APLIKASI MOBILE ENKRIPSI SMS MENGGUNAKAN METODE VIGENERE CIPHER DAN BASE64

SKRIPSI

Oleh:
GILANG KURNIAWAN
NIM. 08650009

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2013

APLIKASI MOBILE ENKRIPSI SMS MENGGUNAKAN METODE VIGENERE CIPHER DAN BASE64

SKRIPSI

Diajukan Kepada :
Dekan Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
GILANG KURNIAWAN
NIM. 08650009

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN)
MAULANA MALIK IBRAHIM MALANG
2013

HALAMAN PERSETUJUAN

APLIKASI MOBILE ENKRIPSI SMS MENGGUNAKAN METODE VIGENERE CIPHER DAN BASE64

SKRIPSI

Oleh:
GILANG KURNIAWAN
NIM. 08650009

Telah Diperiksa dan Disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Ririen Kusumawati, M.Kom	**A'la Syauqi, M.Kom**
NIP. 19720309 200501 2 002	**NIP. 19771201 200801 1 007**

Tanggal 28 Maret 2013

Mengetahui,
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

HALAMAN PENGESAHAN

APLIKASI MOBILE ENKRIPSI SMS MENGGUNAKAN METODE VIGENERE CIPHER DAN BASE64

SKRIPSI

Oleh :
GILANG KURNIAWAN
NIM. 08650009

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal, 12 April 2013

Susunan Dewan Penguji:		Tanda Tangan
1. Penguji Utama	: Fatchurrochman, M.Kom NIP. 19700731 200501 1 002	( )
2. Ketua	: Syahiduz Zaman, M.Kom NIP. 19700502 200501 1 005	( )
3. Sekretaris	: Ririen Kusumawati, M.Kom NIP. 19720309 200501 2 002	( )
4. Anggota	: A'la Syauqi, M.Kom NIP. 19771201 200801 1 007	( )

Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

SURAT PERNYATAAN
ORISINALITAS PENELITIAN

Saya yang bertanda tangan di bawah ini :

Nama : Gilang Kurniawan

NIM : 08650009

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : APLIKASI MOBILE ENKRIPSI SMS MENGGUNAKAN METODE VIGENERE CIPHER DAN BASE64

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan, maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 12 April 2013
Penulis

Gilang Kurniawan
NIM. 08650009

MOTTO

فَإِنَّ مَعَ ٱلۡعُسۡرِ يُسۡرًا ۝٥ إِنَّ مَعَ ٱلۡعُسۡرِ يُسۡرًا ۝٦

Karena Sesungguhnya sesudah kesulitan itu ada kemudahan, Sesungguhnya sesudah kesulitan itu ada kemudahan. (QS. Al-Inshiroh:5-6)

لَئِن شَكَرۡتُمۡ لَأَزِيدَنَّكُمۡۖ وَلَئِن كَفَرۡتُمۡ إِنَّ عَذَابِي لَشَدِيدٞ ۝٧

"Sesungguhnya jika kamu bersyukur, pasti Kami akan menambah nikmat kepadamu, dan jika kamu mengingkari (nikmat-Ku), Maka Sesungguhnya azab-Ku sangat pedih". (QS. Ibrahim:7)

PERSEMBAHAN

Dari relung hati yang terdalam

Ku ucapkan beribu syukur dan sujud sedalam kalbu atas segala karunia nikmat-Mu Ya Allah ...

Dengan Ridhomu tlah memberiku kekuatan dalam setiap langkah kehidupanku

Sholawat serta salam kepada Junjungan Rasulullah SAW yang telah memberiku kebanggaan dengan menjadi salah satu dari ummatnya.

Ku persembahkan karya tulis ini untuk

Ayahanda Mohammad Yakut Anas dan Ibunda Supinatun tercinta yang setiap saat selalu bersujud dan bermunajat kepada Allah SWT untuk kebaikan putra-putrinya, serta senantiasa mendukung, memotivasi dan memberiku inspirasi untuk terus berjuang. Adikku tersayang Romantika Mayang Asri dan Dharu Habib Mukhlis yang selalu menjadi motivasiku untuk menjadi teladan yang lebih baik, munajatku agar kalian bisa lebih baik lagi dariku. My special one, I praise Allah for sending you in my life, terima kasih telah menjadi semangatku dan inspirasiku untuk terus berusaha menjadi insan yang lebih baik lagi.

Terima kasih untuk semua

Masyaikh, Asatidz, dan Sahabat-sahabatku yang selalu memberi bimbingan, dukungan dan do'a, serta seluruh teman-teman seperjuangan jurusan Teknik Informatika angkatan 2008, terima kasih kalian telah memberi warna dan pembelajaran dalam hidupku, you're all great people, Jazakumullah ahsanal jaza'

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji dan syukur kehadirat Allah SWT Dzat yang Maha Berilmu di atas mereka yang merasa diri berilmu, serta pencipta Maha Sempurna di atas segala yang dianggap sempurna. Ucapan sholawat serta salam tertuju kepada Rasulullah SAW insan termulia yang telah menghabiskan waktu hanya untuk menuntun umat pengikutnya ke arah keselamatan hidup.

Adapun benar skripsi sulit untuk dapat terwujud manakala penulis tidak dapat dukungan dari berbagai pihak, baik berupa saran maupun kritik, lebih-lebih bantuan yang bersifat moral. Karena itulah sepatutnya diucapkan terimakasih yang tak terhingga, terutama penulis tujukan kepada yang terhormat :

1. Prof. Dr. H. Imam Suprayogo, selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Prof. DR. Sutiman Bambang Sumitro. SU.DSc, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Ririen Kusumawati, M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. Suhartono, M.Kom, selaku Dosen Wali, yang membimbing perencanaan studi selama menuntut ilmu di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

5. Ririen Kusumawati, M.Kom dan A'la Syauqi, M.Kom selaku Dosen Pembimbing, karena atas bimbingan, pengarahan, dan kesabarannya penulisan skripsi ini dapat terselesaikan.
6. Mohammad Yakut Anas, S.Pd dan Supinatun selaku Ayahanda dan Ibunda tercinta yang dengan sepenuh hati memberikan dukungan moral maupun spiritual sehingga penulisan tugas akhir ini dapat terselesaikan.
7. Andriani Susanti, S.Pd yang memberikan motivasi dan semangat.
8. Teman-teman Jurusan Teknik Informatika angkatan 2008 dan teman-teman kampus tercinta di UIN Maulana Malik Ibrahim Malang.
9. Teman-teman kost Ibu Dewi saya ucapkan terima kasih karena telah memberikan motivasi.
10. Teman-teman komunitas uinbuntu, komunitas linux arek malang (kolam) saya ucapkan terima kasih karena telah banyak memberikan ilmu dan pengalaman serta menemani dalam banyak hal.
11. Dan semua pihak yang telah membantu dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Amin Ya Rabbal Alamin.

Wassalamu'alaikum Wr. Wb.

Malang, 12 April 2013

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

ABSTRAK

Gilang Kurniawan. 2013. **Aplikasi Mobile Enkripsi SMS Menggunakan Metode Vigenere Cipher dan Base64.**
Pembimbing : (I) Ririen Kusumawati, M.Kom (II) A'la Syauqi, M.Kom

Kata Kunci : Aplikasi *Mobile*, SMS, Enkripsi, *Vigenere Cipher*, *Base64*.

Perkembangan teknologi saat ini sangat pesat, dengan ditandai munculnya teknologi *smartphone*. Meskipun adanya *smartphone*, SMS (*Short Message Service*) masih tetap menjadi alat komunikasi yang paling popular di masyarakat. Namun SMS memiliki kelemahan dalam sisi privasi pengguna, banyaknya kasus penyadapan yang terjadi ditengah masyarakat sehingga membuat ketidaknyamanan dalam menggunakan fitur SMS. Dengan ada permasalahan tersebut, perlu adanya penanggulangan dengan menggunakan enkripsi pesan sehingga pesan tersebut terjaga privasi dan kemurniannya.

Tujuan dari penelitian ini adalah membuat aplikasi yang dapat mengenkripsi pesan menggunakan metode *Vigenere Cipher* dan *Base64* untuk menjaga privasi pesan. *Vigenere Cipher* merupakan salah satu kriptografi klasik dengan menggunakan teknik subtitusi, dengan menambahkan *Base64* yang digunakan untuk menutupi kelemahan dari metode tersebut.

Dari hasil penelitian ini, terciptanya aplikasi *mobile* yang digunakan untuk mengamankan pesan. Dalam penggunaan enkripsi dan dekripsi, kunci yang digunakan adalah *alphabet*. Jika kunci yang digunakan berkarakter selain itu, proses enkripsi tetap berjalan tetapi hasil dari dekripsi berbeda dengan teks aslinya. Berdasarkan dari pengujian kelayakan produk, 75.2% dari 50 responden menyatakan aplikasi ini layak digunakan dan dipublikasikan ke masyarakat.

ABSTRACT

Gilang Kurniawan. 2013. **Mobile Applications SMS Encryption Using Methods Vigenere Cipher and Base64**.

Promotor : (I) Ririen Kusumawati, M.Kom (II) A'la Syauqi, M.Kom.

Now days, the progress of technology has growth quickly. One of evidence about this case is, there are Smartphone technologies appear. Yet, the Smartphone is very popular now, some people prefer to use Short Message Service (SMS) to communicate each others. Besides SMS is really useful, it also has disadvantages for the user's privacy. There are a lot of tapping case which make the user cannot maximize the SMS features comfortably. In this condition, users need to control the SMS by using encryption, so that, the message's privacy and purity can be guarded.

Therefore, the purpose of this research is about make an application. This application can encrypt the message which uses Vigenere Chiper and Base 64 method. So that, the user can keeps the privacy of SMS. Vigenere Chiper is one of classic cryptography which uses substitution technique. Then, the researcher adds Base 64 method to complete this research result.

Finally, from this research, the researcher creates a mobile application to secure the message. In the use of encryption and decryption, the key used is alphabet. If, the key used has another character, the process of encryption still run well, but the result of decryption will be different with the original text. Based on the test of feasibility product 75, 2 % from 50 respondents state that this application is appropriate to use and publish in society.

