SISTEM KEAMANAN ATM DENGAN MENGGUNAKAN ENKRIPSI AES PADA KARTU ATM.

Arief Agung Gumelar[1)], Latief Adam Busyairi[2)], Muhammad Fajrian Noor[3)]

[1), 2), 3)] *Teknik Informatika STMIK AMIKOM Yogyakarta*
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : mail@ariefagung.com[1)], latief0027@students.amikom.ac.id[2)], mfajrian09@gmail.com[3)]

Abstrak

Sistem keamanan atm seperti yang di terapkan belakangan ini belum sepenuhnya mampu untuk membuat sistem berjalan secara *secure* , berbagai serangan dan ancaman terhadap sistem masih terjadi dengan intensitas yang cukup tinggi hal ini tentunya membuat nasabah tidak nyaman akan dampak negatif yang di hasilkan, pasalnya diperkirakan total kerugian yang terjadi dalam satu kali aksi pembobolan jumlahnya bisa mencapai jutaan bahkan milyaran rupiah, Pada penelitian yang kami ajukan ini akan dibahas sebuah metode pengamanan pada atm dengan memadukan kriptografi di dalamnya, metode yang digunakan adalah melakukan enkripsi pada sisi kartu atm dengan algoritma AES sehingga data yang di simpan pada kartu atm yang sebelumnya berupa nomor tanpa enkripsi akan berubah mejadi kode-kode enkripsi sehingga hal ini dapat meminimalisir kejahatan pembobolan atm dengan melakukan duplikasi kartu atm.

Kesadaran perlunya menjaga kartu atm juga perlu di perhatikan karena ini merupakan upaya awal yang dipat dilakukan untuk menjaga kartu atm yang di berikan oleh pihak bank. Penelitian ini akan menjelaskan metode kriptografi AES yang digunakan dalam enkripsi *plaintext* pada kartu atm menjadi *ciphertext* sehingga nantinya ketika kartu digunakan untuk transaksi maka mesin atm harus mampu melakukan deskripsi terlebih dahulu dari hasil enkripsi yang ada pada kartu atm.

Kata Kunci: Keamanan, menjaga, kriptografi, atm, enkripsi, kartu, aes.

1. Pendahuluan

Keamanan sebuah sistem dalam jaringan merupakan hal penting yang sangat krusial terlebih jika sistem tersebut melibatkan transaksi keuangan seperti yang di jalankan oleh perbankan, berbagai metode keamanan telah diterapkan oleh pihak bank mulai dari penerapan sandi pada atm dengan ketentuan yang tidak boleh mengkombinasikan tanggal lahir, penggunaan sandi dengan minimal 6 karakter, pemeliharaan sandi yang harus ganti minimal 1 tahun sekali. Hal tersebut dilakukan semata-mata untuk menjaga keamanan dalam bertransaksi, banyaknya kasus pembobolan atm yang sering terjadi memiliki dampak *negative* bagi yang memanfaatkan jasa layanan bank karena hal ini bisa menurunkan dampak kepercayaan pengguna kepada pihak bank terkait kasus yang marak terjadi belakangan ini jika di biarkan dalam jangka panjang hal ini kemungkinan akan menimbulkan dampak pada bidang ekonomi karena ketika pengguna jasa bank sudah tidak lagi percaya untuk menitipkan uangnya dibank maka tentunya mereka akan menguras uang simpanan mereka di bank dan akan memindahkannya ketempat yang lain yang dirasa aman namun perlu diketahui bahwa tindakan tersebut bisa menimbulkan guncangan terhadap kestabilan perokonomian suatu negara.

Peran kriptografi dalam hal ini memiliki tingkatan yang sangat penting karena dengan metode kriptografi yang di terapkan setidaknya akan mengamankan sistem yang sebelumnya komunikasi data berjalan apa adanya tanpa adanya enkripsi namun dengan diterapkannya metode kriptografi maka data yang keluar masuk pada sistem akan terenkripsi menjadi bentuk gabungan dari angka dan huruf yang acak, hal ini akan menambah tingkat ke amanan pada sebuah sistem juga akan mempersulit bagi mereka yang akan melakukan kejahatan, kriptografi yang di terapakan pada proses enkripsi kali ini berfokus pada enkripsi nomor yang ada pada kartu atm sehingga nantinya tujuan akhir dari metode ini seluruh nomor yang ada pada kartu atm berbentuk enkripsi selain itu pada jurnal ini penulis juga akan mengulas tentang cara kerja atm, berbagai macam kejahatan pembobolan yang sering dilakukan dan juga sistem keamanan yang menjadi solusi untuk mengatasi masalah tersebut[1].

