PENGAMANAN DATA VIDEO *SURVEILLANCE* SECARA *REAL-TIME* MENGGUNAKAN *VIDEO ENCRYPTION ALGORITHM*

SECURING VIDEO SURVEILLANCE DATA BY REAL-TIME MODE USING VIDEO ENCRYPTION ALGORITHM

Dorynita Fenny Illyan, Surya Michrandi Nasution,S.T.,M.T., Anton Siswo S.T., M.T

Sistem Komputer, Fakultas Teknik Elektro – Univesitas Telkom
Jln. Telekomunikasi No.1 Terusan Buah Batu Bandung 40257 Indonesia dorynitafenny@gmail.com[1], michrandi@telkomuniversity.ac.id[2], worldliner@telkomuniversity.ac.id[3]

ABSTRAK

Keamanan merupakan aspek yang sangat penting bagi jalannya pertukaran data. Pada umumnya data yang dikirim hanya ditunjukan bagi pihak-pihak tertentu saja. Suatu data harus sampai pada pihak yang berhak dengan kerahasiaan yang tetap terjaga, tanpa harus diketahui oleh pihak-pihak yang tidak berkepentingan. Oleh karena itu untuk menjaga keamanan dan kerahasiaan data tersebut, perlu adanya metode enkripsi data, yang merupakan ilmu untuk menyembunyikan informasi dari pihak ketiga. Salah satu algoritma yang cukup handal dalam mengamankan suatu data yang *real-time* adalah VEA *(Video Encryption Algorithm)*. VEA dapat diimplementasikan di lingkungan video *streaming* karena algoritmanya yang dapat berbasis stream cipher maupun block cipher. Pada penelitian ini, akan dirancang suatu system pengamanan data pada video *Surveillance,* dengan cara mengenkripnya menggunakan VEA dan kunci tertentu, lalu memberikan hak akses secara aman kepada orang yang benar-benar berhak tersebut. Tujuan dari penelitian ini adalah untuk menganalisa performansi dari algoritma VEA dalam hal waktu proses enkripsi dan dekripsi, maupun *delay*-nya. Dalam sistem ini di peroleh pengujian yang menunjukan bahwa modifikasi algorima VEA *(Video Encryption Algorithm)* dengan *generate key* tertentu dapat mengenkripsi dan mendekripsi video streaming secara real-time karena menghasilkan delay kurang dari satu second.

Kata kunci: VEA, video streaming, surveillance, real-time

ABSTRACT

Security is the most important aspect of data exchange. Usually, the data is transmitted shown only for certain parties the data must accepted to entitled user with maintained confidentiality, without known by other people who want to see. Therefore, to keep the data, it needs a data encrption method, which is the science hiding information. One algorithm reliable enough in securing a real-time data is VEA (Video Encryption Algorithm). VEA can be implemented in the video streaming because the algorithm that can be based stream ciphers and block ciphers. In this final project has been designed a system to secure data on video surveillance, with VEA and encrypt it using a secret keyword, and then providing secure access to the person who really entitled to it. The purpose of this study was to analyze the performance of the algorithm in terms of time VEA encryption and decryption process, and delay it. The software is built using the Java programming language. The result of this system shows that VEA *(Video Encryption Algorithm*) algorithm modification with particular generated key could encrypt and decrypt video streaming with real-time because the delay lesser than one second.

