IMPLEMENTASI ALGORITMA RIJNDAEL UNTUK KEAMANAN LOGIN (STUDI KASUS: PERANGKAT LUNAK KEUANGAN PEMBERIAN TUNJANGAN DI KANTOR KOPERTIS WILAYAH IV)

Andy Victor Pakpahan
Program Studi Teknik Informatika
STMIK LPKIA
Email: abang@lpkia.ac.id

Novi Fibriani Prayino
Program Studi Teknik Informatika
STMIK LPKIA
Email: nfibriani@gmail.com

ABSTRAK

Pada perangkat lunak bagian keuangan pengelolaan pemberian tunjangan di Kopertis Wilayah IV didalamnya menangani data – data sensitif yang tidak boleh diakses oleh orang yang tidak berkepentingan. Maka, keamanan informasi pada perangkat lunak merupakan suatu aspek penting yang perlu diperhatikan. Salah satunya yaitu keamanan pada proses *authentication,* Dimana *user* melakukan login untuk masuk pada perangkat lunak dengan memasukan *username* dan *password*. Jika aspek keamanan tidak diperhatikan kemungkinan suatu perangkat lunak dapat dengan mudah disalahgunakan oleh pihak – pihak yang tidak bertanggung jawab. Adapun serangan pada perangkat lunak yang dapat terjadi pada proses *authentication* yaitu penyadapan dengan *tools sniffing.* Proses penyadapan dengan *tools sniffing* digunakan untuk mendapatkan informasi yang ada pada sistem jaringan komputer. Usaha yang dapat dilakukan untuk meningkatkan keamanan antara lain adalah dengan menggunakan teknologi kriptografi. Algoritma Rijndael terpilih sebagai algoritma kriptografi yang dapat melindungi informasi dengan baik, Maka dari itu penulis melakukan penelitian tentang implementasi algoritma tersebut. Dari hasil pengujian Algoritma Rijndael di implementasikan menggunakan javascript pada view login mampu mengamankan data *username* dan *password* dan data mengamankan data password di database. Namun, terdapat titik lemah yaitu mengenai transmisi kunci awal dari proses enkripsi.

Kata kunci: kriptografi**,** rijndael, AES, mode CBC

ABSTRACT

In the financial management software the provision of benefits in Kopertis Region IV in it handles sensitive data that may not be accessed by people who are not interested. So, information security in software is an important aspect that needs attention. One of them is security in the authentication process, where users log in to enter the software by entering a username and password. If the security aspects are not taken into consideration, the possibility of software can be easily misused by irresponsible parties. The attack on the software that can occur in the authentication process is tapping with sniffing tools. The process of tapping with sniffing tools is used to obtain information that is on a computer network system. Efforts that can be made to improve security include the use of cryptographic technology. Rijndael's algorithm was chosen as a cryptographic algorithm that can protect information well, so the authors conducted research on the implementation of the algorithm. From the results of testing the Rijndael Algorithm implemented using javascript in the login view is able to secure username and password data and data secure password data in the database. However, there is a weak point regarding the initial key transmission of the encryption process.

Keywords: *cryptograhy, rijndael, AES, CBC mode*

1. PENDAHULUAN

Perangkat lunak keuangan pemberian tunjangan di Kopertis Wilayah IV digunakan untuk mengelola proses pemberian tunjangan profesi Dosen dan tunjangan kehormatan Profesor. Tunjangan ini merupakan bentuk penghargaan untuk kinerja Dosen [1]. Tentunya pada perangkat lunak ini menangani data – data sensitif dan tidak boleh diakses oleh orang tidak berkepentingan. Maka dari itu keamanan informasi pada perangkat lunak merupakan suatu aspek penting yang perlu diperhatikan. Salah satunya yaitu keamanan pada proses *authentication*, Dimana *user* melakukan login untuk masuk pada perangkat lunak dengan memasukan *username* dan *password*. Proses *authentication* adalah metode untuk melakukan pengecekan apakah dia adalah pengguna yang telah terdaftar atau tidak [2]. Ini merupakan salah satu usaha yang dapat dilakukan untuk menjaga informasi dari orang – orang yang tidak berhak mengakses atau pada aspek keamanan ini disebut dengan *confidentiality* sedangkan jika data tersebut terkait dengan data yang bersifat pribadi disebut dengan istilah *Privacy* [2].

Jika aspek keamanan tidak diperhatikan kemungkinan suatu perangkat lunak dapat dengan mudah disalahgunakan oleh pihak – pihak yang tidak bertanggung jawab. Tujuan dari pengamanan data atau informasi yaitu sebagai tindakan *preventif* supaya tidak terjadi exploitasi data demi menjaga 3 aspek utama yaitu *Confidentiality*, *Integrity* dan *Availability* [3]. Adapun serangan pada aspek *privacy* yaitu dengan usaha untuk melakukan penyadapan (dengan *tools sniifer)*. Proses penyadapan ini digunakan untuk mendapatakan informasi yang ada pada sistem jaringan komputer seperti *password* dan *username* [4] .Pada perangkat lunak berbasis web, saat ini terdapat pengamanan jaringan menggunakan protokol HTTPS dengan memanfaatkan *Secure Socket Layer* (SSL) atau *Transport Layer Security* (TLS) karena protokol tersebut mempunyai keamanan yang tinggi, Tetapi protokol tersebut membutuhkan modal yang besar untuk sertifikat pengamananya dan protokol tersebut biasanya digunakan pada web yang berada pada jaringan internet [5]. sedangkan implementasi perangkat lunak pada penelitian ini dilakukan pada jaringan *Local Area Network* (LAN).