2. Pembahasan

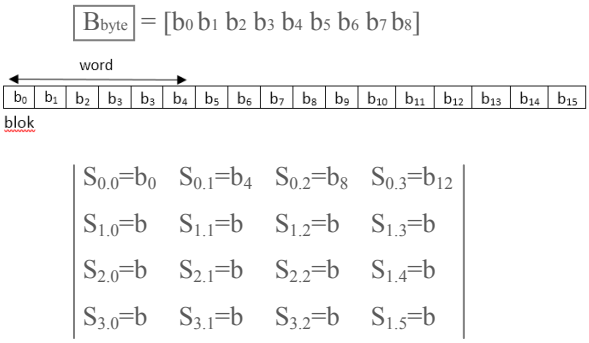
Kriptografi awalnya dikenal sebagai ilmu yang mempelajari bagaimana cara menyandikan pesan namun ketika kita merujuk pada pengertian kriptografi secara modern adalah sebuah teori yang didasarkan pada teori matematika dimana digunakan sekumpulan teknik untuk menyediakan keamanan bagi informasi ataupun data. Ada banyak algoritma yang bisa di gunakan dalam melakukan enkripsi sebuah informasi namun pada pembahasan kali ini penulis akan lebih focus untuk pembahasan algoritma menggunakan AES (*Advanced Encryption Standard*)

AES memiliki invers dengan panjang blok 128 bit[1] sehingga system penyandian blok disebut non-Feistel. Sistem penyandioan AES memiliki proses yang berulang biasanya hal ini sering disebut juga dengan ronde penentuan jumlah ronde dalam AES bergantung pada panjang kunci yang di gunakan.

Tabel 1. *Hubungan jumla ronde dan panjang kunci AES*

Panjang Kunci AES (bit)	Jumlah Ronde (Nr)
128	10
192	12
256	14

Sebelum berlanjut untuk membahas lebih jauh mengenai AES perlu kita ketahui terlebih dahulu bahwa karakteristik AES menggunakan 5 ukuran data, diantaranya adalah byte,bit,word,blok dan state dalam nilai digit sistem biner bit merupakan nilai satuan data terkecil lebih kecil dari Byte karena terdapat perbedaan antara keduanya, jika bit adalah nilai terkecil dalam satuan data sedangkan Byte terdiri dari 8 bit, word berukuran 4 byte atau sama dengan 32 bit, blok berukuran 16 byte atau sama dengan 128 bit dan state adalah blok yang di atur sebagai matrik byte berukuran 4x4 seperti gambar berikut[1].



Gambar 1. *unit data AES*

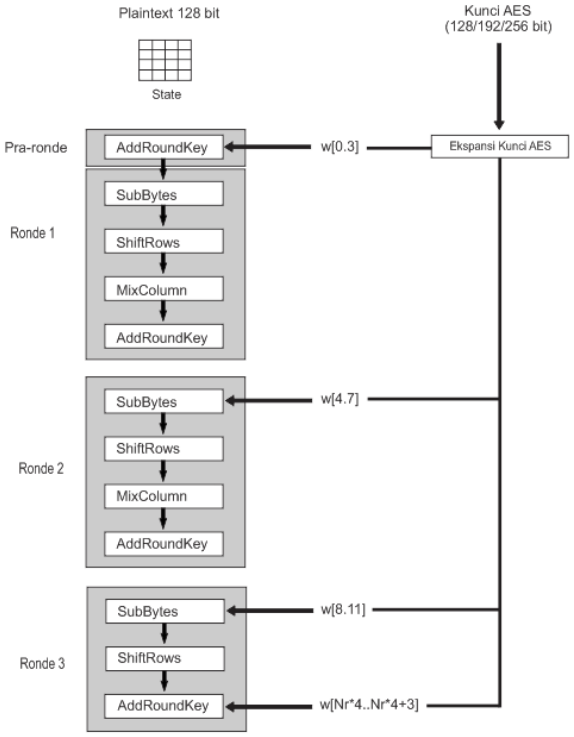
2.1 Struktur Enkripsi Pada AES

Dalam enkripsi struktur AES merupakan transformasi terhadap State sebuah teks asli berupa plaintext dalam blok 128 bit terlebih dahulu akan diorganisir sebagai State. enkripsi AES adalah suatu transformasi terhadap State yang dilakukan secara berulang dalam beberapa ronde state yang menjadi keluaran ronde k menjadi masukan untuk ronde ke-k+1[2].

