkan teknik kompresi yang lebih baik. Secara umum, teknik kompresi data dibagi menjadi dua bagian yaitu *lossy* dan *lossless* (Nelson dan Gailly, 1995). Kompresi data *lossy* memiliki kelemahan dalam akurasi data karena untuk meningkatkan kompresi tersebut terdapat beberapa informasi yang hilang ketika data dikompres. Sedangkan pada kompresi *lossless*, tidak ada informasi yang hilang ketika kompresi dilakukan. Kompresi *lossy* biasanya digunakan pada gambar dan suara digital, sedangkan kompresi *lossless* biasanya digunakan ketika menyimpan catatan database, *spreadsheet*, dan *file* pengolah kata.

Kompresi pada DNA telah ditemukan sejak tahun 1993 dan telah menghasilkan beberapa teknik kompresi seperti BioCompress (Grumbach dan Tah, 1993), BioCompress-2 (Grumbach dan Tah, 1994), GenCompress (Chen, et al., 1999), DNACompress (Chen, et al., 2002), dan DNASequitur (Cherniavsky

dan Ladner, 2004). Rantai DNA, yang merupakan kombinasi acak antara a, c, g dan t, dimasukkan ke dalam suatu *file* teks lalu dikompres. Hasil kompresi tersebut dapat dikembalikan lagi (dekompres) tanpa ada informasi yang hilang (*lossless compression*).

Karena hanya terdiri dari empat jenis basa Nitrogen (a, c, g, dan t), maka 2 bit cukup untuk menyimpan setiap basa Nitrogen tersebut (Chen, et. al., 1999). Hal ini merupakan kelebihan dari teknik kompresi DNA. Jika menggunakan teknik kompresi standar seperti "compress" dan "compact" yang ada di Unix atau program arsip "pkzip" yang ada di MS-DOS, teknik-teknik kompresi tersebut menyimpan setiap basa Nitrogen ke dalam lebih dari 2 bit. Dengan menyimpan setiap basa ke dalam lebih dari 2 bit, artinya teknik kompresi tersebut memakai lebih banyak memori daripada teknik kompresi DNA.

Akan tetapi, teknik-teknik kompresi DNA, misalnya GenCompress, tidak dapat digunakan untuk media lain, misalnya untuk gambar, karena informasi pada rantai DNA hanya terdiri dari huruf a, c, g dan t. Oleh sebab itu, kompresi tersebut hanya dapat digunakan pada *file* DNA dalam bentuk teks. Pada kenyataannya, *file* teks dan gambar disimpan dalam komputer dalam bentuk *binary*. Jika dibandingkan dengan *tools* yang sering digunakan dalam kompresi *file*, *tools* tersebut mampu untuk melakukan kompresi pada berbagai jenis *file*.

Kompresi DNA mampu melakukan kompresi pada setiap basa Nitrogen menjadi 2 bit atau kurang dari 2 bit. Akan tetapi, permasalahan yang ditemui adalah teknik-teknik kompresi pada DNA hanya dapat diterapkan pada *file* DNA yang berbentuk teks. Teknik kompresi DNA tersebut tidak dapat diterapkan pada jenis *file* yang lain.

Karena teknik kompresi DNA hanya dapat diterapkan pada *file* DNA dalam bentuk teks, maka tujuan dari penelitian ini adalah melakukan konversi data dari *binary* ke dalam bentuk DNA, agar dapat dikompres dengan menggunakan teknik kompresi DNA. Selain itu, penelitian ini membandingkan hasil kompresi tersebut dengan *software* kompresi data WinRAR.

2. Landasan Teori

DNA (Asam Deoksiribonukleat) adalah suatu materi genetik yang merupakan bagian kerja dari kromosom. DNA memiliki struktur seperti batang kecil dalam sel dan mengandung gen, suatu unit hereditas (Silverstein, et al., 2008). Gen membawa informasi yang menentukan karakteristik dari sel. DNA berbentuk seperti benang rantai unit kimia yang disebut Nukleotida, masing-

masing terdiri dari tiga bagian, yaitu Fosfat, gula (Deoksiribosa), dan zat yang disebut basa Nitrogen. Gula dan Fosfat terhubung secara bersama-sama dan membentuk rantai yang menjadi tulang punggung rantai. Basa Nitrogen dapat bereaksi dengan basa lainnya pada rantai yang berbeda. Jadi, bentuk DNA adalah rantai ganda, yang terdiri dari dua untai panjang Nukleotida yang dihubungkan oleh ikatan kimia antara basa Nitrogen.

