

Teknik Kompresi Citra Digital untuk Penyimpanan File menggunakan Format Data XML

Veronica Lusiana

Program Studi Teknik Informatika, Universitas Stikubank Semarang
email: verolusiana@yahoo.com

Abstrak

Berkas digital (*file*) yang disimpan hasil dari proses kompresi biasanya akan berbentuk *raw data*, informasi pendukung mengenai berkas tersebut yang biasa dikenal dengan *meta data* ikut diproses juga. Informasi meta data ini menjadi hilang, hal ini disebabkan proses kompresi pada berkas akan dikerjakan mulai dari karakter awal data sampai dengan karakter terakhir data (*end of file*). Untuk mengatasi hal tersebut maka akan dicoba untuk membuat berkas pendamping dari berkas yang dikompresi. Berkas pendamping ini disimpan dalam bentuk data XML (*Extensible Markup Language*).

Format XML adalah baku namun dapat didefinisikan secara fleksibel sesuai dengan keinginan pengguna. XML dapat mendeskripsikan isi berkas itu sendiri dan sekaligus berfungsi sebagai meta data. Pada penelitian ini akan menggunakan algoritma kompresi data Shannon-Fano. Algoritma ini memiliki karakteristik *lossless compression* yang bekerja berdasarkan frekuensi kemunculan karakter pada berkas untuk menyusun kode baru menggantikan kode ASCII. Kode baru ini digunakan untuk menyusun berkas terkompresi. Data yang digunakan adalah berkas citra digital. Penelitian ini akan membuat kamus data kode Shannon-Fano yang digunakan untuk proses kompresi berkas citra dan membuat struktur dokumen XML untuk menyimpan informasi (*meta data*) milik berkas citra tersebut.

Kata kunci: kompresi citra, kode ASCII, Shannon-Fano, XML, *lossless compression*

PENDAHULUAN

Berkas yang disimpan hasil dari proses kompresi akan berbentuk *raw data*, informasi pendukung mengenai berkas digital tersebut yang biasa dikenal dengan *meta data* ikut diproses juga, sehingga informasi ini menjadi hilang. Hal ini disebabkan proses kompresi pada berkas biasanya akan dikerjakan mulai dari *byte* awal data sampai dengan *byte* terakhir data (*end of file*). Untuk mengatasi hal tersebut maka akan dicoba hasil proses kompresi disimpan menggunakan bentuk data XML (*Extensible Markup Language*). Format XML adalah baku namun dapat didefinisikan secara fleksibel sesuai dengan keinginan pengguna. XML dapat mendeskripsikan isi berkas itu sendiri sekaligus berfungsi sebagai *meta data*.

Proses kompresi data menggunakan menggunakan algoritma Shannon-Fano yaitu jenis *lossless compression*. Algoritma ini banyak

diimplementasikan untuk penelitian yang berkaitan dengan kompresi data. Disini data yang digunakan adalah berkas citra digital. Algoritma Shannon-Fano digunakan untuk membuat kamus data kode biner Shannon-Fano yang digunakan untuk proses kompresi berkas citra, yang dilanjutkan dengan membuat struktur dokumen XML untuk menyimpan informasi (*meta data*) milik berkas citra tersebut. Manfaat dari penelitian ini adalah memudahkan pengguna untuk mengelola berkas citra dalam bentuk terkompresi, dengan cara pengguna masih dapat melakukan *preview* dokumen XML pada saat *browsing* berkas citra.

TINJAUAN PUSTAKA

Kompresi Citra

Proses kompresi bertujuan untuk mereduksi ukuran data dengan mempertimbangkan kualitas data yang masih memadai untuk dapat dinikmati. Pada umumnya

sebuah citra memiliki duplikasi data didalamnya, yaitu suatu piksel mempunyai intensitas warna yang sama dengan piksel-piksel tetangganya. Semakin besar ukuran citra maka kemungkinan terjadinya duplikasi ini menjadi semakin besar pula. Prinsip umum yang digunakan pada proses kompresi citra adalah mengurangi duplikasi data pada citra sehingga memori yang dibutuhkan untuk merepresentasikan citra menjadi lebih sedikit daripada representasi citra semula atau yang tidak terkompresi.

