semanTIK, Vol.2, No.1, Jan-Jun 2016, pp. 321-330
ISSN : 2502-8928 (Online)

APLIKASI KEAMANAN *E-MAIL* MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD*) BERBASIS *ANDROID*

Nur Ayun Qolbu M *[1], Sutardi [2], LM. Tajidun [3]
[*1,2,3]Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari
e-mail: [1*]**ayunqm@gmail.com**, [2]sutardi_hapal@yahoo.com, [3]moeh_tajidun@yahoo.com

Abstrak

Aplikasi *e-mail* merupakan salah satu fasilitas yang disediakan oleh *handphone* berbasis *android* sehingga semakin mengubah cara masyarakat dalam berkomunikasi. Dulu komunikasi jarak jauh masih menggunakan cara yang konvensional, yaitu dengan cara saling mengirim surat, tetapi sekarang komunikasi jarak jauh dapat dilakukan dengan mudah dan cepat. Namun dengan banyaknya teknologi baru yang berkembang, banyak pula terdapat informasi yang disebarluaskan baik bersifat umum maupun bersifat pribadi atau dirahasiakan, sehingga keamanan informasi tersebut belum terjamin.

Aplikasi keamanan *e-mail* diperlukan untuk mengamankan informasi tersebut dengan menerapkan ilmu kriptografi algoritma AES (*Advanced Encryption Standard*). Algortima AES merupakan algoritma kriptografi yang cukup kuat keamanannya dan mudah untuk dipecahkan dan diaplikasikan. Pada aplikasi ini diimplementasikan menggunakan bahasa pemrograman java, android studio.

Pengujian *file* yang digunakan pada aplikasi *e-mail* ini adalah *file* txt, docx, pdf dan ppt. Hasil penelitian ini menyatakan algoritma AES (*Advanced Encryption Standard*) dapat diimplementasikan ke dalam proses enkripsi dan dekripsi pada aplikasi keamanan *e-mail*. Hal ini dapat memastikan bahwa pihak yang mengerti pesan atau *file* yang dikirimkan melalui *e-mail* adalah pihak yang benar

Kata kunci— Algoritma AES (*Advanced Encryption standard*), Keamanan *E-mail*, Android

Abstract

The e-mail is one of the facilities provided by the android based mobile phone so that increasingly changing the way people communicate. First long distance communication is still using the conventional manner, that is by sending each other letters, but now remote communication can be done easily and quickly. But with so many new technologies are evolving, there are many who disseminated information is both public and private, or secret, so that information security is not guaranteed.

E-mail security applications needed to secure the information by applying the science of cryptography algorithm AES (Advanced Encryption Standard). AES algorithm is a cryptographic algorithm that is sufficiently robust security and easy to solve and apply. In this application is implemented using the Java programming language, android studio.

The test files used in the e-mail application is txt, docx, pdf and ppt. The results of this study states algorithm AES (Advanced Encryption Standard) can be implemented into the process of encryption and decryption at the application of e-mail security. This can ensure that the parties understand the message or file is sent via e-mail is the right side.

Keywords— *AES (Advanced Encryption Standard) Algorithms, E-mail Security, Android.*
.

1. PENDAHULUAN

Era globalisasi saat ini, kebutuhan manusia semakin kompleks. Sehingga manusia termotivasi untuk membuat inovasi baru yang memudahkan kita dalam menyelesaikan suatu masalah. *Handphone* adalah salah satu inovasi manusia untuk membantu dalam penyelesaian masalah tersebut.

Salah satu fasilitas yang ditawarkan oleh *handphone* berbasis *android* adalah penggunaan aplikasi *e-mail*, sehingga semakin mengubah cara masyarakat dalam berkomunikasi.

Seiring dengan perkembangan dan kemudahan dari teknologi-teknologi tersebut, banyak informasi baru bermunculan baik yang layak disebarluaskan atau dirahasiakan. Namun keamanan dari informasi tersebut belum terjamin.

Untuk mengatasi masalah tersebut, maka dibutuhkan sebuah sistem aplikasi kemanan e-mail dengan menggunakan algoritma AES (*Advanced Encryption Standard*) dimana algoritma AES dianggap cukup kuat dan sulit untuk dipecahkan, tetapi dapat diaplikasikan disebuah handphone yang berbasis Android.

Dari penelitian sebelumnya [1] yang berjudul "Implementasi Algoritma Enkripsi AES Pada Aplikasi SMS (*Short Message Service*) Berbasis *Android*" penelitian tersebut mengimplementasikan kriptografi pada android yang dimana pengiriman SMS yang bersifat penting dan rahasia diharapkan dapat dengan aman dikirim ke penerima tanpa takut kebocoran informasi.

Selanjutnya penelitian yang dilakukan oleh [2] yang berjudul "Implementasi Algoritma *Triple Data Encryption Standard* untuk Keamanan *E-mail"* yang pada penelitiannya algoritma *Triple*-DES digunakan untuk mengamankan pesan *e-mail* yang dapat melampirkan teks, gambar, dan dokumen yang akan dikirimkan dengan menggunakan tiga kunci.

Penelitian yang dilakukan oleh [3] yang berjudul "Enkripsi *Email* Dengan Menggunakan Metode Elgamal Pada Perangkat *Mobile*" yang pada penelitiannya dengan menggunakan metode enkripsi ElGamal, diharapkan proses pengiriman *email* yang dilakukan melalui perangkat *mobile* menjadi lebih *secure* karena adanya *public key* dan *private key* yang hanya diketahui oleh pengirim dan penerima pesan.

