khazanah informatika

Jurnal Ilmu Komputer dan Informatika

Penerapan Algoritma AES dan Konversi SMS ke dalam Bahasa Khek pada Aplikasi Enkripsi Berbasis *Mobile Application*

Chandra Kirana*, Edi Sugianto
[1]Program Studi Teknik Informatika
STMIK Atma Luhur
Pangkalpinang
*Chandra.kirana@atmaluhur.ac.id

Abstrak-Telepon Selular (ponsel) atau dikenal dengan nama HP (*handphone*) memiliki banyak keunggulan dan kelebihan baik dari segi fasilitas yang dimilikinya, salah satu fasilitas yang banyak digunakan berupa SMS. Akan tetapi fasilitas yang berupa SMS ini memiliki kerentanan berupa penyadapan, maka dari itu diusulkan sebuah aplikasi enkripsi menggunakan algoritma AES dan konversi bahasa Khek. Algortima *Advanced Encryption Standard* (AES) merupakan algoritma kriptografi modern yang bersifat simetris. Pada algoritma AES kunci yang dipakai memiliki panjang bervariasi yaitu 128, 192, 256 dengan memiliki jumlah ronde yang berbeda pula tergantung panjang kuncinya. Bahasa Khek adalah bahasa yang dituturkan oleh orang Hakka, memiliki 9 jenis dialek, salah yang digunakan adalah dialek lufang. Dengan menerapkan algoritma AES dan konversi bahasa Khek pada aplikasi enkripsi, pengamanan pesan yang dikirim akan terjamin kerahasiaannya, sehingga pihak yang tidak berwenang, tidak dapat mendapatkan informasi pesan yang dikirimkan. Dalam pengembangan aplikasi selanjutnya diharapkan aplikasi yang dibangun dapat mempunyai fasilitas untuk menyembunyikan sebuah folder yang digunakan untuk menyimpan hasil enkripsi dan juga dekripsi.

Kata kunci: Algoritma AES; SMS; Kriptografi

1. Pendahuluan

Setiap Short Message Service (SMS) yang masuk pada perangkat seseorang merupakan suatu privasi bagi dirinya. Sebagai contoh penyadapan SMS singkat yang pernah dialami oleh beberapa petinggi negara. Bagi dirinya penyadapan itu merugikan dirinya karena beberapa rahasia pribadinya terbongkar ke khalayak ramai. Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting, dalam hal ini sangat terkait dengan betapa pentingnya informasi tersebut dikirim dan diterima oleh orang yang berkepentingan. Oleh karena itu, untuk menjaga kerahasian SMS diperlukan sebuah sistem keamanan yang berupa aplikasi keamanan dari suatu pesan.

Bahasa adalah kemampuan yang dimiliki manusia untuk berkomunikasi dengan manusia lainnya menggunakan tanda, misalnya kata dan gerakan. Bahasa Hakka secara harfiah berarti bahasa keluarga tamu atau di Indonesia umumnya dipanggil Khek adalah bahasa yang dituturkan oleh orang Hakka, yakni suku Han yang tersebar di kawasan pegunungan provinsi Guangdong, Fujian, dan Guangxi di Tiongkok. Masing-masing daerah ini juga memiliki khas dialek Hakka yang agak berbeda tergantung provinsi dan juga bagian gunung sebelah mana mereka tinggal.

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan atau informasi yang dapat dibaca. Pesan biasanya disebut juga sebagai plaintext [1]. Algoritma AES merupakan substitusi-permutasi khas enkripsi jaringan (SPN) primitif berdasarkan struktur cipher blok simetris. Jumlah untuk operasi putaran pada AES adalah 10, 12, 14 untuk AES-128, AES-192, dan AES-26 secara terpisah, dengan jumlah bit kunci menjadi 128, 129, dan 256[2]. Algortima Advanced Encryption Standard (AES) merupakan algoritma kriptografi modern yang bersifat simetris. Pada algoritma AES kunci yang dipakai memiliki panjang bervariasi yaitu 128, 192, 256 dengan memiliki jumlah ronde yang berbeda pula tergantung panjang kunci-nya, sehingga algoritma ini sangat baik untuk pengaman teks maupun data [1].

Pada awalnya sender mengisi pesan, kemudian pesan diterjemahkan ke bahasa Khek dialek lufang, pesan hasil terjemahan kemudian dikirim dan dienkripsi dengan 16 kunci dan menghasilkan Ciphertext, saat mendekripsi pesan recipient menggunakan kode yang digunakan saat enkripsi, kemudian pesan diterjemahkan kembali ke dalam bahasa indonesia. Berdasarkan latar belakang di