Empat jenis basa Nitrogen yang ditemukan dalam DNA adalah Adenin, Timin, Guanin, dan Sitosin (biasanya disingkat dengan A, T, G, dan C). Basa Nitrogen tersebut selalu berikatan satu sama lain menurut aturan yang spesifik. A selalu berpasangan dengan T sedangkan C berpasangan dengan G.

Dalam penelitian Church, et. al.(2012), mereka berhasil mengkonversikan buku 'Regenesis' yang dalam bentuk HTML menjadi format DNA. Buku tersebut terdiri dari 53.426 kata, sebelas gambar, dan sebuah *file* Javascript. Setelah itu, kode biner dari HTML buku tersebut dikonversikan ke dalam bentuk DNA. Bit yang bernilai nol ditentukan sebagai huruf A atau C pada basis DNA Adenin dan Sitosin, dan bit yang bernilai satu ditentukan sebagai huruf G dan T pada basis DNA Guanin dan Timin.

Proses konversi yang dilakukan dalam penelitian Church, et. al (2012) adalah sebagai berikut: Pertama, teks tersebut diubah ke dalam bentuk biner. Sebagai contoh, karakter 'f' ketika diubah menjadi bentuk biner menjadi 01100110. Proses konversi dari biner ke DNA dilakukan dengan cara mengubah angka nol menjadi A atau C, dan angka satu menjadi T atau G. Setelah mengkonversi 01100110 ke dalam bentuk DNA, maka diperoleh suatu untai DNA 'aTGaaTTc'.

Namun, jika melihat aturan konversi dari biner ke DNA yang digunakan, maka biner 01100110 tidak selalu dikonversi menjadi 'aTGaaTTc'. Bisa saja kode biner 01100110 tersebut menjadi bentuk yang lain, misalnya 'cTTacGTa' atau 'aGGaaTTa'. Hal ini disebabkan karena kode biner nol dapat ditentukan sebagai A atau C, dan kode biner satu dapat ditentukan sebagai T atau G. Karena dalam 1 bit kode biner terdapat dua kemungkinan untai DNA yang berbeda, maka dalam 8 bit kode biner dapat menghasilkan 28 atau 256 kemungkinan untai DNA yang berbeda.

Selain itu, dengan melakukan konversi 01100110 menjadi 'aTGaaTTc', itu berarti mengubah 8 bit biner (1 *byte*) menjadi delapan karakter teks. Padahal, satu karakter dalam teks disimpan sebagai 1 *byte*. Hal ini menunjukkan bahwa 8

bit biner (setara dengan 1 *byte*) disimpan menjadi 8 *byte* karakter teks dalam bentuk DNA. Oleh sebab itu, ukuran *file* DNA-nya menjadi delapan kali lebih besar dari ukuran sebenarnya. Dengan mempertimbangkan kedua hal tersebut, maka penelitian ini menggunakan cara konversi yang berbeda dengan cara konversi pada penelitian dari Church, et.al. (2012).

3. Metodologi

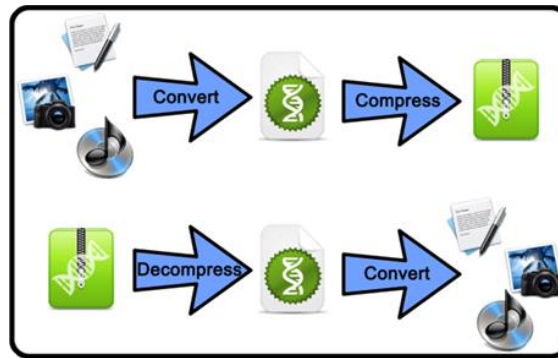
Penelitian ini menggunakan beberapa *dataset* yang terdiri dari sekumpulan *file* avi, bmp, txt, exe, wav dan *dataset* Brodatz (gif). Untuk lebih memudahkan, maka penamaan sampel-sampel tersebut menggunakan suatu kode seperti ditunjukkan pada Tabel 1. Huruf pertama menunjukkan jenis *file*, sedangkan huruf kedua (huruf x) menyatakan urutan atau sampel ke-x.

Proses konversi data menjadi format DNA menggunakan kode biner yang terdapat pada *file* dengan cara mengubah kode biner 00 menjadi 'a', 01 menjadi 'c', 10 menjadi 'g' dan 11 menjadi 't'. Proses ini menghasilkan suatu *file* yang berisi kode DNA dari *file*. Setelah itu, *file* yang berisi kode DNA tersebut dikompresi dengan menggunakan teknik kompresi DNA, yaitu GenCompress.

