SISTEM KEAMANAN DATABASE MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD(AES-128) STUDI KASUS : RED AVENUE INDONESIA

Faris Akbar[1)], **Sejati Waluyo**[2)]
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : 1311502288@student.budiluhur.ac.id[1)], *sejati.waluyo@budiluhur.ac.id*[2)]

ABSTRAK

Red Avenue Indonesia adalah perusahaan yang berdiri sejak tahun 2003 dan bergerak dalam bidang event organizer & experiential learning, yang mempunyai client dari Indonesia dan bahkan sampai dengan Asia. Dengan banyaknya instansi, organisasi, maupun kelompok masyarakan yang mempercayai menggunakan jasa dari red avenue Indonesia untuk keperluan outbound atau membentuk jiwa kepemimpinan untuk para karyawan, membuat client-client tersebut percaya memberikan data informasi pribadi maupun informasi data perusahaannya tersebut kepada perusahaan Red Avenue Indonesia. Dengan banyaknya data-data penting client tersebut, tidak sedikit juga ada kelompok maupun individual yang ingin mencuri data-data client tersebut dari perusahaan red avenue Indonesia untuk kepentingan atau keuntungan sendiri maupun kelompoknya. Untuk mencegah hal tersebut, maka dibuatlah suatu aplikasi penyimpanan dan pengamanan database berbasis desktop menggunakan algoritma Advanced Encryption Standard 128 (AES 128). AES 128 merupakan algoritma simetris blok yang cukup aman karena memiliki 10 putaran dalam proses enkripsi dan dekripsinya. Perancangan aplikasi ini menggunakan metodologi pengembang waterfall dari Winston W. Royce, 1970 agar mempermudah dalam proses penelitian. Bahasa pemograman yang digunakan adalah Java NetBeans IDE 8.2. Saat data client diinput kedalam database, data sudah otomatis terenkripsi dengan AES 128, ketika user ingin melihat data kembali, user harus mendekripsi terlebih dahulu dengan mengisi password dekripsi tersebut dengan benar, user dapat mengedit data dengan mendekripsi data terlebih dahulu, dan dapat menghapus data yang diinginkan dengan memilih record pada tabel. Dengan menggunakan aplikasi ini, perusahaan red avenue indonesia dapat menyimpan data client ke dalam database tanpa adanya rasa takut atau khawatir orang lain mencuri atau membaca isi dari data client tersebut.

Kata kunci : Kriptografi, AES 128, Enkripsi, Dekripsi, Database

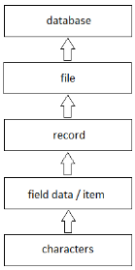
1. PENDAHULUAN

Seiring berkembangnya teknologi dan telekomunikasi yang semakin pesat, kemampuan untuk mengakses dan menyediakan informasi data secara cepat dan akurat menjadi sangat penting bagi sebuah organisasi ataupun instansi. Sangat pentingnya sebuah data menjadikan data yang diinginkan hanya boleh diakses oleh orang-orang yang dipercaya saja. Red Avenue Indonesia adalah perusahaan yang berdiri sejak tahun 2003 dan bergerak dalam bidang jasa *event organizer* dan *experiential learning*. Di perusahaan tersebut terdapat data-data penting dari berbagai client yang mempercayai red avenue Indonesia untuk menggunakan jasanya, yaitu informasi data pribadi ataupun informasi data instansi client red avenue indonesia, dan juga jadwal kegiatan. Di perusahaan tersebut masih belum memiliki aplikasi penyimpanan database dan pengamanan database tersebut, perusahaan tersebut memakai nota atau kwitansi untuk transaksi pembayaran serta menyimpan data *client* tersebut didalam suatu *folder* tanpa ada suatu pengamanan data. Aplikasi ini dibuat untuk dapat menyimpan dan mengamankan data *client* dalam penyimpanan database di Red Avenue Indonesia untuk menghindari pencurian data dari pihak-pihak yang akan merugikan perusahaan tersebut. Didalam aplikasi yang dibuat, data akan otomatis terenkripsi saat *user* meng-input data kedalam *database* dan apabila data dipanggil kembali data harus didekripsi terlebih dahulu. Algoritma yang dipakai pada aplikasi pengamanan data atau informasi adalah algoritma AES-128, dan data yang akan di amankan yaitu hanya data yang berupa karakter atau *plaintext* didalam *database*. Dan aplikasi hanya dapat mengenkripsi dan mendekripsi *database* per-*record*. Dalam pengembangan aplikasi ini penulis menggunakan metode *waterfall* oleh Winston W. Royce tahun 1970, agar mempermudah dalam proses penelitian yang penulis lakukan.