atas, maka yang menjadi pembahasan utama di dalam penelitian ini adalah bagaimana mengimplementasikan algoritma AES yang dikombinasikan dengan bahasa Khek untuk meningkatkan keamanan sebuah informasi pada perangkat lunak mobile android, sehingga data hanya dapat dibaca oleh orang memiliki hak akses. Beberapa penelitian yang sudah ada dan memiliki hubungan dengan penulis teliti di antaranya seperti Penelitian Asri Prameshwari dan Nyoman Putra Sastra dengan judul "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen". Penelitian ini menyatakan bahwa algoritma AES-128 bit dapat dijadikan salah satu alternatif untuk proses keamanan data dalam hal ini enkripsi dan dekripsi file dokumen [3]. Kemudian Penelitian G.C. Prasetyadi, dkk dengan judul "File Encryption and Hiding Application Based on AES and Append Insertion Steganography". Penelitian ini menghasilkan bahwa aplikasi sebagai implementasi dari algoritma yang diusulkan terbukti layak, tetapi hanya untuk penggunaan pribadi karena beberapa perbaikan masih harus diimplemntasikan. [4]. Penelitian Dedi Darwis, dkk mengenai "Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data". Penelitian ini berhasil menggabungkan metode AES dengan steganografi Gifshuffle dengan tingkat pengujian sebesar 85% [5]. Selanjutnya penelitian yang dilakukan oleh Aji Purwinarko dan Wahyu Hardyanto dengan judul penelitian "A Hybrid Algorithm AES and Blowfish for Authentication in Mobile Applications". Penelitian ini memberikan hasil bahwa algoritma AES dan Blowfish adalah sebuah algoritma kunci simetris yang sangat cepat dan kuat[6]. Penelitian yang dilakukan oleh Bagus Satrio Waluyo Poetro dan Retantyo Wardoyo mengenai "Perbandingan Afisiensi, Efektivitas, dan Kualitas Algoritma Rijndael dengan Algoritma Camellia pada Citra Digital". Penelitian ini menghasilkan kesimpulan bahwa algoritma Rijndael lebih efisien dibandingkan dengan algoritma camellia[7].

2. Metode

a. Pengumpulan Data

Dalam penelitian ini, proses pengumpulan data dilakukan dengan melakukan studi literatur, di mana penulis mempelajari berbagai bahan baik dalam bentuk buku, jurnal, prosiding yang berkaitan dengan apa yang penulis teliti.

b. Analisis Kebutuhan

Tahap analisis kebutuhan bertujuan untuk mendefinisikan kebutuhan dari penelitian ini. Analisis kebutuhan dibedakan menjadi kebutuhan fungsional dan kebutuhan non fungsional

1. Kebutuhan Fungsional
   Ada pun spesifikasi kebutuhan fungsional adalah sebagai berikut:
   a) Aplikasi memiliki fitur enkripsi pesan yang akan mengubah pesan yang dikirim ke kode ASCII menggunakan Advanced Encryption System dan akan mengirimkan pesan yang sudah diubah tersebut ke penerima pesan.
   b) Aplikasi ini juga dilengkapi dengan fitur penterjemah kode pesan yang diterima dengan menggunakan kunci yang telah ditetapkan sebelumnya.
   c) Aplikasi hanya dapat mengelola text pesan.
2. Kebutuhan Non Fungsional
   Dalam pembangunan aplikasi ini ada 3 (tiga) kebutuhan non fungsional yang digunakan, yaitu: Kebutuhan Pengguna, Kebutuhan Hardware, dan Kebutuhan Software.
   a) Kebutuhan Pengguna
      Kebutuhan pengguna berhubungan dengan aplikasi yang dibuat adalah:
      1) Performance: memiliki response time yang cepat.
      2) Security: aplikasi yang digunakan mengubah pesan yang akan dikirim ke bentuk sandi yang tidak bisa dibaca tanpa kunci sandi sehingga pesan tersebut akan aman dari orang-orang yang tidak diinginkan.
      3) Control: kontrol aplikasi harus mudah digunakan
      4) Efficiency: pengguna aplikasi dapat menekan penggunaan kertas dan waktu yang dibutuhkan untuk memecahkan kode pesan.
      5) Service: aplikasi harus mudah digunakan.
   b) Kebutuhan Hardware
      Perangkat keras yang diperlukan untuk menjalankan aplikasi ini dengan spesifikasi sebagai berikut:
      1) Android OS, minimum versi 5.0 (Lolipop)
      2) Processor 1 Ghz
      3) Memory Internal 512 MB
      4) RAM 512 MB
      5) Screen Size, minimum 5.0 inch
   c) Kebutuhan Software
      Kebutuhan software yang dibutuhkan untuk membuat aplikasi adalah sebagai berikut:
      1) Java Development Kit versi 6 atau 7, untuk mengkompilasi kode program
      2) Java Runtime, sebagai platform untuk menjalankan aplikasi
      3) Eclipse, sebagai perangkat lunak untuk membangun aplikasi tersebut
      4) Android SDK, untuk pengembangan aplikasi android
      5) Android Development Tool, sebagai plugin android pada eclipse
      6) Android Virtual Device, sebagai emulator untuk menjalankan aplikasi android

c. Analisis Terjemahan Pesan ke dalam Bahasa Khek

Dalam penelitian ini, fitur untuk menerjemahkan sebuah pesan penulis menggunakan kosakata yang ditampung sebanyak 100 kosakata dan dapat dilihat pada Gambar 1.

