Perangkat Lunak Enkripsi Video MPEG-1 dengan Modifikasi *Video Encryption Algorithm* (VEA)

Tessa Ramsky
Laboratorium Ilmu dan Rekayasa Komputasi
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
email: if14124@students.if.itb.ac.id

Abstract – *Makalah ini membahas tentang rancangan dan implementasi model enkripsi pada video MPEG-1 dengan tujuan keamanan pada penyimpanan file video. Proses enkripsi pada video akan menghasilkan video dengan gambar yang acak. Sebaliknya, proses dekripsi akan mengembalikan video acak tersebut kembali menjadi video aslinya. Kunci yang digunakan pada proses enkripsi dan dekripsi ini harus sama, jika tidak proses dekripsi tidak akan mengembalikan video yang aslinya.*

Pada model enkripsi ini dipilih algoritma VEA (Video Encryption Algorithm) untuk diimplementasikan pada enkripsi video MPEG-1. Algoritma VEA umum digunakan karena dinilai ringan dan cocok untuk diterapkan pada video yang pada umunya berukuran besar. Untuk meningkatkan keamanan proses enkripsi, algoritma VEA yang ada dimodifikasi dengan menambahkan algoritma kriptografi DES dengan mode operasi CBC. Hal ini mengakibatkan operasi yang dilakukan bukan lagi bit per bit, melainkan per blok-blok slice dari gambar video. Blok-blok ini selanjutnya akan dienkripsi dengan menggunakan algoritma DES. Perangkat lunak ini dibangun dengan menggunakan bahasa pemograman Java dan kakas NetBeans 6.0.

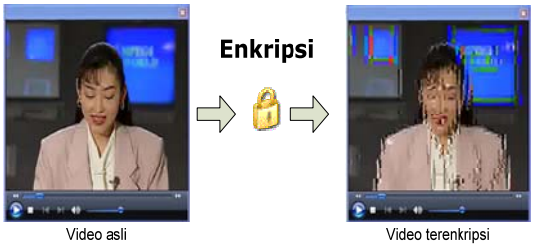
Kata Kunci*: VEA, DES, MPEG-1, CBC, enkripsi video*

1. PENDAHULUAN

Dewasa ini, multimedia telah berkembang sangat pesat dan digunakan secara luas di berbagai bidang. Akan tetapi perkembangan video ini menimbulkan berbagai permasalahan seperti penyalahgunaan akses dan penjiplakan yang telah menimbulkan dampak serius terhadap permasalahan legal, sosial, dan ekonomi. Tidak semua video yang ada dibuat untuk konsumsi masyarakat umum. Banyak dari video tersebut yang bersifat pribadi hanya ditujukan untuk kelompok tertentu saja. Teknologi baru telah meningkatkan kebutuhan akan keamanan multimedia serta perlindungan hak cipta. Hal ini mengakibatkan kebutuhan keamanan dalam penyimpanan video digital menjadi sangat penting.

Untuk itu diperlukan suatu teknik enkripsi yang dapat menjaga keamanan data multimedia tersebut.

Faktor penting yang harus diperhatikan dalam enkripsi video adalah efisiensi dan tingkat keamanan. Tidak seperti plainteks, enkripsi data multimedia memiliki ukuran data yang sangat besar. Sebagai contoh, ukuran sebuah video MPEG-1 yang berdurasi 2 jam berukuran sekitar 1 GB. Sedangkan faktor keamanan yang dimaksud ada dua, yaitu keamanan dari sisi algoritma serta keamanan dari sisi gambar video hasil enkripsi. Untuk itu, diperlukan suatu algoritma enkripsi video yang dapat mengatasi hal-hal tersebut. Gambaran enkripsi video dapat dilihat pada Gambar 1 [2].



Gambar 1 Enkripsi Video

Terdapat beberapa algoritma enkripsi video yang telah dibangun hingga saat ini, salah satunya adalah *Video Encryption Algorithm*, atau sering disebut dengan VEA. Alasan penggunaan algoritma ini adalah karena tingkat keamanannya yang cukup bagus dan komputasi yang ringan. Untuk meningkatkan faktor kemanan, maka VEA akan dimodifikasi dengan algoritma DES. Implementasi dari algoritma ini diharapkan dapat memenuhi kebutuhan keamanan penyimpanan video.

