

BAB V PENGUJIAN

Untuk mengetahui apakah sistem bekerja dengan baik dan sesuai dengan perancangan, maka diperlukan serangkaian pengujian. Pengujian kriptografi video *streaming* pada *server* dan *client* meliputi pengujian-pengujian sebagai berikut.

5.1 Pengujian Server Berkoneksi dengan Client

Pengujian ini bertujuan untuk mengetahui apakah *server* dapat berkoneksi dengan *client* sehingga bisa digunakan sesuai fungsionalitasnya. Hasil pengujian dilakukan pada satu *server* dan beberapa *client* dalam jaringan komputer lokal (LAN) menggunakan *protocol* TCP/IP dimana data yg dikirim secara serial atau sequensial. Pada Tabel 5.1 hasil pengujian koneksi *server* dan *client* sesuai dengan perancangan dan batasan masalah yang telah dijelaskan pada bab sebelumnya.

Table 5.1 Hasil pengujian koneksi *server* dengan *client*

No.	Jumlah <i>Client</i>	Hasil Pengujian
1	1	Bisa terkoneksi
2	2	Bisa terkoneksi
3	3	Bisa terkoneksi

(Sumber : Pengujian)

Hasil pengujian pada Tabel 5.1 diperoleh dengan cara pada setiap percobaan jumlah *client* yang aktif ditambahkan, yaitu dengan menjalankan program “VeaTube Client“, kemudian *server* mencoba melakukan hubungan koneksi pada *client* tersebut. Jadi dapat disimpulkan bahwa *server* bisa terkoneksi dengan banyak *client* apabila *service* yang ada di dalam program “VeaTube Client“ pada *client* sudah dijalankan.

5.2 Rancangan Kasus Uji pada Server dan Client

Rancangan kasus uji pada *server* dan *client* dapat dilihat pada Tabel 5.2

Tabel 5.2 Rancangan kasus uji

No.	Fungsi	Kasus Uji
1	<i>Transmitter</i> menerima data masukan video.	Masukan berupa format video mpeg-1 dan mpeg-2 (.mpg, .mpeg).
2	<i>Transmitter (server)</i> melakukan enkripsi dan <i>receiver (client)</i> melakukan dekripsi jika kunci sama.	Masukan kunci pada <i>encryptor</i> dan <i>decryptor</i> sama.
3	Proses dekripsi tidak berhasil dilakukan jika kunci tidak sama.	Masukan kunci pada <i>decryptor</i> berbeda dengan masukan kunci pada <i>encryptor</i> .
4	<i>Display</i> video pada <i>receiver</i> tidak muncul jika video tidak diterima.	Masukan alamat ip dan <i>port</i> pada <i>transmitter</i> tidak sama dengan <i>receiver</i> .
5	<i>Transmitter</i> dan <i>receiver</i> dijalankan disatu komputer saja.	Menjalankan <i>transmitter</i> dan <i>receiver</i> bersamaan disatu komputer.
6	<i>Transmitter</i> dan <i>receiver</i> dijalankan dibeberapa komputer.	Menjalankan <i>transmitter</i> dan <i>receiver</i> bersamaan dibeberapa komputer. Satu komputer <i>transmitter</i> , komputer lain adalah <i>receiver</i> .
7	Mengukur performansi model enkripsi pada video <i>streaming</i> yang telah dibangun.	Masukan kunci untuk proses enkripsi berbeda-beda dari sisi panjang kunci
8	Mengukur perbedaan waktu performansi antara video <i>streaming</i> yang menggunakan algoritma enkripsi VEA dan MD5 dengan video <i>streaming</i> yang tidak menggunakan enkripsi dalam <i>streaming</i> videonya.	Hitung perbedaan waktu yang dibutuhkan <i>receiver</i> untuk memunculkan <i>display</i> videonya ketika menjalankan video <i>streaming</i> baik yang menggunakan enkripsi maupun yang tidak menggunakan enkripsi.

(Sumber : Pengujian)

5.2.1 Rekapitulasi Hasil Uji pada *Server* dan *Client*

Rekapitulasi hasil uji dapat dilihat pada Tabel 5.3 merupakan hasil-hasil yang didapat ketika melakukan proses pengujian berdasarkan rancangan kasus uji.

