APLIKASI ENKRIPSI SMS (*SHORT MESSAGE SERVICE*) MENGGUNAKAN METODE AES (*ADVANCED ENCRYPTION STANDARD*) 128 bit BERBASIS ANDROID

Hendra Gunawan

Teknik Informatika, STMIK-IM
Jl.Jakarta No.79 Bandung
hendra_gunawan@engineer.com

ABSTRAK

Perkembangan pesat pada dunia teknologi, salah satunya adalah telepon selular. Mulai dari ponsel yang hanya bisa digunakan untuk bicara dan sms hingga "ponsel cerdas" (*smartphone*) seperti android yang memiliki berbagai fungsi dan fitur. Perangkat *smartphone* seperti android memiliki keamanan yang dirancang demi kenyamanan dari pengguna, namun beberapa layanan pada fitur-fitur tertentu sama sekali tidak memiliki metode pengamanan pada data yang disimpannya, salah satunya adalah layanan SMS (*Short Message Service*) yang pada perangkat android secara *standard* sama sekali tidak memiliki metode pengamanan yang cukup. Keamanan dan kerahasiaan sangat penting dalam segala aspek untuk melindungi data. Pesan teks pada *smartphone* android yaitu SMS (*Short Message Service*) merupakan salah satu data penting yang perlu sistem keamanan data.

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. AES (*Advanced Encryption Standard*) adalah teknik merahasiakan sandi atau data sesuai standarisasi dari FIPS (*Federasi Information Processing Standard*) versi 197 dengan menggunakan algoritma rijndael. AES dibuat oleh kriptografer asal Belgia, yakni Vincent Rijmen dan Joan Daemen.

Sehingga untuk dapat mengamankan pesan teks pada *smartphone* android yaitu SMS (*Short Message Service*) dapat digunakan algoritma AES sebagai algoritma enkripsi dan juga deskripsi pesan. Dengan menggunakan enkripsi diharapkan dapat memenuhi sistem keamanan pengiriman pesan yang sangat diperlukan bagi para pengguna layanan SMS (*Short Message Service*).

Kata Kunci : *Short Messages Service* (SMS), Android, Enkripsi, Dekripsi, *Advanced Encryption Standard* (AES)

1. PENDAHULUAN

1.1. Latar Belakang

Perkembangan pesat pada dunia teknologi, salah satunya adalah telepon selular. Mulai dari ponsel yang hanya bisa digunakan untuk bicara dan sms hingga "ponsel cerdas" (*smartphone*) seperti android yang memiliki berbagai fungsi dan fitur.

Seperti yang diketahui bahwa perangkat *smartphone* seperti android memiliki keamanan yang dirancang demi kenyamanan dari pengguna, namun beberapa layanan

pada fitur-fitur tertentu sama sekali tidak memiliki metode pengamanan pada data yang disimpannya, salah satunya adalah layanan SMS (*Short Message Service*) yang pada perangkat android secara *standard* sama sekali tidak memiliki metode pengamanan yang cukup.

Keamanan dan kerahasiaan sangat penting dalam segala aspek untuk melindungi data. Pesan teks pada *smartphone* android yaitu SMS (*Short Message Service*) merupakan salah satu data penting yang perlu sistem keamanan data. Menurut pengguna disaat sebuah pesan atau data menjadi penting dan bersifat rahasia serta tidak ingin diketahui *public* maka perlu diadakan perlindungan terhadap data tersebut.

AES (*Advanced Encryption Standard*) adalah teknik merahasiakan sandi atau data sesuai standarisasi dari FIPS (*Federation Information Processing Standard*) versi 197 dengan menggunakan algoritma rijndael. AES dibuat oleh kriptografer asal Belgia, yakni Vincent Rijmen dan Joan Daemen. Sehingga untuk dapat mengamankan pesan teks pada *smartphone* android yaitu sms dapat digunakan algoritma AES (*Advanced Encryption Standard*) sebagai algoritma enkripsi dan juga deskripsi pesan.

Oleh karena itu penulis tertarik untuk membuat suatu aplikasi pesan singkat SMS (*Short Message Service*) yang dapat memberikan layanan keamanan bagi pengguna *smartphone* android. Kita bisa merahasiakan pesan singkat tersebut dengan metode enkripsi AES (*Advanced Encryption Standard*) 128 bit menggunakan bahasa pemrograman java yang dipakai saat mengembangkan suatu aplikasi berbasis android.

1.2. Tujuan

Tujuan penelitian ini adalah merancang dan membangun aplikasi berbasis android yang mampu mengenkripsi dan deskripsi pesan teks SMS (*Short Message Service*) menggunakan metode AES (*Advanced Encryption Standard*) 128 bit.

1.3. Metode Penelitian

Metodologi yang digunakan untuk membangun sistem ini adalah Model *Waterfall*. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematik, dengan beberapa tahapan, yaitu: *System Engineering*, *Analysis*, *Design*, *Coding*, *Testing* dan *Maintenance*.

1.3.1. Teknik Pengumpulan Data

Dalam penelitian ini digunakan teknik pengumpulan data yang dilakukan dengan beberapa tahap, diantaranya :

1. Observasi, yaitu melihat dan mengamati secara langsung proses pengolahan data yang ada.
2. Wawancara, yaitu mengumpulkan data yang dilakukan dengan cara melakukan tanya jawab secara langsung kepada pihak-pihak yang terkait guna mendapatkan keterangan-keterangan yang diperlukan.
3. Studi pustaka, yaitu membaca buku–buku atau mencari referensi dari internet yang terkait secara langsung maupun tidak langsung untuk mengetahui secara teoritis permasalahan yang sedang dihadapi.

1.3.2. Metode Pengembangan Perangkat Lunak

Metodologi yang digunakan untuk membangun sistem ini adalah Model *Waterfall*. Model ini merupakan sebuah pendekatan terhadap pengembangan perangkat lunak yang sistematik, dengan beberapa tahapan, yaitu: *System Engineering*, *Analysis*, *Design*, *Coding*, *Testing* dan *Maintenance*. *System Engineering*, merupakan bagian awal dari pengerjaan suatu proyek perangkat lunak. Dimulai dengan mempersiapkan segala hal yang diperlukan dalam pelaksanaan proyek.

1. *Analysis*, merupakan tahapan dimana *System Engineering* menganalisis segala hal yang ada pada pembuatan proyek atau pengembangan perangkat lunak yang bertujuan untuk memahami sistem yang ada, mengidentifikasi masalah dan mencari solusinya.
2. *Design*, tahapan ini merupakan tahap penerjemah dari keperluan atau data yang telah dianalisis ke dalam bentuk yang mudah dimengerti oleh pemakai *(user).*
3. *Coding*, yaitu menerjemahkan data yang dirancang ke dalam bahasa pemrograman yang telah ditentukan.
4. *Testing*, merupakan uji coba terhadap sistem atau program setelah selesai dibuat.
5. *Maintenance,* yaitu penerapan sistem secara keseluruhan disertai pemeliharaan jika terjadi perubahan struktur, baik dari segi *software* maupun *hardware*.

2. LANDASAN TEORI

2.1. Pengertian Perangkat Lunak

Pengertian perangkat lunak menurut Rosa A.S.M.Shalahuddin (2013:2) dalam bukunya yang berjudul "Rekayasa Perangkat Lunak", perangkat lunak didefinisikan sebagai berikut : "Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)".

Menurut Rosa A.S.M.Shalahuddin (2013:3), dalam bukunya "Rekayasa Perangkat Lunak" karakteristik perangkat lunak adalah sebagai berikut :

1. Perangkat lunak dibangun dengan rekayasa *(software engineering)* bukan diproduksi secara manufaktur atau pabrikan.
2. Perangkat lunak tidak pernah usang *("wear out")* karena kecacatan dalam perangkat lunak bisa diperbaiki.
3. Barang produksi pabrikan biasanya komponen barunya akan terus diproduksi, sedangkan perangkat lunak biasanya terus diperbaiki seiring bertambahnya kebutuhan.