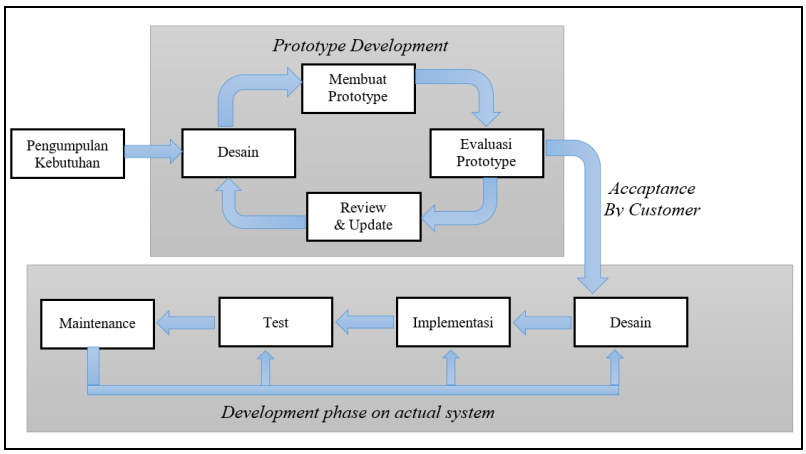
Maka dari itu, usaha yang dapat dilakukan untuk meningkatkan keamanan pada aspek *privacy* dan *confidentiality* antara lain adalah dengan menggunakan teknologi kriptografi [6]. Kriptografi adalah ilmu mengenai teknik enkripsi dimana dilakukan pengacakan pada data – data yang akan di enkripsi dengan menggunakan suatu kunci enkripsi, data yang telah diacak tidak bisa diketahui oleh seseorang yang tidak memiliki kunci enkripsi tersebut [6]. Salah satu algortima kriptografi yang dapat digunakan untuk meningkatkan keamanan adalah Algoritma Rijndael. Algoritma Rijndael terpilih pada kompetisi yang digelar oleh *National Institute of Standard and Technology* (NIST) menjadi algoritma kriptografi *Advanced Encryption Standard (*AES). Algoritma Rijndael terpilih karena dapat melindungi informasi dengan baik dan pada proses implementasinya sangat efisien, selain telah dilakukan analisis pada algoritma rijndael ini bahwa tidak ditemukan celah keamanan seperti lawannya pada kompetisi AES tersebut dan jika terkena serangan algortima ini tidak menyebabkan kerusakan yang berarti. [2]

AES adalah algoritma kriptografi simetri, pada teknik kriptografi ini dilakukan dengan mode penyandian blok *(block chiper).* Salah satu mode operasi yang dapat diterapkan pada AES adalah mode CBC, Pada mode ini proses yang dilakukan jauh lebih rumit dikarenakan bit bit data yang di enkripsi bukan berasal dari *plaintext* langsung melainkan dari bit bit yang telah di enkripsi sebelumnya [7], Oleh karena itu dengan menggunakan mode CBC dapat meningkatkan keamanan data.

Maka pada penelitian ini akan mengimplementasikan Algoritma Rijndael yaitu AES 128bit dengan mode CBC pada sistem login sebagai usaha meningkatkan keamanan pada perangkat lunak.

2. METODOLOGI PENELITIAN

Pada penelitian ini penulis menggunakan konsep SDLC (*Systems Development Life Cycle)* dimana dalam rekayasa perangkat lunak SDLC merupakan konsep yang mendasari berbagai jenis metodologi pengembangan perangkat lunak [8]. Dari berbagai metodologi yang terdapat pada SDLC, dipilih model *prototype* sebagai pilihan metodologi pengembangan perangkat lunak yang akan dibangun, Karena metodologi tersebut sesuai untuk melakukan simulasi sistem serta merupakan suatu teknik untuk mendefinisikan kebutuhan- kebutuhan dari perangkat lunak dengan cepat. Adapun tahapan – tahapan dari model pengembangan *prototype* ini dapat dilihat pada gambar 1.



Gambar 1. *Prototype Model of Software Development*

2.1. Pengumpulan Kebutuhan

Pada tahapan ini dilakukan pengumpulan informasi terkait kebutuhan dari perangkat lunak yakni mengenai sistem *login* yang telah berjalan dan bagaimana proses penyimpanan data autentikasi *user*. Untuk mendapatkan informasi tersebut dilakukan observasi tentang proses sistem *login* pada perangkat lunak dengan melakukan percobaan *login user* dan melakukan *testing* menggunakan *tools sniffing* Adapun *tools* yang digunakan yaitu *wireshark,* Selain itu mengumpulkan informasi bagaimana data *login user* disimpan ke *database*. Dari sini didapatkan bahwa pada sistem berjalan, data *login* yaitu *username* dan *password* berhasil di dapatkan dengan *tools sniffing* dan data *login user* yang tersimpan pada *database* masih berupa *plaintext* atau teks aslinya. Maka dari itu dibutuhkan sistem yang dapat mengamankan data - data tersebut.

2.2. Membuat Prototyping

Setelah melakukan pengumpulan informasi terkait perangkat lunak maka dilanjutkan pada pembuatan *prototyping* yaitu dengan membuat desain rancangan dari sistem yang akan di buat. Kemudian dilanjutkan dengan pembuatan *prototyping* berdasarkan desain yang telah dibuat yaitu dengan melakukan enkripsi untuk *username* dan *password* yang digunakan untuk melakukan *login* ke perangkat lunak dan melakukan enkripsi pada data *user* sebelum disimpan ke dalam *database*. Enkripsi yang digunakan yaitu Algoritma Rijndael yang mana merupakan algoritma terpilih untuk kriptografi *Advanced Encryption Standard* (AES) dengan ukuran kunci 128 bit dan menerapkan mode operasi *Chiper Block Chaining* (CBC).

