
TEKNIK OPTIMASI ANIMASI ADOBE FLASH

Ezrifal Sany, ST. M.Kom

Dosen Tetap STMIK Nurdin Hamzah Jambi, Jambi 36121

E-mail : ezrifalsany@yahoo.com

Abstract - Adobe Flash is one of the software to create an animation. Animation involves objects with complex motions, therefore the resulting file has a very large size, which causes the animation display slower. It takes a method in making an animation to minimize the resulting file size, which is called optimization technique. Optimization techniques make the animation file size becomes smaller, but the resulting animation quality remains the same. The qualities of the animation are determined by the amount of movement generated by an object. Animated optimization results will have the same animated qualities with animations that do not use optimizations if both have the same number of motions.

Keywords: Adobe Flash, Animate, Multimedia, Optimatio.

I. PENDAHULUAN

Kebutuhan manusia modern yang haus informasi telah memacu semangat para pengembang teknologi informasi untuk berlomba-lomba mencoba mengimbangi tuntutan tersebut. Permasalahan yang penting dalam dunia teknologi informasi adalah bagaimana cara mengolah data dari informasi-infomasi yang semakin besar dan kompleks tersebut, sehingga lebih cepat, mudah, aman, dan efisien baik dalam proses penyimpanannya maupun transfer data.

Untuk mengatasi permasalahan di atas salah satu solusinya adalah dengan cara mengompres data informasi tersebut sehingga berukuran lebih kecil dari ukuran semula dengan tanpa mengurangi isi dari data tersebut. Maka terciptalah berbagai macam algoritma mengenai kompresi data.

Algoritma kompresi data baik yang bersifat *lossy* maupun *lossless* telah banyak diketahui dan diteliti. Demikian pula halnya pengembangan dengan algoritma *lossless Compression*, yang mana untuk jenis algoritma ini dipakai untuk keperluan

transfer data-data penting yang mengharuskan tidak adanya kehilangan data. Hal tersebut telah memunculkan berbagai algoritma *lossless* baru yang memiliki kinerja dan kualitas yang beragam. Menurut David Salomon (2000), pada prinsipnya kompresi data dapat dicapai dengan mereduksi *redundancy*. Terdapat banyak metode untuk untuk kompresi data yang sudah dikenal saat ini. Metode-metode ini berdasarkan pada ide-ide yang berbeda, dan cocok untuk tipe data yang berbeda pula, serta hasil yang berbeda. Meskipun demikian, semua metode tersebut berdasarkan pada prinsip yang sama yakni semuanya memampatkan data dengan menghilangkan *redundancy* dari data asli dalam *file* sumber.

Salah satu cara penyajian sistem informasi adalah dengan menggunakan teknik animasi, karena dengan menggunakan teknik ini penyajian sistem informasi akan lebih menarik dan mampu menambah daya tarik dari konsumen. Animasi digunakan agar tampilan informasi menjadi lebih menarik.

Macromedia FlashMx merupakan salah satu perangkat lunak untuk membuat suatu animasi [1]. Animasi melibatkan objek dengan gerakan yang kompleks, oleh karena itu file yang dihasilkan memiliki ukuran yang sangat besar, yang menyebabkan tampilan animasi menjadi lebih lambat. Diperlukan suatu metode dalam pembuatan suatu animasi untuk memperkecil ukuran file yang dihasilkan, yang disebut dengan teknik optimasi. Teknik Optimasi yang dilakukan dalam pembuatan animasi agar ukuran file menjadi lebih kecil, tetapi kualitas animasi yang dihasilkan tetap sama.

Kualitas animasi ditentukan oleh jumlah gerakan yang dihasilkan oleh suatu objek. Animasi hasil optimasi akan memiliki kualitas animasi yang sama dengan animasi yang tidak menggunakan optimasi yang jika keduanya memiliki jumlah gerakan yang sama.

Dengan dibuatnya penelitian ini diharapkan dapat memberikan informasi mengenai metode pembuatan animasi dengan ukuran file yang lebih kecil, karena selama ini pemanfaatan fasilitas flash masih terbatas cara untuk mempermudah pembuatan animasi.