Tabel 5.3 Rekapitulasi hasil pengujian

No.	Pengujian	Hasil
1	Masukan video berupa format MPEG-1 dan MPEG-2 dengan ekstensi .mpeg atau .mpg.	Format video dapat diterima <i>receiver (client)</i> dan dapat dikirim <i>transmitter (server)</i> .
2	Masukan video berupa format selain MPEG-1 dan MPEG-2.	Format video tidak dapat diterima <i>receiver</i> dan dapat dikirim <i>transmitter</i> .
3	Masukan kunci pada <i>transmitter</i> dan <i>receiver</i> sama.	<i>Display</i> video dapat berjalan pada <i>receiver</i> dan tampilan video telah terdekripsi.
4	Masukan kunci pada <i>transmitter</i> dan <i>receiver</i> berbeda.	<i>Display</i> video dapat berjalan pada <i>receiver</i> namun tampilan video acak atau terenkripsi.
5	Masukan alamat ip dan <i>port</i> pada <i>transmitter</i> dan <i>receiver</i> berbeda.	<i>Receiver</i> tidak bisa berkoneksi dengan <i>transmitter</i> dan <i>Display</i> video tidak dapat berjalan pada <i>receiver</i> .
6	<i>Transmitter</i> dan <i>receiver</i> dijalankan disatu komputer.	<i>Transmitter</i> dan <i>receiver</i> dapat berjalan dengan baik.
7	<i>Transmitter</i> dan <i>receiver</i> dijalankan di beberapa komputer.	<i>Transmitter</i> dan <i>receiver</i> dapat berjalan dengan baik.
8	Antarmuka <i>transmitter</i> dan <i>receiver</i> .	Seluruh tombol pada antarmuka <i>transmitter</i> dan <i>receiver</i> dapat berjalan dengan baik.
9	Hitung waktu yang dibutuhkan proses enkripsi pada <i>transmitter</i> dan proses dekripsi pada <i>client</i> .	Performansi yang dihasilkan dapat dilihat pada Tabel 5.4 dan 5.5
10	Hitung perbedaan waktu yang dibutuhkan oleh <i>receiver</i> untuk memunculkan <i>display</i> videonya baik dari video <i>streaming</i> yang menggunakan enkripsi maupun video <i>streaming</i> yang tidak menggunakan enkripsi.	Performansi yang dihasilkan dapat dilihat pada Tabel 5.6 dan 5.7
11	Masukan panjang kunci berbeda-beda.	Performansi yang dihasilkan konstan. Detail dapat dilihat pada Tabel 5.8 dan 5.9

(Sumber : Pengujian)

5.3 Pengujian Kriptografi VEA pada *Server* dan *Client*

Pengujian kriptografi algoritma *Video Encryption Algorithm* (VEA) dilakukan untuk menghitung dan mengetahui waktu yang dibutuhkan oleh *server* dalam proses enkripsi video dan juga waktu yang dibutuhkan *receiver* dalam proses dekripsi video. Tujuan pengujian ini agar dapat mengetahui kinerja dari algoritma VEA tersebut apakah sudah berjalan dengan baik dan apakah sudah dapat digunakan sebagai algoritma yang tepat dalam keamanan data khususnya untuk *streaming* data video.

Pada pengujian ini terdapat 3 waktu yang akan dicatat yakni waktu *display*, total waktu enkripsi, total waktu dekripsi. Waktu *display* adalah waktu yang didapat saat *user* menekan tombol start sampai muncul form *display*. Total waktu enkripsi adalah waktu total saat *server* melakukan membaca *byte* dari file video kemudian melakukan proses enkripsi dan mengirimkan *byte-byte* tersebut pada *client*. Total waktu dekripsi adalah waktu total saat *client* mendapatkan *byte-byte* dari *server* kemudian melakukan proses dekripsi dan menuliskan *byte-byte* tersebut menjadi *file* video *temporary*.

Pada pengujian tahap ini akan dilakukan percobaan dengan menggunakan dua *file* video mpeg yaitu video mpeg-1 dan mpeg-2, dimana kedua video tersebut mempunyai spesifikasi yang berbeda. Percobaan ditujukan untuk mengetahui waktu rata-rata hasil dari percobaan pertama menggunakan mpeg-1 apakah jauh berbeda dengan hasil dari percobaan kedua menggunakan mpeg-2.

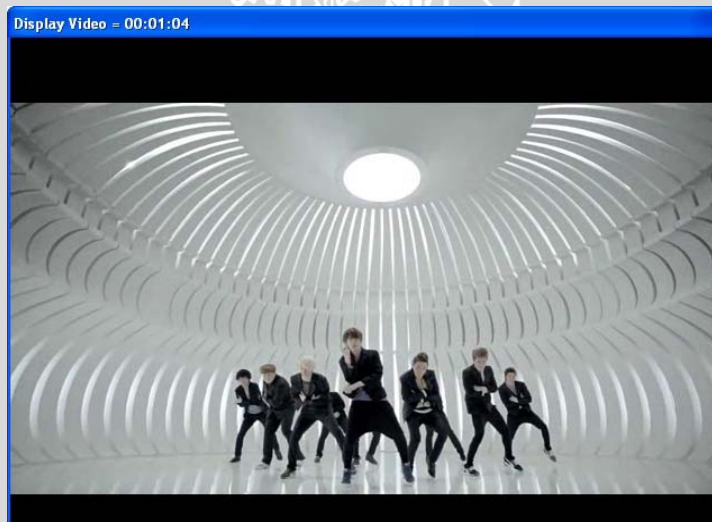
Pengujian ini menggunakan beberapa perangkat keras berupa *Personal Computer* (PC) dengan spesifikasi yaitu Intel Core 2 Duo E7500 2.93 GHz (2 CPUs), *mainboard* FOXCONN ETON, DDR3 1024MB, VGA intel GMA X4500 *Graphics*, *harddisk* 320GB. Sehingga hasil atau waktu enkripsi dan dekripsi dalam pengujian ini hanya didapat dari perangkat keras yang telah ditentukan tersebut. Apabila aplikasi dijalankan pada perangkat keras yang berbeda maka dapat menghasilkan atau menunjukkan waktu enkripsi dan dekripsi yang berbeda pula.

