2

JURNAL INFORMATIKA

ISSN: 2087-5266

NO.2 VOL.IV | MEI-AGUSTUS 2013

PEMBANGUNAN APLIKASI *WEB EVENT CALENDAR* MENGGUNAKAN ALGORITMA RIJNDAEL UNTUK ENKRIPSI DATA

DEWI ROSMALA[1], HANIF WIBOWO[2]

Jurusan Teknik Informatika
Institut Teknologi Nasional Bandung

d_rosmala@itenas.ac.id, bowo.wowowo30@gmail.com

ABSTRAK

Web Event Calendar pada dasarnya digunakan untuk perusahaan event organizer sebagai sarana pengumpul informasi. Aplikasi ini membantu para developer serta pengembang sistem untuk mempermudah menggelar sebuah event dalam mengatur konsep dan rencana. Namun ketika aplikasi web menjadi suatu aplikasi yang mudah diakses, menjadikan sebuah jaringan rentan dari ancaman pencurian informasi penting, menurut buku "Membangun Bisnis Event Organizer" yang ditulis oleh Yudhi Megananda, akses data yang sering terjadi menjadikan sebuah jaringan rentan dari ancaman pencurian informasi penting. Kerentanan suatu jaringan dapat membuat suatu data event ataupun informasi menjadi tidak terorganisir dengan baik. Baik itu kerahasiaan data event ataupun originalitas data. Oleh karena itu diterapkan algoritma Rijndael yang berfungsi untuk menjaga kerahasiaan data, keaslian data serta keaslian pengirim. Kerahasiaan data dapat dienkripsi menjadi ciphertex yang disimpan didalam database. Pesan ciphertext berisi seluruh informasi yang tidak dapat dibaca manusia ataupun komputer tanpa menggunakan mekanisme yang tepat untuk melakukan dekripsi. Penggunaan algoritma ini akan mengamankan konten dari halaman web, server web sampai client.

***Kata kunci**: Web Event Calendar, enkripsi, ciphertex, Algoritma Rijndael*

ABSTRACT

Web Event Calendar is basically used for corporate event organizer as a means of collecting information. The application helps developers system to make it easier to roll out an event in organizing the concepts and plans. However, when a web application is easily accessible, making a network vulnerable from the threat of theft of important information. Therefore implemented the algorithm Rijndael, which serves to maintain the confidentiality of data, the authenticity of the data and authenticity of the sender. The confidentiality of encrypted data can be stored in the ciphertex database. Ciphertext message contains all the information that is not human readable, or the computer without using the proper mechanism to decrypt. The use of this algorithm will secure content from a web page, the web server to the client.

***Keywords**:* Web Event *Calendar, encryption,* ciphertex, *Algoritma Rijndael*

PENDAHULUAN

Web event calendar adalah sebuah aplikasi yang digunakan untuk perusahaan *event organizer* sebagai sarana pengumpul informasi. Hal – hal mengenai penjadwalan sebuah *event* diinformasikan terpusat didalam aplikasi ini. Aplikasi *web event calendar* membantu para developer serta pengembang sistem untuk mempermudah perwujudan rencana menggelar sebuah *event.* Mewujudkan kesuksesan sebuah *event* merupakan sebuah kerja keras dan dibutuhkan konsep dan rencana yang benar-benar matang. Tanpa adanya rencana yang jelas dan terarah, tentu saja acara tidak bisa berlangsung dengan sukses. Oleh karena itu aplikasi *web event calendar* ini mampu mengorganisir konsep dan rencana agar acara yang direncanakan jelas arahnya.

Namun ketika aplikasi *web* menjadi suatu aplikasi yang mudah diakses oleh semua orang, disebutkan didalam buku "Membangun Bisnis Event Organizer" yang ditulis oleh Yudhi Megananda, akses data yang sering terjadi menjadikan sebuah jaringan rentan dari ancaman pencurian informasi penting. Terhitung sejak tahun 2011 sekitar 12 EO (*Event Organizer)* di Surabaya mengalami kerugian yang sangat besar akibat adanya perubahan-perubahan data yang tidak diketahui yang menyebabkan *miss* komunikasi antara developer dan pengembang sistem. Kerentanan suatu jaringan dapat membuat suatu data *event* ataupun informasi menjadi tidak terorganisir dengan baik. Kerahasiaan data dan *originalitas* data dapat dicuri dengan menggunakan serangan tertentu melalui jaringan. Jenis serangan yang sering dilakukan yaitu *sniffing, replay attack, spoofing, man-in-the-middle.*