2. LANDASAN TEORI

2.1. Database

Database yaitu kumpulan data yang berhubungan satu dengan lainnya, dan tersimpan di luar komputer dan menggunakan perangkat lunak tertentu untuk memanipulasinya. *Database* merupakan komponen yang penting didalam sistem, karena berfungsi sebagai penyedia informasi bagi para penggunanya. Penerapan yang didalam sistem disebut dengan *database system*[1].



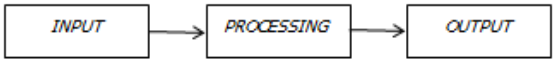
Gambar 1 : Jenjang Data

Jenjang data :

1) *Characters,* adalah bagian data yang terkecil, bisa berupa karakter numeric, huruf atau karakter khusus (*special characters*) yang membentuk suatu data.
2) *Field,* menampilkan suatu atribut dari *record* yang menunjukan item dari data, seperti nama perusahaan, alamat, telepon. Kumpulan dari *field* membentuk menjadi sebuah *record.*
3) *Record,* kumpulan dari *field* yang membentuk suatu *record*. Sebuah *Record* menunjukan suatu unit data individu tertentu.
4) *File,* adalah *record-record* yang menggambarkan kesatuan data yang sejenis. Misalnya *file client* salah satu perusahaan berisi data tentang semua client perusahaan tersebut.
5) *Database,* adalah kumpulan dari *file* atau sebuah tabel membentuk suatu data yang disebut *database*[2].

2.2. Sistem Pengolahan Data Dalam *Database*

Proses pengolahan data dalam database terdiri dari tiga tahap dasar yang disebut *data processing cycle* atau siklus pengolahan data. Yaitu *input, processing*, dan *output* sebagai gambar berikut[3]:



Gambar 2 : Data Processing Cycle

2.3. Definisi Kriptografi

Kriptografi yaitu berasal dari bahasa yunani, *crypto* dan *graphia*. *Crypto* artinya *secret* (rahasia) dan *graphia* artinya writing (tulisan). Jadi, kriptografi dapat diartikan menjadi tulisan rahasia. Kriptografi adalah ilmu tentang teknik enkripsi yang dimana data diacak dengan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh orang yang tidak mempunyai kunci dekripsi. Kriptografi merupakan teknik pengamanan informasi dilakukan dengan mengolah informasi awal (*plain text*) dengan suatu kunci dan menggunakan metode enkripsi tertentu sehingga dapat menghasilkan informasi baru (*ciphertext*) yang tidak dapat dibaca. Dan *ciphertext* tersebut dapat dikembalikan menjadi *plaintext* kembali dengan cara proses dekripsi. Kriptografi merupakan bidang ilmu yang bisa menjadi solusi dari masalah keamanan data.

2.4. Tujuan Kriptografi

Tujuan kriptografi adalah memberikan keamanan. Dan yang dinamakan aspek keamanan yaitu :

1) *Confidentially* (Kerahasiaan)
   adalah layanan untuk menjaga pesan supaya tidak dapat dibaca oleh orang yang tidak berhak.
2) *Data Integrity* (Integritas data)
   yaitu layanan yang menjamin pesan masih asli atau tidak pernah dimanipulasi selama pengiriman.
3) *Authentication* (Autentifikasi)
   yaitu layanan yang berhubungan dengan identifikasi, mengidentifikasi kebenaran pihak yang berkomunikasi (*user authentication*).
4) *Non-repudiation* (Penyangkalan)
   Adalah layanan untuk menjaga entitas yang berkomunikasi melakukan penyangkalan[4].

2.5. Algoritma Kriptografi

Algoritma kriptografi adalah sekumpulan aturan untuk proses enkripsi dan dekripsi. Di beberapa metode kriptografi, ada perbedaan antara fungsi enkripsi maupun fungsi dekripsi. Konsep matematis yang mendasari algoritma relasi antara himpunan yang berisi elemen *ciphertext.* Enkripsi dan dekripsi adalah fungsi yang memetakan elemen-elemen diantara kedua himpunan tersebut. Misalnya himpunan *plaintext* diibaratkan P dan himpunan elemen *ciphertext* diibaratkan C, maka fungsi E memetakan dari himpunan P ke himpunan C.

$$E(P) = C$$

Dan fungsi dekripsi memetakan dari himpunan C ke himpunan P

$$D(C) = p$$

Karena fungsi dekripsi D mengembalikan himpunan C menjadi himpunan P awal semula, jadi algoritma kriptografi harus memenuhi persamaan

$$D(E(P)) = P$$

Tingkat keamanan algoritma kriptografi sering diukur dari kuantitas proses yang dilakukan didalam suatu fungsi, baik fungsi enkripsi atau fungsi dekripsi. Proses itu dapat dihubungkan dengan sumber data yang diperlukan, menampilkan semakin

kuatnya algoritma kriptografi tersebut. Pada kriptografi klasik keamanan kriptografi ada pada kerahasiaan algoritma kriptografinya. Salah satu contohnya yaitu mesin enigma yang dibuat oleh pemerintah Jerman pada saat perang dunia ke-2.