5.3.1 Pengujian Kriptografi VEA pada Video MPEG-1

Untuk spesifikasi dari video MPEG-1 antara lain :

- Nama *file* = mr_simple.mpg
- Jumlah *frame* total = 7065
- Jumlah *frame* I = 605
- Bit *rate* = 2 512 Kbps
- *Frame rate* = 23.976 fps
- Resolusi = 640 x 480 pixels
- Durasi video = 4 menit 52 detik
- Ukuran *file* MPEG = 98508800 bytes (93.9 MB)

➤ Tampilan video mpeg-1 sebagai berikut:



Gambar 5.1 Screen shoot hasil dekripsi video mr_simple.mpg

(Sumber : Pengujian)

Gambar 5.1 merupakan *screen shoot* hasil dekripsi dari *file* video mr_simple.mpg, dimana *user (client)* memasukkan kunci yang sama sehingga menampilkan video yang dapat dinikmati. Namun apabila dimasukkan kunci yang salah atau beda maka akan dilakukan proses enkripsi sehingga akan menghasilkan tampilan video yang acak atau rusak yang tidak dapat dinikmati oleh pengguna. Untuk tampilan

dari file video mr_simple.mpg yang telah terenkripsi ditunjukkan seperti pada Gambar 5.2



Gambar 5.2 Screen shoot hasil enkripsi video mr_simple.mpg
(Sumber : Pengujian)

Dalam pengujian ini didapat perbandingan waktu proses enkripsi dan dekripsi pada video mpeg-1 menggunakan algoritma VEA.

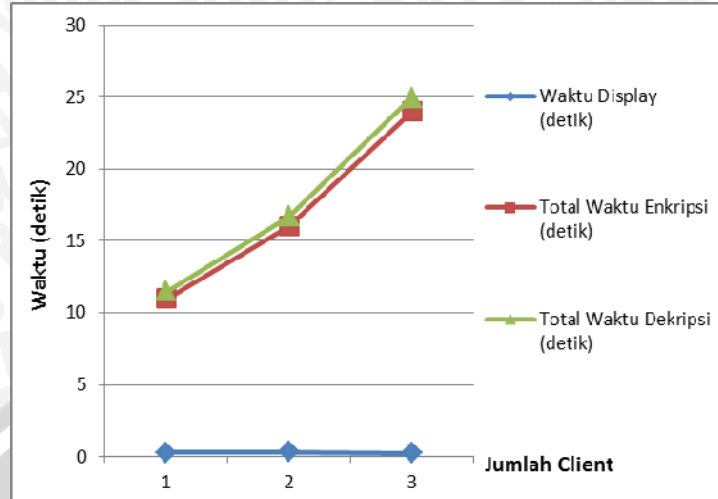
Untuk algoritma VEA pada video mpeg-1 ditunjukkan pada Tabel 5.4

Tabel 5.4 Waktu enkripsi dan dekripsi pada video mpeg-1

Jumlah Client	Total Waktu Enkripsi (detik)	Total Waktu Dekripsi (detik)
1	11	11.485
2	16	16.686
3	24	24.956

(Sumber : Pengujian)

Apabila Tabel 5.4 dibentuk menjadi kurva atau grafik akan terlihat seperti pada Gambar 5.3



Gambar 5.3 Kurva perbandingan waktu kriptografi video mr_simple.mpg

(Sumber : Pengujian)

5.3.2 Pengujian Kriptografi VEA pada Video MPEG-2

Untuk spesifikasi dari video MPEG-2 antara lain :

- Nama *file* = hipnotis.mpg
- Jumlah *frame* total = 8684
- Jumlah *frame* I = 714
- *Bit rate* = 1 365 Kbps
- *Frame rate* = 29.970 fps
- Resolusi = 640 x 480 pixels
- Durasi video = 4 menit 46 detik
- Ukuran *file* MPEG = 54514294 bytes (51.9 MB)

➤ Tampilan video mpeg-2 sebagai berikut:



Gambar 5.4 *Screen shoot* hasil dekripsi video hipnotis.mpg
(Sumber : Pengujian)

Gambar 5.1 merupakan *screen shoot* hasil dekripsi dari video hipnotis.mpg, dimana *user (client)* memasukkan kunci yang sama sehingga menampilkan video yang dapat dinikmati. Namun apabila dimasukkan kunci yang salah atau beda maka akan dilakukan proses enkripsi sehingga akan menghasilkan tampilan video yang acak atau rusak yang tidak dapat dinikmati oleh pengguna. Untuk tampilan dari *file* video hipnotis.mpg yang telah terenkripsi ditunjukkan seperti pada Gambar 5.2



Gambar 5.5 *Screen shoot* hasil enkripsi video hipnotis.mpg
(Sumber : Pengujian)

Dalam pengujian ini didapat perbandingan waktu proses enkripsi dan dekripsi pada video mpeg-2 menggunakan algoritma VEA.