Penelitian yang berkaitan dengan kompresi data dilakukan oleh Kaimal, dkk (2013). Secara khusus menganalisa keuntungan dan kekurangan dari *lossles compression* dan *lossy compression* terhadap kompresi data audio, video, dan citra. Sementara itu Maan (2013) membandingkan beberapa buah algoritma kompresi jenis *lossles compression* yaitu algoritma Run-Length Encoding (RLE), Huffman Coding, dan Arithmetic Coding.

Kriteria Kompresi Citra

Kriteria yang digunakan dalam mengukur metode kompresi citra adalah (Munir 2004):

1. Waktu kompresi dan dekompresi

Waktu yang dibutuhkan untuk proses kompresi dan dekompresi.

2. Kebutuhan memori

Memori yang dibutuhkan untuk merepresentasikan citra.

3. Kualitas kompresi

Kualitas kompresi yang baik biasanya dicapai pada proses kompresi yang menghasilkan rasio kompresi tidak terlalu besar, disini terdapat hubungan antara kualitas citra dengan ukuran hasil kompresi.

4. Format keluaran

Format citra hasil kompresi sebaiknya cocok untuk proses pengiriman dan penyimpanan data.

Jenis Kompresi Citra

Empat pendekatan yang dapat digunakan pada proses kompresi citra (Munir 2004):

1. Pendekatan statistik

Kompresi citra berdasarkan frekuensi kemunculan intensitas warna atau derajat keabuan pada seluruh daerah citra. Contoh: Huffman Coding, Shannon-Fano.

2. Pendekatan ruang

Kompresi citra berdasarkan hubungan spasial antar piksel di dalam suatu kelompok yang memiliki intensitas warna atau derajat keabuan sama di dalam suatu daerah citra. Contoh: Run Length Encoding.

3. Pendekatan kuantisasi

Kompresi citra dilakukan dengan mengurangi intensitas warna atau derajat keabuan yang dimiliki oleh citra. Contoh: kompresi kuantisasi.

4. Pendekatan fraktal

Kompresi citra berdasarkan pada kemiripan bagian-bagian pada citra yang dapat dieksploitasi menggunakan suatu matriks transformasi. Contoh : *Fractal Image Compression*.

Sifat kompresi dan dekompresi citra

Metode lossless compressions

Citra hasil kompresi dapat dikembalikan seperti kondisi semula melalui proses dekompresi. Disini tidak ada informasi yang hilang akibat dari proses kompresi dan dekompresi.

Metode Lossy Compression

Citra hasil kompresi tidak dapat dikembalikan seperti kondisi semula pada saat proses dekompresi. Disini ada informasi yang hilang akibat dari kedua proses tersebut namun masih relatif baik atau masih dapat diterima oleh indera pengelihatan kita.

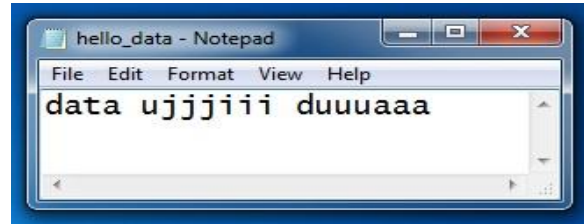
Algoritma Shannon-Fano

Algoritma Shannon-Fano dikembangkan secara bersamaan oleh Claude Shannon dari Bell Labs dan Robert Fano dari MIT. Proses kompresi data berdasarkan pada probabilitas kemunculan setiap karakter pada berkas. Berkas ini digunakan untuk menyimpan serangkaian karakter yang memiliki panjang bit tetap untuk

setiap karakternya yaitu 8 bit sesuai dengan kode ASCII (American Standard Code for Information Interchange). Pembentukan kode biner Shannon-Fano pengganti adalah unik untuk setiap karakter yang muncul. Karakter dengan probabilitas kemunculan yang tinggi memiliki kode pengganti dalam jumlah bit yang lebih pendek, demikian sebaliknya (Putra, 2010; Salomon, 2010; Thyagarajan, 2011). Berikut ini adalah langkah-langkah untuk membentuk kode pengganti berdasarkan algoritma Shannon-Fano.