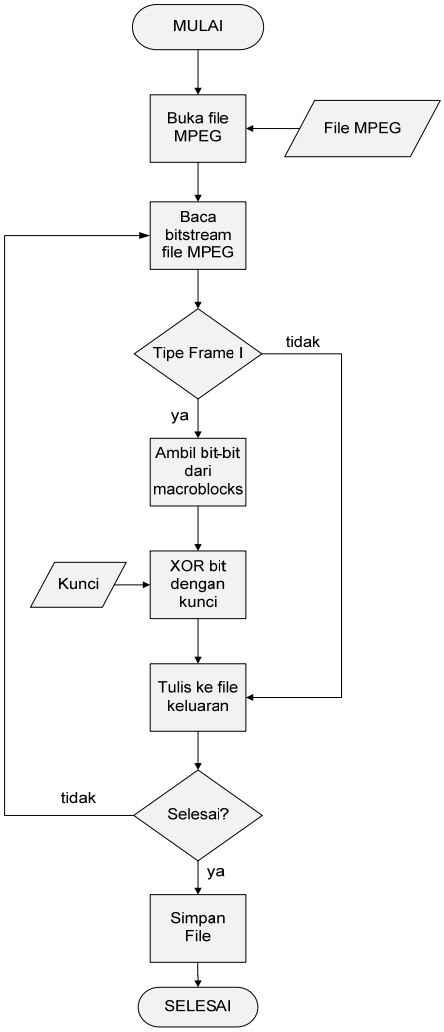
1.1. VEA

Algoritma VEA merupakan algoritma enkripsi video yang hanya beroperasi pada *sign bits* dari koefisien DCT pada *frame* I dari sebuah *file* video [1]. Algoritma VEA akan menghasilkan sebuah kunci rahasia *k* secara random dalam bentuk bitstream dengan panjang m yang dapat dituliskan sebagai $k = b1b2 \cdots bm$.

Berikut adalah skema global dari algoritma VEA:

1. Buka sebuah *file* video MPEG
2. Baca *stream* bit dari video MPEG tersebut bit per bit.
3. Setiap ketemu *frame*, baca tipe *frame* tersebut.
4. Baca *stream* bit dari *frame* tersebut.
5. Jika *stream* bit tersebut bukan berasal dari *frame* I, maka *stream* bit tersebut langsung ditulis ke *file* tujuan
6. Jika *stream* bit tersebut berasal dari *frame* I, maka *stream* bit tersebut di-*XOR*-kan dengan bit kunci yang berkorespondensi. Tulis hasil enkripsi ke *file* tujuan.
7. Lanjutkan pembacaan *stream* bit seperti langkah 4,5,dan 6 hingga akhir dari *frame*.
8. Kembali ke langkah 3 untuk membaca *frame* selanjutnya. Ulangi proses ini sampe akhir dari *file*.

Diagram alir dari algoritma VEA dapat dilihat pada Gambar 2.



Gambar 2 Diagram Alir VEA

Contoh hasil enkripsi menggunakan VEA dapat dilihat pada Gambar 3.

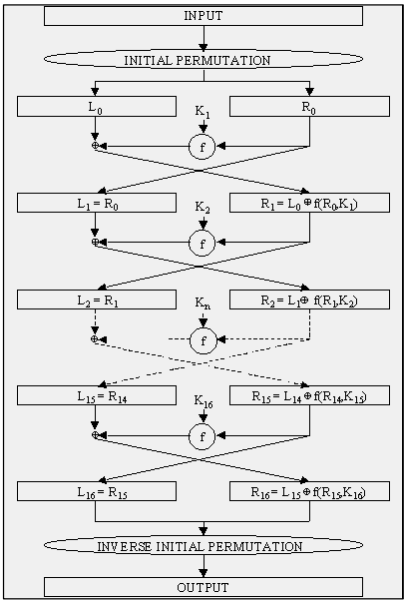
Gambar 3 Contoh hasil enkripsi VEA

1.2. DES

Skema enkripsi yang paling umum digunakan saat ini adalah *Data encryption Standard* (DES). DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit.

Skema global dari algoritma DES adalah sebagai berikut (lihat Gambar II-9):

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP)
2. Hasil permutasi awal kemudian di*enciphering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP-1) menjadi blok cipherteks.

