

PENERAPAN ALGORITMA HUFFMAN UNTUK APLIKASI PENGAMANAN SMS PADA *MOBILE PHONE* BERBASIS ANDROID

Fitria Purwaningsih¹, Desmulyati²

Program Studi Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer Nusa Mandiri Jakarta
fitriappurwaningsih@gmail.com¹, desmulyati.dmy@bsi.ac.id²

Abstrak – Perkembangan perangkat *mobile phone* yang dikenal dengan telepon genggam atau telepon seluler pada teknologi informasi dan komunikasi saat ini semakin marak. Dimana telepon seluler juga menyediakan media komunikasi yang beragam, salah satunya adalah SMS. Penggunaan SMS menjadi populer di kalangan masyarakat karena dengan begitu mudahnya 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. Menggunakan Algoritma Huffman untuk mengkodekan setiap karakter ke dalam representasi bit. Aplikasi ini diterapkan pada perangkat Android, menggunakan bahasa pemrograman *Java*, dengan *tools system* UML (*Unified Modeling Language*), untuk membangun aplikasi pengamanan SMS yang mampu mengenkripsi dan dekripsi pesan teks sesuai algoritma yang diterapkan.

Kata kunci: Algoritma Huffman, Android, Enkripsi, SMS

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 diprediksi 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 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 mengetahuinya. 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.

Primartha (2011:371) menyatakan bahwa salah satu mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang dikirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Jadi enkripsi adalah proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah enkripsi (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Terminologi yang lebih tepat digunakan adalah "*encipher*". Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut dekripsi (*decryption*). Terminologi yang lebih tepat untuk proses ini adalah "*decipher*".

II. KAJIAN PUSTAKA

Android

Safaat (2014:1) mengemukakan bahwa Android adalah sebuah sistem operasi untuk perangkat *mobile*

berbasis *linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. Aplikasi android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Fitur-Fitur Android

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

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

Java

Bahasa *Java* merupakan karya Sun Microsystems 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

Menurut Huda (2009:2) menyatakan bahwa, "*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 (*Software Development Kit*)

Safaat (2014:5) mengemukakan bahwa "Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada *platform android* menggunakan bahasa pemrograman *java*."

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 (ADT)

Safaat (2014:6) mengemukakan bahwa "*Android Development Tools (ADT)* adalah *plugin* yang didesain untuk IDE *Eclipse* yang memberi kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE *Eclipse*."

Dengan menggunakan ADT untuk *Eclipse* akan memudahkan dalam membuat aplikasi proyek 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 dirancang.

Metode Enkripsi dan Dekripsi

Juju (2008:181) mengemukakan bahwa "Enkripsi adalah salah satu cara yang digunakan untuk mengubah teks "*asli*" (sebenarnya) menjadi teks "*buatan*". 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.

Arryawan (2010:74) mengemukakan bahwa "Dekripsi adalah proses mengembalikan pesan yang dalam kondisi tersandikan sehingga dapat dibaca lagi apa pesan sebenarnya".

Algoritma Huffman

Menurut Nugroho (2015:11) "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. 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. 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. Representasi bit untuk setiap karakter berbeda satu sama lain berdasarkan frekuensi kemunculan karakter. Semakin sering karakter tersebut muncul, maka semakin pendek panjang representasi bit-nya. Sebaliknya bila semakin jarang frekuensi suatu karakter untuk muncul, maka semakin panjang representasi bit untuk karakter tersebut. Dalam perkembangannya, implementasinya bisa juga digunakan dalam bidang kriptografi atau untuk pengkompresi jenis data lainnya.

Pengujian Sistem

Ada 2 metode untuk melakukan unit *testing* yaitu:

1. *Black Box Testing*
Menurut Ayuliana (2009:3) mengemukakan bahwa "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."
2. *White Box Testing*
White box testing adalah cara pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis 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 dicek satu persatu dan diperbaiki, kemudian di-*compile* ulang.

UML (*Unified Modeling Language*)

Jenis diagram UML antara lain:

- a. Diagram Kelas
Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi.

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). 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. 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 di dalamnya.

III. METODE PENELITIAN

Teknik Pengumpulan Data

- a. Observasi
Metode ini dilakukan dengan mencari dan membandingkan aplikasi sejenis yang telah lebih dahulu dibuat.
- b. Wawancara
Metode pengumpulan data yang dilakukan dengan cara bertanya kepada narasumber yang dianggap mengerti dengan permasalahan yang dibahas, yaitu Manager perusahaan dengan asisten, atau perorangan.
- c. Studi Kepustakaan
Pengambilan data dengan cara mengambil materi-materi yang berhubungan dengan judul karya ilmiah melalui buku-buku, studi literatur dan jurnal ilmiah.