Keywords: *VEA, video streaming, surveillance, real-time*

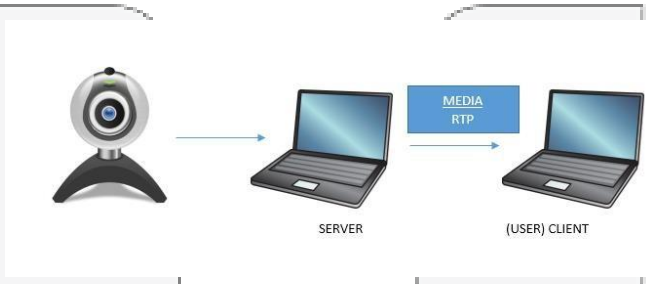
1. PENDAHULUAN

Kemajuan teknologi membuat penggunaan video *surveillance* di beberapa ruangan semakin penting guna meningkatkan keamanan dan *privacy* bagi penggunanya. Adanya video *real-time* yang berfungsi merekam suatu gambar pada suatu kegiatan tentu penting bagi beberapa instansi seperti perbankan, perkantoran, pertahanan negara, dan lain-lain. Maka dari itu kerahasiaan data pun semakin ditingkatkan, salah satunya dengan metode kriptografi yang menggunakan algoritma VEA.[1] Kriptografi adalah proses mengubah data asli menjadi bersandi atau tidak dapat dipahami oleh pembaca jika tidak memiliki kunci sandinya. Kunci yang digunakan biasa nya bersifat simetris dan asimetris. Pembaca yang telah memiliki kuncinya maka dapat mengubah data bersandi atau tidak bisa dibaca tadi menjadi data asli. Video terenkripsi itulah yang nantinya diterima oleh user yang tidak berhak memasukkan kunci tertentu yang telah didefinisikan. Algoritma ini memiliki karena tingkat keamanannya yang cukup memuaskan, komputasi yang ringan, dan cocok diimplementasikan di lingkungan video *streaming* karena algoritmanya yang dapat berbasis stream cipher maupun block cipher, tergantung kebutuhan saat *streaming* video tersebut.

2. TEORI DASAR

2.1 Video *Streaming*

Streaming adalah sebuah jenis layanan yang dapat langsung mengolah data yang diterima tanpa menunggu seluruh data selesai terkirim.[3]. Keunggulan streaming adalah cocok digunakan pada content yang tidak terbatas waktunya. Ide dasar dari video *streaming* adalah membagi paket video ke dalam beberapa bagian, mentransmisikan paket tersebut, kemudian penerima dapat mendecode dan memainkan potongan paket file video tanpa harus menunggu seluruh file terkirim ke mesin penerima.[4]



Gambar 1 Ilustrasi Proses Streaming

2.2 VEA *(Video Encryption Algorithm)*

Video Encryption Algorithm dikembangkan oleh Shi and Bhargava (1998)[5]. Algoritma ini memiliki kemudahan dalam berbagai macam modifikasi yang disesuaikan berdasarkan kebutuhan. Vea merupakan algoritma berbasis stream cipher yang memiliki waktu enkripsi yang pendek, namun memiliki tingkat keamanan yang relative rendah, sehingga dibutuhkan algoritma lain sebagai kunci untuk menambah keamanan pada video yang akan dienkripsi.

Algoritma VEA umum digunakan untuk keperluan enkripsi video karena kemudahannya dalam implementasi, terutama karena algoritma ini mengenkripsi video bit per bit. Algoritma VEA selanjutnya dimodifikasi sedemikian rupa agar cocok diimplementasikan terhadap model enkripsi video *streaming*. Modifikasi dilakukan dengan menambahkan algoritma kriptografi kunci rahasia Rabbit. Selain itu juga, operasi yang akan dilakukan oleh modifikasi algoritma VEA bukan lagi bit per bit namun per paket-paket data yang akan dikirim transmitter.

2.3 Kunci Rabbit

Algoritma Rabbit adalah algoritma yang temukan oleh *Fast Software Encryption* pada tahun 2003 oleh martin Boesgaard, Mette Vesterager, Jesper Christiansen, dan Ove Scavenius. Algoritma ini menerima kunci