Tabel 1 Keterangan format penamaan sampel

Format Nama Sampel	Keterangan
Dx.gif	Gambar pada Dataset Brodatz dengan format .gif dimensi 640x640 pixel
Ix.bmp	<i>Printscreen wallpaper</i> dengan format .bmp dimensi 320x320 pixel.
Wx.wav	<i>File</i> suara di Windows 7 yang terdapat di C:\Windows\Media.
Ex.exe	Hasil <i>compile sourcecode</i> Tx.txt
Ax.avi	Potongan suatu film yang berformat .avi. Sampel <i>file</i> ini memiliki durasi 2-3 detik
Tx.txt	<i>Source code</i> program C sederhana.
Px.bmp	Gambar berpola dengan format .bmp dimensi 100x100 pixel

Demikian juga dengan proses dekompresi *file* tersebut. *File* hasil kompresi selanjutnya di-*decompress* menggunakan teknik dekompresi DNA, GenCompress, sehingga menghasilkan suatu *file* yang berisi kode DNA suatu *file*. Kode DNA tersebut kemudian diubah kembali menjadi *file* semula. Proses pengubahan menjadi *file* semula, merupakan kebalikan dari proses pengubahan pada proses kompresi yaitu, 'a' dikembalikan menjadi kode biner 00, 'c' menjadi 01, 'g' menjadi 10 dan 't' menjadi 11. Rancangan aplikasi untuk penelitian ini adalah seperti ditunjukkan pada Gambar 1.



Gambar 1 Rancangan Aplikasi

4. Hasil dan Pembahasan

Hasil dari penelitian yang dilakukan diantaranya adalah perangkat lunak untuk melakukan konversi dari *file* ke DNA dan DNA ke *file*. Perangkat lunak yang pertama mengubah sebuah *file* ke dalam bentuk teks DNA dan kemudian melakukan kompresi data dengan menggunakan perangkat lunak GenCompress. Sedangkan perangkat lunak yang kedua melakukan proses dekompresi dengan menggunakan perangkat lunak GenDecompress dan kemudian mengubah bentuk DNA tersebut ke dalam bentuk *file* semula.

Proses konversi yang dilakukan dalam penelitian ini, berbeda dengan proses konversi yang digunakan oleh Church, et. al. (2012). Dalam proses konversi dari *file* ke DNA, kode biner 00 menjadi 'a', 01 menjadi 'c', 10 menjadi 'g' dan 11 menjadi 't'. Jadi, proses konversi ini mengubah 8 bit kode biner (1 *byte*) menjadi empat karakter (4 *byte*) teks DNA. Walaupun ukuran *file* DNA-nya menjadi empat kali lipat lebih besar, tapi ukuran *file* DNA yang dihasilkan lebih kecil dari ukuran *file* dalam penelitian Church, et. al. (2012). Selain itu, konversi *file* ke DNA menghasilkan untaian DNA yang tetap. Sebagai contoh, kode biner 01 10 01 10, ketika diubah ke dalam bentuk DNA menjadi 'cgcg'.

Setelah seluruh kode biner dari file dikonversi ke DNA, untaian DNA tersebut disimpan dalam sebuah *file* baru yang berekstensi '.dna'. Selanjutnya *file* dengan ekstensi '.dna' tersebut dikompres dengan menggunakan perangkat lunak GenCompress dan kemudian menghasilkan *file* dengan ekstensi '.GEN'.

Proses dekompresi merupakan kebalikan dari proses kompresi. Pada tahap ini, *file* hasil kompresi yang berekstensi '.GEN' didekompres menggunakan perangkat lunak GenDecompress sehingga menghasilkan *file* dalam bentuk DNA. Hasil dekompresi dari GenDecompress ternyata menghasilkan *file* dengan ekstensi '.OUT'. Oleh sebab itu, *file* hasil dekompresi tersebut diubah menjadi ekstensi '.dna' dengan cara mengubah nama *file* tersebut.

Selanjutnya, *file* dalam bentuk DNA tersebut dikonversi menjadi bentuk biner. Proses konversi ini mengubah 'a' menjadi 00, 'c' menjadi 01, 'g' menjadi 10 dan 't' menjadi 11. Maka, empat karakter teks DNA diubah menjadi 8 bit kode biner. Sebagai contoh, DNA 'cgag', ketika diubah ke dalam bentuk biner menjadi 01 10 00 10.