Metode Penelitian

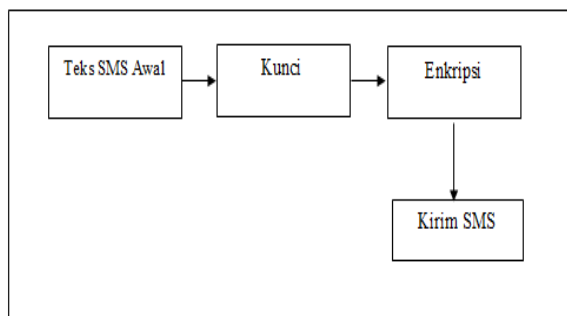
- a. Analisis Kebutuhan
Menganalisis *hardware* dan *software* yang digunakan untuk membuat aplikasi pengamanan

- 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. *Code Generation*
Tahap *coding* merupakan tahap pengkodean dari desain ke dalam bahasa pemrograman. Dalam aplikasi ini pengkodean dibuat menggunakan *eclipse*.
 - d. *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*.
 - e. *Implementasi*
Instal aplikasi ke perangkat *mobile* dengan operasi sistem android. Maka aplikasi siap digunakan.

IV. HASIL DAN PEMBAHASAN

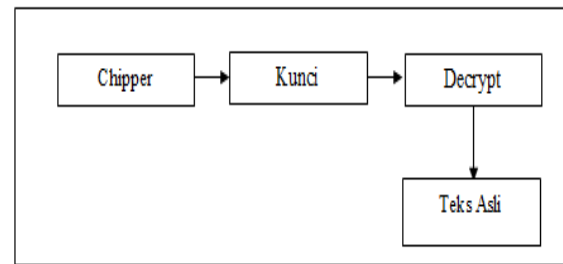
Rancangan Algoritma pada Kasus

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



Gambar 1. Proses Enkripsi

Gambar 1 menunjukkan proses Enkripsi yang dilakukan untuk enkripsi teks SMS. Proses meng-input-kan teks SMS kemudian pesan tersebut akan dienkripsi sebelum mengirimkan SMS.

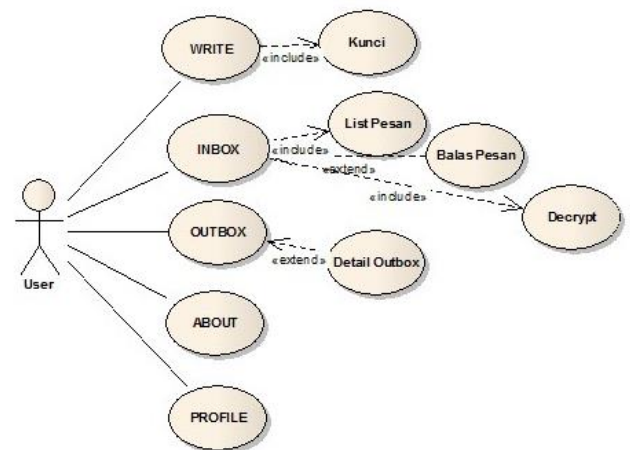


Gambar 2. Proses Dekripsi

Sedangkan gambar 2 merupakan proses yang menggambarkan dekripsi teks SMS yang diterima. Proses ini dimulai dengan pesan *chipper* teks, karena teks SMS yang diterima merupakan teks yang telah dienkripsi. Langkah selanjutnya adalah mendekripsi teks SMS yang sudah dienkripsi.