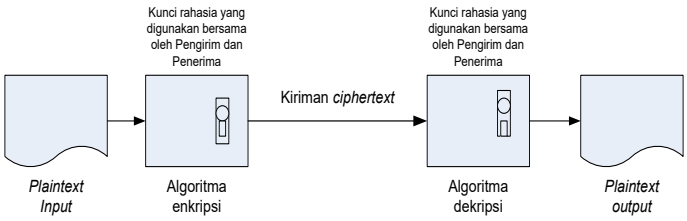
Gambar 1. Kamus bahasa Indonesia dan Khek

Dalam simulasi ini penulis akan menterjemahkan "**saya suka kamu**", ke dalam bahasa Khek:
Saya disubstitusikan menjadi "ngai"
Suka disubstitusikan menjadi "oi"
Kamu disubstitusikan menjadi "nyi"

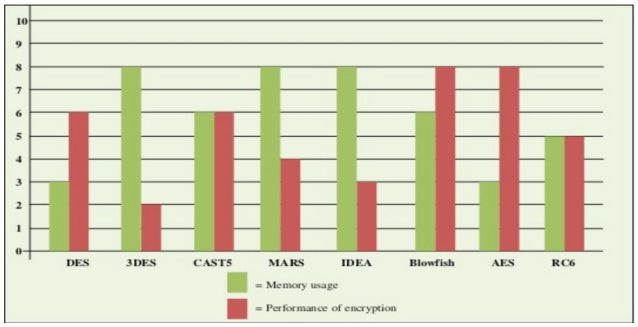
Dan dari hasil terjemahkan perkata di gabung menjadi suatu kalimat menjadi **"ngai oi nyi".**

a. Analisis Algoritma AES

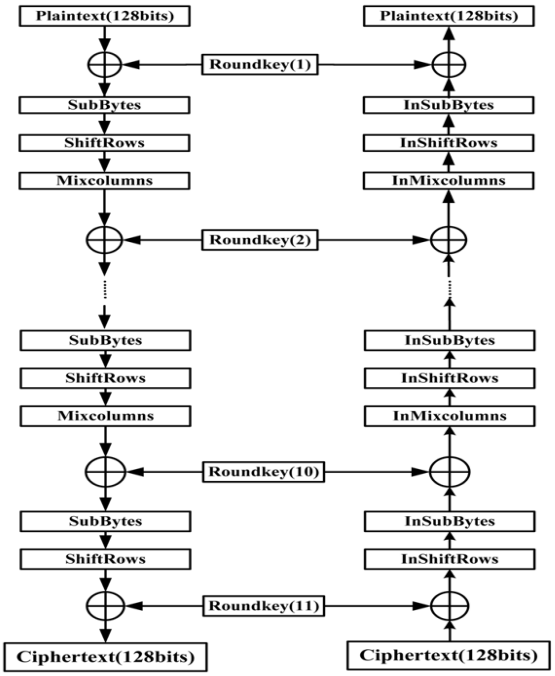
Algoritma kriptografi telah diklasifikasikan sebagai metode simetris dan asimetris. Metode simetris seperti AES menggunakan sebuah kunci rahasia yang sama untuk proses enkripsi dan dekripsi. Mereka memiliki kekuatan dan enkripsi atau dekripsi yang cepat. Proses enkripsi dan dekripsi di dalam metode simetris dapat dilihat pada Gambar 2.



Gambar 2. Skema *Symmetric Chipers Models*[8]



Gambar 3. Perbandingan Algoritma Simetris[9]



Gambar 4. Diagram alur enkripsi dan dekripsi yang tidak dioptimalkan[6]

Proses enkripsi dengan menggunakan metode AES lebih cepat jika dibandingkan dengan DES, 3DES, CAST5, MARS, IDEA, Blowfish, dan RC6. Selain kecepatan, di dalam metode AES penggunaan memory pada proses enkrispi lebih ringan bila dibandingkan dengan ketujuh algoritma lainnya.

Algoritma AES merupakan standar pemrosesan informasi Federal Pemerintah Amerika Serikat yang digunakan untuk enkripsi simetris. Algoritma AES merupakan kombinasi dari suatu algoritma yang kuat dan kunci aman, di mana algoritma ini memiliki panjang kunci variabel seperti 128,192, dan 256 bit yang menghasilkan tingkat kecepatan dan keamanan. AES adalahCipher blok simetris dengan 10 putaran untuk kunci 128-bit, 12 putaran dengan kunci 192-bit, dan juga 14 putaran untuk kunci 256-bit. Pada Gambar 4 dapat dilihat bagaimana alur diagram dari proses enkripsi dan dekripsi algoritma AES yang tidak dioptimalkan dengan 11 putaran[6].

b. Analisis Proses

Dalam penelitian ini dilakukan beberapa tahapan di antaranya tahapan konversi, enkripsi, dan dekripsi.

