ENKRIPSI DAN DESKRIPSI MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD)* UNTUK IMPLEMENTASI SMS (*SHORT MESSAGE SERVICE*)

TUGAS AKHIR

Oleh :

Maria Olva 3311201043

Disusun untuk memenuhi syarat kelulusan Program Diploma III

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM

2015

LEMBAR PENGESAHAN

ENKRIPSI DAN DESKRIPSI MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD)* UNTUK IMPLEMENTASI SMS (*SHORT MESSAGE SERVICE*)

Oleh :

Maria Olva 3311201043

Tugas Akhir ini telah diterima dan disahkan

sebagai persyaratan untuk memperoleh gelar

Ahli Madya

di

PROGRAM STUDI DIPLOMA 3 TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

Batam, 13 Januari 2015

Disetujui oleh :

Pembimbing,

Meyti Eka Apriyani, M.T
NIK. 111081

HALAMAN PERNYATAAN

Dengan ini, saya :

NIM : 3311201043

Nama : Maria Olva

adalah mahasiswa Teknik Informatika Politeknik Negeri Batam yang menyatakan bahwa tugas akhir dengan judul :

ENKRIPSI DAN DESKRIPSI MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD)* UNTUK IMPLEMENTASI SMS (*SHORT MESSAGE SERVICE*)

Disusun dengan :

1. Tidak melakukan plagiat terhadap naskah karya orang lain
2. Tidak melakukan pemalsuan data
3. Tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan diatas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Negeri Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, Januari 2015

Maria Olva
3311201043

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang telah memberikan anugerah dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini yang berjudul "ENKRIPSI DAN DESKRIPSI MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD)* UNTUK IMPLEMENTASI SMS (*SHORT MESSAGE SERVICE*)".

Pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu proses penyelesaian Tugas Akhir ini diantaranya adalah :

1. Kepada Allah SWT atas selesainya Tugas Akhir ini.
2. Kedua orang tua tercinta yang selalu memberikan doa, perhatian serta dukungan kepada penulis.
3. Bapak Dr. Ir. Priyono Eko Sanyoto selaku Direktur Politeknik Negeri Batam.
4. Ibu Meyti Eka Apriyani, M.T selaku dosen pembimbing tugas akhir yang telah membimbing penulis hingga dapat menyelesaikan tugas akhir ini.
5. Teman-teman seperjuangan Teknik Informatika angkatan 2012 atas dukungan dan bantuannya.

Dalam penulisan ini, penulis masih terdapat kekurangan dalam penyusunannya. Untuk itu, penulis menharapkan kritik dan saran yang membangun dari berbagai pihak-pihak lain.

Batam, Januari 2015

Penulis

ABSTRAK

ENKRIPSI DAN DESKRIPSI MENGGUNAKAN ALGORITMA AES (*ADVANCED ENCRYPTION STANDARD)* UNTUK IMPLEMENTASI SMS (*SHORT MESSAGE SERVICE*)

Saat ini penggunaan *Smartphone* sangat digemari oleh banyak kalangan, khususnya *Smartphone* berbasis Android. Salah satu fitur *Smartphone* yang juga banyak diminati oleh masyarakat adalah SMS. Layanan pesan singkat (SMS) adalah salah satu dari pertukaran pesan media komunikasi yang populer digunakan meskipun banyak aplikasi pertukaran pesan berbasis internet cenderung lebih hemat dan mudah tetapi masih terhalang oleh jaringan yang tidak stabil. Pengiriman suatu informasi dengan menggunakan SMS memerlukan suatu proses yang menjamin keamanan pesan yang dikirim. Kriptografi merupakan salah satu solusi yang dapat dimanfaatkan dan dikembangkan dalam menghadapi permasalahan tentang keamanan pesan SMS. Dengan melakukan enkripsi pada pesan SMS, maka kemanan pesan SMS akan lebih terjaga dan aman. Kriptografi memiliki banyak teknik dalam melakukan pengenkripsian pesan SMS dan salah satu yang memiliki tingkat keamanan yang tinggi dan tingkat kesulitan dalam pemecahannya adalah algoritma AES (*Advanced Encryption Standard)*. Pada saat melakukan pengenkripsian pesan harus memasukkan kunci yang akan mengamankan pesan SMS. Hal ini dapat mengurangi bocornya informasi kepada pihak-pihak yang tidak berkepentingan seperti operator telepon seluler atau pihak lainnya.

Kata Kunci : *Smartphone*, SMS, Kriptografi, Enkripsi, Deskripsi, AES.

ABSTRACT

ENCRYPTION and DESCRYPTION USE AES (*ADVANCED ENCRYPTION STANDARD*) ALGORITHMS FOR IMPLEMENTATION OF SMS (SHORT MESSAGE SERVICE)

Currently Smartphone use is very popular with many people, especially Android-based smartphones. One Smartphone features that are also much in demand by the public is SMS. Short message service (SMS) is one of the popularly used message exchange communication medias although many internet based message exchange application tend to be more thrifty and easy but still obstructed by unstable networks. Delivery of an information by using SMS requires a process that ensures the security of messages sent. Cryptography is one solution that can be utilized and developed in dealing with the issue of security of SMS messages. By encrypting the SMS message, then the security of SMS messages will be maintained and safe. Cryptography has many techniques for performing encryption SMS messages and one that has a high level of security and the level of difficulty in solving algorithm is AES (Advanced Encryption Standard). At the time encryption must enter a message that will secure the lock SMS messages. This can reduce the leaking of information to parties who are not interested as a mobile phone operator or others.

***Keywords* :** *Smartphone, SMS, Cryptography, Encryption, Decryption, AES*

DAFTAR ISI

HALAMAN PENGESAHAN …… II
HALAMAN PERNYATAAN …… III
KATA PENGANTAR …… IV
ABSTRAK ……... V
ABSTRACT …… VI
DAFTAR ISI ……... VII
DAFTAR GAMBAR ……... X
DAFTAR TABEL ……. XI
BAB I PENDAHULUAN…….. 1
1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah..... 2
1.3 Batasan Masalah..... 2
1.4 Tujuan Masalah..... 2
1.5 Sistematika Penulisan..... 3
BAB II TINJAUAN PUSTAKA……. 4
2.1 Perbandingan dengan penelitian sebelumnya..... 4
2.2 Perbandingan dengan algoritma lain.....……...4
2.3 Short Message Service (SMS)..... 6
2.4 Kriptografi ……... 6
2.5 Perbandingan SMS Konvensional dan SMS Enkripsi …….. 8
2.6 Algoritma Advanced Encryption Standard (AES)……. 10
2.7 Android……..…… 18
2.8 Android SDK (Software Development Kit)…… 19
2.9 Eclipse……20
BAB III ANALISIS dan PERANCANGAN……..21
3.1 Deskripsi Umum Sistem…… 21
3.2 Analisis Kebutuhan Sistem…… 21
3.2.1 Analisis Kebutuhan Software……. 22
3.2.2 *Analisis Kebutuhan Hardware*……..... 22

3.3 Kebutuhan Fungsioanl……. 23
3.4 Kebutuhan Non Fungsional……. 23
3.5 Diagram *Use Case* ……. 24
3.5.1 Skenario *Use Case* Tulis Pesan…… 24
3.5.2 Skenario *Use Case* Enkripsi…….... 24
3.5.3 Skenario *Use Case* Kirim……..24
3.5.4 Skenario *Use Case* Terima Pesan…… 25
3.5.5 Skenario *Use Case* Deskripsi Pesan……..….. 25
3.5.6 Skenario *Use Case* Baca Pesan.…..…… 25
3.5.7 Skenario *Use Case Inbox*……. 25
3.5.8 Skenario *Use Case* Bantuan……. 26
3.6 Analisis Kelas ……... 26
3.7 Sequence Diagram …… 29
3.7.1 Sequence Diagram Tulis Pesan……. 29
3.7.2 Sequence Diagram Enkripsi dan Kirim Pesan…..……..30
3.7.3 Sequence Diagram Terima Pesan ….……..…….. 31
3.7.4 Sequence Diagram Deskripsi Pesan…..…….…… 31
3.7.5 Sequence Diagram *Inbox*…..…….…… 32
3.7.6 Sequence Diagram Bantuan …..…….....…….. 33
3.8 *Class* Diagram…….. 33
3.9 Algoritma…….. 34
3.9.1 Algoritma Kelas Tulis Pesan……. 34
3.9.2 Algoritma Kelas Tulis Pesan……... 34
3.9.3 Algoritma Kelas Baca Pesan……... 35
3.10 Implementasi AES Pada Aplikasi …….…….. 35
3.11 Perancangan Antarmuka ……...…….. 36
3.11.1 Rancangan Antar Muka Halaman Menu Utama……. 36
3.11.2 Rancangan Antar Muka Tulis Pesan……. …….. 37
3.11.3 Rancangan Antar Muka Terima Pesan…….. 38
3.11.4 Rancangan Antar Muka Bantuan…….. 39

BAB IV IMPLEMENTASI dan PENGUJIAN……. 40
4.1 Implementasi Kelas ….……..... 40
4.2 Implementasi Antarmuka ……... 40
4.2.1 Implementasi Menu Utama …….……... 41
4.2.2 Implementasi Tulis Pesan …….…….... 41
4.2.3 Implementasi *Inbox*/Baca Pesan.…..….……. 42
4.2.4 Implementasi Bantuan.…...…….……..... 43
4.3 Hasil Pengujian ….…...…….……..... 44
BAB V KESIMPULAN dan SARAN……. 46
5.1 Kesimpulan ……..…..……..... 46
5.2 Saran …… ……..... 46
DAFTAR PUSTAKA ……. 48
LAMPIRAN
RIWAYAT HIDUP

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Proses Enkripsi AES…… 11
Gambar 2.2 Matriks Affine ……. 12
Gambar 2.3 Transformasi *ShiftRows*……..... 13
Gambar 2.4 Matriks Transformasi *MixColumns*…… 13
Gambar 2.5 Hasil perkalian dari operasi matriks MixColumns……. 13
Gambar 2.6 Ilustrasi Proses Deskripsi AES ……. 14
Gambar 2.7 Transformasi *InvShiftRows*…….. 15
Gambar 2.8 Matriks Invers Affine…….. 15
Gambar 2.9 Matriks InvMixColumns…… 16
Gambar 2.10 Hasil Perkalian Matriks……. 16
Gambar 3.1 Arsitektur Global Sistem…… 21
Gambar 3.2 Diagram Use Case …….. 24
Gambar 3.3 Analisis Kelas Enkripsi dan Deskripsi Menggunakan Algoritma AES untuk Implementasi SMS…… 26
Gambar 3.4 Sequence Diagram Tulis Pesan……..29
Gambar 3.5 Sequence Diagram Enkripsi & Kirim Pesan…… 30
Gambar 3.6 Sequence Diagram Terima Pesan…… 31
Gambar 3.7 Sequence Diagram Deskripsi Pesan…… 31
Gambar 3.8 Sequence Diagram Inbox……..32
Gambar 3.9 Sequence Diagram Bantuan…… 33
Gambar 3.10 Class Diagram…… 33
Gambar 3.11 Tampilan Halaman Utama Aplikasi……. 36
Gambar 3.12 Tampilan AntraMuka Tulis Pesan…… 37
Gambar 3.13 Tampilan AntarMuka Terima Pesan…… 38
Gambar 3.14 Tampilan AntarMuka Bantuan……. 39
Gambar 4.1 Antarmuka Menu Utama……. 41
Gambar 4.2 Antarmuka Tulis Pesan…….. 41
Gambar 4.3 Antarmuka Inbox/Baca Pesan…… 42
Gambar 4.4 Antarmuka bantuan……. 43

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian …… 4

Tabel 2.2 Perbandingan dengan Algortima Lain…… 5

Tabel 2.3 Perbandingan SMS Konvensional dan SMS Enkripsi…… 8

Tabel 3.1 *Hardware Komputer*……..... 22

Tabel 3.2 *Hardware Mobile*……...……. 23

Tabel 3.3 Kebutuhan Fungsional…….. 23

Tabel 3.4 Kebutuhan Non Fungsional…….. 23

Tabel 3.5 Rincian Kelas Aplikasi…….. 26

Tabel 3.6 Deskripsi Tampilan Antarmuka Halaman Utama…… 37

Tabel 3.7 Deskripsi Tampilan Antarmuka Tulis Pesan…… 38

Tabel 3.8 Deskripsi Tampilan Antarmuka Terima Pesan…… 39

Tabel 3.9 Deskripsi Tampilan Antarmuka Bantuan…… 39

Tabel 4.1 Implementasi kelas…… 40

Tabel 4.2 Implementasi Antarmuka……..… 40

Tabel 4.3 Hasil Pengujian…… 44

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi saat ini mampu menghasilkan perangkat keras yang mempermudah dalam melakukan komunikasi. salah satu perangkat komunikasi yang sering digunakan adalah *Smartphone*. Perkembangan *smartphone* yang sedang populer saat ini adalah *smartphone* berbasis android. *Smartphone* sistem operasi android ini rata-rata memiliki spesifikasi *hardware* yang cukup baik sehingga aplikasi dan fitur banyak digunakan pada *smartphone* tersebut. Salah satu fitur dalam ponsel untuk melakukan pengiriman pesan yaitu *Short Message Service* (SMS).