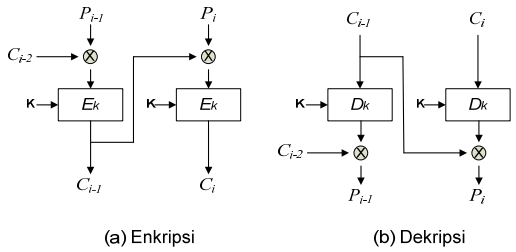


Gambar 4 Algoritma Enkripsi DES

1.3. CBC

CBC merupakan salah satu mode operasi *cipher* blok yang menerapkan umpan balik pada sebuah blok. Caranya, blok plainteks yang *current* di-*XOR*-kan terlebih dahulu dengan blok cipherteks hasil enkripsi sebelumnya, selanjutnya hasil peng-*XOR*-an ini masuk ke dalam fungsi enkripsi. Sedangkan pada proses dekripsi dilakukan dengan memasukkan blok cipherteks yang *current* ke fungsi dekripsi, kemudian meng-*XOR*-kan hasilnya dengan blok cipherteks sebelumnya. Dengan CBC, setiap blok cipherteks bergantung tidak hanya pada blok plainteksnya, tetapi juga pada seluruh blok plainteks sebelumnya. Gambar 5 memperlihatkan skema mode operasi CBC.

Pada mode CBC, blok-blok plainteks yang sama tidak menghasilkan blok-blok plainteks yang sama, sehingga kriptanalis menjadi lebih sulit. Hal ini menyebabkan mode CBC lebih banyak digunakan.



Gambar 5 Skema enkripsi dan dekripsi dengan mode CBC

2. MODIFIKASI ALGORITMA VEA

VEA sebenarnya merupakan algoritma enkripsi dengan waktu enkripsi yang pendek, namun memiliki tingkat keamanan yang relatif rendah. Proses enkripsi akan dilakukan pada video MPEG statik (bukan untuk video *streaming*) namun tetap selektif. Oleh karena itu, akan dilakukan modifikasi pada algoritma selektif VEA dengan menambahkan algoritma DES. Penambahan DES ini bertujuan untuk meningkatkan keamanan dari VEA yang lemah terhadap serangan kriptanalis. Algoritma DES yang ada juga akan dioperasikan pada mode operasi *cipher* blok CBC yang dimodifikasi untuk lebih meningkatkan keamanan DES tersebut.

Algoritma VEA yang ada akan dimodifikasi sesuai dengan kebutuhan model enkripsi video MPEG yang akan dirancang. Aliran data yang akan dienkripsi berupa bit-bit *macroblocks* dan kemudian sekumpulan *macroblocks* ini dienkripsi dengan menggunakan algoritma DES. Data-data selain *macroblocks* tidak akan dienkripsi dan langung ditulis ke *file output* sehingga struktur dari format *file* video tidak rusak. Selain itu, plainteks dan cipherteks yang diproses dari setiap blok akan digunakan dalam proses enkripsi blok yang berikutnya.

Modifikasi yang dilakukan terhadap algoritma VEA secara garis besar adalah sebagai berikut:

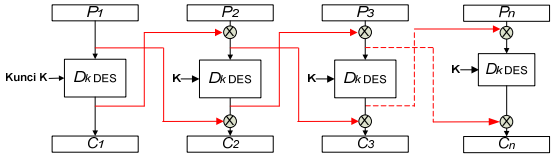
1. Enkripsi dilakukan pada *frame* I dan P dari MPEG.
   Pada VEA, enkripsi dilakukan pada *frame* I, hal ini dikarenakan *frame* I merupakan *frame* utuh tanpa referensi dari *frame* yang lainnya dan merupakan acuan untuk *frame* P dan B lainnya. Pada modifikasi ini, *frame* P ikut dienkripsi bersama dengan *frame* I, karena *frame* P merupakan *frame* yang mengacu kepada *frame* I serta *frame* P lainnya. Dengan mengenkripsi *frame* P dapat membuat gambar semakin acak. *Frame* B tidak dienkripsi karena *frame* B merupakan *frame* yang mengacu pada kedua *frame* P dan I, sehingga tanpa dienkripsi sudah cukup untuk membuat gambar menjadi acak. Selain itu, *frame* B memiliki jumlah yang lebih banyak dibandingkan *frame* I dan P (jika sebuah video memiliki ketiga tipe *frame*), sehingga tanpa mengikutsertakan *frame* B dalam proses enkripsi dapat mempersingkat waktu enkripsi.