2.2 Pengertian SMS (*Short Message Service*)

Layanan pesan singkat atau Surat Masa Singkat (bahasa Inggris: *Short Message Service disingkat SMS*) adalah sebuah layanan yang dilaksanakan dengan sebuah telepon genggam untuk mengirim atau menerima pesan-pesan pendek.

Short message service (SMS) adalah salah satu komunikasi teks melalui telepon seluler. SMS merupakan salah satu media yang paling banyak digunakan saat ini. Selain murah, prosesnya juga berjalan cepat dan langsung sampai pada tujuan, tetapi selama ini SMS (*Short Message Service*) baru digunakan sebatas untuk mengirim dan menerima pesan antara sesama pemilik telepon seluler.

Menurut (Riadi, 2012)," SMS (*Short Message Service*) merupakan layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (nirkabel), memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antar terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal" . SMS berupa pesan teks, jumlah karakter pada setiap pengiriman bergantung pada operatornya. Operator selular di Indonesia umumnya membatasi 160 karakter untuk satu pengiriman dan penerimaan

SMS. Selain itu SMS merupakan metode *store* dan *forward* sehingga keuntungan yang didapat adalah pada saat telepon selular penerima tidak dapat dijangkau, dalam arti tidak aktif atau diluar *service* area, penerima tetap dapat menerima SMS-nya apabila telepon selular tersebut sudah aktif kembali.

2.3 Kriptografi

Menurut Sadikin (2012 : 9), suatu sistem kriptografi (kriptosistem) bekerja dengan cara menyandikan suatu pesan menjadi suatu kode rahasia yang hanya dimengerti oleh pelaku sistem informasi saja. Pada dasarnya mekanisme kerja semacam ini telah dikenal sejak jaman dahulu. Bangsa Mesir kuno sekitar 4000 tahun yang lalu bahkan telah mempraktekkannya dengan cara yang sangat primitif.

Namun pada pengertian modern, Kriptografi adalah ilmu yang bersandarkan pada teknik matematika yang berkaitan dengan keamanan informasi seperti kerahasiaan, keutuhan data, dan otentikasi tentitas. Jadi pengertian kriptografi modern adalah tidak hanya berkaitan dengan penyembunyian pesan namun lebih tertuju pada sekumpulan teknik yang menyediakan kemanan informasi.

Berikut ini diberikan beberapa istilah yang umum digunakan dalam pembahasan kriptografi.

1. *Plaintext*

   Plaintext (*message*) merupakan pesan asli yang ingin dikirimkan dan dijaga keamanannya. Pesan ini tidak lain berupa dari informasi tersebut.

2. *Ciphertext*

   Ciphertext merupakan pesan yang telah dikodekan (disandikan) sehingga siap untuk dikirimkan.

3. *Cipher*

   Cipher merupakan algoritma matematis yang digunakan untuk proses penyandian *Plaintext* menjadi *ciphertext.*

4. Enkripsi

   Enkripsi (*encryption*) merupakan proses yang dilakukan untuk menyandikan *Plaintext* sehingga menjadi *ciphertext.*

5. Dekripsi

Dekripsi (*decryption*) merupakan proses yang dilakukan untuk memperoleh kembali *Plaintext* dari *ciphertext.*

2.4. AES (*Advanced Encryption Standard)*

Menurut Rifki Sadikin dalam bukunya "Kriptografi Untuk Keamanan Jaringan" (2012:152), AES merupakan sistem penyandian blok *non-feisel* karena aes menggunakan komponen yang selalu memiliki *Invers* dengan panjang blok 128bit. AES dapat memiliki panjang kunci bit 128, 192 dan 256 bit. Penyandian AES menggunakan proses yang berulang yang disebut dengan ronde. Jumlah ronde yang digunakan AES tergantung dengan panjang kunci yang digunakan. Setiap ronde membutuhkan kunci ronde dan masukkan dari ronde berikutnya. Kunci ronde dibangkitkan berdasarkan kunci yang diberikan. Relasi antara ronde dan panjang kunci dapat dilihat pada tabel berikut :

Tabel : 2.1 Hubungan Antara Jumlah Ronde dan Panjang Kunci AES.

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

2.5. Pengenalan Sistem Operasi Android

Android dalam buku karya Andry (2011) adalah sistem operasi untuk telepon seluler yang merupakan hasil modifikasi Linux. Sejauh ini Android termasuk sistem operasi yang cepat sekali memperbarui software mereka. Android menyediakan kesempatan terbuka bagi para pengembang (*developer*) untuk menciptakan aplikasi kreasi sendiri untuk ditanamkan pada sistem operasi ini. Awalnya, Google Inc. membeli perusahaan Android Inc., perusahaan *start up* yang saat itu tengah fokus membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat konferensi perdana terkait Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di

bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

3. PEMBAHASAN

3.1. Analisis Sistem

3.1.1 Analisis Kebutuhan Fungsional

Merupakan kebutuhan aktifitas dan *service* yang harus disediakan sistem yang akan dikembangkan, antara lain :

1. Sistem mampu menerima setiap *input* pesan text dan *input* kunci, yang di masukkan oleh pengguna.
2. Sistem mampu mengakses *contact* pada *Smartphone Android.*
3. Sistem mampu melakukan proses enkripsi dan juga deskripsi dengan menggunakan metode AES (*Advanced Encryption Standard*) 128 bit.
4. Sistem mampu membaca setiap pesan yang masuk pada kotak masuk pesan.

3.1.2 Analisis Kebutuhan Non Fungsional

Merupakan fitur-fitur lain yang diperlukan agar sistem dapat beroperasi dengan baik sesuai dengan yang diharapkan, antar lain :

1. Sistem dapat di akses oleh semua pengguna android.
2. Sistem mudah digunakan dengan sentuhan jari tangan dan cukup sederhana.

3.1.3 Kebutuhan Perangkat Lunak

Perangkat lunak ini dapat berjalan baik apabila memnuhi standard minimal dari perangkat lunak. Perangkat lunak minimal memiliki spesifikasi sebagai berikut :

1. *Operating system* Android versi 4.0 (*ICS: Ice Cream Sandwich*)
2. *Support messaging* SMS (*Short Message Services*)

3.1.4 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang untuk dapat mejalankan aplikasi sms enkripsi ini, memiliki spesifikasi sebagai berikut :

1. *Smartphone* android
2. RAM 512 MB
3. CPU 1 GHZ
4. *Internal Memory* 512 MB

3.1.5 Analisis AES (*Advanced Encryption Standard*) 128 bit

Algoritma *Advanced Encryption Standard* (AES) merupakan algoritma kriptografi yang sifatnya simetris dan *cipher block*. Dengan demikian algoritma ini mengunakan kunci yang sama saat enkripsi dan dekripsi serta proses masukan dan keluarannya dibagi berupa blok-blok terlebih dahulu lalu proses enkripsi atau deskripsi akan dilakukan terpisah terhadap masing-masing blok data. Setiap blok AES terdiri atas 4x4 byte dengan jumlah 128 bit per blok. Apabila ada blok yang kurang dari 128 bit maka dilakukan penambalan karakter (*padding*) agar memenuhi syarat 128 bit per blok. Algoritma AES (*Advanced Encryption Standard*) yang digunakaan pada aplikasi enkripsi SMS yang dibangun menggunakan ukuran blok 128 bit akan dilengkapi PKCS7Padding dan MD5 Hash pada kunci serta output *ciphertext encoding* dan *decoding* ke dalam Base64. Algoritma *Advanced Encryption Standard* (AES) terdapat dua proses yaitu proses penjadualan kunci dan enkripsi.

