

PENERAPAN ALGORITMA HUFFMAN UNTUK APLIKASI PENGAMANAN SMS BERBASIS ANDROID

Fitri Purwaningsih¹, Mohammad Badrul²

Program Studi Sistem Informasi – STMIK Nusa Mandiri Jakarta

[1nuri_mbl@yahoo.com](mailto:nuri_mbl@yahoo.com), [2mohammad.mbl@nusamandiri.ac.id](mailto:mohammad.mbl@nusamandiri.ac.id)

Abstrak – Teknologi informasi dan komunikasi di dunia berkembang setiap waktu, salah satunya pada telepon genggam. Telepon seluler menyediakan media komunikasi yang beragam, salah satunya adalah SMS. Penggunaan SMS menjadi populer di kalangan masyarakat karena dengan begitu mudahnya kita dapat saling bertukar informasi tanpa batasan jarak dan waktu. Celah keamanan terbesar pada komunikasi *via* SMS adalah pesan yang dikirimkan akan disimpan di SMSC (*Short Message Service Center*), yaitu tempat dimana SMS disimpan sebelum dikirim ke tujuan. Pesan yang sifatnya *plaintext* ini dapat disadap oleh siapa saja yang berhasil memiliki akses ke dalam SMSC. Manfaat dari metode enkripsi digunakan untuk menjaga keamanan data pada pesan teks yang tersimpan di dalam memori ponsel agar terhindar dari pembajakan oleh pihak yang tidak diinginkan dan data-data yang dikirimkan diubah sedemikian rupa sehingga tidak mudah disadap.

Kata Kunci : sms, huffman, telepon selular

I. PENDAHULUAN

Teknologi informasi dan komunikasi di dunia berkembang setiap waktu, salah satunya pada telepon genggam. Dengan kemajuan teknologi informasi dan komunikasi orang dapat melakukan komunikasi dan transaksi tanpa batas waktu dan tempat. Terobosan baru pada telepon genggam ditandai dengan munculnya sistem operasi android.

Dengan bertambahnya populasi dunia yang menggunakan telepon seluler, jumlah operator seluler juga semakin bertambah. Berdasarkan dari *survei* yang dilakukan oleh *United Nation*, 60% dari populasi dunia kini menggunakan telepon seluler dengan kadar subskripsi 4,1 milyar per tahun. Data menunjukkan peningkatan pengguna telepon seluler kira-kira 1 milyar semenjak tahun 2002. Pengguna telepon seluler di Indonesia tercatat sebanyak 68 juta pada akhir tahun 2006 dan akan tumbuh menjadi 94,7 juta pada tahun 2007. Pada tahun 2010, angka pengguna telepon seluler di Indonesia pun diprediksikan mencapai angka 133 juta. Dengan kata lain, sekitar separuh dari seluruh populasi negeri ini yang diperkirakan mencapai 250 juta jiwa, merupakan pengguna telepon seluler.

Telepon seluler menyediakan media komunikasi yang beragam, salah satunya adalah SMS. Penggunaan SMS menjadi populer di kalangan masyarakat karena dengan begitu mudahnya kita dapat saling bertukar informasi tanpa batasan jarak dan waktu. Celah keamanan terbesar pada komunikasi *via* SMS adalah pesan yang dikirimkan akan disimpan di SMSC (*Short Message Service Center*), yaitu tempat dimana SMS

disimpan sebelum dikirim ke tujuan. Pesan yang sifatnya *plaintext* ini dapat disadap oleh siapa saja yang berhasil memiliki akses ke dalam SMSC. Akibatnya, informasi penting seperti *password*, nomor pin, dan lain-lain dapat dibaca oleh orang yang tidak berhak untuk mengetahuinya[7]. Manfaat dari metode enkripsi digunakan untuk menjaga keamanan data pada pesan teks yang tersimpan di dalam memori ponsel agar terhindar dari pembajakan oleh pihak yang tidak diinginkan.

