ADVANCED ENCRYPTION STANDARD UNTUK KEAMANAN BASIS DATA DI KANTOR KECAMATAN KARANG TENGAH

Nadiah Safitri[1]. Dewi Kusumaningsih[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: nadiasafitri777@gmail.com[1]. Dewi.kusumaningsih@budiluhur.ac.id[2]

ABSTRAK

Masalah keamanan dan kerahasian database adalah salah satu hal yang penting udah dilakukan perlindungandan terhadap data yang disimpan disuatu instansi terutama pada kantor Kecamatan Karang Tengah Kota Tangerang. Data tersebut adalah data Penduduk yang tersimpan didalam Basis Data yang harus terjamin keamanan sehingga tidak disalahgunakan oleh orang-orang yang tidak bertanggung jawab ataupun yang tidak berwenangan atas data tesebut. Untuk itu perlu dilakukan perlindungan terhadap data Penduduk yang ada di Kantor Kecamatan Karang Tengah Kota Tangerang. Metode kriptografi yang digunakan pada penelitian ini adalah Advanced Encryption Standard (AES) dengan panjang kunci 128 bit. Kriptografi merupakan seni dalam mengamankan data penduduk yang didalam database menjadi suatu data yang tidak bisa dikenali atau dibaca oleh siapapun. Algoritma kriptografi ini sangat sederhana dan mudah diimplementasikan. Aplikasi pengamanan database ini berbasis desktop dengan bahasa pemrograman Java. Dengan adanya aplikasi ini maka data penduduk dapat dienkripsi oleh user menjadi data yang tidak dapat dibaca oleh siapapun sehingga keamanan dan kerahasiaan datanya dapat terjaga dengan baik dan aman.

Kata Kunci: Basis Data, Kriptografi, *Advanced Encryption Standard* (AES 128)

1. PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi komputer dan telekomunikasi yang semakin pesat, keamampuan untuk mengakses dan menyediakan informasi data secara cepat dan akurat yang dibutuhkan oleh suatu instansi.

Semakin tinggi tingkat teknologi komputer, semakin tinggi pula tingkat ancaman yang mengancam keamanan data didalam komputer. Terlebih jika data yang disimpan bersifat rahasia yang sangat penting. Salah satu dampak negatif dalam perkembangan teknologi adalah pencurian data yang sangat merugikan bagi pemilik data atau instansi, untuk menghindari kejahatan tersebut maka dibutuhkannya pengamananan dalam penyimpanan data yang dianggap penting agar terhindarnya dari kejahatan teknologi informasi.

Kecamatan Karang Tengah merupakan sebuah Kecamatan yang beralamat di Jl. Sunan Giri No.20 Karang Tengah Kota Tangerang, yang tugasnya melayani warga dan menyimpan data-data warga sehingga memiliki banyak data-data penting dan tidak boleh ada orang lain yang mengetahuinya selain yang bersangkutan, salah satunya seperti data kependudukan. Data tersebut hanya disimpan begitu saja didalam *database server* tanpa adanya sistem keamanan.

Aplikasi ini dibuat agar lebih memudahkan dalam penyimpanan data kependudukan di Kecamatan Karang Tengah dan memberi cadangan data bila terjadinya suatu musibah seperti bencana alam, kebakaran, dan pencurian data oleh pihak-pihak yang tidak bertanggung jawab. Pada aplikasi ini data yang masuk ke *database* sudah terenkripsi dan apabila data dipanggil kembali data sudah terdekripsi.

Melihat permasalahan yang telah di uraikan diatas maka penulis memilih sebuah algoritma kriptografi yang digunakan dalam penyusunan tugas akhir ini, penulis memilih algoritma AES 128 karena dapat di implementasikan untuk pengamanan *database*.