1. Mencatat daftar frekuensi kemunculan simbol ASCII dari seluruh data.
2. Urutkan secara descending (karakter dengan frekuensi kemunculan tertinggi sampai dengan yang terendah) daftar yang diperoleh dari langkah nomor 1.
3. Bagilah menjadi dua bagian daftar yang diperoleh dari langkah nomor 2. Pembagian berdasarkan pada jumlah total frekuensi suatu bagian (akan menjadi bagian atas) sedekat mungkin dengan jumlah total frekuensi bagian yang lain (akan menjadi bagian bawah).
4. Daftar bagian atas dinyatakan dengan digit 0 dan bagian bawah dinyatakan dengan digit 1. Kode karakter pada bagian atas dimulai dengan 0 dan kode karakter pada bagian bawah dimulai dengan 1.
5. Kerjakan proses secara rekursif untuk langkah 3 dan 4 pada bagian atas dan bawah, sampai dengan setiap karakter telah memiliki kodenya. Tambahkan digit 0 atau 1 sesuai dengan posisi bagiannya.

Karakter yang sering muncul atau memiliki frekuensi tinggi di dalam berkas citra akan dikodekan dengan jumlah bit yang lebih sedikit, sebaliknya untuk karakter yang jarang muncul dikodekan dengan jumlah bit yang lebih panjang. Setiap kode yang dihasilkan adalah bersifat kode prefix unik yaitu tidak ada kode biner pengganti setiap karakter yang sama antara satu dengan lainnya. Sebagai contoh, rasio kompresi terhadap berkas hello_data.txt beserta proses untuk memperoleh kode pengganti Shannon-Fano dapat dilihat pada Gambar 2 dan Gambar 3. Isi berkas hello_data.txt dapat dilihat pada Gambar 1.



Gambar 1. isi berkas hello_data.txt

```

--- kompresi data menggunakan metode Shannon Fano (kode SF)---
nama_file= D:\sf_xml\hello_data.txt
panjang_data= 20
variasi_karakter= 7
jumlah_probabilitas= 1

no. - ASCII8bit Frek. prob. - kode SF (nbit) rasio1 rasio2
1 97 a 5 0.250 00..... 2 75.000 18.750
2 117 u 4 0.200 01..... 2 75.000 15.000
3 105 i 3 0.150 100..... 3 62.500 9.375
4 106 j 3 0.150 101..... 3 62.500 9.375
5 32 2 0.100 110..... 3 62.500 6.250
6 100 d 2 0.100 1110..... 4 50.000 5.000
7 116 t 1 0.050 1111..... 4 50.000 2.500
rasio1 (%) pemampatan per karakter:
100-(nbit/8*100)
rasio2 (%) pemampatan per karakter terhadap seluruh data:
rasio1*(frek/panjang_data)
rasio (%) pemampatan data (sigma rasio2) atau
100-((sigma(frek*nbit)/(panjang_data*8))*100) = 66.25
    
```