Gambar 3 : Mesin Enigma

Dan inilah yang menjadi titik lemah pada saat algoritma bocor ke pihak yang tidak berwenang sehingga mengharuskan menyusun suatu algoritma baru agar tidak adanya rasa khawatir akan kebocoran informasi tersebut, karena informasi tersebut hanya dapat didekripsi, yaitu oleh pihak yang memang mempunyai kunci private tersebut [5]. Berikut ini adalah istilah yang digunakan dalam kriptografi :

a) *Plaintext* adalah pesan data asli yang akan dikirim.
b) *Ciphertext* artinya pesan yang telah terenkripsi, *ciphertext* tersebut ialah hasil dari enkripsi.
c) Enkripsi yaitu proses perubahan dari *plaintext* menjadi *ciphertext.*
d) Dekripsi yaitu mengubah *ciphertext* menjadi *plaintext,* sehingga menjadi data awal atau data asli.
e) Kunci adalah bilangan yang dirahasiakan, dan digunakan dalam proses enkripsi dan dekripsi.

2.6. Algoritma Simetris

Dimana kunci yang digunakan harus sama pada saat proses enkripsi dan saat dekripsi. Dalam kriptografi dengan kunci simetris dapat diasumsikan bahwa pengirim telah terlebih dahulu membagikan kunci kepada penerima sebelum pesan dikirimkan kepada ke penerima pesan. Keamanan dari sistem ini ada pada kerahasiaan kuncinya. Yang termasuk dalam kriptografi simetris berjalan dalam mode blok (*block cipher*), setiap kali proses enkripsi atau dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu *bit* atau satu *byte* data.

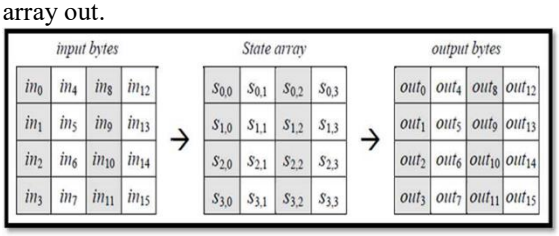
2.7. Algoritma AES-128 *(Advanced Encryption Standard - 128)*

Algoritma AES adalah algoritma kriptografi yang sifatnya simetri dan *cipher block.* Algoritma ini menggunakan kunci yang sama saat enkripsi dan dekripsi, masukan dan keluarannya berupa blok dan kunci yang akan digunakan. AES memiliki ukuran blok dan kunci yang tetap yaitu sebesar 128, 192, 256 bit. Berikut perbandingan jumlah proses Panjang kunci yang dilalui untuk masing-masing masukan bit[6].

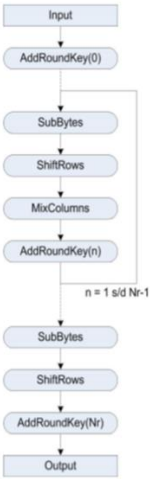
Tabel 1 : Perbandingan Panjang Kunci AES

Tipe	Jumlah key (Nk)	Besar Blok (Nb)	Jumlah Round (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Operasi AES dilakukan terhadap *array of byte* dua dimensi (state). State memiliki ukuran NROWS X NCOLS. Pada saat awal enkripsi, data yang berupa in0, in2, in4, in5, in6, in7, in8, in9, in10, in11, in12, in13, in14, in15 disalin kedalam array state. State ini yang nantinya dilakukan enkripsi/dekripsi. Kemudian keluarannya akan ditampung kedalam array out.



Gambar 4 : State Array, Input, dan Output



Gambar 5 : Ilustrasi Algoritma Enkripsi AES

Masing-masing tipe memakai kunci internal berbeda yaitu *round key* untuk setiap proses putaran. Proses putaran enkripsi AES-128 dilakukan sebanyak 10 putaran, yaitu sebagai berikut.