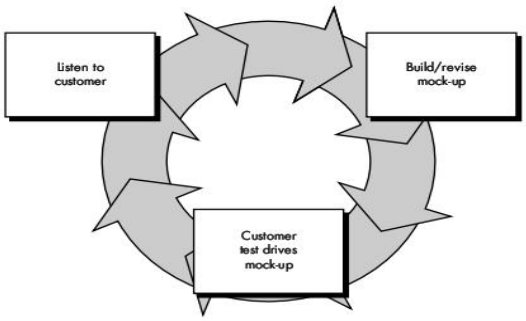
Untuk menghindari pencurian data tersebut, di implementasikan sebuah dukungan *web security* yang dapat menjadikan aplikasi menjadi lebih aman dari pencurian informasi data. Pengamanan sistem informasi memiliki beberapa dasar-dasar dan teori-teori yang digunakan. Kriptografi, enkripsi, dan dekripsi merupakan dasar dari pengamanan sistem informasi. Ada beberapa algoritma atau metoda dalam enkripsi dan dekripsi yang saat ini sedang berkembang seperti *Data Encryption Standard* (DES), *International Data Encryption Algorithm* (IDEA), Rivest-Shamir and Adleman (RSA), Algoritma Rijndael dan masih banyak lagi.

Pada penelitian ini dititik beratkan pada algoritma Rijndael yang dikenal juga dengan *Advanced Encryption Standard* (AES) yang merupakan perkembangan dari DES. Algoritma Rijndael melakukan enkripsi data dari *plaintext* menjadi *ciphertext* agar data tidak bisa dibaca oleh siapapun sehingga dapat menjaga kerahasiaan data, keaslian data serta keaslian pengirim.

METODOLOGI

Identifikasi Prototype pada Aplikasi Web *Event Calendar*

Dalam pembangunan sistem digunakan metodologi prototype yang memiliki 3 tahapan dasar, di setiap tahapan tersebut di implementasikan untuk membantu pembangunan aplikasi *web event calendar* ini. Dari setiap fase dijelaskan tahapan yang harus dilakukan dan fase tersebut akan berhubungan satu sama lainnya. Untuk lebih jelasnya bagan metodologi prototype dapat dilihat pada Gambar 1.



Gambar 1. Bagan Life Cycle Prototype

Dari Gambar 1 terlihat fase-fase yang terdapat pada proses prototype, dari setiap fase akan berhubungan satu dengan yang lain dan tidak terpisah. Pada sub bab selanjutnya dijelaskan secara spesifik apa saja yang di lakukan dari setiap fase yang ada. Sehingga dalam pembangunan aplikasi *web event calendar* sesuai dengan kerangka kerja yang telah ditentukan sebelumnya agar mendapatkan hasil akhir yang baik sesuai dengan apa yang diharapkan.

User Requirement

Tahapan ini merupakan tahapan awal pembangunan aplikasi *web event calendar*. Seperti yang diketahui dalam membangun sebuah sistem harus dilakukan proses *User* requirement yang pada Gambar 1 digambarkan dengan "*listen to customer*" dengan wawancara dan mendengarkan keinginan atau kebutuhan konsumen terhadap aplikasi yang akan dibuat. Dalam hal ini ada dua analisis kebutuhan yang dicapai.

Analisis Kebutuhan Aplikasi

Aplikasi yang dibutuhkan adalah aplikasi *web event calendar* yang mampu mengatur *event* dan informasi secara terpusat dengan memenuhi fungsi *insert, update, delete* sebagai fitur untuk mengatur data *event* dan juga ditambahkan fungsi keamanan agar terhindar dari pencurian dan perubahan data *event* atau informasi.

Analisis Kebutuhan Keamanan

Keamanan yang dibutuhkan adalah yang mampu memberikan perlindungan terhadap bagian *login* dan konten dari serangan *sniffing*. Keamanan yang dilakukan berupa enkripsi pada input data sehingga data yang dicuri tidak bisa dibaca. Data yang di enkripsi yaitu data *event* dan password *user*.

