Jurnal Nasional Informatika dan Teknologi Jaringan
E-ISSN : 2540-7600
P-ISSN : 2540-7597

ANALISIS KRIPTO SISTEM ALGORITMA AES DAN ELLIPTIC CURVE CRYPTOGRAPHY (ECC) UNTUK KEAMANAN DATA

Edy Budi Harjono Sibarani.M.Kom[1],Prof.Dr.Muhammad Zarlis[2], Rahmat Widya Sembiring,M, PhD[3]
[12]*Universitas Sumatera Utara*
Jl. Universitas no.123. 2O140 Medan
edybudi@gmail.com

Abstrak—Kriptografi merupakan salah satu solusi atau metode pengamanan data yang tepat untuk menjaga kerahasiaan dan keaslian data, serta dapat meningkatkan aspek keamanan suatu data atau informasi. Metode ini bertujuan agar informasi yang bersifat rahasia dan dikirim melalui suatu jaringan, seperti LAN atau Internet, tidak dapat diketahui atau dimanfaatkan oleh orang atau pihak yang tidak berkepentingan. Kriptografi mendukung kebutuhan dua aspek keamanan informasi, yaitu perlindungan terhadap kerahasiaan data informasi dan perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan. AES-Rinjdael merupakan salah satu algoritma kriptografi yang digunakan dalam mengamankan pesan menggunakan panjang kunci sampai 256 bit, yang mana untuk meghindari kriptanalisis, maka dilakukan metode kombinasi dengan algoritma Eliptic Curve Criptografi (ECC) dalam pengenkripsian pesan.

Keywords— KriptografI, ECC, AES-RINJDAEL.

I. PENDAHULUAN

Keamanan data merupakan salah satu hal penting dalam pertukaran data, khusunya pertukaran data didunia maya yang didalamnya terdapat banyak ancaman untuk proses itu sendiri. Bagi suatu organisasi keamanan data bernilai sangat rahasia (private and confidential). Sama halnya dengan dokumen konvensional, dokumen dalam format digital pun membutuhkan aspek keamanan[1]. Teknik enkripsi untuk pengamanan pesan ada dua yaitu teknik enkripsi asimetris dan teknik enkripsi simteris. Pengamanan pesan dengan menggunakan teknik enkripsi simetri sudah banyak dilakukan, misalnya dengan menggunakan teknik enkripsi simteris dengan menggunakan algoritma RC6, kekurangan dari algoritma RC6 ini adalah pesan menjadi lebih besar karena harus bekerja pada 8 bit dan dibutuhkan padding untuk memenuhi panjang blok. Algoritma enkripsi simetris lainnya adalah AES. AES adalah salah satu teknik enkripsi simetris yang popular dan banyak digunakan dalam berbagai aplikasi. AES terdiri dari 128 bit, 192 dan 256 bit. AES dapat dimplementasikan pada perangkat mobile.

II. TINJAUAN PUSTAKA

Keamanan data dalam penggunaan komputer tidak hanya tergantung dari sebuah teknologi, melainkan bagian dari prosedur atau suatu kebijakan keamanan yang digunakan serta cerdas dalam pemilihan sumber daya manusia. Jika *firewall* dan perangkat keamanan lainnya bisa dibobol oleh individu yang tidak memiliki hak, maka peran utama dari kriptografi untuk mengamankan data atau dokumen dengan menggunakan teknik enkripsi sehingga data atau dokumen tidak bisa terbaca[2].

kriptografi adalah suatu studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperi kerahasiaan, integritas data, otentikasi entitas dan otentikasi keaslian data. Kriptografi tidak hanya berarti penyediaan keamanan informasi, melainkan sebuah himpunan taknik-teknik. (Pitchaiah, 2012). Untuk perbandingan dengan algoritma simetris lainnya dalam hal performa AES masih kalah dengan Blowfish akan tetapi lebih aman. Sedangkan dengan DES, AES menang penuh dari segi keamanan dan segi kecepatan AES juga lebih unggul dari DES[3].

Selain teknik enkripsi simetris, pengamanan data bisa juga dilakukan dengan menggunakan teknik enkripsi asimetris yaitu algoritma RSA dan IBC, penggunaan algoritma IBC diimplementasikan bersamaan dengan algoritma RSA dikarenakan adanya kelemahan dari algoritma IBC yang kinerjanya sangat lambat dan tidak cocok untuk perangkat yang memilki resource terbatas sehingga diperlukan algoritma RSA untuk menutupi kelemahan tersebut. RSA dan IBC adalah salah satu algoritma enkripsi kunci publik[4].

Algoritma enkripsi kunci publik digunakan untuk menutupi kelemahan dari enkripsi kunci simteris. Kelemahan enkripsi kunci simteris terletak pada distribusi kunci dan sangat susah dalam manajemen kuncinya [1].

Dari beberapa penelitian yang telah dilaksanakan, maka pada penelitian ini penulis akan membahas bagaimana mengamankan sebuah data dengan menggunakan algoritma kriptografi simetris yang cukup aman dan algoritma kriptografi asimetris yang dapat digunakan pada resource yang terbatas namun memiliki tingkat keamanan yang tinggi.

III. LANDASAN TEORI

A. Algoritma AES (Advanced Encryotion Standart)

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunakan untuk mengamankan data. Algoritma *AES* adalah blok chipertext simetri yang dapat mengenkripsi *(encipher)* dan dekripsi *(decipher)* informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut ciphertext; sebaliknya dekripsi adalah merubah ciphertext data menjadi bentuk semula yang kita kenal sebagai plaintext. Algoritma *AES* mengunkan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekrip data pada blok 128 bits. Kriteria pemilihan *AES* didasarkan pada 3 kriteria utama yaitu : keamanan, harga, dan karakteristik algoritma beserta implementasinya.