2. LANDASAN TEORI

2.1. Pengertian *Database*

Database merupakan sekelumpulan dari data yang saling berhubungan satu dengan lain yang tersimpan diluar komputer dan digunakan pada perangkat lunak tertentu untuk memanipulasi. *Database* merupakan salah satu komponen yang

penting pada suatu sistem informasi, untuk penggunanya. [1]

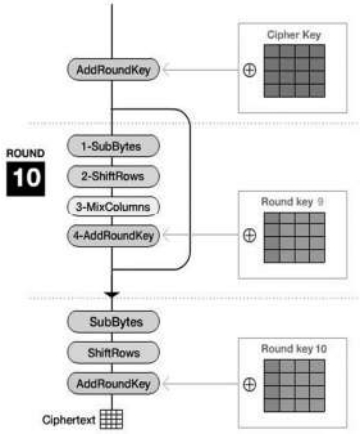
2.2. Definisi Kriptografi

Kriptografi berasal daru bahasa Yunani crypto dan Graphia. Crypto berati rahasia (secret) dan Graphia berarti tilisan (writing). Menurut terminologinya, kriptografi merupakan seni dan ilmu menyembunyikan informasi dari penerima yang tidak berhak (unauthorized persons). Kriptotografi dengan cara mendapatkan plaintext dan ciphertext yang digunakan untuk mendapatkan informasi berharga mengubah atau memalsukan data dengan tujuan untuk menipu penerima yang sesunggunya. [2]

Tujuan utama dari kriptografi yang merupakan aspek keamanan sistem informasi sebagai berikut:

1) Kerahasiaan (*confidentiality*)
2) Integritas Data
3) Otentikasi (*authentication*)
4) Anti-penyangkalan (*non-repuditation*)

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



Gambar 1: Algoritma Enkripsi AES

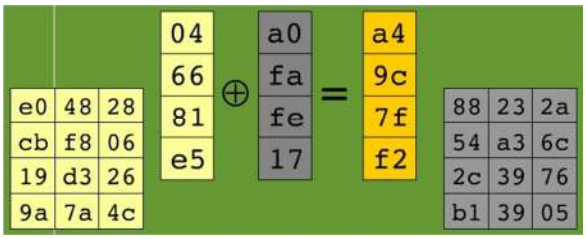
2.4. Proses Enkripsi AES

Pada proses enkripsi, pesam (M) akan disandikan dengan menggunkan kunci enkripsi (key) atau sandi sehingga menjadi suatu sandi yang tidak dapat dimengerti (C). [3]

Berikut adalah enkripsi Algoritma AES:

1. *AddRoundKey*

   AddRoundKey() adalah menggabungkan antara chiper teks yang sudah ada dengan cipher key yang cipher key nya tersebut yang telah dihubungkan dengan XOR.



Gambar 2: *AddRoundKey*

2. *SubBytes*

   SubBytes() merupakan proses subtitusi byte dengan menggunakan table S-Box. [4]

Tabel 1: Tabel S-Box Untuk Enkripsi

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

3. *ShiftRows*

   Pada *ShiftRows*() ini adalah proses melakukan pergeseran dengan cara wrapping. [5]

4. *MixColumns*

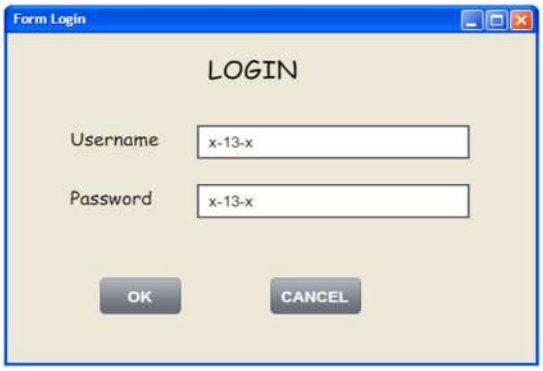
   Transformasi menggunakan *Mix Columns*() dalah proses ketiga dalam satu ronde enkripsi AES. Disini mixcolumns untuk mengacak data pada kolom masing-masing array state.