2. METODE PENELITIAN

2.1 E-mail

E-mail (*Electronic Mail*) adalah layanan yang memudahkan *user* untuk saling bertukar pesan. Tiap *user e-mail* mempunyai kotak surat (*mailbox*) yang digunakan untuk menerima dan menyimpan *e-mail* dari *user* yang lain. Salah satu keuntungan *e-mail* adalah kemampuannya dalam menghantarkan pesan ke *user* lain dengan cepat, bahkan hanya dalam waktu hitungan detik, meskipun kedua *user* tersebut berada di lokasi yang saling berjauhan. *E-mail* pertama kali diperkenalkan oleh seorang ilmuan BBN *Technologies*, Ray Tomlinson, lebih dari tiga puluhan tahun yang lalu. Sejak kemunculannya pertama kali, *e-mail* telah memprakarsai sebuah revolusi besar dalam sejarah komunikasi manusia [4].

2.2 Algoritma AES (*Advanced Encryption Standard*)

Algoritma AES (*Advanced Encryption Standard)* merupakan algoritma kriptografi yang sifatnya simetris dan *cipher block*. Dengan demikian algoritma ini menggunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah *bit* tertentu. Algoritma AES yang digunakan pada aplikasi ini dibangun menggunakan ukuran blok dan kunci 128 *bit*. Algoritma AES terdapat tiga proses yaitu proses penjadwalan kunci, enkripsi, dan dekripsi [5].

a) Analisis Penjadwalan Kunci

Proses penjadwalan kunci merupakan proses dimana *cipherkey* dijadwalkan untuk menghasilkan *subkey-subkey* yang digunakan untuk proses enkripsi dan dekripsi pada algoritma AES. Contoh penjadwalan kunci pada algoritma AES jika diketahui panjang kunci 16 *byte*, yaitu:

Cipherkey = nurayunqolbu9921

dengan solusi atau penyelesaian dengan tahap-tahap sebagai berikut:

a. Tahap pertama ubah *cipherkey* ke dalam bentuk *hexadecimal* menggunakan tabel ASCII menjadi sebagai berikut:

Cipherkey = 6e, 75, 72, 61, 79, 75, 6e, 71, 6f, 6c, 62, 75, 39, 39, 32, 31

Tahap selanjutnya melakukan operasi-operasi penjadwalan kunci. Operasi-operasi yang dilakukan yaitu sebagai berikut:

1. Memasukan *cipherkey* tersebut ke dalam blok 16 *byte*.

$$Chiperkey = \begin{bmatrix} 6e & 79 & 6f & 39 \\ 75 & 75 & 6c & 39 \\ 72 & 6e & 62 & 32 \\ 61 & 71 & 75 & 31 \end{bmatrix}$$

2. Melakukan operasi *RotWord* pada kolom terakhir dari *ciphertext*.

$$\begin{bmatrix} 39 \\ 39 \\ 32 \\ 31 \end{bmatrix} = \begin{bmatrix} 39 \\ 32 \\ 31 \\ 39 \end{bmatrix}$$

3. Melakukan operasi *SubByte* dengan tabel *s-box*.

$$\begin{bmatrix} 39 \\ 32 \\ 31 \\ 39 \end{bmatrix} = \begin{bmatrix} 12 \\ 23 \\ c7 \\ 12 \end{bmatrix}$$

Hasil dari operasi *SubByte* dilakukan operasi XOR dengan *rcon* dan W_1 (kolom ke-1 dari *Chiperkey*).

$$rcon = \begin{bmatrix} 01 & 02 & 04 & 08 & 10 & 20 & 40 & 80 & 1b & 36 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \\ 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

$$\begin{bmatrix} 6e \\ 75 \\ 72 \\ 61 \end{bmatrix} \oplus \begin{bmatrix} 39 \\ 39 \\ 31 \\ 39 \end{bmatrix} \oplus \begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix}$$

$$\begin{bmatrix} 0110\ 1110 \\ 0111\ 0101 \\ 0111\ 0010 \\ 0110\ 0001 \end{bmatrix} \oplus \begin{bmatrix} 0011\ 1001 \\ 0011\ 0010 \\ 0011\ 0001 \\ 0011\ 1001 \end{bmatrix} \oplus \begin{bmatrix} 0000\ 0001 \\ 0000\ 0000 \\ 0000\ 0000 \\ 0000\ 0000 \end{bmatrix}$$

$$= \begin{bmatrix} 0101\ 0110 \\ 1100\ 0111 \\ 0100\ 0011 \\ 0101\ 1000 \end{bmatrix} = \begin{bmatrix} 56 \\ 47 \\ 43 \\ 58 \end{bmatrix}$$

$$\begin{bmatrix} 79 \\ 75 \\ 6e \\ 71 \end{bmatrix} \oplus \begin{bmatrix} 56 \\ 47 \\ 43 \\ 58 \end{bmatrix}$$

$$\begin{bmatrix} 0111\ 1001 \\ 0111\ 0101 \\ 0110\ 1110 \\ 0111\ 0001 \end{bmatrix} \oplus \begin{bmatrix} 0101\ 0110 \\ 0100\ 0111 \\ 0100\ 0011 \\ 0101\ 1000 \end{bmatrix} =$$

$$\begin{bmatrix} 0010\ 1111 \\ 0011\ 0010 \\ 0010\ 1101 \\ 0010\ 1001 \end{bmatrix} = \begin{bmatrix} 2f \\ 32 \\ 2d \\ 29 \end{bmatrix}$$