2. Enkripsi dengan DES

   Enkripsi dilakukan pada setiap bit-bit *macroblocks* pada semua *slice* yang terdapat pada setiap *frame* I dan P. Setiap 64 bit dari *macroblocks* akan dienkripsi dengan DES. Pada algoritma VEA yang sebenarnya, hanya dilakukan operasi *XOR* pada *sign bits* dari koefisien DCT terhadap kunci.

3. Modifikasi CBC

   Algoritma DES yang ada dioperasikan pada mode enkripsi cipherblok CBC. Pada modifikasi CBC ini, proses *XOR* dilakukan sebanyak dua kali, yaitu saat sebelum dilakukan enkripsi dengan DES, dan setelah dilakukan enkripsi dengan DES.



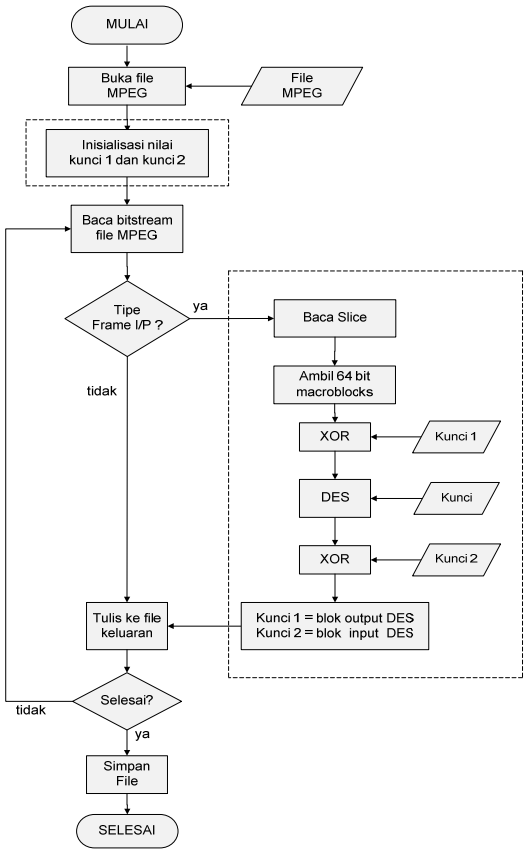
Gambar 6 DES dan modifikasi CBC

Berikut adalah skema global dari modifikasi algoritma VEA:

1. Buka *file* video MPEG.
2. Inisialisasi nilai kunci 1 dan kunci 2.
3. Baca *stream* bit dari video byte per byte hingga ditemukan *frame* header.
4. Cek tipe *frame*. Apabila *frame* merupakan *frame* I atau P, lanjutkan operasi algoritma. Apabila *frame* merupakan *frame* B, lanjutkan pembacaan *bitstream* hingga ditemukan *frame* I atau P.
5. Baca *bitstream* hingga ditemukan *slice*/data gambar (*picture data*).

6. Ambil 64 bit *macroblocks* dari data gambar tersebut.
7. *XOR*-kan 64 bit *macroblocks* tersebut dengan kunci 1.
8. Enkripsi dengan algoritma DES dengan kunci (kunci masukan user).
9. *XOR*-kan blok hasil DES dengan kunci 2, maka diperoleh blok cipherteksnya.
10. Ganti nilai kunci 1 dengan *bytes* blok sebelum DES, serta kunci 2 dengan *bytes* blok hasil DES. Kunci 1 dan 2 digunakan untuk *XOR* blok selanjutnya.
11. Ulangi proses 5 hingga 10 hingga seluruh *macroblocks* dari slices terenkripsi.
12. Lakukan kembali langkah 3 untuk mengenkripsi slices berikutnya hingga akhir *frame*.
13. Lakukan kembali langkah 2 hingga akhir dari *file*.

Diagram alir dari modifikasi algoritma enkripsi video yang dilakukan dapat dilihat pada Gambar III-3.