SMS dapat digunakan baik untuk kebutuhan individu maupun bisnis. Penggunaan SMS telah mengalami pengembangan saat ini di berbagai lapisan masyarakat. Namun seiring dengan berkembangnya teknologi membuat SMS yang digunakan bisa saja diketahui orang lain dengan cara menyadapnya. Sebagai contoh terkadang beberapa orang bertukar informasi rahasia seperti *password* atau data penting lain melalui SMS. Bank di seluruh dunia menggunakan SMS untuk melakukan beberapa layanan perbankan seperti SMS Banking.Teknologi SMS memiliki beberapa resiko seperti penyadapan dan akses dari pihak yang tidak berwenang. Pesan yang dikirim menggunakan aplikasi SMS bawaan ponsel masih berupa teks terbuka yang belum terproteksi. Selain itu, pengiriman SMS yang dilakukan tidak sampai ke penerima secara langsung akan tetapi SMS yang dikirimkan harus melalui SMS *Centre* (SMSC). SMSC berfungsi mencatat komunikasi yang terjadi di antara pengirim dan penerima. Seorang operator dapat memperoleh informasi atau membaca SMS tersebut. Bukan hanya operator, pihak lain yang tidak berwenang pun dapat membaca pesan tersebut. Hal ini tentunya akan sangat berbahaya untuk pengguna yang menggunakan layanan SMS (Syaifullah M, 2014).

Karena itu, dibutuhkan suatu cara untuk mengamankan informasi yang sifatnya penting atau rahasia. Dengan melakukan enkripsi terhadap teks SMS, maka

tingkat keamanan informasi dari pesan tersebut dapat ditingkatkan. Saat ini, AES (*Advanced Encryption Standard*) digunakan sebagai standar algoritma kriptografi terbaru (Irwan,2012). AES menggantikan DES (*Data Encryption Standar*) yang pada tahun 2002 sudah berakhir masa penggunaannya. DES juga dianggap tidak mampu lagi untuk menjawab tantangan perkembangan teknologi komunikasi yang sangat cepat. AES sendiri adalah algoritma kriptografi dengan menggunakan algoritma Rijndael yang dapat mengenkripsi dan mendekripsi blok data sepanjang 128 bit dengan panjang kunci 128 bit, 192 bit, atau 256 bit. (Mubarak, 2012). Dengan memanfaatkan algoritma AES ini, maka dapat dikembangkan suatu aplikasi SMS yang memungkinkan pengguna untuk mengirimkan pesan singkat dengan enkripsi teks dan dapat melakukan dekripsi terhadap pesan terenkripsi. Aplikasi SMS ini akan dibangun berbasis *mobile* pada platform Android.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut maka dapat disusun rumusan masalah yaitu

1. Menerapkan jenis pengamanan informasi pesan pada media SMS berbasis android.
2. Mengimplementasikan metode algoritma AES sehingga pesan dapat terkirim dengan aman.

1.3 Batasan Masalah

Batasan masalah dalam Tugas Akhir ini yaitu :

1. Algoritma kriptografi yang digunakan adalah Algoritma AES.
2. Algoritma AES yang digunakan adalah AES 128 bit.
3. Aplikasi SMS dibangun untuk *smartphone* Android minimal versi 2.2 (Froyo).
4. Aplikasi tidak menangani distribusi kunci bagi pengirim maupun penerima.
5. Pengirim dan penerima harus memiliki aplikasi.

1.4 Tujuan

Berdasarkan permasalahan di atas, maka tujuan yang ingin dicapai adalah :

1. Aplikasi yang dibangun menerapkan algoritma AES untuk pengamanan informasi SMS berbasis android.
2. Hasil implementasi saat pengiriman dan pembacaan pesan menggunakan metode Algoritma AES.

1.5 Sistematika Penulisan

Sistematika dalam penulisan Tugas Akhir ini, meliputi :

BAB I **: PENDAHULUAN**, bagian ini dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II : **LANDASAN TEORI**, bagian ini diuraikan dan dijelaskan mengenai tinjauan pustaka dan dasar teori.

BAB III : **ANALISIS dan PERANCANGAN**, bagian ini dijelasakn mengenai langkah-langkah penyelesaian masalah.

BAB IV : **IMPLEMENTASI dan PENGUJIAN**, bagian ini memuat uraian langkah implementasi dan pengujian/validasi.

BAB V : **KESIMPULAN dan SARAN**, bagian ini memuat simpulan yang merupakan rangkuman dari hasil analisis kineja pada bagian sebelumnya serta berisi saran-saran pengembangan dari penelitian yang dibuat dan aspek yang belum terselesaikan.

BAB II
TINJAUAN PUSTAKA

Pada bab ini berisi tentang teori-teori yang berhubungan dengan Enkripsi dan Deskripsi Algoritma AES untuk Implementasi SMS berdasarkan penelitian-penelitian yang sudah ada, serta penelitian yang telah dilakukan.

2.1 Perbandingan dengan Penelitian Sebelumnya

Penelitian Sebelumnya sebagai referensi Tugas akhir ini adalah "Jurnal Perbandingan Algoritma AES dengan Algoritma XTS-AES untuk Enkripsi dan Dekripsi Teks SMS Berbasis Java ME" oleh Mariana dan Martha Sari di Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) MDP. Perbandingan antara penelitian sebelumnya dengan Tugas Akhir yang akan dibuat dapat dilihat dari Tabel 2.1 sebagai berikut :

Tabel 2.1 Perbandingan Penelitian

No	Perbandingan	Penelitian Sebelumnya	Tugas Akhir
1.	Algoritma Enkripsi	AES	AES
2.	Implementasi	SMS	SMS
3.	Sistem Operasi	Java ME	Android
4.	Bahasa Pemrograman	Java	Java

Berdasarkan Tabel 2.1 diatas mengenai perbandingan antara pembuatan tugas akhir dengan penelitian sebelumnya bahwa terdapat perbedaan dari segi sistem operasi yaitu pada penelitian sebelumnya sistem operasi yang digunakan adalah Java ME sedangkan untuk pembuatan tugas akhir ini sistem operasi yang digunakan adalah android.

2.2 Perbandingan dengan Algoritma Lain

Untuk mengetahui efektivitas dari algoritma AES maka disajikan Tabel 2.2 yang merupakan perbandingan antara algoritma AES dengan algoritma lain yaitu algoritma Caesar Chiper. Perbandingan diukur berdasarkan parameter yang dimiliki oleh masing-masing algoritma.

Tabel 2.2 Perbandingan dengan algoritma lain

Parameter	**Algortima AES**	**Algoritma Caesar Chiper**
Berdasarkan jumlah karakter *ciphertext*	Lebih panjang : Pada proses enkripsi dengan menggunakan kriptografi AES terjadi penambahan bit tiap kelipatan 32 bit, sehingga *output* akan selalu berjumlah kelipatan dari 32 bit.	Lebih singkat : Pada proses enkripsi satu huruf didalam sebuah pesan akan diganti dengan huruf yang berada tiga posisi dalam urutan alphabet huruf tersebut.
Keamanan (dilihat dari proses enkripsi pesan)	Lebih aman	Kurang aman
Output Generator	*Unpredictable* (tidak dapat diprediksi). Contoh : Karakter a = ;â?$Èè??aÖ0ÙÙô5	*Predictable* (dapat diprediksi). Contoh : karakter a = c
Kompleksitas Algoritma	Kurang Kompleks	Lebih Kompleks
Oppurtunities	Lebih Besar	Lebih Kecil

Berdasarkan Tabel 2.2 diatas mengenai perbandingan algoritma AES dengan algoritma Caesar Chiper bahwa penggunaan algoritma AES lebih efektif dibandingkan dengan Caesar Chiper. Hal tersebut dapat dilihat berdasarkan tingkat keamanan dari proses enkripsi pesan meskipun dalam penggunaan karakter *chipertext* algoritma Caesar chiper lebih unggul namun jika terjadi penyadapan algoritma ini mudah untuk dipecahkan. Selain itu berdasarkan *Output* Generator algoritma AES bersifat *unpredictable* yaitu tidak dapat diprediksi dalam bentuk karaker sehingga menyulitkan para *cryptanalyst* untuk menerjemahkan isi pesan. Dari parameter kompleksitasi algoritma Caesar chiper lebih unggul dibandingkan dengan algoritma AES karena proses enkripsi

algoritma Caesar chiper untuk setiap karakter hanya mengalami pergeseran 3 atau 5 kali sedangkan untuk algorima AES proses enkripsi terdiri dari 4 tahapan yaitu *SubBytes*, *ShiftRows*, *MixColumns*, *AddRoundKey*. Kemudian dari parameter *oppurtunities* algoritma AES lebih unggul karena algoritma ini termasuk algorima kriptografi modern sedangkan algoritma Caesar chiper termasuk kriptografi klasik.

2.3 *Short Message Service* (SMS)

Pengertian *Short Message Service* (SMS) adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, dalam hal ini perangkat nirkabel yang digunakan adalah telepone selular. Salah satu kelebihan dari SMS adalah biaya yang murah. Selain itu SMS merupakan metode *store* dan *forward* sehingga keuntungan yang di dapat 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. SMS menyediakan mekanisme untuk mengirimkan pesan singkat dari dan menuju media-media *wireless* dengan menggunakan sebuah *Short Message Service Center* (SMSC), yang bertindak sebagai sistem yang berfungsi menyimpan dan mengirimkan kembali pesan-pesan singkat.

2.4 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Kriptografi adalah suatu ilmu pengetahuan yang mempelajari teknik-teknik yang berkaitan dengan keamanan informasi, teknik-teknik yang digunakan pada umumnya menggunakan dasar pengetahuan matematika (Ariyus, 2008). Kriptografi bukanlah satu-satunya jalan dalam menjaga keamanan dokumen tetapi kriptografi menyediakan kumpulan teknik untuk menjaga keamanan dokumen. Secara garis besar kriptografi dibagi menjadi 2 jenis kriptografi klasik dan kriptografi moderen. Perbedaan mendasar yang terdapat pada ke dua jenis tersebut adalah pada kriptografi moderen,

algoritma kriptografi umumnya beroperasi pada mode bit sedangkan pada kriptografi klasik beroperasi pada mode karakter. Teknik kriptografi moderen, secara umum dibagi menjadi 2 jenis, yaitu:

1. Algoritma kriptografi kunci simetris

   Pada algoritma kriptografi ini, kunci yang digunakan dalam proses dekripsi dan enkripsi merupakan kunci yang sama. Berdasarkan pemrosesan bit, algoritma kunci simetris dibagi menjadi dua bagian, yaitu; algoritma *block cipher* yang melakukan pemrosesan bit per-blok dan algoritma *stream cipher* yang memproses blok secara mengalir atau per-bit.