Keywords: Mobile Applications, SMS, Encryption, Vigenere Cipher, Base64

BAB I

PENDAHULUAN

1.1 Latar Belakang

Beberapa tahun terakhir ini terjadi perkembangan yang pesat pada teknologi, salah satunya adalah telepon selular (ponsel). Mulai dari ponsel yang hanya bisa digunakan untuk bicara dan mengirim pesan SMS hingga "ponsel cerdas" (*smartphone*) yang memiliki berbagai fungsi seperti *multimedia*, *multiplayer games*, transfer data, *video streaming* dan lain-lain. SMS (*Short Message Service*) merupakan salah media komunikasi yang populer di masyarakat. Selain mudah digunakan, biaya yang digunakan juga murah. Tapi pada sisi lain, SMS memiliki kelemahan yaitu pada sisi keamanan dan kerahasiaan pesan yang dikirim, salah cara memanfaatkan sisi kelemahan SMS dengan menyadap pesan SMS. Sehingga dapat menyebabkan terganggunya privasi pengguna dan ketidaknyamanan dalam menggunakan SMS karena keamanan dan kerahasiaan pesan SMS yang seharusnya jadi privasi pengguna tidak terjaga. Dalam Kamus Besar Bahasa Indonesia, kata sadap berarti mendengarkan (merekam) informasi (rahasia, pembicaraan) orang lain dengan sengaja tanpa sepengetahuan orangnya.

Kasus penyadapan SMS banyak terjadi di Indonesia, diantaranya adalah kasus penyadapan SMS yang terjadi pada wartawan salah satu media surat kabar yang dilakukan oleh salah satu *provider* telekomunikasi di Indonesia.(Hukum online, 2007), dan penyadapan SMS yang terjadi pada salah satu artis indonesia

dan oknum polisi.(TribunNews, 2013) Dari kasus tersebut dapat disimpulkan bahwa penggunaan media komunikasi SMS memiliki kelemahan pada keamanan dan kerahasian pesan. Dengan adanya kasus penyadapan SMS, maka perlu mengamankan pesan SMS sehingga data yang ada pada pesan tersebut terjaga privasinya, dengan cara mengenkripsi data SMS, akan mengamankan data tersebut. Ada beberapa metode yang digunakan untuk enkripsi teks, salah satunya *Vigenere Chiper* merupakan metode klasik yang sering digunakan untuk proses enkripsi pada jaman dulu. Dengan menggabungkan metode *Base64* yang merupakan pengubahan bit pada setiap karakter, maka teks akan mengalami dua kali proses enkripsi, sehingga SMS akan lebih aman.

Metode *Vigenere Cipher* ini dikenal sebagai *polyalphabetic substitution cipher*, karena enkripsi terhadap satu huruf yang sama bisa menghasilkan huruf yang berbeda. *Vigenere Cipher* ini bukan yang terbaik yang dihasilkan oleh *Vigenere*. Dia mengembangkan kriptografi lain yang dikenal sebagai kriptografi *Autokey* yang konon lebih handal dari pada *Vigenere Cipher*, tetapi nama *Vigenere* sudah terlanjur melekat pada kriptografi sebelumnya. Sampai dengan kurun waktu 300 tahun, kedua kriptografi ini dianggap '*unbreakable*', tetapi pada pertengahan abad 19, *Charles Babbage* dan *Friedrich Kasiski* secara terpisah mampu memecahkan cara penyandian ini.(Fahmi, 2006) Kelebihan dari *Vigenere Cipher* adalah meng-enkripsi huruf yang sama bisa menghasilkan huruf yang berbeda. Kekurangan *Vigenere Cipher* adalah dapat dipecahkan dengan menggunakan analisis frekuensi.

Dengan adanya kekurangan dari metode *Vigenere cipher*, maka perlu penambahan untuk menutupi kekurangan tersebut. Dengan penggunaan metode *Base64* dapat menutupi kekurangan tersebut, dikarenakan adanya pengubahan karakter menjadi bentuk biner dari 8 bit menjadi 6 bit. *Base64* berasal dari konten pengkodean MIME (*Multipurpose Internet Mail Extensions*) yang diterbitkan pada RFC (*Request For Comments*) 2045 tahun 1996. Metode *Base64* melibatkan karakter ASCII, mengkonversi setiap karakter teks ke bilangan biner, dan kemudian membagi bilangan biner untuk setiap karakter teks ke 6 bit dan mengkonversinya ke nilai-nilai yang sesuai karakter dalam *Base64*.(Ahmad, 2012) Kelebihan dari *Base64* adalah representasi dari 8 bit menjadi 6 bit, sehingga setiap karakter dapat menjadi lebih panjang dari 3 karakter menghasilkan *ciphertext* 4 karakter. Sedangkan kekurangannya adalah data yang direpresentasi menjadi 33% lebih panjang dari teks asli.(Calhoun, 2011) Dengan penggabungan dua metode *Vigenere Cipher* dan *Base64*, diharapkan bisa menjadikan pesan SMS yang akan dikirim ter-enkripsi dengan baik, data SMS pengguna dapat terlindungi sehingga mendapatkan hak privasinya.

Berkaitan dengan dengan keamanan SMS, sesuai yang telah dinyatakan oleh Allah swt di dalam Al Qur'an-Nya.

ٱلَّذِىٓ أَطۡعَمَهُم مِّن جُوعٖ وَءَامَنَهُم مِّنۡ خَوۡفِۭ ۝٤

"Yang telah memberi makan kepada mereka untuk menghilangkan lapar dan mengamankan mereka dari ketakutan." (Q.S. Al-Quraisy:4)

Pada Surat Al-Quraisy ayat 4 menerangkan secara jelas bahwa Allah lah yang memberikan kita nikmat berupa makanan sehingga hilanglah segala bentuk kelaparannya dan hanya Allah lah yang memberi keamanan dari segala ketakutan. Maha Suci Allah yang maha mengaruniakan dan menjamin keamanan bagi manusia.

Keamanan merupakan karunia Allah yang diberikan kepada manusia yang wajib untuk disyukuri. Allah juga yang telah mengaruniakan manusia akal untuk berfikir. Masalah keamanan merupakan hal yang terpenting dalam kehidupan di dunia ini. Oleh karena itu manusia wajib menggunakan akalnya untuk mempelajari dan menciptakan keamanan itu, sehingga manusia dapat terbebas dari ketakutan.

Seperti yang telah dijabarkan, SMS belum memiliki sistem keamanan untuk menjaga privasi data pengguna, oleh karena itu dibutuhkan pengamanan data SMS dengan cara meng-enkripsi data tersebut. Dengan menggunakan metode *Vigenere Cipher* dan *Base64* diharapkan bisa menjadikan data SMS menjadi aman. Dengan penambahan fitur *inbox*, pesan yang telah diterima dapat dibaca kembali. Sehingga pengguna dapat merasakan nyaman dalam menggunakan fasilitas SMS.

1.2 Rumusan Masalah

Dari latar belakang yang dikemukakan di atas, maka yang menjadi rumusan masalah dalam penelitian ini adalah:

Bagaimana membangun aplikasi *mobile* enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64*?

1.3 Batasan Masalah

Penelitian ini memiliki batasan masalah, yaitu :

a. Input pesan SMS berkarakter *alphanumeric*.
b. Input kunci enkripsi dan dekripsi pada *Vigenere Cipher* berkarakter *alphabet*.
c. Spesifikasi SMS (panjang 1 pesan 160 karakter) disesuaikan dengan standart teknologi *Global System for Mobile* (GSM).
d. Pengiriman pesan tidak dapat melebihi 160 karakter.
e. Bahasa pemrograman yang akan digunakan adalah Java berbasis Android.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menghasilkan aplikasi *mobile* untuk pengiriman dan penerimaan pesan teks dengan fitur pengamanan informasi.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk menjaga privasi dan pengamanan informasi isi dari pesan tersebut bagi pengguna fitur sms.

1.6 Sistematika Pembahasan

Sistematika pembahasan skripsi ini dikelompokkan menjadi 5 Bab sebagai berikut:

BAB I : PENDAHULUAN

Bab ini bersisi tentang latar belakang, batasan masalah, perumusan masalah, tujuan dan manfaat penelitian serta sistematika pembahasan.

BAB II : LANDASAN TEORI

Bab ini membahas tentang teori-teori yang mendukung dan berhubungan dengan judul penulisan skripsi.

BAB III : ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi tentang rancangan pemecahan masalah sesuai dengan judul penulisan dan perancangan sistem.

BAB IV : IMPLEMENTASI HASIL DAN PEMBAHASAN

Bab ini membahas tentang laporan tugas akhir berupa tahapan implementasi dan uji coba dari perancangan sistem serta analisis hasil yaitu implementasi tabel dan pembuatan program aplikasi.

BAB V : PENUTUP

Dalam bab ini berisi kesimpulan dan saran tentang hasil perancangan dan implementasi program.

BAB II

TINJAUAN PUSTAKA

2.1 Keamanan dalam Perspektif Agama Islam

Keamanan adalah suatu hal yang dituntut dalam kehidupan, dimana seluruh makhluk sangat membutuhkannya dalam memenuhi hal-hal yang berkaitan dengan mashlahatan kepentingan manusia, baik yang bersifat keduniaan maupun keagamaan.

Dan tiadalah seorang insan yang hidup di muka bumi ini kecuali mencari sebab-sebab keamanan untuk dirinya dan mencurahkan segenap kemampuan guna menjauhi sebab-sebab ketakutan yang boleh jadi akan mendatangkan ancaman bahaya dalam perjalan hidupnya.

Bagaimana seorang manusia meraih keselamatan badan dan keluasan rizki, maka hal tersebut tidaklah bernilai dan tiada terasa menfaatnya kecuali dengan keamanan dan ketentraman.

Betapapun manusia diberikan sebab-sebab kemajuan dan segala unsur keberhasilan, maka manusia tidak akan mencapai kebahagiannya dan tidak pula dapat menua kehidupan yang indah kecuali dengan tuntunan dan syari'at yang Allah *'Azza wa Jalla*, Sang Pencipta manusia ridhoi untuk mereka.

Dan manusia harus bersyukur dan memuji Allah SWT yang telah menerangkan segala sebab keamanan dalam Islam. Dan manusia harus senantiasa menyanjung-Nya atas segala kemurahan-Nya yang diantaranya adalah

dijadikannya syari'at Islam ini sebagai syari'at yang bertujuan menegakkan keamanan di tengah manusia.