II. TINJAUAN PUSTAKA

2.1 Metode Huffman

Pandangan dasar dari *source coding* adalah menghilangkan redundansi dari sumber. *Source Coding* menghasilkan data kompresi dan mengurangi rate dari transmisi. Pengurangan dari rate transmisi dapat mengurangi biaya dari sambungan dan memberikan kemungkinan untuk pengguna berbagi dalam sambungan yang sama. Secara umum kita dapat mengkompres data tanpa menghilangkan informasi (*lossless source coding*) atau mengkompres data dengan adanya kehilangan informasi (*loss source coding*). Teori *Source Encoding* merupakan salah

satu dari ketiga teorema dasar dari teori informasi yang diperkenalkan oleh Shannon (1948) [3]. Teori *Source Encoding* mencanangkan sebuah limit dasar dari sebuah ukuran dimana keluaran dari sebuah sumber informasi dapat dikompresi tanpa menyebabkan probabilitas error yang besar. Kita telah mengetahui bahwa entropy dari sebuah sumber informasi merupakan sebuah ukuran dari isi informasi pada sebuah sumber. Sehingga, dari pendapat teori *source encoding* bahwa entropy dari sebuah sumber sangat penting. Persamaan 1 berikut adalah rumus entropy [2] .

$$H(X) = - \sum_{i=1}^m P_i \log_2 P_i \dots \dots \text{Persamaan 1}$$

Di mana :

-) Efisiensi dari sumber ditentukan oleh $H(X)/H(X)_{max}$
-) p_i merupakan probabilitas pada simbol ke-I,
-) $H(X)$ maksimum ketika sumber mempunyai simbol probabilitas yang sama.

Teori Shannon *Noiseless Source Coding* menyatakan bahwa nilai rata-rata dari simbol biner per keluaran sumber dapat digunakan untuk mendekati entropi dari sumber [2]. Dalam kata lain, efisiensi dari sumber dapat dihasilkan dari *source coding*. Untuk sumber dengan probabilitas simbol yang sama, dan/atau secara statistic tidak terikat satu dengan yang lain, maka dapat dipetakan (*encode*) setiap simbol dalam sebuah *codeword* dengan panjang n .

2.2 Algoritma Huffman

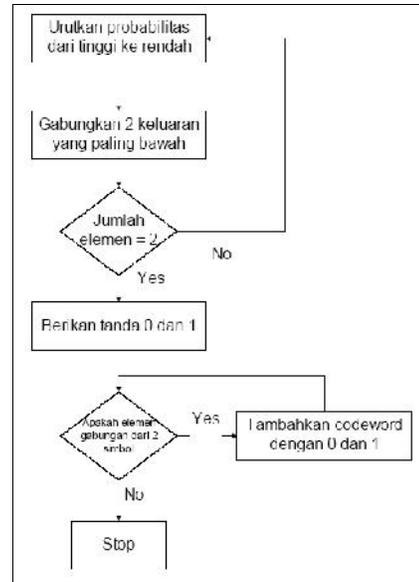
Dalam Huffman Coding, panjang blok dari keluaran sumber dipetakan dalam blok biner berdasarkan pajang variable. Cara seperti ini disebut sebagai *fixed to variable-length coding*. Ide dasar dari cara Huffman ini adalah memetakan mulai simbol yang paling banyak terdapat pada

sebuah urutan sumber sampai dengan yang jarring muncul menjadi urutan biner. Dalam *variable-length coding*, sinkronisasi merupakan suatu masalah. Ini berarti harus terdapat satu cara untuk memecahkan urutan biner yang diterima kedalam suatu *codeword* [2].

Seperti yang disebutkan diatas, bahwa ide dari *Huffman Coding* dalam memilih panjang *codeword* dari yang paling besar probabilitasnya sampai dengan urutan *codeword* yang paling kecil probabilitasnya. Apabila kita dapat memetakan setiap keluaran sumber dari probabilitas p_i ke sebuah *codeword* dengan panjang $1/p_i$ dan pada saat yang bersamaan dapat memastikan bahwa dapat didekodekan secara unik, kita dapat mencari rata-rata panjang kode $H(x)$. *Huffman Code* dapat mendekodekan secara unik dengan $H(x)$ minimum, dan optimum pada keunikan dari kode-kode tersebut. Algoritma dari *Huffman encoding* adalah [2]:

1. Pengurutan keluaran sumber dimulai dari probabilitas paling tinggi.
2. Menggabungkan 2 keluaran yang sama dekat kedalam satu keluaran yang probabilitasnya merupakan jumlah dari probabilitas sebelumnya.
3. Apabila setelah dibagi masih terdapat 2 keluaran, maka lanjut kelangkah berikutnya, namun apabila masih terdapat lebih dari 2, kembali ke langkah 1.
4. Memberikan nilai 0 dan 1 untuk kedua keluaran
5. Apabila sebuah keluaran merupakan hasil dari penggabungan 2 keluaran dari langkah sebelumnya, maka berikan tanda 0 dan 1 untuk *codeword*-nya, ulangi sampai keluaran merupakan satu keluaran yang berdiri sendiri.