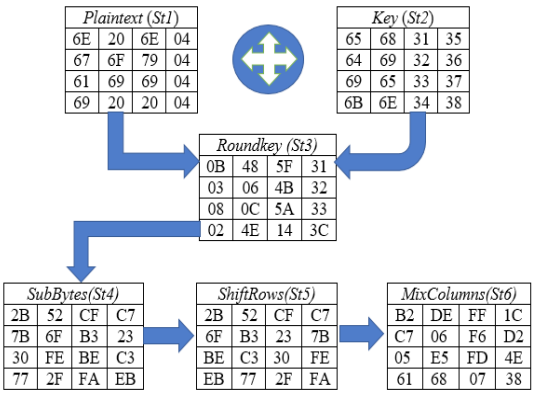
1. Proses Enkripsi AES

Plaintext	=	ngai oi nyi
Kunci	=	edikhien12345678
Chiphertext	=	÷-^6½m'k'ÉK$ž¡x

Enkripsi yang merupakan proses pengacakan pesan dengan menunjukan state sebagai objek utama yang akan di simulasikan secara per block untuk kunci panjang 128 bit ke dalam bentuk hexadecimal sebagai berikut:

Plaintext 6e 67 61 69 20 6f 69 20 6e 79 69 20 04 04 04 04

Kunci 65 64 69 6b 68 69 65 6e 31 32 33 34 35 36 37 38



Gambar 5. Proses Enkripsi AES

a) AddroundKey()
Langkah pertama yaitu mengkopi plaintext sebagai St1 dan kunci sebagai St2. St3 didapat dari proses AddRoundKey (Operasi XOR lihat gambar 5.) antara St1 dan St2 yang dikonversikan ke dalam bentuk hexadecimal, Dijelaskan sebagai berikut:

01101110	=	6E	00100000	=	20	01101110	=	6E	00000100	=	4
01100101	=	65	01101000	=	68	00110001	=	31	00110101	=	35
00001011	=	**0B**	**01001000**	=	**48**	**01011111**	=	**5F**	**00110001**	=	**31**
01100111	=	67	01101111	=	6F	01111001	=	79	00000100	=	4
01100100	=	64	01101001	=	69	00110010	=	32	00110110	=	36
00000011	=	**03**	**00000110**	=	**06**	**01001011**	=	**4B**	**00110010**	=	**32**
01100001	=	61	01101001	=	69	01101001	=	69	00000100	=	4
01101001	=	69	01100101	=	65	00110011	=	33	00110111	=	37
00001000	=	**08**	**00001100**	=	**0C**	**01011010**	=	**5A**	**00110011**	=	**33**
01101001	=	69	00100000	=	20	00100000	=	20	00000100	=	4
01101011	=	6B	01101110	=	6E	00110100	=	34	00111000	=	38
00000010	=	**02**	**01001110**	=	**4E**	**00010100**	=	**14**	**00111100**	=	**3C**

Gambar 6 Operasi Xor St1 dan St2

Sehingga dihasilkan {'0B','48','5F','31','03','06','4B', '32','08','0C','5A','33'.'02','4E','14','3C'}.

b) SubByte()
Langkah selanjutnya SubBytes() yaitu mensubtitusikan St3 dalam bentuk hexadecimal ke dalam tabel S-box (gambar 7) sehingga menghasilkan St4. Di mana diketahui S,r,c sebagai state 3 serta r (row) merupakan baris dan c merupakan kolom. Digambarkan '00' menjadi '63' sebagai berikut:

	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

Gambar 7. S-box

c) Shiftrows()
St5 merupakan hasil dari proses shiftrows() dengan menggeser secara cyclic .lihat gambar 8. sebagai berikut:

Sub Bytes			
2B	52	CF	C7
7B	6F	B3	23
30	FE	BE	C3
77	2F	FA	EB

			2B	52	CF	C7
		7B	6F	B3	23	<--
	30	FE	BE	C3	<--	<--
77	2F	FA	EB	<--	<--	<--

Gambar 8. Shiftrows

2B	52	CF	C7
6F	B3	23	7B
BE	C3	30	FE
EB	77	2F	FA

Gambar 9. Hasil Shitfrows

d) MixColumns()
Langkah Selanjutnya adalah mixcolumn() (lihat gambar 4.8), dalam langkah ini saya menjelaskan 1 langkah saja.
Contoh: Mencari nilai mixcolumn 2B, maka nilai '{2B','6F','BE','EB'} Dikalisilangkan dengan matrix {'2','3','1','1'}, seperti di bawah ini:

Shiftrows()			
2B	52	CF	C7
6F	B3	23	7B
BE	C3	30	FE
EB	77	2F	FA

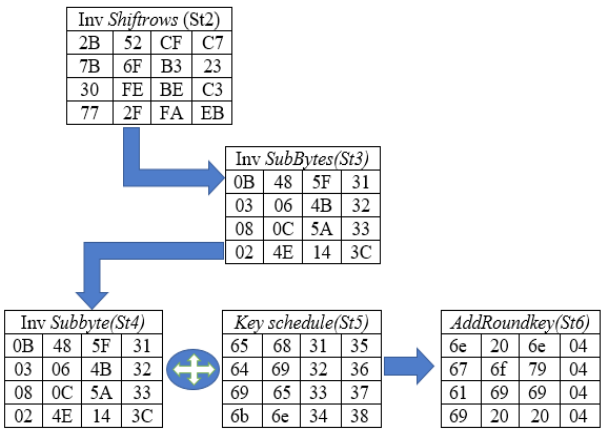
Matrix			
2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