Operasi XOR dilanjutkan ke kolom selanjutnya pada matriks *chiperkey*

$$\begin{bmatrix} 6f \\ 6c \\ 62 \\ 75 \end{bmatrix} \oplus \begin{bmatrix} 2f \\ 32 \\ 2d \\ 29 \end{bmatrix} = \begin{bmatrix} 40 \\ 5e \\ 4f \\ 5c \end{bmatrix}$$

$$\begin{bmatrix} 39 \\ 39 \\ 32 \\ 31 \end{bmatrix} \oplus \begin{bmatrix} 40 \\ 5e \\ 4f \\ 5c \end{bmatrix} = \begin{bmatrix} 79 \\ 67 \\ 7d \\ 6d \end{bmatrix}$$

Hasil XOR dari setiap matriks *chiperkey* kemudian disimpan ke dalam *subkey, yaitu:*

$$Subkey = \begin{bmatrix} 56 & 2f & 40 & 79 \\ 47 & 32 & 5e & 67 \\ 43 & 2d & 4f & 7d \\ 58 & 29 & 5c & 6d \end{bmatrix}$$

b) Proses Enkripsi Algoritma AES

Proses enkripsi algoritma AES terdiri dari empat operasi yaitu *AddRoundKey*, *SubBytes*, *ShiftRows*, dan *MixColumns*. Operasi-operasi ini diulang terus-menerus hingga menghasilkan *ciphertext*.

Contoh enkripsi pada algoritma AES, yaitu:

Cipherkey = nurayunqolbu9921

Plaintext = apa kabar unhalu

dengan solusi atau penyelesaian dengan tahap-tahap sebagai berikut:

1) Tahap awal mengubah *cipherkey* dan *plaintext* ke dalam bentuk *hexadecimal* menjadi sebagai berikut:
   Cipherkey = 56, 47, 43, 58, 2f, 32, 2d, 29, 40, 5e, 4f, 5c, 79, 67, 7d, 6d
   Plaintext = 61, 70, 61, 20, 6b, 61, 62, 61, 72, 20, 75, 6e, 68, 61, 6c, 75
2) Memasukan *cipherkey* dan *plaintext* ke dalam blok 16 *byte* sehingga menjadi:

$$Plaintext = \begin{bmatrix} 61 & 6b & 72 & 68 \\ 70 & 61 & 20 & 61 \\ 61 & 62 & 75 & 6c \\ 20 & 61 & 6e & 75 \end{bmatrix}$$

$$Cipherkey = \begin{bmatrix} 56 & 2f & 40 & 79 \\ 47 & 32 & 5e & 67 \\ 43 & 2d & 4f & 7d \\ 58 & 29 & 5c & 6d \end{bmatrix}$$

3) *Cipherkey* dan *plaintext* yang telah dimasukan ke dalam blok selanjutnya dapat dilakukan operasi-operasi enkripsi pada algoritma *Rijndael* sebagai berikut:

a. Melakukan operasi *AddRoundKey* dengan melakukan operasi XOR pada setiap kolom di *plaintext* dengan kolom di *cipherkey*.

$$\begin{bmatrix} 61 & 6b & 72 & 68 \\ 70 & 61 & 20 & 61 \\ 61 & 62 & 75 & 6c \\ 20 & 61 & 6e & 75 \end{bmatrix} \oplus \begin{bmatrix} 56 & 2f & 40 & 79 \\ 47 & 32 & 5e & 67 \\ 43 & 2d & 4f & 7d \\ 58 & 29 & 5c & 6d \end{bmatrix} = \begin{bmatrix} 37 & 44 & 32 & 11 \\ 37 & 53 & 7c & 06 \\ 22 & 2f & 3a & 11 \\ 78 & 44 & 32 & 1a \end{bmatrix}$$

b. Setelah dilakukan operasi *AddRoundKey* tersebut dilakukan perulangan dengan urutan operasi pertama yaitu operasi *SubByte*. Operasi ini yaitu melakukan subtitusi menggunakan Tabel 1.

Tabel 1 *S-box*

	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

$$\begin{bmatrix} 37 & 44 & 32 & 11 \\ 37 & 53 & 7c & 06 \\ 22 & 2f & 3a & 11 \\ 78 & 44 & 32 & 1a \end{bmatrix} \rightarrow \begin{bmatrix} 9a & 1b & 23 & 82 \\ 9a & ed & 10 & 6f \\ 93 & 15 & 80 & 82 \\ bc & 1b & 23 & a2 \end{bmatrix}$$

c. Hasil dari operasi *SubByte* dilakukan operasi *ShiftRows* yaitu memutar tiga baris terakhir dari state seperti berikut:

$$\begin{bmatrix} 9a & 1b & 23 & 82 \\ 9a & ed & 10 & 6f \\ 93 & 15 & 80 & 82 \\ bc & 1b & 23 & a2 \end{bmatrix} \rightarrow \begin{bmatrix} 9a & 1b & 23 & 82 \\ ed & 10 & 6f & 9a \\ 80 & 82 & 93 & 15 \\ a2 & bc & 1b & 23 \end{bmatrix}$$

d. Melakukan operasi *MixColumns* yaitu melakukan perkalian tiap kolom pada hasil operasi *ShiftRows* dengan matriks seperti berikut:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 9a & 1b & 23 & 82 \\ ed & 10 & 6f & 9a \\ 80 & 82 & 93 & 15 \\ a2 & bc & 1b & 23 \end{bmatrix}$$

- 02*18 = 0000 0010 * 0001 1000 = x * $x^4+x^3 = x^5+x^4$ = 0011 0000

- 03*c3 = 0000 0011 * 1100 0011 = x+1 * $x^7+x^6+x+1 = x^8+x^7+x^2+x \oplus x^7+x^6+x+1 = x^8+x^6+x^2+1$ = 1 0100 0101