Gambar 1 menggambarkan *flowchart* alur program Huffman code



Gambar 2.1 Alur Program Huffman Code

Contoh : Apabila kita mempunyai kalimat:

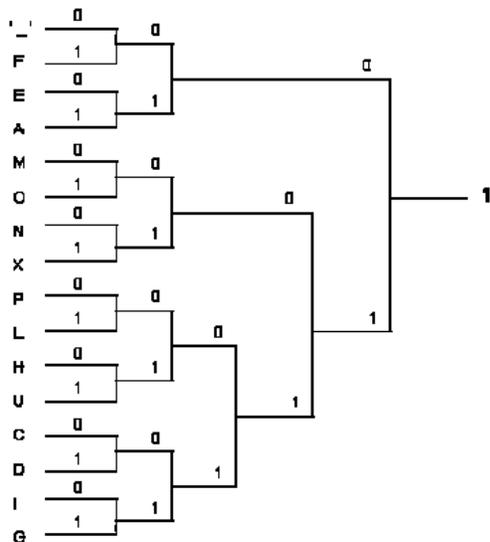
"EXAMPLE OF HUFFMAN CODING"

Pertama, kita kalkulasi probabilitas dari setiap simbol :

Tabel 1. Kalkulasi Probabilitas Dari Setiap Simbol

Simbol	Probabilitas
E	2/25
X	1/25
A	2/25
M	2/25
P	1/25
L	1/25
O	2/25
F	2/25
H	1/25
U	1/25
C	1/25
D	1/25
I	1/25
N	2/25
G	1/25
spasi	3/25

Huffman Tree dari kalimat tersebut dijelaskan pada Gambar 2 berikut ini



Gambar 2. Huffman Tree dari Hasil Kalkulasi

2.3 Teknik Optimasi Animasi

Teknik Optimasi Animasi adalah teknik untuk memperkecil ukuran file pada pembuatan animasi. Selain ukuran file, Kualitas animasi menjadi faktor utama dalam optimasi. Animasi optimal jika ukuran file yang dihasilkan lebih kecil dan kualitas animasi tidak berkurang. Kualitas animasi ditentukan oleh banyaknya gerakan yang dihasilkan oleh dua objek yang sama. Kualitas animasi sama jika jumlah gerakan objek hasil optimasi sama dengan sebelum optimasi. Jumlah gerakan objek direpresentasikan dalam jumlah *frame* yang digunakan karena satu *frame* mewakili satu gerakan. Sehingga jumlah gerakan dapat dilihat dari jumlah penggunaan *frame* [4].

Kualitas flash secara standar akan mengatur *actual rate* pada kecepatan 5.0 Kbps untuk menampilkan suatu *movie*. *Actual rate* adalah kecepatan rata-rata untuk menampilkan flash pada saat dilakukan publikasi animasi. Publikasi animasi adalah proses konversi file flash dari format .fla ke format .swf agar dapat dilihat oleh pengguna. Berdasarkan data Macromedia flash Player Statistic bahwa pengguna tidak akan menelusuri antar muka yang kompleks atau menunggu sistem informasi untuk

tampil selama lebih dari 30 detik, untuk itu diperlukan konversi maksimal agar ukuran file tidak melebihi dari perhitungan 5.0 KB perdetik x 30 detik yaitu 150 KB.

Kompresi file dilakukan sebelum dilakukan proses publikasi animasi. Jika file animasi membutuhkan file yang besar, maka secara langsung mempengaruhi performa kecepatan tampilan setelah dilakukan proses publikasi. Pada hakekatnya, Flash secara otomatis akan melakukan optimasi ketika dilakukan proses publikasi, misalnya mendeteksi bentuk ganda hanya satu bentuk yang akan disimpan. Flash juga secara otomatis akan melakukan pengelompokan terhadap setiap objek yang mungkin untuk dikelompokkan. Ukuran file yang besar menyebabkan waktu tanggap menjadi lebih lama. Lebih lanjut tentang Adobe Flash, bahwa besar file yang dihasilkan Adobe Flash dipengaruhi oleh banyak animasi. Semakin banyak dimensi *frame* animasi yang akan ditampilkan maka akan semakin besar pula file yang akan dihasilkan. Untuk menyajikan tampilan animasi dengan lebih efisien dilakukan optimasi pada saat pembuatan tampilan animasi dan *movie*, yang dilakukan dengan optimasi yang dilakukan pada *tools* Adobe Flash dan membandingkannya dengan teknik metode huffman. Berdasarkan elemen animasi yang dibuat, dapat didefinisikan optimasi animasi dengan membuat acuan untuk hal berikut.