Untuk algoritma VEA pada video mpeg-2 ditunjukkan pada Tabel 5.5

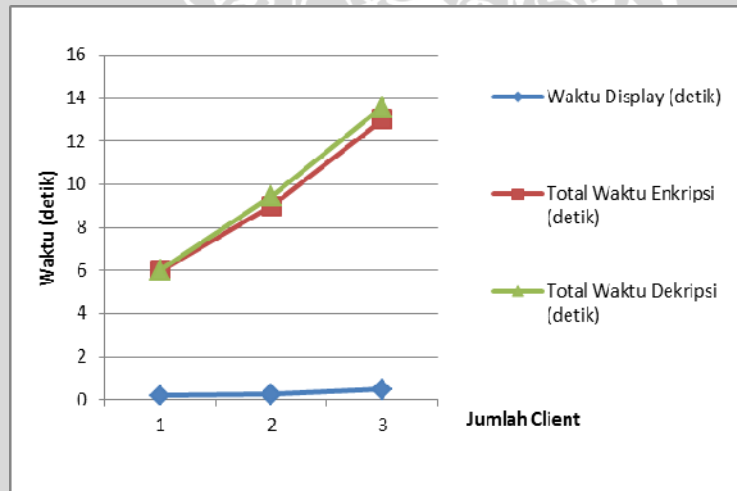
Tabel 5.5 Waktu enkripsi dan dekripsi pada video mpeg-2

Jumlah Client	Total Waktu Enkripsi (detik)	Total Waktu Dekripsi (detik)
1	6	6.021
2	9	9.445
3	13	13.595

(Sumber : Pengujian)

Tabel 5.5 jika dibentuk menjadi kurva atau grafik akan terlihat seperti

Gambar 5.6



Gambar 5.6 Kurva perbandingan waktu kriptografi video hipnotis.mpg

(Sumber : Pengujian)

5.4 Pengujian Kriptografi VEA dengan Panjang Kunci Berbeda

Pengujian *Video Encryption Algorithm* (VEA) yang digunakan dalam membangun model enkripsi video *streaming* diuji coba dengan cara memasukkan panjang kunci yang berbeda-beda dan kemudian menghitung waktu yang

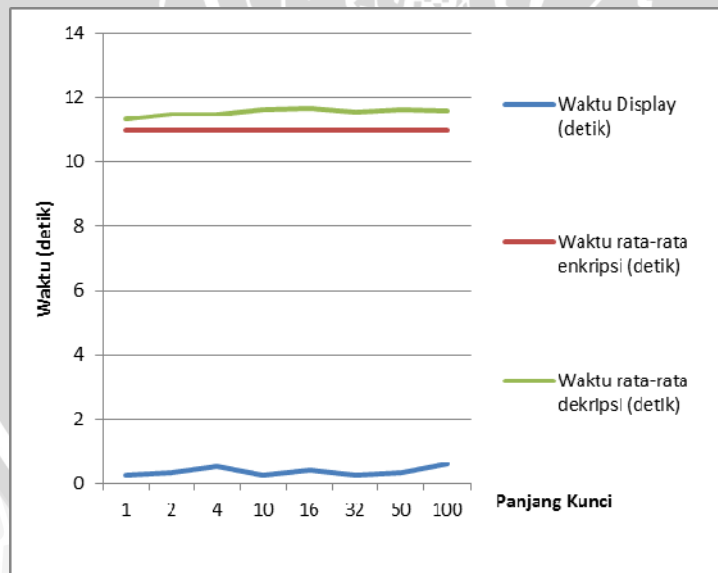
dibutuhkan transmitter untuk mengenkripsi dan *receiver* untuk mendekripsi. Tabel hasil uji algoritma enkripsi video mpeg-1 dengan panjang kunci berbeda-beda dapat dilihat pada Tabel 5.6 sedangkan Grafik fungsi waktu dan panjang kunci dari hasil pengujian dapat dilihat pada Gambar 5.7

Tabel 5.6 Hasil uji algoritma enkripsi video mpeg-1

Panjang Kunci (karakter)	Waktu Rata-rata Enkripsi (detik)	Waktu Rata-rata Dekripsi (detik)
1	11	11.324
2	11	11.467
4	11	11.495
10	11	11.624
16	11	11.668
32	11	11.556
50	11	11.634
100	11	11.607

(Sumber : Pengujian)

Tabel 5.6 jika dibentuk menjadi kurva atau grafik akan terlihat pada Gambar 5.7



Gambar 5.7 Grafik hasil pengujian algoritma enkripsi pada video mpeg-1

(Sumber : Pengujian)

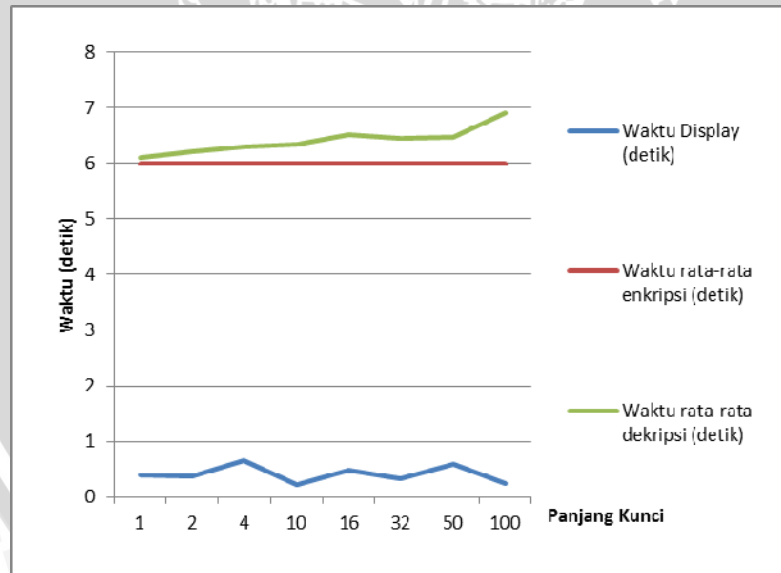
Sedangkan tabel dari hasil uji performansi algoritma enkripsi video mpeg-2 dapat dilihat pada Tabel 5.7 dan Grafik fungsi waktu dan panjang kunci dari hasil pengujian dapat dilihat pada Gambar 5.8