Karena lebih dari 8 *bit* maka harus dilakukan XOR dengan 11B

1 0100 0101 $\oplus$ 1 0001 1011 = 0101 1110

- 01*15 = 0000 0001 * 0001 0101 = 0001 0101
- 01*d1 = 0000 0001 * 1101 0001 = 1101 0001

  0011 0000 $\oplus$ 0101 1110 $\oplus$ 0001 0101 $\oplus$ 1101 0001 = 1010 1010 = aa

Setelah melakukan perkalian matriks tiap baris dan kolom, diperoleh:

$$\begin{bmatrix} aa & 48 & 7a & da \\ 6b & 61 & dc & b4 \\ 99 & 3d & 36 & 2c \\ 27 & 38 & 39 & 3e \end{bmatrix}$$

e. Melakukan *AddRoundKey* kembali dengan menggunakan *SubKey* hasil dari penjadwalan kunci *CipherKey*.

$$\begin{bmatrix} aa & 48 & 7a & da \\ 6b & 61 & dc & b4 \\ 99 & 3d & 36 & 2c \\ 27 & 38 & 39 & 3e \end{bmatrix} \oplus \begin{bmatrix} 56 & 2f & 40 & 79 \\ 47 & 32 & 5e & 67 \\ 43 & 2d & 4f & 7d \\ 58 & 29 & 5c & 6d \end{bmatrix} = \begin{bmatrix} 55 & d2 & 89 & 44 \\ a1 & cd & 1a & 1c \\ ab & 68 & 8 & 7d \\ 7f & 8 & 65 & 12 \end{bmatrix}$$

Semua operasi tersebut diulang sebanyak 10 iterasi hingga mendapatkan *ciphertext*. Untuk iterasi 1 sampai 9 dilakukan operasi *SubByte*, *ShiftRow*, *MixColumn*, dan *AddRoundKey*. Sedangkan untuk iterasi terakhir hanya dilakukan operasi *SubByte*, *ShiftRow* dan *AddRoundKey*.

- Analisis Dekripsi AES

Proses dekripsi menggunakan algoritma AES merupakan kebalikan dari proses enkripsi. Operasi-operasi yang dilakukan yaitu *InvSubByte*, *InvShiftRow*, *InvMixColumn*, dan *AddRoundKey*. Penjadwalan kunci pada proses dekripsi pada tiap *round* berkebalikan dengan proses enkripsi yaitu dimulai dari *SubKey* ke-10 sampai dengan *cipher key*.

Contoh dekripsi pada algoritma AES, jika diketahui kunci dan *ciphertext* yang akan

digunakan untuk dekripsi dengan panjang 16 *byte*, yaitu:

$$Chipertext \begin{bmatrix} 55 & d2 & 89 & 44 \\ a1 & cd & 1a & 1c \\ ab & 68 & 8 & 7d \\ 7f & 8 & 65 & 12 \end{bmatrix} \longrightarrow subkey\ ke-10 \begin{bmatrix} 5c & da & 64 & db \\ 3b & 25 & d8 & ed \\ 68 & 3e & 38 & cb \\ 15 & bb & f5 & 33 \end{bmatrix}$$

1. Melakukan operasi *AddRoundKey* dengan melakukan operasi XOR pada setiap kolom di *ciphertext* dengan kolom di *subkey* ke-10

$$\begin{bmatrix} 55 & d2 & 89 & 44 \\ a1 & cd & 1a & 1c \\ ab & 68 & 8 & 7d \\ 7f & 8 & 65 & 12 \end{bmatrix} \oplus \begin{bmatrix} 5c & da & 64 & db \\ 3b & 25 & d8 & ed \\ 68 & 3e & 38 & cb \\ 15 & bb & f5 & 33 \end{bmatrix} = \begin{bmatrix} 73 & ee & 95 & f4 \\ 08 & 7f & f3 & aa \\ 27 & 64 & 1a & cb \\ 04 & c1 & 68 & 74 \end{bmatrix}$$

2. Melakukan operasi *Invers ShiftRows*

$$\begin{bmatrix} 73 & ee & 95 & f4 \\ 08 & 7f & f3 & aa \\ 27 & 64 & 1a & cb \\ 04 & c1 & 68 & 74 \end{bmatrix} \rightarrow \begin{bmatrix} 73 & ee & 95 & f4 \\ aa & 08 & 7f & f3 \\ 1a & cb & 27 & 64 \\ c1 & 68 & 74 & 04 \end{bmatrix}$$

3. Melakukan operasi *Invers SubBytes* menggunakan Tabel 2.

Tabel 2 *Invers S-box*

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
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
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

$$\begin{bmatrix} 0f & 99 & ad & ba \\ 62 & bf & 6b & 7e \\ 43 & 59 & 3d & 8c \\ dd & f7 & ca & 30 \end{bmatrix}$$

Setelah itu masuk *round* ke-2 sampai *round* ke-10 yang terdiri dari operasi:

1. Operasi *AddRoundKey*. Melakukan operasi XOR pada *Invers SubBytes* dengan *SubKey* ke-9

$$\begin{bmatrix} 8f & 99 & ad & ba \\ 62 & bf & 6b & 7e \\ 43 & 59 & 3d & 8c \\ dd & f7 & ca & 30 \end{bmatrix} \oplus \begin{bmatrix} ea & d3 & 5a & 1a \\ 66 & c3 & b6 & d9 \\ 44 & 7c & f5 & 02 \\ 3b & 81 & 00 & 00 \end{bmatrix} = \begin{bmatrix} 7c & 5d & 03 & 1a \\ 7f & c3 & b6 & d9 \\ 40 & 29 & b8 & 6d \\ 1e & 03 & f7 & 3c \end{bmatrix}$$