Contoh :

Pesan (*Plaintex*) : **STMIK IM BDG OK**

Dalam Hexadesimal : **53 54 4D 49 | 4B 20 49 4D | 20 42 44 47 | 20 4F 4B _**

PKCS7 : **53 54 4D 49 | 4B 20 49 4D | 20 42 44 47 | 20 4F 4B 01**

Kunci (*key*) : **IWAN H 361341002**

Cipherkey MD5 Hash : **C5 14 2B 1D | 55 2E 69 A4 | B4 4B 8B B3 | 37 69 E0 26**

3.1.5.1 Analisis Proses Penjadwalan Kunci

Proses penjadwalan kunci merupakan proses dimana *cipherkey* dijadwalkan untuk menghasilkan kunci ronde (*key scedule* / *subkey-subkey*) yang digunakan untuk proses enkripsi dan dekripsi pada algoritma AES (*Advanced Encryption Standard*). Proses

penjadalan kunci terdiri dari beberapa operasi meliputi *SubWord, SubBytes* dan operasi XOR dengan *RCon* pada Tabel 2.4.

Tahap selanjutnya melakukan operasi-operasi penjadwalan kunci AES 128 bit. Operasi- operasi penjadwalan yang dilakukan yaitu *RootWord, SubByte,* dan melakukan operasi XOR dengan R*con* pada Tabel 2.4 yang sudah ditentukan untuk menghasilkan kunci ronde (*key schedule* / *subkey*). Operasi-operasi yang dilakukan yaitu sebagai berikut :

1. Langkah pertama yang harus dilakukan adalah menyediakan array kunci berukuran 4x4 yang terdiri dari 128 bit di mana pada tiap selnya terdiri atas 1 byte, ini dinamakan kunci ronde ke-0. Masukan *cipherkey* tersebut kedalam blok array 4x4 (16 *Byte*)

   Cipherkey : C5 14 2B 1D | 55 2E 69 A4 | B4 4B 8B B3 | 37 69 E0 26

   w_0 w_1 w_2 w_3

C5	55	B4	37
14	2E	4B	69
2B	69	8B	E0
1D	A4	B3	26

2. Selanjutnya lakukan operasi *RotWord* pada kolom terakhir dari *ciphertext*.

   Yaitu tukar posisi antarbaris. Baris ke-1 menjadi baris ke-4, ke-2 menjadi ke-1, ke-3 menjadi ke-2, dan ke-4 menjadi ke-3. Kolom terakhir yang sudah ditukar posisinya bernama *RotWord*.

   w_0 w_1 w_2 w_3

C5	55	B4	37		69
14	2E	4B	69	=	E0
2B	69	8B	E0		26
1D	A4	B3	26		**37**

   Hasil *RootWord* : 69 E0 26 37

3. Melakukan operasi *SubBytes,* yaitu substitusi dengan Tabel 2.2 substitusi *SubBytes* (*S-box*).

69	=	**F9**
E0		**E1**
26		**F7**
37		**9A**

Hasil *SubBytes* : F9 E1 F7 9A

4. Hasil dari operasi *SubBytes* dilakukan operasi XOR dengan *rcon* pada tabel RC yang telah ditentukan dengan w_0 (kolom ke-1 kunci ronde ke-0).

Tabel : 3.1. Konstan RC dalam hexadesimal

I	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
RC[i]	01	02	04	08	10	20	40	80	1B	36

w_0 Rcon w_4

C5		F9		01		**3D**
14		E1		00		**F5**
2B	⊕	F7	⊕	00	=	**DC**
1D		9A		00		**87**

Hasil XOR tersebut menjadi kolom ke-1(w_4) *Round Key* 1 : 3D F5 DC 87

5. Selanjutnya untuk mendapatkan kolom ke-2 *Round Key* 1 (w_5), dilakukanlah peng-XOR an kolom ke-1 (w_4) *Round Key* 1 dengan kolom 2 array kunci ronde ke-0 (w_1)

w_1		w_4		w_5
55	⊕	3D	=	**68**
2E		F5		**DB**
69		DC		**B5**
A4		87		**23**

Hasil w_5 : 68 DB B5 23

6. Selanjutnya untuk mendapatkan Kolom ke-3 Round Key ke-1 (w_6),didapatkan dari peng-XOR-an kolom 2 Round Key 1 (w_5) dengan kolom 3 array Kunci ronde ke-0 (w_2).

w_2		w_5		w_6
B4	⊕	68	=	**DC**
4B		DB		**90**
8B		B5		**3E**
B3		23		**90**

Hasil w_6 : DC 90 3E 90

7. Untuk mendapatkan kolom ke-4 Round Key 1 (w_7) didapatkan dari peng XOR-an kolom 3 Round Key 1(w_6) dengan kolom 4 array Kunci ronde ke-0 (w_3)

w_3		w_6		w_7
37	⊕	DC	=	**EB**
69		90		**F9**
E0		3E		**DE**
26		90		**B6**

Hasil w_7 : EB F9 DE B6

Sehingga didapat hasil kunci ronde (*subkey*) ke- 1 :

w_4 w_5 w_6 w_7

3D	**68**	**DC**	**EB**
F5	**DB**	**90**	**F9**
DC	**B5**	**3E**	**DE**
87	**23**	**90**	**B6**

Kunci Ronde ke-1 : 3D F5 DC 87 | 68 DB B5 23 | DC 90 3E 90 | EB F9 DE B6

Subkey ini yang akan digunakan untuk proses enkripsi atau deskripsi pada algoritma *Advanced Encryption Standard* (AES) 128 bit pada round ke-1 untuk round selanjutnya dilakukan penjadualan kunci kembali sampai round ke-10.

Kemudian langkah-langkah di atas diulang 9x lagi untuk mendapatkan 9 buah *RoundKey* berikutnya. *Resume* hasil kunci rounde ke -0 sampai ke-10 ($w_0 \ldots w_{43}$, selengkapnya pada Tabel 3.2 dan Tabel 3.3 sebagai berikut :

Tabel: 3.2. Hasil Kunci Ronde (*Subkey*) ke-0 sampai 5

Ronde	**KeyScedule**			
Ronde 0	C5	55	B4	37
	14	2E	4B	69
	2B	69	8B	E0
	1D	A4	B3	26
Ronde 1	3D	68	DC	EB
	F5	DB	90	F9
	DC	B5	3E	DE
	87	23	90	B6
Ronde 2	A6	CE	12	F9
	E8	33	A3	5A
	92	27	19	C7
	6E	4D	DD	6B
Ronde 3	1C	D2	C0	39
	2E	1D	BE	E4
	ED	CA	D3	14
	F7	BA	67	0C

Ronde 4	7D	AF	6F	56
	D4	C9	77	93
	13	D9	0A	1E
	E5	5F	38	34
Ronde 5	B1	1E	71	27
	A6	6F	18	8B
	0B	D2	D8	C6
	54	0B	33	07

Tabel : 3.3. Hasil Kunci Ronde *Subkey* ke-6 sampai 10

Ronde	**KeyScedule**			
Ronde 6	AC	B2	C3	E4
	12	7D	65	EE
	CE	1C	C4	02
	98	93	A0	A7
Ronde 7	C4	76	B5	51
	65	18	7D	93
	92	8E	4A	48
	F1	62	C2	65
Ronde 8	98	EE	5B	0A
	37	2F	52	C1
	DF	51	1B	53
	20	42	80	E5
Ronde 9	FB	15	4E	44
	DA	F5	A7	66
	06	57	4C	1F
	47	05	85	60
Ronde 10	FE	EB	A5	E1
	1A	EF	48	2E
	D6	81	CD	D2
	5C	59	DC	BC

3.1.5.2 Analisis Proses Enkripsi AES

Proses enkripsi pada algoritma *Advanced Encryption Standard* (AES) terdiri dari empat operasi yaitu *Add Round Key, Sub Bytes, Shift Rows,* dan *Mix Columns*. Operasi-operasi ini diulang terus-menerus hingga menghasilkan *ciphertext*. Jumlah perulangan yang dilakukan tergantung pada ukuran blok dan kunci yang digunakan, dalam hal ini ukuran blok dan kunci yang digunakan yaitu 128 bit, sehingga berdasarkan pada teori, maka perulangan/ronde yang dilakukan sebanyak 10 kali.