Prototype Building

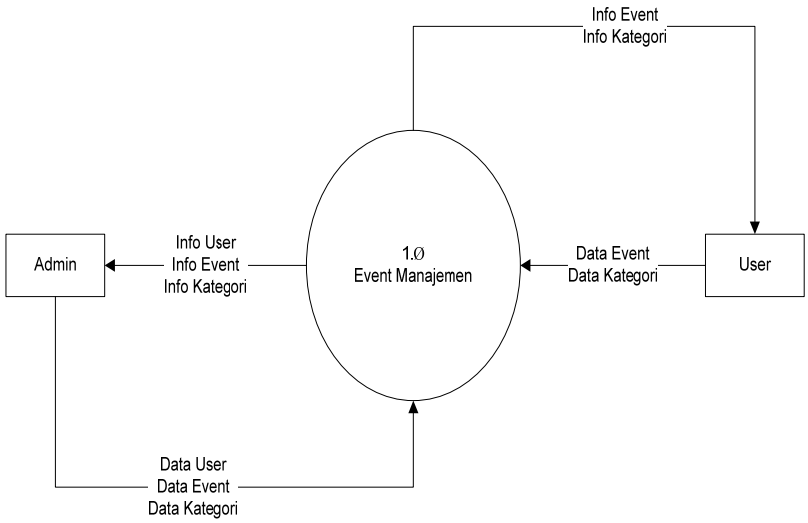
Merupakan tahap lanjutan dari fase sebelumnya yang bertujuan untuk memindahkan kebutuhan-kebutuhan dari fase *User Requirement* kepada rancangan pembangunan contoh aplikasi. Pada tahap ini dibangun prototype dari aplikasi yang dapat dilihat pada Gambar 1 sebagai "*buid/revise mock-up*" yang memiliki tahapan perancangan proses sistem, perancangan *database*, perancangan alur sistem dan perancangan antarmuka.

Perancangan Proses Sistem

Dalam tahap ini dijelaskan proses-proses yang berlangsung dalam aplikasi sistem web *event calendar*. Proses tersebut digambarkan dengan menggunakan DFD (*Data Flow Diagram*). Dalam penggambaran proses tersebut, DFD dibagi menjadi dua bagian yaitu *Context Diagram* dan DFD Level 1.

Context Diagram

Context diagram merupakan tingkatan tertinggi dalam diagram alir data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Diagram context ini menggambarkan alur proses yang dilakukan oleh sistem seperti output yang diberikan oleh sistem dan input yang diterima oleh sistem yang digambarkan secara umum yang dapat dilihat pada Gambar 2.