2. Operasi *Invers MixColumns*

$$\begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \times \begin{bmatrix} 2c & 5d & 03 & 1a \\ 7f & c3 & b6 & d9 \\ 40 & 29 & b8 & 6d \\ 1e & 03 & f7 & 3c \end{bmatrix}$$

Setelah melakukan perkalian matriks tiap baris dan kolom diatas, diperoleh:

$$\begin{bmatrix} 4d & c7 & fc & 6e \\ 67 & d0 & 79 & 8f \\ 1e & 49 & 4f & 56 \\ c7 & 64 & 65 & 8f \end{bmatrix}$$

3. Operasi *Invers ShiftRows*

$$\begin{bmatrix} 4d & c7 & fc & 6e \\ 67 & d0 & 79 & 8f \\ 1e & 49 & 4f & 56 \\ c7 & 64 & 65 & 8f \end{bmatrix} \rightarrow \begin{bmatrix} 4d & c7 & fc & 6e \\ 8f & 67 & d0 & 79 \\ 4f & 56 & 1e & 49 \\ 64 & 65 & 8f & c7 \end{bmatrix}$$

4. Operasi *Invers SubBytes*

$$\begin{bmatrix} 4d & c7 & fc & 6e \\ 8f & 67 & d0 & 79 \\ 4f & 56 & 1e & 49 \\ 64 & 65 & 8f & c7 \end{bmatrix} \rightarrow \begin{bmatrix} 65 & 31 & 55 & 45 \\ 73 & 0a & 60 & af \\ 92 & b9 & e9 & a4 \\ 8c & bc & 73 & 31 \end{bmatrix}$$

Semua operasi tersebut diulang sebanyak 10 iterasi hingga mendapatkan *plaintext*. Untuk iterasi 1 dilakukan operasi *AddRoundKey*, *Invers ShiftRows*, dan *Invers SubBytes*. Sedangkan untuk iterasi 2 sampai 10 dilakukan operasi *AddRoundKey*, *Invers MixColumns*, *Invers ShiftRows*, dan *Invers SubBytes*.

2.3 Android

Android adalah sebuah sisitem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk

Google, HTC, Intel, Motorola, Qualcomn, T-*mobile*, dan Nvidia [6].

Sistem operasi Android, memiliki keunggulan tersendiri bagi para penggunannya antara lain:

a. *Platform* Android disediakan melalui lisensi *Open Source*, pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan *linux Karnel*.
b. Android adalah *platform*/aplikasi yang bebas untuk *develop*, tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan yang diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk Android dapat didistribusikan atau diperdagangkan dalam bentuk apapun.
c. Dari segi tampilan, terlihat *elegant*, sehingga penggunanya tidak membuat bosan.
d. Bersifat *Multitasking* yang artinya bisa menjalankan berbagai aplikasi sekaligus, artinya anda bisa menjalankan *browsing*, *Facebook*, YM, sambil mendengarkan lagu sekaligus, namun semua itu juga tergantung dari *processor handphone* tersebut..
e. Kemudahan dalam Notifikasi, Setiap mendapatkan *Misscall*, *SMS*, *Chat* baru baik dari YM maupun *Facebook*, *E-mail* atau bahkan artikel terbaru dari *RSS Reader*.
f. Tampilan (*themes*), jika anda bosan dengan tampilan yang disajikan oleh produsen, anda bisa mengganti sesuka hati, hanya dengan men*download* di *market* Android. (rudyes.com)
g. *Widget*, yang berfungsi untuk mempermudah penggunanya dalam melakukan *setting* atau memilih aplikasi yang akan dijalankan.
h. *Syncronisasi*, jika anda pengguna Gmail ataupun Ymail, anda dapat menginterigrasikan dengan *handphone* anda, sehingga akan mempermudah anda mengecek atau mengirim *E-mail*.

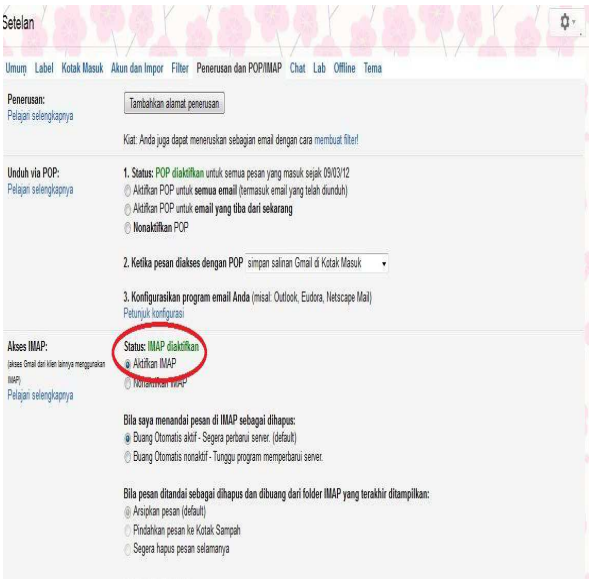
3. HASIL DAN PEMBAHASAN

Implementasi merupakan tahap dimana sistem siap untuk dioperasikan. Hasil analisis dan perancangan diimplementasikan dalam bentuk aplikasi keamanan *e-mail* dengan menggunakan bahasa pemrograman Java yang berbasis *Android.*

Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam implementasi adalah sebagai berikut:

1. Perangkat lunak yang dibutuhkan (*required software*):
   a. Sistem Operasi, *Windows 7*
   b. *Android Studio*
   c. jdk1.8.0_11
   d. SDK
2. Perangkat keras yang dibutuhkan (*required hardware*) dengan spesifikasi:
   a. *Processor Intel Core 5 Quad 2.4 GHz*
   b. RAM 2 GB
   c. Monitor dengan resolusi 1024 x 768 *pixel*
   d. *Hanpphone Android kitkat 4.4*
   e. *Hard disk 320GB*

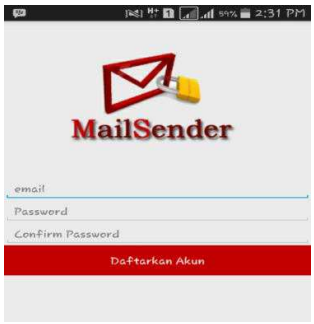
Sebelum *login* diaplikasi *MailSender*, pastikan *Account permissions* sudah diatur untuk *login* diaplikasi selain aplikasi Google. Untuk mengubah pengaturan pengguna harus masuk di *website* gmail.com dan google.com, ditunjukkan oleh Gambar 1.



Gambar 1 Pengaturan Akun

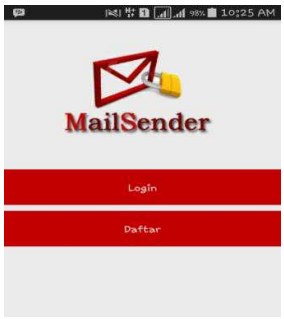
Halaman menu daftar (Gambar 2) adalah halaman dimana *user* dapat *login* pada aplikasi. Namun sebelum mengakses halaman menu utama, terlebih dahulu *user* harus

melakukan daftar dengan mengisi *username*, *password*, dan *confirm password.*



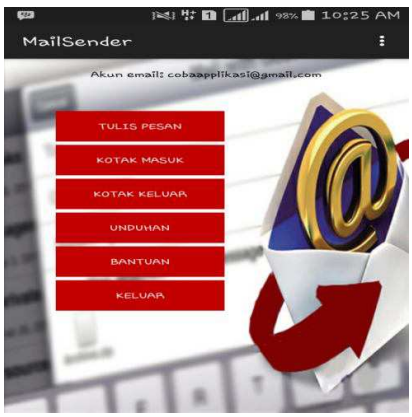
Gambar 2 Halaman Daftar

Halaman menu *login* (Gambar 3) adalah halaman dimana *user* dapat mengakses menu utama. Namun sebelum mengakses halaman menu utama, terlebih dahulu *user* harus melakukan *login* dengan mengisi *username* dan *password*.



Gambar 3 Halaman *Login*

Halaman menu utama (Gambar 4) adalah halaman awal pada saat *user* berhasil melakukan *login*, yang menampilkan semua menu utama. Terdapat 5 menu dalam *interface* ini yaitu: Tulis pesan, Kotak masuk, Kotak keluar, Unduhan, Bantuan, Keluar.



Gambar 4 Halaman Menu Utama

Gambar 5 menampilkan *form* tulis pesan.

Gambar 5 Halaman Tulis Pesan

Gambar 6 menampilkan *list* kotak masuk.

Gambar 6 Halaman Kotak Masuk

Gambar 7 menampilkan *list* kotak keluar.

Gambar 7 Halaman Kotak Keluar

Gambar 8 menampilkan *list* unduhan berdasarkan *e-mail* yang dipilih pada unduhan.

Gambar 8 Halaman Unduhan

Gambar 9 menampilkan menu bantuan.

Gambar 10 Halaman Bantuan

Pada tahap ini akan dijelaskan analisis hasil kinerja dari aplikasi *MailSender* Kripto AES

1. Analisis Enkripsi Pesan

Pesan dan kunci pesan akan ditransformasi terlebih dahulu ke *binary*. Kemudian pesan akan dienkripsi menggunakan kunci yang telah diinputkan dengan metode AES. Setelah dienkripsi maka pesan teks tidak dapat dibaca.

Pesan : halo
Kunci : abcdefghijklmnop
Ciphertext: XzRoLhMFeW+a8VIp3ItWw==
:QswhxrCz3UVGWEZS3drNRQ==

2. Analisis Dekripsi Pesan

Ciphertext tidak bisa dibaca, sebelum melewati proses dekripsi terlebih dahulu dengan menggunakan kunci yang sama, seperti berikut:

Ciphertext: XzRoLhMFeW+a8VIp3ItWw==
: QswhxrCz3UVGWEZS3drNRQ==
Kunci : abcdefghijklmnop
Pesan : halo

3. Analisis Enkripsi *File*

File yang akan dienkripsi yaitu *file* yang berekstensi *.docx. *file* yang akan dienkripsi merupakan *file* yang masih bisa terbaca atau *plainteks*, setelah dilakukan enkripsi *file* maka *file* hasil enkripsi tidak dapat dibaca.

Hasil pengujian yang didapat, akan dijadikan sebagai tolak ukur dalam proses pengembangan selanjutnya. Pengujian ini dilakukan untuk mengetahui hasil yang didapat dari aplikasi yang telah dibuat.

1. Pengujian Enkripsi Teks

No	*Input*	*Output*	Waktu proses enkripsi (s)
1	- pesan: halo - kunci: abcdefghijklmnop	- *ciphertext*: XzRoLhMFeW+a8VIp3ItWw== :QswhxrCz3UVGWEZS3drNRQ==	2 detik
2	- pesan: halo apa kabar - kunci: 123456789123456	- *ciphertext*: Á2Á@u=Å¨Á@u¨¡=Á^Á@u	2 detik
3	- pesan: halo 123 #* - kunci: abcdefgh1234567	- *ciphertext*: Ñ—x ‰Ú‰‰UÚ U¤	2 detik

Berdasarkan Pengujian 1 dapat dilihat bahwa proses enkripsi dengan panjang pesan dan kunci berbeda tidak terdapat perbedaan waktu proses yang signifikan.