Setelah *prototype* dibuat maka masuk pada pada tahapan evaluasi dari *prototyping* tersebut apakah sudah sesuai dengan kebutuhan pada tahapan ke 1 atau tidak, Jika sudah maka akan lanjut ke tahapan berikutnya namun jika belum sesuai maka diperlukan *review* dan *update prototype* untuk memperbaikinya agar dapat sesuai dengan kebutuhan pada tahapan 1. Dari *Prototype* yang telah di buat hasilnya telah sesuai

dengan kebutuhan pada tahapan 1. Maka dari itu dilanjut pada fase *development* di sistem yang sesungguhnya.

2.3. Fase *Development*

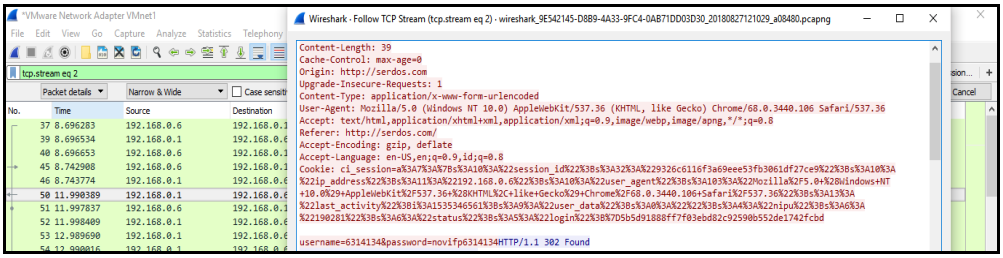
Setelah prototyping dibuat maka dilanjutkan pada fase development di sistem sesungguhnya. Pada fase ini terdiri dari beberapa tahapan sebagai berikut :

1. Desain
   Tahap ini bertujuan untuk menggambarkan bagaimana sistem nantinya akan di implementasikan. Desain dibuat berdasarkan hasil dari *prototype* yang telah di buat sebelumnya. Pada penelitian ini penulis membuat skenario desain penelitian pengiriman data pada sistem *login* yang dapat dilihat pada gambar 2 serta membuat skenario desain penelitian enkripsi penyimpanan data *user* yang dapat dilihat pada gambar 3.
2. Implementasi
   Pada tahapan ini penulis melakukan implementasi sesuai dari desain yang telah di buat dengan melakukan pengkodean. Adapun pengkodean yang digunakan yaitu dengan menggunakan bahasa pemograman PHP serta *Javascript* dan untuk *database* menggunakan MySQL.
3. Test
   Pada tahapan ini penulis melakukan proses pengujian terhadap sistem yang telah dibuat. Pada proses ini hal yang di uji yakni dari logika pengkodean dan fungsionalitas sistem yang bertujuan untuk memastikan bahwa sistem yang dibuat telah sesuai dengan kebutuhan yang diinginkan di awal.
4. Maintenance
   Tahapan ini dilakukan ketika sistem sudah berjalan yaitu kegiatan untuk melakukan pemeliharaan dan perawatan dari sistem yang telah dibuat.

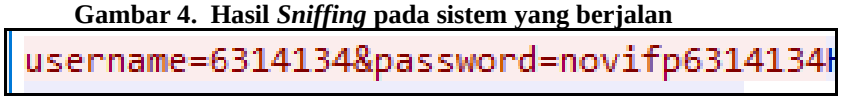
3. HASIL DAN PEMBAHASAN

3.1. Hasil dan Pembahasan Tahapan Pengumpulan Kebutuhan

Pengumpulan informasi untuk menentukan kebutuhan dari sistem yang akan dibuat dilakukan dengan melakukan observasi pada sistem login yang sudah berjalan dengan melakukan percobaan *login user* dan melakukan *testing* menggunakan *tools sniffing*. Adapun hasil dari percobaan tersebut ditampilkan pada gambar 4 dan gambar 5.



Gambar 4. Hasil *Sniffing* pada sistem yang berjalan



Gambar 5. Data *User* Hasil *Sniffing* dengan wireshark

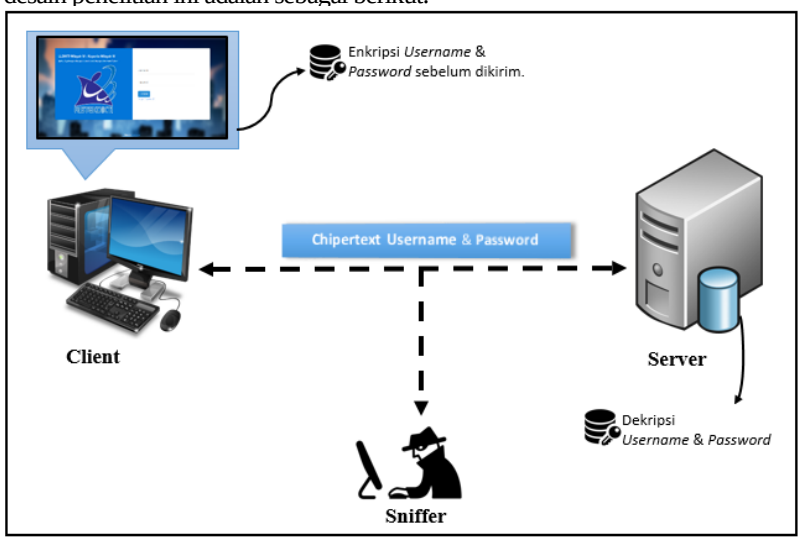
Dari percobaan tersebut dapat dilihat bahwa dengan menggunakan *tools sniffing wireshark* data autentikasi user yakni *usename* dan *password* dapat dengan mudah di dapatkan. Percobaan kedua yaitu dengan melihat data user pada *database* ternyata data yang disimpan masih berupa *plaintext* atau data terang.