B. Parameter Algoritma AES

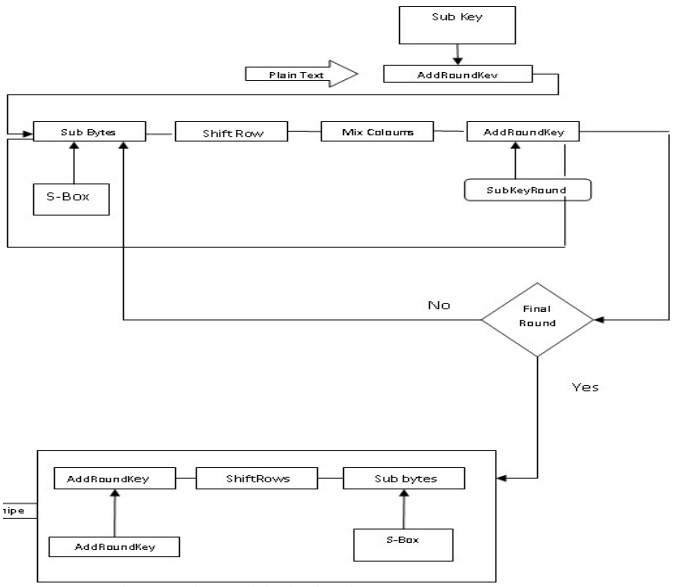
Pada algoritma kriptografi Advanced Encryption Standart (AES), terdiri atas 128 bit, 192 bit dan 256 bit dimana panjang kunci akan menentukan jumlah putaran total kunci. AES-128 bit menggunakan panjang kunci Nk = 4 word (kata) yang mana setiap kata terdiri dari 32 bit sehingga menghasilkan total kunci 128 bit, ukuran blok teks asli 128 bit dan memiliki 10 putaran. Untuk putaran kunci terdiri dari Ki = 4 kata dan total putaran kunci 128 bit dan memiliki ukuran kunci yang diperluas 44 kata dan 176 byte. Dalam beberapa kasus, blok ini juga dianggap sebagai *array* satu dimensi dari vektor 4-*byte*, di mana setiap vektor terdiri dari kolom yang sesuai dalam representasi *array* dua dimensi. Array ini memiliki panjang masing-masing 4, 6, atau 8, dan indeks dalam rentang 0..3, 0..5, atau 0..7. Vektor 4-*byte* ini disebut dengan *word*.

TABEL I
PARAMETER AES[5]

	AES 128	**AES 192**	**AES 256**
Key size	4 word (16 byte)	6 word (24 byte)	8 word (32 byte)
Plaintext blok size	4 word (16 byte)	4 word (16 byte)	4 word (16 byte)
Number of round	10	12	14
Round key size	4 word (16 byte)	4 word (16 byte)	4 word (16 byte)
Expanded key size	44 word (176 byte)	52 word (208 byte)	60 word (240 byte)

C. Enkripsi algortma AES

Enkrpsi pada algorima aes terdri dari 4 jenis transformasi bytes, yaitu *SubBytes*, *ShitfRows*, *MixColom* dan *AddRoundKey*. Pada awal proses enkripsi, input yang dikopikan kedalam state akan mengalami transformasi *Byte AddRoundKey*. Setelah itu state akan mengalami tanformasi *SubBytes*, *ShiftRow*, *MixColoums* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam AES disebut sebagai *Round Function*. Round yang terakhir berbeda dengan Round-Round sebelumnya dimana pada Round terakhir state tidak mengalami *MixColoums*.



Gbr. 1 : Alur Enkripsi Algoritma AES

D. AddRoundKey

Pada proses enkripsi dan dekripsi AES proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada *state* dengan operasi XOR. Setiap *round key* terdiri dari Nb *word* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*Nb+c}] \text{untuk } 0 \le c \le Nb$$

[wi] adalah *word* dari *key* yang bersesuaian dimana i = round*Nb+c. Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada *round* = 0 untuk *round* selanjutnya *round* = *round* + 1, pada proses dekripsi pertama kali pada *round* = 14 untuk *round* selanjutnya *round* = *round* - 1.

E. Transformasi SubBytes

SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Proses Subbytes adalah operasi yang melakukan susbtitusi tidak liniar dengan cara menggantis etiap byte state dengan byte pada tabel S-Box. Sebuah tabel S-Box terdiri dari 16 X 16 baris dan kolom dengan masing-masing ukuran 1 byte. Tabel S-Box dapat dilihat pada tabel II.