II. METODOLOGI PENELITIAN

Untuk mempermudah penelitian ini, beikut penulis jabarkan analisa penelitian yang penulis gunakan :

- a. *Planning*
Tahap ini penulis melandasi permasalahan yang penulis lakukan dengan cara studi literatur dari beberapa jurnal ilmiah yang relevan dengan penelitian yang penulis bahas.
- b. Analisis
Dalam tahap ini, penulis melakukan analisa terhadap permasalahan yang yang akan penulis lakukan dengan melihat lebih detail metode pengamanan apa yang akan digunakan untuk penelitian ini.
- c. Desain
Tahap perancangan (*design*) memiliki tujuan untuk mendesain aplikasi yang dapat melakukan pengamanan terhadap sms yang ada di perangkat mobile. Kegiatan yang dilakukan dalam tahap ini meliputi perancangan *input* dan *output*. Langkah-langkah yang dilakukan adalah menyiapkan rancangan aplikasi yang terinci. Dalam tahap desain dilakukan penentuan grafis dari pembuatan *interface*.

Penentuan dimulai dari komposisi tampilan dengan rancangan program yang ada sehingga membentuk tampilan yang menarik.

d. Implementasi

Tahap implementasi memiliki beberapa tujuan, yaitu untuk melakukan kegiatan spesifikasi rancangan logika ke dalam kegiatan yang sebenarnya dan dari aplikasi yang akan dikembangkan, kemudian mengimplementasikan aplikasi tersebut kedalam salah satu bahasa pemrograman yang paling sesuai. Kegiatan pada tahap ini juga menjamin bahwa aplikasi dapat berjalan secara optimal.

III. DASAR TEORI

Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi[1]. Aplikasi android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Android memiliki beberapa fitur yang menarik yang ingin mengembangkan aplikasi, di antaranya sebagai berikut:

- a. *Application framework* yang memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- b. *Dalvik Virtual Machine*, yaitu mesin yang dioptimalkan untuk perangkat *mobile*.
- c. *Graphic Library*, yang mendukung grafik 2D, dan 3D berdasarkan *OpenGL Library*.
- d. *Media Supported*, yang mendukung beberapa media seperti: *audio*, *video*, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- e. *Hardware Independent*, mendukung GSM, *Bluetooth*, EDGE, 3G, *Wifi*, kamera, *Global Positioning System* (GPS), kompas, dan *accelerometer*.

Java

Bahasa *Java* merupakan karya Sun Microsystem Inc. Rilis resmi *level beta* dilakukan pada November 1995. Dua bulan berikutnya *Netscape* menjadi perusahaan pertama yang memperoleh lisensi bahasa *Java* dari Sun. Sebagian besar bahasa pemrograman modern berdiri di atas pustaka-pustaka kelas yang telah ada untuk mendukung fungsionalitas. Pada bahasa *Java*, kelompok-kelompok kelas yang berkaitan erat dimasukkan di satu paket, bervariasi sesuai edisi *Java*. Masing-masing paket untuk maksud tertentu yaitu *applet*, aplikasi standar, skala *enterprise*, dan produk konsumen.

Eclipse

Eclipse adalah IDE (*Integrated Development Environment*) merupakan aplikasi pengembangan *java/android*, *eclipse* memiliki *plugin* yang dapat membuat proyek yang berbasis android. ADT (*Android*

Development Tools)*Eclipse* sangat diminati oleh para pengembang perangkat lunak, ini disebabkan karena *eclipse* merupakan *software free* dan *open source*. *Eclipse* juga menyediakan *plugins* untuk mengembangkan perangkat lunak yang digunakan untuk memodifikasi dan membuat aplikasi-aplikasi yang inovatif. Pada saat ini *eclipse* merupakan salah satu IDE favorit karena gratis dan *open source* berarti setiap orang boleh melihat kode program perangkat lunak ini. Selain itu kelebihan dari *eclipse* yang membuatnya populer adalah kemampuannya untuk dikembangkan oleh pengguna dengan membuat komponen yang disebut *plugin*.