Gambar 2. Context Diagram

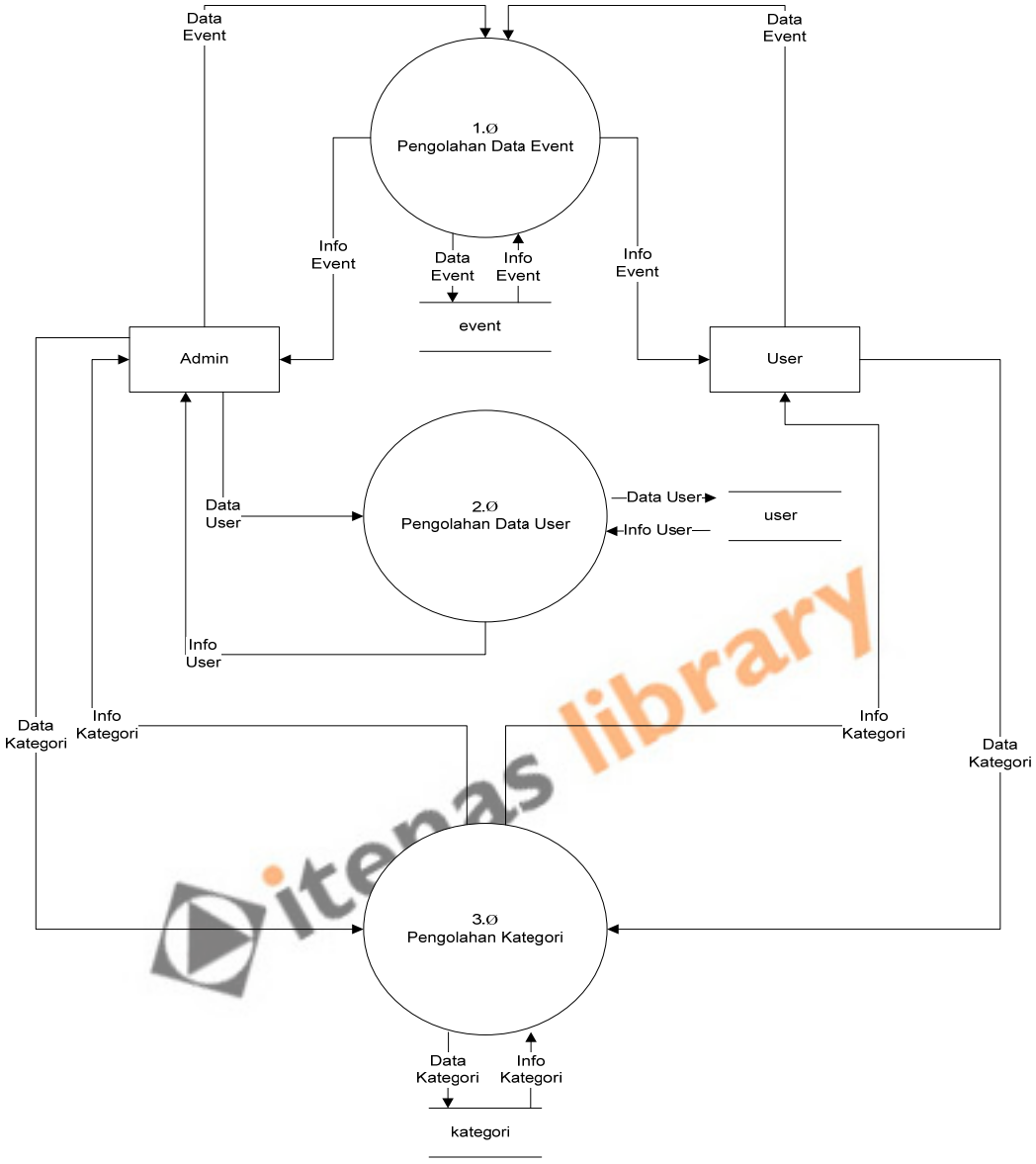
Terdapat dua entitas yang berhubungan langsung dengan proses tersebut yaitu admin dan *user*. Dalam aplikasi ini admin mengelola segala proses yang ada di dalam sistem baik itu *insert, update, delete,* manajemen *user* maupun manajemen *event*. *User* hanya dapat mengelola *insert, update dan delete*. Admin akan memberikan inputan berupa data *user* dan data *event* kemudian sistem akan mengeluarkan output kepada admin berupa info *user* dan info *event.User* akan memberikan inputan data *event* kemudian system akan mengeluarkan output berupa info *event* kepada *user*. Pada Tabel 3 berikut memperlihatkan alur input dan output dari entitas eksternal ke proses sistem aplikasi web *event calendar*.

Tabel 1. Tabel Alur Data

Entitas	Alur Input	Alur Output
Admin	Data User	Info User
	Data Event	Info Event
	Data Kategori	Info Kategori
User	Data Event	Info Event
	Data Kategori	Info Kategori

***DFD (Data Flow Diagram)* Level 1**

Pada tahap selanjutnya adalah DFD Level 1, pada tahapan ini akan dijelaskan secara mendalam proses-proses yang terdapat di aplikasi web *event calendar*. DFD level 1 pada aplikasi ini akan lebih menjelaskan proses yang terjadi pada sistem yang dapat dilihat pada Gambar 3.



Gambar 3. DFD level 1

Perancangan *Database* Sistem

Berikut ini merupakan perancangan dari sistem *database* yang dibuat

ERD

Sesuai dengan kebutuhan sistem yang akan dibangun, terdapat entitas yang saling berhubungan, dapat dilihat pada Gambar 4.

Gambar 4. ERD

Dari Gambar 4 dapat diketahui relasi yang terjadi antar entitas, derajat relasi yang digunakan dalam perancangan ini adalah satu ke banyak (one to many). Penentuan relasi ini didapat dari hasil analisa yang disesuaikan dengan kebutuhan sistem yang dibangun.