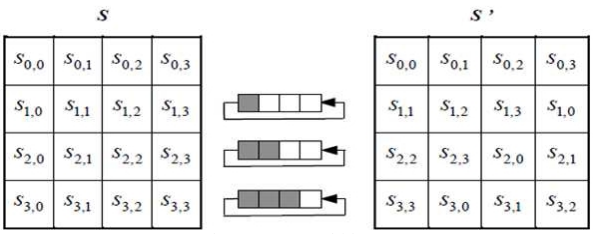
TABEL II

S-BOX SUBBYTES [5]

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
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
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

F. Shift Rows

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran *bit* dimana *bit* palingkiri akan dipindahkan menjadi *bit* paling kanan (rotasi *bit*). penggeseran baris ke-i pada *state* ke arah kanan sejauh i. Proses pergeseran *Shiftrow* ditunjukkan dalam Gambar 2 berikut:



Gbr. 2 Proses Shiftrows

G. MIX Coloum

Transformasi *Mix columns* merupakan operasi yang beroperasi terhadap setiap kolom pada *state*. *Mix column* merupakan perkalian matriks pada sebuah kolom. Perkalian tersebut merupakan perkalian pada Persamaan 3. Di sini, kolom-kolom pada array state akan diperlukan sebagai suatu polynomial yang berada dalam GF(28) dan akan dikalikan dengan modulo x4 + 1, dengan suatu polynomial tertentu : a(x) = {03}x3 + {01}x2 + {01}x + {02}. Transformasi MixColoums dapat dilihat pada perkalian matriks berikut:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Contoh: Misalkan *state* seperti berikut, dan akan dilakukan transformasi *mixColumns*,

D4	E0	B8	1E
BF	B4	41	27
5D	52	11	98
30	AE	F1	E5

·

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

=

04	E0	48	28
66	CB	F8	06
81	19	D3	26
E5	9A	7A	4C

Proses penambahan pada operasi ini berarti melakukan operasi Bitwise XOR maka, hasil perkalian matriks diatas dapat dianggap seperti perkalian dibawah ini

$$S0=(D4\cdot02)\oplus(B\ \cdot03)\oplus(5D\cdot01)\oplus(30\cdot01)=04$$

$$S1=(D4\cdot01)\oplus(B\ \cdot02)\oplus(5D\cdot03)\oplus(30\cdot01)=66$$

$$S2=(D4\cdot01)\oplus(B\ \cdot01)\oplus(5D\cdot02)\oplus(30\cdot03)=81$$

$$S3=(D4\cdot03)\oplus(B\ \cdot01)\oplus(5D\cdot01)\oplus(30\cdot02)=5E$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$S15=(1E\cdot03)\oplus(27\cdot01)\oplus(98\cdot01)\oplus(E5\cdot02)=4C$$

H. Elliptic Curve Criptografi (ECC)

Eliptic Curve Cryptography merupakan sistem kriptografi kunci publik yang memanfaatkan persamaan kurva eliptik. Algoritma ini dirancang dan diajukan oleh Neal Koblitz dan Victor S. Miller. Jika dibandingkan dengan kriptografi kunci asimetrik lainnya, ECC dianggap lebih unggul. Penyebab utamanya adalah karena dengan menggunakan kunci yang jauh lebih kecil atau pendek, ECC tetap dapat memberikan tingkat keamanan yang sama dengan algoritma asimetrik lainnya yang menggunakan kunci yang lebih besar. Dengan ukuran kunci yang lebih kecil dan tingkat keamanan yang sama tinggi, implementasi ECC menjadi lebih efisien, Perbandingan ukuran kunci ECC dengan ukuran kunci asimetrik lainnya dapat lebih jelas terlihat pada tabel III.

TABEL III
PERBANDINGAN UKURAN KUNCI PUBLIK PADA KRIPTOGRAFI ASIMETRIK SUMBER: [6]

ECC Key size (bit)	RSA Key size (bit)	Key size ratio	AES Key size (bit)
163	1024	1:6	
256	3072	1:12	128
384	7680	1:20	192
512	15360	1:30	256

H. Operasi Pada Kurva Eliptik

Terdapat beberapa operasi pada Kurva Eliptik dalam Fp, yaitu:

a. Penambahan (*Addition*)

Berikut ini adalah sifat-sifat operasi penambahan pada kurva eliptik :

Jika ϕ adalah *point at infinity*, maka penjumlahan dua buah ϕ akan menghasilkan ϕ pula. Secara matematis dapat dituliskan sebagai berikut :

$$\phi + \phi = \phi.$$

Jika P = (x1,y1) ∈ E(Fp), maka :

$$P + \phi = \phi + P = P$$

Jika P = (x1,y1) ∈ E(Fp), maka negasi P,–P = (x1,-y1) juga merupakan *point* pada kurva. Penambahan dua buah *point* ini memberikan :

P + (-P) = P – P = ϕ
Jika P = (x1,y1) ∈ E(Fp) dan Q = (x2,y2) ∈ E(Fp) dengan x1 ≠ x2, dan Q ≠ -P, maka jumlah mereka : P + Q = R = (x3,y3)
didefinisikan sebagai :
x3 = λ2 – x1 – x2
y3 = λ(x1-x3) –y1
dengan
λ = y2 – y1
x2-x1

Sifat penting lain dari aturan penambahan ini adalah bahwa urutan penambahan tidak penting, P + Q = (x1,y1) + (x2,y2) = (x3,y3) = (x2,y2) + (x1,y1) = Q + P.

b. Penggandaan (*Doubling*)

Jika P = (x1,y1) ∈ E(Fp) adalah suatu *point* dengan y1 ≠ 0 dan P bukan ϕ, maka 2P = (x3,y3), dengan :

λ = (3x1
2 + a)
2y
x3 = λ2 - 2x1, dan
y3 = -y1 + λ (x1-x3)
Jika P=ϕ atau y1=0, maka 2P=ϕ.

c. Perkalian Skalar (*Scalar Multiplication*)

Point kurva eliptik tidak dapat dikalikan, namun dapat dilakukan *scalar multiplication*, yaitu penambahan berulang untuk *point* yang sama. Jika n adalah suatu integer positif dan P suatu point pada kurva eliptik, perkalian skalar nP adalah hasil penambahan P sejumlah n kali. Sehingga,

5P = P+P+P+P+P.