Tabel 5.7 Hasil uji algoritma enkripsi video mpeg-2

Panjang Kunci (karakter)	Waktu Rata-rata Enkripsi (detik)	Waktu Rata-rata Dekripsi (detik)
1	6	6.105
2	6	6.221
4	6	6.293
10	6	6.357
16	6	6.522
32	6	6.462
50	6	6.473
100	6	6.921

(Sumber : Pengujian)

Tabel 5.7 jika dibentuk menjadi kurva atau grafik akan terlihat pada Gambar 5.8



Gambar 5.8 Grafik hasil pengujian algoritma enkripsi pada video mpeg-2

(Sumber : Pengujian)

Berdasarkan pengujian dihasilkan algoritma enkripsi video yang digunakan untuk membangun model enkripsi video *streaming* ini adalah konstan dan hanya berkisar disekitar 11 detik pada *file* video mpeg-1 sedangkan pada *file* video mpeg-2 berkisar sekitar 6 detik. Pada proses dekripsi terjadi sedikit perbedaan waktu karena terdapat proses pengiriman data atau proses *streaming* sehingga terdapat penambahan waktu, akan tetapi pada proses dekripsinya diperkirakan waktu yang dibutuhkan adalah tetap atau konstan.

Oleh karena itu, dapat disimpulkan bahwa algoritma enkripsi yang digunakan (algoritma VEA dan MD5) pada model enkripsi video *streaming* ini konstan, tidak berpengaruh walau dengan panjang kunci yang berbeda-beda.

Algoritma enkripsi yang menimbulkan waktu yang konstan tersebut adalah MD5, karena berapapun panjang masukan kunci dari pengguna (*user*), kunci tersebut akan di-*hash* menggunakan kunci fungsi *hash* MD5 yang mengakibatkan panjang akhir kunci selalu 128 bit.

5.5 Pengujian Model Video Streaming

Pengujian model video *streaming* yang telah dibangun, diuji coba dengan cara menghitung total waktu yang dibutuhkan transmisi *file* video dari *server* pada *client*, baik dari model video *streaming* yang menggunakan enkripsi maupun dari model video *streaming* yang tidak menggunakan enkripsi dalam proses *streaming* videonya. Dari pengujian didapat hasil perbandingan waktu yang ditunjukkan pada Tabel 5.8, dimana hasil tersebut didapat dari lama waktu hasil uji model video *streaming* pada mpeg-1.

Tabel 5.8 Hasil uji model video *streaming* mpeg-1

Model Video Streaming	Waktu Display (detik)	Total Waktu Rata-rata receiver (detik)	Total Waktu Rata-rata transmitter (detik)
Menggunakan VEA	0.463	11.916	11
Tanpa VEA	0.207	9.872	9

(Sumber : Pengujian)

Sedangkan pada Tabel 5.9 merupakan lama waktu yang didapat dari hasil uji model video *streaming* pada mpeg-2.

Tabel 5.9 Hasil uji model video *streaming* mpeg-2

Model Video Streaming	Waktu Display (detik)	Total Waktu Rata-rata receiver (detik)	Total Waktu Rata-rata transmitter (detik)
Menggunakan VEA	0.753	7.891	7
Tanpa VEA	0.521	5.633	5

(Sumber : Pengujian)

Berdasarkan pengujian yang dilakukan, dapat dilihat bahwa waktu yang dibutuhkan model video *streaming* menggunakan enkripsi tidak berbeda jauh dengan model video *streaming* tanpa menggunakan enkripsi. Hal ini menunjukkan bahwa penambahan proses enkripsi pada video *streaming* tidak akan memberatkan kinerja dari proses *streaming*-nya itu sendiri, karena data yang dienkrpsi menggunakan algoritma VEA tersebut bukan lagi *file* video yang ukurannya sangat besar, namun *file* video yang dienkrpsi hanya bagian *frame* I, kemudian disimpan dalam paket-paket data yang ukuran satu paketnya relatif kecil dan proses pengiriman *file* video tersebut dilakukan secara *bit per-bit* (*stream chipper*).