Android SDK

Android SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman *java*[1]. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Saat ini disediakan android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk memulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemrograman *java*.

Android Development Tools

Android Development Tools (ADT) adalah *plugin* yang didesain untuk IDE *Eclipse* yang memberi kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE *Eclipse*[1]. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan dalam membuat aplikasi projek android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, serta dapat melakukan *running* aplikasi menggunakan Android SDK melalui *Eclipse*. Dengan ADT juga dapat melakukan pembuatan *package android* (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Algoritma Huffman

Enkripsi adalah salah satu cara yang digunakan untuk mengubah teks "*asli*" (sebenarnya) menjadi teks "buatan"[2]. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus". Dikarenakan enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Dekripsi adalah proses mengembalikan pesan yang dalam kondisi tersandikan sehingga dapat dibaca lagi apa pesan sebenarnya[3].

Kode Huffman adalah algoritma yang menggunakan frekuensi/probabilitas kemunculan dari simbol pada sebuah *string* sebagai *input* dan menghasilkan *output* berupa *prefix code* yang mengkodekan *string* menggunakan *bit* paling sedikit dari seluruh kemungkinan *binary prefix code* yang

mungkin[4]. Algoritma ini dikembangkan oleh David A. Huffman". Huffman Coding menggunakan struktur pohon dalam pemrosesannya. Pohon adalah *graf* tak berarah yang tidak mengandung sirkuit. Di dalam struktur pohon dikenal terminologi *parent* (orang tua) dan *child* (anak). *Parent* (orang tua) yaitu sebuah simpul yang memiliki lintasan ke simpul lain dengan tingkatan di bawahnya. *Child* yaitu sebuah simpul yang memiliki lintasan ke simpul lain dengan tingkatan di atasnya[8]. Berdasarkan jumlah anak, pohon dapat dikategorikan sebagai pohon *n-ary*. Pohon dengan orang tua yang hanya memiliki satu anak, disebut pohon *uner*. Pohon dengan orang tua yang memiliki dua anak, disebut pohon *biner*, dan seterusnya. Pohon Huffman menggunakan struktur pohon *biner*, yaitu struktur pohon dengan setiap simpul orang tua yang hanya memiliki maksimal dua simpul anak. Penyusunan data dalam struktur pohon Huffman menggunakan kode awalan (*prefix code*). Kode awalan adalah himpunan kode (*biner*) dimana anggota kumpulan yang satu bukan merupakan anggota kumpulan yang lain. Huffman adalah mengkodekan setiap karakter ke dalam representasi bit[8]. Representasi bit untuk setiap karakter berbeda satu sama lain berdasarkan frekuensi kemunculan karakter. Semakin sering karakter tersebut muncul, maka semakin pendek panjang representasi bitnya. Sebaliknya bila semakin jarang frekuensi suatu karakter untuk muncul, maka semakin panjang representasi bit untuk karakter tersebut[8]. Dalam perkembangannya, pengimplementasinya bisa juga digunakan dalam bidang kriptografi atau untuk pengkompresi jenis data lainnya.

Pengujian Sistem

Ada 2 metode untuk melakukan unit *testing* yaitu:

a. Black Box Testing

Pengujian *black box* sering disebut juga dengan pengujian tingkah laku yang lebih terfokus kepada kebutuhan fungsional dari perangkat lunak. Pengujian *black box* memungkinkan pembuat perangkat lunak untuk menentukan kondisi yang terjadi untuk suatu masukan yang akan menjalankan semua kebutuhan fungsional[5].

b. White Box Testing

White box testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisa apakah ada kesalahan atau tidak. Jika ada modul yang menghasilkan *output* yang tidak sesuai dengan proses bisnis yang dilakukan, maka baris-baris program, variabel, dan parameter yang terlibat pada unit tersebut akan di cek satu persatu dan diperbaiki, kemudian di-*compile* ulang.