Contoh analisis enkripsi pada algoritma AES (*Advanced Encryption Standard) 128 bit*, jika diketahui kunci dan plaintext yang akan digunakan untuk enkripsi dengan panjang 16 byte, sebagai berikut :

Plaintext : **53 54 4D 49 | 4B 20 49 4D | 20 42 44 47 | 20 4F 4B 01**

Cipherkey : **C5 14 2B 1D | 55 2E 69 A4 | B4 4B 8B B3 | 37 69 E0 26**

Langkah pertama masukan *cipherkey* dan *plaintext* ke dalam masing-masing blok array 4x4 (16 byte) yang disebut *state* sehingga menjadi :

State =

53	4B	20	20
54	20	42	4F
4D	49	44	4B
49	4D	47	**01**

Cipherkey =

C5	55	B4	37
14	2E	4B	69
2B	69	8B	E0
1D	A4	B3	26

Karena *plaintext* kurang dari 16 byte, maka untuk memenuhi blok AES dilakukan padding / penambalan dengan 01.

Kemudian *Cipherkey* dan *plaintext* yang telah dimasukan kedalam blok selanjutnya, lakukan operasi-operasi enkripsi pada algoritma *Advanced Encryption Standard*, sebagai berikut :

1. Pra Ronde (*Initial Round)*

   Pada *initial round* (rode ke-0), Melakukan operasi *AddRoundKey* yaitu dengan melakukan operasi XOR pada setiap kolom di *state* dengan kolom di *cipherkey*, sehingga menghasilkan *state* baru seperti berikut :

State					*Cipherkey*					Hasil *AddroundKey*			
53	4B	20	20	⊕	C5	55	B4	37	=	96	1E	94	17
54	20	42	4F		14	2E	4B	69		40	0E	09	26
4D	49	44	4B		2B	69	8B	E0		66	20	CF	AB
49	4D	47	01		1D	A4	B3	26		54	E9	F4	27

   Hasil *AddRoundKey* : 96 40 66 54 | 1E 0E 20 E9 | 94 09 CF F4 | 17 26 AB 27

2. *SubBytes*

   Selanjutnya proses ronde ke-1, yaitu *State* yang telah dilakukan operasi *AddRoundKey* tersebut dilakukan perulangan dengan urutan operasi pertama yaitu operasi *SubByte*. Operasi ini yaitu melakukan subtitusi *state* dengan tabel *s-box* pada Tabel : 2.3, sehingga menghasilkan *state* baru :

90	72	22	F0
09	AB	01	F7
33	B7	8A	62
20	1E	BF	CC

   Hasil *SubBytes* : 90 09 33 20 | 72 AB B7 1E | 22 01 8A BF | F0 F7 62 CC

3. *ShifRows*

   Selanjutnya hasil dari operasi *SubByte* dilakukan operasi *ShiftRows* yaitu menjalankan operasi *sirkular left* sebanyak *i* pada baris ke-*i* pada *state*, sebagai berikut :

State hasil *SubBytes*

90	72	22	F0
09	AB	01	F7
33	B7	8A	62
20	1E	BF	CC

ShiftRows →

State hasil *ShiftRows*

90	72	22	F0
AB	01	F7	**09**
8A	62	**33**	**B7**
CC	**20**	**1E**	**BF**

Hasil *ShiftRows* : 90 AB 8A CC | 72 01 62 20 | 22 F7 33 1E | F0 09 B7 BF

4. *MixColumns*

Selanjutnya, melakukan operasi *MixColumns* yaitu melakukan perkalian tiap kolom pada *state* dengan matriks seperti berikut :

90	72	22	F0
AB	01	F7	09
8A	62	33	B7
CC	20	1E	BF

•

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

=

9B	A5	6B	E8
94	F6	9C	9F
7B	D7	91	56
09	B5	9E	D0

Hasil *MixColumns* : 9B 94 7B 09 | A5 F6 D7 B5 | 6B 9C 91 9E | E8 9F 56 D0

5. *AddRoundKey*

Kemudian, melakukan *AddRoundKey* kembali dengan menggunakan operasi XOR Kunci ronde (*SubKey*) ke-1 hasil dari penjadwalan kunci *Cipherkey* dengan hasil *MixColumns*.

State

9B	A5	6B	E8
94	F6	9C	9F
7B	D7	91	56
09	B5	9E	D0

⊕

Cipherkey 1

3D	68	DC	EB
F5	DB	90	F9
DC	B5	3E	DE
87	23	90	B6

=

Hasil *AddroundKey*

A6	CD	B7	03
61	2D	0C	66
A7	62	AF	88
8E	96	0E	66

Hasil *AddRoundKey* : A6 61 A7 8E | CD 2D 62 96 | B7 0C AF 0E | 03 66 88 66

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

3.2. Perancangan Sistem

Dalam perancangan aplikasi, untuk menggambarkan proses-proses perancangan aplikasi enkripsi pesan singkat SMS (*Short Message Service*) digunakan beberapa model diagram, yaitu : *use case diagram, activity diagram, sequence diagram, deployment diagram dan class diagram*.

3.2.1 Pemodelan *Use Case Diagram*

Dalam *use case diagram* ini menjelaskan apa yang dilakukan oleh sistem dan siapa saja yang berinteraksi dengan sistem. Dalam pemodelan *use case* ini terbagi menjadi empat bagian, yaitu identifikasi aktor, identifikasi kebutuhan *use case*, *use case diagram* dan skenario *use case*.

1. Identifikasi Aktor

   Berikut adalah deskripsi pendefinisian aktor pada aplikasi enkripsi SMS (*Short Message Service*) :

   Tabel : 3.6 Identifikasi Aktor

No	Aktor	Deskripsi
1	User (Pengirim /Penerima)	Aktor yang memiliki akses buat pesan, baca pesan, balas pesan, pilih kontak, pilih pesan, enkripsi pesan, dekripsi pesan, input nomor tujuan, tulis pesan, input kunci enkripsi, input kunci dekripsi.