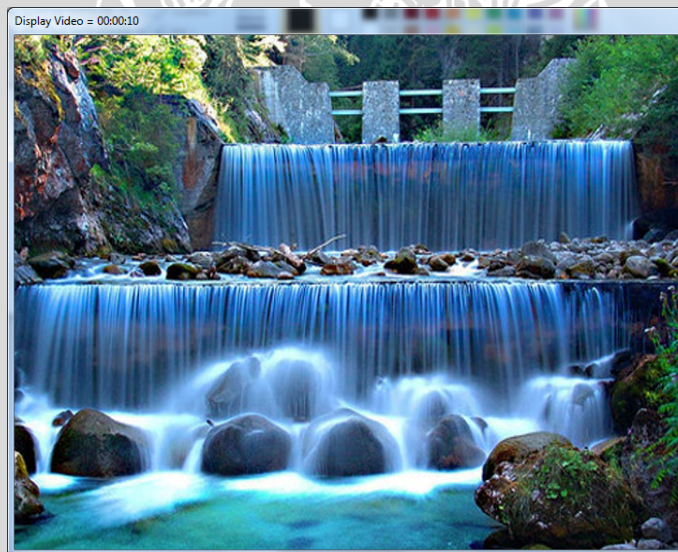
Selain itu, pengujian model video *streaming* juga dilakukan terhadap berbagai jenis video mpeg-1 dan mpeg-2 baik yang memiliki *frame rate* rendah maupun tinggi. Hasil pengujian pada video mpeg-1 menunjukkan waktu rata-rata yang dibutuhkan berkisar 9 detik sampai 11 detik. Dan hasil pengujian video mpeg-2 menunjukkan waktu rata-rata yang dibutuhkan berkisar 5 detik sampai 7 detik. Hal ini menunjukkan bahwa *frame rate* video tidak berpengaruh pada performansi model enkripsi video *streaming*.

5.6 Pengujian *Still Picture Video* (Video Gambar Diam)

Dalam Pengujian ini dilakukan pengujian kriptografi namun video yang digunakan adalah video yang dibuat dari satu gambar dan disalin menjadi banyak kemudian disusun menjadi video sehingga menjadi video gambar diam. Untuk spesifikasi dari video mpeg pada gambar diam antara lain :

- Nama *file* = waterfall.mpg
- Jumlah *frame* total = 400
- Jumlah *frame* I = 27
- *Bit rate* = 2 527 Kbps
- *Frame rate* = 25 fps
- Resolusi = 720 x 576 pixels
- Durasi video = 16 detik
- Ukuran *file* MPEG = 5156864 bytes (4.91 MB)

➤ Tampilan video sebagai berikut:

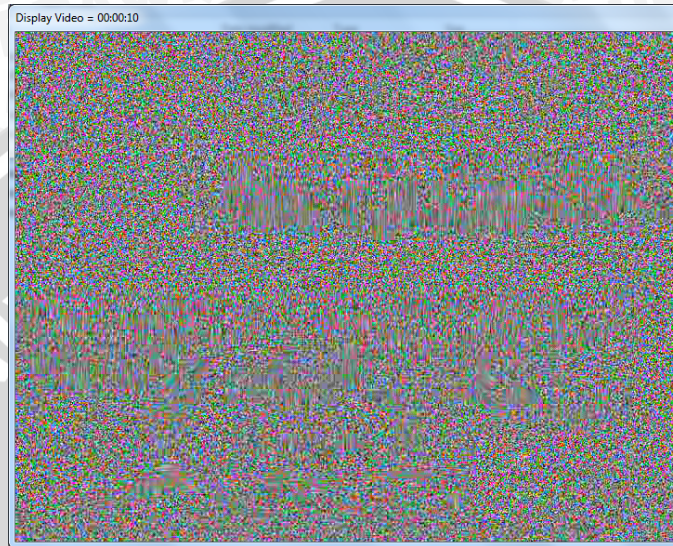


Gambar 5.9 *Screen shoot* hasil dekripsi video waterfall.mpg

(Sumber : Pengujian)

Gambar 5.9 merupakan *screen shoot* hasil dekripsi dari *file* video waterfall.mpg, dimana *user (client)* memasukkan kunci yang sama sehingga menampilkan video yang dapat dinikmati. Namun apabila dimasukkan kunci yang

salah atau beda maka akan dilakukan proses enkripsi sehingga akan menghasilkan tampilan video yang acak atau rusak yang tidak dapat dinikmati oleh pengguna. Untuk tampilan dari *file* video waterfall.mpg yang telah terenkripsi ditunjukkan seperti pada Gambar 5.10



Gambar 5.10 Screen shoot hasil enkripsi video waterfall.mpg
(Sumber : Pengujian)

Dari hasil pengujian kriptografi pada video gambar diam dihasilkan video enkripsi yang jauh lebih rusak atau lebih acak dibandingkan dengan pengujian kriptografi pada video yang diperoleh dari gambar bergerak. Perbedaan ini terjadi karena video gambar diam mempunyai perubahan antar *frame*-nya tidak ada (sama) dalam setiap detiknya.

Dengan kata lain antara *frame* I dengan *frame* P (*Predicted*) tidak ada perbedaan, sehingga pada *frame* P akan merujuk *frame* I yang telah terenkripsi dan didapatlah hasil enkripsi yang lebih acak daripada video gambar bergerak karena hanya bagian tertentu saja yang dienkripsi.