Gambar 2. Rasio kompresi terhadap berkas hello_data.txt

```

D:\000sf_xml\hello_data_catatan_stepbystep - Copy.txt 11 Agustus 2014 16:18
--- membentuk kode biner metode Shannon fano ---

no: 0 kode: 0..... no: 0 kode: 00..... no: 0 kode: 00.....
no: 1 kode: 0..... no: 1 kode: 01..... no: 1 kode: 01.....
no: 2 kode: 1..... no: 2 kode: 10..... no: 2 kode: 100.....
no: 3 kode: 1..... no: 3 kode: 10..... no: 3 kode: 101.....
no: 4 kode: 1..... no: 4 kode: 11..... no: 4 kode: 110.....
no: 5 kode: 1..... no: 5 kode: 11..... no: 5 kode: 111.....
no: 6 kode: 1..... no: 6 kode: 11..... no: 6 kode: 111.....
n_round.....= 1 n_round.....= 3 n_round.....= 5

no: 0 kode: 00..... no: 0 kode: 00..... no: 0 kode: 00.....
no: 1 kode: 01..... no: 1 kode: 01..... no: 1 kode: 01.....
no: 2 kode: 1..... no: 2 kode: 100..... no: 2 kode: 100.....
no: 3 kode: 1..... no: 3 kode: 101..... no: 3 kode: 101.....
no: 4 kode: 1..... no: 4 kode: 11..... no: 4 kode: 110.....
no: 5 kode: 1..... no: 5 kode: 11..... no: 5 kode: 1110.....
no: 6 kode: 1..... no: 6 kode: 11..... no: 6 kode: 1111.....
n_round.....= 2 n_round.....= 4 n_round.....= 6
    
```

Gambar 3. Proses memperoleh kode pengganti Shannon-Fano

XML

XML (*Extensible Markup Language*) adalah bahasa *markup* untuk keperluan umum yang aturannya telah dibakukan oleh konsorsium W3C (*World Wide Web Consortium*, <http://www.w3.org/>). XML didisain untuk dapat

menyimpan data secara terstruktur, sehingga memudahkan untuk melakukan pertukaran data dan mudah untuk dibaca oleh mesin dan juga manusia. Struktur dokumen XML dapat dibuat secara fleksibel sesuai kebutuhan pengguna dengan tetap mengikuti aturan yang telah dibakukan (Hunter dkk., 2007).

XML merupakan pengembangan dari HTML (*Hyper Text Markup Language*) yaitu kumpulan perintah dalam bentuk skrip yang telah dikenal luas untuk menyusun *website*. Dokumen XML menyimpan data di dalam tag-tag yang dapat didefinisikan dalam DTD (Document Type Definition) atau XML Schema. Tag yang telah didefinisikan dapat digunakan untuk memberikan makna pada bagian dari data yang ada di dalam sebuah dokumen XML.

Struktur dokumen XML yang diusulkan adalah seperti dapat dilihat pada Gambar 4. Melalui struktur dokumen tersebut diharapkan dapat digunakan untuk menyimpan informasi jenis berkas, nama berkas, dan keterangan singkat penjelasan isi berkas. Dua tag yang terakhir yaitu <preview> dan <data> digunakan untuk menyimpan data citra berukuran 64x64 piksel untuk keperluan *preview* dan menyimpan kamus kode biner metode Shannon-Fano untuk proses kompresi.

Penelitian yang pernah dilakukan oleh Grun (2010) adalah menganalisa data XML yang berukuran besar khususnya pada proses menyimpan dan *query*. Proses transaksi data menggunakan perangkat lunak manajemen basis

data XML yaitu BaseX. Penelitian Grun tidak secara spesifik menggunakan data citra.

```
D:\sf_xml\general template.xml
<?xml version="1.0" encoding="UTF-8"?>
<datacompressed>
<typeoffile>
<!--jenis file teks|citra|suara|video -->
</typeoffile>
<name>
<!--nama dan ekstensi berkas -->
</name>
<description>
<!--deskripsi atau keterangan singkat berkas -->
</description>
<preview>
<!--untuk preview (icon 64x64 piksel) -->
</preview>
<data>
<!--kamus data (kode biner untuk proses kompresi) -->
</data>
</datacompressed>
```

Gambar 4. Struktur dokumen XML

PROSES PENELITIAN

Proses penelitian dibagi menjadi dua tahap, yaitu:

1. Menguji kode biner yang dihasilkan oleh algoritma kompresi Shannon-Fano dan menghitung rasio kompresi terhadap berkas citra.
2. Membuat struktur dokumen XML yang dapat digunakan untuk menyimpan kode biner Shanon-Fano dan deskripsi berkas.