4.1 Konversi *file* ke DNA

Pada tahap ini, sejumlah sampel diambil secara acak kemudian dikonversi ke dalam bentuk DNA. Berdasarkan proses konversi yang digunakan, ukuran dari *file* yang menyimpan untaian DNA menjadi empat kali lipat lebih besar daripada ukuran *file* yang asli, seperti yang telah dijelaskan pada Landasan Teori.

Ukuran *file* dalam format DNA memiliki ukuran empat kali lebih besar dari ukuran *file* semula. Hal ini disebabkan karena setiap 1 *byte* isi dari *file* asli diubah menjadi 4 *byte* teks dalam format DNA. Proses konversi ke dalam format DNA ini sangat dibutuhkan agar *file* tersebut dapat dikompres dengan menggunakan teknik kompresi DNA GenCompress. GenCompress hanya dapat melakukan kompresi jika isi dari *file* tersebut berupa untaian DNA.

4.2 Kompresi dengan menggunakan GenCompress

File DNA kemudian dikompres dengan menggunakan perangkat lunak GenCompress. Hasil dari kompresi tersebut dapat dilihat pada Tabel 2. Perhitungan GenCompress *ratio* pada Tabel 2 ditunjukkan pada Persamaan 1.

$$Compression_ratio = 1 - \frac{|Compressed_filesize|}{|Original_filesize|} \dots\dots\dots(1)$$

Pada Persamaan 1, jika ukuran *file* hasil kompresi semakin kecil, maka *compression ratio*-nya menjadi semakin lebih besar. Dari Tabel 2 terlihat bahwa hasil GenCompress *ratio* menunjukkan hasil diatas 75%. Ini berarti bahwa *file* DNA, yang ukurannya lebih besar empat kali lipat dari *file* asli, dapat dikompresi sehingga ukuran *file* hasil kompresi lebih kecil dari *file* semula. Perbandingan ukuran *file* asli dengan *file* hasil kompresi dapat dilihat pada Tabel 2.

Pada Tabel 2, perhitungan *compression ratio* tidak menggunakan rumus Persamaan 1, tetapi menggunakan rumus seperti yang ditunjukkan pada Persamaan 2.

$$Compression_ratio = \frac{Compressed_size}{Uncompressed_size} \dots\dots\dots(2)$$

Pada Persamaan 2, jika ukuran *file* hasil kompresi semakin kecil, maka *compression ratio* menjadi semakin lebih kecil. *Compression ratio* yang semakin kecil menunjukkan bahwa kompresi tersebut semakin bagus. Berdasarkan hasil pada Tabel 2, terlihat hasil kompresi untuk txt, exe dan pola dalam bmp menunjukkan hasil yang baik dibanding *file* avi, wav dan gif.

Dalam melakukan kompresi ini, komputer yang digunakan hanya membuka 2 (dua) buah aplikasi yaitu Command Prompt dan Microsoft Excel. Command Prompt untuk menjalankan perintah kompresi sedangkan Microsoft Excel untuk mencatat lamanya proses dan ukuran *file* hasil kompresi. Lamanya proses kompresi selain dipengaruhi oleh ukuran *file*, juga dipengaruhi oleh isi untaian DNA yang ada dalam *file* itu sendiri. Hal ini bisa dilihat pada *file* I7.bmp dan I29.bmp. Walaupun keduanya memiliki ukuran *file* awal yang sama, tetapi waktu yang diperlukan dalam proses kompresi terlihat berbeda jauh.

4.3 Dekompresi menggunakan GenCompress dan konversi *file*

Pada bagian ini, *file* hasil kompresi didekompres dan kemudian dikonversi untuk menghasilkan *file* semula. Hasil dari proses ini adalah isi *file* yang sama seperti pada saat sebelum dikompres dan lama proses dekompres dan konversi. Hasil dari proses konversi dan dekompresi ditunjukkan pada Tabel 3.

Pada Tabel 3 terlihat bahwa sebagian besar *file* yang ada berhasil didekompres dan menghasilkan suatu *file* yang sama seperti saat sebelum dikompres. Waktu yang dibutuhkan untuk melakukan dekompresi terlihat lebih cepat dari proses kompresi yaitu sekitar 3 detik. Akan tetapi untuk *file* exe, proses dekompresinya tidak berhasil karena *software* GenDecompress mengalami error ketika dijalankan.