secara garis besar desain enkripsi aes diberikan oleh gambar 2. mulanya sebuah plaintext di organisasi sebagai sebuah state setelah itu sebelum ronde 1 mulai teks asli yang berupa plaintext dicampur dengan kunci ronde ke-0 pada bagian ini disebut sebagai *AddRoundKey* kemudian ronde ke 1 sampai dengan ronde ke (NR-1) dengan Nr merupakan jumlah ronde menggunakan 4 jenis transformasi, yaitu *SubBytes, ShiftRows, MixColumns dan AddRoundKey*. Pada ronde ke-Nr akan dilakukan transformasi serupa dengan ronde lain namun tanpa MixColumns[3].

2.2 Struktur Deskripsi Pada AES

Dalam proses enkripsi yang dilakukan sebelumnya untuk mengubah sebuah text asli (*plaintext*) menjadi *ciphertext* didasarkan pada beberapa langkah-langkah yang disebut sebagai ronde, namun jika dalam proses deskripsi ada sedikit perbedaan karena pada dasarnya proses ini adalah melakukan dekrip dari sebuah kode enkripsi yang di sebut ciphertext menjadi text aslinya lagi (*plaintext*). Algoritma dekripsi AES menggunakan transformasi invers semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Sehingga dalam transformasi dasar AES memiliki invers, yaitu *InvSubBytes, InvShiftRows dan InvMixColumns.*



Gambar 2. *Struktur Enkripsi AES*

AddRoundKey adalah transformasi yang sifatnya *self-invers* memiliki syarat menggunakan *Key* yang sama seperti ditunjukkan pada gambar 2. diatas adalah algoritma deskripsi untuk AES.

Algoritma Enkripsi AES

Input: P,K {Teks Asli 16 bytes, kunci AES(128,192,256 bit)}

Output: CT {Teks sandi 16 bytes}

(*Nr*,*w*) ← **EkspansiKunci**(K) {*Nr* : Jumlah Ronde, *w* : larik bytes kunci ronde}

CT = P

AddRoundKey(CT,*w*[0..3])
For *i* = 1 → *Nr* **do**
SubBytes(CT)
ShiftRows(CT)
if ≠ *Nr* **then**
MixColumns(CT)
end if
AddRoundKey(CT,*w*[(*i**4)..(*i**4)+3])
end for

Algoritma Deskripsi AES

Input: CT,K{Teks sandi 16 bytes, kunci AES (128,192, 256 bit)}
Output: P {Teks asli 16 bytes}
(*Nr*,*w*) ← **EkspansiKunci**(K) {*Nr* : Jumlah Ronde, *w* : larik bytes kunci ronde}
P=CT

AddRoundKey(P,*w*[*Nr**4..*Nr**4-3])
For *i* = 1 → *Nr* **do**
InvSubBytes(P)
InvShiftRows(P)
AddRoundKey(P,*w*[(*Nr-i*) *4)..((*Nr-i*)*4+3])
if ≠ *Nr* **then**
MixColumns(P)
end if
end for

2.3 Transformasi-transformasi AES

Algoritma enkripsi AES menggunakan 4 jenis transformasi: substitusi yang disebut dengan **SubBytes**, permutasi yang disebut dengan **ShiftRows**, pencampuran yang disebut dengan **MixColumns**, dan penambahan kunci yang disebut **AddRoundKey.**

2.3.1 SubBytes

AES menggunakan substitusi nonlinier pada ukuran **byte** yang disebut **SubBytes.** Setiap elemen pada *state* dari elemn $s_{(0,0)}$ sampai dengan $s_{(3,3)}$ dikenakan transformasi **SubBytes**.