Gambar 7 Diagram alir algoritma modifikasi

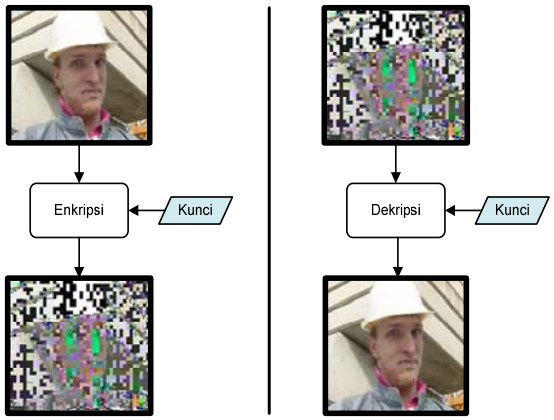
Dua operasi XOR ini menyebabkan algoritma ini sedikit lebih rentan terhadap serangan *key-search* dibandingkan dengan algoritma DES [3]. Dua buah kunci yang digunakan pada operasi *XOR* juga selalu berbeda untuk setiap plainteksnya dan bukan merupakan plainteks atau pun cipherteks sehingga membuat kriptanalis lebih sulit dalam memecahkan blok cipherteks.

3. ANALISIS

Perangkat lunak ini merupakan aplikasi yang berbasis desktop. Data yang digunakan dalam sistem secara umum dibagi menjadi dua, yaitu data untuk proses enkripsi dan data untuk proses dekripsi. Untuk kedua proses tersebut, data yang dilibatkan adalah sebagai berikut:

1. Video asli (plainteks)
   Video MPEG-1 digunakan sebagai data masukan pada proses enkripsi.
2. Video terenkripsi (cipherteks)
   Video ini merupakan video hasil enkripsi yang memiliki gambar yang rusak dan berbeda dari video yang sebenarnya. Video ini digunakan sebagai masukan pada proses dekripsi.
3. Kunci
   Kunci merupakan data rahasia yang digunakan untuk mengenkripsi dan mendekripsi *file* video. Kunci yang digunakan memiliki panjang 64 bit, yang disesuaikan dengan panjang kunci yang digunakan pada algoritma DES.

Perangkat lunak ini digunakan untuk melakukan enkripsi terhadap suatu *file* video MPEG sesuai dengan kunci yang didefinisikan oleh user. Hasil yang diperoleh adalah sebuah *file* video baru terenkripsi dengan format yang sama yang akan menghasilkan gambar yang kabur atau acak apabila dijalankan. Pada proses dekripsi, apabila kunci yang dimasukkan sama dengan kunci pada saat enkripsi maka akan menghasilkan video yang sebenarnya. Sebaliknya, apabila kunci yang dimasukkan tidak benar, maka gambar video yang dihasilkan masih dalam keadaan kabur atau acak. Skema kerja aplikasi dapat dilihat pada Gambar 8.



Gambar 8 Skema kerja aplikasi

Perangkat lunak ini mampu mengimplementasikan fungsi-fungsi berikut:

a. Melakukan enkripsi *file* video tanpa merubah struktur video tersebut.
b. Melakukan dekripsi *file* video hasil enkripsi dan mampu mengembalikan video terenkripsi menjadi video yang sebenarnya.
c. Menyimpan video hasil enkripsi dan dekripsi ke sebuah *file* baru.
d. Kunci yang digunakan pada enkripsi dan dekripsi harus sama. Jika kunci tidak sama, proses dekripsi video tetap dapat dilaksanakan, namun video hasil dekripsi bukan merupakan video yang sebenarnya dan masih dalam keadaan acak dan kabur.

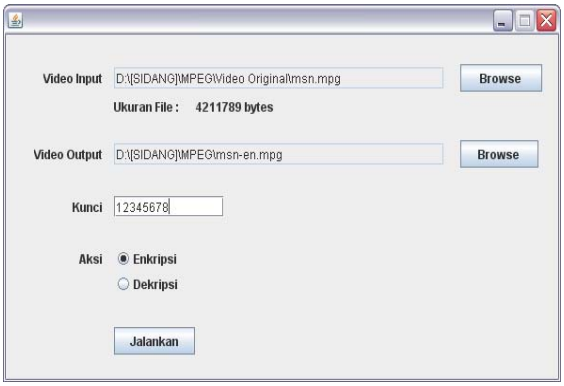
4. IMPLEMENTASI

Lingkungan yang digunakan dalam membangun perangkat lunak enkripsi video yang telah dirancang adalah lingkungan berbasis *windows*. Bahasa pemograman yang digunakan untuk membangun perangkat lunak adalah Java.