4.4 Perbandingan hasil kompresi GenCompress dan WinRAR

Pada bagian ini, hasil kompresi dengan menggunakan GenCompress dibandingkan dengan hasil kompresi dengan menggunakan WinRAR. Berdasarkan Tabel 4, terlihat hasil kompresi *file* gif, avi, txt, dan sebagian gambar pola dengan menggunakan GenCompress memiliki ukuran lebih kecil dibandingkan dengan menggunakan WinRAR.

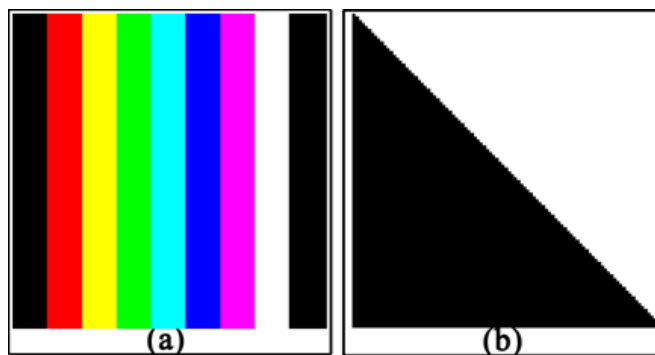
4.5 Pembahasan

Berdasarkan hasil yang diperoleh, teknik kompresi GenCompress terbukti mampu untuk melakukan kompresi pada semua jenis *file*. Untuk *dataset* Brodatz, *file* avi, sebagian besar txt dan beberapa pola dalam format bmp, terlihat hasil

kompresi GenCompress menunjukkan hasil yang lebih baik dari WinRAR. Sedangkan untuk *file* exe, wav dan bmp, hasil kompresinya kurang baik.

Jika melihat hasil kompresi dari beberapa pola dalam format bmp, terlihat beberapa sampel memiliki hasil yang baik sedangkan yang lainnya memiliki hasil yang kurang baik. Sampel pola yang memiliki hasil kompresi yang lebih baik daripada WinRAR ditunjukkan pada Gambar 2(a), sedangkan sampel pola yang hasil kompresinya kurang baik pada Gambar 2(b). Untuk bentuk *byte* dalam hexadesimal dari *file* tersebut ditunjukkan pada Gambar 3.

Di dalam Gambar 2(a), jumlah *pixel* dari setiap warna memiliki nilai yang sama pada setiap barisnya. Oleh sebab itu, pada Gambar 3(a), *byte* penyusun Gambar 2(a) juga menunjukkan adanya pola pengulangan yang sama. Sedangkan pada Gambar 2(b), jumlah *pixel* warna hitam dan putih berbeda setiap baris. Pada *byte* penyusun Gambar 2(b) juga terlihat bahwa walaupun hanya terdiri dari angka 00 dan 01, tetapi panjang setiap pengulangan tersebut memiliki jumlah yang berbeda. Berdasarkan hasil tersebut maka dapat ditarik kesimpulan bahwa untuk suatu pola pengulangan yang sama, GenCompress memberikan hasil yang lebih baik dari WinRAR. Tapi sebaliknya, untuk pola pengulangan yang tidak sama jumlahnya, WinRAR memberikan hasil yang lebih baik dari GenCompress.



Gambar 2 Contoh sampel yang memiliki pengulangan

Pada pola pengulangan yang tidak teratur, WinRAR memberikan hasil yang lebih baik dari GenCompress karena WinRAR menggunakan metode hibrid yang berdasarkan pada teknik Lempel-Ziv-Storer-Szymanski (LZSS) dan *Prediction by partial matching* (PPM). LZSS menggunakan *dictionary method* sedangkan PPM menggunakan *statistical method*. Sedangkan, GenCompress hanya menggunakan *dictionary method* yang berdasarkan pada teknik LZ77. Maka, dapat disimpulkan bahwa untuk pengulangan yang tidak sama jumlahnya, teknik kompresi WinRAR lebih baik karena WinRAR menggunakan metode hibrid

yang dapat mengompres secara statistik. Akan tetapi untuk jumlah pengulangan yang sama, teknik kompresi GenCompress lebih baik, karena semua untaian DNA yang disimpan memiliki peluang yang sama.