3. RANCANGAN SISTEM DAN APLIKASI

3.1. Rancangan Layar

Rancangan layar yang diberikan merupakan representasi dari aplikasi yang akan dibuat nantinya. Oleh karena itu rancangan layar yang dibuat harus mudah dipahami dan dimengerti, agar dalam menggunakan aplikasi ini *user* dapat merasakan nyaman dalam menggunakannya sehingga rancangan layar tidak membuat bingung pengguna dan tidak mengalami kesulitan saat menggunakan aplikasi ini. Dalam aplikasi ini akan digambarkan rancangan layar masing-masing *form*, yaitu rancangan layar *form login*, Menu Utama, *Form* Penduduk, *Form* Kartu Keluarga, Form Detail Pindah, Help dan About.

Berikut ini merupakan gambar 4 rancangan layar *Login*. *Login* adalah antarmuka yang pertama kali muncul pada saat aplikasi dijalankan. *User* dapat menggunakan aplikasi ini dengan cara menginput *username* dan *password*.



Gambar 4: Rancangan Layar *Login*

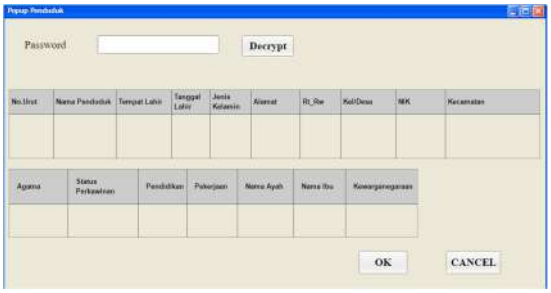
Kemudian gambar 5 merupakan rancangan layar Menu Utama yang merupakan *form* pertama ditampilkan pada saat *user* selesai *Login*. Pada form ini terdapat 6 menu diantaranya menu Penduduk, Kartu Keluarga, Detail Pindah, *Help*, *About* dan *Logout*.

Gambar 5: Rancangan Layar Menu Utama

Gambar 6 merupakan rancangan layar form penduduk. Pada *form* penduduk *user* dapat melakukan penyimpanan data penduduk baru untuk di *encrypt*, pengeditan data penduduk dan penghapusan data penduduk pada *form* ini.

Gambar 6: Rancangan Layar Form Penduduk

Rancangan layar pada gambar 7 merupakan rancangan layar *form Popup* Penduduk. *User* dapat melakukan *decrypt* yang telah di *encrypt* dengan menggunakan *password* dan tombol *decrypt* pada form ini.

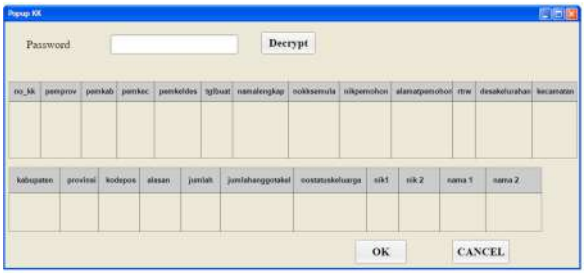


Gambar 7: Rancangan Layar *Popup* Penduduk

Rancangan layar pada gambar 8 merupakan *form* Kartu Keluarga ini digunakan untuk menginput data kartu keluarga yang akan disimpan dalam *database*. Juga dapat *Update*, *Delete* data jika mengalami perubahan. Jika ingin mengubah informasi tentang akun, maka menggunakan *button update*. Jika ingin menghapus akun dari *database* maka menggunakan *button delete*. Sebelum mengubah atau menghapus data dari database, maka harus mendekrip terlebih dahulu menggunakan menu *popup* kartu keluarga.