Model UML

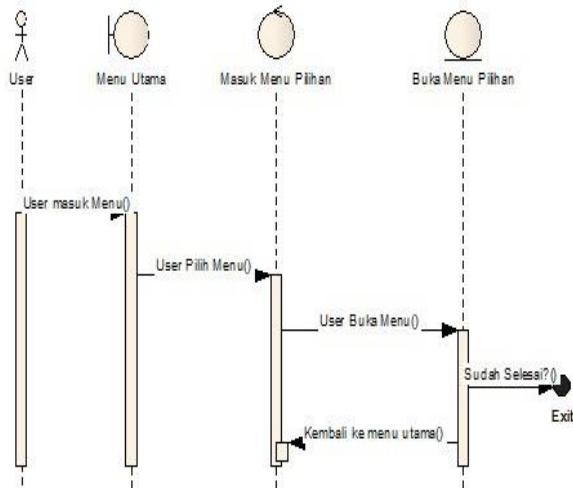
a) Use case diagram



Gambar 3. Use Case Diagram

b) Sequence diagram

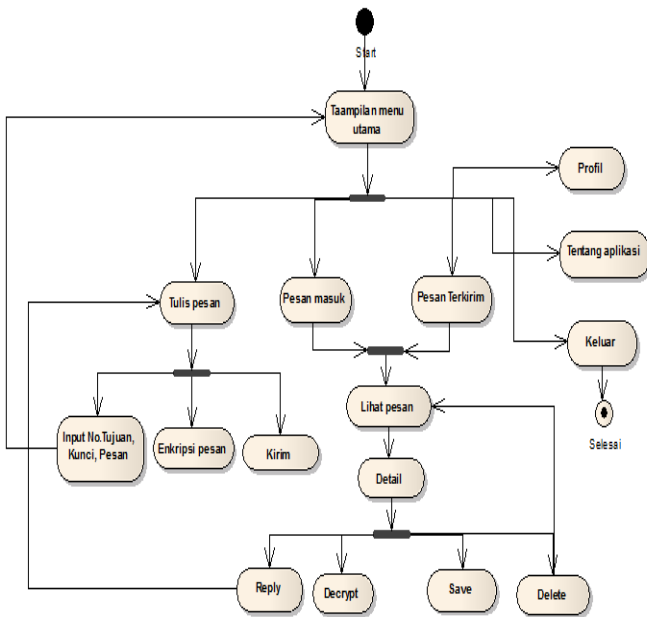
Adapun *software* arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan Diagram *Sequence* adalah sebagai berikut :



Gambar 4. Sequence Diagram

c) Activity Diagram

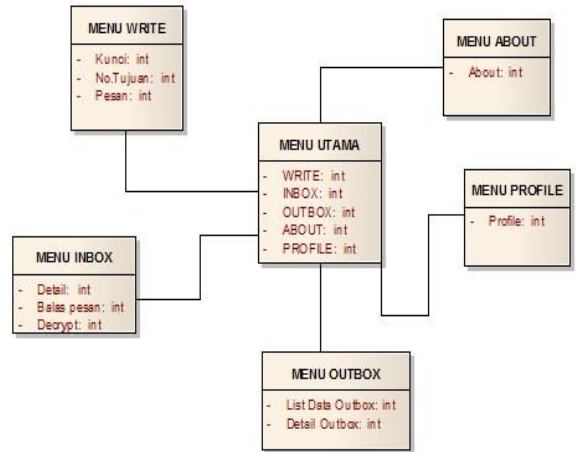
Adapun software arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan Diagram Activity adalah sebagai berikut :



Gambar 5. Activity Diagram

d) Class Diagram

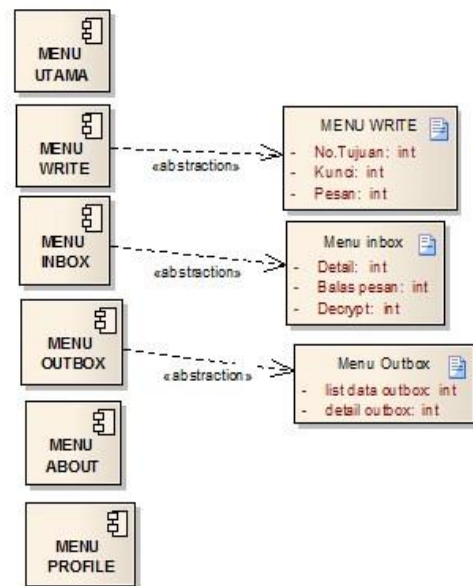
Adapun software arsitektur untuk perancangan sistem aplikasi pengamanan SMS ini dengan menggunakan Diagram Class adalah sebagai berikut :



Gambar 6. Class Diagram

e) Deployment diagram

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



Gambar 7. Deployment Diagram

Testing

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

Pengujian Interface Aplikasi

Pengujian ini bertujuan untuk mengetahui apakah *interface* aplikasi sudah berfungsi dengan baik atau belum. Hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Pengujian Interface Aplikasi