5.7 Pengujian Informasi Kriptografi VEA dengan Algoritma Big (O)

Kinerja sebuah algoritma dapat diketahui dengan beberapa cara, namun pada pengujian ini digunakan algoritma Big O untuk mengetahui kinerja atau performansinya. Berikut perhitungan performansi pada fungsi enkripsi :

```
private void enkripsi(NetworkStream clientStream, byte[]
bufferTmp, byte[] hash, int panjang, ref byte[] bufferPattern, ref bool
flagEncrypt,ref int indexHash, ref int totalFrame,ref int totalFrameI)
{
```

```
byte[] bufferFlag = new byte[8];
bool next_encrypt = false;
int shift = 0;
int coding_type;
```

```
for (int k = 0; k < bufferPattern.Length; k++ )
{
    bufferFlag[k] = bufferPattern[k];
} (1)
```

```
for (int i = 0; i < panjang; i++) (2)
{
    bufferFlag[0] = bufferFlag[1];
    bufferFlag[1] = bufferFlag[2];
    bufferFlag[2] = bufferFlag[3];
    bufferFlag[3] = bufferFlag[4];
    bufferFlag[4] = bufferFlag[5];
    bufferFlag[5] = bufferFlag[6];
    bufferFlag[6] = bufferFlag[7];
    bufferFlag[7] = bufferTmp[i];
```

```
if(next_encrypt == true || flagEncrypt == true) (3)
{
    bufferTmp[i]=(byte)((int)bufferFlag[7]^hash[indexHash%hash.Length]
);
    indexHash++;
}
```

```
if(bufferFlag[0]==0x00&&bufferFlag[1]==0x00&&bufferFlag[2] == (4)
0x01)
```

```
{
    if (bufferFlag[3] == 0x00) (5)
    {
        totalFrame++;
        coding_type = ((int)bufferFlag[5] & 0x38) >> 3;
```

```
if (coding_type == 1 ) (6)
{
    totalFrameI++;
    next_encrypt = true;
    flagEncrypt = true;
}
```

```
else
{
    next_encrypt = false;
    flagEncrypt = false;
}
```



```

        if (i < 7) (7)
        {
            shift = i;
        }
        else
        {
            shift = 7;
        }

        for (int m = shift; m >= 0; m--) (8)
        {
            bufferTmp[i - m] = bufferFlag[7 - m];
        }
    }
}

clientStream.Write(bufferTmp, 0, panjang); (9)

for (int l = 0; l < bufferPattern.Length; l++) (10)
{
    bufferPattern[l] = bufferFlag[l];
}
}

```

❖ Analisis Enkripsi Menggunakan Algoritma Big O :

Langkah 10 :

Misal : $\text{bufferPattern.Length} = n$; banyak Statement = p

Banyak langkah = $(\text{akhir} - \text{awal} + 2) + (\text{akhir} - \text{awal} + 1)(p + 1)$

$$= (\text{bufferPattern.Length} - 0 + 2) + (\text{bufferPattern.Length} - 0 + 1) * (1 + 1)$$

$$= (k + 2) + (k + 1) * (2)$$

$$= (k + 2) + (k + 1) * 2$$

$$= k + 2 + 2k + 2$$

$$= 3k + 4$$

Langkah 9 :

Banyak langkah = 1 (Karena hanya melakukan 1 kali proses pemanggilan fungsi)

Langkah 8 :

Misal : $p = \text{banyak Statement} = 1$

$$\begin{aligned}
 \text{Banyak langkah} &= (\text{akhir} - \text{awal} + 2) + (\text{akhir} - \text{awal} + 1) (p + 1) \\
 &= (m - 0 + 2) + (m - 0 + 1) * (1 + 1) \\
 &= (m + 2) + (m + 1) * (2) \\
 &= m + 2 + 2m + 2 \\
 &= 3m + 4
 \end{aligned}$$

Langkah 7 :

Misal : c = banyak Kondisi yang dicek pada Percabangan

$P1$ = banyak Statement pada Kondisi 1

$P2$ = banyak Statement pada Kondisi 2

$$\begin{aligned}
 \text{Banyak langkah} &= c + \max (P1, P2) \\
 &= 1 + \max (1, 1) \\
 &= 1 + 1 \\
 &= 2
 \end{aligned}$$

Langkah 6 :

Misal : c = banyak Kondisi yang dicek pada Percabangan

$P1$ = banyak Statement pada Kondisi 1

$P2$ = banyak Statement pada Kondisi 2

$$\begin{aligned}
 \text{Banyak langkah } P2 &= 2 + \text{Langkah 7} + \text{langkah 8} \\
 &= 2 + 2 + 3m + 4 \\
 &= 3m + 8 \\
 \text{Banyak langkah} &= c + \max (P1, P2) \\
 &= 1 + \max (3, 3m + 8) \\
 &= 1 + 3m + 8 \\
 &= 3m + 9
 \end{aligned}$$

Langkah 5:

Misal : c = banyak Kondisi yang dicek pada Percabangan

$P1$ = banyak Statement pada Kondisi 1 = Langkah 6

$P2$ = banyak Statement pada Kondisi 2

$$\begin{aligned}
 \text{Banyak langkah} &= c + \max (P1, P2) \\
 &= 1 + \max (2 + (3m + 9), 0) \\
 &= 1 + 3m + 11 \\
 &= 3m + 12
 \end{aligned}$$