Bahan penelitian berupa berkas citra digital dalam format BMP dan JPG. Sebagai pembandingan, kode Shanon-Fano akan diterapkan

Tabel 1. Keterangan berkas citra

no	nama berkas (file) citra	format (ekstensi)	dimensi (piksel)	panjang data (karakter/ bytes)	deskripsi isi citra (keterangan)
1	citra02	bmp	251 x 256	193.590	Tingkat kecerahan sedang, variasi obyek tinggi, variasi tekstur sedang
2	citra02	jpg	251 x 256	35.040	
3	citra04	bmp	256 x 192	147.510	Tingkat kecerahan tinggi, variasi obyek tinggi, variasi tekstur tinggi
4	citra04	jpg	256 x 192	49.854	
5	citra08	bmp	192 x 256	147.510	Tingkat kecerahan tinggi, variasi obyek sedang, variasi tekstur sedang
6	citra10	bmp	256 x 192	147.510	Tingkat kecerahan tinggi, variasi obyek sedang, variasi tekstur sedang
7	citra10	jpg	256 x 192	30.635	

untuk proses kompresi pada berkas teks yang berukuran lebih kecil dibandingkan berkas citra. Pada Tabel 1 berisi keterangan berkas teks dan berkas citra. Beberapa contoh citra percobaan dapat dilihat pada Gambar 5.



Citra02



Citra04



Citra08



Citra10

Gambar 5. Citra percobaan

HASIL DAN PEMBAHASAN

Membentuk Kode Shannon-Fano dan menghitung Rasio Pemampatan

Berikut ini adalah kode Shannon-Fano dan rasio pemampatan berkas menggunakan contoh berkas hello_data.txt tampak pada gambar 6.

nama_berkas hello_data.txt									
data = "data ujjiiii duuuuaaa" (tanpa tanda petik)									
panjang_data= 20									
variasi_karakter= 7									
no.	- ASCII8bit	Frek.	prob.	- kode SF (nbit)	rasio1	rasio2			
1	97	a	5	0.25	00.....	2	75.00	18.750	
2	117	u	4	0.20	01.....	2	75.00	15.000	
3	105	i	3	0.15	100.....	3	62.50	9.375	
4	106	j	3	0.15	101.....	3	62.50	9.375	
5	32		2	0.10	110.....	3	62.50	6.250	
6	100	d	2	0.10	1110.....	4	50.00	5.000	
7	116	t	1	0.05	1111.....	4	50.00	2.500	

Gambar 6. Contoh Rasio Pemampatan Berkas

Karakter "a" muncul sebanyak 5 kali di dalam data yang memiliki panjang 20 karakter, sehingga memiliki probabilitas $5/20 = 0.25$. Karakter "u" muncul sebanyak 4 kali sehingga memiliki probabilitas $4/20 = 0.20$, demikian seterusnya. Setelah diurutkan berdasarkan frekuensi atau probabilitas dari yang terbesar sampai dengan terkecil maka diperoleh urutan

karakter a, u, i, j, (spasi), d, dan t. Kode pengganti Shannon-Fano diperoleh dengan mengikuti langkah-langkah seperti dapat dilihat pada Gambar 3 yaitu proses memperoleh kode pengganti Shannon-Fano.