1. *Addroundkey,* XOR antara data awal (*plaintext*) dengan *cipher key.* Tahap ini disebut *Initial round.*
2. Putaran sebanyak 9 kali, proses yang dilakukan pada setiap putaran ini adalah:

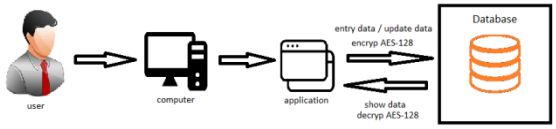
SubBytes, ShiftRows, MixColumns, dan *AddRoundKey.*

3. *Final round,* yaitu proses untuk putaran ke-10 yang meliputi *SubBytes, ShiftRows,* dan *AddRoundKey.*

3. RANCANGAN SISTEM DAN APLIKASI

3.1. Arsitektur Sistem

Gambar dibawah ini adalah rancangan arsitektur kerja sistem aplikasi yang akan dibuat.

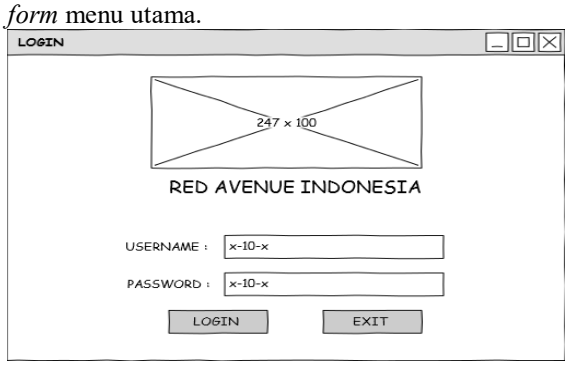


Gambar 6 : Arsitektur Kerja Aplikasi

3.2. Rancangan Layar

3.2.1. *Form* Menu Login

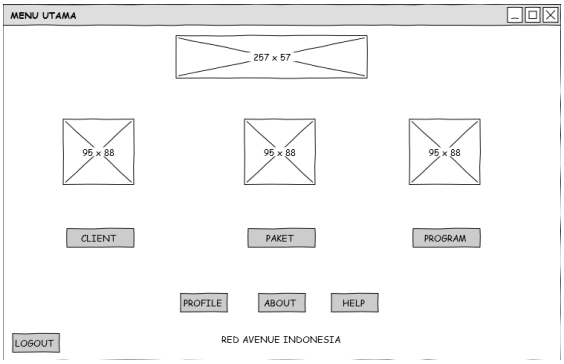
Form ini adalah tampilan awal saat aplikasi pertama kali dibuka, seperti gambar dibawah ini, berfungsi untuk akses menuju menu utama. *Form* ini menyediakan pengisian *username* dan *password*. Tombol *login* digunakan untuk proses validasi, jika *username* dan *password* benar maka akan tampil *form* menu utama.



Gambar 7 : Rancangan Layar Form Login

3.2.2. Rancangan Layar *Form* Menu Utama

Form menu utama akan tampil jika *user* memasukan *username* dan *password* dengan benar. Di *form* ini pengguna bisa melihat semua menu yang disediakan didalam aplikasi. *User* dapat memilih menu *client*, paket, program, *profile, about, help*, atau logout. Seperti gambar dibawah ini:



Gambar 8 : Rancangan Layar Form Menu Utama

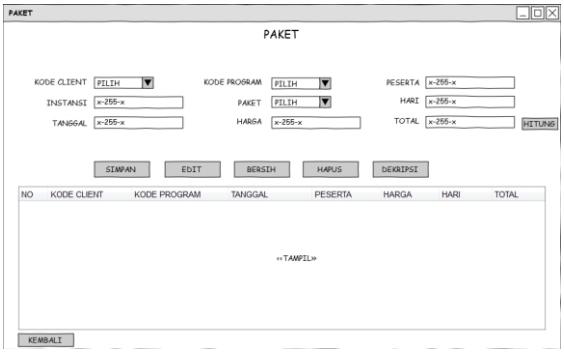
3.2.3. Rancangan Layar *Form Client*

Pada rancangan menu *Form Client,* berfungsi untuk melakukan *input,* simpan, *edit,* dan hapus data *client*. Seperti gambar berikut ini:

Gambar 9: Rancangan Layar Form Client

3.2.4. Rancangan Layar *Form* Paket

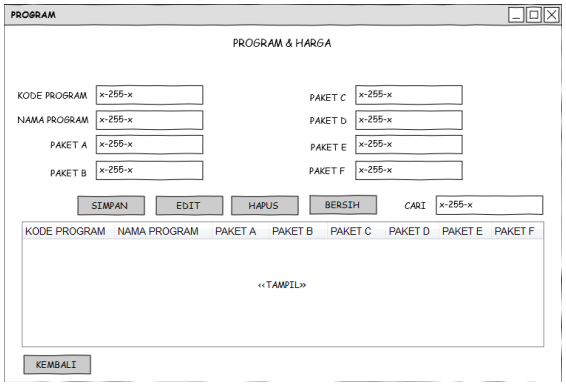
Pada rancangan menu *Form* Paket, berfungsi untuk melakukan *input,* simpan, *edit,* dan hapus program paket dan jadwal kegiatan yang disetujui oleh *client,* serta otomatis mengenkripsi data saat *user* menyimpan, dan terdapat tombol dekripsi untuk melihat data asli dari paket *client*. Seperti gambar berikut ini:



Gambar 10 :Rancangan Layar Form Paket

3.2.5. Rancangan Layar *Form* Program

Pada rancangan menu *Form* Program, berfungsi untuk melihat menyimpan, mengedit, serta menghapus program acara dan daftar harga yang ada didalam *database*. Seperti gambar berikut ini:

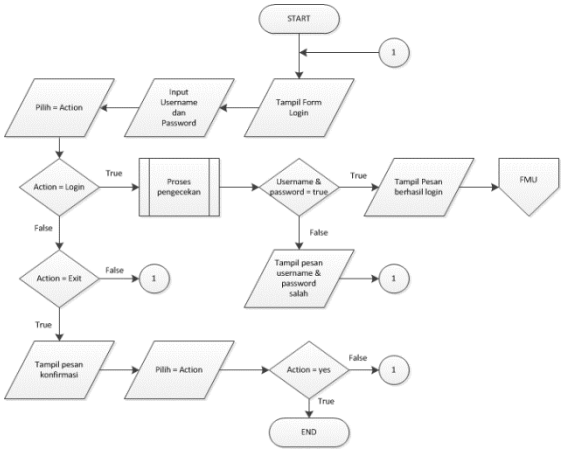


Gambar 11 : Rancangan Layar Form Program

3.3. Flowchart

3.3.1. Flowchart *Form Login*

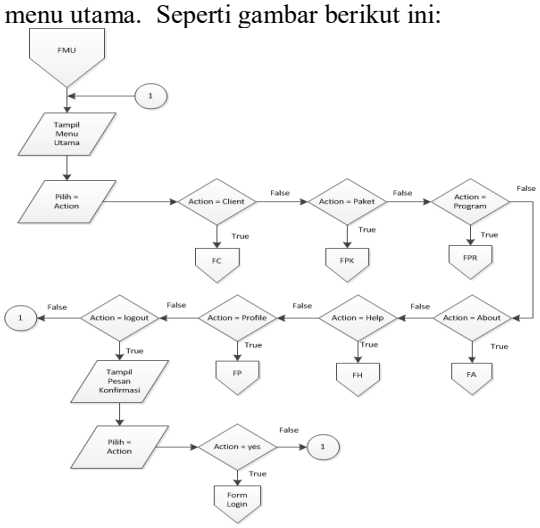
Flowchart ini menerangkan cara *user* untuk *login* agar dapat menggunakan aplikasi ini. *User* harus memasukan *username* dan *password* dengan benar. Seperti gambar berikut ini :



Gambar 12 : Flowchart Form Login

3.3.2. Flowchart *Form* Menu Utama

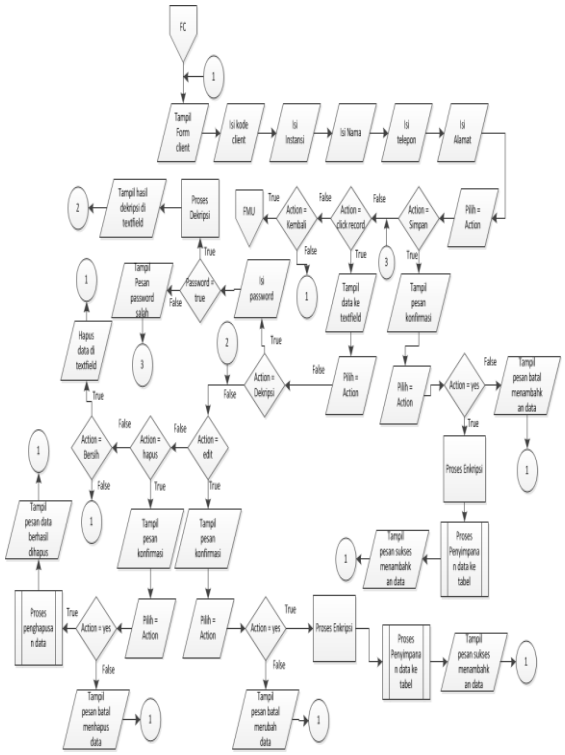
Flowchart ini menjelaskan alur proses penampilan menu utama. Seperti gambar berikut ini:



Gambar 13 : Flowchart Form Menu Utama

3.3.3 Flowchart *Form Client*

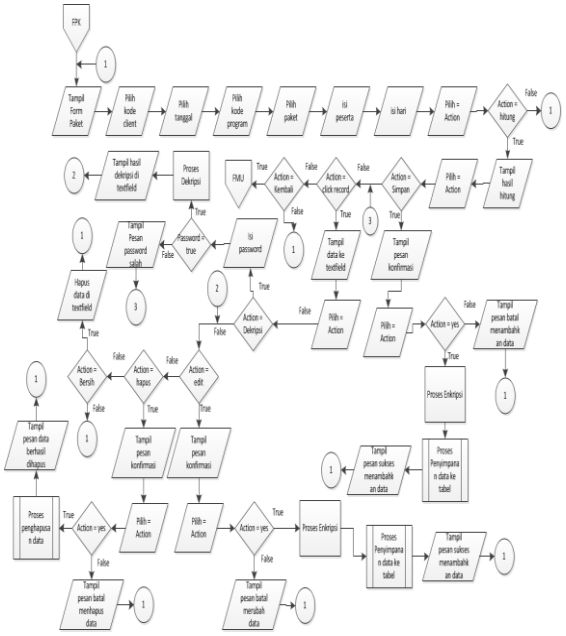
Flowchart form client merupakan gambaran alur proses dari *form client*. *User* dapat menyimpan, mengubah, dan menghapus data *client.* Jika *user* memilih tombol dekripsi, maka data yang dipilih oleh *user* akan terdekripsi. Seperti gambar berikut ini:



Gambar 14 : Flowchart Form Client

3.3.4. Flowchart *Form* Paket

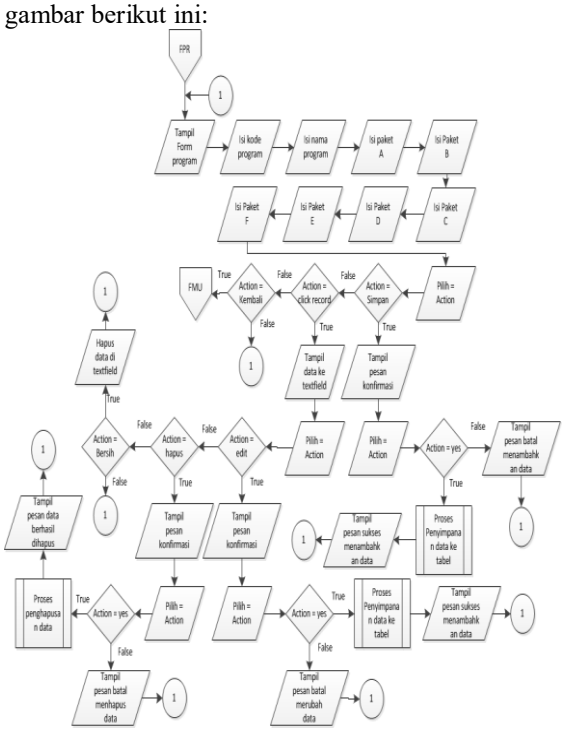
Flowchart form paket merupakan gambaran alur proses dari *form* paket. Pengguna dapat menyimpan, mengubah, dan menghapus data paket *client* yang sudah disepakati. Jika *user* memilih tombol dekripsi maka *record* yang dipilih oleh *user* akan terdekripsi. Seperti gambar berikut ini:



Gambar 15: Flowchart Form Paket

3.3.5 Flowchart *Form* Program

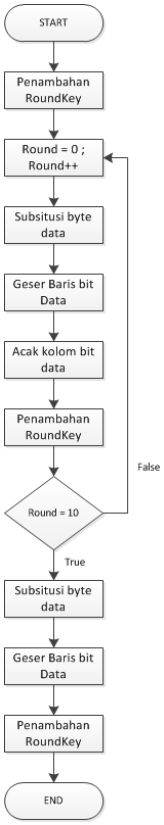
Flowchart form program merupakan gambaran alur proses dari *form* program. U*ser* dapat menyimpan, mengubah, dan menghapus data program. Seperti gambar berikut ini:



Gambar 16 : Flowchart Form Program

3.3.6 Flowchart Enkripsi AES 128

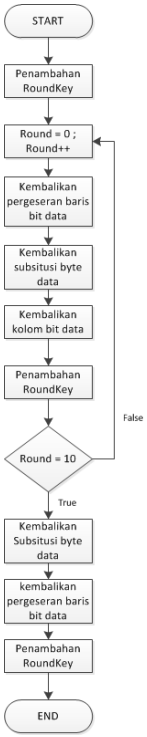
Gambar dibawah ini menjelaskan proses perubahan dari *plaintext* menjadi *ciphertext* menggunakan *algoritma AES 128. Seperti gambar berikut ini:*



Gambar 17 : Flowchart Enkripsi AES 128

3.3.7 Flowchart Dekripsi AES 128

Flowchart dibawah ini menjelaskan proses mengembalikan dari *ciphertext* menjadi *plaintext* menggunakan algoritma *AES 128.* Seperti gambar berikut ini:



Gambar 18 : Flowchart Dekripsi AES 128

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Layar *Form Login*

Form login pada gambar dibawah ini tampil pertama kali saat aplikasi dijalankan, yang berisi *textbox username* dan *password* juga *button* untuk *login* dan tombol *exit* untuk keluar aplikasi.

Gambar 19 : Tampilan Layar Form Login

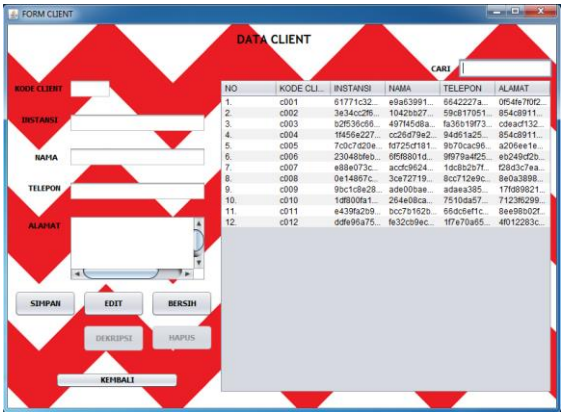
4.2. Tampilan Layar *Form* Menu Utama

Form menu utama berisi macam-macam menu yang dapat digunakan seperti *Client*, Paket, Program, *Profile, About, Help,* dan *logout* untuk kembali ke halaman login.

Gambar 20 : Tampilan Form Menu Utama

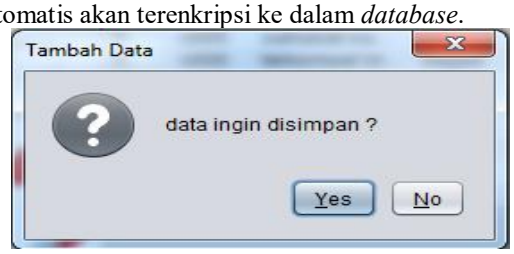
4.3. Tampilan Layar *Form Client*

Tampilan layar *form client* dibawah ini tampil saat pengguna memilih tombol *client* pada form menu utama.



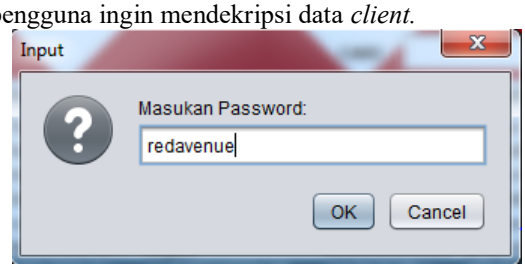
Gambar 21 : Tampilan Layar Form Client

Tampilan gambar dibawah menggambarkan tampilan *popup* konfirmasi jika *user* ingin menyimpan data *client*. Data yang ingin disimpan otomatis akan terenkripsi ke dalam *database*.



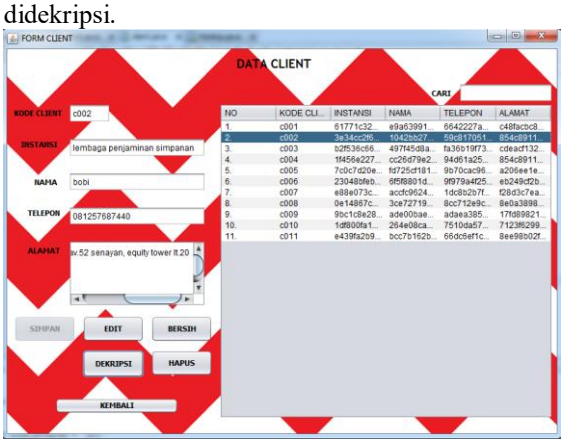
Gambar 22 : Tampilan Popup Konfirmasi Simpan

Tampilan gambar dibawah ini menggambarkan tampilan *popup* konfirmasi *input password* jika pengguna ingin mendekripsi data *client.*



Gambar 23 : Tampilan Popup Konfirmasi Dekripsi

Tampilan gambar dibawah ini menggambarkan tampilan hasil dekripsi saat *user* sebelumnya memilih data pada *record* tabel yang ingin didekripsi.