Perkalian skalar ini dapat diperluas untuk integer nol dan negatif yaitu :

0P = ϕ, (-n) P = n (-P)

I. Enkripsi ECC

(Muller & Paulus, 1998) Dalam proses enkripsi pertama-tama dilakukan pembacaan suatu berkas publik yang berisi Kurva Eliptik E, suatu point P yang berada pada E, suatu bilangan prima p ∈ f_p dan kunci publik pemakai lain Q = d*p. Kemudian dipilih suatu bilangan random k ∈ {2,..,p-1} Yang berubah untuk setiap blok data dan dihitung k*Q dan k*P, selanjutnya berkas data dibaca secara perblok (M) dan dienkripsi dengan menggunakan pesamaan (9), dimana $a,b \in Z_p$ dan $4a^3 + 27b^2 \neq 0 \pmod p$, dan sebuah titik ***O*** yang disebut dengan titik *infinity*. Himpunan $E(Z_p)$ adalah semua titik (x,y), untuk x,y $\in Z_p$, yang memenuhi persamaan (9) pada titik ***O***.

M' = [K*P, M ⊕ X (K*Q)]

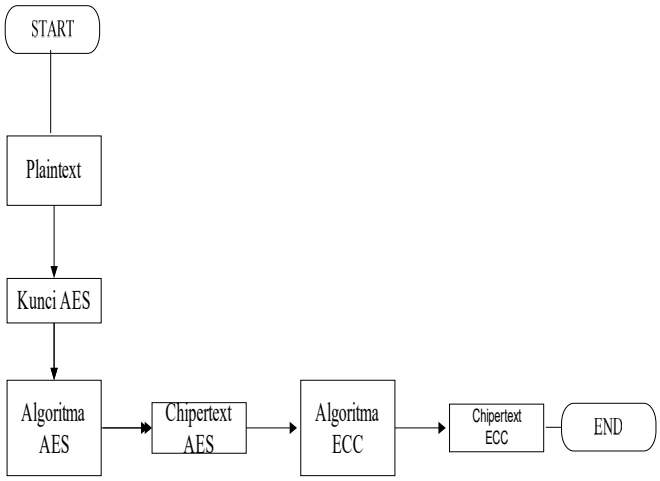
M akan dilakukan operasi logika XOR dengan k*Q, hasilnya berupa string yang lalu ditulis ke berkas dengan k*P ditambahkan sebelumnya. Hasil akhirnya secara sederhana dapat dituliskan sebagai berikut:

M1 M2
K*P M ⊕ X(k*d*P)
M'

IV. METODOLOGI

A. Rancangan Penelitian

Adapun teknik rancangan dalam sistem keamanan data yaitu dengan mengkombinasikan metode algoritma AES-Rinjdael dan metode algoritma Eliptic Curve Criptography (ECC), dimana pada kedua metode tersebut memiki tahap-tahap seperti tahap pembangkitan kunci, tahap enkripsi dan tahap deskripsi. Adapun skema alur proses dalam perancangan sistem yang dibentuk dapat dilihat pada gambar 3.



Gbr. 3 Alur Proses Enkripsi data (file)

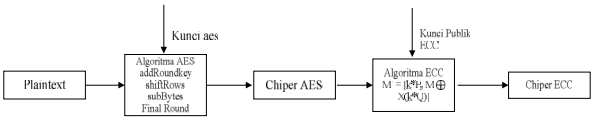
Berdasarkan gambar 3. langkah-langkah rancangan sistem yang akan dilakukan adalah sebagai berikut :

a. Inputlah sebuah file teks yang akan dienkripsi
b. Tentukan Kunci untuk algoritma AES Kemudian file teks tersebut akan dienkripsi menggunkan algoritma AES dan akan meghasilkan cipherteks AES.
c. Bangkitkan kunci publik algoritma ECC, dari kunci *private.*
d. Kemudian file chipertext AES dienkripsi kembali dengan menggunkan algoritma ECC, dan akan menghasilkan cipherteks ECC.

B. Alur Proses Enkripsi data (file)

Pada proses enkripsi data, diambil sebuah plaintext, karakter tersebut pertama kali dienkripsi dengan menggunkan metode algoritma AES, dengan menggunakan kunci private AES, dimana kunci yang digunakan memiliki jumlah panjang kunci 16 byte, kemudian dilakukan proses enkripsi dengan menggunakan rumus yang ada dari perhitungan tersebut didapatkan sebuah chipertext1. Cipherteks1 yang dihasilkan dari proses enkripsi algoritma AES ini kemudian akan di enkripsikan kembali menggunakan algoritma Elliptic Curve Cryptogaraphy (ECC), maka

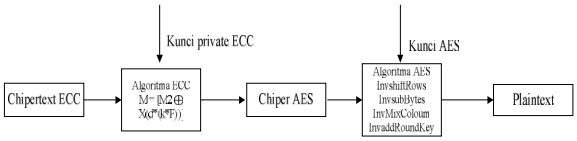
menghasilkan cipherteks2. Alur proses enkripsi ini dapat dilihat pada gambar 4.