3.2. Hasil dan Pembahasan Tahapan Desain

3.2.1. Desain Skenario

Penelitian ini berkaitan dengan proses enkripsi dan dekripsi data menggunakan AES 128bit dengan mode CBC. Untuk mempermudah pemahaman penelitian yang akan dilakukan, maka desain penelitian yang akan dibuat akan dijelaskan pada gambar 6 dan gambar 7. Adapun untuk mekanisme atau cara kerja dari AES 128bit dengan mode CBC akan dijelaskan pada sub bab 3.2.2.

Skenario desain penelitian ini adalah sebagai berikut:

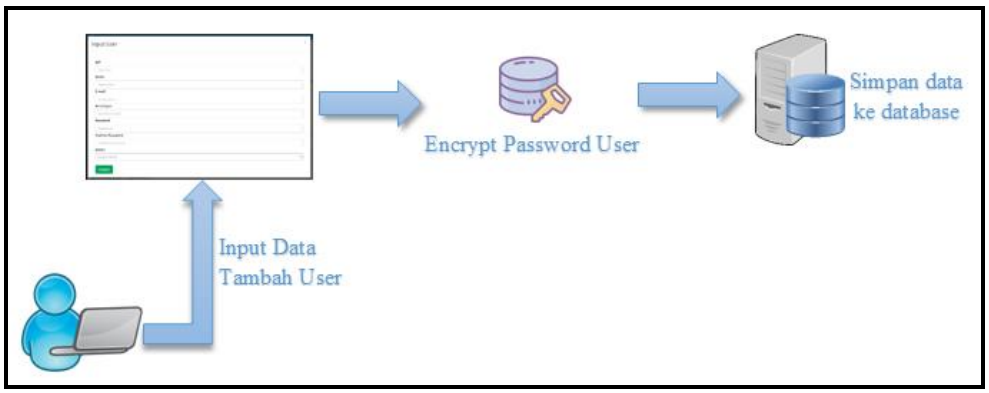


Gambar 6. Skenario Desain Penelitian Pengiriman Data pada Sistem Login

Penjelasan dari gambar diatas adalah sebagai berikut:

1. Untuk dapat masuk ke perangkat lunak *User* sebagai *client* melakukan login dengan memasukan *username* dan *password* dan klik tombol "Login".
2. Sebelum *username* dan *password* dikirimkan, Pada *view* login u*sername* dan *password* di enkripsi menggunakan *javascript* kriptografi. Enkripsi yang digunakan yaitu AES 128bit.
3. Setelah dilakukan enkripsi maka data dikirimkan ke server, Hasil enkripsi berupa sebuah objek yang dapat diubah ke notasi JSON yang didalamnya mengandung *variable – variable* yaitu *initialization vector*, *chipper text*, dan *salt*. Setelah itu pada sisi server *username* dan *password* di dekripsi dengan algoritma yang sama.
4. Hasil dari proses dekripsi kemudian diolah dan di bandingkan dengan data yang disimpan pada database server. Apabila kedua data tersebut sesuai maka akan masuk ke halaman utama perangkat lunak, sedangkan jika tidak sesuai maka perangkat lunak akan menampilkan pesan error.

Pada proses yang telah diuraikan diatas maka ketika ada serangan dari *sniffer* untuk mengambil data saat data ditransmisikan maka data yang didapat adalah data yang telah terenkripsi. Adapun skenario penelitian berikutnya ditampilkan pada gambar 7.



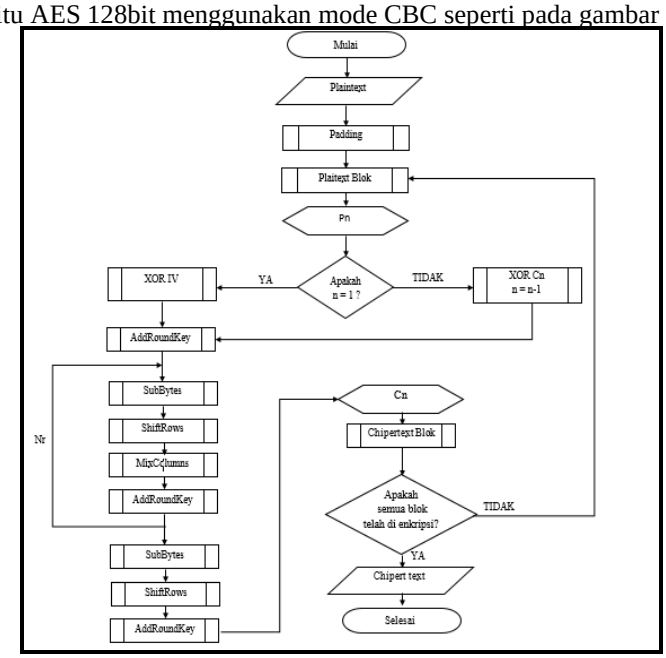
Gambar 7. Skenario Desain Penelitian Enkripsi Penyimpanan Data *User*

Penjelasan dari gambar diatas adalah sebagai berikut:

1. User mengakses halaman untuk tambah data *user* baru.
2. Kemudian isi data – data *user* pada *form input* yang disediakan.
3. Sebelum data dikirim untuk disimpan di *database*, data password dilakukan enkripsi terlebih dahulu dengan menggunakan Algoritma Rijndael AES 128bit mode CBC.
4. Setelah dilakukan enkripsi maka data user baru disimpan pada database.