2. Identifikasi Kebutuhan *Use Case*

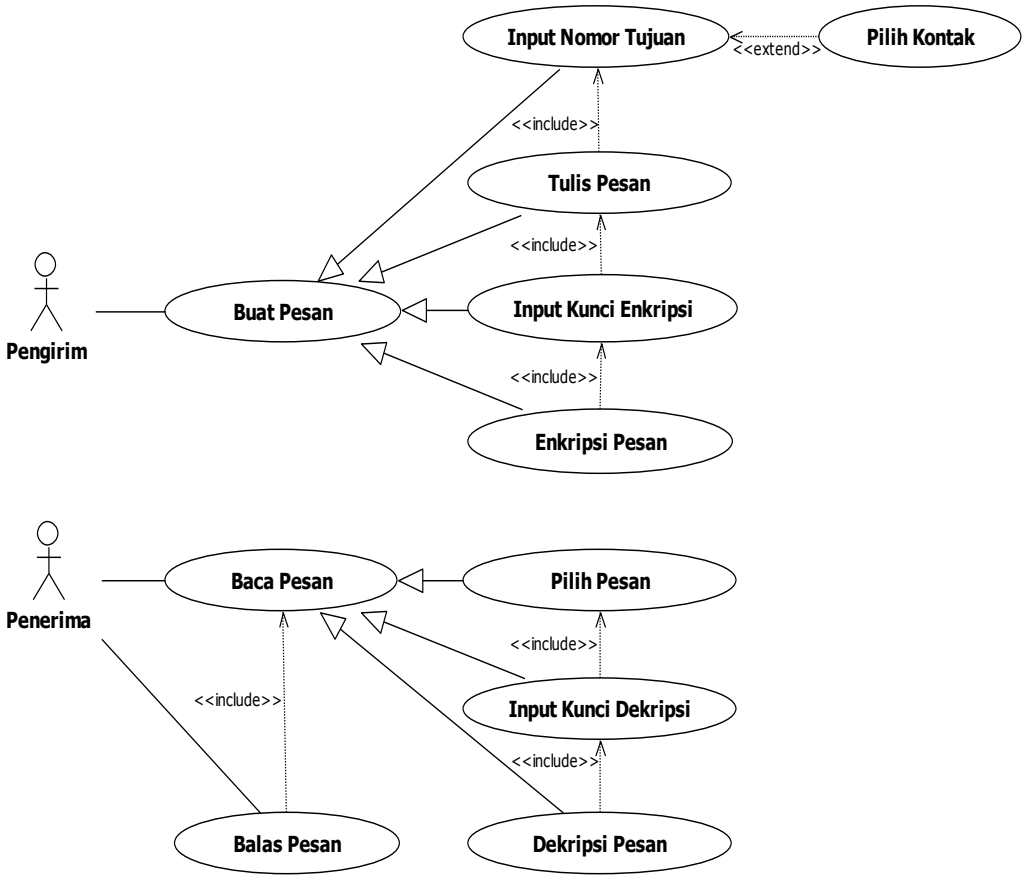
   Berikut adalah deskripsi pendefinisian use case pada aplikasi enkripsi SMS (*Short Message Service*) :

Tabel : 3.7 Identifikasi Kebutuhan *Use Case*

No	*Use Case*	Deskripsi	Aktor
1	Buat pesan	*Use case* ini menggambarkan proses penulisan pesan untuk dikirim, termasuk input nomor tujuan, pilih kontak, tulis pesan, input kunci enkripsi, enkripsi pesan	*User*
2	Baca pesan	*Use case* ini menggambarkan proses penerimaan pesan untuk dibaca, termasuk pilih pesan, input kunci dekripsi, dekripsi pesan	*User*
3	Balas pesan	*Use case* ini menggambarkan proses balas pesan	*User*

3.2.1.1 *Use Case Diagram*

Pada Gambar 3.1. menunjukan *use case diagram* yang membentuk fungsionalitas dari aplikasi enkripsi SMS (*Short Message Service*) Aman Aes.



Gambar : 3.1. *Use Case Diagram* Aplikasi Enkripsi Sms Aman Aes

3.2.1.2 *Use Case* Skenario

Use case skenario digunakan untuk menjelaskan aktifitas secara rinci dari use case diagram yang telah digambarkan. Berikut *use case* scenario dari masing-masing *use case* yang telah didefinisikan sebelumnya :

Tabel : 3.8 *Use Case* Skenario Buat Pesan

Nama Use case	: Buat Pesan
Deskripsi	: Untuk menulis, mengirim pesan text yang terenkripsi
Aktor yang terlibat	: User
Kondisi Awal	: User ingin mengirim pesan sms yang terenkripsi
Kondisi Akhir	: Pesan sms terenkripsi terkirim
Skenario	
Aktor	Respon Sistem
1. User memasukkan nomor tujuan atau memilih kontak	
2. User menulis pesan teks	
3. User memasukkan kunci enkripsi	
4. User menekan tombol enkripsi	5. Sistem melakukan proses enkripsi
	6. Sistem menampilkan hasil enkripsi pesan
7. User menekan tombol kirim pesan	8. Sistem mengirimkan pesan

Tabel : 3.9 *Use Case* Skenario Baca Pesan

Nama Use case	: Baca Pesan
Deskripsi	: Untuk membaca pesan teks enkripsi yang masuk dan mendeskripsikan kembali
Aktor yang terlibat	: User
Kondisi Awal	: User ingin membaca pesan terenkripsi
Kondisi Akhir	: Menampilkan dekripsi pesan teks yang terenkripsi
Skenario	
Aktor	Respon Sistem
1. User memilih pesan masuk yang akan didekripsi	2. Menampilkan pesan yang terenkripsi
3. User memasukkan kunci untuk dekripsi	
4. User menekan tombol dekripsi	5. Memproses Dekripsi
	6. Menampilkan pesan teks hasil dekripsi

Tabel : 3.10 *Use Case* Skenario Balas Pesan

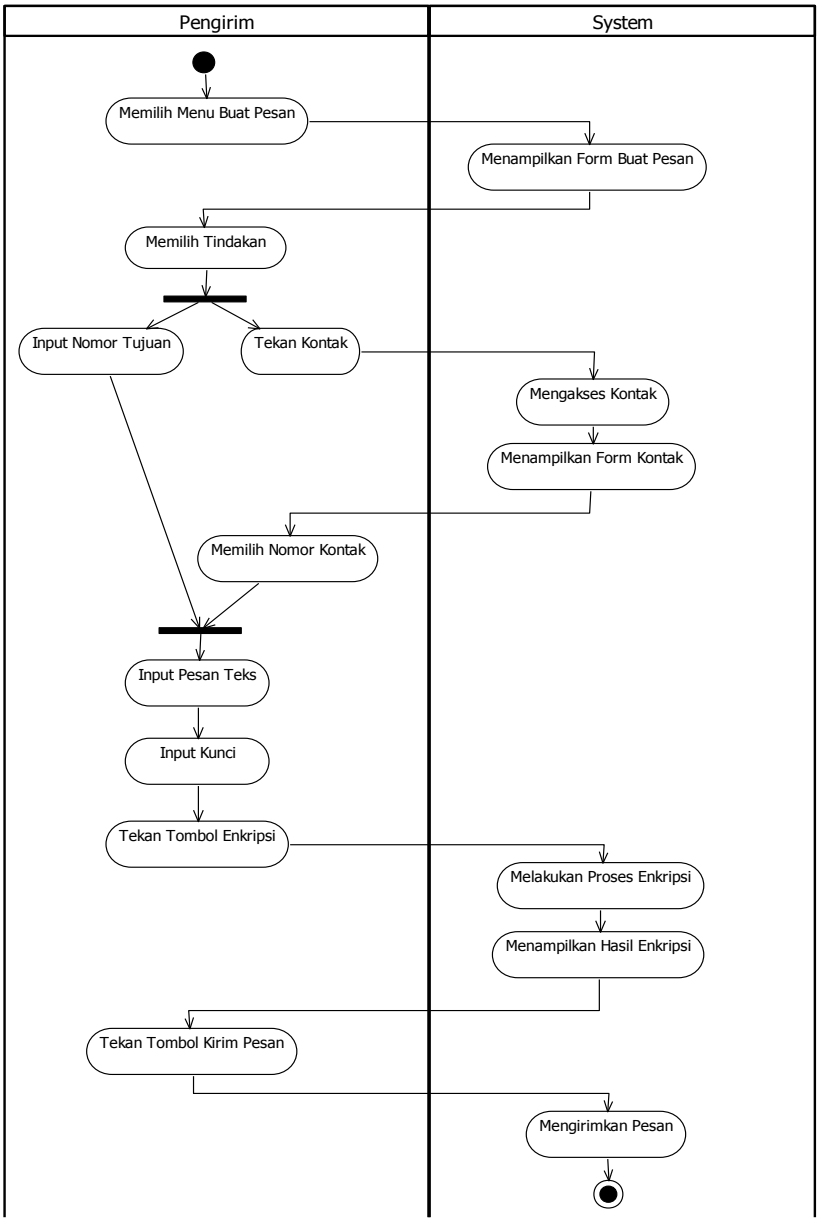
Nama Use case	: Balas Pesan
Deskripsi	: Untuk membalas pesan sms yang masuk
Aktor yang terlibat	: User
Kondisi Awal	: User telah membaca pesan yang telah di dekripsi
Kondisi Akhir	: Sistem menampilkan form buat pesan dengan nomor pengirim sebagai nomor tujuan
Skenario	
Aktor	Respon Sistem
1. User menekan tombol balas	2. Sistem menampilkan form buat pesan
	3. Sistem mengisi nomor telpon pengirim sebagai nomor telpon tujuan