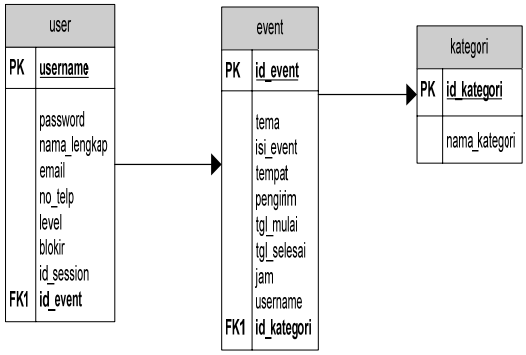
TRD (Table Relationship Diagram)

Setelah menentukan entitas yang terhubung dalam sistem yang akan dibangun maka dapat ditentukan atribut-atribut yang terdapat pada setiap entitas sesuai dengan yang dibutuhkan, ditunjukan pada Tabel 2.

Tabel 2. Atribut *Database*

Entitas	Atribut
Event	id_event
	Tema
	Isi_event
	Tempat
	Pengirim
	Tgl_mulai
	Tgl_selesai
	Jam
	Username
	Id_kategori
User	Username
	Password
	Nama_lengkap
	Email
	No_telp
	Level
	Blokir
	Id_session
Kategori	Id_kategori
	Nama_kategori

Dari atribut yang telah ditentukan maka dapat dibentuk sebuah relasi antar tabel berdasarkan entitas-entitas yang telah ada sebelumnya. Berikut gambar relasi antar tabel pada Gambar 5.



Gambar 5. TRD

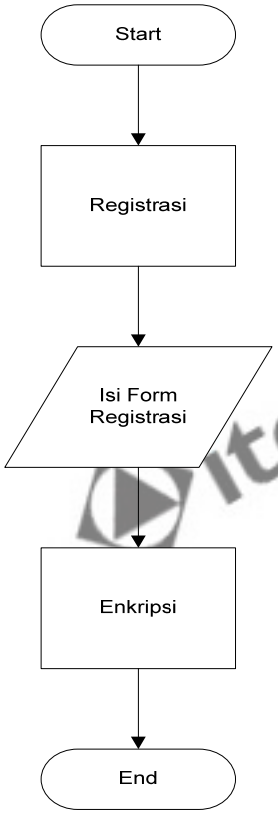
Dari Gambar 5 dapat dijelaskan TRD yang dibuat sesuai dengan yang telah ada pada ERD sebelumnya.

Perancangan Alur Sistem

Penggambaran alur kerja sistem dalam aplikasi *web event calendar* digambarkan dengan Flowchart. Proses alur sistem yang dibahas adalah sistem *Login*, Registrasi, dan Algoritma *Rijndael*.

Proses Registrasi

Berikut merupakan rancangan seperti yang terlihat pada Gambar 6 *flowchart* dari proses registrasi yang diberikan fungsi keamanan pada data seperti melakukan enkripsi untuk merubah data *plaintext* menjadi *ciphertext*.



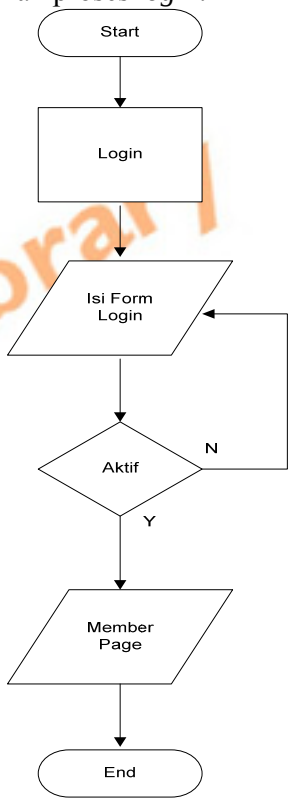
Gambar 6. Flowchart Registrasi

Keterangan:

1. Proses dimulai, *User* yang belum terdaftar tidak bisa langsung *Login*, maka *User* perlu melakukan proses registrasi.
2. Setelah itu *User* diminta untuk mengisi form registrasi.
3. Kemudian dilakukan proses enkripsi terhadap data yang di inputkan yang bertujuan untuk menghindari dari pencurian data kemudian data akan disimpan di dalam *database*.
4. Proses selesai.