Gbr 4. Enkripsi Plaintext Menggunakan Metode AES

C. Alur Proses Dekripsi data

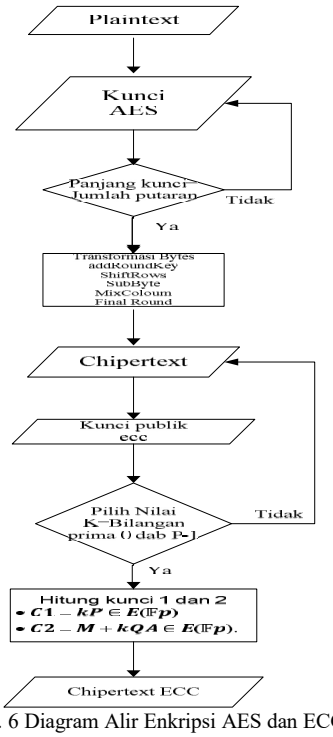
Pada proses dekripsi data, chipertext akan didekripsikan menggunakan algoritma *Eliptic Curve Cryptography* (ECC) dengan menggunakan kunci *private* yang mana nantinya akan menghasilkan chipertext dari hasil enkripsi algoritma AES. Selanjutnya *chipertext* kembali akan dienkripsi dengan menggunakan algoritma enkripsi yang pertama yaitu AES dan akan menghasilkan data semula (*plaintext*). Alur proses dekripsi data ini dapat dilihat pada gambar 5.



Gbr 5. Proses Dekripsi Data

D. Diagram Alir Proses Enkripsi

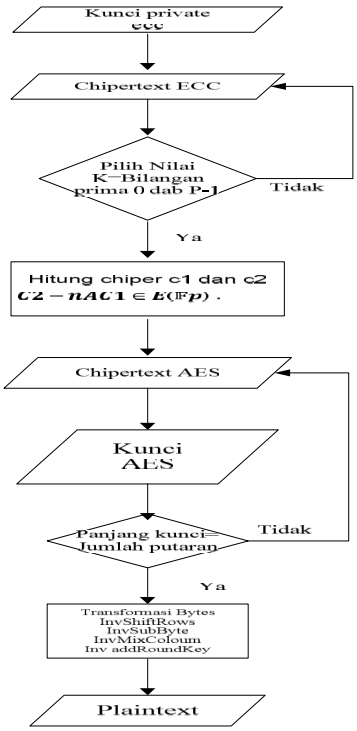
Adapun diagram alir untuk proses kombinasi enkripsi chipertext AES menggunakan algoritma Elliptic Curve Cryptography (ECC) dapat dilihat pada gambar 6.



Gbr. 6 Diagram Alir Enkripsi AES dan ECC

E. Diagram Alir Proses Dekripsi

Adapun diagram alir proses dekripsi data chipertext dengan menggunakan algoritma Eliptic Curve Cryptography (ECC) dapat dilihat pada gambar 7.



Gbr. 7 Diagram alir proses dekripsi algoritma ECC dan AES

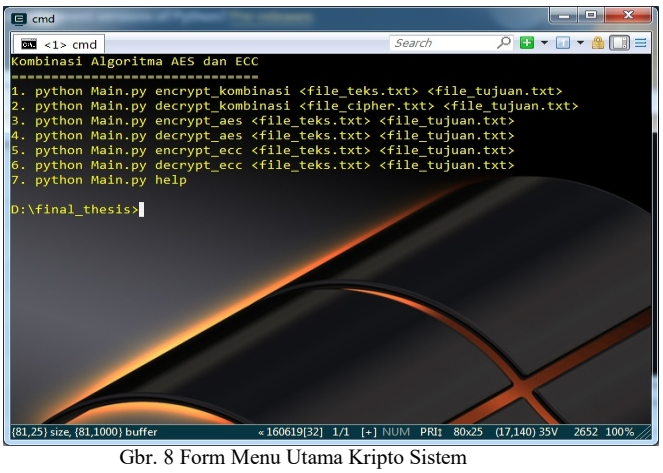
V. IMPLEMENTASI

A. Proses Enkripsi Dekripsi Algoritma AES

Untuk mempermudah implementasi dari proses penelitian ini digunakan program java versi 2.7 dengan sistem operasi yang digunakan windows 7 ultimate. Adapun hasil uji coba ini menggunakan beberapa sampel data yang digunakan pada tabel 4, dan pada gambar 8 ditunjukkan sebagai form menu utama dalam proses enkripsi menggunakan AES, ECC dan kombinasi dari kedua algoritma.

TABEL IV
SAMPEL DATA ENKRIPSI

No.	Nama File	Jenis File	Kapasitas
1	Coba	text	1 Kb
2	Test1	html	114 Bytes
3	Atom	doc	22 Kb
4	Coba1	pas	8 Kb
5	Crack	bas	29 Kb



Gbr. 8 Form Menu Utama Kripto Sistem

B. Proses Pengujian Enkripsi Algoritma AES

Pengujian enkripsi tahap awal akan dilakukan dengan menggunakan algoritma AES dengan blok masukan dan kode kunci memiliki panjang 16 byte.