Pengujian Sistem

Jenis diagram UML antara lain:

a. Diagram Kelas.

Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi[6]. Diagram ini umum dijumpai pada pemodelan sistem

berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. Diagram kelas adalah inti dari proses pemodelan objek dan merupakan kumpulan kelas-kelas objek. Oleh karena itu pengertian kelas sangat penting sebelum merancang diagram kelas.

b. Diagram Use-Case.

Bersifat statis. Diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas)[6]. Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna. Komponen pembentuk diagram *use -case* adalah:

- 1) Aktor (*actor*), menggambarkan pihak-pihak yang berperan dalam sistem.
- 2) *Use-case*, aktivitas/sarana yang disiapkan bisnis/sistem.
- 3) Hubungan (*link*), aktor mana saja yang terlibat dalam *use-case* ini.

c. Diagram Interaksi dan Sequence (urutan).

Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.

d. Diagram Aktivitas (Activity Diagram).

Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram aktivitas menunjukkan aktivitas dalam bentuk kumpulan aksi-aksi.

e. Deployment Diagram

Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya.

IV. TEKNIK PENGUMPULAN DATA

Dalam memudahkan pembuatan dan pengumpulan data-data yang diperlukan dalam penelitian ini, maka peneliti menggunakan metode penelitian sebagai berikut:

a. Analisa Kebutuhan

Menganalisa *hardware* dan *software* yang digunakan untuk membuat aplikasi pengamanan SMS. *Hardware* menggunakan *notebook* PC dan *software* yang digunakan adalah sistem operasi *Windows 8*, *Eclipse*, ADT versi 21.1.0, *Android SDK* dan *Java JDK*.

b. Desain

Merancang alir kerja dari sistem dalam bentuk diagram UML (*Unified Modelling Language*) yaitu *class* diagram, *use-case* diagram, *activity* diagram, dan lain-lain.

c. Testing

Tahap ini dilakukan pengujian terhadap sistem dengan tujuan untuk melihat semua kesalahan dan kekurangan pada sistem. Pengujian sistem dilakukan dengan metode *blackbox* dan *whitebox*.

d. Implementasi

Instal aplikasi ke perangkat *mobile* dengan operasi sistem android. Maka aplikasi siap digunakan.

V. ANALISA KEBUTUHAN

Pada pembangunan aplikasi ini dibutuhkan spesifikasi minimal *hardware* sebagai berikut :

Tabel 1. Kebutuhan Hardware

No	Nama Hardware	Spesifikasi	Jml
1	Laptop	Processor : Intel Core i5 RAM : 4 GB Hardisk : 750 GB	1
2	Handphone	Processor : 1 GHz Qualcomm	1

Software yang digunakan untuk pembangunan aplikasi sebagai berikut.

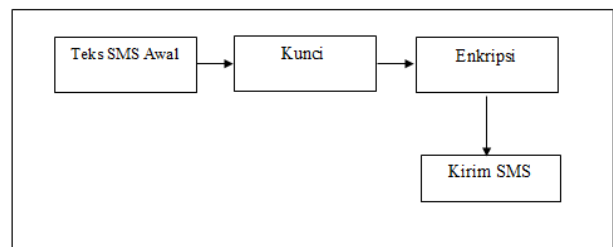
Tabel 2. Kebutuhan Software

No	Kebutuhan	Keterangan	Fungsi
1	Sistem Operasi	Windows 7 Ultimate 64 Bit	Sebagai Sistem Operasi yang dipakai untuk menjalankan Software
2	Aplikasi	Java Development Kit 7	Develop program
		Eclipse Indigo	Tampilan interface program
		ADT-18 Plugin	Android plugin
		Android SDK Manager	Emulator Android di PC