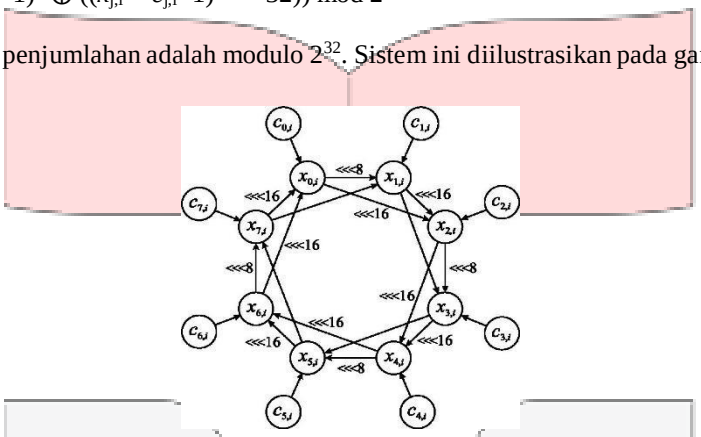
sepanjang 128 bit dan *initial vector* sepanjang 64 bit. Algoritma ini mengolah kunci dan *initial vector* tersebut menjadi sebuat deretan bit semu acak sepanjang 128 bit [7]. Algoritma ini adalah algoritma kunci simetri[8].

Algoritma Rabbit memiliki tujuh belas peubah. Delapan peubah *inner state* dengan ukuran masing-masing 32 bit, delapan *counter* dengan ukuran maing-masing 32 bit, dan satu carry berukuran satu bit. Algoritma Rabbit memiliki beberapa skema. Skema-skema tersebut diantaranya skema persiapan kunci, skema next state function dan skema ekstraksi. Skema persiapan kunci adalah skema untuk memberikan nilai awal pada semua *inner state, counter, dan carry*. Skema *next state function* adalah skema yang menjelaskan aturan pengubahan nilai nilai semua peubah. Skema ekstraksi berfungsi untuk megambil bit-bit dari inner state dengan susunan tertentu menjadi rangkaian bit berukuran 128 dan semu acak [9].

Fungsi utama dari algoritma Rabbit terletak pada persamaan berikut:

$g_{j,i} = ((x_{j,i} + c_{j,i}+1)^2 \oplus ((x_{j,i} + c_{j,i}+1)^2 >> 32)) \bmod 2^{32}$

dimana semua penjumlahan adalah modulo 2^{32}. Sistem ini diilustrasikan pada gambar



Gambar 2 Grafik Ilustrasi Sistem

2.4 *Surveillance*

Surveillance dalam Bahasa Ingris memiliki arti "pengawasan, penjagaan, pengamatan". Video *surveillance* merupakan video yang digunakan untuk keperluan pengawasan. Video yang direkam berasal dari kamera. Kamera pengintai adalah teknologi yang dirancang sebagai alat pemantauan keamanan. Tindak pidana menjadi alasan utama penggunaan kamera pengintai. Kamera pengintai telah diterapkan di banyak teknologi dan dalam beberapa versi seperti CCTV, IP Camera, maupun WebCam.

3. PERANCANGAN SISTEM

3.1. Gambaran Umum Sistem

Perancangan yang di buat adalah aplikasi tampilan video. Dimana terdapat dua user yang dapat mengaksesnya, yaitu dari sisi server dan dari sisi client. Jika user berperan sebagai server, maka server akan merekam video dari webcam. Kriptografi yang digunakan untuk mengamankan video adalah algoritma VEA dengan kunci Rabbit. Kunci yang di masukan terdiri dari 128 bit kunci rahasia dan 64 bit *Initialization Vector* (IV). Setelah itu video akan terenkripsi dan di kirimkan oleh user melalui *Transmission Control Protocol (TCP)*. Video terenkripsi tersebut akan di terima oleh sisi client yang telah terhubung dengan server. Jika client memiliki kunci yang sama dengan kunci yang di server, maka video dapat terdekripsi. Kondisi ini dapat dilakukan jika antara server dan client saling terhubung pada koneksi jaringan yang sama.