Contoh penerapan :
Plain Teks : magisterusu2016!
In Hex : 6d 61 67 69 73 74 65 72 75 73 75 32 30 31 36 21
Kode Kunci : fasilkomtiusu008
In Hex : 66 61 73 69 6c 6b 6f 6d 74 69 75 73 75 30 30 38
Dengan menggunakan algoritma perhitungan pseudocode dibawah ini

Cipher (byte in [4*Nb], Byte out [4*Nb], word w [Nb*(Nr+1)])
Begin
    Byte state [4,Nb]
    State = in
    AddRoundKey (state, w)

    For round = 1 step 1 to Nr-1
        SubBytes x (state
        ShiftRows (state)
        MixColoum (state)
        AddRoundKey (state,w+round*Nb)
    End for
    SubBytes x (state)
    ShiftRows (state)
    AddRoundKey (state, w +Nr*Nb)
    Out = State
End

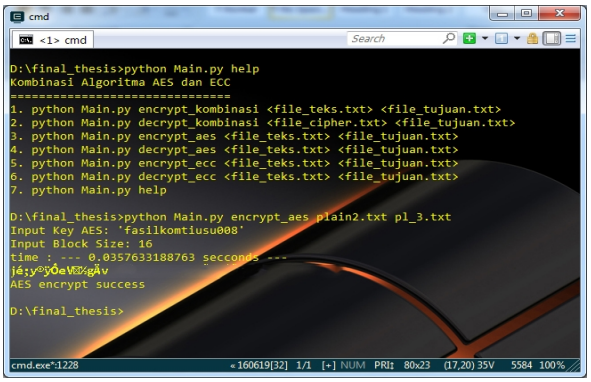
Nilai dari putaran kunci :
W [0] = 66 61 73 69
W [1] = 6c 6b 6f 6d
W [2] = 74 69 75 73
W [3] = 75 30 30 38

Sehingga kode kunci menjadi 66 61 73 69 6c 6b 6f 6d 74 69 75 73 75 30 30 38, nilai lanjut dari state dapat dilihat pada tabel 5. dibawah ini.

TABEL V
PROSES ENKRIPSI AES 128 BIT

	SubBytes	ShiftRows	MixColumns	AddRoundKey	Key Schedule
R 0				0B 1F 01 45 00 1F 1A 01 14 0A 00 06 00 1F 41 79	66 6C 74 75 61 6B 69 30 73 6F 75 30 69 6D 73 58
R 1	2B C0 7C 6E 63 C0 A2 7C FA 67 63 6F 63 C0 83 86	2B C0 7C 6E C0 A2 7C 63 63 6F FA 67 86 63 C0 83	D8 6A 46 9D A3 4D 51 82 EC 19 B4 5D A9 50 99 AB	BB 65 3D 93 C6 43 36 D5 F5 6F B7 6E 5D C9 73 19	63 0F 78 0E 65 0E 67 57 19 76 03 33 F4 99 EA B2
R 2	EA 4D 27 DC B4 1A 05 03 E6 A8 A9 9F 4C DD 8F D4	EA 4D 27 DC 1A 05 03 B4 A9 9F E6 A8 D4 4C DD 8F	9C 46 70 43 EA B1 CD C3 DE B9 8F A9 25 D5 2D 66	A6 73 3E 03 4C 19 02 5B F0 E1 D4 C1 7A 13 01 F8	3A 35 4E 40 A6 AB CF 98 2E 58 5B 68 5F C6 2C 9E
R 3	24 8F B2 7B 29 D4 77 39 8C F8 48 78 DA 7D 7C 41	24 8F B2 7B D4 77 39 29 48 78 8C F8 41 DA 7D 7C	26 3E C5 09 0E 33 32 46 A3 7D 0F 3D 72 2A B2 A4	5E 73 C6 4A ED 78 B6 5A B6 00 29 73 24 BA 3E 86	78 4D 03 43 E3 4B B4 1C 25 7D 26 4E 56 90 BC 22
R 4	58 8F B4 D6 55 BC 4E BE 44 63 A5 8F 36 F4 B2 44	58 8F B4 D6 BC 4E BE 55 A5 8F 44 63 44 36 F4 B2	8E 6E 1A 99 BB AF EB 6B 79 9E 85 88 79 27 CE 28	62 CF B8 78 47 28 E8 74 CF 55 68 2B 35 FB AE 6A	EC A1 A2 E1 CC B7 03 1F B6 CB ED A3 4C DC 60 42
R 5	AA 8A 6C BC A0 34 9B 92 8A FC 45 F1 96 0F E4 02	AA 8A 6C BC 34 9B 92 A0 45 F1 8A FC 02 96 0F E4	54 DE F0 80 C9 78 9B 41 12 49 ED CB 90 D8 B2 50	68 43 CF 5E C6 41 42 5D 88 18 5C D7 24 B0 BA 1A	3C 9D 3F DE B4 68 08 4A
R 6	45 1A 8A 58 DD BC 14 83 C4 AD 4A 0E 36 E7 F4 A2	45 1A 8A 58 BC 14 83 DD 4A 0E C4 AD A2 36 E7 F4	BD 30 B2 95 5A 16 27 E1 90 4B AB C3 66 58 17 6B	ED FD 40 B9 5C 51 22 B9 DC 55 09 7D CF 99 DE EB	50 CD F2 2C 06 47 05 58 4C 1D A1 BE A9 C1 C9 83
R 7	55 54 09 56 4A D1 93 56 86 FC 01 FF 8A EE 1D 9B	55 54 09 56 D1 93 56 4A 01 FF 86 FC 8A EE 1D 9B	58 73 80 93 74 F9 DA C0 30 A7 61 D8 02 9F 0C 76	22 C4 C5 FA DC 16 30 72 90 1A 7D 7A DA B6 DC 25	7A B7 45 69 A8 EF EA B2 A0 BD 1C A2 D8 19 D0 53
R 8	93 1C A6 2D 86 47 04 40 60 A2 FF DA 57 44 86 3F	93 1C A6 2D 47 04 40 86 FF DA 60 A2 3F 57 44 86	34 B9 B3 EF 3B 36 C2 41 70 4E EA 65 68 54 59 44	F9 C3 8C B9 AA 4B 55 64 3D BE 06 2B 49 6C B1 FF	CD 7A 3F 56 92 7D 97 25 4D F0 EC 4E 21 38 E8 B8
R 9	99 2E 64 56 AC B5 FC 43 27 AE 6F F1 3B 50 C8 16	99 2E 64 56 B5 FC 43 AC 6F F1 27 AE 16 3B 50 C8	9E 89 7A 25 45 FE D8 34 CE 66 99 FE 40 09 68 73	77 1A D6 DF FE 3E 8C 46 69 31 22 0B D0 A1 28 88	E9 93 AC FA BD C0 57 72 A7 57 BB F5 90 A8 40 F8
R 10	F5 A2 F6 9E B6 B2 64 5A F9 C7 93 2B 70 32 34 C4	F5 A2 F6 9E B2 64 5A B6 93 2B F9 C7 C4 70 32 34		***6A AE 56 C4*** ***E9 FF 96 05*** ***3B D4 BD 76*** ***79 65 67 9A***	9F 0C A0 5A BE CC BE A8 FF 44 B1 BD 15 55 AE