Tabel 2 Hasil kompresi dengan menggunakan aplikasi GenCompress

Nama file	Ukuran file asli (byte)	Ukuran file setelah dikompresi (byte)	Lama kompresi	Gencompress ratio	Compress ratio
D18.gif	359.434	353.928	13 detik	75,375173%	98,468147%
D42.gif	350.969	341.822	14 detik	75,643433%	97,393787%
D66.gif	295.743	291.468	9 detik	75,351995%	98,554488%
D83.gif	366.538	360.509	14 detik	75,403642%	98,355150%
D109.gif	290.104	282.665	11 detik	75,631498%	97,435747%
I4.bmp	307.254	245.780	18 detik	80,001888%	79,992449%
I7.bmp	307.254	211.801	8 detik	82,766620%	68,933521%
I16.bmp	307.254	249.571	15 detik	79,693430%	81,226282%
I22.bmp	307.254	289.757	16 detik	76,423659%	94,305363%
I29.bmp	307.254	164.086	127 detik	86,648994%	53,404024%
W7.Wav	88.236	77.369	7 detik	78,078959%	87,684165%
W14.Wav	214.828	138.855	101 detik	83,841143%	64,635429%
W20.Wav	100.780	71.074	27 detik	82,369022%	70,523913%
W22.Wav	23.308	19.830	2 detik	78,730479%	85,078085%
W27.Wav	353.836	222.823	300 detik	84,256619%	62,973524%
A1.avi	177.288	160.305	68 detik	77,394832%	90,420671%
A6.avi	367.796	349.099	43 detik	76,270881%	94,916475%
A16.avi	554.584	536.214	41 detik	75,828098%	96,687607%
A22.avi	438.934	422.074	54 detik	75,960281%	96,158876%
A29.avi	337.342	323.590	39 detik	76,019144%	95,923425%
T1.txt	33	37	0 detik	71,969697%	112,121212%
T7.txt	754	361	0 detik	88,030504%	47,877984%
T16.txt	578	293	0 detik	87,326990%	50,692042%
T22.txt	319	185	0 detik	85,501567%	57,993730%
T30.txt	689	376	0 detik	86,357039%	54,571843%
E2.exe	15.663	5.676	7 detik	90,940433%	36,238269%
E13.exe	15.838	5.948	7 detik	90,611188%	37,555247%
E16.exe	16.632	6.188	7 detik	90,698653%	37,205387%
E23.exe	15.661	5.740	7 detik	90,837111%	36,651555%
E30.exe	15.691	5.827	6 detik	90,716016%	37,135938%
P2.bmp	11.078	56	4 detik	99,873623%	0,505506%
P10.bmp	11.078	827	3 detik	98,133688%	7,465246%
P26.bmp	11.078	109	1 detik	99,754017%	0,983932%
P27.bmp	11.078	7.094	1 detik	83,990793%	64,036830%
P28.bmp	11.078	6.411	2 detik	85,532136%	57,871457%

Tabel 3 Proses dekompresi dan konversi *file*

<i>File hasil kompresi</i>	<i>Ukuran file kompres</i>	<i>Ukuran file output</i>	<i>Lama Dekompres</i>
D18.gif.GEN	353.928	359.434	3 detik
D42.gif.GEN	341.822	350.969	3 detik
D66.gif.GEN	291.468	295.743	2 detik
D83.gif.GEN	360.509	366.538	3 detik
D109.gif.GEN	282.665	290.104	2 detik
I4.bmp.GEN	245.780	307.254	1 detik
I7.bmp.GEN	211.801	307.254	1 detik
I16.bmp.GEN	249.571	307.254	2 detik
I22.bmp.GEN	289.757	307.254	2 detik
I29.bmp.GEN	164.086	307.254	1 detik
W7.Wav.GEN	77.369	88.236	0 detik
W14.Wav.GEN	138.855	214.828	1 detik
W20.Wav.GEN	71.074	100.780	0 detik
W22.Wav.GEN	19.830	23.308	0 detik
W27.Wav.GEN	222.823	353.836	1 detik
A1.avi.GEN	160.305	177.288	3 detik
A6.avi.GEN	349.099	367.796	3 detik
A16.avi.GEN	536.214	554.584	3 detik
A22.avi.GEN	422.074	438.934	3 detik
A29.avi.GEN	323.590	337.342	2 detik
T1.txt.GEN	37	33	0 detik
T7.txt.GEN	361	754	0 detik
T16.txt.GEN	293	578	0 detik
T22.txt.GEN	185	319	0 detik
T30.txt.GEN	376	689	0 detik
E2.exe.GEN	5.676	-	-
E13.exe.GEN	5.948	-	-
E16.exe.GEN	6.188	-	-
E23.exe.GEN	5.740	-	-
E30.exe.GEN	5.827	-	-
P2.bmp.GEN	56	-	-
P10.bmp.GEN	827	-	-
P26.bmp.GEN	109	-	-
P27.bmp.GEN	7.094	11.078	0 detik
P28.bmp.GEN	6.411	11.078	0 detik