3.2.2 Perancangan *Activity Diagram*

Activity diagram menggambarkan aktivitas-aktivitas terperinci yang terjadi pada sistem yang tidak cukup digambarkan oleh *use case diagram.*

Berikut activity diagram pada aplikasi enkripsi Sms Aman Aes :

1. *Activity Diagram* Buat Pesan



Gambar : 3.2 *Activity Diagram* Buat Pesan

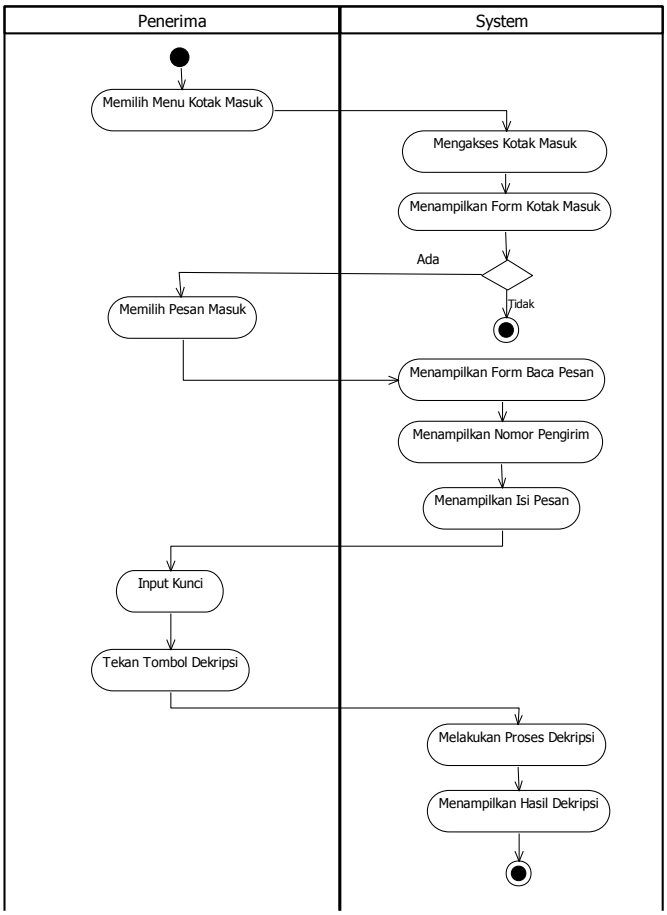
Penjelasan *Activity Diagram* Buat Pesan :

Aktifitas yang terjadi pada Gambar 3.2. adalah aktifitas aktor pengirim yang sedang melakukan proses pengiriman pesan yang di enkripsi.

1. Aktifitas dimulai setelah aktor memilih menu buat pesan
2. Kemudian sistem akan menampikan form buat pesan
3. Setelah itu, aktor memasukkan nomor tujuan penerima atau dapat menekan tombol kontak untuk mengakses kontak pada *Smartphone*

4. Setelah input kontak, kemudian aktor menuliskan isi pesan yang akan disampaikan kepada penerima pesan
5. Kemudian aktor menulis kunci yang digunakan untuk melakukan enkripsi
6. Untuk melakukan enkripsi aktor diharuskan menekan tombol enkripsi, Kemudian sistem akan melakukan enkripsi pesan yang telah di tuliskan oleh aktor menggunakan metode AES (*Advanced Encryption Standard*) 128 bit dengan menggunakan kunci yang telah dituliskan oleh aktor, dan sistem akan menampilan hasil enkripsi isi pesan.
7. Kemudian aktor menekan tombol kirim pesan untuk mengirimkan isi pesan, dan sistem akan melakukan pengiriman pesan ke nomor yang telah di tulis atau dipilih dari kontak oleh aktor.

2. *Activity Diagram* Baca Pesan



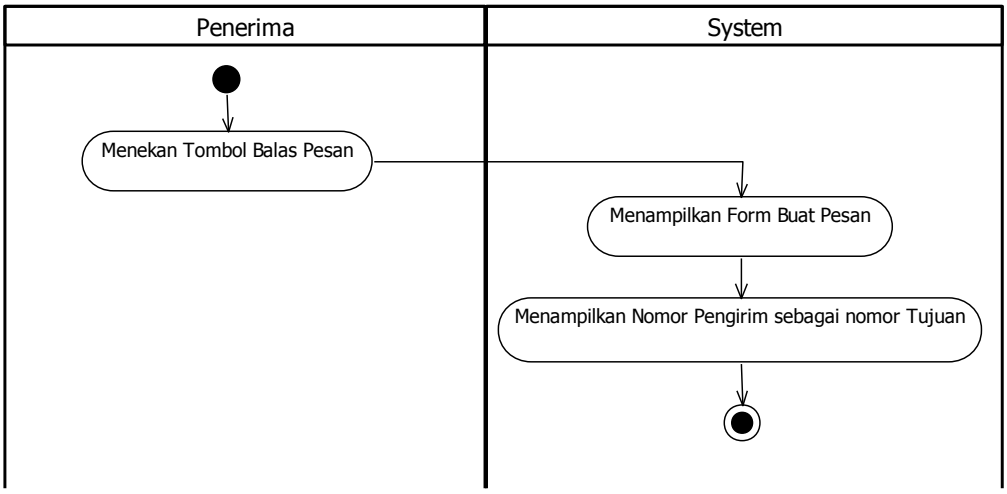
Gambar : 3.3 *Activitiy Diagram* Baca Pesan

Penjelasan *Activity Diagram* Baca Pesan :

Aktifitas yang terjadi pada Gambar 3.3. adalah aktifitas aktor penerima yang sedang membaca isi pesan.

1. Aktifitas dimulai setelah aktor masuk memilih menu kotak masuk.
2. Sistem akan mengakes kotak masuk pesan pada *Smartphone*, dan kemudian menampilkan kotak masuk pesan.
3. Kemudian *user* memilih pesan yang akan dibaca, jika tidak ada pesan yang masuk maka sistem tidak akan menampilkan form baca pesan.
4. Sistem menampilkan form baca pesan, menampilkan nomor pengirim dan menampilkan isi pesan.
5. Selanjutnya, *user* memasukkan kunci pesan dan menekan tombol dekripsi untuk melakukan dekripsi isi pesan.
6. Kemudian sistem akan melakukan dekripsi isi pesan menggunakan metode AES (*Advanced Encryption Standard*) 128 bit dengan menggunakan kunci yang telah dimasukkan oleh *user*, dan kemudian sistem akan menampilkan hasil dekripsi pesan teks.