3.2 Perancangan Sistem

Adapun perancangan system dari aplikasi video adalah sebagai berikut :

3.2.1 Skema Umum Proses Enkripsi Dekripsi

Data yang di rekam melalui kamera merupakan Data Raw. Data Raw atau *Raw Data* adalah data mentah yang didapatkan dari berbagai sumber data dan informasi, yang belum diolah maupun di analisa. Data tersebut kemudian di convert menjadi byte lalu di XOR-kan dengan kunci yang di hasilkan dari algoritma Rabbit. Data

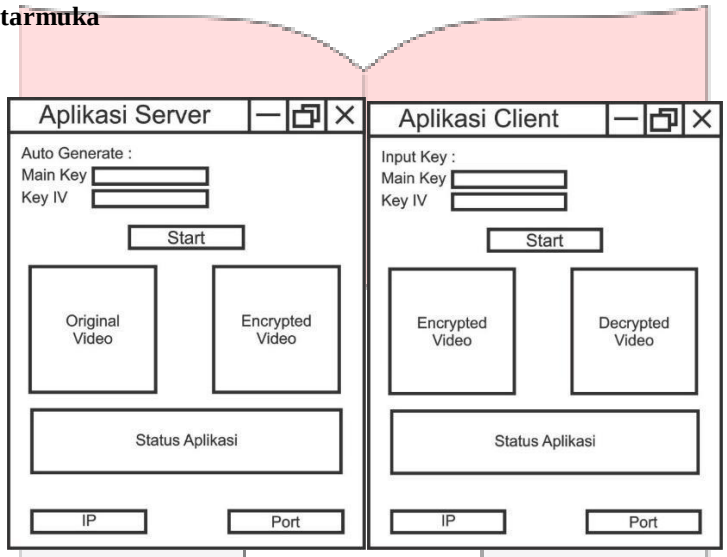
yang telah terenkripsi kemudian di tampilkan. Untuk mendekripsi video, data terenkripsi tadi di XOR-kan kembali dengan kunci yang sama, karena Algoritma VEA maupun Rabbit merupakan algoritma kunci simetris. Berikut adalah data flow untuk proses enkripsi dan dekripsi nya.

3.2.2 **Skema Umum Pembangkitan Kunci**

Penjelasan secara singkat mengenai kerja algoritma kunci Rabbit sebagai berikut.

1. Menentukan nilai awal pada *inner state, counter, dan carry* sesuai dengan skema persiapan kunci
2. Nilai di ubah dengan fungsi *next state*
3. Hasil nilai yang telah di ubah kemudian di ekstrak dengan skema ekstraksi menjadi 16 bilangan bulat (s1)
4. Blok pertama plaintext (Bi) di xor-an dengan Si, menghasilkan blok ciphertext pertama (Ci)
5. Perulangan terjadi jika masih ada blok yang akan di enkripsi dengan memanggil fungsi next state kembali. Untuk menghasilkan Sn yang akan di xor-an dengan bn menjadi cn.
6. fungsi next state di panggil sebanyak 4 kali