3.3. Flowchart

Flowchart adalah bagan yang terdiri dari simbol – simbol yang digunakan untuk menggambarkan langkah – langkah arus penyelesaian suatu masalah seperti untuk menggambarkan penyajian dari algoritma. Adapun mekanisme enkripsi dan dekripsi dengan algoritma Rijndael yaitu AES 128bit menggunakan mode CBC seperti pada gambar 8.



Gambar 8. *Flowchart* Enkripsi AES 128bit mode CBC

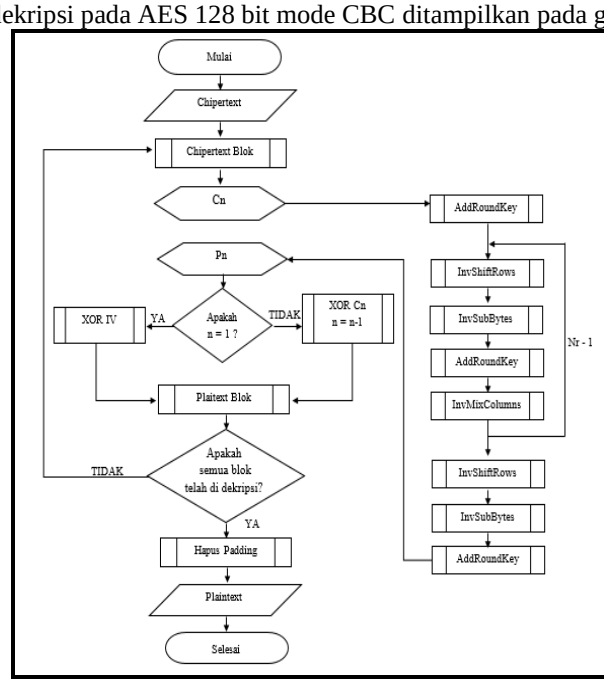
Berdasarkan gambar 8, akan dijelaskan langkah – langkah enkrpisi AES 128 bit Mode CBC adalah sebagai berikut:

1. Pertama, sistem akan mengambil data plaintext yaitu daam kasus ini adalah *username* atau *password*.

2. Plaintext disusun menjadi blok – blok data berkukuran sama. AES yang digunakan memiliki ukuran blok 128bit. Pada mode CBC data harus ada pada blok yang ukurannya sama, maka perlu dilakukan proses padding byte yang berfungsi untuk pengganjal yaitu menggenapi data supaya data pas dengan ukuran blok. *Padding* dilakukan dengan mengisi *byte* bernilai N bila dibutuhkan *padding* sebanyak N *byte*. Contoh dibutuhkan *padding* 4byte maka *padding* berisi "04 04 04 04". [3] Setelah itu *plaintext* sudah menjadi blok - blok data.
3. Jika Blok *plaintext* pertama maka dilakukan proses XOR dengan *initialization vector* atau IV sedangkan blok selanjutnya dilakukan XOR dengan hasil *chipertext* dari blok sebelumnya.
4. Kemudian dilakukan enkripsi AES yaitu pada proses:
   - *AddRoundKey*: Pada tahap awal ini disebut juga dengan *Initial Round* yaitu Melakukan XOR antara *state* awal (Pn) dengan *chiper key.* [4]
   - Putaran sebanyak Nr. Proses yang dilakukan setiap putaran adalah:
     a. *SubBytes* : Subtitusi byte dengan table S-Box
     b. *Shift Rows* : Baris – baris *array state* dilakukan pergeseran dengan cara *wrapping*.
     c. *MixColumn* : Pengacakan data dengan mengalikan setiap elemen dari blok chipper dengan matriks.
     d. *AddRoundKey* : XOR *current state* dengan *round key.*

   Pada AES 128bit proses diatas dilakukan Putaran Nr sebanyak 9 kali (9 *round)* yaitu pada round ke 10 dilakukan *SubBytes, ShiftRows, AddRoundKey.* Dari sini menghasilkan chipertext blok ke n.
5. Selanjutnya dilakukan pengecekan apakah semua blok telah di enkripsi jika belum maka lakukan proses no 4. Sedangkan jika semua blok telah selesai maka telah mendapatkan *chipertext* hasil enkripsi AES 128bit dengan mode CBC.

Sedangkan proses dekripsi pada AES 128 bit mode CBC ditampilkan pada gambar 9.



Gambar 9. *Flowchart* Dekripsi AES 128bit mode CBC

Berdasarkan gambar diatas, akan dijelaskan langkah – langkah dekripsi AES 128bit Mode CBC sebagai berikut:

1. Pertama, sistem akan mengambil data chipertext yaitu dalam kasus ini adalah *username* atau *password* serta mengambil key.
2. Chipertext dirubah menjadi blok – blok *chipertext.*
3. Setiap blok chipertext (Cn) dilakukan dekripsi AES 128 Mode CBC yaitu:
   - *AddRoundKey* yaitu XOR antara chipertext dengan chiper key. (*Initial Round)*
   - Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
     a. *InvShiftRow* : baris – baris array *state* digeser secara *wrapping.*
     b. *InvSubByte* : Substitusi byte dengan substitusi kebalikan (inverse S-box).
     c. *AddRoundKey* : XOR *state* dengan *round key.*
     d. *InvMixColumn* : Pekalian elemen dari blok chiper dengan matriks.
   - *Final Round* putaran ke-10, proses untuk putaran terakhir ini adalah: *InvShifRow, InvByteSub, AddRoundKey*.
4. Dari proses dekripsi diatas menghasilkan plaintext ke n (Pn). Jika itu merupakan n = 1 maka lakukan XOR dengan *Initialixzation Vector* (IV). Sedangkan Pn selanjutnya di XOR kann dengan Blok Chipertext sebelumnya.
5. Setelah dilakukan XOR akan membentuk *Plaintext* Blok, dan di cek apakah semua blok sudah di dekripsi jika belummaka kembali ke proses No. 3 dan 4. Sedangkan jika semua blok sudah di dekripsi maka akan dilakukan pengahpusan *padding*. Maka proses dekripsi telah selesai dilakukan dan menghasilkan *Plaintext*.

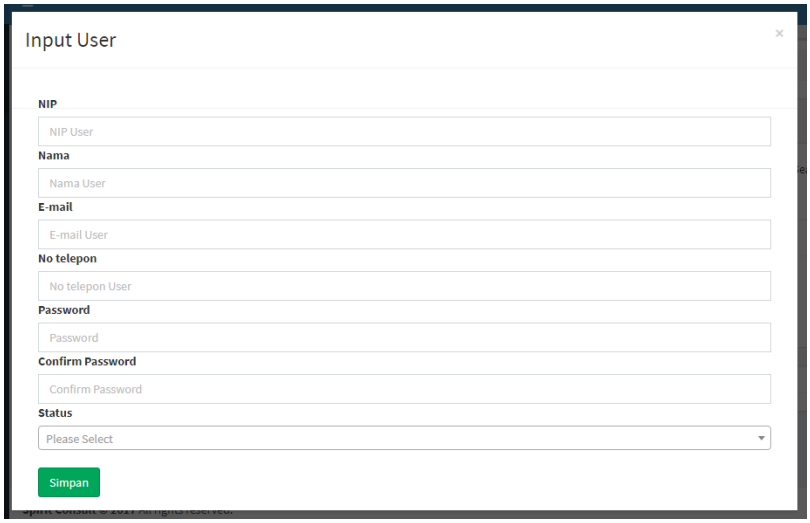
3.3. Hasil dan Pembahasan Tahapan Implementasi

3.3.1. Implementasi Antarmuka

Dalam pembuatan perangkat lunak, antarmuka memegang peranan penting. Antarmuka dapat memudahkan User dalam mengoperasikan perangkat lunak yang telah dibuat. Berikut antarmuka perangkat lunak yang berhubugan dengan implementasi Algoritma Rijndael (AES 128bit mode CBC) beserta uraian cara penggunaan ditampilkan pada gambar 10.

Gambar 10. Antarmuka *Login*

Halaman ini akan muncul saat pertama kali perangkat lunak dijalankan. Untuk melakukan login User memasukan *username* dan *password* kemudian menekan tombol login, Jika login valid, maka halaman utama akan muncul sesuai dengan hak akses masing-masing jika gagal maka akan menampilkan pesan *error*. Halaman tambah user ditampilkan pada gambar 11.



Gambar 11. Antarmuka Tambah Data *User*

Halaman ini akan muncul saat user mengklik tombol tambah data setelah itu User mengisi data User Lalu klik Tombol simpan untuk menyimpan data user.

3.4. Hasil dan Pembahasan Tahapan Pengujian (Test)

Pada penelitian ini dilakukan pengujian dengan cara eksperimen sesuai dengan skenario eksperimen yang telah dibuat dan dengan variabel eksperimen yang telah ditentukan.

3.4.1. Variabel Eksperimen

Variabel yang digunakan dalam penelitian ini adalah :

1. *Plaintext Username* dan *Password User*.
2. *Chippertext Username* dan *Password*.
3. *Library Javascript* AES Rijndael 128bit sebagai metode keamanan yg digunakan.
4. Tools *Sniffing* Wireshark

3.4.2. Kebutuhan Sumber Daya

Untuk mendukung penelitian ini maka maka *hardware* dan *software* yang digunakan adalah sebagai berikut:

Perangkat Keras (*Hardware*)

a. Prosessor : Intel (R) *Core* (TM) i3 - 2310M CPU @ 2,40 GHz
b. Memori : 2 GB RAM
c. Harddisk Drive : free space (± 4 GB)
d. Monitor
e. Mouse & Keyboard

Perangkat Lunak (*Software*)

a. Sistem operasi:
- Windows 10 (client)
- Linux Debian 9.5.0 (server)

b. Service dan Database (Apache, MySql)
c. Web Browser
d. VMware
e. Tools Sniffing Wireshark

f. Programming Language: PHP, Javascript

3.5. Populasi dan Sampel

Populasi dari penelitian ini adalah data akun login *user* yaitu. Pada akun login *user* terdapat 3 (tiga) hak akses yaitu sebagai Admin, Pemroses dan Pemegang Wilayah. Maka dari itu penulis mengambil 5 data sampel dengan hak akses berbeda – beda. Selain itu penulis juga melakukan *sampling* penambahan data akun *user* dengan 3 panjang karakter *password* yang berbeda beda. Adapun sample yang digunakan ditampilkan pada tabel 1 dan tabel 2.