Dari proses diatas maka diperoleh chipertext 6A E9 3B 79 AE FF D4 65 56 96 BD 67 C4 05 76 9A. Proses simulasi dapat dilihat pada gambar 9.



Gbr. 9 Enkripsi data dengan AES

Hasil dari enkripsi data teks “magisterusu2016!” dengan kunci “fasilkomtiusu008” adalah “jé;y®ÿÔeV½gÄv”, dan waktu yang dibutuhkan untuk proses enkripsi yaitu “0.0357”, detik.

C. Proses Dekripsi Algoritma AES

Proses dekripsi, untuk mendekripsi hasil dari penerapan diatas dengan menginversikan cipherteks yang didapat.

Contoh penerapan :
Chiperteks : 6A E9 3B 79 AE FF D4 65 56 96 BD 67 C4 05 76 9A
Char : jé;y®ÿÔeV½gÄvš
Kode Kunci : fasilkomtiusu008
In Hex : 66 61 73 69 6c 6b 6f 6d 74 69 75 73 75 30 30 38

Dengan menggunakan algoritma perhitungan pseudocode untuk kode inversi dibawah ini.

```
EqInvCipher (byte in [4*Nb], Byte out [4*Nb], word dw [Nb*(Nr+1)] )
Begin
        Byte state [4,Nb]
        State = in
        AddRoundKey (state, dw + Nr * Nb)
        For round = Nr - 1 step  - 1 to 1
                InvSubBytes x (state)
                InvShiftRows (state)
                InvMixColoum (state)
                AddRoundKey (state, dw  +  round * Nb)
        End for
        InvSubBytes x (state)
        InvShiftRows (state)
        AddRoundKey (state, dw)
        Out = State
End
```

Kode kunci yang digunakan untuk proses dekripsi sama dengan enkripsi. Nilai dari putaran kunci :

W [0] = 66 61 73 69
W [1] = 6c 6b 6f 6d
W [2] = 74 69 75 73
W [3] = 75 30 30 38

Sehingga kode kunci menjadi 66 61 73 69 6c 6b 6f 6d 74 69 75 73 75 30 30 38, nilai lanjut dari state dapat dilihat pada tabel 6. dibawah ini.