Proses Login

Pada Gambar 7 merupakan rancangan proses *login* yang dilakukan oleh *user* aktif agar dapat mengelola *event*. Tidak semua *user* dapat melakukan *login*, hanya *user* aktif atau *user* yang telah diberi izin oleh admin yang dapat melakukan proses *login*.



Gambar 7. Flowchart Login

Keterangan:

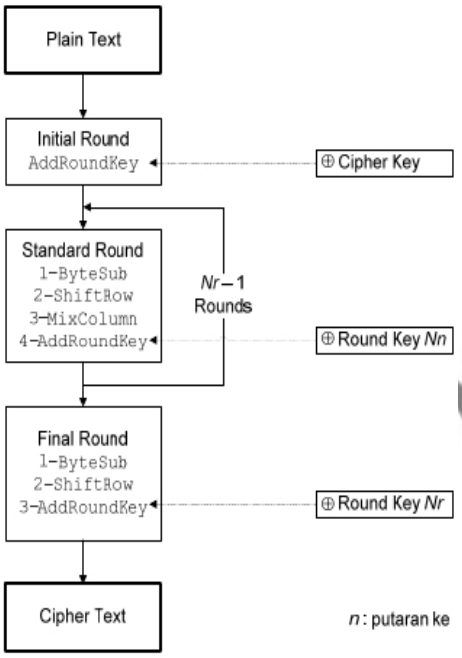
Proses Login pada Gambar 7 dijelaskan sebagai berikut:

1. Pertama *user* berada di halaman login
2. Kemudian *user* mengisi form login yang berisi data *username* dan password

3. Jika telah selesai melakukan input data maka akan dilakukan pengecekan keaktifan user. Apabila *user* tidak aktif maka proses tidak akan dilanjutkan dan kembali ke form *login*
4. Jika *user* aktif proses dilanjutkan menuju *member page*
5. Proses *login* selesai.

Proses Algoritma *Rijndael*

Adapun proses Algoritma *Rijndael* didalam sistem yang dijelaskan pada Gambar 8.



Gambar 8. Flowchart Algoritma Rijndael

Berikut adalah penjelasan flowchart *Rijndael* pada Gambar 8:

1. *Plaintext* merupakan array yang berukuran 16-*byte* yang berisi data masukan.
2. Pada proses ini *Initial Round* melakukan *XOR* antara *state* awal (*plaintext*) dengan *cipherkey*, tahap ini disebut juga *AddRoundKey*.
3. Lalu putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:

a. *SubBytes*: operasi melakukan substitusi dengan cara mengganti setiap *byte state* dengan *byte* pada sebuah tabel yang dinamakan tabel S-Box.
b. *ShiftRows*: proses ini beroperasi pada tiap baris dari tabel *state.* Proses ini bekerja dengan cara memutar *byte-byte* dengan jumlah putaran yang berbeda-beda.
c. *MixColumns:* proses ini beroperasi pada tiap kolom dari tabel *state.* Operasi ini menggabungkan *byte* dari setiap kolom tabel *state* dan menggunakan transformasi linear
d. *AddRoundKey:* Melakukan *XOR* antara *array state* sekarang dengan *round key*.
e. *Final round* merupakan proses untuk putaran terakhir *SubBytes*, *ShiftRows*, dan *AddRoundKey*.
f. *Ciphertext* merupakan array yang berukuran 16-*byte* yang berisi hasil enkripsi.

HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan mengenai hasil pengujian *alpha* yang telah dilakukan sebelumnya baik itu pengujian *alpha* fungsi proses maupun pengujian alpha *source code* beserta kesimpulan dari hasil pengujian kedua tes tersebut. Kasimpulan ini berisi hasil dari keluaran yang diharapkan berdasarkan dengan hasil tes yang telah dilakukan.

Pada Tabel 3 dijelaskan hal-hal yang dilakukan pada saat pengujian alpha fungsi proses *login* pada aplikasi yang dibuat. Tabel ini berisi fitur yang diuji dilengkapi dengan pengamatan berdasarkan hasil pengujian dan kesimpulan yang di dapat.