2. Algoritma kriptografi kunci asimetris

   Proses enkripsi dan dekripsi pada algoritma kriptografi kunci publik menggunakan kunci yang berbeda. Seperti namanya algoritma ini menggunakan kunci enkripsi yang bersifat publik atau tidak rahasia, namun menggunakan kunci dekripsi yang bersifat rahasia. Kunci dekripsi pada umumnya merupakan hasil perhitungan dari kunci enkripsi yang bukan merupakan pemetaan satu ke satu, sebuah kunci dekripsi dapat memiliki beberapa kunci enkripsi. Dalam penggunaannya, algoritma kriptografi kunci publik tidak hanya digunakan untuk menyembunyikan pesan, tetapi dapat juga digunakan untuk melakukan otentikasi dokumen.

Tujuan kriptografi adalah untuk mencegah dan mendeteksi orang yang tidak bertanggung jawab melakukan hal-hal yang mengganggu seperti membaca data rahasia atau mengubah suatu data penting. Untuk tujuan itu, kriptografi menyediakan empat aspek keamanan yaitu kerahasiaan, integritas data, otentikasi dan penyangkalan. Algoritma kripografi melibatkan proses pengubahan pesan menjadi tersembunyi atau tidak dikenali isi dan maksudnya. Pesan yang belum diubah tersebut disebut dengan *plainteks* dan pesan yang telah diubah disebut dengan *ciphertext*. Proses pengubahan *plainteks* menjadi *ciphertext* disebut dengan enkripsi dan proses pengembalian *ciphertext* menjadi *plainteks* disebut dengan dekripsi.

2.5 Perbandingan SMS Konvensional dan SMS Enkripsi

SMS Konvensional atau SMS (*Short Message Service)* merupakan suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, dalam hal ini perangkat nirkabel yang digunakan adalah telepone selular. Sedangkan SMS Enkripsi merupakan pesan dalam bentuk *hexadesimal* atau kode bertujuan untuk menjaga tingkat keamanan pada saat proses pengirirman pesan. Pada Tabel 2.3 berikut akan dijelaskan mengenai perbandingan antara SMS Konvensional dan SMS Enkripsi.

Tabel 2.3 Perbandingan SMS Konvensional dan SMS Enkripsi

Perbandingan	SMS Konvensional	SMS Enkripsi	Keterangan
Efektivitas	Tidak Efektif	Efektif	SMS enkripsi lebih efektif dibandingkan dengan sms konvensional. Hal ini dilihat dari segi. keamanan pada saat proses pengiriman pesan dan isi dari pesan tersebut.
Efisiensi	Efisien	Efisien	Dilihat dari segi waktu pengiriman pesan. SMS konvensional dan SMS enkripsi sama-sama menggunakan jaringan pada saat pengiriman pesan.

Perbandingan	SMS Konvensional	SMS Enkripsi	Keterangan
Isi Pesan	Pesan Biasa	Pesan Rahasia	Dilihat dari proses pengiriman isi pesan. SMS konvensional biasanya digunakan untuk mengirimkan pesan biasa contoh dalam kegiatan sehari-hari. Sedangkan untuk SMS enkripsi pesan yang dikiriman adalah pesan yang bersifat penting dan rahasia sehingga isi didalam pesan pun hanya teks seperlunya.
Besar Data	Lebih Kecil	Lebih Besar	Dilihat dari ukuran pesan. SMS konvensional tidak mengalami perubahan karakter pada saat pengiriman, sedangkan SMS enkripsi pesan asli yang ingin dikirimkan harus di ubah menjadi bentuk *heksadesimal*.

Perbandingan	SMS Konvensional	SMS Enkripsi	Keterangan
Kecepatan Transfer Data	Sama	Sama	Kecepatan transfer data antara sms enkripsi dengan sms konvensional adalah sama karena keduanya menggunakan jaringan yang sama.

Berdasarkan uraian Tabel 2.3 mengenai perbedaan antar SMS Konvensional dan SMS Enkripsi maka dapat diambil kesimpulan bahwa dilihat dari efektifitas SMS Enkripsi lebih efektif dibandingkan dengan SMS Konvensional. Kemudian dari efisiensi SMS Konvensional dan SMS Enkripsi sama-sama memiliki tingkat efisiensi yang sama yaitu dilihat dari segi waktu pengiriman. Dilihat dari kecepatan transfer data SMS Enkripsi lebih cepat dibandingkan dengan SMS Konvensional karena dilihat dari segi kebutuhan dari isi pesan. Perbandingan terakhir SMS Konvensional memiliki ukuran lebih kecil dibandaingkan dengan SMS Enkripsi dari segi besar data.

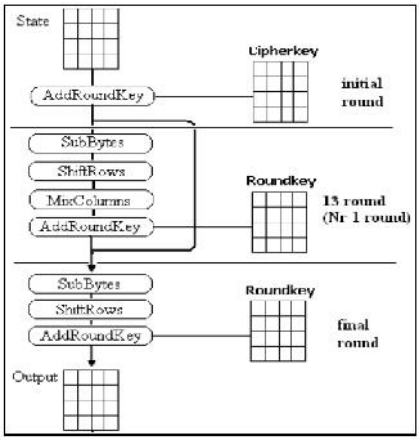
2.6 Algoritma *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) merupakan algoritma kriptografi yang dapat digunakan untuk mengamakan data. Menurut Ariyana (2011), Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*; sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Kriteria pemilihan AES didasarkan pada 3 kriteria utama yaitu keamanan, harga dan karakteristik algoritma beserta implementasinya. AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit. Berdasarkan ukuran block yang tetap, AES bekerja pada matriks berukuran 4 x 4 dimana tiap-tiap sel matriks terdiri atas 1 *byte* (8 bit). Pengelompokkan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka di belakang kata AES

menggambarkan panjang kunci yang digunakan pada tipa-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya *round* yang dipakai. AES-128 menggunakan 10 *round,* AES-192 sebanyak 12 *round,* dan AES-256 sebanyak 14 *round*. Proses algoritma *Advanced Encryption Standard (*AES) terdiri dari 2 proses yaitu proses Enkripsi dan proses Deskripsi.

1. Proses Enkripsi

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *byte*s, yaitu Sub*Byte*s, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, input yang telah di*copy*kan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, state akan mengalami transformasi Sub*Byte*s, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar 2.1 di bawah ini :



Gambar 2.1 Ilustrasi Proses Enkripsi AES

Dikutip dari : *Pengantar ilmu kriptografi teori analisis dan implementasi*, 2008, halaman 172

SubBytes

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (*S-Box*). Hasil yang didapat dari pemetaan dengan menggunakan tabel *S-Box* ini sebenarnya adalah hasil dari dua proses transformasi bytes, yaitu :

1. *Invers* perkalian dalam GF(2^8) adalah fungsi yang memetakan 8 bit ke 8 bit yang merupakan *invers* dari elemen *finite field* tersebut. Suatu byte a merupakan *invers* perkalian dari byte b bila a•b = 1, kecuali {00} dipetakan ke dirinya sendiri. Setiap elemen pada *state* akan dipetakan pada tabel *invers*. Sebagai contoh, elemen "01010011" atau {53} akan dipetakan ke {CA} atau "11001010".

2. Transformasi *affine* pada *state* yang telah dipetakan. Transformasi *affine* ini apabila dipetakan dalam bentuk matriks adalah sebagai berikut :

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

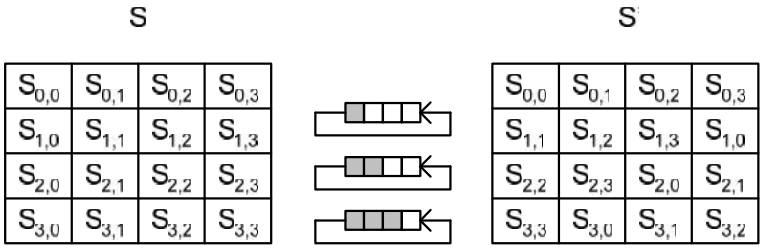
Gambar 2.2 Matriks Affine

$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ adalah urutan bit dalam elemen state atau array byte dimana b_7 adalah *most significant bit* atau bit dengan posisi paling kiri.

ShiftRows

Transformasi Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit).

Transformasi ini diterapkan pada baris 2, baris 3, dan baris 4. Baris 2 akan mengalami pergeseran bit sebanyak satu kali, sedangkan baris 3 dan baris 4 masing-masing mengalami pergeseran bit sebanyak dua kali dan tiga kali.



Gambar 2.3 Transformasi *ShiftRows*

MixColumns

Mixolumns mengoperasikan setiap elemen yang berada dalam satu kolom pada *state*. Elemen pada kolom dikalikan dengan suatu polinomial tetap a(x) = $\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. Secara lebih jelas, transformasi mixcolumns dapat dilihat pada perkalian matriks berikut ini :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 2.4 Matriks Transformasi *MixColumns*

Melakukan proses penambahan pada operasi ini berarti melakukan operasi *bitwise* XOR. Maka hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada di bawah ini :

$$s'_{0,c} = (\{02\} \bullet s_{o,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

Gambar 2.5 Hasil perkalian dari operasi matriks MixColumns

AddRoundKey

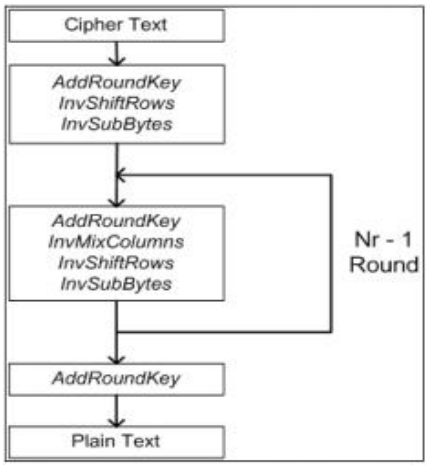
Pada proses AddRoundKey, sebuah *round key* ditambahkan pada *state* dengan operasi bitwise XOR. Setiap *round key* terdiri dari *Nb word* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :

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

$[w_i]$ adalah *word* dari key yang bersesuaian dimana $i = round*Nb+c$. Transformasi AddRoundKey diimplementasikan pertama kali pada *round* = 0, dimana *key* yang digunakan adalah *initial key* (*key* yang dimasukkan oleh kriptografer dan belum mengalami proses *key expansion*) (Wibowo A, 2004).

2. Proses Deskripsi

Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubByte*s, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada Gambar 2.6 berikut ini :

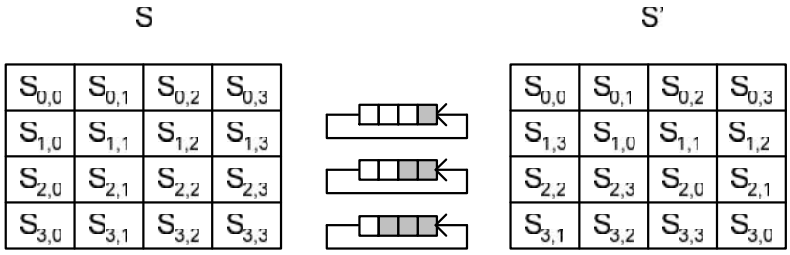


Gambar 2.6 Ilustrasi Proses Deskripsi AES

Dikutip dari : *Kriptografi Keamanan Data dan Komunikasi*, 2005, halaman 169

InvShiftRows

InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri. Pada baris kedua, pergeseran bit dilakukan sebanyak 3 kali, sedangkan pada baris ketiga dan baris keempat, dilakukan pergeseran bit sebanyak dua kali dan satu kali.