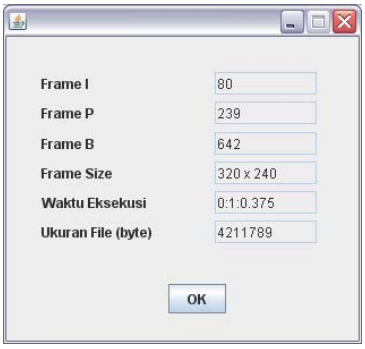
Perangkat lunak yang dikembangkan memiliki batasan sebagai berikut:

1. Algoritma DES pada perangkat lunak ini bekerja pada kunci dengan panjang 64 bit. Oleh karena itu, apabila terdapat sisa bit-bit pada *slice* yg berukuran kurang dari 64 bit, maka akan langsung ditulis ke *file* tujuan tanpa dilakukan enkripsi ataupun *padding*. Sehingga ukuran file video hasil enkripsi tetap sama dengan ukuran video yang sebenarnya.
2. Data yang akan dienkripsi hanyalah *slice* dari *frame* I dan P dari MPEG video. *Header* dari *MPEG audio* tidak ikut dienkripsi.

Gambar 9 dan Gambar 10 memperlihatkan implementasi antarmuka dari perangkat lunak.



Gambar 9 Implementasi antarmuka utama



Gambar 10 Implementasi antarmuka hasil

5. PENGUJIAN

Pengujian dilaksanakan pada lingkungan dengan spesifikasi perangkat keras dan perangkat lunak yang sama dengan lingkungan implementasi perangkat lunak. Format video yang digunakan dalam pengujian adalah video dengan format MPEG-1. Perangkat lunak tidak melakukan validasi *file* video apakah video tersebut merupakan video MPEG-1 atau tidak.

Data uji yang digunakan pada pengujian dapat dilihat pada Tabel 1.

No	Nama Video	Ukuran (*kilobytes*)	Durasi Video	Ukuran *frame*
1	"biotherm.mpg"	5.288	00:00:31	352x240
2	"msn.mpg"	4.114	00:00:31	320x240
3	"clip01.mpg"	3.485	00:01:45	176x112
4	"toyota.mpg"	2.121	00:00:45	256x192
5	"devilcar.mpg"	1,071	00:00:23	320x240

Tabel 1 Data Uji

Terdapat empat buah kasus uji yang digunakan untuk menguji kebenaran dan kinerja perangkat lunak:

1. Menguji kebenaran proses enkripsi dan dekripsi
2. Proses dekripsi gagal jika kunci tidak sama
3. Menguji kesalahan bit pada *file* video sebelum dekripsi
4. Menguji dampak perubahan *file* video setelah proses enkripsi

Berikut adalah hasil pengujian dari masing-masing kasus uji:

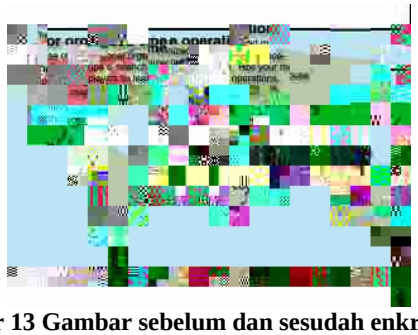
1. Kasus Uji 1
   Pengujian ini dilakukan dengan cara memilih sebuah *file* video MPEG-1 untuk dienkripsi dengan kunci masukan user, yaitu '12345678'. Selanjutnya *file* video hasil ekripsi video tersebut didekripsi dengan kunci yang sama dengan kunci proses enkripsi sebelumnya. Video hasil dekripsi ini menghasilkan video yang sama dengan video sebelum dienkripsi.