Nabi Ibrahim as. pada awal mula beliau menginjakkan kakinya di kota Makkah, beliau berdoa kepada Rabb-Nya,

وَإِذْ قَالَ إِبْرَٰهِـۧمُ رَبِّ ٱجْعَلْ هَٰذَا بَلَدًا ءَامِنًا وَٱرْزُقْ أَهْلَهُۥ مِنَ ٱلثَّمَرَٰتِ مَنْ ءَامَنَ مِنْهُم بِٱللَّهِ وَٱلْيَوْمِ ٱلْءَاخِرِ ۖ قَالَ وَمَن كَفَرَ فَأُمَتِّعُهُۥ قَلِيلًا ثُمَّ أَضْطَرُّهُۥٓ إِلَىٰ عَذَابِ ٱلنَّارِ ۖ وَبِئْسَ ٱلْمَصِيرُ ﴿١٢٦﴾

"Ya Tuhanku, jadikanlah negeri ini negeri, negeri yang aman sentosa, dan berikanlah rezki berupa buah-buahan kepada penduduknya yang beriman di antara mereka kepada Allah dan hari kemudian." (QS. Al-Baqarah:126)

Setelah beliau merintis kota Makkah, maka beliau dengan perintah Allah meninggalkan keluarganya di negeri baru tersebut untuk sementara waktu. Kemudian beliau kembali lagi ke negeri tersebut dan beliau berdoa kepada-Nya,

وَإِذْ قَالَ إِبْرَٰهِيمُ رَبِّ ٱجْعَلْ هَٰذَا ٱلْبَلَدَ ءَامِنًا وَٱجْنُبْنِى وَبَنِىَّ أَن نَّعْبُدَ ٱلْأَصْنَامَ ﴿٣٥﴾ رَبِّ إِنَّهُنَّ أَضْلَلْنَ كَثِيرًا مِّنَ ٱلنَّاسِ ۖ فَمَن تَبِعَنِى فَإِنَّهُۥ مِنِّى ۖ وَمَنْ عَصَانِى فَإِنَّكَ غَفُورٌ رَّحِيمٌ ﴿٣٦﴾

"Ya Rabb-ku, jadikanlah negeri ini (Makkah), negeri yang aman, dan jauhkanlah aku beserta anak cucuku daripada menyembah berhala-berhala. Ya Rabb-ku, sesungguhnya berhala-berhala itu telah menyesatkan kebanyakan daripada manusia, maka barangsiapa yang mengikutiku, maka ssesungguhnya orang itu termasuk golonganku, dan barang siapa yang mendurhakai aku, maka sesungguhnya Engkau, Maha Pengampun lagi Maha Penyayang". (QS. Ibrahim : 35-36)

Pada ayat-ayat ini Allah SWT, memerintahkan kepada Nabi Muhammad agar menyampaikan kepada umatnya kisah di waktu Nabi Ibrahim berdoa kepada

Tuhannya agar doa itu menjadi iktibar dan pelajaran bagi orang Arab waktu itu, karena Ibrahim a.s. itu adalah cikal bakal dan asal keturunan mereka. Doa itu ialah: *"Ya Tuhan kami, jadikanlah negeri Makkah ini, negeri yang aman dan tenteram, negeri yang sentosa, terpelihara dari peperangan dan serangan musuh."* Doa Nabi Ibrahim itu dikabulkan Tuhan, dan Dia telah menjadikan negeri Makkah dan sekitarnya menjadi tanah dan tempat yang aman bagi orang-orang yang berada di sana. Di negeri itu dilarang menumpahkan darah, menganiaya orang, membunuh binatang dan menebang tumbuh-tumbuhan yang berada di sana. (Yayasan Indonesia Membaca, 2009)

Dalam Surat Ibrahim ayat 35-36, Nabi Ibrahim as. berdoa dengan memohon keamanan untuk kota Makkah. Hal tersebut karena Nabi Ibrahim as. sangat mengetahui bahwa keamanan adalah lambang kebahagian masyarakat, bangsa dan Negara, dan dengan keamanan segala kemashlahatan dan kebaikan yang dibutuhkan manusia akan tercapai. (Dzulqarnain, 2009)

Dan Allah SWT mengingatkan nikmat keamanan kepada penduduk tanah haram dan kepada seluruh makhluk agar mereka senantiasa mengingat nikmat tersebut dan bersyukur kepada Allah karenanya dan beribadah kepada-Nya di bawah teduhannya,

أَوَلَمْ يَرَوْا۟ أَنَّا جَعَلْنَا حَرَمًا ءَامِنًا وَيُتَخَطَّفُ ٱلنَّاسُ مِنْ حَوْلِهِمْ ۚ أَفَبِٱلْبَٰطِلِ يُؤْمِنُونَ وَبِنِعْمَةِ ٱللَّهِ يَكْفُرُونَ ﴿٦٧﴾

"Dan apakah mereka tidak memperhatikan, bahwa sesungguhnya Kami telah menjadikan (negeri mereka) tanah suci yang aman, sedang manusia disekitarnya saling rampok-merampok. Maka mengapa (sesudah nyata kebenaran) mereka masih percaya kepada yang bathil dan ingkar kepada nikmat Allah?" (QS. Al-'Ankabuut : 67)

Pada Surat Al-'Ankabuut ayat 67 menjelaskan bahwa mengingatkan orang-orang musyrik Makkah akan nikmat yang dilimpahkan kepada mereka. Mereka diistimewakan Allah dari penduduk negeri-negeri yang berada di sekitar mereka dengan menjadikan kota Makkah sebagai negeri yang aman dan tenteram diharamkan berperang di sana, dan Allah menjaga negeri itu dari musuh-musuhnya yang hendak menghancurkan dan menguasainya, seperti yang pernah terjadi pada tahun kelahiran Nabi Muhammad saw. Pada waktu itu tentara Abrahah yang mengendarai gajah dihancurkan Allah sebelum mereka sempat menjamah Kabah.

Dengan adanya keamanan maka manusia akan merasakan kenikmatan yang diberikan oleh Allah SWT. Sebagai contoh pada kehidupan sehari-hari pada kehidupan manusia. Misalkan ada seorang mahasiswa memiliki laptop yang didalamnya terdapat data penting, seperti data-data skripsi. seseorang pasti berhati-hati dalam meletakkan barang tersebut, bukan berarti tidak ingin laptopnya hilang, melainkan akan pentingnya datanya. Jika manusia merasa tenang dan yakin kalau barang tersebut diletakkan ditempat yang aman, maka perasaan yang dirasakan adalah tenang, melakukan kegiatan pasti lebih fokus. Dan jika sebaliknya, maka perasaan tidak tenang akan terus menyelimuti. Sehingga tidak bisa merasakan nikmatnya dari keamanan dan ketenangan yang diberikan Allah SWT kepada manusia.

Begitu pula penggunaan keamanan pada SMS, yang memberikan kenyamanan pengguna dalam menggunakan SMS. Pada Aplikasi Enkripsi SMS, pengguna dapat mengenkripsi guna mengamankan pesan yang akan dikirim,

sehingga jika terjadi panyadapan SMS maka pesan tersebut akan terjaga privasi dan informasi yang terkandung didalamnya.

2.2 SMS (*Short Message Service*)

Short Message Service (SMS) adalah transmisi pesan teks yang singkat untuk dan dari telepon selular, mesin fax, dan / atau alamat IP. Sebuah pesan haruslah tidak melebihi 160 karakter alfanumerik dan tidak mengandung gambar atau grafis. SMS adalah sistem komunikasi yang relatif sederhana yang disediakan oleh jaringan telepon seluler. Pesan SMS yang didukung oleh GSM, TDMA dan CDMA berbasis jaringan telepon selular yang saat ini digunakan. Meskipun layanan berbasis SMS sudah ada selama bertahun-tahun, penetrasi telepon seluler yang dilakukan baru-baru ini, dan penerapan skala besar dari layanan yang ada oleh pengguna, telah membuat layanan berbasis SMS lebih menarik bagi penyedia layanan.

Short Message Service (SMS), sebagaimana didefinisikan dalam standar digital ponsel GSM yang populer di Eropa, Timur Tengah, Asia, Afrika dan beberapa bagian Amerika Utara, memiliki beberapa fitur unik:

a. Satu SMS bisa mencapai panjang 160 karakter per teks. 160 karakter dapat terdiri dari kata-kata atau angka atau kombinasi alfanumerik, juga mendukung SMS yang tidak berbasis teks (misalnya, dalam format biner).
b. SMS adalah *store* dan *forward service*, dengan kata lain, SMS tidak dikirim langsung dari pengirim ke penerima, tetapi selalu melalui SMS *Center* sebagai gantinya. Setiap jaringan telepon seluler yang mendukung SMS,

memiliki satu atau lebih pusat pesan untuk menangani dan mengelola pesan singkat.

c. SMS memiliki fitur konfirmasi pengiriman pesan. Tidak seperti *paging*, pengguna tidak cukup mengirim SMS dan berharap bahwa pesan yang dikirim itu tersampaikan ke tujuan. Sebaliknya pengirim pesan singkat dapat menerima pesan kembali untuk memberitahukan kepada pengguna apakah SMS sudah dikirim atau tidak.

d. SMS dapat dikirim dan diterima secara bersamaan dengan penerimaan GSM *voice*, data dan panggilan Fax. Hal ini dikarenakan suara, data dan panggilan faks menggunakan saluran radio khusus, sedangkan SMS diatas saluran radio menggunakan jalur sinyal. Dengan demikian, pengguna SMS jarang sekali mendapatkan jaringan yang sibuk dikarenakan mereka menggunakan SMS dan GSM *voice* secara bersamaan.

e. Tersedia beberapa cara untuk mengirim SMS. Rangkaian SMS (pengiriman SMS secara bersamaan) dan kompresi SMS (mendapatkan lebih dari 160 karakter informasi dalam satu pesan) telah ditetapkan dan dimasukkan dalam SMS standar GSM.(Rathke, 2012)

2.3 Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Prinsip-prinsip yang mendasari kriptografi yakni:

a. *Confidelity* (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
b. *Data integrity* (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
c. *Authentication* (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
d. *Non-repudiation* (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya). (Menezes, 1996)

2.3.1 Algoritma Kriptografi

Definisi terminologi algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara matematis. Sedangkan kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut.

Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu:

a. Enkripsi, merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *Plaintext*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *Cipher* atau kode.
b. Dekripsi, merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (tesk-asli), disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma untuk enkripsi
c. Kunci, yang dimaksud adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

Keamanan dari algoritma kriptografi tergantung pada bagaimana algoritma itu bekerja. Oleh sebab itu algoritma semacam ini disebut dengan algoritma terbatas. Algoritma terbatas merupakan algoritma yang dipakai sekelompok orang untuk merahasiakan pesan yang mereka kirim. Jika salah satu dari anggota itu keluar dari kelompoknya maka algoritma yang dipakai diganti dengan yang baru. Jika tidak maka hal itu bisa menjadi masalah di kemudian hari.

Keamanan dari kriptografi modern didapat dengan merahasiakan kunci yang dimiliki dari orang lain, tanpa harus merahasiakan algoritma itu sendiri. Kunci memiliki fungsi yang sama dengan *password*. Jika keseluruhan dari keamanan algoritma tergantung pada kunci yang dipakai maka algoritma ini bisa dipublikasikan dan dianalisis oleh orang lain. Jika algoritma yang telah dipublikasikan bisa dipecahkan

dalam waktu yang singkat oleh orang lain maka algoritma tersebut tidaklah aman untuk digunakan.

2.3.2 Macam-macam Algoritma Kriptografi

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya:

a. Algoritma Simetri

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi maupun dekripsi. Algoritma ini sudah ada sejak lebih dari 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsikan pesan yang terkirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan. Algoritma yang memakai kunci simetri di antaranya adalah:

1. *Data Encryption Standard* (DES),
2. RC2, RC4, RC5, RC 6,
3. *International Data Encryption Algorithm* (IDEA),
4. *Advanced Encryption Standard* (AES),
5. *On Time Pad* (OTP),
6. A5, dan lain sebagainya.

b. Algoritma Asimetri

Algoritma asimetri sering juga disebut dengan algoritma kunci public, dengan arti kata kunci yang digunakan melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian, yaitu :

1. Kunci umum (*public key*), kunci yang boleh semua orang tahu (dipublikasikan).
2. Kunci rahasia (*private key*), kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).

Kunci-kunci tersebut berhubungan satu sama lain. Dengan kunci public orang dapat mengenkripsi pesan tetapi tidak bisa mendekripsikannya. Hanya orang yang memiliki kunci rahasia yang dapat mendekripsikan pesan tersebut. Algoritma asimetri bisa mengirimkan pesan dengan lebih aman daripada algoritma simetri.

Algoritma yang memakai kunci public di antaranya adalah :

1. *Digital Signature Algorithm* (DSA),
2. *RSA*,
3. *Diffle-Hellman* (DH),
4. *Elliptic Curve Cryptography* (ECC),
5. Kriptografi Quantum, dan lain sebagainya.

c. Fungsi Hash

Fungsi Hash sering disebut dengan fungsi satu arah (*one-way function*), *message digest*, *fingerprint*, fungsi kompresi dan *message authentication code* (MAC), merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. Fungsi Hash biasanya diperlukan bila ingin membuat sidik jari dari suatu pesan. Sidik jari pada pesan

merupakan suatu tanda bahwa pesan tersebut benar-benar berasal dari orang-orang yang diinginkan.

2.3.3 Kriptografi Klasik

Kriptografi klasik merupakan suatu algoritma yang menggunakan satu kunci untuk mengamankan data. Teknik ini sudah digunakan beberapa abad yang lalu. Dua teknik dasar yang biasa digunakan pada algoritma jenis ini adalah sebagai berikut :

a. Teknik subtitusi, penggantian setiap karakter teks-asli dengan karakter lain.
b. Teknik transposisi (permutasi), dilakukan dengan menggunakan permutasi karakter. (Ariyus, 2008)

Salah satu teknik enkripsi menggunakan kunci simetri adalah teknik subtitusi, yaitu mengganti setiap karakter *Plaintext* dengan karakter lain. Terdapat empat cara dalam menggunakan teknik subtitusi, yaitu :

a. *Monoalphabet*, dimana setiap karakter *Ciphertext* mengganti satu macam karakter *Plaintext* tertentu.
b. *Polialphabet*, dimana setiap karakter *Ciphertext* mengganti lebih dari satu macam karakter *Plaintext*.
c. *Monograf/unilateral*, dimana satu enkripsi dilakukan terhadap satu karakter *Plaintext*.
d. Poligraf/multilateral, dimana satu enkripsi dilakukan terhadap lebih dari satu karakter *Plaintext*. (Menezes, 1996)

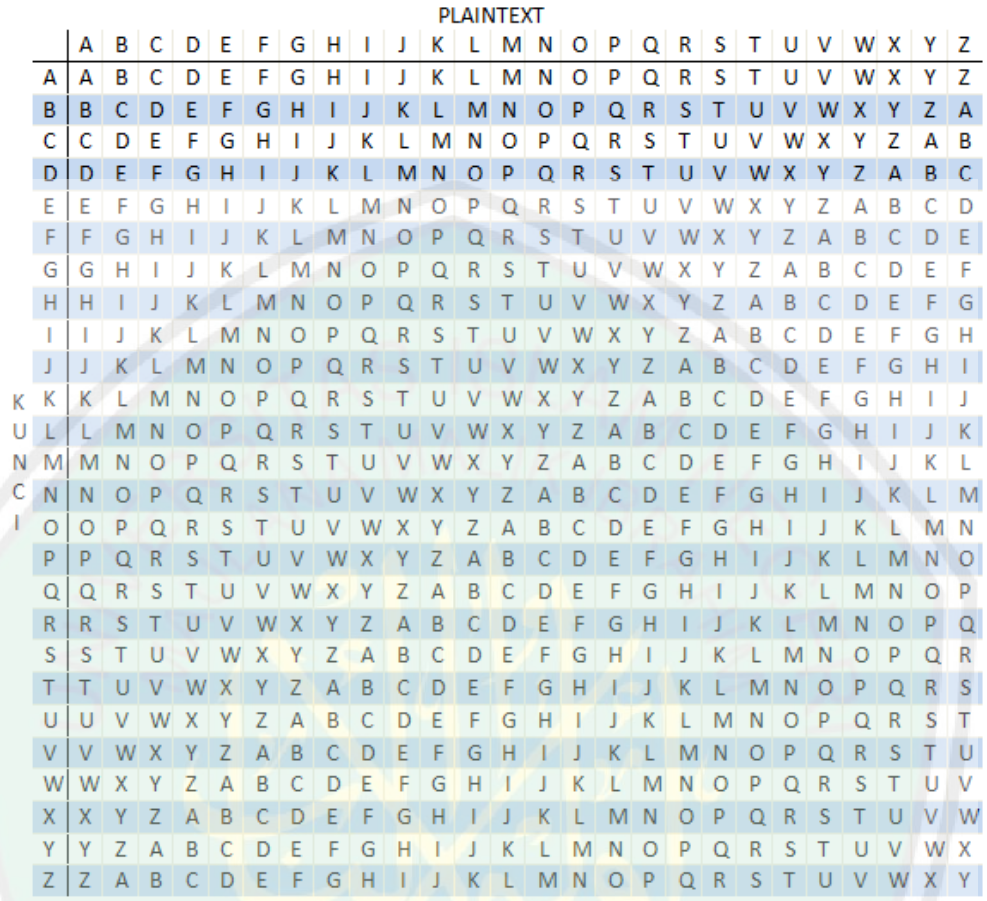
2.3.4 Kriptografi Modern

Kriptografi modern merupakan suatu algoritma yang digunakan pada saat sekarang ini, yang mana kriptografi modern mempunyai kerumitan yang sangat komplek, karena dalam pengoperasiannya menggunakan komputer. (Ariyus, 2006)

2.4 *Vigenere Cipher*

Kode *Vigenere* termasuk kode abad-majemuk (*polyalphabetic substitution Cipher*). Dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis, *Blaise de Vigenere* pada abad 16, tahun 1586. Sebenarnya *Giovan Batista Belaso* telah menggambarkannya untuk pertama kali pada tahun 1553 seperti ditulis di dalam buku *La Cifra Del Sig*. Algoritma ini baru dikenal luas 200 tahun kemudian dan dinamakan kode *Vigenere*.

Pada teknik subtitusi *Vigenere* setiap teks-kode bisa memiliki banyak kemungkinan teks-asli. Ide dasarnya adalah dengan menggunakan kode kaisar, tetapi jumlah pergeseran hurufnya berbeda-beda untuk setiap periode beberapa huruf tertentu. untuk mengenkripsi pesan dengan kode *Vigenere* digunakan *tabula recta* (disebut juga bujursangkar *Vigenere*) seperti pada gambar 2.1.



Gambar 2.1 *Tabula Recta.*

Tabula Recta digunakan untuk memperoleh teks-kode dengan menggunakan kunci yang sudah ditentukan. Jika panjang kunci lebih pendek daripada panjang teks-asli maka pengunaan kunci diulang.

Contoh kode *Vigenere* adalah sebagai berikut :

Plaintext : "KEAMANAN DATA MENGGUNAKAN CIPHER VIGENERE"

Key : "KRIPTOGRAFI"

Ciphertext : "UVIBTBGE DFBK WVVVZCTRKFV FZOTGSXHE HQZYMP". (Ariyus, 2008)

Secara matematis *Vigenere* dilakukan dengan penjumlahan index *plaintext* dan index *key*, A - Z diubah ke dalam bentuk angka dari 0 - 25 yang kemudian di modulo 26.(Kester, 2012) Cara ini hampir sama dengan kode geser atau *Caesar Cipher*. Rumus enkripsi *Vigenere Cipher* dapat ditulis dengan, (*Plaintext+Key*)*Modulo*26, dan rumus dekripsi *Vigenere Cipher* dapat ditulis dengan, (*Ciphertext-Key*)*Modulo*26.

Tabel 2.1. *Caesar Cipher*

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Misalkan teks-aslinya "*This Cryptosystem is Not Secure*" dan kunci yang digunakan adalah "CIPHER" di ubah ke bentuk angka dalam tabel *Caesar Cipher* manjadi (2, 8, 15, 7, 4, 17).