VI. PERANCANGAN

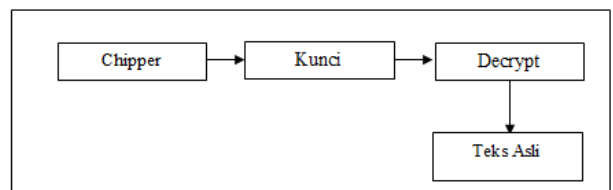
1. Rancangan Algoritma Pada Kasus

Rancangan algoritma pada pembahasan ini dibagi menjadi dua yaitu proses enkripsi dan dekripsi. Dalam aplikasi penangamanan SMS ini, penulis akan menggunakan metode algoritma “*Huffman*”, di mana metode ini digunakan untuk merahasiakan pesan teks SMS.



Gambar 1. Proses Enkripsi

Gambar di atas menunjukkan proses Enkripsi yang dilakukan untuk enkripsi teks SMS. Proses meng-*input*-kan teks SMS kemudian pesan tersebut akan dienkripsi sebelum mengirimkan SMS.



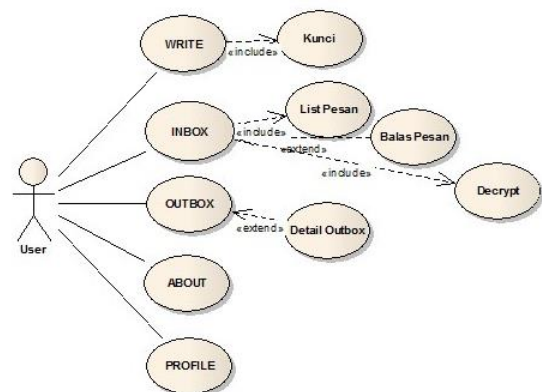
Gambar 2. Proses Dekripsi

Sedangkan gambar di atas merupakan proses yang menggambarkan dekripsi teks SMS yang diterima. Proses ini di mulai dengan pesan *chipper* teks, karena teks SMS yang diterima merupakan teks yang telah di enkripsi. Langkah selanjutnya adalah mendekripsi teks SMS yang sudah di enkripsi tadi.

2. Model UML

a) Use case diagram

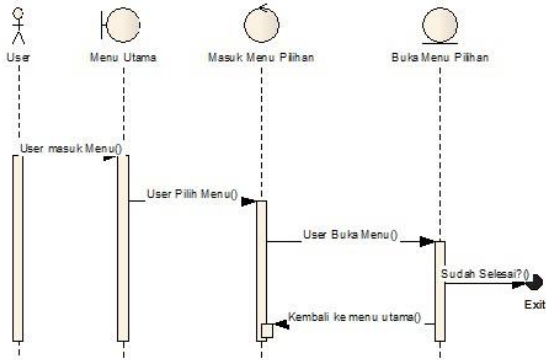
Adapun *software* arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan diagram *Use Case* adalah sebagai berikut :



Gambar 2. Use Case Diagram

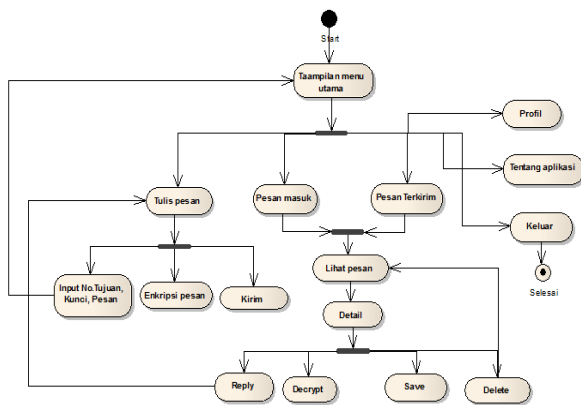
b) Sequence diagram

Adapun *software* arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan Diagram *Sequence* adalah sebagai berikut :



Gambar 3. Sequence Diagram

c) Activity Diagram
Adapun *software* arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan Diagram Activity adalah sebagai berikut :



Gambar 4. Activity Diagram

VII. HASIL DAN PEMBAHASAN

Untuk melakukan pengujian pada aplikasi yang telah dibuat penulis menggunakan metode *BlackBox* dan *WhiteBox testing*. Pengujian dibagi 2, yaitu pengujian *interface* aplikasi, dan pengujian *form handle* aplikasi.