Gambar 24 : Tampilan Layar Hasil Dekripsi Client

4.4. Hasil Pengujian

4.4.1 Hasil Pengujian Tabel *Client*

Berikut adalah tabel pengujian proses enkripsi *database* dengan algoritma AES 128.

Tabel 2: Hasil Pengujian Enkripsi Tabel Client

Nama Data	Karakter Asli	jumlah	Karakter Hasil Enkripsi	Jumlah Karakter Hasil Enkripsi
kode_client	C001	4	-	-
instansi	fakultas hukum universitas padjajaran	37	61771c32453e64711f70446b195a8b1d4f33045e44e5a0815eea59b2c631fc1b91dd6af55cf3397634e65c8d193b6242	96
nama	andrian	7	e9a639911428957d9960508eb918cd65	32
telepon	085795839281	12	6642227a6922e137ce055103fda645de	32
alamat	jl, raya bandung sumedang	25	c48facbc8a54c75d060b0bc8f89301c713f1459ca866f28b0acd0abeee11855f	64

4.5 Evaluasi Program

Setelah melakukan Analisa dari pengujian aplikasi tersebut, maka terdapat beberapa kelebihan dan kekurangan dari aplikasi tersebut, yaitu sebagai berikut:

a. Kelebihan Program

1) Aplikasi akan otomatis mengenkripsi data ke dalam *database*.
2) *Database* yang telah terenkripsi tidak dapat terbaca oleh orang yang tidak diinginkan.
3) Tidak ada perubahan didalam *database* setelah didekripsi kembali.

b. Kekurangan Program

1) Aplikasi hanya bisa enkripsi dan dekripsi *database* per-*record* saja.
2) Aplikasi masih sangat sederhana.

5. KESIMPULAN

5.1. Kesimpulan

Kesimpulan yang bisa diperoleh dari perancangan, pembuatan, uji coba dan Analisa aplikasi tersebut, maka dapat dibuat beberapa kesimpulan sebagai berikut :

a. Dengan dibuatnya aplikasi ini, penyimpanan data ke database akan menjadi lebih aman.
b. Proses enkripsi dan dekripsi tidak memakan waktu banyak.
c. Algortma AES 128 bisa diimplementasikan ke dalam aplikasi pengamanan database.

5.2. Saran

Saran yang diperlukan agar aplikasi tersebut menjadi lebih baik lagi, adalah :

a. Pengguna yang tidak mengerti database sekalipun diharapkan dapat mengoperasikan aplikasi ini.
b. Interface masih sangat sederhana, diharapkan dapat ditambah beberapa fitur progress bar untuk proses enkripsi dan dekripsi
c. Aplikasi ini diharapkan dapat dikembangkan lagi, dapat menjadi enkripsi per-tabel.

DAFTAR PUSTAKA

[1] Candra, B., Wahyudi, J., & Hermawansyah. , 2014. Pengembangan Sistem Keamanan Untuk Toko Online Berbasis Kriptografi AES Menggunakan Bahasa Pemrograman PHP dan MySql. *Jurnal Media Infotama*, *11*(1), 31–39.

[2] Minarni, & Susanti. , 2014. Sistem Informasi Inventory Obat Pada Rumah Sakit Umum Daerah (Rsud) Padang. *Momentum*, *16*(1), 103–111.

[3] Hasrul, H., & Siregar, L. H. , 2016. Penerapan Teknik Kriptografi Pada Database Menggunakan Algoritma One Time Pad, *2*(2), 41–52.

[4] Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. , 2015. Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Jurnal Informatika Mulawarman*, *10*(1), 20–31.

[5] Andreas, E. L. Y. , 2014. Aplikasi Kriptografi Pengamanan Doc, Docx , Jpg Dan Pdf Dengan Metode AES Dan Kompresi Huffman.

[6] Rahmawati, R., & Rahardjo, D. , 2016. Aplikasi Pengamanan Data Menggunakan Algoritma Steganografi Discrete Cosine Transform dan Kriptografi grafi AES 128 BIT pada SMK PGRI 15 Jakarta. *Jurnal Teknik Informatika Dan Sistem Informasi*, *2*(April), 67–74.