Plaintext	T	H	I	S	C	R	Y	P	T	O	S	Y	S	T
Index Plaintext	19	7	8	18	2	17	24	15	19	14	18	24	18	19
Key	C	I	P	H	E	R	C	I	P	H	E	R	C	I
Index Key	2	8	15	7	4	17	2	8	15	7	4	17	2	8
(*Plaintext* + *key*) mod 26	21	15	23	25	6	8	0	23	8	21	22	15	20	1
Ciphertext	V	P	X	Z	G	I	A	X	I	V	W	P	U	B

Plaintext	E	M	I	S	N	O	T	S	E	C	U	R	E
Index Plaintext	4	12	8	18	13	14	19	18	4	2	20	17	4
Key	P	H	E	R	C	I	P	H	E	R	C	I	P

Index Key	15	7	4	17	2	8	15	7	4	17	2	8	15
(*Plaintext* + *key*) mod 26	19	19	12	9	15	22	8	25	8	19	22	25	19
Ciphertext	T	T	M	J	P	W	I	Z	I	T	W	Z	T

Plaintext : "*This cryptosystem is not secure*"

Kunci : CIPHER

Ciphertext : "VPXZGIAXIVWPUBTTMJPWIZITWZT"

2.5 *Base64*

Base64 adalah istilah umum untuk sejumlah skema pengkodean serupa yang mengkodekan data biner dan menerjemahkannya ke dalam representasi basis 64. Istilah *Base64* berasal dari konten pengkodean MIME (*Multipurpose Internet Mail Extensions*) pada RFC (*Request For Comments*) 2045 tahun 1996.

Base64 biasanya digunakan ketika ada kebutuhan untuk menyandikan data biner yang perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani data tekstual. Hal ini untuk memastikan bahwa data tetap utuh tanpa perubahan selama pengiriman. *Base64* digunakan umum dalam beberapa aplikasi termasuk email melalui MIME, dan penyimpanan data yang kompleks dalam XML (*Extensible Markup Language*). Transformasi *Base64* banyak digunakan di dunia Internet sebagai media data format untuk mengirimkan data. Dikarenakan hasil dari transformasi *Base64* berupa *Plaintext*, maka nilai ini akan jauh lebih mudah dikirim, dibandingkan format data berupa binary.

Tabel 2.2. *Encoding ASCII.*

Char	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
DEC	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Char	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
DEC	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Char	SP	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
DEC	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Char	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
DEC	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Char	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
DEC	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Char	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
DEC	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Char	`	a	b	c	D	e	f	g	h	I	j	k	l	m	n	o
DEC	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
Char	p	q	r	s	T	u	v	w	x	Y	z	{	|	}	~	DEL
DEC	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127

Tabel 2.3. *Encoding Base64*

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
pad	=						

2.5.1 Penggunaan *Base64*

Berikut ini adalah contoh penggunaan dari MIME *Base64* dalam melakukan *encoding* karakter.

Kutipan dari Albert Einstein :

"Imajinasi lebih penting daripada pengetahuan".

Hasil Transformasi :

"SW1hamluYXNpIGxlYmloIHBlbnRpbmcgZGFyaXBhZGEgcGVuZ2V0YWh1YW4="

Kutipan dari Thomas Alfa Edison:

"Banyak orang yang sebenarnya sudah sangat dekat dengan sukses tapi sayangnya, mereka kemudian menyerah"

Hasil Transformasi:

"QmFueWFrIG9yYW5nIHlhbmcgc2ViZW5hcm55YSBzdWRhaCBzYW5nYXQgZGVrYXQgZGVuZ2FuIHN1a3NlcyB0YXBpIHNheWFuZ255YSwgbWVyZWthIGtlbXVkaWFuIG1lbnllcmFo"

Pada hasil encoding kata "Rio" diganti menjadi "Umlv"/ Pada tabel ASCII huruf R, i, o disimpan sebagai 82, 105, 111 atau dengan kata lain 01010010, 01101001, 01101111 pada bilangan berbasis 2. Apabila ketiga *byte* tersebut digabungkan, maka akan dihasilkan 24 bit *buffer* yaitu 010100100110100101101111. Angka tersebut harus dikonversi sehingga berbasis 64, caranya dengan membagi 24 bit tersebut dengan 6. Maka dihasilkan 4 bagian dengan masing-masing 6 bit. Kemudian masing-masing bagian tersebut dikonversi ke nilai yang ada di *Base64*.

Tabel 2.4 *Contoh proses encoding Base64*

Huruf	R								i								o							
ASCII	82								105								111							
Bit	0	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	1	0	1	1	1	1
Index	20								38								37							
Base64 Encoded	U								m								l							
	v																							

Proses *padding* akan dilakukan apabila sekelompok karakter yang dimiliki tidak bernilai 3 Byte (24 bit). *Padding* dilakukan dengan menambahkan karakter '='. Contoh penggunaan *padding* dapat dilihat pada tabel berikut.

Tabel 2.5 *Contoh penggunaan padding*

Huruf	i								s															
ASCII	105								115															
Bit	0	1	1	0	1	0	0	1	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0
Index	26								23								12							
Base64 Encoded	a								X								M							
	=																							

Seperti pada tabel 2, apabila terdapat *single byte* maka jumlah padding yang ditambahkan adalah 2 *Byte* yang bernilai 0. Sehingga memenuhi aturan 3 *Byte* (24 bit), Sedangkan jumlah *byte padding* yang ditambahkan adalah 1 *Byte* karena sebelumnya telah memiliki 2 *Byte*. (Kurniawan, 2008)

2.6 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.

Awalnya, Google Inc membeli Android Inc. yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung penuh dari Google atau *Google mail Services* (GMS) dan kedua adalah benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produksi perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikkan bug dan penambahan fitur baru.

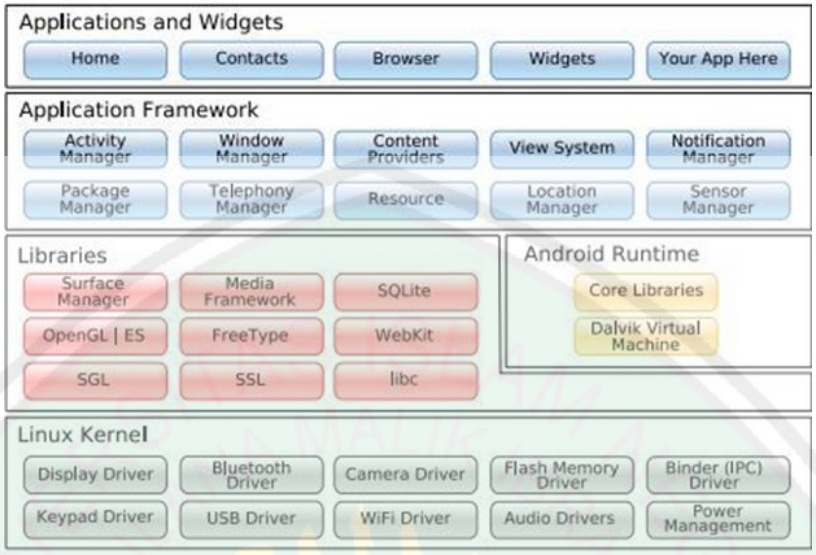
Pada masa saat ini sebagian besar *vendor-vendor smartphone* sudah memproduksi *smartphone* berbasis Android, *vendor-vendor* itu diantaranya lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi,

Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi android. Hal ini, karena Android adalah sistem operasi yang open source sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, Aplikasi dan *Tool* pengembangan, *Market* aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari jumlah *device* yang ada di dunia. (Safaat H, 2012)

2.6.1 Arsitektur Android

Secara garis besar Arsitektur Android dapat digambarkan sebagai berikut:



Gambar 2.2 *Diagram komponen utama android.*(Syafaat, 2010)

Pada gambar 2.2 diperlihatkan sistem operasi Android memiliki 4 lapisan (*layer*) yang merupakan komponen sistem Android. Google mengibaratkan Android sebagai sebuah tumpukan *software*. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan-lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas.

a. *Applications* dan *Widgets*

Applications dan *Widgets* adalah *layer* dimana pengguna berhubungan dengan aplikasi. Di mana biasanya pengguna men-*download* aplikasi kemudian menginstalasi dan menjalankan aplikasi tersebut. Di *layer* ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman Java.

b. *Applications Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberikan kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya pengguna dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*).

Applications Frameworks adalah *layer* di mana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android. Karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

1) *Views*

2) *Content Provider*

3) *Resource Manager*

4) *Notification Manager*

5) *Activity Manager*

c. *Libraries*

Libraries adalah *layer* di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya berjalan di atas kernel, *layer* ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

1) *Libraries* media untuk pemutaran media audio dan video.
2) *Libraries* untuk manajemen tampilan.
3) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4) *Libraries* SQLite untuk dukungan *database*.
5) *Libraries* SSL dan *Webkit* terintegrasi dengan *web browser* dan *security*.
6) *Libraries* LiveWebcore mencakup modern *web browser* dengan *engine embedded web view*.
7) *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's.

d. Android *Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android *Run Time* dibagi menjadi dua bagian yaitu :

1) *Core Libraries*, aplikasi Android dibangun dalam bahasa Java, sementara DVM sebagai virtual mesinnya buka *Virtual Mechine* Java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *Core Libraries*.

2) *Dalvik Virtual Machine*, Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux *kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

e. Linux *Kernel*

Linux *Kernel* adalah *layer* di mana inti dari *operating system* dari Android itu berada. Berisi *file-file system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi Android. Linux *kernel* yan digunakan android adalah linux *kernel release* 2.6. (Safaat H, 2012)

2.6.2 Aplikasi Fundamental

Aplikasi Android ditulis dalam bahasa pemrograman Java. Kode Java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi, di mana prosesnya di-*package* oleh *tools* yang dinamakan "*apt tools*" ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk. *File* apk itulah yang disebut dengan aplikasi, dan nantinya dapat di install di perangkat *mobile*.

Ada empat jenis komponen pada aplikasi Android yaitu :

1) *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga penguna dapat melakukan interaksi. Sebuah aplikasi Android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi yang bertindak sebagai *user*

interface (UI) saat aplikasi diperlihatkan kepada pengguna. Untuk pindah dari satu *activity* ke *activity* lain, pengembang dapat menggunakan dengan satu *even*, misalnya click tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *window activity* dinyatakan dengan *method* Activity.setContentView(). ContentView adalah objek yang berada pada *root* hirarki.

2) *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *device* harus berada dalam kelas induknya. Misalnya, *media player* sedang memutar lagu dari list yang ada, aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu misalnya, atau menulis SMS dengan menjalankan *media player* juga. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* ini dijalankan pada *thread* utama dari proses aplikasi.