2.3.1.1 Transformasi dengan Tabel Substitusi

Transformasi **SubBytes** dapat menggunakan tabel substitusi, yaitu dengan cara menginterpretasikan *byte* masukan $s_{i,j}$ sebagai 2 bilangan heksadesimal, kemudian digit kiri menunjukkan indeks baris dan digit kanan menunjukkan indeks kolom di tabel substitusi. Nilai *byte* pada tabel substitusi yang dirujuk oleh indeks baris dan kolom menjadi nilai yang mensubstitusi $s_{i,j}$. Tabel substitusi untuk **SubBytes** diberikan oleh Tabel 2.1. sedangkan tabel invers substitusi **SubBytes** (transformasinya diberi nama **InvSubBytes**) diberikan oleh tabel 2.

Tabel 2. *Tabel substitusi untuk transformasi SubBytes*

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

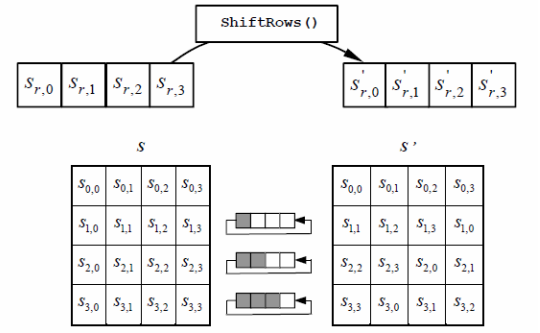
Tabel 3. *Tabel substitusi untuk transformasi InvSubBytes*

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

2.3.2 ShiftRows dan InvShiftRows

2.3.2.1 ShiftRows

Selain menggunakan substitusi untuk mengganti nilai pada elemen *state*, AES menggunakan permutasi pada *state*. Transformasi permutasi pada *state* disebut dengan transformasi **ShiftRows. ShiftRows** dilakukan dengan menjalan operasi *circular shift left* sebanyak i pada baris ke-i pada *state*. Ilustrasi transformasi **ShiftRows** diberikan oleh Gambar 3.



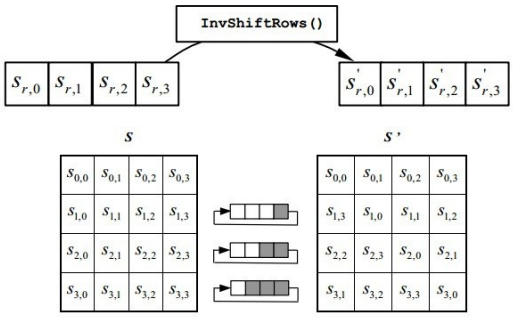
Gambar 3. *Transformasi ShiftRows*

Transformasi **ShiftRows** merupakan jenis transformasi permutasi, yaitu pengubahan posisi elemen pada *state* tanpa mengubah nilainya. Transformasi **ShiftRows** terlihat sederhana jika dilihat melalui representasi *state*. Namun, karena *state* adalah representasi blok dengan oriental per kolom menjadikan transformasi **ShifRows** menjadi rumit jika dilihat dari sudut pandang blok.

2.3.2.2 InvShiftRows

Tranformasi invers terhadap **ShiftRows** disebut **InvShiftRows.** Transformasi **InvShiftRows** terhadap sebuah *state* menggunakan operasi (*circular shift right*)

pada tiap barisnya yang banyak gesernya sesuai dengan indeks baris seperti yang ditunjukkan oleh Gambar 4.



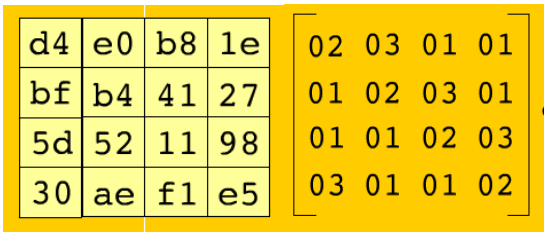
Gambar 4. *Transformasi InvShiftRows*

2.3.3 MixColumn

Tujuan transformasi **MixColumn** adalah mencampur nilai kolom pada *state* pada satu elemen *state* keluaran. Untuk melakukan pencampuran itu, transformasi **MixColumn** menggunakan operasi perkalian matriks dengan operasi perkalian matriksdengan operasi perkalian.