Gambar 2.7 Transformasi *InvShiftRows*

InvSubBytes

InvSubBytes juga merupakan transformasi bytes yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada *state* dipetakan dengan menggunakan tabel *inverse S-Box*. Tabel ini berbeda dengan tabel *S-Box* dimana hasil yang didapat dari tabel ini adalah hasil dari dua proses yang berbeda urutannya, yaitu transformasi *affine* terlebih dahulu, baru kemudian perkalian *invers* dalam GF(2^8).

$$\begin{bmatrix} b_7' \\ b_6' \\ b_5' \\ b_4' \\ b_3' \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Gambar 2.8 Matriks Invers Affine

Perkalian *invers* yang dilakukan pada transformasi InvSubBytes ini sama dengan perkalian *invers* yang dilakukan pada transformasi SubBytes.

InvMixColumns

Pada InvMixColumns, kolom-kolom pada tiap *state* (*word*) akan dipandang sebagai polinom atas GF(2^8) dan mengalikan modulo $x^4 + 1$ dengan polinom tetap $a^{-1}(x)$ yang diperoleh dari :

$$a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}.$$

Atau dalam matriks :

$$s'(x) = a(x) \otimes s(x)$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 2.9 Matriks InvMixColumns

Hasil dari perkalian diatas adalah :

$$s'_{0,c} = (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

Gambar 2.10 Hasil Perkalian Matriks

Inverse AddRoundKey

Transformasi Inverse AddRoundKey tidak mempunyai perbedaan dengan transformasi AddRoundKey karena pada transformasi ini hanya dilakukan operasi penambahan sederhana dengan menggunakan operasi bitwise XOR.

Ekspansi Kunci

Algoritma AES mengambil kunci cipher, K, dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Algoritma ini membutuhkan set awal *key* yang terdiri dari *Nb word*, dan setiap *round Nr* membutuhkan data kunci sebanyak *Nb word*. Hasil *key schedule* terdiri dari array 4 byte word linear yang dinotasikan dengan [w_i].

SubWord adalah fungsi yang mengambil 4 byte *word* input dan mengaplikasikan S-Box ke tiap-tiap data 4 byte untuk menghasilkan *word* output. Fungsi *RotWord* mengambil *word* [a_0, a_1, a_2, a_3] sebagai input, melakukan permutasi siklik, dan mengembalikan *word* [a_1, a_2, a_3, a_0]. Rcon[i] terdiri dari nilai-nilai yang diberikan oleh [x^{i-1}, {00}, {00}, {00}], dengan x^{i-1} sebagai pangkat dari x (x dinotasikan sebagai {02} dalam *field GF*(2^8). *Pseudocode* dari proses ekspansi *key* dapat dilihat seperti yang ada di bawah berikut ini :

```
KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
    word temp
    i = 0
    while (i < Nk)
            w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
            i = i +1
    end while
    i = Nk
    while (i < Nb*(Nr+1))
            temp = w[i -1]
            if (i mod Nk = 0)
                    temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
            else if (Nk > 6 and i mod Nk = 4)
                    temp = SubWord(temp)
```

```
        end if
        w[i]=w[i-Nk] xor temp
        i = i + 1
    end while
end
```

Dari *pseudocode* dapat dilihat bahwa *word* ke *Nk* pertama pada ekspansi kunci berisi kunci *cipher*. Setiap word berikutnya, w[i], sama dengan XOR dari *word* sebelumnya, w[i-1] dan *word Nk* yang ada pada posisi sebelumnya, w[i-Nk]. Untuk *word* pada posisi yang merupakan kelipatan *Nk*, sebuah transformasi diaplikasikan pada w[i-1] sebelum XOR, lalu dilanjutkan oleh XOR dengan konstanta *round*, Rcon[i]. Transformasi ini terdiri dari pergeseran siklik dari byte data dalam suatu word *RotWord*, lalu diikuti aplikasi dari *lookup tabel* untuk semua 4 byte data dari word *SubWord*. (Wibowo A, 2004).

2.7 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Pada saat perilisan perdana android 5 November 2007 bersamaan Open Handset Alliance menyatakan mendukung open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode android di bawah lisensi Apache sebuah lisensi perangkat lunak dan open platform perangkat seluler. Android dipuji sebagai "platform mobile pertama yang lengkap, terbuka dan bebas".

- Lengkap (Complete Platform) : Para desainer dapat melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan platform andorid. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang mengembangkan aplikasi.
- Terbuka (Open Source Platform) : Platform android disediakan melalui lisensi open source. Pengembang dapat dengan bebas mengembangkan aplikasi.

- Free (Free Platform) : Android adalah platform/aplikasi yang bebas untuk develop. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada platform android tidak ada biaya keanggotaan diperlukan tidak diperlukan biaya pengujian tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apapun. (Safaat.N, 2012).

Pembuatan Aplikasi Enkripsi dan Deskripsi Algoritma AES untuk Implementasi SMS dibuat menggunakan sistem operasi Android. Sistem operasi ini dipilih karena dilihat dari beberapa kelebihan yang dimiliki dibandingkan dengan sistem operasi lain. sistem operasi Android memiliki *platform* terbuka sehingga mudah untuk melakukan pengembangan dalam pembuatan aplikasi. Selain itu, sistem operasi android ini juga bersifat *free platform* sehingga bebas untuk di develop karena tidak ada lisensi atau biaya royalti jika ingin menggunakan sistem operasi tersebut.

2.8 Android SDK (*Software Development Kit*)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android mengunakan bahasa java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang di-release oleh Google (Safaat.N, 2012). Beberapa fitur-fitur Android yang paling penting adalah :

- Framework aplikasi yang mendukung penggantian komponen dan reusable.
- Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi open ES 1,0 (Opsional akselerasi hardware).
- SQLite untuk penyimpanan data.
- Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

2.9 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

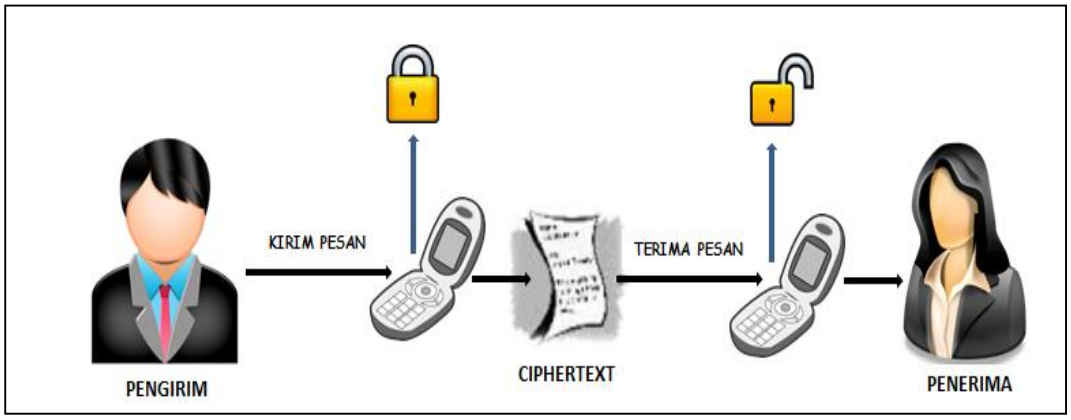
- Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- Mulit-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
- Multi-role: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

BAB III
ANALISIS dan PERANCANGAN

Pada bab ini akan menjelaskan mengenai analisis dan perancangan aplikasi Enkripsi dan Deskripsi Menggunakan Algoritma AES (*Advanced Encryption Standard*) untuk Implementasi SMS (*Short Message Service*) yang dikembangkan pada platform Android dan IDE yang digunakan untuk membangun aplikasi.

3.1 Deskripsi Umum Sistem



Gambar 3.1 Arsitektur Global Sistem

Berdasarkan Gambar 3.1 diatas pengirim membuat pesan atau *plaintext* kepada penerima. Pesan yang akan dikirim di enkripsi terlebih dahulu dengan menggunakan *key* atau kunci. Pesan yang telah terenkripsi akan menghasilkan *ciphertext*. Kemudian untuk mendeskripsikan pesan yang telah terenkripsi, harus menggunakan *key* atau kunci yang telah dibuat oleh pengirim. Pesan yang telah terdeskripsi akan kembali menghasilkan *plaintext* sehingga dapat dibaca oleh penerima.

3.2 Analisis Kebutuhan Sistem

Aplikasi utama pada sms enkripsi dan deskripsi merupakan aplikasi yang dapat digunakan oleh *user* melalui *platform* android. *User* yang menggunakan aplikasi utama ini dapat dibedakan menjadi 2 jenis, yaitu pengirim dan penerima pesan. Pengirim melakukan pengiriman pesan yang terenkripsi sedangkan penerima

adalah user yang dapat mendeskripsikan pesan yang terenkripsi oleh pengirim pada aplikasi Android.

3.2.1 Analisis Kebutuhan *Software*

- *Software* yang digunakan dalam pembuatan aplikasi ini adalah :
    1. Sistem Operasi yang digunakan adalah Android
    2. *Integrated Development Environment* (IDE) yang digunakan adalah eclipse karena IDE ini mempunyai ADT (*Android Development Tool*)
    3. *Android Sistem Development Kit* (Android SDK) yang menyediakan development environment dengan semua komponen pengembangan yang diperlukan.
    4. ADT android membuat kostum *plugin* untuk IDE eclipse yang memberikan kemudahan untuk pengembangan sebuah sistem aplikasi berbasis Android.
- *Software* untuk Penerapan

Software yang digunakan dalam penerapan aplikasi adalah Android Froyo 2.2.

3.2.2 Analisis Kebutuhan *Hardware*

Untuk merancang aplikasi dibutuhkan *hardware* yang medukung aplikasi tersebut. Adapun *hardware* yang dibutuhkan dibagi menjadi dua yaitu :

- *Hardware* untuk proses pembuatan dapat dilihat pada Tabel 3.1.

Tabel 3.1 *Hardware* Komputer

Processor	AMD E2-1800 APU
Memory	2048 MB
Hard Drive	465,76 GB SATA
Video Card	AMD Radeon HD 7340
Display	10.1 HD LED LCD
Audio	*Realtek High Definition Audio*

- *Hardware mobile* untuk proses pembuatan dapat dilihat pada Tabel 3.2.

Tabel 3.2 *Hardware mobile*

RAM	512 MB
Memory	4 GB
Konektivitas	3G, WLAN Wi-Fi, Bluetooth, USB
CPU	Dual-core 1.2 GHz

3.3 Kebutuhan Fungsional

Kebutuhan fungsional Aplikasi Enkripsi dan Deskripsi Menggunakan Algoritma AES untuk Implementasi SMS dapat dilihat pada Tabel 3.3 berikut.

Tabel 3.3 Kebutuhan Fungsional

F-001	Aplikasi dapat menerima *inputan* nomor tujuan, pesan dan kunci.
F-002	Aplikasi dapat melakukan enkripsi pesan menggunakan kunci yang telah di *input*.
F-003	Aplikasi dapat mengirim ke nomor tujuan.
F-004	Aplikasi dapat menampilkan nomor pengirim pesan.
F-005	Aplikasi dapat melakukan deskripsi menggunakan button deskripsi pada aplikasi.
F-006	Aplikasi dapat menampilkan pesan asli setelah di deskripsi kepada penerima.
F-007	Aplikasi memiliki menu bantuan untuk memudahkan *user* dalam penggunaan.