3) *Broadcast Receiver*

Broadcast receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh broadcast seperti zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh camera, atau mengubah referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi broadcast misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan.

Broadcast receiver tidak memiliki user interface (UI), tetapi memiliki sebuah activity untuk merespon informasi yang mereka terima, atau mungkin menggunakan Notification Manager untuk memberitahukan kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

4) *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti *database* SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (*Map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi *content provider*. (Safaat H, 2012)

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis *Input*

Input adalah data-data yang dimasukkan yang nantinya akan diproses sehingga bisa menghasilkan *output* sesuai dengan yang diharapkan. Pada sistem aplikasi enkripsi SMS ini terbagi menjadi dua macam *input*, yaitu *input* yang digunakan oleh *handphone* pengirim dan *input* yang digunakan oleh *handphone* penerima. *Input* yang digunakan oleh pengirim adalah nomor tujuan dan teks SMS yang akan dikirimkan. Teks SMS ini nantinya akan menjalani proses *encoding*. Sedangkan *input* yang digunakan oleh *handphone* penerima adalah teks SMS yang telang di-*encoding* menjadi kode hasil dari *Vigenere Cipher* dan *Base64*, yang dikirimkan oleh *handphone* pengirim. *Input* ini nantinya akan menjalani proses *decoding*.

3.2 Analisis *Output*

Output adalah hasil pengolahan data-data yang telah dimasukkan, sehingga menghasilkan keluaran yang bisa digunakan. *Output* yang dihasilkan oleh sistem aplikasi enkripsi SMS, yaitu *output* yang dihasilkan oleh *handphone* pengirim. *Output* yang dihasilkan oleh *handphone* pengirim berupa kode yang sudah di-*encode* yang merupakan hasil *encoding* dari teks SMS yang akan dikirim. Kode inilah yang kemudian dikirimkan ke nomor tujuan.

3.3 Analisis Kebutuhan

Analisis kebutuhan merupakan analisis terhadap komponen-komponen yang digunakan untuk pembuatan sistem aplikasi enkripsi SMS. Dalam hal ini, komponen yang dibutuhkan terbagi menjadi dua macam, komponen *software* dan *hardware*.

3.3.1 *Software*

Software adalah program atau aplikasi komputer lain yang dibutuhkan untuk membangun sebuah sistem. *Software* yang dibutuhkan untuk proses pembuatan aplikasi enkripsi SMS antara lain:

a. Eclipse

   Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk pengembangan perangkat lunak dari berbagai bahasa pemograman, dan dapat dijalankan di semua *platform*.

b. Java SE Runtime Environment (JRE)

   Untuk menjalankan program java, maka file berekstensi .java harus dikompilasi menjadi *byte code*. JRE berfungsi untuk mengeksekusi file *byte* yang memungkin pemakai untuk menjalankan program java.

c. Android Software Development Kit (SDK)

   Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

d. Android Development Tools (ADT)

Android Development Tools adalah plugin yang didesain untuk IDE Eclipse yang menjadikan penghubung dengan Android SDK. (Nazruddin Safaat H, 2012).

3.3.2 *Hardware*

Hardware adalah perangkat keras atau *device* yang digunakan untuk menunjang dalam pembuatan sebuah sistem. Dalam pembuatan aplikasi enkripsi SMS, *hardware* yang dibutuhkan antara lain:

a. Komputer

Komputer digunakan untuk membangun sistem aplikasi enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64*.

b. *Smartphone*

Smartphone digunakan untuk mengaplikasikan program yang telah dibuat untuk mengetahui apakah program tersebut telah memenuhi spesifikasi yang diingikan. Program yang berjalan di komputer dan emulator, belum tentu dapat berjalan dengan baik di *mobile devices* yang sesungguhnya. Semuanya tergantung spesifikasi dari *smartphone* yang dipakai.

3.4 Tahap-Tahap Pembuatan Aplikasi

Dalam pembuatan sebuah sistem aplikasi akan melewati beberapa proses, mulai dari pengumpulan data hingga uji coba program. Berikut adalah tahapan-tahapan yang dilakukan untuk membuat aplikasi enkripsi SMS:

a. Pengumpulan Data

Merupakan tahapan untuk mengumpulkan informasi yang dibutuhkan dalam proses pembuatan aplikasi enkripsi SMS. Dalam pengumpulan data, penulis menggunakan metode studi pustaka, yaitu mencari sumber dari buku dan internet.

b. Analisis Data dan Sistem

Merupakan tahapan untuk mengidentifikasi semua permasalahan yang terdapat didalam sistem berdasarkan data-data yang telah dikumpulkan.

c. Perancangan Sistem

Merupakan tahapan mendesain dan merancang sistem, yang meliputi bagaimana proses jalannya sistem, bagaimana tampilan *interface* sistem, dan fitur apa saja yang akan ditampilkan di dalam sistem.

d. Pembuatan Program

Setelah data terkumpul dan sistem telah dirancang, maka selanjutnya adalah memulai pembuatan aplikasi dengan menggunakan bahasa pemrograman Java berbasis Android.

e. Uji Coba

Tahapan uji coba dilakukan untuk mengetahui apakah aplikasi yang dibuat telah sesuai dengan yang diharapkan atau belum.

f. Revisi Program

Jika dalam proses uji coba terdapat *error* ataupun masih terdapat fitur yang kurang di dalam aplikasi yang dibuat, maka dilakukan revisi program untuk menyempurnakan kembali aplikasi tersebut.

g. Pembuatan Laporan

Tahap akhir adalah pembuatan laporan yang membahas mengenai keseluruhan proses pembuatan aplikasi enkripsi SMS.

3.5 Spesifikasi Aplikasi

Aplikasi Enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64* ini memiliki kemampuan sebagai berikut :

a. Enkripsi SMS

Sebelum SMS dikirimkan ke nomor tujuan, maka teks pesan tersebut akan di-enkripsi dengan menggunakan metode *Vigenere Cipher* dan *Base64* untuk mengamankan teks asli dari pesan tersebut.

b. Mengirim SMS

Setelah teks SMS di-enkripsi menjadi kode hasil dari *Vigenere Cipher* dan *Base64*, maka pesan dapat dikirimkan ke nomor tujuan.

c. Dekripsi SMS

Dekripsi SMS merupakan pengolahan atau pengubahan kode hasil yang telah di-enkripsi menjadi bentuk teks asli dengan menggunakan metode *Base64* dan *Vigenere Cipher*.

d. Mengatur Kunci

Semua proses enkripsi dan dekripsi *Vigenere Cipher* membutuhkan kunci yang digunakan untuk mengolah *plaintext*.

e. Menyimpan pesan masuk

Semua pesan yang masuk akan disimpan dalam *database*, baik SMS yang ter-enkripsi maupun SMS yang tidak ter-enkripsi.

3.6 Spesifikasi Pengguna

Aplikasi enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64* ini dapat digunakan oleh semua orang yang memiliki *smartphone* Android. Dengan aplikasi ini maka pengguna dapat merasakan nyaman dalam menggunakan fitur SMS.

3.7 Dekripsi Sistem

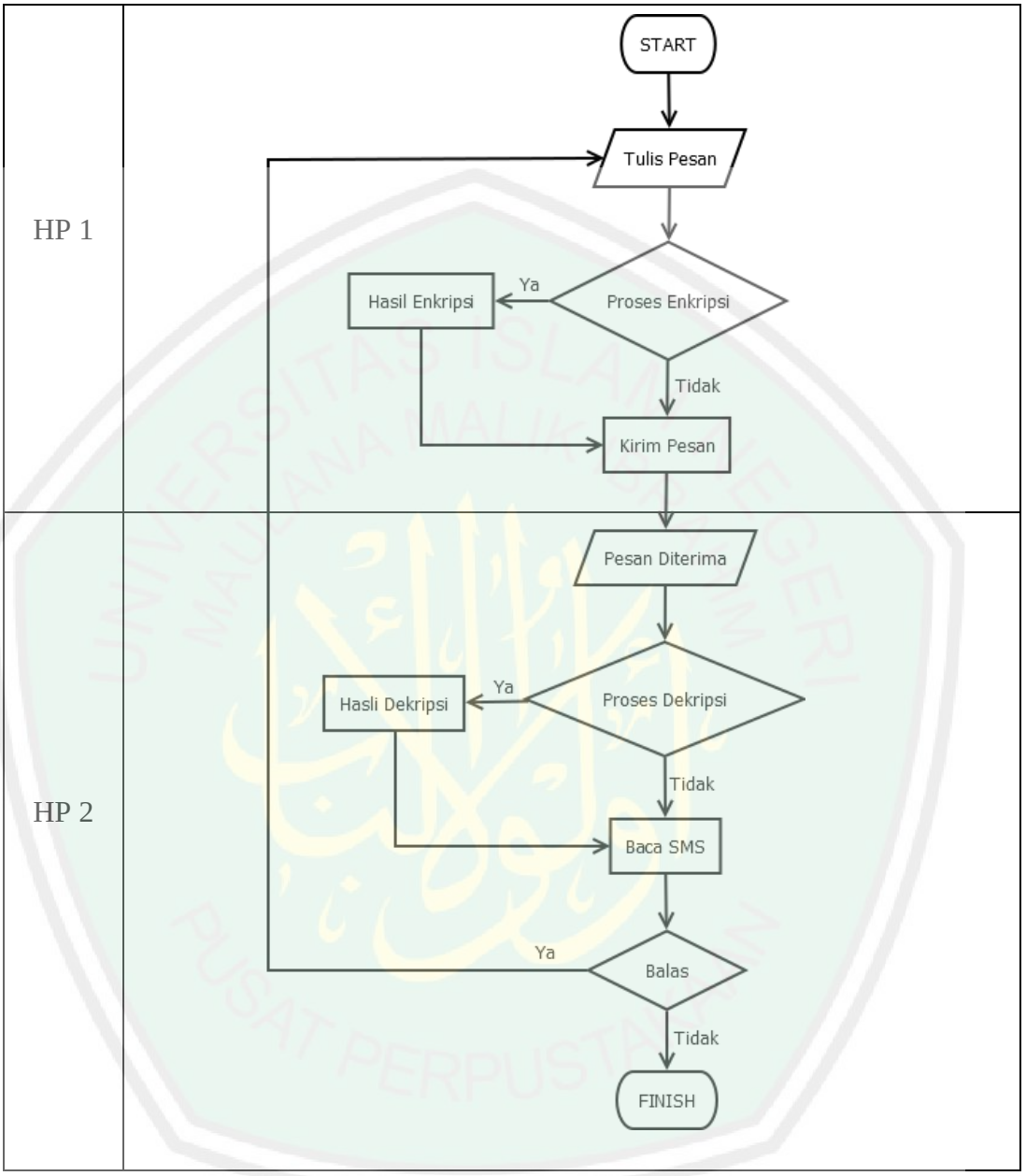
Aplikasi enkripsi SMS dengan menggunakan metode *Vigenere Cipher* dan *Base64* ini memiliki tujuan utama untuk memberikan keamanan dalam menggunakan SMS, sehingga pengguna tidak perlu khawatir dengan adanya aplikasi penyadap SMS.