Tabel 1. *Item Sample* akun login *user*

No	*NIP/Username*	*Password*	*Status*
1	6314134	novifp6314134	Admin
2	9965124257	lpkiajaya1984	Admin
3	104008	fibrianiPray	Pemroses
4	209012	tunjangan123	Pemegang wilayah
5	190281	LPKIAjaya	Pemegang wilayah

Tabel 2. *Item Sample* tambah akun *user*

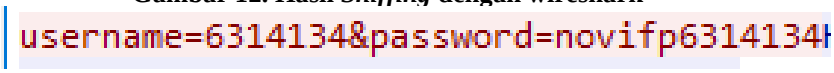
No	*Data Password*	*Panjang karakter*
1	novi6314134	11 karakter
2	kopertiswilayah1	16 karakter
3	KopertisWilayahIV	17 karakter

3.5.1. Hasil Pengujian

Berikut adalah hasil pengujian yang dilakukan sesuai skenario eksperimen yang mengacu kepada prosedur eksperimen, variabel eksperimen serta *sample* yang telah ditentukan yaitu data akun *user* pada perangkat lunak. Adapun hasil pengujian dari penelitian ini adalah sebagai berikut:

1. Pengujian dengan melakukan *sniffing* menggunakan *tools sniffing* wireshark pada form login yang tidak menggunakan enkripsi. Dari 5 (lima) data *sample* diambil dan dilakukan pengujian keseuluruhan data dapat dibaca oleh *tools sniffing* wireshark. Hasil dari proses *sniffing* dengan menggunakan wireshark adalah seperti gambar 12 dan gambar 13.

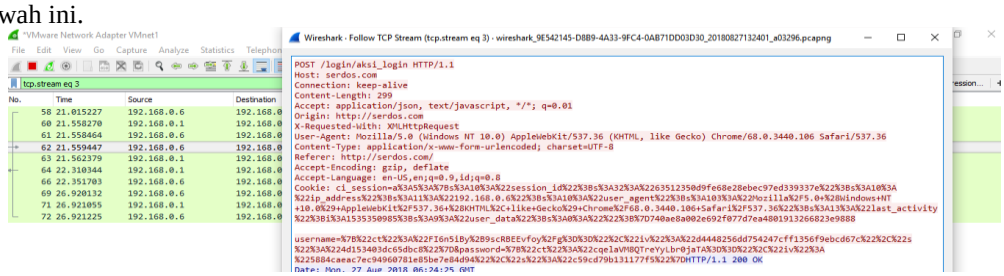
Gambar 12. Hasil *Sniffing* dengan wireshark



Gambar 13. Data *User* Hasil Sniffing dengan wireshark

2. Melakukan *sniffing* menggunakan *tools sniffing* wireshark pada form *login* yang menggunakan enkripsi. Dari data *sample* yang telah diambil, *username* dan *password*

user telah berhasil dienkripsi, Hal itu dibuktikan seperti gambar 14 dan gambar 15 dibawah ini.



Gambar 14. Hasil Sniffing dengan wireshark

```
username=%7B%22ct%22%3A%22FI6n5iBy%2B9scRBEEvfoy%2Fg%3D%3D%22%2C%22iv%22%3A%22d4448256dd754247cff1356f9ebcd67c%22%2C%22s
%22%3A%224d153403dc65dbc8%22%7D&password=%7B%22ct%22%3A%22cqelaVM8QTreYyLbr0jaTA%3D%3D%22%2C%22iv%22%3A
%225884caeac7ec94960781e85be7e84d94%22%2C%22s%22%3A%22c59cd79b131177f5%22%7DHTTP/1.1 200 OK
```

Gambar 15. Data Hasil Sniffing yang telah terenkripsi

3. Pada pengujian ini dilakukan proses login beberapa kali dengan akun login yang sama untuk melihat enkripsi yang dihasilkan adapun data login yang digunakan adalah:
   a. Username : 6314134
   b. Password : novifp6314134
   c. Status : Admin

   Adapun hasil pengujian yang telah dilakukan adalah sebagai tabel 3:

Tabel 3. Pengujian Hasil Enkripsi

No	*Hasil*
1	username=%7B%22ct%22%3A%22pS64m6A%2Fn%2BAszmczD0XSSw%3D%3D%22%2C%22iv%22%3A%229bb05aff765743421eb7c6d905b0e920%22%2C%22s%22%3A%224b4b57aa687a5c63%22%7D&password=%7B%22ct%22%3A%22qD7r1fjSwIehQNbJK%2Bmnyg%3D%3D%22%2C%22iv%22%3A%22f67645a7cd64977e14b00abc54bbbfa6%22%2C%22s%22%3A%22246cbf495533979f%22%7DHTTP/1.1 200 OK
2	username=%7B%22ct%22%3A%22U%2FXRfqtSCUshpWfuAv2QZA%3D%3D%22%2C%22iv%22%3A%2284cb877a066f2eaf28c4785df0bd6bc0%22%2C%22s%22%3A%222b1a4cd8fa848e52%22%7D&password=%7B%22ct%22%3A%22%2F6M06ahkychpHl0JHmvMvA%3D%3D%22%2C%22iv%22%3A%22f29e494b326fb936f0ddc34b531733b0%22%2C%22s%22%3A%22a5952f7c59a69466%22%7DHTTP/1.1 200 OK
3	username=%7B%22ct%22%3A%22j5nNDVD2MDqaFFPg0OU0wQ%3D%3D%22%2C%22iv%22%3A%22156495c09b0457d3966640396de68a2c%22%2C%22s%22%3A%22a42100d1e1c4a4bd%22%7D&password=%7B%22ct%22%3A%222DdXMAhbAXaAI3mz%2FUJnwQ%3D%3D%22%2C%22iv%22%3A%22e30ed5d9a417a6f791993c27eba80f60%22%2C%22s%22%3A%2216f7cdf462284331%22%7DHTTP/1.1 200 OK
4	username=%7B%22ct%22%3A%22dZsWsSIgq1m9iBPo6PHwlw%3D%3D%22%2C%22iv%22%3A%229e001ebcc9ca6394dbaecb3b93b9e82c%22%2C%22s%22%3A%22da7be9d39aac436d%22%7D&password=%7B%22ct%22%3A%22E2GDZidk8HAO%2F%2FLpIcM1qA%3D%3D%22%2C%22iv%22%3A%220adff6f207b715b0354c030fd2e9d97f%22%2C%22s%22%3A%22d2cc38c1d35e80b0%22%7DHTTP/1.1 200 OK
5	username=%7B%22ct%22%3A%22%2BkRwjF6cDgIKKrVKtI%2FInA%3D%3D%22%2C%22iv%22%3A%22342ce16ee8cd96ac6034321ef5a427f5%22%2C%22s%22%3A%22be63891b3f9ad84a%22%7D&password=%7B%22ct%22%3A%22%2FquhiEnMEZIMBzmK0Zg8Kg%3D%3D%22%2C%22iv%22%3A%22a3567eaf9d8a2b85097758a2c66531ad%22%2C%22s%22%3A%2211c8d5a25d7dc556%22%7DHTTP/1.1 200 OK