3. *Activity Diagram* Balas Pesan



Gambar : 3.4 *Activitiy Diagram* Balas Pesan

Penjelasan *Activity Diagram* Balas Pesan :

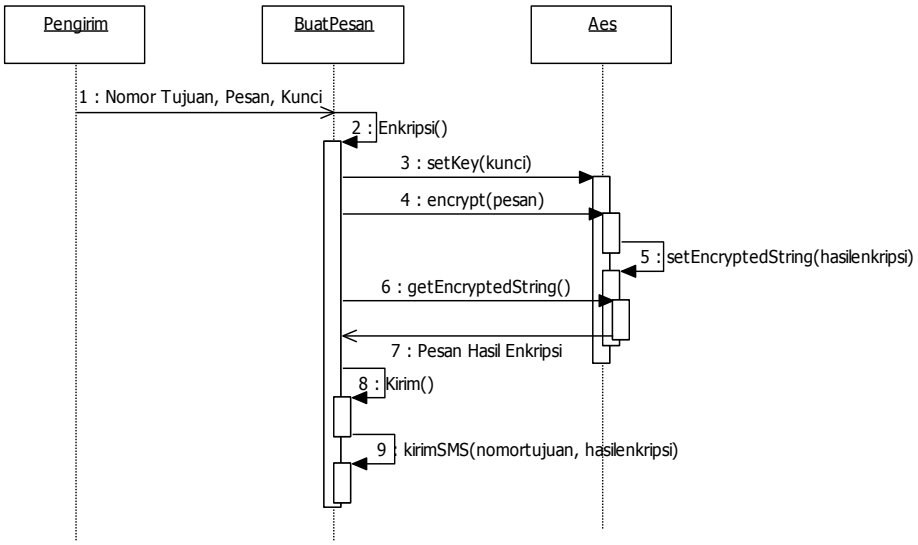
Aktifitas yang terjadi pada Gambar 3.4. adalah aktifitas aktor penerima yang hendak membalas pesan :

1. Aktifitas dimulai setelah *user* membaca pesan hasil dekripsi
2. User menekan tombol balas pesan
3. Kemudian sistem menampilkan form buat pesan
4. Sistem menampilkan nomor pengirim sebagai nomor tujuan

3.2.3 Perancangan *Sequence Diagram*

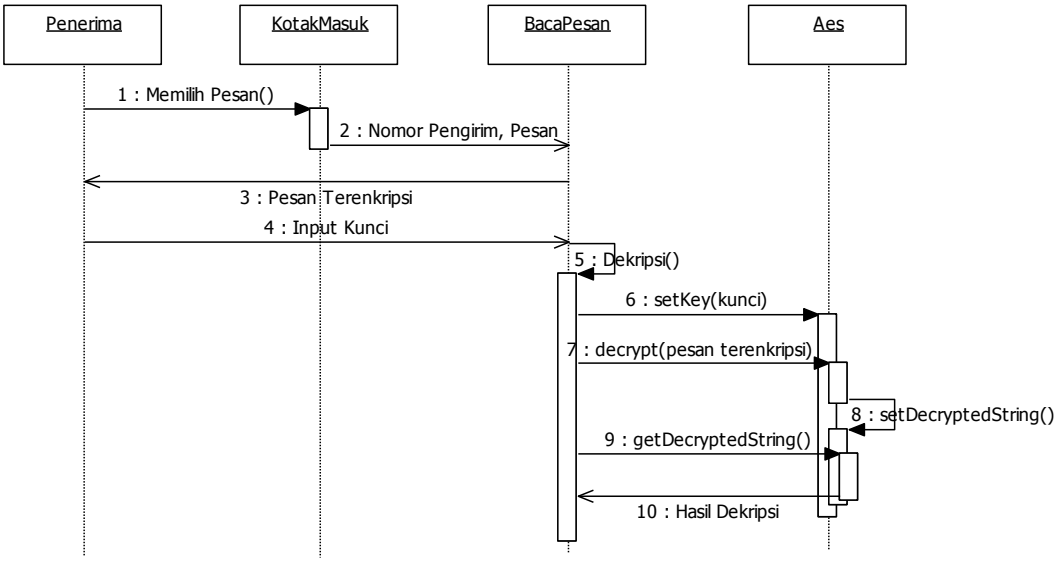
Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu.

1. *Sequence Diagram* Buat Pesan



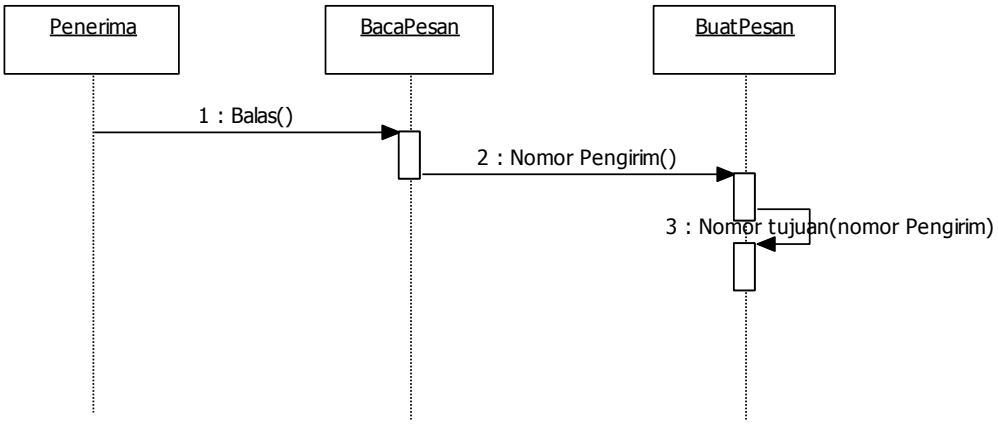
Gambar : 3.5. *Sequence Diagram* Buat Pesan