Dalam melakukan enkripsi terhadap teks SMS, dilakukan dengan menggunakan metode *Vigenere Cipher* dan *Base64*. Dua metode tersebut dipakai karena memiliki tingkat enkripsi yang cukup baik dibanding dengan metode enkripsi lainya. Algoritma *Vigenere Cipher* menggunakan teknik subtitusi dimana setiap huruf akan diubah dalam bentuk angka kemudian ditambahkan dengan kunci yang sudah dibentuk ke angka dan menghasilkan sebuah kode *Vigenere Cipher* kemudian di-modulo 26 sesuai dengan jumlah huruf yang berjumlah 26. Setelah terbentuknya kode hasil dari proses *Vigenere Cipher*, maka akan di proses lagi menggunakan *Base64*. *Base64* adalah pengubahan karakter ke dalam bilangan biner, dimana dari 8 bit diubah ke dalam 6 bit.

Pada sistem aplikasi enkripsi SMS ini, untuk melakukan proses enkripsi dan dekripsi menggunakan metode yang sama, yaitu menggunakan *Vigenere*

Cipher dan *Base64*. Penggunaan dua metode tersebut menghasilkan kode enkripsi yang cukup baik, tetapi masih memiliki kekurangan dikarenakan hasil ukuran teks SMS yang diproses dengan dua metode tersebut menjadi lebih besar dibandingkan dengan teks aslinya.

Aplikasi ini juga dilengkapi dengan fitur pesan masuk, dimana pengguna dapat membaca SMS yang telah diterimanya. Dalam fitur SMS ini pesan yang diterima masih dalam bentuk enkripsi kemudian dapat di-dekripsi ke dalam bentuk teks aslinya. Secara garis besar, proses yang terjadi dapat digambarkan pada gambar 3.1.



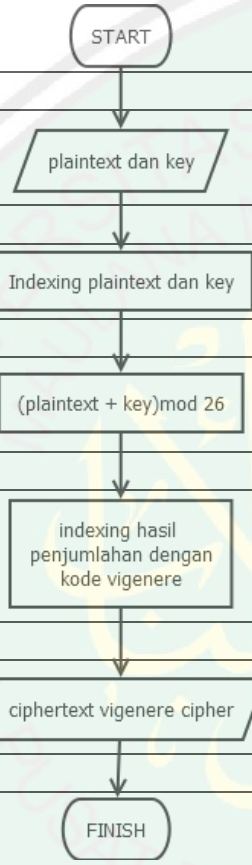
Gambar 3.1 *Flowchart sistem aplikasi enkripsi SMS secara umum*

3.7.1 Tahap *Encoding* SMS

Proses-proses yang dilakukan pada tahapan *encoding* SMS adalah:

a. Pesan (*plaintext*) akan di-enkripsi dengan *Vigenere Cipher*.
b. Sebelum proses enkripsi, ada kunci yang akan digunakan untuk proses enkripsi.
c. Mengubah *plaintext* dan kunci menjadi kode *vigenere* (*indexing*).
d. Kemudian menjumlah hasil *indexing plaintext* dan kunci, dan hasilnya di modulo 26 dan menghasilkan *ciphertext* hasil dari proses enkripsi *vigenere cipher*.
e. Hasil tersebut akan di proses lagi menggunakan *Base64*, dengan mengubah setiap karakter menjadi kode ASCII.
f. Setelah itu, diubah ke bentuk *byte* (8 bit), kemudian dari 8 bit diubah ke bentuk 6 bit.
g. Hasil dari perubahan tersebut akan di ubah menjadi kode *Base64* dan menjadi *ciphertext* dari *Base64*.

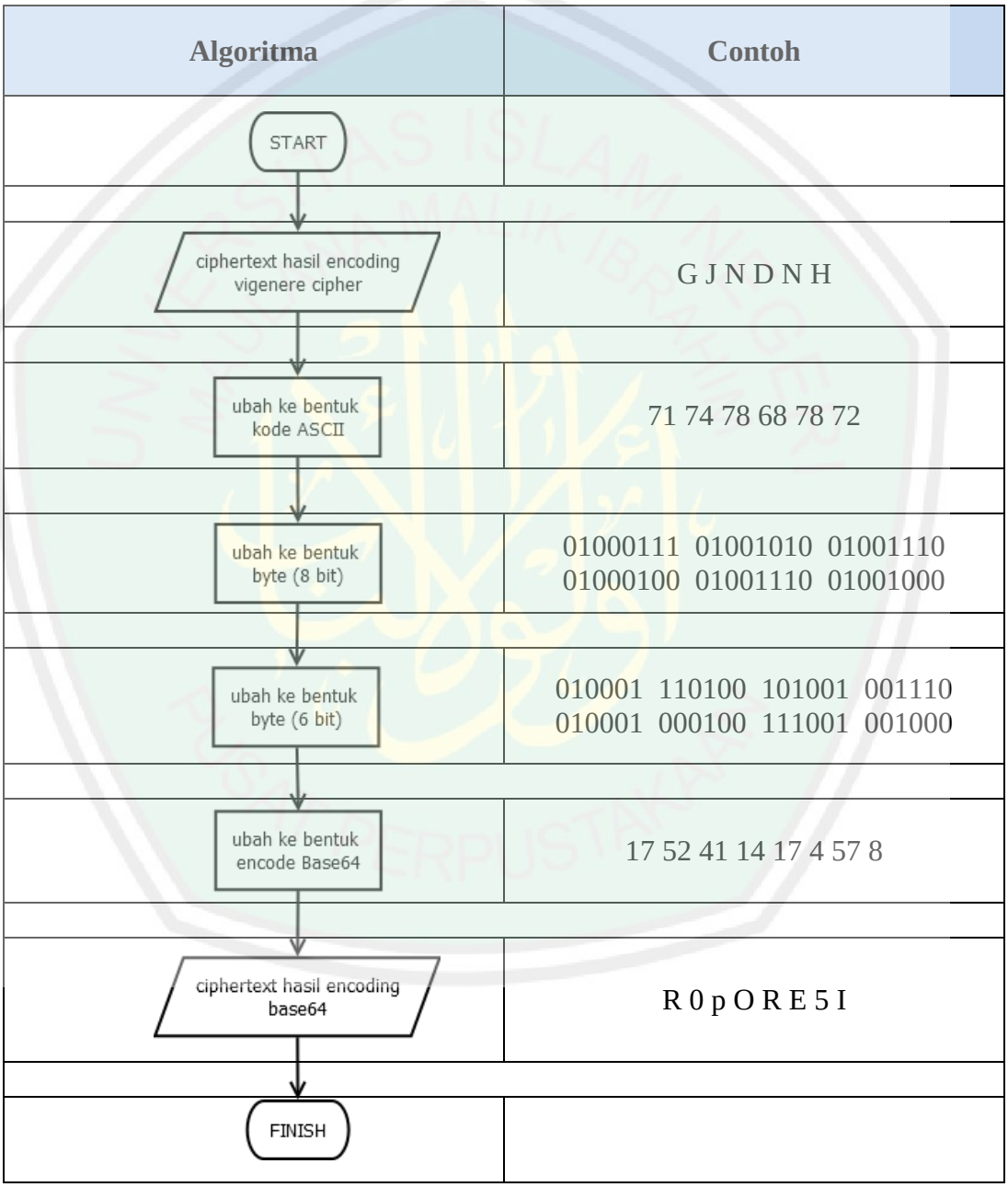
Tahap *encoding Vigenere Cipher* dapat digambarkan dengan diagram alur pada gambar 3.2. Misalkan Plaintext (p) = G I L A N G, Kunci (k) = A B C D.

Algoritma	Contoh
	p = G I L A N G k = A B C D *Indexing* p = 6 8 11 0 13 6 *Indexing* k = 0 1 2 3 Menjumlahkan *plaintext* dengan kunci dengan rumus *((p + k)mod26)* Hasil penjumlahan = 6 9 13 3 13 7 *Ciphertext* = G J N D N H

Gambar 3.2 *Flowchart encoding Vigenere Cipher*.

Hasil dari *encoding Vigenere Cipher* adalah "G J N D N H".

Tahap *encoding Base64* dapat digambarkan dengan diagram alur pada gambar 3.3. Misalkan *ciphertext* hasil dari *encode Vigenere Cipher* adalah "G J N D N H".



Gambar 3.3 *Flowchart encoding Base64.*

Hasil dari enkripsi *Base64* adalah "R0pORE5I".