TABEL VI
PROSES DEKRIPSI AES 128 BIT

	SubBytes	ShiftRows	MixColumns	AddRoundKey	Key Schedule
R 0				60 73 75 30 61 74 73 31 67 65 75 36 69 72 32 21	66 6C 74 75 61 6B 69 30 73 6F 75 30 69 6D 73 58
R 1	08 1F 01 45 00 1F 1A 01 14 0A 00 06 00 1F 41 79	2B C0 7C 6E 63 C0 A2 7C FA 67 63 6F 63 C0 83 86	2B C0 7C 6E C0 A2 7C 63 63 6F FA 67 B6 63 C0 83	D8 6A 46 9D A3 4D 51 82 EC 19 B4 5D A9 50 99 AB	63 0F 7B 0E 65 0E 67 57 19 76 03 33 F4 99 EA B2
R 2	BB 65 3D 93 C6 43 36 D5 F5 6F B7 6E 5D C9 73 19	EA 4D 27 DC 1A 05 03 B4 B6 AB A9 9F 4C DD 8F D4	EA 4D 27 DC 1A 05 03 B4 A9 9F B6 AB D4 4C DD 8F	9C 46 70 43 EA B1 CD C3 DE B9 8F A9 25 D5 2D 66	3A 35 4E 40 A6 AB CF 98 2E 58 5B 6B 5F C6 2C 9E
R 3	A6 75 38 05 4C 19 02 5B F0 81 D4 C1 7A 13 01 FE	24 8F B2 75 29 D4 77 39 BC FE 46 78 DA 7D 7C 41	24 8F B2 75 D4 77 39 29 46 78 BC FE 41 DA 7D 7C	26 3E C9 09 D6 33 B2 46 A5 7D DF 5D 72 2A B2 A4	7B 4D D5 43 E5 45 E4 1C 25 7D 26 4E 56 9D BC 22
R 4	3E 73 C6 4A ED 76 B6 5A B6 DD 29 73 24 BA 3E B6	3E 8F B4 D6 BC 4E B6 55 44 63 A5 8F B6 F4 B2 44	3E 8F B4 D6 BC 4E B6 55 A5 8F 44 63 44 B6 F4 B2	8E 6E 1A B9 B6 A9 B6 B6 79 9E E5 B6 90 D6 B2 50	BC A1 A2 E1 CC B7 D5 1F B6 C6 ED A5 B4 66 D6 4A
R 5	B2 CF B6 7B 47 2E 6E 74 CF 55 6B 2E 35 F6 A2 BA	AA BA 6C BC AD B4 9E 92 BA FC 4E F1 9E CF B4 D2	AA BA 6C BC B4 9E 92 AD 4E F1 BA FC D2 9E CF B4	B4 D6 F0 BD CF 39 D9 1C 12 49 BD CB 9D D6 B2 50	3C 9D 3F D6 C6 41 42 BD 9A 51 BC 1F B4 6B D6 4A
R 6	6B 43 CF 5E C9 7B 9B 41 B6 16 BC D7 24 BD BA 1A	4E 1A BA 5B DD BC 14 B3 C4 AD 4A D6 B6 E7 F4 A2	4E 1A BA 5B BC 14 B3 DD 4A D6 C4 AD A2 B6 E7 F4	BD 3D B2 95 BA 16 27 E1 9D 4E A6 C3 6B 16 17 6B	5D CD F2 2C D6 47 D5 B6 4C 1D A1 B6 A9 C1 C9 B3
R 7	BD FD 4D 69 BC B1 22 B9 DC B5 09 7D CF 99 D6 B6	B5 B4 09 B6 4A D1 B3 B6 B6 FC D1 FF BA E6 1D 9B	B5 B4 09 B6 D1 B3 B6 4A D1 FF B6 FC 9B BA E6 1D	B6 7B BD B3 74 F9 DA CD BD A7 B1 D6 D2 9F DC 76	7A B7 4B 69 AE EF EA B2 AD BD 1C A2 D6 19 DD B3
R 8	22 C4 C5 FA DC 1B 3D 72 9D 1A 7D 7A DA B6 DC 25	B3 1C A6 2D B6 47 D4 4D 6D A2 FF DA B7 44 E6 37	B3 1C A6 2D 47 D4 4D B6 FF DA 6D A2 2F B7 44 E6	B4 B9 B3 EF B6 B6 C2 41 7D 4E EA B5 E6 B4 B9 44	CD 7A EF B6 B2 7D B7 25 4D FD E6 4E 21 B6 E6 B6
R 9	FD C3 BC B9 AA 4E B3 B4 3D E6 D6 2B 49 BC B1 FF	B9 2E B4 B6 AC B3 FC 4B B3 FC 4B AC 2F B7 44 E6	B9 2E B4 B6 B3 FC 4B AC 4B FE D6 B4 E6 E6 B9 FE	B6 B9 7A 2B 4E FE D6 B4 2C E6 B9 FE 4D D9 E6 7B	B9 B3 AC FA BD CD B7 72 A7 B7 B6 FB 9D AE 4D FB
R 10	77 1A D6 DF FB 3E BC 46 B9 B1 22 D6 DD A1 2B B6	FB A2 FB 9B B6 B2 B4 BA FB C7 B3 2B 7D B2 B4 C4		FB A2 FB 9B B2 B4 BA B6 B3 2B FB C7 C4 7D B2 B4	BF DC AD BA B6 9B CC B6 AE FF 44 B1 BD 1B B6 AE

Dari proses dekripsi diatas didapatkan hasil dekripsi adalah :

InHex : 6A E9 3B 79 AE FF D4 65 56 96 BD 67 C4 05 76 9A
Char : jé;y®ÿÔeV–½gÄvš
InHex : 6d 61 67 69 73 74 65 72 75 73 75 32 30 31 36 21
Char : magisterusu2016!

Gbr. 10 Proses Dekripsi AES

VI. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Keamanan data elektronik merupakan salah hal yang paling teknis dan wajib diterapkan agar tidak salah digunakan oleh pihak yang tidak bertanggung jawab, algoritma AES dan ECC dapat diterapkan pada data (*file*) sebagai proses keamanan.
2. Algoritma ECC menawarkan ukuran panjang kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama dengan algoritma asimetris lainnya.
3. Proses enkripsi data pada algoritma AES memiki 4 tahap yang mana pada tahap awal *AddRoundKey* digunakan 10 kali putaran terhadap kunci dan yang membuat data terproteksi dengan aman.
4. Waktu proses enkripsi lebih cepat dibandingkan dengan waktu proses dekripsi dengan menggunakan data, dan besar data yang sama pada spesifikasi perangkat komputer yang sama.

REFERENCE

[1] Abomhara, M.,Khalifa, O., Zakaria, O., Zaidan, A., Zaidan, A., Alanazi, H., 2010, Suitability of Using Symmetric Key to Secure Multimedia Data: An Overview, Journal of Applied Sciences, 15, Vol.10, pp.1656-1661.
[2] Ariyus, Dony.2006.KRIPTOGRAFI Keamanan Data dan Komunikasi.Graha ilmu.Yogyakarta.
[3] Sumitra, 2013, Comparative Analysis of AES and DES security Algorithm, International Journal of Scientific and Research Publication, Issues 1, Vol.3, pp.1-5.
[4] Enany, A., 2007, Achieving Security in Messaging and Personal Content in Symbian Phone, Thesis, Department of Interaction and System Design School of Engineering, Blekinge Institute of Technology, Sweden.
[5] Ariyus, D. 2008. Pengantar Ilmu Kriptografi. Andi Offset: Yogyakarta.
[6] Certicom. (2004). An Elliptic Curve Cryptography (ECC) Primer. The Certicom 'Cacht the Curve' White Paper Series.