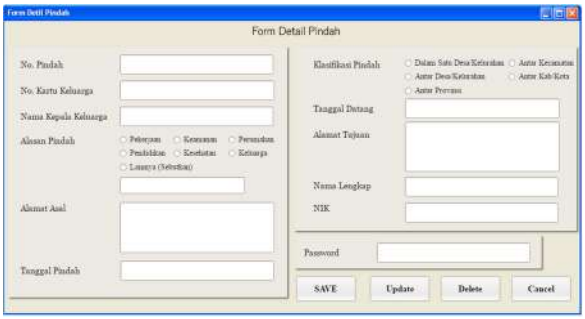
Gambar 8: Rancangan Layar *Form* Kartu Keluarga

Pada gambar 9 merupakan rancangan layar *form popup* Kartu Keluarga. *User* dapat melakukan *decrypt* data kartu keluarga yang telah di *encrypt* dengan menggunakan *password* dan tombol *decrypt* pada *form* ini.



Gambar 9: Rancangan Layar Popup Kartu Keluarga

Pada gambar 10 merupakan rancangan layar *form* Detail Pindah. pada *form* Detail Pindah *user* dapat melakukan penyimpanan data Detail Pindah baru untuk di *encrypt*, mengupdate dan menghapus data Detail Pindah pada *form* ini.



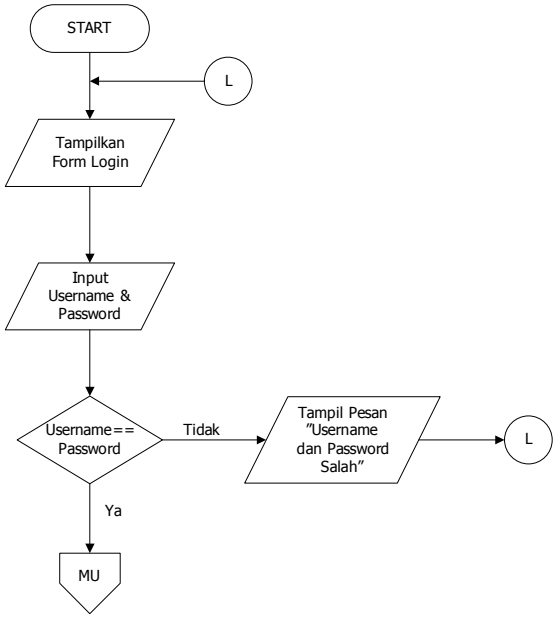
Gambar 10: Rancangan Layar Detail Pindah

Pada gambar 11 merupakan rancangan layar form detail pindah. *user* dapat melakukan *decrypt* yang telah di *encrypt* dengan menginput *password* dan memilih tombol *decrypt* pada *form* ini.

Gambar 11: Rancangan Layar *Form Popup* Detail Pindah

3.2. *Flowchart* Program

Pada gambar 12 menggambarkan proses yang terjadi pada halaman *login*. Jika benar memasukan *username* dan *password* maka akan diarahkan ke menu utama. Jika tidak akan tampil pesan *error*.



Gambar 12: *Flowchart* Proses *Login*

Gambar 13 merupakan alur proses dari *form* penduduk terenkripsi.

Gambar 14: *Flowchart Popup* Penduduk

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Layar

Gambar 15 adalah tampilan layar utama *user* wajib melakukan *login* untuk menggunakan aplikasi.

Gambar 15: Tampilan Layar *Form Login*

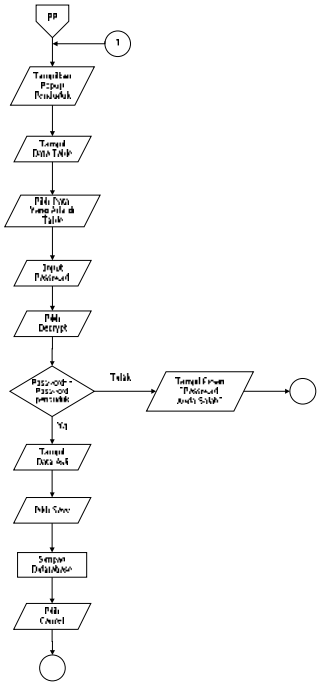
Gambar 16 adalah tampilan layar dari Menu Utama, form pertama yang ditampilkan pada saat user selesai *login*, pada *form* ini terdapat 9 menu diantaranya menu Penduduk, Kartu Keluarga, Detail Pindah , *Decrypt* Penduduk, *Decrypt* Kartu Keluarga, *Decrypt* Detail Pindah, *Help* dan *About*.