Panjang kode Shannon-Fano (nbit kode_SF) yang dihasilkan bervariasi antara 2 sampai dengan 4 bit. Untuk kode_sf "00" digunakan untuk menyimpan atau menggantikan karakter "a" yang memiliki kode ASCII (American Standard Code for Information Interchange) "0110001", kode_sf "01" digunakan untuk menyimpan atau menggantikan karakter "u" yang memiliki kode ASCII "01110101" demikian seterusnya sampai dengan kode_sf "1111" untuk karakter "t" yang memiliki kode ASCII "01110100". Kode ASCII mengkodekan secara unik 256 macam karakter dengan panjang yang sama yaitu 8 bit per karakter. Sekilas tampak bahwa fungsi pemampatan terjadi dengan menggantikan panjang kode 8 bit menjadi panjang kode yang lebih kecil. Jika sering ditemukan sebuah karakter atau sekumpulan karakter yang memiliki frekuensi kemunculan tinggi maka secara umum rasio pemampatan data menjadi semakin tinggi. Rasio pemampatan data dapat dihitung sebagai berikut:

- Rasio 1 (%) adalah pemampatan setiap karakter

$$100 - (\text{nbit}/8 * 100)$$
- Rasio 2 (%) adalah pemampatan setiap karakter terhadap seluruh data

$$\text{rasio1} * (\text{frekuensi}/\text{panjang_data})$$
- Rasio (%) adalah pemampatan berkas atau seluruh data

$$\text{sigma} \quad \text{rasio2} \quad \text{atau} \quad 100 - ((\text{sigma}(\text{frekuensi} * \text{nbit}) / (\text{panjang_data} * 8)) * 100)$$

HASIL PENELITIAN

Rangkuman hasil penelitian dapat dilihat pada Tabel 2. Dari tabel tersebut diperoleh 4 buah citra format BMP memiliki rasio pemampatan lebih tinggi dari 3 buah citra format JPG. Rasio pemampatan citra format BMP

adalah citra10_256bmp sebesar 27,3% (terbesar), citra08_256bmp sebesar 5,55%, citra04_256bmp sebesar 3,78%, dan citra02_256bmp sebesar 3,29% (terkecil).

Isi citra10_256bmp adalah sebuah obyek tunggal stupa candi dengan warna obyek tidak terlalu bervariasi dan memiliki tekstur cenderung seragam. Pada latar belakang adalah langit cerah dengan dominan warna putih. Terdapat sebuah karakter dengan kode ASCII 255 yang memiliki frekuensi muncul sangat tinggi yaitu 43.179 akan dikodekan dengan kode_sf 2 bit yaitu "00". Dilihat dari ukuran berkas (panjang data) yaitu 147.510 maka lebih dari 25%-nya adalah karakter kode ASCII 255 ini. Dilihat dari hasil pembentukan kode_sf, maka kode_sf yang dihasilkan cukup panjang yaitu 9 sampai dengan 18 bit dengan total frekuensi kemunculan tidak terlalu besar.

Pada karakter yang memiliki kode_sf dengan panjang 8 bit akan menghasilkan rasio1 dan rasio2 sama dengan 0. Hal ini terjadi karena panjang bit kode_sf sama dengan panjang bit kode ASCII sehingga tidak terjadi unsur pemampatan untuk karakter tersebut. Sementara itu untuk karakter yang memiliki kode_sf lebih panjang dari 8 bit menghasilkan rasio1 dan rasio2 negatif. Disini tidak terjadi pemampatan data tetapi justru sebaliknya, karena panjang bit kode_sf melebihi panjang bit kode ASCII.

Pada Tabel 2 mencatat 3 buah citra format JPG yang memiliki rasio pemampatan relatif kecil dengan satu diantaranya negatif. Berkas citra02_256jpg memiliki rasio pemampatan data -0,18% yang berarti ukuran berkas baru tidak menjadi lebih kecil namun bertambah besar walaupun hanya 0,18%.

KESIMPULAN DAN SARAN

Melalui hasil penelitian maka diperoleh kesimpulan dan saran sebagai berikut:

- Struktur dokumen XML dapat digunakan untuk menyimpan kode Shanon-Fano, data citra berukuran 64x64 piksel dan keterangan berkas (*meta data*).