3.3 Perancangan Antarmuka



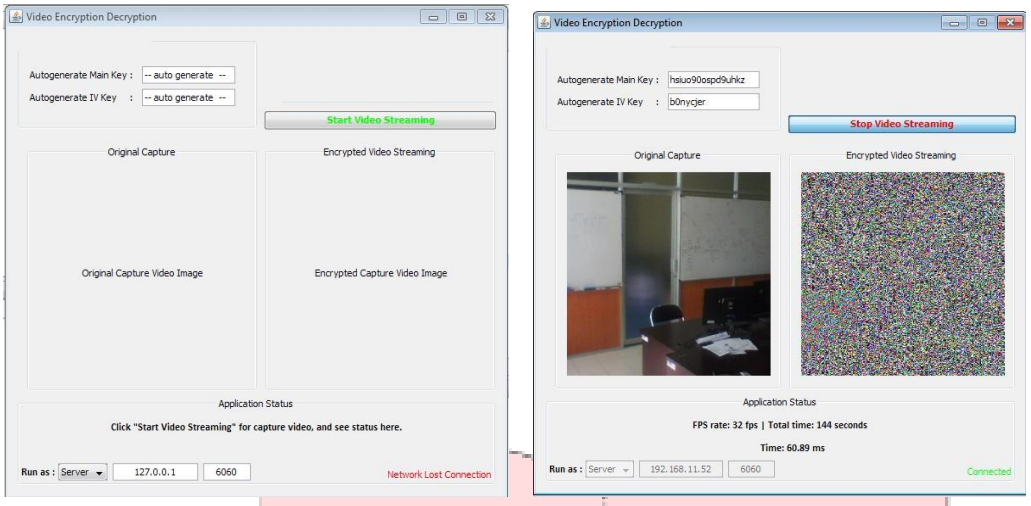
Gambar 4. Perancangan Server dan Client

3.4 Implementasi Antarmuka Server

Implementasi antarmuka /user interface merupakan tampilan dari aplikasi video yang akan menampilkan video asli dan video terenkripsi pada sisi *server*, serta video terenkripsi dan terdekripsi pada sisi server. Berikut adalah tampilan antar muka yang telah di implementasikan :

3.4.1 Tampilan Server

Tampilan ini merupakan tampilan awal yang hanya dapat diakses oleh user sebagai servernya. Pada tampilan server terdapat I untuk inputan IP dan Port yang dimiliki oleh server. Selain itu user dapat mengklik tombol Start Video *Streaming* untuk mulai merekam.



Gambar 5 Tampilan Server

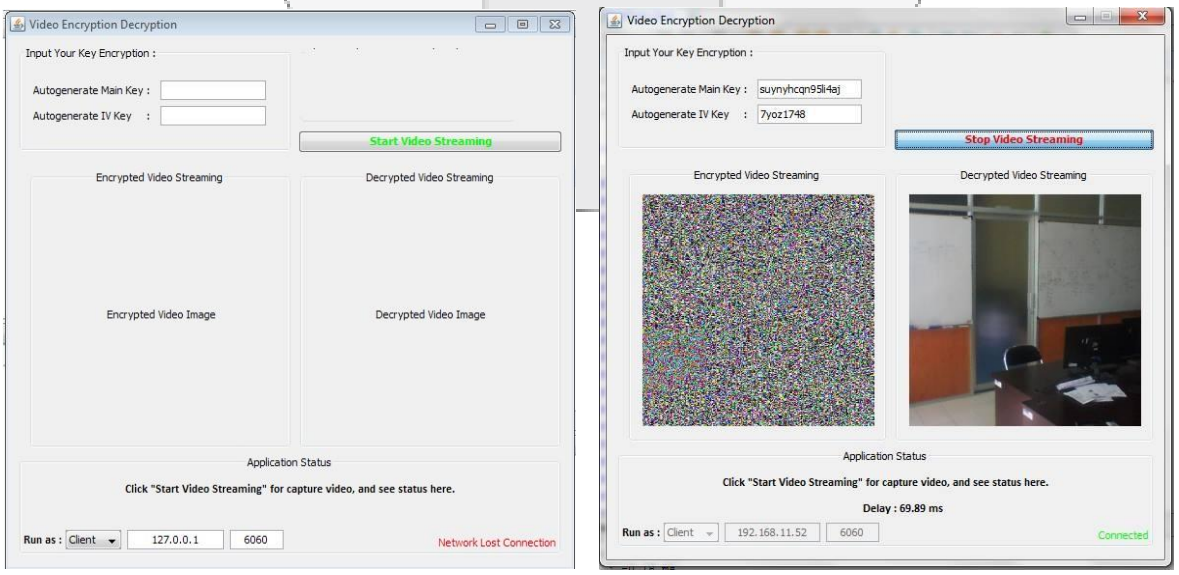
Pada saat video mulai direkam, secara otomatis aplikasi menampilkan generate key yang di hasilkan secara random pada *text field Main Key* dan *IV Key*. Tampilan server yang mulai merekam video akan otomatis menampilkan *generate key*. Pada tampilan server menampilakn video dalam dua kondisi, yaitu kondisi video asli dan video yang telah terenkripsi.