Langkah 4 :

Misal : c = banyak Kondisi yang dicek pada Percabangan
 $P1$ = banyak Statement pada Kondisi 1 = Langkah 5
 $P2$ = banyak Statement pada Kondisi 2

$$\begin{aligned}
 \text{Banyak langkah} &= c + \max (P1, P2) \\
 &= 5 + \max (3m + 12, 0) \\
 &= 5 + 3m + 12 \\
 &= 3m + 17
 \end{aligned}$$

Langkah 3 :

Misal : c = banyak Kondisi yang dicek pada Percabangan
 $P1$ = banyak Statement pada Kondisi 1
 $P2$ = banyak Statement pada Kondisi 2

$$\begin{aligned}
 \text{Banyak langkah} &= c + \max (P1, P2) \\
 &= 3 + \max (2, 0) \\
 &= 3 + 2 \\
 &= 5
 \end{aligned}$$

Langkah 2 :

$$\begin{aligned}
 \text{Banyak Langkah P} &= 8 + \text{Langkah 3} + \text{Langkah 4} \\
 &= 8 + 5 + 3m + 17 \\
 &= 3m + 30
 \end{aligned}$$

Misal : panjang = a

$$\begin{aligned}
 \text{Banyak Langkah} &= (\text{akhir} - \text{awal} + 2) + (\text{akhir} - \text{awal} + 1) (p + 1) \\
 &= (\text{panjang} - 0 + 2) + (\text{panjang} - 0 + 1) * (3m + 30 + 1) \\
 &= (a + 2) + (a + 1) * (3m + 31) \\
 &= a + 2 + 3am + 31a + 3m + 31 \\
 &= 3am + 32a + 3m + 33
 \end{aligned}$$

Langkah 1 :

Misal : $\text{bufferPattern.Length} = k$; banyak Statement = p

$$\begin{aligned} \text{Banyak langkah} &= (\text{akhir} - \text{awal} + 2) + (\text{akhir} - \text{awal} + 1) (p + 1) \\ &= (\text{bufferPattern.Length} - 0 + 2) + (\text{bufferPattern.Length} - 0 + 1) * (1 + 1) \\ &= (k + 2) + (k + 1) * (2) \\ &= (k + 2) + (k + 1) * 2 \\ &= k + 2 + 2k + 2 \\ &= 3k + 4 \end{aligned}$$

Jadi, total langkah pada fungsi Enkripsi :

$$\begin{aligned} &= 4 + \text{Langkah 1} + \text{Langkah 2} + \text{Langkah 9} + \text{Langkah 10} \\ &= 4 + (3k + 4) + (3am + 32a + 3m + 33) + (1) + 3k + 4 \\ &= 3ma + 32a + 3m + 6k + 46 \end{aligned}$$

$$\begin{aligned} \text{Notasi Big (O)} &= 3N^2 + 32N + 3N + 6N + 46 \\ &= 3N^2 + 41N + 46 \\ &= O(n^2) \end{aligned}$$

Dari analisa hasil perhitungan menggunakan Algoritma Big O didapat persamaan $3N^2 + 41N + 46$. Persamaan tersebut termasuk Notasi Big O Kuadratik karena pangkat tertinggi adalah 2.

5.8 Kesimpulan Hasil Pengujian

Dari data yang diperoleh selama pengujian, dapat dianalisis bahwa:

1. Rancangan model enkripsi video pada *streaming* berdasarkan model yang telah dirancang penulis dapat diterapkan dan berhasil diimplementasikan dengan baik.
2. Model enkripsi yang menggunakan algoritma VEA yang telah dimodifikasi dengan menggunakan algoritma enkripsi kunci rahasia MD5 dan disesuaikan untuk kebutuhan *streaming* paket data video dapat berjalan pada video *streaming*.

3. Model enkripsi pada video *streaming* yang menggunakan algoritma enkripsi VEA dan MD5 berdasarkan perbedaan panjang kunci yang digunakan konstan. Waktu yang dihasilkan tetap walaupun dengan panjang kunci yang beragam.
4. Waktu yang didapat pada saat pengujian video *streaming* yang menggunakan algoritma enkripsi VEA dan MD5 dengan video *streaming* yang tidak menggunakan enkripsi dalam *streaming* videonya hampir tidak ada bedanya. Oleh karena itu, pengguna enkripsi pada video *streaming* tidak akan membebani kinerja video *streaming*-nya sendiri.
5. Pengujian kriptografi pada video gambar diam menghasilkan tampilan video yang lebih buruk atau lebih acak dibandingkan pada video gambar bergerak dikarenakan pada video gambar diam perubahan antar *frame*-nya tidak ada (sama) dalam setiap detiknya.
6. Perangkat lunak yang dibangun dapat berjalan dengan baik, baik disatu komputer maupun dibeberapa komputer.
7. Fungsi-fungsi yang terdapat dalam perangkat lunak dapat berjalan dengan baik.
8. Format video yang dapat digunakan dan dapat berjalan hanya pada *file* video mpeg-1 dan mpeg-2. Beberapa format video lain tidak dapat diuji coba karena sumber video dengan format tersebut tidak kompatibel dengan program yang dibuat oleh penulis.