4. Pengkodean kriptografi yang dilakukan untuk enkripsi data login menggunakan javascript yaitu dengan melakukan enkripsi pada view sebelum dikirim ke controller. Seperti gambar 16.

Gambar 16. Implementasi Javasccript Kriptografi pada Aplikasi

Pada penggunaan javascript kode program dapat mudah sekali dibaca, Pada kasus ini kunci pembangkit untuk melakukan enkripsi ditransmisikan pada saat proses enkripsi. Dari sini *cryptoanalys* dapat menggunakannya untuk mendapatkan data yang sebenarnya.

5. Pada pengujian ini dilakukan penambahan data user dimana sebelum data disimpan ke database *password* harus di enkripsi terlebih dahulu. Adapun tabel hasil pengujian ini adalah sebagai tabel 4.

6. Tabel 4. Pengujian Enkripsi Untuk Penyimpanan Ke Database

No	*Data Password*	*Hasil yang diharapkan*	*Keluaran*	*Hasil*
1	novi6314134 (11 karakter)	Data password terenkripsi	Password terenkripsi. 78bb02fd625164476b3878bc4e61bf25	[√] Berhasil [ ] Tidak Berhasil
2	kopertiswilayah1 (16 karakter)	Data password terenkripsi	Password terenkripsi. 5088128e49cbf3c5576cb5fcb2d1524b	[√] Berhasil [ ] Tidak Berhasil
3	KopertisWilayahIV (17 karakter)	Muncul peringatan “Maximum panjang password 16 karakter”	Muncul peringatan “Maximum panjang password 16 karakter”	[√] Berhasil [ ] Tidak Berhasil
4	Password yang dimasukan sama seperti sebelumnya : novi6314134	Data password terenkripsi	Password terenkripsi. 78bb02fd625164476b3878bc4e61bf25	[√] Berhasil [ ] Tidak Berhasil

4. KESIMPULAN

Berdasarkan implementasi dan pengujian yang telah dilakukan, maka diperoleh kesimpulan bahwa penelitian mengenai Implementasi Algoritma Rijndael untuk keamanan sistem *login* adalah:

1. Pada format *login* standar, proses *sniffing* mampu mendapatkan *username* dan *password user*.
2. Penerapan algoritma kriptografi AES Rijndael pada perangkat lunak terbukti dapat mengamankan sistem *login* dikarenakan saat proses *sniffing username* dan *password user* sudah terenkripsi.
3. Penggunaan algoritma AES dengan mode CBC menghasilkan *chipertext* yang berbeda – beda meskipun dengan plaintext dan kunci pembangkit yang sama.

4. Pemanfaatan kriptografi dengan javascript mempunyai kelmahan yaitu pada transmisi kunci.

DAFTAR PUSTAKA

[1] M. Keuangan, "Menteri keuangan republik indonesia," vol. 2004, 2008.
[2] B. Rahardjo, *Keamanan Sistem Informasi Berbasis Internet*. 1999.
[3] A. Victor and T. M. Putra, "PENERAPAN V.S.N HARDWARE KEY SCHEME DENGAN RSA CRYPTOSYSTEM UNTUK PENGAMANAN PERANGKAT LUNAK," *J. Rekayasa Sist. Ind.*, 2015.
[4] D. M. Khairina, "ANALISIS KEAMANAN SISTEM LOGIN," *J. Inform. Mulawarman*, 2011.
[5] R. Hikmah *et al.*, "Implementasi Enkripsi AES Pada Pembangunan Aplikasi Accounting Pada PT PRO Sistematika Automasi," pp. 2–5, 2011.
[6] S. Kromodimoeljo, *Teori dan Aplikasi Kriptograf*. 2009.
[7] T. Zebua, "Pengamanan Data Teks Dengan Kombinasi Cipher Block Chaining dan LSB-1," in *Seminar Nasional Inovasi dan Teknologi (SNITI)*, 2015.
[8] R. Susanto and A. D. Andriana, "PERBANDINGAN MODEL WATERFALL DAN PROTOTYPING UNTUK PENGEMBANGAN SISTEM INFORMASI," *Maj. Ilm. UNIKOM*, 2016.