3.5 Implementasi Antarmuka Client

Implementasi antarmuka /user interface merupakan tampilan aplikasi video yang akan menampilkan video enkripsi yang di peroleh dari server, dan menampilkan tampilan video yang telah terdekripsi. Berikut adalah tampilan antar muka yang telah di implementasikan :

3.5.1 Tampilan awal Client

Tampilan awal client merupakan tampilan yang akan digunakan *user* sebagai *client*. Tampilan awal client tidak jauh berbeda dengan tampilan awal server. Hanya saja client harus menginputkan IP dan Port yang sesuai dengan yang telah di tentukan oleh server. Begitupun dengan kunci untuk mendekripnya, *client* harus memiliki kunci yang sama dengan kunci ketika mengenkrip.

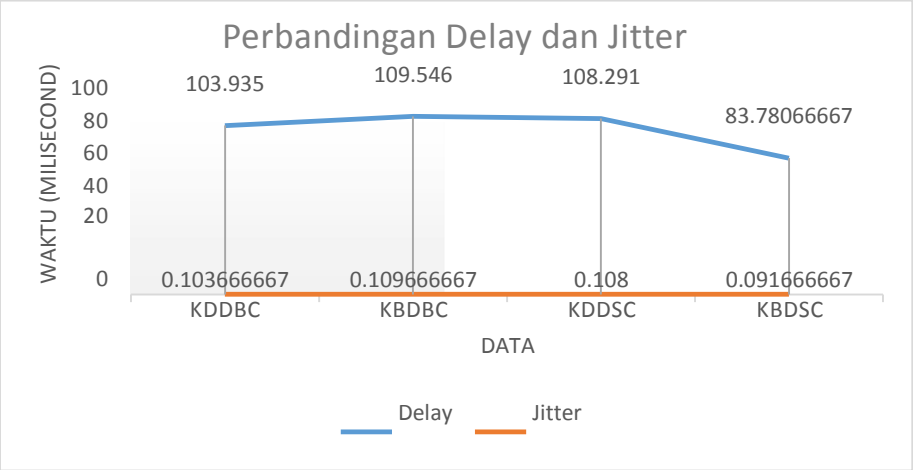


Gambar 6 Tampilan Awal Client

5. Pengujian dan Analisis

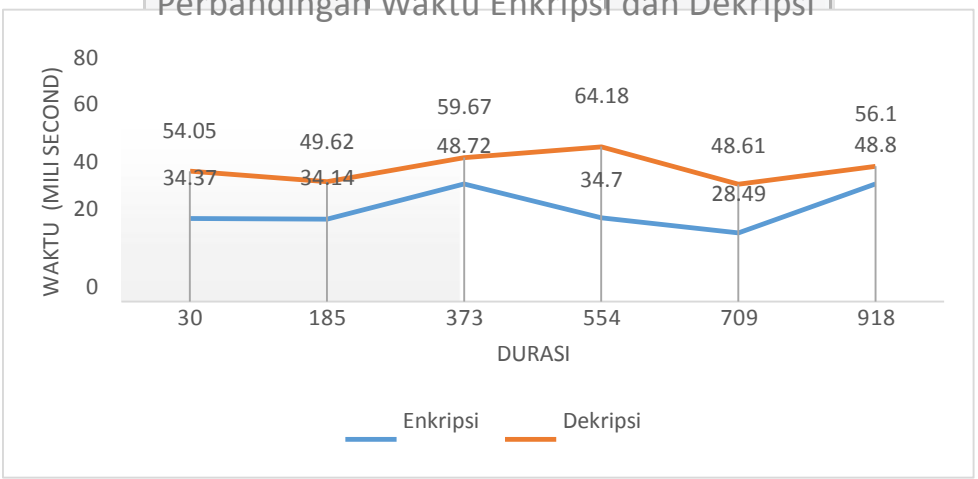
Proses perhitungan enkripsi dimulai saat video mulai merekam video sampai menampilkannya. Rata-rata waktu yang dibutuhkan untuk mengenkripsi video adalah 45.2235 ms. Sedagkan rata-rata proses dekripsi di mulai saat menerima gambar sampai menampilkan video terdekripsi yaitu sebesar 71 ms.