Gambar 16: Tampilan Layar *Form* Menu Utama

Pada gambar 17 merupakan tampilan form input Pendudukm user dapat melakukan penyimpanan data penduduk baru yang akan di *encrypt*, *update* dan *delete* data penduduk yang sudah di *encrypt*. Untuk melakukan proses *encrypt*, *user* harus menginput *password* untuk melakukan *decrypt*.

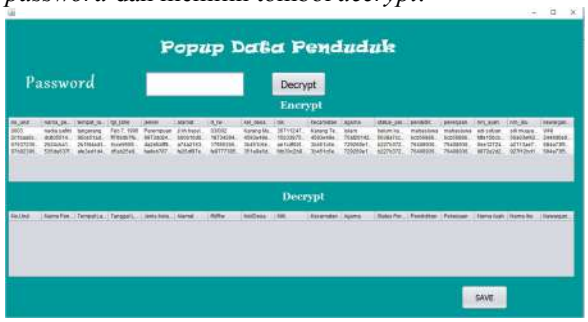
Gambar 13: *Flowchart* Proses *Form* Penduduk

Gambar 14 merupakan alur proses dari *popup* penduduk terdekripsi.



Gambar 17: Tampilan *Form* Input Penduduk

Pada gambar 18 merupakan *form* popup penduduk, *user* dapat melakukan decrypt data penduduk yang telah di *encrypt* dengan cara mengisi *password* dan memilih tombol *decrypt*.



Gambar 18: Tampilan *Form Popup* Penduduk

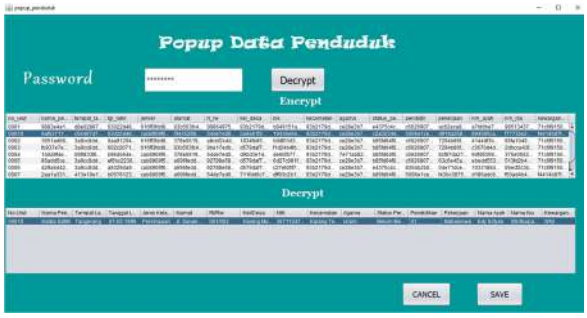
4.2. Pengujian Aplikasi

Pada tahap ini dilakukan proses tahap uji coba aplikasi yang bertujuan untuk mengetahui kinerja dari aplikasi yang telah dibuat. Pada tahap ini aplikasi akan diuji mengenai proses *encrypt* dan proses *decrypt*.

Untuk melakukan proses *encrypt*, *user* terlebih dahulu memilih menu "penduduk". Kemudian pada menu "Penduduk" *user* dapat melakukan input data penduduk.

Gambar 19: *Form* Penduduk

Selanjutnya untuk melihat hasil proses enkripsi dan ingin melakukan proses dekrip, *user* dapat memilih menu *decrypt* penduduk.

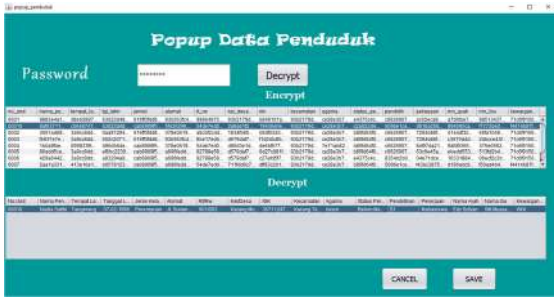


Gambar 20: Tampilan Hasil Proses Enkripsi

Pada gambar 21 ini memperlihatkan tampilan data penduduk yang sudah disimpan di *database*.