3.4 Kebutuhan Non-Fungsional

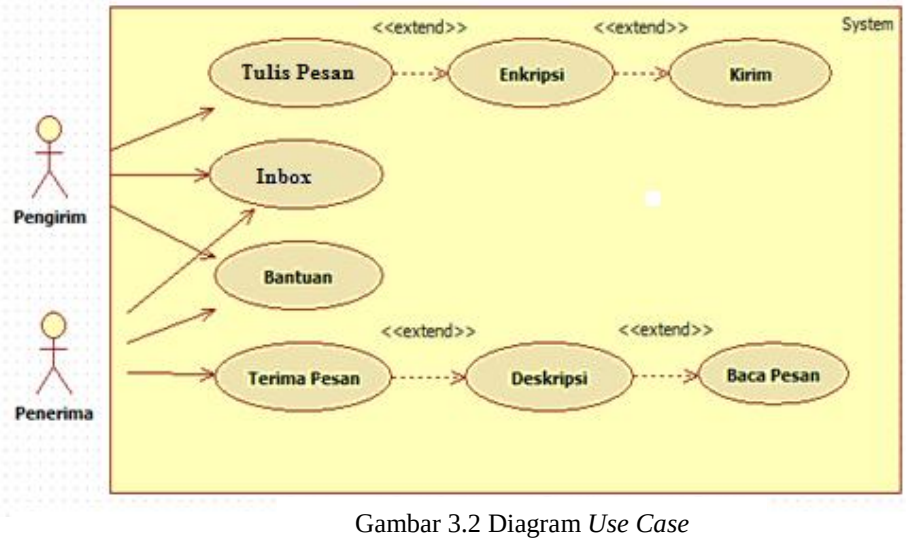
Kebutuhan Non-Fungsional Aplikasi Enkripsi dan Deskripsi Menggunakan Algoritma AES untuk Implementasi SMS dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4 Kebutuhan Non Fungsional

NF-001	Tampilan di *design* menarik sehingga pengguna tertarik untuk menggunakan aplikasi.
NF-002	Cara kerja sistem mudah dimengerti oleh pengguna.

3.5 Diagram *Use Case*

Use Case diagram digunakan untuk mengambarkan hubungan sejumlah *external* aktor dengan *Use Case* yang terdapat dalam sistem aplikasi. *Use Case* diagram ini hanya menggambarkan keadaan lingkungan sistem yang dapat dilihat dari luar aktor. *Use Case* diagram aplikasi Enkripsi dan Deskripsi Menggunakan Algoritma AES untuk Implementasi SMS dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram *Use Case*

3.5.1 Skenario *Use Case* Tulis Pesan

Aktor : Pengirim.

Kondisi Awal : Pengirim memilih menu Tulis Pesan.

Skenario : Pengirim memilih menu tulis pesan kemudian menulis pesan pada aplikasi untuk dikirim.

Kondisi Akhir : Pengirim selesai menulis pesan.

3.5.2 Skenario *Use Case* Enkripsi

Aktor : Pengirim.

Kondisi Awal : Pengirim membuat pesan

Skenario :Pengirim membuat pesan kemudian memasukkan kunci enkripsi agar pesan tidak dapat dibaca.

Kondisi Akhir : Pengirim sudah mengenkripsi pesan.

3.5.3 Skenario *Use Case* Kirim

Aktor : Pengirim.

Kondisi Awal : Pengirim sudah membuat dan mengenkripsi pesan.

Skenario : Pengirim membuat dan mengenkripsikan pesan terlebih dahulu, kemudian setelah selesai pesan akan dikirim.

Kondisi Akhir : Pesan dikirim dan akan menampilkan status pengiriman.

3.5.4 Skenario *Use Case* Terima Pesan

Aktor : Penerima.

Kondisi Awal : Pesan telah dikirim oleh pengirim.

Skenario : Penerima mendapatkan pesan dari pengirim kemudian pesan akan dideskripsi sehingga dapat dibaca.

Kondisi Akhir : Pesan sudah diterima.

3.5.5 Skenario *Use Case* Deskripsi Pesan

Aktor : Penerima.

Kondisi Awal : Penerima mendapat pesan dari pengirim.

Skenario :Penerima mendapatkan pesan terlebih dahulu dari pengirim, setelah pesan diterima masukkan kode untuk mendeskripsikan pesan sehingga pesan dapat dibaca.

Kondisi Akhir : Pesan telah di deskripsi.

3.5.6 Skenario *Use Case* Baca Pesan

Aktor : Penerima.

Kondisi Awal : Penerima telah mendapat pesan

Skenario :Penerima mendapatkankan pesan kemudian mendeskripsikan pesan.

Kondisi Akhir : Sudah membaca pesan.

3.5.7 Skenario *Use Case Inbox*

Aktor : Pengirim atau Penerima.

Kondisi Awal : Pesan diterima.

Skenario : Pengirim atau penerima dapat melihat pesan masuk .

Kondisi Akhir : Pesan dapat dilihat.

3.5.8 Skenario *Use Case* Bantuan

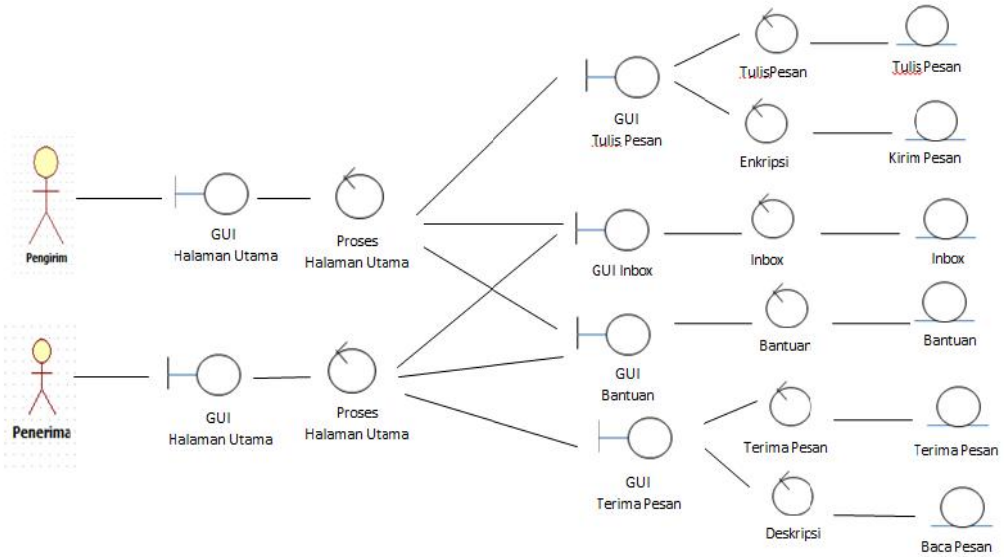
Aktor : Pengirim atau Penerima.

Kondisi Awal : Pengirim atau Penerima telah memilih menu bantuan

Skenario :Pengirim atau penerima memilih bantuan pada menu utama yang terdapat didalam aplikasi .

Kondisi Akhir : Bantuan dapat dilihat.

3.6 Analisis Kelas



Gambar 3.3 Analisis Kelas Enkripsi dan Deskripsi Menggunakan Algoritma AES untuk Implementasi SMS

Berdasarkan gambar diatas terlihat bahwa diagram dibagi menjadi tiga kelas, yatu Kelas *Boundary*, Kelas *Control* dan Kelas *Entity*. Rincian kelas-kelas tersebut akan dijelaskan dalam Tabel 3.5.

Tabel 3.5 Rincian Kelas Aplikasi

Jenis Kelas	Nama Kelas	Deskripsi	Perancangan *Use Case*
Kelas *Boundary*	GUI TulisPesan	Kelas yang berperan sebagai antarmuka untuk melakukan pembuatan pesan.	*Use Case* menangani TulisPesan.

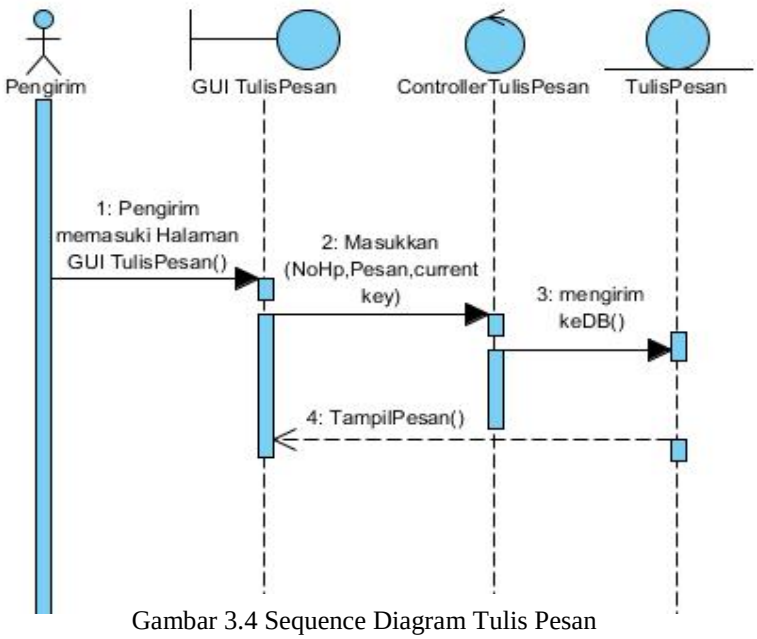
Jenis Kelas	Nama Kelas	Deskripsi	Perancangan *Use Case*
Kelas *Boundary*	GUI *Inbox*	Kelas yang berperan sebagai antarmuka untuk menampilkan pesan masuk.	*Use Case* Menampilkan pesan masuk.
	GUI Bantuan	Kelas yang berperan sebagai antarmuka untuk menampilkan bantuan.	*Use Case* menampilkan bantuan.
	GUI Terima Pesan	Kelas yang berperan sebagai antarmuka untuk menampilkan Terima Pesan.	*Use Case* menampilkan Terima Pesan.
Kelas *Control*	Tulis Pesan	Kontroler yang berfungsi untuk menampilkan buat pesan.	*Use Case* melakukan buat pesan.
	Enkripsi	Kontroler yang berfungsi untuk menampilkan pesan enkripsi.	*Use Case* melakukan enkripsi
	Inbox	Kontroler yang berfungsi untuk menampilkan pesan masuk.	*Use Case* menampilkan pesan masuk.
	Bantuan	Kontroler yang berfungsi untuk menampilkan bantuan	*Use Case* menampilkan bantuan.
	Deskripsi	Kontroler yang berfungsi untuk menampilkan pesan deskripsi.	*Use Case* Deskripsi.

Jenis Kelas	Nama Kelas	Deskripsi	Perancangan *Use Case*
Kelas *Control*	Baca Pesan	Kontroler yang berfungsi menmpilkan hasil pesan.	*Use Case* Baca Pesan.
Kelas *Entity*	Tulis Pesan	Kelas yang berfungsi untuk melakukan penyimpanan pesan.	*Use Case* Tulis pesan.
	Enkripsi Pesan	Kelas yang berfungsi untuk melakukan penyimpanan pesan enkripsi.	*Use Case* Enkripsi Pesan.
	Inbox	Kelas yang berfungsi untuk melakukan penyimpanan pesan masuk.	*Use Case* Kotak Masuk.
	Bantuan	Kelas yang berfungsi untuk menyimpan bantuan.	*Use Case* bantuan.
	TerimaPesan	Kelas yang berfungsi untuk menyimpan pesan masuk	*Use Case* Terima Pesan.
	Baca Pesan	Kelas yang berfungsi untuk menyimpan pesan yang telah di deskripsi.	Use Case Baca Pesan.

3.7 Sequence Diagram

Sequence Diagram menggambarkan urutan proses yang akan terjadi dalam sistem. Diagram ini juga menggambarkan *method* yang dijalankan oleh masing-masing kelas setiap proses yang terjadi pada sistem.