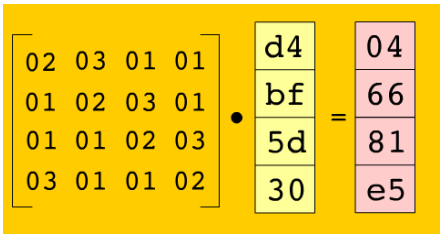
$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Gambar 5. *Formula penyelesaian MixColumn*



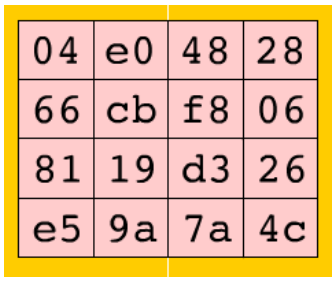
Gambar 6. *Contoh MixColumn dan nilai matriks*

Gambar matriks diatas dikalikan berurutan dimulai dengan mengambil pada kolom ke-1 ($a_{(0,0)}$ sampai $a_{(0,3)}$) dengan matriks 1 dilanjutkan hingga kolom terakhir ($a_{(3,0)}$ sampai $a_{(3,3)}$) dengan matriks 4.



Gambar 7. *Metode perkalian*

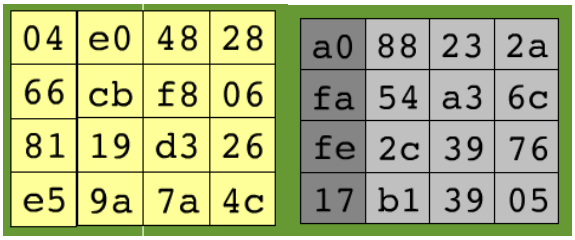
Dan akan menemukan hasil untuk melanjutkan ke metode selanjutnya yang ada pada Gambar 8.



Gambar 8. *Hasil dari perkalian*

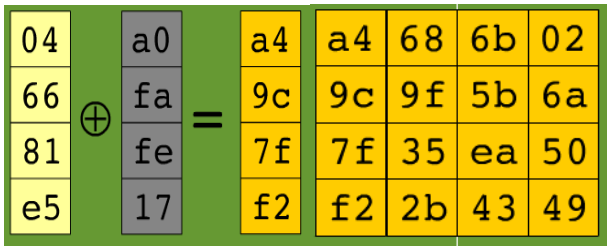
2.3.4 AddRoundKey

Transformasi keempat yang digunakan pada penyandian AES adalah transformasi **AddRoundKey**. Transformasi **AddRoundKey** mencampur sebuah OR(⊕). Setiap elemen pada *state* masukan yang merupakan sebuah byte dikenakan operasi eksklusif OR dengan bytepada posisi yang sama di kunci ronde (kunci ronde direpresentasikan sebagai *state*).



Gambar 9. *Hasil MixColumn dan RoundKey*

Kalikan kolom pertama *Round Key* dengan kolom pertama hasil *MixColumn* begitu juga setelahnya



Gambar 10. *Meode XOR*

Kalikan begitu juga seterusnya hingga menemukan hasil dari round pertama yang ada dan akan ketemu hasil akhir setelah melakukan *looping* hingga 9 kali.

2.3 Kriptografi Pada ATM

Dalam proses transaksi ATM biasanya menggunakan sebuah kartu debet yang dilengkapi dengan pin dengan jumlah sebanyak 6 karakter, pin yang telah diterapkan bersifat rahasia termasuk pegawai bank bersangkutanpun tidak boleh mengetahui pin debet dari setiap nasabahnya karena ini merupakan privasi yang harus terjamin bagi setiap pemegang kartu atm.

Dalam proses transaksi atm mulanya pemilik kartu memasukkan kartu debetnya kedalam mesin atm, kemudian setelah kartu masuk kedalam mesin atm

pengguna kartu diminta untuk memasukkan pin biasanya sejumlah 6 karakter (tergantung kebijakan bank bersangkutan) setelah itu mesin atm akan mengirimkan data kepada server pusat untuk melakukan validasi terhadap kartu atm dan juga pin yang telah dimasukkan oleh pemegang kartu, ketika data yang dilakukan validasi ternyata valid maka transaksi akan bisa di lanjutkan namun pada saat mesin server pusat menyatakan bahwa data kartu dan pin yang di masukkan tidak sesuai dengan *database* yang telah disimpannya maka mesin atm akan memberikan kesempatan sebanyak 2 kali kepada pemegang kartu untuk mengulanginya, setelah dinyatakan 3 kali mesin atm gagal melakukan validasi maka transaksi akan dihentikan dan kartu atm akan di blokir, sejauh ini kriptografi sudah mengambil peran penting dalam proses transaksi yang dilakukan pada mesin atm, dimana salah satu algoritma kriptografi telah berperan dalam melakukan enkripsi data pada saat terjadi komunikasi antara mesin atm yang digunakan untuk transaksi dan juga komputer server pusat yang melakukan validasi dan menyimpan semua data user (nasabah bank terkait) kriptografi berperan untuk melakukan pengacakan data baik yang dikirim maupun diterima antara mesin atm pusat dan mesin atm yang digunakan untuk transaksi sehingga keamanan pada saat transaksi akan lebih terjamin karena lalu lintas data akan di enkripsi sebelum dikirimkan.