2. Kasus Uji 2
   Pengujian ini dilakukan dengan cara melakukan dekripsi dari video keluaran hasil enkripsi pada kasus uji 1 dengan kunci yang berbeda, yaitu “87654321”. Pengujian ini berhasil menghasilkan video hasil dekripsi masih dalam keadaan acak, dan bukan merupakan video aslinya
3. Kasus Uji 3
   Pengujian ini dilakukan dengan cara mengubah bit-bit *slice* dari *frame* I atau B pada video hasil enkripsi. *Frame* yang akan diubah bit-bitnya adalah *frame* yang berada ditengah video sehingga perubahannya akan terlihat dengan jelas pada saat video dijalankan. Kemudian, video yang telah diubah tersebut didekripsi dengan menggunakan kunci yang benar. Video ini selanjutnya dilihat perubahan gambarnya untuk melihat pengaruh dari modifikasi CBC yang diterapkan. Hasil pengujian ini berhasil menghasilkan gambar yang sama dengan video aslinya di awal video dijalankan, kemudian ditengah video mulai berubah menjadi acak hingga akhir video.
4. Kasus Uji 4
   Pengujian ini dilakukan dengan cara membandingkan hasil enkripsi video dengan video sebelum dienkripsi. Untuk membandingkan kedua video tersebut, seluruh *frame* dari kedua video tersebut diekstrak terlebih dahulu menjadi gambar dengan format JPEG. Kemudian, diambil *frame* dengan urutan yang sama dari kedua video tersebut untuk dibandingkan. Pengujian ini dilakukan secara objektif, yaitu dengan melihat perbedaan dari kedua gambar tersebut.

   Hasil perbandingan gambar sebelum dan sesudah enkripsi dapat dilihat pada Gambar 11 hingga Gambar 13.

Gambar 11 Gambar sebelum dan sesudah enkripsi dari video ‘biotherm.mpg’

Gambar 12 Gambar sebelum dan sesudah enkripsi dari video ‘devilcar.mpg’



Gambar 13 Gambar sebelum dan sesudah enkripsi dari video ‘msn.mpg’

Dari data yang diperoleh selama proses pengujian dapat dianalisis bahwa:

1. Rancangan proses enkripsi video pada video MPEG-1 yang telah dirancang dapat diimplementasikan dan berhasil dijalankan dengan baik.
2. Proses enkripsi maupun dekripsi dapat menghasilkan video dengan format yang

sama dengan video aslinya, yaitu MPEG-1, serta dapat dijalankan pada *video player*.

3. Perangkat lunak dapat menangani penggunaan kunci dengan baik. Proses dekripsi dengan kunci yang salah dapat ditangani dengan menghasilkan video yang berbeda dengan video yang asli.
4. Model enkripsi yang menggunakan VEA yang telah dimodifikasi dengan DES dan CBC modifikasi dapat berjalan baik. Video hasil dekripsi dari video yang telah diubah bit-bit *frame* I-nya dapat menunjukkan efek perubahan bit tersebut saat dijalankan.
5. Hasil perbandingan kedua gambar dari video sebelum enkripsi dan sesudah enkripsi menghasilkan gambar yang acak, bahkan bisa dikatakan tidak mirip.
6. Fungsi-fungsi yang terdapat pada perangkat lunak dapat berjalan dengan baik.

6. KESIMPULAN

Kesimpulan yang dapat diambil adalah:

1. Secara umum, penerapan modifikasi algoritma VEA terhadap enkripsi video ini dapat diimplementasikan dan berhasil dijalankan dengan baik. Pengujian menunjukkan bahwa enkripsi dan dekripsi serta penanganan kunci berhasil dilakukan.
2. Performansi dari modifikasi algoritma VEA tidak dipengaruhi oleh durasi video. Pengujian menunjukkan bahwa terdapat video dengan waktu eksekusi yang lebih sebentar dan lebih lama dari durasinya.
3. Modifikasi algoritma VEA dengan DES pada enkripsi video ini dapat meningkatkan keamanan tanpa mengubah struktur dari video. Video hasil enkripsi merupakan video dengan format yang sama, yaitu MPEG-1 yang dapat dijalankan di berbagai video player.
4. Enkripsi video dengan algoritma enkripsi video ini dapat menghasilkan gambar yang sangat acak, bahkan berbeda dari gambar aslinya.

DAFTAR REFERENSI

[1] Daniel, Socek. *Comparison and Analysis of Selected Video Encryption Algorithms Implemented for MPEG-2 Streams*. Department of Computer Science and Engineering Florida Atlantic University, U.S.A.

[2] Bharat Bhargava, Changgui Shi and Sheng-Yih Wang. *MPEG Video Encryption Algorithms*. Department of Computer Science Purdue University, USA, 2002

[3] Kilian, Joe and Phillip Rogaway. *How to protect DES against exhaustive key search*. Advances in Cryptology - Crypto '96, Springer-Verlag.