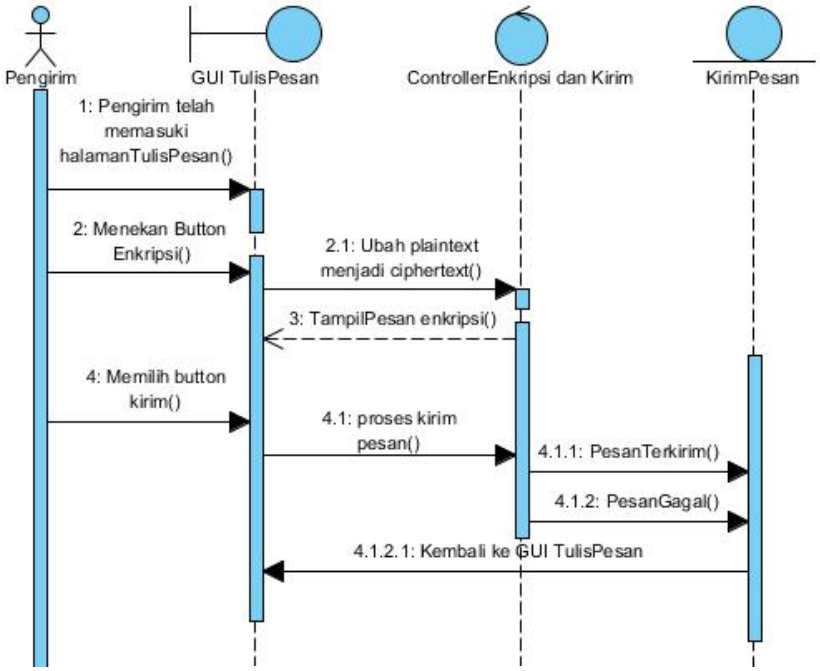
3.7.1 Sequence Diagram Tulis Pesan



Gambar 3.4 Sequence Diagram Tulis Pesan

Pada Gambar 3.4 diatas menjelaskan urutan proses pembuatan pesan pada aplikasi. Pengirim awalnya memilih menu pesan pada GUI Halaman Utama kemudian GUI tersebut diproses menuju ke GUI tulis pesan. Setelah itu, pengirim memasukkan no tujuan, isi pesan dan *current key* lalu kemudian di proses kembali oleh controller tulis pesan untuk dikirimkan ke *database* tulis pesan. Setelah dikirim maka no tujuan, pesan dan *current key* akan ditampilkan kembali ke GUI tulis pesan. Kemudian hasil yang telah ditampilkan akan dikirim. Jika kirim pesan berhasil maka pesan akan disimpan dan jika pengiriman pesan gagal maka pesan tidak disimpan dan akan kembali ke GUI Tulis Pesan.

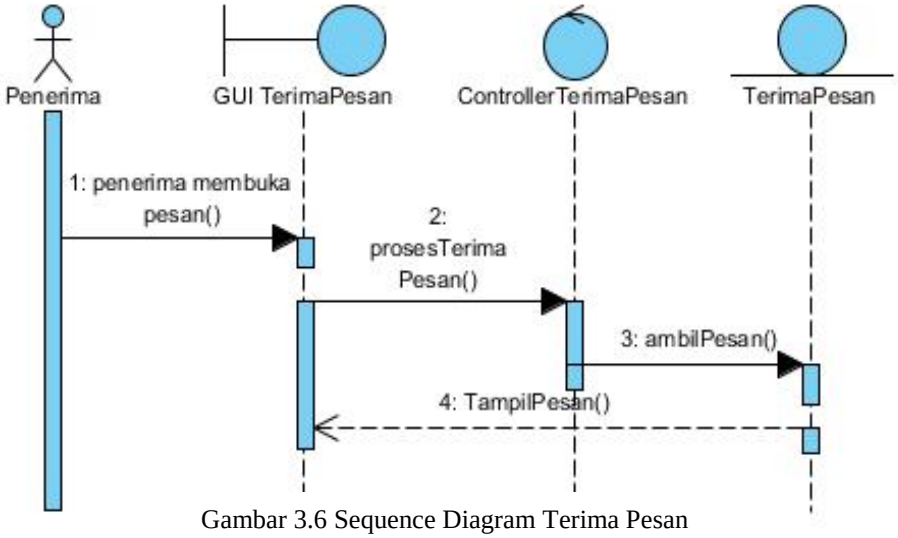
3.7.2 Sequence Diagram Enkripsi & Kirim Pesan



Gambar 3.5 Sequence Diagram Enkripsi & Kirim Pesan

Pada Gambar 3.5 menggambarkan urutan proses enkripsi dan kirim pesan. Pengirim telah memasuki GUI tulis pesan, kemudian pengirim memasukkan *current key*. Setelah *current key* diisi maka pengirim menekan *button* enkripsi yang kemudian akan diproses pada controller enkripsi untuk mengubah pesan asli menjadi *ciphertext*. Setelah itu ciphertext akan disimpan ke db enkripsi pesan dan ditampilkan ke GUI tulis pesan. Setelah enkripsi pesan selesai maka pengirim akan memilih *button* kirim dan kemudian *button* tersebut akan diproses oleh *controller* kirim. Selanjutnya pesan akan terkirim akan tersimpan pada db kirim pesan sedangkan pesan yang gagal tidak disimpan kedalam *database* dan akan kembali ke GUI tulis pesan.

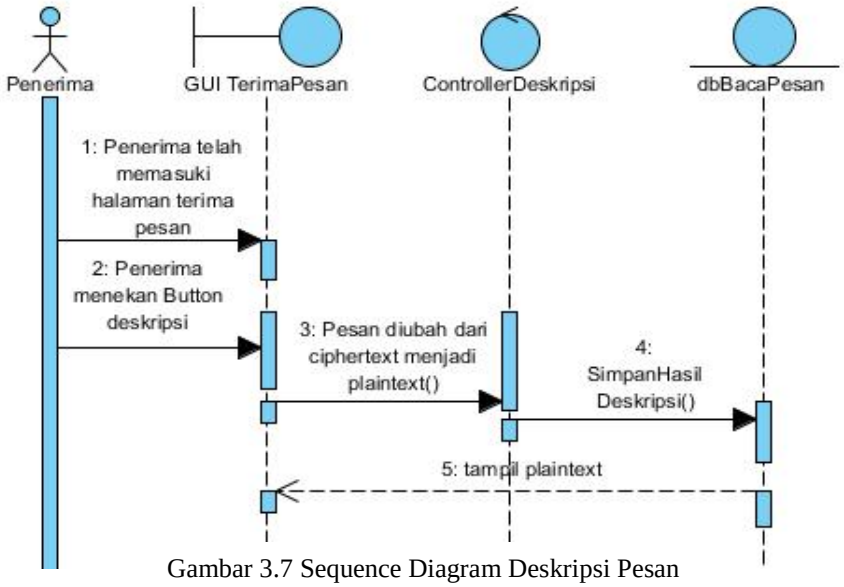
3.7.3 Sequence Diagram Terima Pesan



Gambar 3.6 Sequence Diagram Terima Pesan

Pada Gambar 3.6 menggambarkan urutan proses terima pesan. Kondisi awal penerima melihat pesan yang telah diterima pada GUI halaman utama kemudian akan diproses untuk menuju ke halaman GUI terima pesan. Setelah itu pesan akan diambil dari db terima pesan dan kemudian akan ditampilkan pada GUI terima pesan. Pada GUI terima pesan terdapat no pengirim, pesan yang telah terenkripsi serta *current key* dari penerima.

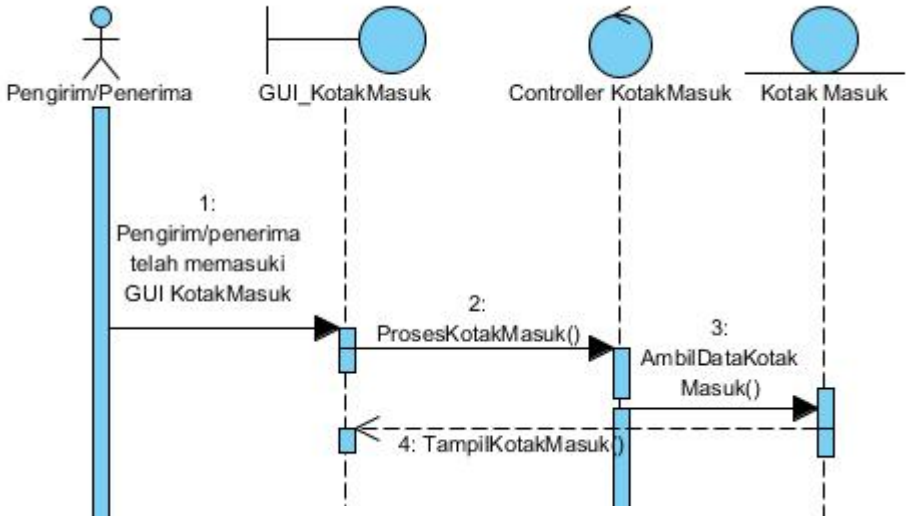
3.7.4 Sequence Diagram Deskripsi Pesan



Gambar 3.7 Sequence Diagram Deskripsi Pesan

Pada Gambar 3.7 menggambarkan urutan proses deskripsi pesan. Kondisi awal penerima telah memasuki GUI terima pesan. Pada GUI terima pesan tersebut terdapat no pengirim, isi pesan yang telah terenkripsi dan *current key* dari pengirim. Setelah itu penerima menekan *button* deskripsi yang akan diproses oleh deskripsi untuk mengubah *ciphertext* menjadi *plaintext.* Hasil dari deskripsi akan disimpan ke db baca pesan dan ditampilkan ke GUI terima pesan.

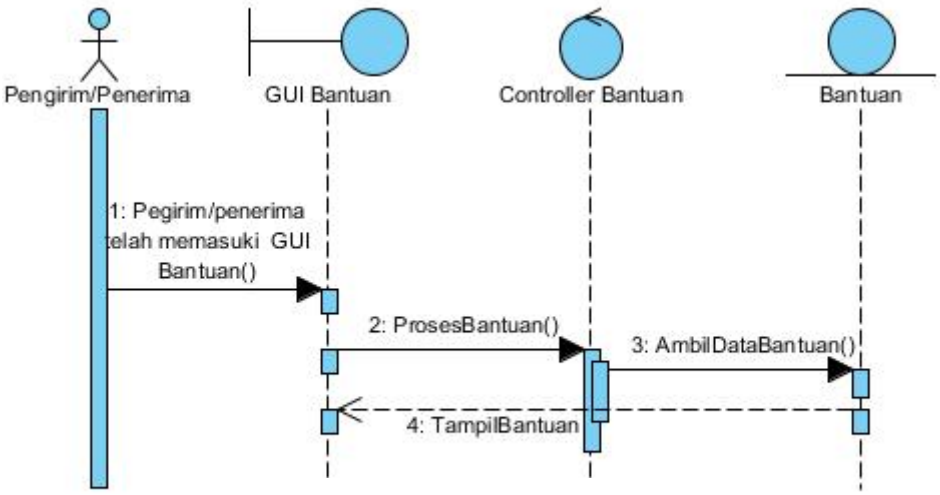
3.7.5 Sequence Diagram Inbox



Gambar 3.8 Sequence Diagram *Inbox*

Pada Gambar 3.8 menggambarkan urutan proses *inbox*. Kondisi awal pengirim atau penerima memasuki GUI halaman utama untuk memilih menu *inbox* kemudian diproses. Setelah itu pengirim atau penerima memasuki GUI kotak masuk kemudian diproses oleh *controller* kotak masuk setelah itu data *inbox* akan diambil dari *database* dan kemudian akan ditampilkan ke GUI kotak masuk.