Gambar 10. Mixcolumn()

Langkah awal mixcolomn adalah mengkonversi nilai '{2B','6F','BE','EB'} ke dalam biner, sehingga menghasilkan biner {'00101011',' 01101111',

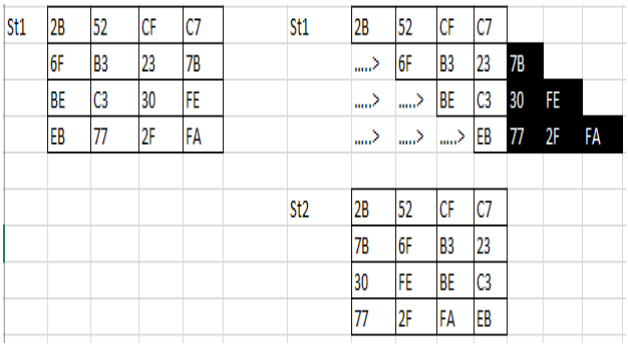
'10111110',' '11101011'}
Jika nilai yang dikalikan dengan matrix bernilai '1' maka nilai biner nya tetap, jika nilai dikalikan matrix bernilai '2' maka nilai biner digeser ke kiri dan ditambah '0', jika nilai dikalikan matrix bernilai '3' maka nilai biner digeser ke kiri dan ditambah '0' dan dixor nilai awal,
Nilai setelah di geser = {'00101011+0',' 01101111+0 ⊕ ' **01101111** ', '10111110',' '11101011'}
= {'001010110',' 011011110', '10110001',' '11101011'}
Langkah ke 2 adalah jika matrix bernilai '3' dan nilai biner hasil geser nya melebihi 255 desimal maka nilai hasil digeser di xor dengan '100011011'. jika matrix bernilai '3' dan nilai biner hasil geser nya kurang dari 255 desimal maka nilai nya tetap. Jika nilai matrix bernilai '2' dan nilai biner hasil geser nya melebihi 255 desimal maka nilai hasil digeser di xor dengan '100011011'. jika matrix bernilai '3' dan nilai biner hasil geser nya kurang dari 255 desimal maka nilai nya tetap.
Nilai langkah ke 2 = {'1010110',' 10110001', '10111110','11101011'}
Langkah ke 3 adalah semua nilai langkah ke 2 di xor = {'1010110' ⊕' 10110001' ⊕'10111110'⊕ '11101011'} = '10110010'
Langkah ke 4 adalah nilai hasil langkah 3 di ubah ke hexadecimal
Langkah 4 = '10110010'

2. Proses Dekripsi AES
Dekripsi merupakan penterjemah chiphertext menjadi ke bentuk semula atau plaintext. Berikut ini akan disimulasikan pada roundkey ke -10 (final Round) lihat lampiran 2 yang merupakan invers dari cipher contoh enkripsi. Yang mana proses mixcolumn tidak diikut sertakan pada round ini.



Gambar 11. Dekripsi AES

a) InvShiftRows()
InvShiftRows() adalah dengan menggeser St1 menjadi St2 sebagai berikut :



Gambar 12. Invers Shift Rows

b) InvSubBytes()
Langkah Selanjutnya Invers SubBytes() yaitu mensubstitusikan St3 ke dalam bentuk hexadecimal ke dalam tabel S-box^{-1} (invers S-box) lihat gambar 13. sehingga menghasilkan St3. Digambarkan '**2B'** Menjadi '**0B**' Sebagai berikut :

x \ y	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

Gambar 13. S-box^{-1}

c) AddRoundKey()
Langkah terakhir adalah AddRoundKey dengan mengoperasikan XOR (lihat gambar 14.) antara InvSubByte dengan KeySchedule, sehingga menghasilkan AddRoundKey ke 10 sebagai plaintext, dijelaskan sebagai berikut:

00001011	=	0b	01001000	=	48	01011111	=	5f	00110001	=	31
01100101	=	65	01101000	=	68	00110001	=	31	00110101	=	35
01101110	=	**6E**	**00100000**	=	**20**	**01101110**	=	**6E**	**00000100**	=	**04**
00000011	=	3	00000110	=	6	01001011	=	4b	00110010	=	32
01100100	=	64	01101001	=	69	00110010	=	32	00110110	=	36
01100111	=	**67**	**01101111**	=	**6F**	**01111001**	=	**79**	**00000100**	=	**04**
00001000	=	8	00001100	=	0c	01011010	=	5a	00110011	=	33
01101001	=	69	01100101	=	65	00110011	=	33	00110111	=	37
01100001	=	**61**	**01101001**	=	**69**	**01101001**	=	**69**	**00000100**	=	**04**
00000010	=	2	01001110	=	4e	00010100	=	14	00111100	=	3c
01101011	=	6B	01101110	=	6E	00110100	=	34	00111000	=	38
01101001	=	**69**	**00100000**	=	**20**	**00100000**	=	**20**	**00000100**	=	**04**