2. *Sequence Diagram* Baca Pesan



Gambar : 3.6. *Sequence Diagram* Baca Pesan

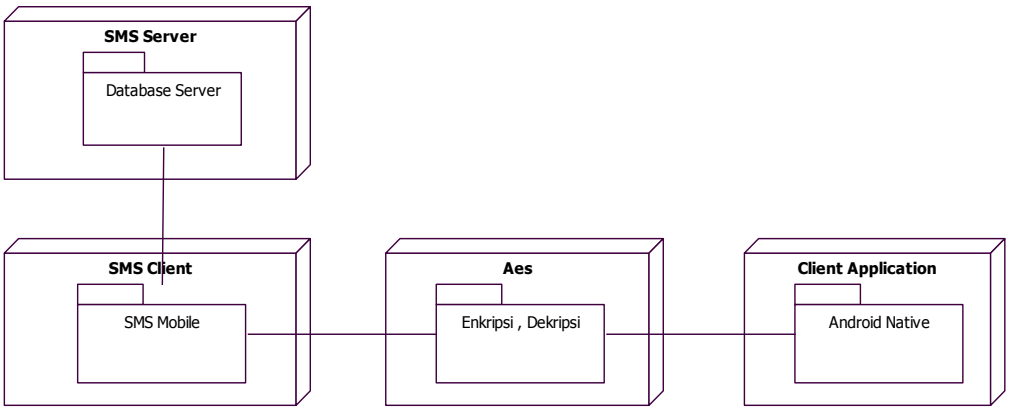
3. *Sequence Diagram* Balas Pesan



Gambar : 3.7. *Sequence Diagram* Balas Pesan

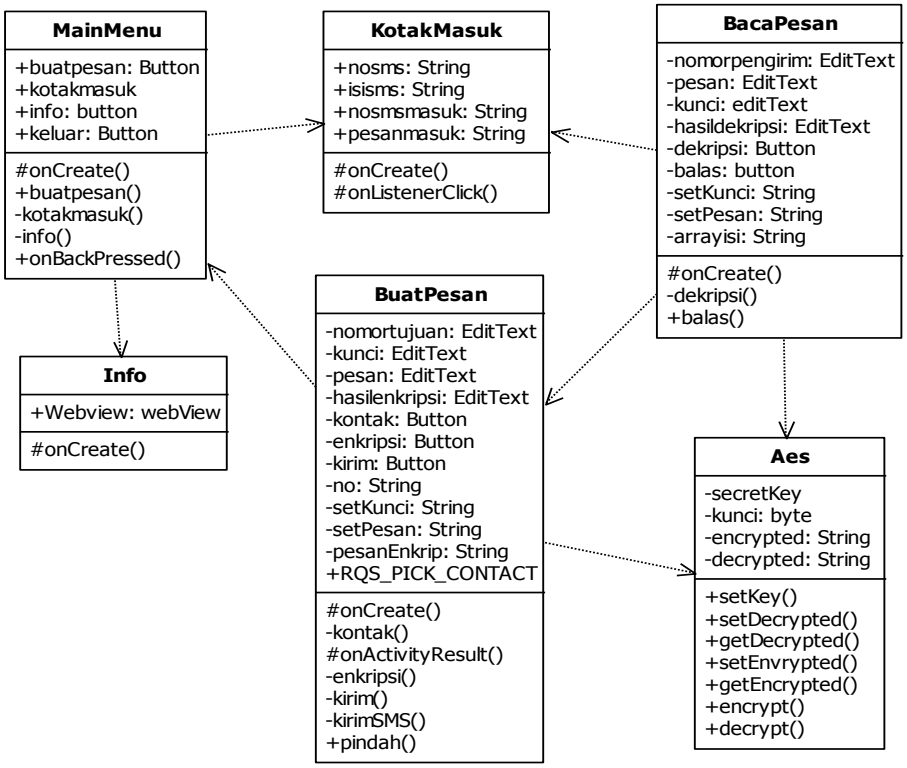
3.2.4 *Deployment Diagram*

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Berikut merupakan *deployment diagram* dari aplikasi enkripsi Sms Aman Aes :



Gambar : 3.8. *Deployment Diagram*

3.2.5 *Class Diagram*



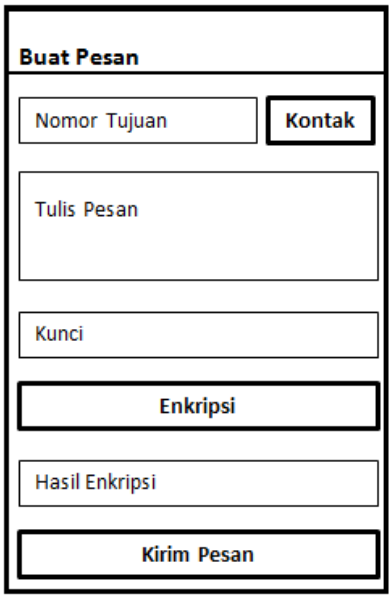
Gambar : 3.9. *Class Diagram* Aplikasi Enkripsi Sms Aman Aes

3.2.6 Perancangan Antar Muka

1. Perancangan Form Menu Utama

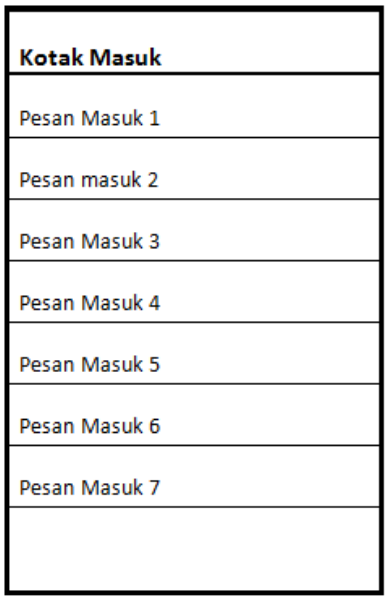
Gambar : 3.10. Perancangan Form Menu Utama

2. Perancangan Form Buat Pesan



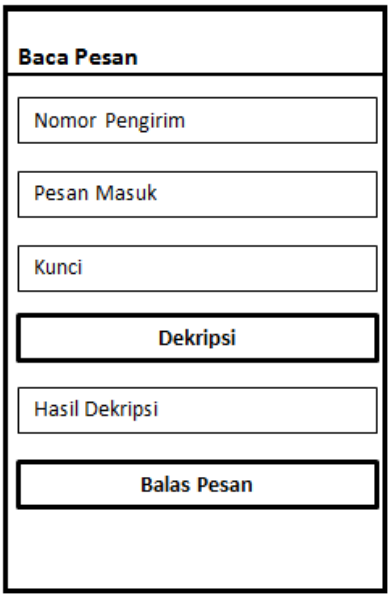
Gambar : 3.11. Perancangan Form Buat Pesan

3. Perancangan Form Kotak Masuk



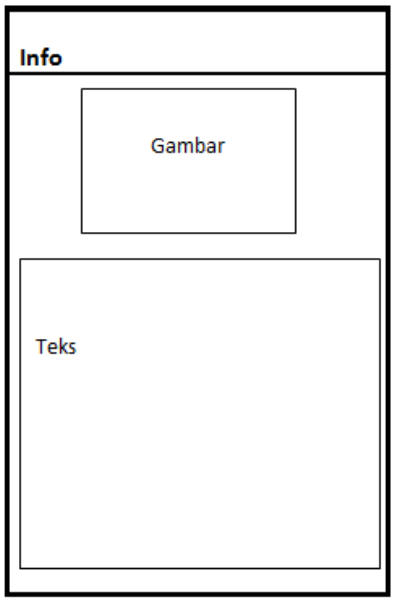
Gambar : 3.12. Perancangan Form Kotak Masuk

4. Perancangan Form Baca Pesan



Gambar : 3.13. Perancangan Form Baca Pesan

5. Perancangan Form Info



Gambar : 3.14 Perancangan Form Info

4. KESIMPULAN

Berdasarkan hasil analisa, maka dapat disimpulkan sebagai berikut :

1. Aplikasi ini dibuat untuk melakukan perlindungan terhadap pesan teks SMS (*Short Message Service*) yang bersifat rahasia, perlindungan tersebut di implementasikan dengan melakukan metode enkripsi dan juga dekripsi menggunakan metode AES (*Advanced Encryption Standard*) 128 bit.
2. Aplikasi SMS ini memiliki kemampuan untuk mengakses daftar kontak, kotak masuk pesan, menerima input pesan teks, menerima input kunci enkripsi dan dekripsi serta melakukan pengiriman pesan teks yang telah terenkripsi pada *smartphone* android.
3. Aplikasi ini yang dibangun dapat berjalan di *smartphone* android, untuk menggunakannya dapat menginstall AplikasiSmsAmanAes.apk yang penulis buat.
4. Dengan menggunakan Aplikasi SMS Aman Aes ini, pesan yang dikirim oleh para pengguna layanan SMS (*Short Message Service*) dapat terjaga keamanan serta kerahasiaan pesannya.

5. DAFTAR PUSTAKA

H.M, Jogiyanto, 2010. **Analisis dan Desain Sistem Informasi**. Yogyakarta : Andi Offset.

Rosa, A.S., Shalahuddin, M., 2013. **Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek).** Bandung : Informatika.

Sadikin, Rifki. 2012. **Kriptografi Untuk Keamanan Jaringan**. Yogyakarta : Andi Offset.

Safaat, Nazaruddin. 2014. **Pemrograman Aplikasi Berbasis Mobile Smartphone dan Tablet PC Berbasis Android**. Bandung : Informatika.

Safaat, Nazruddin. 2013. **Aplikasi Berbasis Android**. Bandung : Informatika.

Supriyanto, Dodit Dan Rini Agustina. 2012. **Pemrograman Aplikasi Android**. Yogyakarta : Mediakom.