3.7.6 Sequence Diagram Bantuan

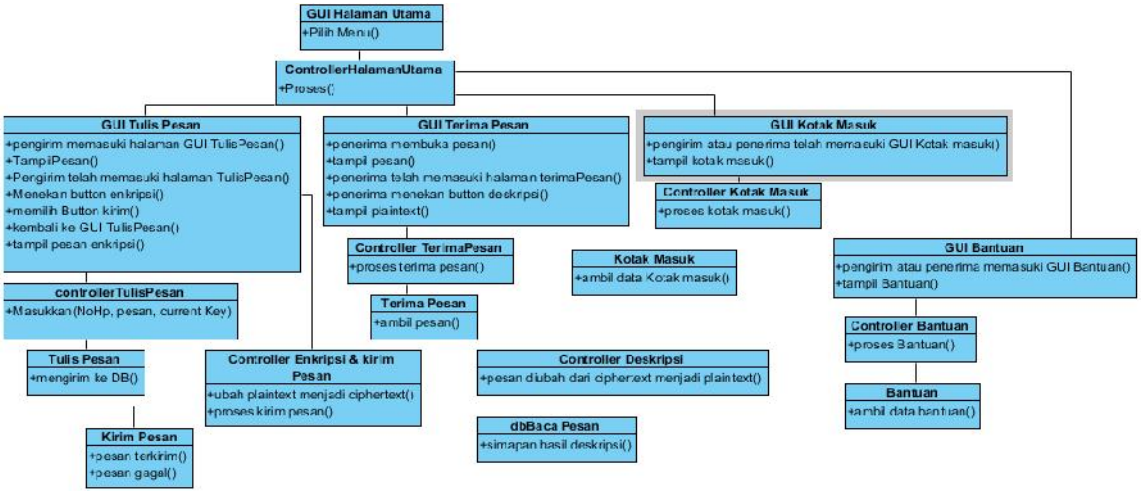


Gambar 3.9 Sequence Diagram Bantuan

Pada Gambar 3.9 menggambarkan urutan proses menampilkan Bantuan. Kondisi awal pengirim atau penerima memasuki halaman GUI bantuan. Kemudian setelah memilih akan diproses oleh c*ontroller* bantuan untuk mengambil data dari *database* bantuan setelah itu akan ditampilkan kembali ke GUI bantuan.

3.8 *Class* Diagram

Pemodelan pada Gambar 3.10 c*lass* diagram untuk menggambarkan perancangan struktur class-class yang menyusun aplikasi Enkripsi dan Deskripsi menggunakan algortima AES untuk implementasi SMS adalah sebagai berikut.



Gambar 3.10 *Class* Diagram

3.9 Algoritma

3.9.1 Algoritma Kelas Tulis Pesan

Nama Kelas : tulis_pesan

Nama Operasi : kirimpesan : send()

Algoritma : *(Algo-001)*

```
{User mengirim pesan}
Initial state : Belum mengirim pesan
Final state   : Sudah mengirim pesan

Algoritma
if(no.length()>0 && pesan_enkripsi.length()>0 &&Skunci.length()>0) {
                    sendSMS(no,pesan_enkripsi);

                }
                else if (no.length()>0 &&kunci.length()>0 && hasil.length()==0) {
                    Toast.makeText(getBaseContext(),"Pesan Kosong / Belum
Terenkripsi", Toast.LENGTH_SHORT).show();
                }
                else {
                    Toast.makeText(getBaseContext(),"Nomor Tujuan Belum
Terisi", Toast.LENGTH_SHORT).show();
                }

                }
```

3.9.2 Algoritma Kelas Tulis Pesan

Nama Kelas : tulis_pesan

Nama Operasi : enkripButton : onClick(View v)

Algoritma : *(Algo-002)*

```
{User mengenkripsi pesan}
Initial state : Pesan belum dienkripsi
Final state  : Pesan sudah dienkripsi

Algoritma
if(noPK.length()>0 && noPesan.length()>0)
            {
            teksEnkrip.setText(enKata);
            }
```

```
                else
                {
                    Toast.makeText(getBaseContext(),"current key dan Pesan tidak boleh
kosong", Toast.LENGTH_LONG).show();
                }
                }
                    });
```

3.9.3 Algoritma Kelas Baca Pesan

Nama Kelas : baca_pesan

Nama Operasi : dekripButton : onClick(View v)

Algoritma : *(Algo-003)*

```
{User mendeskripsi pesan}

Initial state : Pesan belum dideskripsi
Final state  : Pesan sudah dideskripsi
Algoritma
if(!ambilKata.equals(null)) {
                    try {
                    deKata = GenerateAES.decrypt(pecah1, pecah0);
                    } catch(Exception e) {

                    }
                    teksDekrip.setText(deKata);
                }
                }
                    });
```

3.10 Implementasi AES Pada Aplikasi

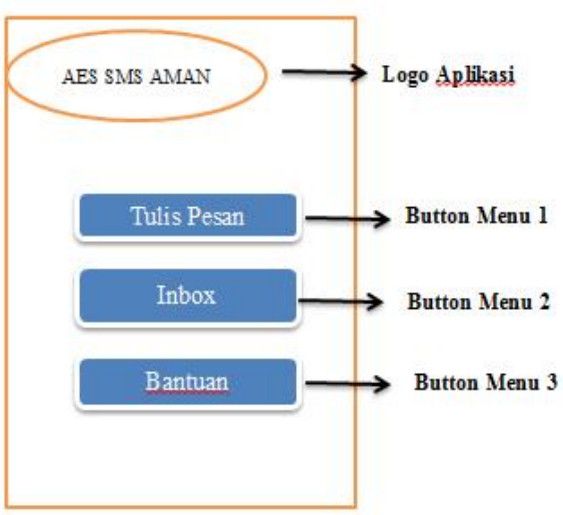
Pembuatan aplikasi keamanan SMS dengan menggunakan algoritma AES ini adalah dengan mengembangkan *project* Proyek Akhir yang dibuat oleh M. Acep Syaifullah dan Yogi Mardion mahasiswa Politeknik Negeri Batam dengan judul Proyek Akhir adalah "Mobile Application SMS Enkripsi" dengan metode algoritma yang digunakan adalah RC6. Untuk pembuatan aplikasi Tugas Akhir ini menggunakan *library* yang tersedia dari *project* tersebut kemudian dilakukan penambahan yang sesuai dengan kebutuhan aplikasi pada Tugas Akhir. Pada penelitian sebelumnya distribusi kunci dilakukan secara manual, jadi anatar

pengirim dan penerima harus membuat kesepakatan terlebih dahulu untuk kunci yang digunakan. Pada Tugas Akhir ini distribusi kunci langsung ditangani oleh sistem jadi kunci yang di*input* langsung dikirimkan bersama dengan pesan rahasia. Kunci yang dikirimkan tersebut adalah parameter agar dapat melakukan proses deskripsi pesan. sehingga penerima hanya tinggal melakukan deskripsi dari pesan yang dikirimkan. Jika tidak terdapat kunci maka pesan tidak dapat dibaca.

3.11 Perancangan AntarMuka

Antarmuka atau *interface* dari aplikasi akan dideskripsikan pada bagian ini. Antarmuka dikategorikan untuk 2 (dua) aktor, yaitu Pengirim dan Penerima. Berikut antarmuka Aplikasi Enkripsi dan Deskripsi Menggunakan Algortima AES untuk Implementasi SMS.

3.11.1 Rancangan Antar Muka Halaman Menu Utama



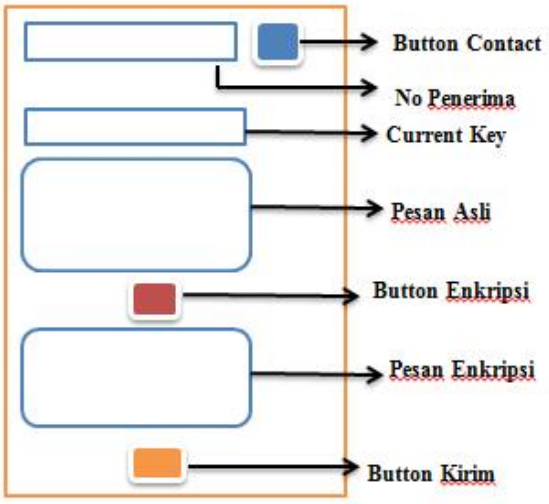
Gambar 3.11 Tampilan Halaman Utama Aplikasi

Berdasakan tampilan rancangan aplikasi di atas, deskripsi tampilan antarmuka halaman menu utama dapat dilihat pada Tabel 3.6 berikut.

Tabel 3.6 Deskripsi Tampilan Antarmuka Halaman Utama

Id_Objek	Jenis	Nama	Keterangan
Jtulis_p esan	jButton	Tulis Pesan	Menampilkan Halama Buat Pesan.
JInbox	jButton	Inbox	Menampilkan Halaman Pesan Masuk.
Jbantuan	jButton	Bantuan	Menampilkan Halaman Bantuan.

3.11.2 Rancangan Antar Muka Tulis Pesan



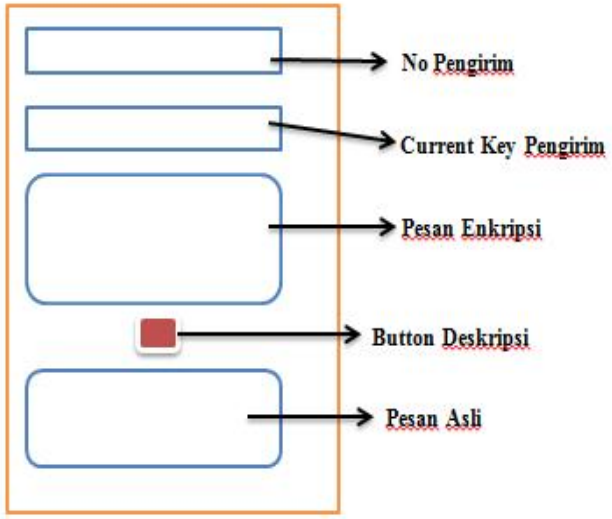
Gambar 3.12 Tampilan AntraMuka Tulis Pesan

Berdasakan tampilan rancangan aplikasi di atas, deskripsi tampilan antarmuka buat pesan dapat dilihat pada Tabel 3.7 berikut.

Tabel 3.7 Deskripsi Tampilan Antarmuka Tulis Pesan

Id_Objek	Jenis	Nama	Keterangan
jKontak	jButton	Kontak	Menampilkan Halaman Kontak
jNoTujuan	jText	No Tujuan	Masukkan nomor tujuan
jPesan	jText	Pesan	Masukkan isi pesan.
jKode	jText	Kode	Masukkan kode untuk enkripsi pesan.
jPesanEnkripsi	jText	Pesan Enkripsi	Menampilkan hasil pesan enkripsi.
jKirim	jButton	Kirim	Mengirim pesan.

3.11.3 Rancangan Antar Muka Terima Pesan



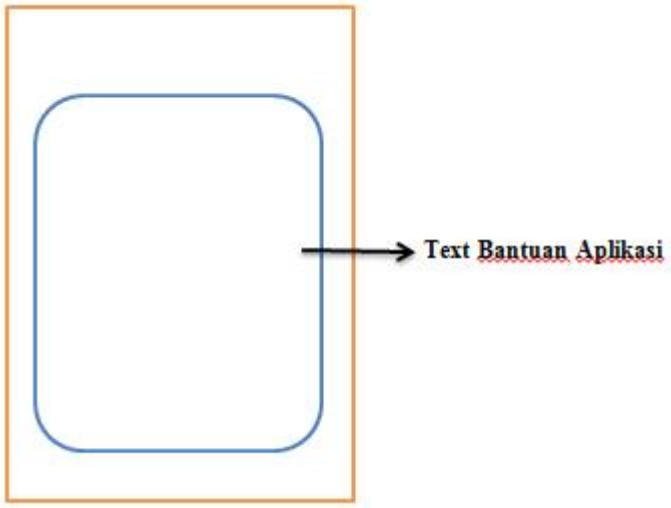
Gambar 3.13 Tampilan AntarMuka Terima Pesan

Berdasakan tampilan rancangan aplikasi di atas, deskripsi tampilan antarmuka Terima Pesan dapat dilihat pada Tabel 3.8 berikut.