1. Pengujian *Black Box*

a. Pengujian *Interface* Aplikasi

Pengujian *interface* aplikasi bertujuan untuk mengetahui fungsionalitas dari elemen-elemen *interface* yang terdapat di dalam halaman pada aplikasi. Elemen-elemen *interface* yang diujikan terutama adalah elemen *button*.

Tabel 3. Pengujian *Interface* Aplikasi

Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Ket
Pengujian <i>Image View Tulis</i>	Aplikasi akan menampilkan halaman buat pesan	Aplikasi menampilkan halaman buat pesan baru	Sesuai
Pengujian <i>Image View Inbox</i>	Aplikasi akan menampilkan halaman pesan masuk	Aplikasi menampilkan halaman daftar pesan masuk	Sesuai
Pengujian	Aplikasi akan	Aplikasi	Sesuai

<i>Image View Outbox</i>	menampilkan halaman keluar	menampilkan halaman daftar pesan keluar	
Pengujian <i>Image View Tentang Aplikasi</i>	Aplikasi akan menampilkan info pesan berupa <i>alert dialog</i>	Aplikasi menampilkan info pesan berupa <i>alert dialog</i>	Sesuai
Pengujian <i>Image View Profil</i>	Aplikasi akan menampilkan info pesan berupa <i>alert dialog</i>	Aplikasi menampilkan info pesan berupa <i>alert dialog</i>	Sesuai
Pengujian <i>button Enkripsi</i>	Teks yang dibuat pada form buat pesan akan berubah menjadi <i>ciphertext</i>	Teks yang dibuat pada form buat pesan berubah menjadi <i>ciphertext</i>	Sesuai
Pengujian <i>Image View kontak</i>	Aplikasi akan masuk ke daftar kontak pada ponsel	Aplikasi masuk ke daftar kontak pada ponsel	Sesuai
Pengujian <i>button Reply</i>	Aplikasi akan masuk ke halaman tulis pesan	Aplikasi masuk ke halaman tulis pesan	Sesuai
Pengujian <i>button Decrypt</i>	Aplikasi akan menerjemahkan pesan yang telah dienkripsi	Aplikasi telah menerjemahkan pesan yang telah dienkripsi	Sesuai
Pengujian <i>button Clear</i>	Tombol ini untuk menghapus	Tombol ini telah menghapus	Sesuai

b. Pengujian *Form Handle* Aplikasi

Pengujian *form handle* aplikasi bertujuan untuk mengetahui kemampuan aplikasi untuk menanggapi bermacam-macam *input* yang diberikan oleh *user*.

Tabel 4. Pengujian *Form Handle* Aplikasi

Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Ket
Pengujian kirim pesan dengan data yang benar, yaitu form nomor tujuan, kunci dan nomor tujuan di isi	Pesan akan terkirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa pesan terkirim	Pesan terkirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa pesan terkirim	Sesuai
Pengujian kirim pesan dengan data yang salah, yaitu form nomor tujuan tidak di isi	Pesan tidak akan terkirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa nomor tujuan belum di isi	Pesan tidak bisa dikirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa nomor tujuan belum di isi	Sesuai
Pengujian kirim pesan dengan data yang salah, yaitu salah satu dari form pesan kunci tidak di isi	Pesan tidak akan terkirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa kunci belum di isi	Pesan tidak bisa dikirim dan memberikan <i>feedback</i> pada <i>user</i> bahwa kunci belum di isi	Sesuai