Gambar 14. Operasi Add Round Key

Sehingga kembali kebentuk aslinya atau plaintext {'6E','67','61','69','20','6F','69','20','6E','79','69','20','04','04','04','04'}

Diasumsikan juga seperti cipher bahwa simulasi round 1 sampai dengan round 9 caranya sama dengan proses sebelumnya, sehingga pesan yang acak dapat dikembalikan secara semula, ditujukan dalam karakter sebagai berikut:

Plaintext	6E 67 61 69 20 6F 69 20 6E 79 69 20 04 04 04 04
Kunci	65 64 69 6B 68 69 65 6E 31 32 33 34 35 36 37 38
Chiphertext	F7 2D 5E 36 BD 6D 27 6B 92 C9 4B 24 9E A1 06 78

d) Simulasi Ekspansi Kunci

Ekspansi Kunci yang dibangkitkan oleh kunci primer menghasilkan Key schedule. Kunci direpresentasikan menjadi word (w[i]).

Misalnya diketahui:

Chipher Key	65 64 69 6B 68 69 65 6E 31 32 33 34 35 36 37 38

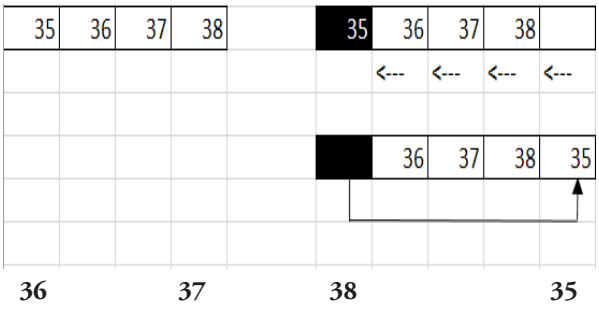
w_0 =	w_1 =	w_2 =	w_3 =
6564696B	6869656E	31323334	35363738

Maka akan diselesaikan dengan langkah di bawah ini:

1) Temp sebagai variabel menyimpan key schedule sebelumnya, untuk yang pertama diperoleh dari w_3 =

35 36 37 38

2) RotWord() merupakan proses pergeseran satu kali ke kiri secara cyclic seperti ShiftRows().



3) SubWord() merupakan substitusi tabel nonlinear (S-Box) seperti SubBytes()

36	-->	05
37	-->	9A
38	-->	07
35	-->	96

05 9A 07 96

4) Operasikan XOR antara hasil langkah 3 dengan Rcon[i] lihat persamaan 2.11.

00000101	=	5	10011010	=	9a	00000111	=	7	10010110	=	96
00000001	=	1	00000000	=	0	00000000	=	0	00000000	=	0
00000100	=	04	10011010	=	9A	00000111	=	07	10010110	=	96

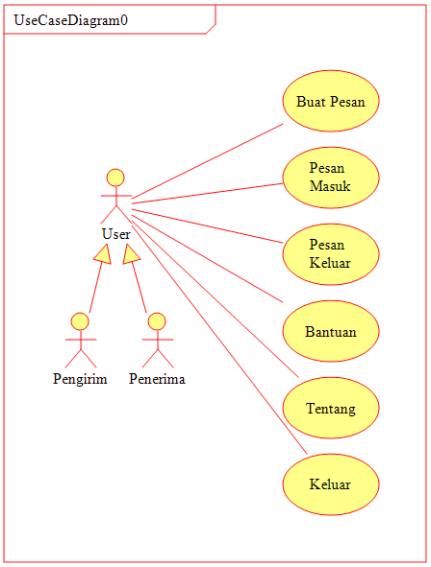
5) Langkah terakhir operasikan XOR antara hasil langkah 4 dengan w[i-Nk].

00000100	=	4	10011010	=	9a	00000111	=	7	10010110	=	96
01100101	=	65	01100100	=	64	01101001	=	69	01101011	=	6b
01100001	=	61	11111110	=	FE	01101110	=	6E	11111101	=	FD

Langkah Selanjutnya seperti 5 langkah di atas, untuk AES-128 sampai mencapai 44 words key schedule.

c. Rancangan Use Case Diagram

Use case Diagram adalah teknik untuk merekam persyaratan fungsional sebuah system. Use case mendeskripsikan interaksi tipikal antara pengguna system dengan system itu sendiri, dengan memberi sebuah narasi tentang bagaimana system tersebut digunakan. Use Case Diagram dari penelitian ini dapat dilihat pada Gambar 15.