Tabel 3.8 Deskripsi Tampilan Antarmuka Terima Pesan

Id_Objek	Jenis	Nama	Keterangan
jNoPengirim	Jtext	No Pengirim	Menampilkan No pengirim
jPesan	Jtext	Pesan	Menampilkan pesan.
jKode	Jtext	Kode	Masukkan kode.
jPesanAsli	Jtext	Pesan Asli	Menampilkan hasil deskripsi pesan.

3.11.4 Rancangan Antar Muka Bantuan



Gambar 3.14 Tampilan AntarMuka Bantuan

Berdasakan tampilan rancangan aplikasi di atas, deskripsi tampilan antarmuka pesan masuk dapat dilihat pada Tabel 3.9 berikut.

Tabel 3.9 Deskripsi Tampilan Antarmuka Bantuan

Id_Objek	Jenis	Nama	Keterangan
jBantuan	jLabel	Bantuan	Menampilkan petunjuk penggunaan aplikasi.

BAB IV
IMPLEMENTASI dan PENGUJIAN

4.1 Implementasi Kelas

Berdasarkan perancanaan yang telah dilakukan, maka hasil implementasi kelas dan antarmuka ynag dibuat secara detail dapat dilihat pada Tabel 4.1.

Tabel 4.1 Implementasi kelas

No	Nama Kelas	Nama File Fisik	Nama File Executable
1	tulis_pesan	tulis_pesan.java	Tulis_pesan.class
2	Bantuan	bantuan.java	bantuan.class
3	MainActivity	MainActivity.java	MainActivity.class
4	Inbox	inbox.java	inbox.class
5	GenerateAES	GenerateAES.java	GenerateAES.class
6	baca_pesan	baca_pesan.java	baca_pesan.class

Dari perancangan yang telah dilakukan, saat melakukan implementasi menghasilkan 6 kelas yaitu kelas tulis_pesan, bantuan, MainActivity, inbox, GenerateAES dan baca_sms. Kelas-kelas tersebut mewakili fungsional dari aplikasi.

4.2 Implementasi Antarmuka

Berdasarkan dokumen perancangan yang telah dilakukan, maka hasil implementasi dari antarmuka yang dibuat secara detail dapat dilihat pada table 4.2.

Tabel 4.2 Implementasi Antarmuka

No	Antarmuka	Nama File Fisik	Nama File Executable
1	activity_main	activity_main.xml	activity_main.xml
2	baca_sms	baca_sms.xml	baca_sms.xml
3	Inbox	inbox.xml	inbox.xml
4	Tulis Pesan	Tulis_pesan.xml	Tulis_pesan.xml
5	Bantuan	Bantuan.xml	Bantuan.xml

Pada tahap desain dan tahap implementasi tetap teradapat lima antarmuka yaitu activity_main, Main, Baca_sms, Inbox dan Bantuan.

4.2.1 Implementasi Menu Utama

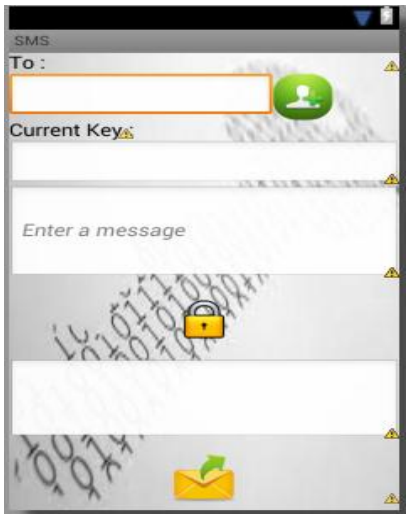
Pada gambar 4.1 merupakan antarmuka Menu Utama.

Gambar 4.1 Antarmuka Menu Utama

Gambar 4.1 menjelaskan komponen-komponen menu utama yang terdapat pada antarmuka aplikasi Enkripsi SMS yaitu Tulis Pesan, Inbox dan Petunjuk.

4.2.2 Implementasi Tulis Pesan

Pada gambar 4.2 merupkan antarmuka tulis pesan aplikasi SMS.



Gambar 4.2 Antarmuka Tulis Pesan

Gambar 4.2 menjelaskan komponen-komponen yang terdapat pada antarmuka tulis pesan. Komponen tulis pesan terdiri antara lain :

1. Text field To yang berfungsi untuk mengisi no tujuan untuk penerima pesan.
2. Button Contact yang berfungsi untuk melink contact yang ada diponsel kita. Kemudian current key yang berfungsi untuk memasukkan kunci yang nantinya akan digunakan untuk proses enkripsi.
3. Text field pesan untuk mengisikan pesan yang akan kita kirim.
4. Button Enkripsi berfungsi untuk mengenkripsi pesan yang telah dibuat.
5. Button Kirim berfungsi untuk mengirimkan pesan kepada penerima.

4.2.3 Implementasi *Inbox*/Baca Pesan

Pada gambar 4.3 merupakan antarmuka Baca pesan aplikasi SMS.

Gambar 4.3 Antarmuka *Inbox*/Baca Pesan

Gambar 4.3 menjelaskan komponen-komponen yang terdapat pada antarmuka *Inbox.* Komponen *Inbox*/Baca Pesan antara lain:

1. Text field from menampilkan No pengirim
2. Current key : Kunci yang dikirim oleh pengirim.
3. Text field Pesan : Pesan enkripsi yang dikirmkan oleh pengirim.
4. Button Deskripsi : button yang berfungsi untuk mengenkripsi pesan.
5. Text Field Hasil : pesan asli yang dikirmkan oleh pengirim.

4.2.4 Implementasi Bantuan

Pada gambar 4.4 merupakan antarmuka bantuan aplikasi SMS.

Gambar 4.4 Antarmuka bantuan

Gambar 4.4 menjelaskan tentang bagaimana menggunakan aplikasi Enkripsi SMS pada antarmuka Bantuan.

4.3 Hasil Pengujian

Tabel 4.3 Hasil Pengujian

No	Kelas	Fungsi	Usecase	Skenario	Data Uji	Target	Pengujian	
							Benar	Tidak
1	MainActivity	Select		• User masuk aplikasi Enkripsi SMS	Tampil manual aplikasi Enkripsi SMS	Manual aplikasi SMS tampil	✓	
2	Tulis Pesan	Mengirim Pesan	Tulis Pesan	• User memasukkan no ponsel atau klik Button Contact. • User memasukkan kunci • User menulis pesan • User menekan button enkripsi • User menekan button kirim	**To:** 085765658590 **Kunci** : 123 **Data** : AES SMS AMAN **Enkripsi** : 3882181A5B3882181A5BAA93F64B3871237A6D343B	Pesan berhasil dikirim ke penerima dengan pesan berbentuk enkripsi	✓	

No	Kelas	Fungsi	Usecase	Skenario	Data Uji	Target	Pengujian	
							Benar	Tidak
3	Inbox	Menerima Pesan	Terima Pesan	• User menerima pesan • User membuka menu Inbox • User membuka pesan	Tampilan list pesan yang masuk ke inbox **Data** : 3882181A5B3882181A5BAA93F64B3871237A6D343B	Pesan berhasil diterima sesuai dengan pesan yang dikirim.	✓	
4	Baca Pesan	Baca Pesan	Baca Pesan	• User Menerima Pesan • User menekan tombol kunci • User Membaca Pesan	**From:** 085765658590 **Kunci** : 123 **Data:** 3882181A5B3882181A5BAA93F64B3871237A6D343B **Dekripsi** : AES SMS AMAN	Pesan berhasil dibaca oleh user	✓	
5	Bantuan	Membaca Bantuan	Bantuan	• User melihat bantuan dari aplikasi SMS	Menampilkan petunjuk cara penggunaan aplikasi	User berhasil membaca bantuan dari aplikasi SMS Enkripsi	✓	

BAB V
KESIMPULAN dan SARAN

Pada bab V ini akan diambil kesimpulan dari kegiatan-kegiatan yang telah dilakukan selama pengerjaan Tugas Akhir, selain itu terdapat saran-saran untuk pengembangan lebih lanjut yang dapat diberikan dari Tugas Akhir ini.

5.1 Kesimpulan

Kesimpulan yang didapat selama pengerjaan tugas akhir ini adalah sebagai berikut:

1. Penerapan algoritma kunci simetris untuk aplikasi AES SMS AMAN pada ponsel dapat meningkatkan keamanan karena pesan yang dikirimkan akan dienkripsi dan jika ingin membaca isi pesan maka harus dideskripsikan terlebih dahulu.
2. Aplikasi yang dibangun menggunakan algorimta AES. Salah satu keuntungan dari algoritma ini adalah pengelompokkan panjang kunci yang dapat dipilih dan setiap panjang kunci memiliki proses penyandian yang berbeda. Panjang kunci yang dipilih untuk membangun aplikasi AES SMS AMAN ini adalah AES 128 dengan menggunakan 10 putaran sehingga sulit untuk dideskripsikan. Selain kelebihan algoritma ini juga memiliki kekurangan yaitu jumlah karakter pesan yang dienkripsi lebih panjang dari pesan asli.

5.2 Saran

Berikut adalah saran-saran yang diberikan penulis untuk pengembangan lebih lanjut:

1. Pesan enkripsi yang dikirimkan memiliki panjang pesan yang tidak sama dengan *plainteks* yang ditulis oleh pengirim, sebaiknya diterapkan sebuah algoritma kompresi untuk melakukan kompresi pesan.
2. Algoritma yang digunakan dapat dikembangkan dengan algortima lain. Kemudian dibandingkan performanya antara algortima AES dengan algoritma lain misalnya RSA.

3. Bagi pengembang selanjutnya parameter kunci pada aplikasi dapat diganti nomor *handphone* penerima untuk mendeskripsikan pesan. Sehingga apabila ada pihak ketiga yang tidak berwenang ingin menyadap isi pesan maka pesan tersebut tidak dapat dibaca walaupun pihak ketiga tersebut memiliki aplikasi dan tingkat keamanan aplikasi menjadi lebih tinggi.

DAFTAR PUSTAKA

Ariyus. D, 2008, *Pengantar ilmu kriptografi teori analisis dan implementasi,* ANDI, Yogyakarta.

Ariyus. D, 2005, *Kriptografi Keamanan Data dan Komunikasi*, Andi Offset. Yogyakarta.

Ariyana. Y, 2011, *Advanced Encryption Standard (AES*), SSPPPPTK IPA Bandung.

Mariana, Sari. M , April 2013, *Perbandingan Algoritma AES dengan Algoritma XTS-AES untuk Enkripsi dan Dekripsi Teks SMS Berbasis Java ME*. STMIK MDP. *Volume 1, No 1*, http://eprints.mdp.ac.id/id/eprint/789, 09 September 2014.

Mubarak. Z. El-haq, 2012, *Implementation Of Cryptography Using Advanced Encryption Standard (AES) On Remoting Server Desktop Application By Mobile Phone,* Universitas Gunadarma. Depok.

Safaat, N. 2012 . *ANDROID – Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung : INFORMATIKA.

Syaifullah M. Acep, Mardion Yogi, 2014, *Proyek Akhir Mobile Aplication SMS Enkripsi* : Teknik Informatika. Batam.

Wibowo A. Wihartantyo, 2004, *Advanced Encryption Standard Algoritma Rijndael,* Institut Teknologi Bandung. Bandung.

Irwan, *Perancangan Aplikasi SMS (Short Message Service) dengan Enkripsi Teks Menggunakan Algoritma Block Chiper AES(Advanced Encryption Standard) Berbasis Mobile Pada Platform Android*, http://download.portalgaruda.org/article.php?article=32513&val=2313, (diakses pada tanggal 23 Januari 2015).

Satyanegara.B, *Penerapan Kriptografi dalam Sistem Keamanan SMS Banking*,http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2010-2011/Makalah1/Makalah1-IF3058-Sem1-2010-2011-006.pdf, (diakses pada tanggal 23 Januari 2015).