Tabel 4 Perbandingan antara kompresi menggunakan GenCompress dan WinRAR

Nama file	Compress ratio GEN	Compress ratio RAR	Selisih*
D18.gif	98,468147%	98,854310%	0,386163%
D42.gif	97,393787%	97,845964%	0,452177%
D66.gif	98,554488%	99,099894%	0,545406%
D83.gif	98,355150%	98,748288%	0,393138%
D109.gif	97,435747%	97,933155%	0,497408%
I4.bmp	79,992449%	44,353857%	-35,638592%
I7.bmp	68,933521%	20,557910%	-48,375611%
I16.bmp	81,226282%	40,951786%	-40,274496%
I22.bmp	94,305363%	49,832386%	-44,472977%
I29.bmp	53,404024%	30,809037%	-22,594987%
W7.Wav	87,684165%	45,914366%	-41,769799%
W14.Wav	64,635429%	27,475469%	-37,159961%
W20.Wav	70,523913%	23,697162%	-46,826751%
W22.Wav	85,078085%	29,457697%	-55,620388%
W27.Wav	62,973524%	25,961180%	-37,012345%
A1.avi	90,420671%	90,991494%	0,570823%
A6.avi	94,916475%	95,207126%	0,290650%
A16.avi	96,687607%	96,932836%	0,245229%
A22.avi	96,158876%	96,437961%	0,279085%
A29.avi	95,923425%	96,260768%	0,337343%
T1.txt	112,121212%	357,575758%	245,454545%
T7.txt	47,877984%	48,938992%	1,061008%
T16.txt	50,692042%	56,920415%	6,228374%
T22.txt	57,993730%	76,802508%	18,808777%
T30.txt	54,571843%	56,313498%	1,741655%
E2.exe	36,238269%	26,310413%	-9,927856%
E13.exe	37,555247%	27,238288%	-10,316959%
E16.exe	37,205387%	26,575277%	-10,630110%
E23.exe	36,651555%	26,530873%	-10,120682%
E30.exe	37,135938%	26,760563%	-10,375375%
P2.bmp	0,505506%	1,281820%	0,776313%
P10.bmp	7,465246%	6,932659%	-0,532587%
P26.bmp	0,983932%	1,679003%	0,695071%
P27.bmp	64,036830%	42,390323%	-21,646507%
P28.bmp	57,871457%	37,037371%	-20,834086%

630	00 00 00 00 00 04 04 04 04 04 04 04 04 04 04	a0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
640	06 06 06 06 06 06 06 06 06 06 06 02 02 02 02	b00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
650	02 02 02 02 02 02 03 03 03 03 03 03 03 03 03	b10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
660	03 01 01 01 01 01 01 01 01 01 01 01 05 05 05 05	b20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
670	05 05 05 05 05 05 07 07 07 07 07 07 07 07 07	b30	01 01 01 01 01 01 01 01 01 01 01 01 01 01 00
680	07 07 00 00 00 00 00 00 00 00 00 00 00 00 00	b40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
690	00 00 00 00 00 00 00 00 04 04 04 04 04 04 04	b50	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6a0	04 04 04 04 06 06 06 06 06 06 06 06 06 06 02	b60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6b0	02 02 02 02 02 02 02 02 02 02 03 03 03 03 03	b70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6c0	03 03 03 03 01 01 01 01 01 01 01 01 01 01 01	b80	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6d0	05 05 05 05 05 05 05 05 05 05 07 07 07 07 07	b90	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6e0	07 07 07 07 07 07 00 00 00 00 00 00 00 00 00	ba0	01 01 00 00 00 00 00 00 00 00 00 00 00 00 00
6f0	00 00 00 00 00 00 00 00 00 00 00 00 04 04 04	bb0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
700	04 04 04 04 04 04 04 06 06 06 06 06 06 06 06	bc0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
710	06 06 06 02 02 02 02 02 02 02 02 02 02 03 03	bd0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
720	03 03 03 03 03 03 03 03 01 01 01 01 01 01 01	be0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
730	01 01 01 01 05 05 05 05 05 05 05 05 05 05 07	bf0	00 00 00 01 01 01 01 01 01 01 01 01 01 01 01
740	07 07 07 07 07 07 07 07 00 00 00 00 00 00 00	c00	01 01 01 01 01 01 00 00 00 00 00 00 00 00 00
750	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	c10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
760	00 04 04 04 04 04 04 04 04 04 04 06 06 06 06	c20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
770	06 06 06 06 06 06 02 02 02 02 02 02 02 02 02	c30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
780	02 02 03 03 03 03 03 03 01 01 01 01 01 01 01	c40	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
790	01 01 01 01 01 01 01 05 05 05 05 05 05 05 05	c50	00 00 00 00 00 00 01 01 01 01 01 01 01 01 01
7a0	05 05 05 07 07 07 07 07 07 07 07 07 07 00 00	c60	01 01 01 01 01 01 01 01 01 01 00 00 00 00 00
7b0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	c70	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Gambar 3 Isi dari Gambar 2 dalam bentuk *hexadesimal*