Tabel 2. Hasil rasio pemampatan data

no	nama berkas (file)	panjang data (karakter)	variasi karakter	jangkauan nbit kode Shannon-Fano	frekuensi tertinggi #1 - (nbit)	frekuensi tertinggi #2 - (nbit)	frekuensi tertinggi #3 - (nbit)	rasio (%) pemampatan data
1	hello_ABCDE.txt	39	5	2-3	15 (2 bit)	7 (2 bit)	6 (2 bit)	71.47
2	hello_data.txt	20	7	2-4	5 (2 bit)	4 (2 bit)	3 (3 bit)	66.25
3	hello_sf.txt	60	20	3-6	9 (3 bit)	6 (3 bit)	6 (3 bit)	49.58
4	citra10_256.bmp	147.510	254	2 - 18	43.179 (2 bit)	2.198 (6 bit)	1.571 (6 bit)	27.30
5	citra08_256.bmp	147.510	252	6-18	1.331 (6 bit)	1.326 (7 bit)	1.324 (7 bit)	5.55
6	citra04_256.bmp	147.510	256	6 - 11	2.811 (6 bit)	1.300 (6 bit)	1.291 (7 bit)	3.78
7	citra02_256.bmp	193.590	252	6 - 17	3.930 (6 bit)	3.482 (6 bit)	2.690 (6 bit)	3.29
8	citra10_256.jpg	30.635	256	5 - 9	1.594 (5 bit)	288 (5 bit)	212 (7 bit)	1.11
9	citra04_256.jpg	49.854	256	5-9	1.722 (5 bit)	343 (7 bit)	299 (7 bit)	0.50
10	citra02_256.jpg	35.040	256	6 - 10	549 (6 bit)	204 (7 bit)	203 (8 bit)	-0.18

- Struktur dokumen XML yang dihasilkan merupakan berkas pendamping untuk berkas citra yang terkompresi. Dokumen XML ini dapat mengakomodasi fasilitas *preview*.
- Kode Shannon-Fano yang dihasilkan memiliki variasi jangkauan n-bit yang tidak berhubungan dengan ukuran berkas dan variasi karakter yang dimiliki oleh berkas tersebut.
- Variasi karakter, ukuran berkas (panjang data) tidak berbanding lurus dengan rasio pemampatan data.
- Frekuensi munculnya karakter yang tinggi dengan pengganti kode Shannon-Fano yang pendek akan menghasilkan rasio pemampatan data yang tinggi.
- Pengembangan lebih lanjut dari penelitian ini adalah membangun program sejenis *windows explorer* atau *file manager* yang terintegrasi didalamnya dengan kemampuan untuk membentuk kode Shannon-Fano, proses enkripsi dekripsi, dan *preview* dokumen XML.

DAFTAR PUSTAKA

Grun, C. (2010). Storing and Querying Large XML Instances. Department of computer

science and information science. Mathematics and Natural Sciences Section Universitat Konstanz.

Hunter, D., Rafter, J., Fawcett, J., Vlist, E.van der, Ayers, D., Duckett, J., Watt, A., and McKinnon, L. (2007). *Beginning XML*. 4th Edition, Wiley Publishing, Indianapolis, USA.

Kaimal, A.B., Manimurugan, S., Devadass, C.S.C. (2013). Image Compression Techniques: A Survey. *International Journal of Engineering Inventions*. pp. 26-28, Volume 2, Issue 4 (February 2013).

Maan, A.J. (2013). Analysis and Comparison of Algorithms for Lossless Data Compression. *International Journal of Information and Computation Technology*. pp. 139-14, Volume 3, Number 3.

Munir, R. (2004). *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Penerbit Informatika.

Putra, D. (2010). *Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

Salomon, D., Motta, G. (2010). *Handbook of Data Compression 5th edition*. Springer-Verlag, London.

Thyagarajan, K.S. (2011). *Still Image and Video Compression with Matlab*. John Wiley & Sons, Inc., Hoboken, New Jersey.