Gambar 15. Use Case Diagram

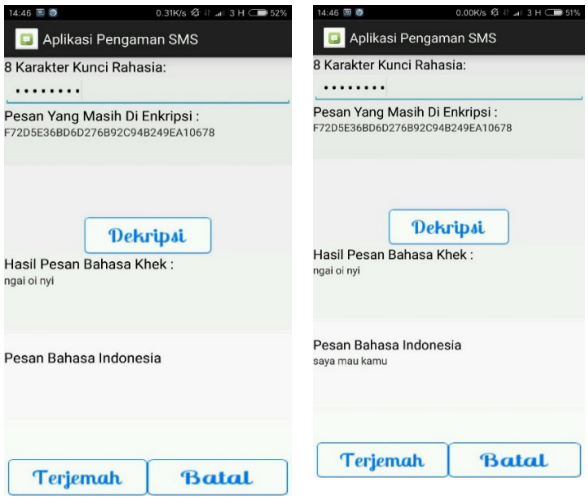
3. Hasil dan Pembahasan

Tampilan menu utama ditunjukkan dalam Gambar 16. Dalam menu utama terdapat beberapa menu, di antaranya:

1. Pilih Buat Pesan untuk membuat pesan
2. Pilih Pesan Masuk untuk melihat pesan yang masuk
3. Pilih Pesan Keluar untuk melihat pesan yang telah dibuat sebelumnya
4. Pilih Bantuan untuk melihat tata cara penggunaan aplikasi
5. Pilih Tentang untuk melihat pengembang dari aplikasi yang telah dibuat
6. Pilih Keluar untuk keluar dari aplikasi

Gambar 16. Tampilan Menu Utama Gambar

17. Tampilan Enkripsi Pesan



Gambar 18. Tampilan Dekripsi Pesan

4. Pengujian

Pengujian perangkat lunak merujuk pada standar kualitas ISO 9126. Proses pengukuran kualitas suatu aplikasi berbasis mobile cukup dengan menggunakan 4 (empat) aspek pengujian, yaitu pengujian functional, pengujian portability, pengujian usability, pengujian efficiency, sehingga pada penelitian ini penulis menggunakan 4 (empat) aspek tersebut untuk melakukan pengujian terhadap perangkat lunak yang telah dibuat[10].

a. Pengujian Usability

Pada pengujian ini, metode yang digunakan yaitu kuesioner, di mana kuesioner diberikan kepada 30 orang diantaranya 15 orang mahasiswa dan 15 orang masyarakat biasa. Sebelum mengisi kuesioner, responden harus mencoba aplikasi terlebih dahulu. Hasil yang didapatkan dari kuesioner yaitu 86,66% dari pengujian usability. Berdasarkan hasil yang didapatkan,kualitas dari perangkat lunak telah sesuai dan dapat digunakan.

b. Pengujian Functionality

Pada pengujian ini, metode yang digunakan adalah memberikan angket kepada 2 orang ahli dibidang software engineering dengan cara mencoba terlebih dahulu aplikasi yang dibuat. Berdasarkan pengujian pada Gambar 17 didapatkan persentase sebesar 93,33%, sehingga kualitas dari perangkat lunak yang dibuat telah sesuai dengan atribut functionality. Pengujian juga dilakukan dengan menggunakan Blackbox,dapat dilihat pada tabel 1.

Tabel 1. Blcakbox Testing

No.	*Test Case*	Hasil yang diharapkan	Hasil Pengujian	Benar/Salah
1	Pengguna Klik Tombol Buat Pesan	Akan tampil form buat pesan	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
2	Pengguna Klik Pilih Kontak	Akan tampil form pilih kontak	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
3	Pengguna Klik Tombol Terjemah	Pesan akan diterjemahkan kebahasa khek dan akan ditampilkan di pesan yang dikirm	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
4	Pengguna Klik Tombol Kirim	Pesan akan di enkripsi dan kirim ke nomor tujuan	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
5	Pengguna Klik Tombol Pesan Masuk	Akan tampil form list pesan masuk	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
6	Pengguna Klik Tombol Teruskan	Akan tampil form buat pesan dengan pesan di set text di pesan yang akan dikirim	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
7	Pengguna Klik Tombol Hapus	Akan tampil konfirmasi hapus pesan	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
8	Pengguna Klik Tombol Terjemah	Akan tampil form terjemah pesan	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
9	Pengguna Klik Tombol Dekripsi	Pesan yang diterima akan kembali ke bahasa Khek	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
10	Pengguna Klik Tombol Terjemah Pada Form Dekripsi	Pesan akan diubah dari bahasa Khek ke bahasa Indonesia	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
11	Pengguna Klik Tombol Batal Pada Form Dekrpsi	Akan tampil form baca pesan	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
12	Pengguna Klik Tombol Bantuan	Akan tampil halaman bantuan penggunaan aplikasi	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
13	Pengguna Klik Tombol Tentang	Akan tampil halaman tentang aplikasi	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar
14	Pengguna Klik Tombol Keluar	Akan keluar dari aplikasi	Tidak memerlukan waktu yang lama, sesuai yang diharapkan	Benar

c. Pengujian Portability

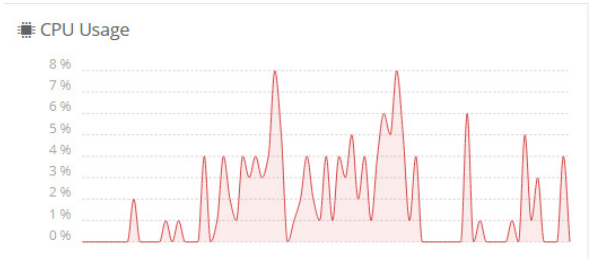
Pada pengujian ini, metode yang digunakan adalah dengan mencoba ke beberapa smartphone dengan sistem operasi android versi Marshmallow, kitkat, dan Nougat, di mana aplikasi yang dibuat apakah dapat terinstal dan dijalankan (lihat tabel 1).