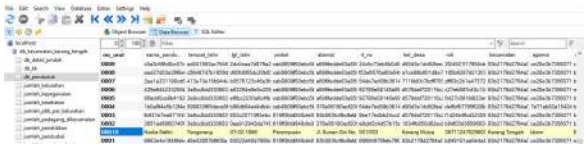
Gambar 21: Data Penduduk di *Database*

User jga bisa melakukan *update* data penduduk dan *delete* data yang sudah di *encrypt*. Untuk melakukan *update* dan *delete* data penduduk, user terlebih dahulu melakukan proses *decrypt* dengan cara memasukan *password* dan memilih tombol button *decrypt*, berikut hasil proses *decrypt*.



Gambar 22: Hasil Proses Dekripsi

Lalu setelah berhasil mendekripsi maka *user* harus memilih tombol button "*save*" untuk menyimpan ke dalam *database*. Dapat dilihat pada gambar 23 dibawah ini:



Gambar 23: Data Penduduk Dekripsi di *Database*

5. KESIMPULAN

Berdasarkan hasil pengujian yang dikeluarkan oleh sistem maka dapat diambil kesimpulan sebagai berikut:

1) Aplikasi kriptografi ini menggunakan algoritma AES-128 dan dibangun agar dapat mengamankan *database*.
2) Dengan adanya aplikasi kriptografi ini maka pihak kecamatan karang tengah dapat mengamankan data penduduk, kartu keluarga, dan detail pindah, agar terjaga kerahasiaannya.
3) Tidak terjadi kerusakan atau perubahan pada isi *database* yang telah didekripsi.
4) Dibuatnya menu bantuan dalam aplikasi ini agar *user* dapat menggunakan aplikasi dengan mudah.

Berdasarkan keterbatasan aplikasi ini, beberapa saran yang dapat diberikan untuk pengembangan aplikasi lebih lanjut dengan adanya perkembangan kedepannya, antara lain:

1) Pengembangan labih lanjut dapat difokuskan pada penggunaan enkripsi AES-128 dengan kombinasi yang lain untuk meningkatkan keamanan *database*.
2) Interface masih sangat sederhana diharapkan bisa ditambah beberapa *fitur* seperti waktu lama proses enkripsi dan dekripsi.

DAFTAR PUSTAKA

[1]. Benni Candra, Jusuf Wahyudi, H. (2014) 'Pengembangan Sistem Keamanan Untuk Toko Online Pemrograman Php Dan Mysql', *Jurnal Media Infotama*, 11(1), pp. 31–39. Available at: http://bit.ly/2Btx23D

[2]. Alyanto, D., Studi, P., Informatika, T., Tinggi, S., Informatika, M., Komputer, D. A. N. and Time, S. (2016) 'Pengenkripsian Data Rahasia'. Available at: http://bit.ly/2j2mbpw

[3]. Haji, W. H. and Mulyono, S. (2012) 'IMPLEMENTASI RC4 STREAM CIPHER UNTUK KEAMANAN BASIS DATA', *Informasi, Jurusan Sistem Komputer, Fakultas Ilmu Mercu, Universitas Jakarta, Buana*, 2012(Snati), pp. 15–16. Available at: http://bit.ly/2HoTcqJ

[4]. Rohman, N. (2010) 'Pembuatan Kunci Lisensi Program Pengubah Atribut File', 4(2), pp. 59–69. Available at: http://bit.ly/2Am6dhD

[5]. Primartha, R. (2013) 'Penerapan Enkripsi dan Dekripsi File menggunakan Algoritma Advanced Encryption Standard (AES)', *Journal of Research in Computer Science and Applications*, 2(1), pp. 13–18