4.6 Masalah yang ditemui

Pada proses dekompresi, ada beberapa *file* hasil kompresi yang tidak dapat didekompres oleh GenDecompress. Dan jika melihat data yang ada, masalah tersebut terjadi pada *file* exe dan sebagian besar gambar pola yang berformat bmp. Masalah yang ditemui terletak pada *software* dekompresi GenDecompress dan bukan pada saat konversi DNA ke kode biner. Kemungkinan besar masalah tersebut terjadi karena terdapat sejumlah pengulangan yang cukup panjang pada isi *file* ketika dikompresi. Contoh pengulangan yang cukup panjang, ditunjukkan pada Gambar 4.

640	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
650	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
660	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
670	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
680	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
690	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6a0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6b0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6c0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6d0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6e0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
6f0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
700	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
710	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
720	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
730	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
740	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
750	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
760	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
770	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
780	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
790	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
7a0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
7b0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
7c0	01 01 01 01 01 01 01 01 01 01 01 01 01 01 01

Gambar 4 Isi dari *file* yang memiliki pengulangan yang cukup panjang

5. Kesimpulan

Setelah melakukan penelitian ini, maka dapat disimpulkan bahwa:

1. Proses konversi *file* media ke DNA terjadi dengan mengubah 00 menjadi a, 01 menjadi c, 10 menjadi g, dan 11 menjadi t. Ketika diubah ke dalam format DNA, ukuran *file* dalam format DNA menjadi 4 kali lebih besar dari ukuran *file* asal.

2. Setelah melakukan konversi ke format DNA, semua *file* tersebut dapat dikompres dengan menggunakan teknik kompresi DNA GenCompress. Rasio kompresi dari GenCompress menunjukkan hasil lebih dari 75% dan ukuran *file* hasil kompresi menjadi lebih kecil dari ukuran *file* sebelumnya.
3. Lama proses kompresi tidak hanya bergantung pada ukuran *file*, tetapi juga bergantung pada isi dari *file*.
4. GenDecompress tidak berhasil mendekompresi *file* exe dan beberapa pola gambar yang memiliki pengulangan cukup panjang pada isi *file*-nya.
5. GenCompress mampu untuk menghasilkan kompresi yang lebih baik ketika *file* tersebut berisi pola pengulangan yang sama, sedangkan WinRAR mampu untuk menghasilkan kompresi yang lebih baik ketika *file* tersebut berisi pola pengulangan yang tidak sama jumlahnya.

Daftar Pustaka

- Chen, X., Kwong, S. & Li, M., 1999. A Compression Algorithm for DNA Sequences and Its Applications in Genome Comparison. *Genome Informatics*, 10, pp. 51-61.
- Chen, X., Li, M., Ma, B. & Tromp, J., 2002. DNACompress: Fast and Effective DNA Sequence Compression. *Bioinformatics*, 18(12), pp. 1696-1698.
- Cherniavsky, N. & Ladner, R., 2004. *Grammar-based Compression of DNA Sequences*. Technical Report. Washington: University of Washington.
- Church, G. M., Gao, Y. & Kosuri, S., 2012. Next-Generation Digital Information Storage in DNA. *Science*, 337(6102), pp. 1628-1628.
- Grumbach, S. & Tahi, F., 1993. Compression of DNA Sequences. Proceeding. *IEEE Data Compression Conference 1993 (DCC'93)*, pp. 340-350.
- Grumbach, S. & Tahi, F., 1994. A New Challenge for Compression Algorithms: Genetic Sequences. *Information Processing & Management*, 30(6), pp. 875-886.
- Nelson, M. & Gailly, J. L., 1995. *The Data Compression Book*. 2nd edition. New York: M & T Books.