III. HASIL DAN PEMBAHASAN

3.1 Tahapan Optimasi

Tahap-tahap yang dilakukan dalam sebuah optimasi dibedakan sesuai dengan jenisnya:

-) Optimasi Movie Interaktif
-) Optimasi Objek
-) Optimasi Vektor dan warna
-) Optimasi Suara
-) Optimasi Action Script

3.2 Perbandingan Penggunaan Optimasi

Perbandingan dilakukan mengacu pada kebutuhan fungsional optimasi yang telah ditentukan sebelumnya, yaitu ukuran file dan kualitas animasi. Pada tahap ini, dibuat perbandingan animasi yang menggunakan dan tidak menggunakan optimasi. Langkah-langkah perbandingan animasi dengan optimasi dan tanpa optimasi adalah sebagai berikut :

1. Setelah optimasi dilakukan, batas-batas yang akan digunakan untuk menentukan perbandingan hasil optimasi yang dikonversi kedalam suatu parameter.
2. Masing-masing animasi yang dibuat dengan menggunakan teknik optimasi, dibandingkan dengan animasi yang tidak menggunakan optimasi.
 -) *Movie comparison*, perbandingan movie, yaitu membandingkan ukuran file dan kualitas animasi tweening dengan animasi frame by frame,
 -) *Object Comparison*, perbandingan objek yaitu membandingkan animasi dengan clip dan layering.
 -) *Action Script Comparison*, perbandingan actionscript yaitu membandingkan animasi yang menggunakan fasilitas panel action frame dengan animasi yang menggunakan frame by frame.
 -) *Vector Comparison*, perbandingan vektor yaitu membandingkan gambar vektor raw (kasar) dengan gambar vektor hasil smoothing yang dilihat dari hasil ukuran file.
 -) *Sound Comparison*, perbandingan suara yaitu membandingkan elemen suara yang mengalami proses cutting dan plain soundtrack atau suara yang tidak menggunakan teknik optimasi.

3.3 Hasil Implementasi

Hasil dari implementasi metode Huffman dalam hal perbandingan animasi movie, objek, gambar vektor, suara dan action script dijelaskan pada Tabel 2 s.d Tabel 6 berikut ini

Tabel 2. Ringkasan Hasil Pengujian Movie

Animasi Movie	Ukuran File		Frame Range	
	Tweening	Frame	Tweening	Frame
Tampilan Pembuka	86KB	428KB	21-30	21-30
Menu Utama	98KB	747KB	45-60	45-60

Tabel 3. Ringkasan Pengujian Objek

Objek	Ukuran File		Frame Range	
	Clip	Layering	Clip	Layering
Tampilan Pembuka	504KB	819KB	1-13	1-13
Menu Utama	74KB	124KB	1-10	1-10

Tabel 4. Ringkasan Hasil Pengujian ActionScript

ActionScript	Ukuran File		Frame Range	
	Action frame	Frame	Action frame	Frame
Tampilan Pembuka	44KB	438KB	21-30	21-30
Menu Utama	49KB	732KB	45-60	45-60

Tabel 5. Ringkasan Hasil Pengujian Gambar Vektor

Elemen Vektor	Ukuran File		Frame Range	
	Smoothing	Raw	Smoothing	Raw
Tampilan Pembuka	126KB	168KB	1-20	1-20
Menu Utama	133KB	176KB	130-175	130-175

Tabel 6. Ringkasan Hasil Pengujian File Suara

Elemen Suara	Ukuran File		Frame Range	
	Cutting frame	Plain Sound Track	Cutting frame	Plain Sound Track
Tampilan Pembuka	126KB	168KB	1-20	1-20
Menu Utama	133KB	176KB	130-175	130-175

IDENTITAS PENULIS

Nama : Ezrifal Sany, ST. M.Kom
 NIDN/NIK : 1001068103/10.066
 TTL : Jambi, 01 Juni 1981
 Gol/Pangkat : III-B
 Jabatan Fungsional : Asisten Ahli
 Alamat : Jl. Kol. Abunjani Sipin,
 Kota Jambi
 Telp./Faks. : -
 Alamat Rumah : Komp BTN Karya Indah
 Telp. : 081366235550

IV. KESIMPULAN

Hasil pengujian menunjukkan bahwa animasi yang menggunakan teknik optimasi lebih efisien dalam penggunaan memori file untuk setiap tahap optimasi. Di sisi lain animasi hasil optimasi memiliki kualitas animasi yang sama baiknya dengan animasi yang tidak menggunakan teknik optimasi. Pada pengujian terhadap keseluruhan tahap optimasi, animasi dengan optimasi memiliki ukuran file yang jauh lebih kecil dibandingkan dengan animasi tanpa optimasi.

DAFTAR PUSTAKA

- [1] Andi, Adobe Flash 2004, MADCOM, Yogyakarta.
- [2] Andreas, Andi Suciadi, Adobe Flash 2004 dengan Action Script, 2005. P.T Elex Media Komputindo, Jakarta.
- [3] Sugiyanto David, Muhammad Farhan Sjaugi, Hasporo Renaldi N, Langkah Demi Langkah Membangun Website dengan PHP, 2003. Datakom Lintas Buana, Jakarta.
- [4] Warung Flash <http://warungflash.com/>. diakses pada 01 November 2013.