No	Test Case	Hasil yang diharapkan	Hasil yang didapatkan	Ket
1.	Pengujian Image View Tulis	Aplikasi akan menampilkan halaman buat pesan	Aplikasi menampilkan halaman buat pesan baru	Berhasil
2.	Pengujian Image View Inbox	Aplikasi akan menampilkan halaman pesan masuk	Aplikasi menampilkan halaman daftar pesan masuk	Berhasil
3.	Pengujian Image View Outbox	Aplikasi akan menampilkan halaman keluar	Aplikasi menampilkan halaman daftar pesan keluar	Berhasil
4.	Pengujian Image View Tentang Aplikasi	Aplikasi akan menampilkan info pesan berupa alert dialog	Aplikasi menampilkan info pesan berupa alert dialog	Berhasil
5.	Pengujian Image View Profil	Aplikasi akan menampilkan info pesan berupa alert dialog	Aplikasi menampilkan info pesan berupa alert dialog	Berhasil
6.	Pengujian button Enkripsi	Teks yang dibuat pada form buat pesan akan berubah menjadi ciphertext	Teks yang dibuat pada form buat pesan berubah menjadi ciphertext	Berhasil
7.	Pengujian Image View kontak	Aplikasi akan masuk ke daftar kontak pada ponsel	Aplikasi masuk ke daftar kontak pada ponsel	Berhasil
8.	Pengujian button Reply	Aplikasi akan masuk ke halaman tulis pesan	Aplikasi masuk ke halaman tulis pesan	Berhasil
9.	Pengujian button Decrypt	Aplikasi akan menerjemahkan pesan yang telah dienkripsi	Aplikasi telah menerjemahkan pesan yang telah dienkripsi	Berhasil
10.	Pengujian button Clear	Tombol ini untuk menghapus	Tombol ini telah menghapus	Berhasil

2. Pengujian WhiteBox

Kompleksitas siklomatis dari grafik alir whitebox dapat diperoleh dengan perhitungan:

$$V(G)=E-N+2 \dots\dots\dots (1)$$

Dimana :

- E = Jumlah edge yang ditentukan gambar panah
- N = Jumlah simpul grafik alir ditentukan dengan gambar lingkaran

$$V(G) = 22 - 18 + 2 = 6$$

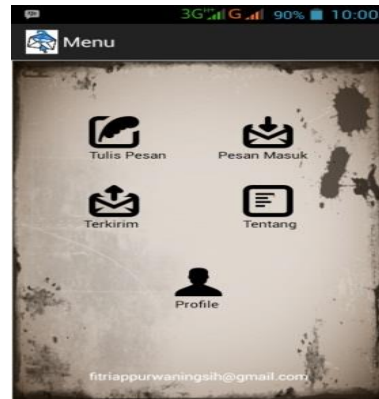
V(G) < 10 berarti memenuhi syarat kompleksitasi siklomatisnya. Baris set yang menghasilkan dari jalur independent adalah sebagai berikut:

- a. 1-2-3-10-15-17-2-9
- b. 1-2-3-4-11-16-18-9
- c. 1-2-3-4-5-12-9
- d. 1-2-3-4-5-6-13-9
- e. 1-2-3-4-5-6-7-14-9
- f. 1-2-3-4-5-6-7-8-9

- g. Ketika aplikasi dijalankan, maka terlihat bahwa satu set baris yang dihasilkan adalah 1-2-3-10-15-17-2-9-1-2-3-4-11-16-18-9-1-2-3-4-5-12-9-1-2-3-4-5-6-13-9-1-2-3-4-5-6-7-14-9-1-2-3-4-5-6-7-8-9