Berikut adalah perbandingan antara *Delay* dan *Jitter* pada kondisi tertentu.



Gambar 4.5.5 (a) Perbandingan Delay dan Jitter

Hasil dari perbandingan diatas dapat disimpulkan bahwa tidak ada delay yang terlalu signifikan. Range delay hanya berada diantara 87-103 milisecond, hal tersebut masih bias dimaklumi karna delay tidak terlalu besar. Sistem memiliki *jitter* yang mendekati 0, itu berarti sitem memiliki nilai *jitter* yang baik.



Gambar 4.5.5 (b) Perbandingan Waktu Enkripsi dan Dekripsi

Hasil dari pengujian bahwa Algoritma VEA dengan key Rabbit dapat melakukan proses enkripsi dan dekripsi dalam waktu singkat yaitu rata-rata 45.2235 ms untuk dan proses dekripsi rata-rata selama 71 ms.

6. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian Tugas Akhir ini adalah:

Kesimpulan yang dapat diambil dari penelitian Tugas Akhir ini adalah:

1. Modifikasi Algoritma VEA dengan kunci rabbit menghasilkan video enkripsi yang sangat acak dan memiliki nilai rata-rata *avalanche effect* 49.9%-50%

2. Berdasarkan hasil pengujian yang telah dilakukan cahaya mempengaruhi frame rate. Semakin terang cahaya yang direkam, semakin banyak pula frame rate yang dihasilkan. Hal ini terlihat dari perbandingan jumlah Frame pada Siang hari yakni rata-rata sebesar 30 FPS, sedangkan pada malam hari rata-rata hanya sebesar 15 FPS.

3. Berdasarkan pengujian proses enkripsi dan dekripsi dalam waktu singkat yaitu rata-rata 45.2235 ms untuk dan proses dekripsi rata-rata selama 71 ms.

4. Berdasarkan pengujian nilai Datarate dan Bandwith adalah hampir sama, yaitu 3,984,187 bps untuk Datarate dan 3.984 Mbps untuk Bandwidth

DAFTAR PUSTAKA

[1] Oni, Marvello. “Algoritma Enkripsi pada Video MPEG”. Sekolah Teknik Elektro dan Informatik Institut Teknologi Bandung

[2] Shanti D.P.M dan Achmad Affandi, “Rancang Bangun Sistem Keamanan Konten Video On Demand (Vod) Pada Internet Protocol Television (Iptv) Menggunakan Video Encryption Algorithm (VEA)”, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

[3] Suhendra, Made. "Analisa Performansi Live Streaming Dengan Menggunakan Jaringan Hsdpa". Institut Teknologi Sepuluh Nopember. 2007

[4] Satwika, I Kadek Susila. "Proses Video Streaming Dengan Protocol Real Time Streaming Protocol (Rtsp)". Universitas Udayana. 2011.

[5] Bhagarva, Bharat. Shi, Changgui. Wang, Sheng-Yih, “MPEG Video Encryption Algorithms”, Purdue University, 2002.

[6] Intania Savitri, Dian. “Perancangan dan Implementasi Modifikasi Algoritma VEA (Video Encryption Algorithm) untuk Video Streaming”. Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[7]] Akhyar, Fikaril.”Rabbit Algorithm for Video on Demand”, IEEE Asia Pacific Conference on Wireless and Mobile, 2015

[8] Martin Boesgaard, Mette Vesterager, Thomas Christens, Erik Zenner “The Stream Chipher Rabbit”. 2003

[9] Paramita “ Studi dan Analisis Mengenai Algoritma Cipher Aliran Rabbit”