d. Pengujian Efficiency

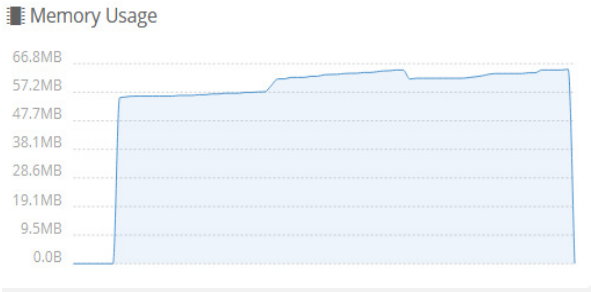
Pada pengujian ini, metode yang digunakan yaitu dengan menggunakan sebuah tools Testdroid yang dapat diakses pada https://cloud.testdroid.com. Dengan tools Testdroid dapat dilihat penggunaan memory dan CPU. Pengujian aspek efficiency menggunakan device yang telah disediakan pada tools Testdroid yaitu device LG Google Nexus 5 6.0.1. Berdasarkan Gambar 20 dapat dilihat bahwa penggunaan memory cukup aman dan perangkat lunak dapat dijalankan dengan baik. Untuk penggunaan CPU dapat dilihat pada Gambar 19 di mana terlihat bahwa penggunaan CPU rata-rata berada pada diangka 8% yang mana angka tersebut berada di bawah batas aman yang telah ditetapkan oleh Little Eye (mobile app analysis tools) yaitu pada angka 15% [9].

Tabel 1. Hasil Aspek Portability

No.	Jenis Perangkat	Jenis Android	Proses Instalasi	Proses Running Aplikasi
1	Xiaomi Redmi Note 4	6.0 (Marshmallow)	Berhasil	Berjalan tanpa ada error
2.	Xiaomi Redmi 4X	7.0 (Nougat)	Berhasil	Berjalan tanpa ada error
3.	LG Google Nexus 5 6.0.1	6.0 (Marshmallow)	Berhasil	Berjalan tanpa ada error
4	Samsung Tab 3	4.4 (Kitkat)	Berhasil	Berjalan tanpa ada error



Gambar 19. Penggunaan CPU oleh aplikasi



Gambar 20. Penggunaan Memory oleh aplikasi

5. Kesimpulan

Penerapan algoritma AES berhasil diterapkan ke dalam aplikasi SMS enkripsi berbasis android dengan baik, sehingga pesan tidak bisa dibaca oleh pihak yang tidak berkepentingan. Konversi bahasa dari bahasa indonesia ke dalam bahasa Khek juga berhasil diterapkan ke dalam aplikasi SMS enkripsi sehingga tingkat keamanan informasi yang disampaikan menjadi lebih aman.

6. Persantunan

Penulis mengucapkan terima kasih kepada pihak kampus yang telah banyak memberikan dukungan finansial, sehingga penelitian yang penulis lakukan dapat berjalan dengan lancar dan dapat terselesaikan dengan baik.

Daftar Pustaka

[1] S. Kromodimoeljo, Teori dan Aplikasi Kriptografi. Jakarta: SPK IT Consulting, 2009.

[2] Y. Yuan, Y. Yang, L. Wu, and X. Zhang, "A High Performance Encryption System Based on AES Algorithm with Novel Hardware Implementation," IEEE, pp. 4–5, 2018.

[3] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," Eksplora Inform., 2018.

[4] G. C. Prasetyadi, R. Refianti, and A. B. Mutiara, "File Encryption and Hiding Application Based on AES and Append Insertion Steganography," vol. 16, no. 1, pp. 361–367, 2018.

[5] D. Darwis, R. Prabowo, and N. Hotimah, "Kombinasi Gifshuffle, Enkripsi AES dan Kompresi Data Huffman Untuk Meningkatkan Keamanan Data," J. Teknol. Inf. dan Ilmu Komput., vol. 5, no. 4, pp. 389–394, 2018.

[6] A. Purwinarko and W. Hardyanto, "A Hybrid Security Algorithm AES and Blowfish for Authentication in Mobile Applications," vol. 5, no. 1, pp. 76–80, 2018.

[7] B. Satrio, W. Poetro, and R. Wardoyo, "Perbandingan Efisiensi , Efektifitas dan Kualitas Algoritma Rijndael dengan Algoritma Camellia pada Citra Digital," J. UGM, vol. 4, pp. 281–291, 2014.

[8] Y. Kurniawan, Kriptography Keamanan Internet dan Jaringan Komunikasi. Bandung: Informatika, 2004.

[9] I. Ahmad and W. Widodo, "khazanah informatika Penerapan Algoritma A Star (A *) pada Game Petualangan Labirin Berbasis Android," khazanah Inform., vol. 3, no. 2, pp. 57–63, 2017.