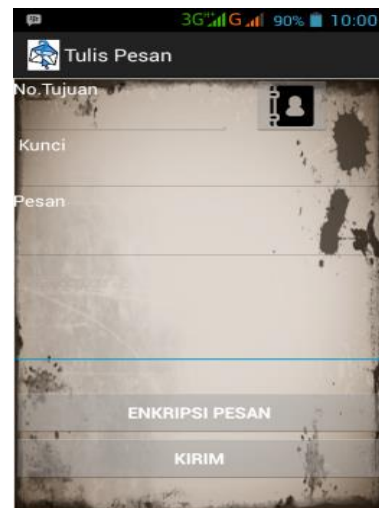
Implementasi

a. Menu Utama



Gambar 8. Tampilan Menu Utama

b. Menu Tulis Pesan



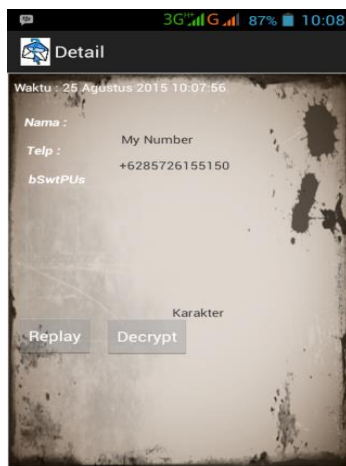
Gambar 9. Tampilan Menu Tulis Pesan

c. Menu Pesan Masuk



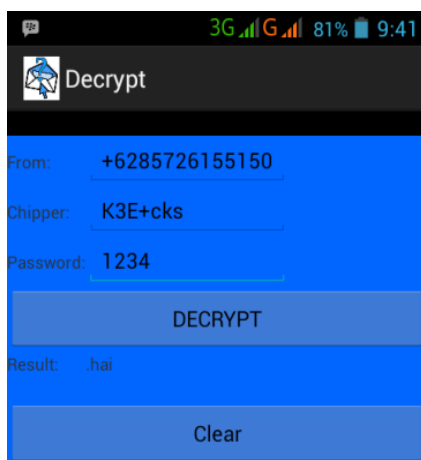
Gambar 10. Tampilan Menu Pesan Masuk

d. Menu *Detail Inbox*



Gambar 11. Tampilan Menu *Detail Inbox*

e. Menu *Decrypt*



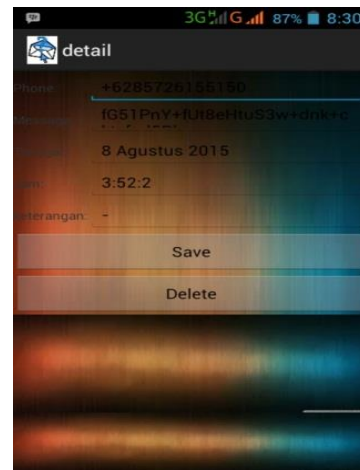
Gambar 12. Tampilan Menu *Decrypt*

f. Menu *Pesan Terkirim*



Gambar 13. Tampilan Menu *Pesan Terkirim*

g. Menu *Detail Outbox*



Gambar 14. Tampilan Menu *Detail Outbox*

V. PENUTUP

Kesimpulan

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*.

Saran

Dari beberapa kesimpulan yang telah dikemukakan, maka penulis mencoba memberikan beberapa saran sebagai alternatif yang dapat dijadikan masukan-masukan yang nantinya akan berguna dalam pembuatan aplikasi. Agar aplikasi ini

dapat menjadi lebih baik terdapat beberapa saran yang dapat dipergunakan diantaranya:

1. Aplikasi yang dikembangkan lebih lanjut diharapkan dapat diimplementasikan dalam aplikasi pengamanan SMS yang lebih baik.
2. Aplikasi ini dapat dikembangkan tidak hanya berbasis android namun juga berbasis *web*.
3. Penelitian yang telah dilakukan membuat suatu aplikasi pengamanan SMS menggunakan bahasa pemrograman *Java*. Perlu dilakukan penelitian untuk membuat aplikasi berdasarkan algoritma *huffman* menggunakan bahasa pemrograman yang berbeda. Bahasa C, C++ atau Pascal dirasa perlu untuk dicoba mengingat masing-masing bahasa pemrograman memiliki karakteristik serta kelebihan masing-masing.

DAFTAR PUSTAKA

- Arrayawan Eko dan SmitDev *Community*. 2010. Mengatasi Investigasi Komputer Forensik. Jakarta: PT. Elex Media Komputindo. (4 Agustus 2015)
- Ayuliana. 2009. Testing dan Implementasi. https://www.google.com.sg/search?q=jurnal+n+pengamanan+sm&oq=jurnal+issn+&aqs=chrome.0.69i59l3j69i57.4165j0j9&sourceid=chrome&es_sm=93&ie=UTF8#q=buku+ayuliana+tentang+blackbox. (17 Agustus 2015)
- Huda, Arif Akbarul. 2009. Aplikasi Android Buat Sendiri. Yogyakarta: CV. Andi Offset
- Juju, Dominikus dan Matamaya Studio. 2009. Teknik Menangkal Kejahatan Internet Untuk Pemula. Jakarta: PT. Elex media Komputindo. (4 Agustus 2015)
- Nugroho, Heru. 2015. Matematika Diskrit dan Implementasinya Dalam Dunia Teknologi. Yogyakarta: CV. Budi Utama.
- 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. <http://ejournal.unsri.ac.id/index.php/jsi/index>. (21 Mei 2015)
- Safaat, Nazruddin. 2014. Pemrograman Aplikasi *Mobile Smartphone* dan Tablet PC Berbasis Android. Bandung Informatik