Tabel 3. Hasil Pengujian Fungsi Proses Sistem

Fitur yang diuji	Hal yang diuji	Pengamatan	kesimpulan
Proses *Insert* Data	Proses *insert* data *event*	Proses *insert* data *event* berhasil	Berhasil
Proses *update* Data	Proses *update* data *event*	Proses *update* data *event* berhasil	Berhasil
Proses *Delete* Data	Proses *delete* data *event*	Proses *delete* data *event* berhasil	Berhasil
Proses *Login*	Proses *insert* data *login* tes positif	Proses *insert* data *login* tes positif	Berhasil
	Proses *insert* data *login* tes negatif	Proses *insert* data *login* tes negative	Berhasil

Berikutnya pada Tabel 4 dijelaskan mengenai hasil pengujian alpha *source code* pada proses algoritma Rijndael yang terdapat pada aplikasi.

Tabel 4. Hasil Pengujian Alpha Source Code

Nama Proses	Input	Hasil Output	Kesimpulan
Proses perubahan data dengan algoritma Rijndael	Pengujian enkripsi	Tp8f+qBYFfN81vIu5iFttb1rljpqvlAb/I9se4IPmj8	Berhasil
	Pengujian dekripsi	Pengujian algoritma rijndael	Berhasil
Serangan *sniffing*	*Capture*	Data tidak dapat dibaca	Berhasil

Dari pengujian yang dilakukan dapat disimpulkan hasil seluruh pengujian algoritma Rijndael pada aplikasi web *event calendar* adalah:

1. Pengujian alpha fungsi sistem

Fungsi *insert*, *update*, *delete* dan *login* yang terdapat pada aplikasi dapat berjalan sesuai dengan yang dirancang. Sehingga dapat disimpulkan fungsi yang pada aplikasi berjalan dengan tepat dan aman.

2. Pengujian alpha source code

Berdasarkan pada Tabel 4 dapat di ketahui bahwa algoritma Rijndael yang di implementasikan berhasil mengubah *plaintext* menjadi sebuah *ciphertext* begitu pula sebaliknya dan juga algoritma yang digunakan telah berhasil menahan dari serangan *sniffing*. Sehingga dapat disimpulkan sistem telah berjalan dan aman sesuai dengan yang diharapkan.

KESIMPULAN

Dari hasil pengujian sistem dapat disimpulkan bahwa aplikasi *web event calendar* mampu mengatur data *event* agar terorganisir sesuai konsep. Serta terlihat pula pada *output* pengujian alpha *source code* bahwa Algoritma Rijndael dapat melakukan proses enkripsi yang dapat merubah *plaintext* menjadi sebuah *ciphertext* dan proses dekripsi yang sebaliknya dapat merubah *ciphertext* menjadi *plaintext.* Dan juga dapat melakukan pengaman terhadap *sniffing* yang membuat semua data yang di inputkan menjadi rahasia. Dengan demikian digunakannya metode algoritma Rijndael memberikan perlindungan yang aman terhadap aplikasi dan juga database karena segala informasi akan dirubah menjadi sebuah *ciphertext.*

DAFTAR PUSTAKA

[1] Munir, Rinaldi. 2006. "Kriptografi". Informatika, Bandung

[2] Sadikin, Rifki. 2012. "Kriptografi untuk Keamanan Jaringan". Andi Publisher

[3] Yolanda, Elfira, 2008. "Implementasi *Disk Encryption* Menggunakan Algoritma Rijndael". Teknik Informatika IT

[4] S'to. 2011. "CEH (*Certified Etchical Hacker*) 400% Illegal". Jasakom

[5] Megananda, Yudhi, 2009, "Membangun Bisnis Event Organizer". BIP (Kelompok Gramedia), Jakarta

[6] Nurdiansyah, Firdaus, 2009, "Rancang Bangun Advance Encryption Standard (AES) Untuk Transfer Email". Teknik Informatika

[7] Wulandari, Ratna, 2010, "Pembuatan Aplikasi Steganografi pada File Audio Mp3 dengan Metode Parity Coding dan Enkripsi Rijndael". Teknik Informatika

[8] Rothe, Jorg. 2005. "*Complexity Theory and Cryptology*". *Springer*