2. Pengujian dekripsi teks

No	*Input*	*Output*	Waktu Proses Dekripsi (s)
1	- *ciphertext*: XzRoLhMFeW+a8VIp3ItWw== :QswhxrCz3UVGWEZS3drNRQ== - kunci: abcdefghijklmnop	- *Plainteks*: halo	2 detik
2	- *ciphertext*: Á2Á@u=Å¨Á@u¨¡=Á^Á@u - kunci: 123456789123456	- *Plainteks*: halo apa kabar	2 detik
3	- *ciphertext*: Ñ—x ‰Ú‰‰UÚ U¤ - kunci: qazwsxedcv	- *Plainteks*: halo 123 #*	2 detik

Berdasarkan Pengujian 2 dapat dilihat bahwa proses dekripsi pesan dengan kunci yang berbeda tidak terdapat perbedaan waktu proses yang signifikan.

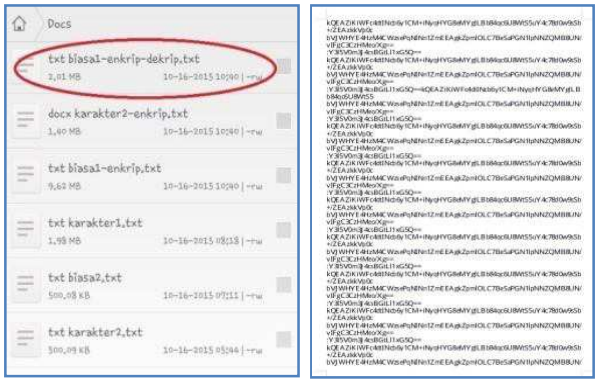
3. Pengujian Berdasarkan Ukuran *File*

No	Jenis *file*	Ukuran *file* *Type file*			Keterangan
		sebelum	proses	sesudah	
1	*File* txt	1. 2 Mb 2. 500 kb 3. 234 kb	1. 6,85 Mb 2. 1,940 kb 3. 439 kb	1. 2 Mb 2. 500 kb 3. 234 kb	Ukuran berubah pada saat proses enkripsi dan kembali keukuran semula pada saat dekripsi
2	*File* docx	1. 1,23 Mb 2. 495 kb 3. 191 kb	1. 2,436 kb 2. 1,765 kb 3. 234 kb	1. 1,23 Mb 2. 495 kb 3. 191 kb	Ukuran berubah pada saat proses enkripsi dan kembali keukuran semula pada saat dekripsi
3	*File* pdf	1. 2 Mb 2.467 kb 3. 254 kb	1. 10,243 kb 2. 1,015 kb 3. 534 kb	1. 2 Mb 2.467 kb 3. 254 kb	Ukuran berubah pada saat proses enkripsi dan kembali keukuran semula pada saat dekripsi
4	*File* ppt	1. 1,24 Mb 2. 439 kb 3. 97,5 kb	1. 2,067 kb 2. 768 kb 3. 193 kb	1. 1,24 Mb 2. 439 kb 3. 97,5 kb	Ukuran berubah pada saat proses enkripsi dan kembali keukuran semula pada saat dekripsi

Berdasarkan Pengujian 3 dapat dilihat bahwa ukuran *file* berubah pada saat sebelum dienkripsi, proses enkripsi, dan pada saat sesudah didekripsi. Pada saat dienkripsi ukuran *file* berubah menjadi lebih besar dari ukuran normal sebelum pengiriman dengan jenis *file* dan ukuran *file* yang berbeda-beda.

4. Pengujian Berdasarkan Inputan Isi *File*

File txt karakter biasa dengan ukuran 2Mb. Gambar 11 menunjukan hasil Pengujian 4.



Gambar 11 Hasil pengujian *file* setelah di enkripsi

Maka setelah didekripsi kembali ke *file* semula. Gambar 12 menunjukkan *file* hasil dekripsi.