Kriptografi yang diterapkan pada proses diatas mungkin bisa menjaga keamanan data pada saat proses transaksi berjalan, namun bagaimana cara melakukan perlindungan pada bagian kartu atmnya itu sendiri, kartu atm disebut juga sebagai *Magnetic Stripe Card* merupakan kartu yang digunakan sebagai media penyimpanan data-data dengan melakukan modifikasi serpihan partikel-partikel pada kartu dengan lempengan magnetic yang tipis

Gambar 11. *Magnetic Stripe Card*

Selain digunakan sebagai kartu atm, *magnetic card* juga biasanya digunakan untuk kartu identitas maupun kartu tiket transportasi cara kerja yang digunakan adalah strip magnetik pada dasarnya adalah deretan magnet kecil data yang dimasukkan dikodekan ke media dengan mengatur polaritas magnet tersebut, kemudian magnetic card digesekkan ke sebuah alat yang biasanya disebut *swipe reader* hingga melewati *stasioner reading head*, alat ini akan berkerja dengan baik untuk melakukan pembacaan pada *magnetic stripe card* jika kartu di gesekkan dengan tidak terlalu cepat dan juga tidak terlalu lambat

Gambar 12. *Swipe Reader*

Tipe lain reader lainnya adalah dengan memasukkan kartu. Biasanya, kartu 'ditelan' oleh reader dan dibaca langsung dengan baik melewati stasioner head reader

Gambar 13. *Model Lain Swipe Reader*

Dari berbagai jenis *swipe reader* di atas fungsi utamanya adalah melakukan pembacaan data yang berada pada *magnetic card* melalui gesekan sebelumnya, disinilah suatu proses kriptografi perlu diterapkan pada sisi keamanan user yang diaplikasikan langsung pada *magnetic card* sehingga bukan hanya proses komunikasinya saja yang dilakukan enkripsi namun pada sisi kartu atm juga di aplikasikan sebuah algoritma kriptografi.

2.3 Proses Enkripsi Pada Kartu ATM

Kartu atm atau disebut juga *magnetic card* memiliki beberapa nomer unik di antaranya biasanya tertera pada bagian depan kartu dan juga bagian belakang kartu selain itu dalam kartu juga masih menyimpan beberapa digit angka yang tersimpan pada bagian magnetik kartu sehingga ketika kartu di gesekkan pada sembarang media *swipe rider* akan terbaca langsung kode yang tercantum di dalamnya untuk mengantisipasi hal ini dibutuhkan sebuah program yang dapat melakukan enkripsi pada nomer yang akan di simpan pada *magnetic card* sehingga nantinya sebuah kartu atm akan menyimpan hasil kode enkripsi tersebut dan ketika di gesekkan pada sembarang magnetic card maka hasil yang terbaca sudah berupa *cipher text* bukan lagi deretan angka-angka text asli (*plain text*).

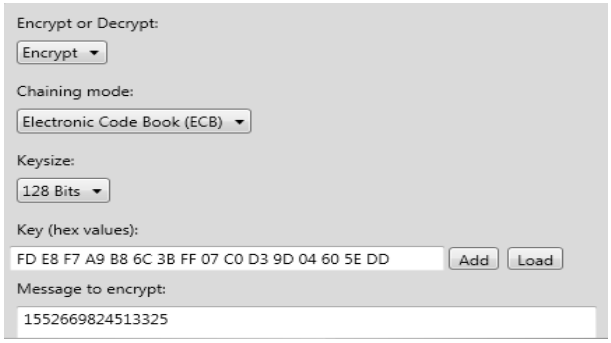
Pada saat proses transaksi maka mesin atm harus mampu melakukan dekrip terhadap kode *ciphertext* tersebut untuk memastikan berapa deretan angka-angka yang sesungguhnya yang tersimpan pada kartu atm, ketika proses dekrip berhasil dan data-data yang ada di dalam kartu atm valid maka dapat melanjutkan ke proses

berikutnya untuk input password dan melakukan transaksi yang dikehendaki

Implementasi untuk melakukan enkripsi penulis menggunakan sebuah program bernama cryptotool.