2. Implementasi



Gambar 5. Menu Utama



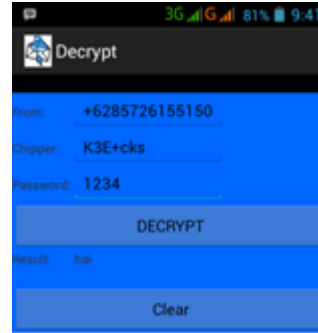
Gambar 6. Menu Tulis Pesan



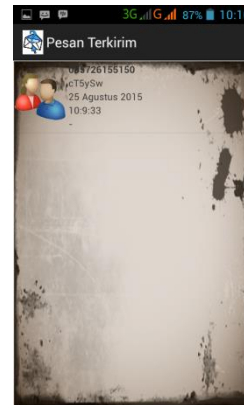
Gambar 7. Menu Pesan Masuk



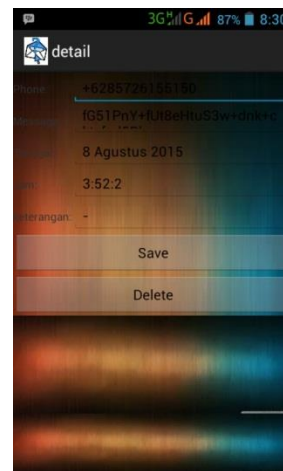
Gambar 8. Menu Detail Inbox



Gambar 9. Menu Decrypt



Gambar 10. Menu Pesan Terkirim



Gambar 11. Menu Detail Outbox

VIII. SIMPULAN

Berdasarkan dari hasil yang diperoleh penulis, maka dapat diambil kesimpulan sebagai berikut:

1. Aplikasi pengamanan SMS ini sudah dapat menjalankan perintah dengan baik dengan mengamankan pesan teks dengan menggunakan kunci/*password*.
2. Tombol enkrip dan dekrip sudah dapat mengenkripsi dan dekripsi pesan teks dengan baik sesuai algoritma yang diterapkan.
3. Aplikasi pengamanan SMS tersebut berhasil dibangun menggunakan bahasa pemrograman *java*

DAFTAR PUSTAKA

- [1] Safaat, Nazruddin. 2014. Pemrograman Aplikasi MobileSmartphone dan Tablet PC Berbasis Android. Bandung: Informatika, 2014
- [2] Juju, Dominikus dan Matamaya Studio. 2009. Teknik Menangkal Kejahatan Internet Untuk Pemula. Jakarta: PT.Elex media Komputindo.
- [3] Arryawan Eko dan SmitDev Community. 2010. Mengatasi Investigasi Komputer Forensik. Jakarta: PT. Elex Media Komputindo
- [4] Nugroho, Heru. 2015. Matematika Diskrit dan Implementasinya Dalam Dunia Teknologi. Yogyakarta: CV. Budi Utama.
- [5] Rusadi, Wildan Agissa. (2013). Whitebox Testing & Blackbox Testing.
- [6] Widodo, Prabowo pudjo. 2011. Menggunakan UML. Bandung: Informatika Bandung.
- [7] Primartha, Rifkie. 2011. Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma *Data Encryption Standard* (DES). ISSN : 2085-1588. Palembang: Jurnal Sistem Informasi (JSI), VOL. 3, NO. 2, Oktober 2011: 371-387. Diambil dari: <http://ejournal.unsri.ac.id/index.php/jsi/index.21> (Mei 2015).
- [8] Syahputra, Agus dan Harvei Desmon Hutahaeen. 2015. Analisa Perbandingan Algoritma *Huffman* Dan *Run Length* Pada Citra *Wavelet Multi Dimensi*. ISSN : 2339-210X. Medan: Majalah Ilmiah Vol: V, No: 3 Februari 2015: 1- 6. (21 Mei 2015)