Tuberkulosis Paru (TB Paru) merupakan salah satu penyakit menular yang disebabkan oleh bakteri mycobacterium tuberkulosa. Penderita TB Paru di Indonesia tidak hanya diderita orang dewasa saja, namun anak-anak juga sangat rentan menderita penyakit ini. Anak lebih beresiko untuk menderita TB Paru sehingga menyebabkan tingginya penderita dan kematian pada anak. Pada tahun 2012 tercatat kasus TB Paru pada anak berkisar 1,7% terjadi di Provinsi Sulawesi Tenggara (tbindonesia.or.id/tb-anak/).
Kendala utama dalam tatalaksana TB Paru pada anak yaitu masalah diagnosis. Gejala TB pada anak seringkali tidak khas dan pengambilan dahak pada anak lebih sulit untuk dilakukan, walaupun terdapat batuk tetapi jarang menghasilkan dahak (sputum). Oleh karena itu, terkadang pasien anak harus melewati serangkaian tes pemeriksaan seperti uji tuberkulin, pemeriksaan radiologis, pemeriksaan bakteriologis, dan uji Bacillus Calmette-Guerin (BCG) yang hanya dapat dilakukan di Rumah Sakit, namun tidak dapat dilakukan di Puskesmas ataupun Klinik, agar diagnosis tetap dapat dilakukan di Puskesmas ataupun Klinik maka diperlukan sebuah alat bantu aplikasi untuk men-diagnosis TB Paru anak dengan lebih efisien dengan menerapkan representasi logika fuzzy berdasarkan inputan gejala TB Paru pada anak berbasis Android.
Logika fuzzy digunakan karena dapat memberikan toleransi terhadap nilai, sehingga dengan adanya perubahan sedikit pada nilai tidak akan memberikan perbedaan yang signifikan. Dengan memanfaatkan kelebihan logika fuzzy dalam toleransi terhadap hal ambigu, diharapkan dapat menjadi pendekatan yang efisien dalam mendiagnosis TB Paru anak.
Hal ini memungkinkan penyakit TB Paru anak dapat di-diagnosis lebih awal dengan alat bantu diagnosis yang dapat digunakan di Unit Pelayanan Kesehatan agar dapat dilakukan proses penyembuhan yang tepat sasaran, mengurangi angka penderita karena lebih mudahnya untuk mengetahui anak positif atau negatif menderita TB Paru, dan sebagai alat penyuluhan bagi masyarakat untuk lebih mengenal penyakit TB Paru anak.Penelitian sebelumnya yang dilakukan oleh peneliti lain yang berkaitan dengan sistem pakar diagnosis penyakit TB Paru anak.
Dwi Purnomo Sidhi (2010) menyimpulkan bahwa riwayat kontak tuberkulosis sebagai faktor risiko hasil uji tuberkulin positif pada anak, dimana riwayat kontak dengan penderita TB paru dewasa merupakan faktor risiko terhadap hasil tes tuberkulin positif pada anak. Riwayat kontak serta riwayat sakit pada anak memberi kontribusi terhadap hasil test tuberkulin positif pada anak sebesar 90,7%.
Desi Nurmala Sari (2011) melakukan penelitian berdasarkan faktor resiko kejadian TB Paru pada anak dimana penelitian ini merupakan penelitian observasional dengan menggunakan desain penelitian kasus kontrol, dengan jumlah sampel 48 anak. Dan hasil penelitian menunjukkan variabel yang menjadi faktor resiko kejadian TB Paru pada anak adalah status gizi (OR = 8,00), tingkat pendidikan orang tua rendah (OR = 2,44), tingkat pengetahuan orang tua sedang (OR = 4,91) dan rendah (OR = 1,71), serta kepadatan hunian (OR = 1,4). Dimana variabel penentu dengan nilai OR > 1,00.
Kusrini (2009) menerapkan model case based reasoning yang digunakan sebagai metode akuisisi pengetahuan dalam aplikasi system pakar diagnosis penyakit TBC pada anak, aturan yang dihatilkan system ini mampu digunakan untuk men-diagnosis penyakit didasarkan pada data-data pasien, namun dalam penentuan diagnosis penyakit belum diimplementasikan derajat kepercayaan terhadap hasil diagnosis tersebut.
Wenny Widiastuti (2012) menerapkan metode forward chaining untuk merancang

Gambar 12 Hasil pengujian *file* setelah didekripsi

4. KESIMPULAN

Berdasarkan pengujian dan hasil penelitian yang dilakukan terhadap aplikasi Keamanan *E-mail* Menggunakan Algoritma AES (*Advanced Encryption Standard*) Berbasis *Android*, maka dapat disimpulkan :

1. Sebuah perangkat lunak yang mengimplementasikan suatu algoritma AES kriptografi kunci publik untuk enkripsi pengiriman *e-mail* telah berhasil dibangun. Secara simulasi perangkat lunak yang dibangun dapat melakukan pengiriman *e-mail* dan penerimaan *e-mail* terenkripsi dengan baik.
2. Sistem dapat memproses pesan yang dienkripsi kedalam sebuah *e-mail* dengan cara, pengguna memasukkan *file* yang akan dikirim kemudian memasukkan kode kunci. Setelah melakukan proses diatas barulah proses enkripsi dapat berjalan.
3. Hasil enkripsi menggunakan algoritma AES memiliki ukuran yang tidak sama sebelum enkripsi dan akan kembali ke ukuran normal *file* setelah didekripsi.

5. SARAN

Saran yang dapat Penulis berikan untuk pengembangan terhadap penelitian selanjutnya yaitu :

1. Metode yang digunakan pada aplikasi kriptografi *e-mail* ini dapat dikembangkan dengan metode lain, kemudian bandingkan performanya antara metode AES dengan metode lain tersebut.
2. Pada aplikasi yang dibangun *file* lampiran yang dapat dikirim hanya yang berformat txt, doc, pdf, dan ppt, sehingga untuk pengembangan selanjutnya mungkin *file* yang dapat dilampirkan bisa bervariasi lagi.

DAFTAR PUSTAKA

[1] Endriani, N., 2014, *Implementasi Algoritma Enkripsi AES Pada Aplikasi SMS (Short Message Service) Berbasis Android.*

[2] Sabir, W., 2014, *Implementasi Algoritma Triple Data Encryption Standard untuk Keamanan E-mail.*

[3] Taufan, Y., 2011, *Enkripsi E-mail Dengan Menggunakan Metode Elgamal Pada Perangkat Mobile.*

[4] Rahmat, R., 2006, *Membangun Server E-mail Berbasis FreeBSD/Linux*, Andi. Yogyakarta.

[5] Roysadi, A., 2009, *Implementasi Algoritma Kriptografi AES Untuk Enkripsi Dan Dekripsi E-mail.* Semarang: Universitas Diponegoro.

[6] Nimodia, C., Deshmukh H. R. 2012. Android Operating System.