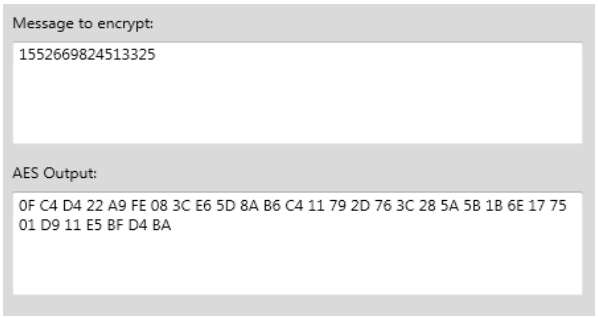
Gambar 14. *Program Cryptool*

Program ini dijalankan pada sistem operasi windows 7 dengan menggunakan ram 4GB kemudian penulis mencoba untuk melakukan percobaan dengan melakukan encrypt deretan angka yang dianalogikan akan di simpan pada kartu atm (*magnetic card*).



Gambar 15. *Proses Enkripsi Data*

Beberapa parameter di tentukan sebelum melakukan enkripsi seperti *KeySize, Key dan Chaining Mode* kemudian proses enkripsi dilakukan dengan menghabiskan waktu percobaan selama 3 detik.



Gambar 16. *Hasil Enkripsi Data*

3. Kesimpulan

Sistem keamanan pada kartu ATM dengan menggunakan algoritma AES(*Advanced Encryption Standard*).

Berdasarakan pembahasan di atas dapat diambil beberapa kesimpulan yaitu:

1. Penelitian ini menggunakan algoritma enkripsi AES 125 bit karena algoritma enkripsi AES memiliki 4 jenis transformasi, yaitu *SubBytes, ShiftRows, MixColumns dan AddRoundKey*. Dan mengulang hingga 10 round.
2. ATM yang biasanya disebut juga sebagai *Magnetic Stripe Card,* ketika kita masukkan pada mesin *swipe card* biasanya akan muncul angka (*plain text*). Disinilah pentingnya sebuah algoritma kriptografi di terapkan untuk mengubah bentuk angka(*plain text*), menjadi sebuah kode-kode (*chipper text*).
3. Aplikasi cryptool ini menggunakan struktur algoritma AES sehingga dapat membuat nomer kartu ATM dapat terenkripsi dengan struktur algoritma AES.
4. Dengan menggunakan aplikasi cryptool disini kita dapat melihat cara kerja enkripsi algoritma AES 128 bit bekerja dan dengan aplikasi ini dapat mengubah bentuk *plain text* menjadi *chiper text* serta metode yang di gunakan aplikasi ini adalah metode AES.

Daftar Pustaka

[1] Sadikin Rifky, *Kriptografi untuk Keamanan Jaringan,* Yogyakarta: Andi Offset, 2012.
[2] Nugraha M. Pasca, "Jurnal Kriptografi pada Kejahatan Pembobolan ATM di Indonesia" *Makalah IF3058 Kriptografi Tahun* 2010.
[3] Sambiangga Roni,"Sistem Keamaan ATM (Automated Teller Macine/Anjungan Tunai Mandiri)" *Makalah IF5054 semester 1 06/07.*
[4] Munir Rinaldi, "Slide kuliah IF3058 Kriptografi, Program Studi Teknik Informatika" *STEI ITB, 2010*
[5] Stalling William, "Cryptography and Network Security", *4th ed., Prentice Hall*

Biodata Penulis

Arief Agung Gumelar, mahasiswa aktif STMIK Amikom Yogyakarta 2014. Kelahiran Pekalongan, 01 November 1994.

Latief Adam Busyairi, mahasiswa aktif STMIK Amikom Yogyakarta 2014. Kelahiran Timor-timur, 22 November 1996.

Muhammad Fajrian Noor, mahasiswa aktif STMIK Amikom Yogyakarta 2014. Kelahiran Amuntai, 09 November 1996.