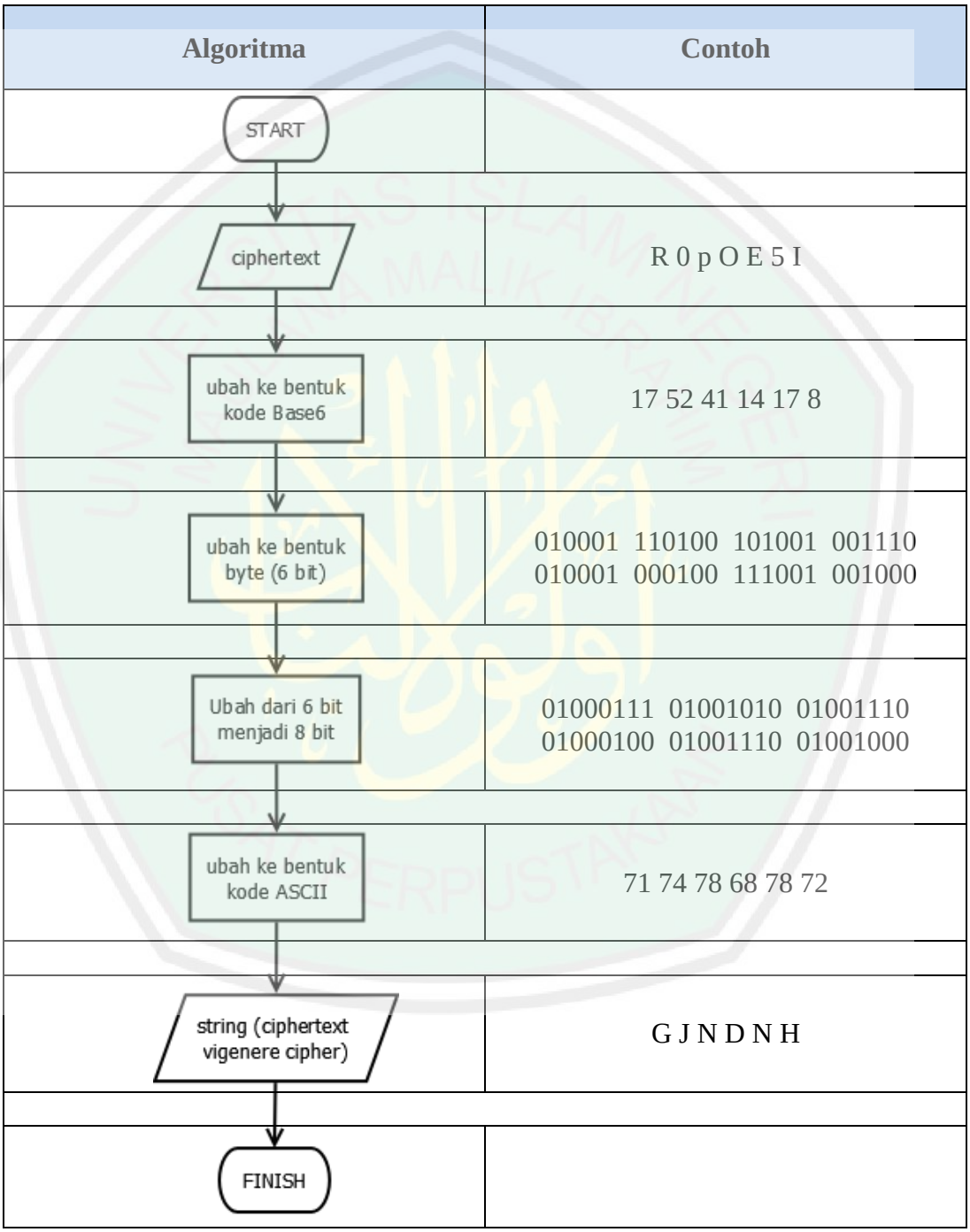
3.7.2 Tahap *Decoding* SMS

Proses-proses yang dilakukan pada tahapan *decoding* SMS adalah:

a. Pesan diterima dalam bentuk *ciphertext*.
b. Mengubah *ciphertext* ke kode *Base64*.
c. Mengubah setiap karakter *ciphertext* ke dalam bentuk *byte* (6 bit), kemudian mengubahnya dari bentuk 6 bit ke dalam bentuk 8 bit.
d. Mengubahnya ke bentuk kode ASCII, mengubah kode ASCII ke bentuk *string* (*ciphertext Vigenere Cipher*).
e. Hasil dari proses dekripsi *Base64* kemudian di-enkripsi dengan *Vigenere Cipher*.
f. Mengubah *ciphertext* dan kunci ke bentuk kode *vigenere* (*indexing*).
g. Menjumlahkan hasil dari *indexing* dengan rumus *"(ciphertext-kunci)mod26*".
h. Hasilnya akan di ubah ke bentuk *string* (*plaintext*).

Tahap *decoding Base64* dapat digambarkan dalam diagram alur pada gambar 3.4.

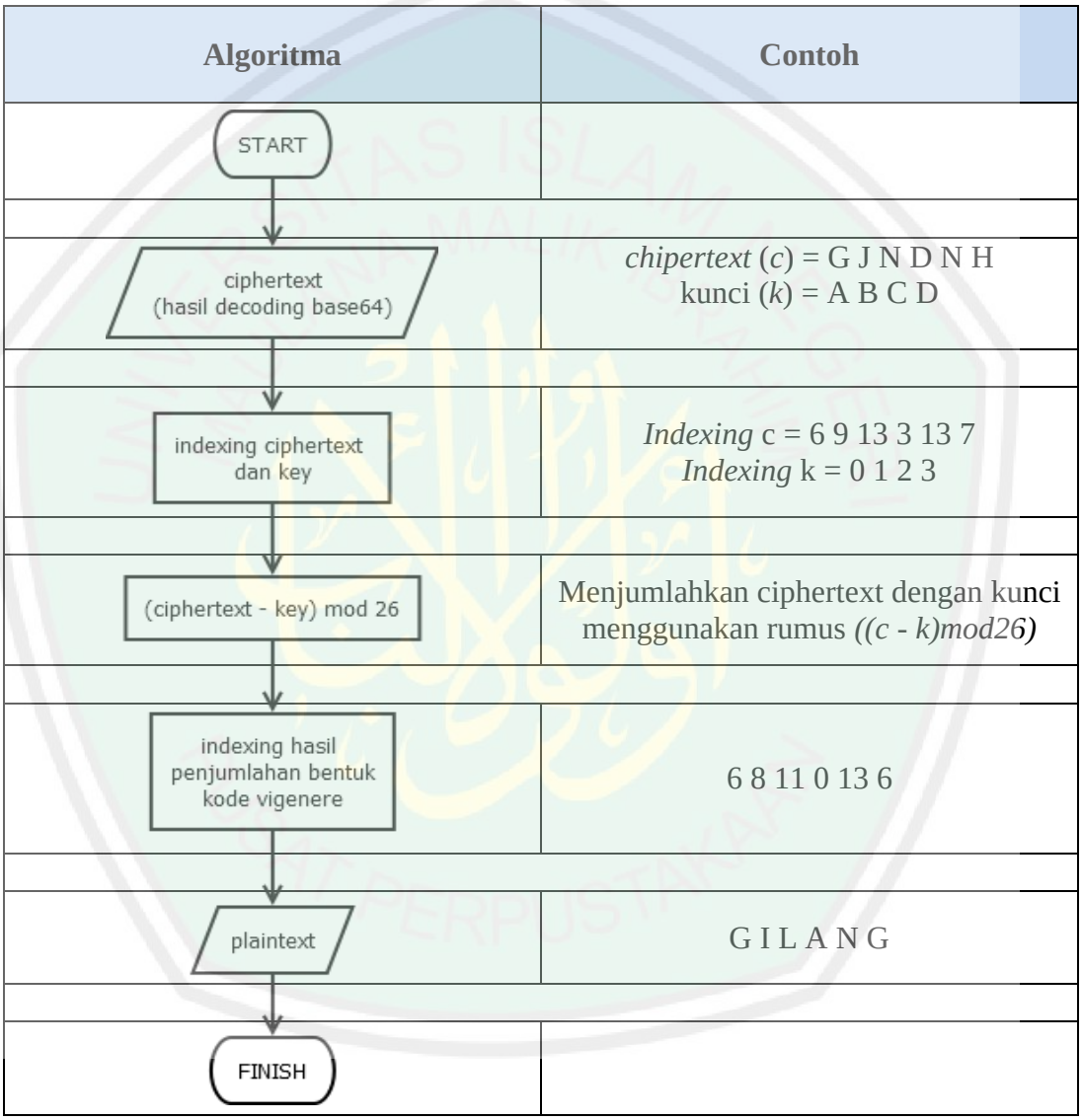
Misalkan *Ciphertext*-nya adalah “R0pORE5l”.



Gambar 3.4 *Flowchart decoding Base64.*

Hasil dari *decoding Base64* adalah “G J N D N H”.

Tahap *decoding Vigenere Cipher* dapat digambarkan dalam diagram alur pada gambar 3.5. Misalkan hasil dari *decoding Base64* adalah “G J N D N H”, dan kunci yang digunakan adalah “ABCD”.



Gambar 3.5 *Flowchart decoding Vigenere Cipher.*

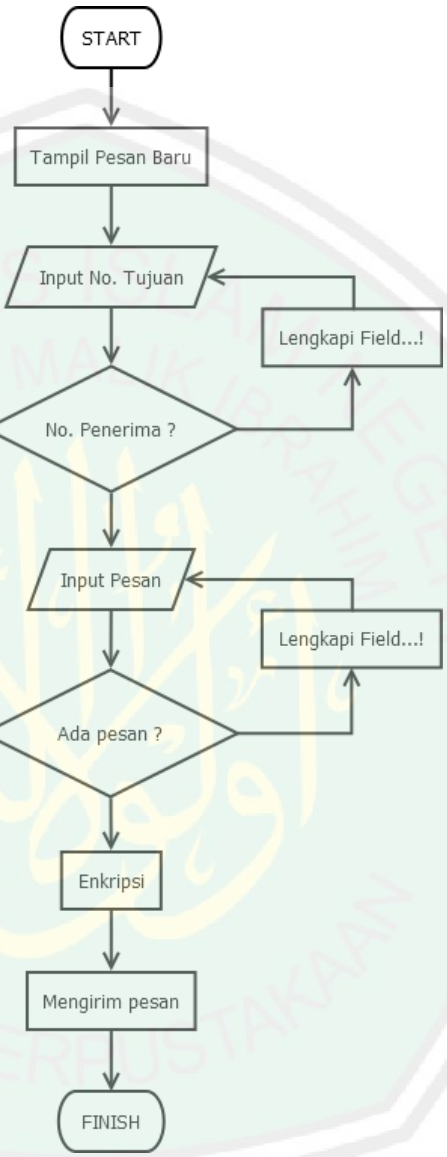
Hasil dari *decoding Vigenere Cipher* adalah “GILANG”.

3.7.3 Tahap Pengiriman SMS

Proses-proses yang dilakukan pada tahapan pengiriman SMS adalah:

a. *User* memasukkan nomor tujuan SMS.
b. *User* mengetikkan teks SMS yang akan dikirimkan.
c. Apabila *user* ingin enkripsi SMS, maka *user* harus menekan tombol yang tertulis enkripsi. Untuk kunci nya telah ter-*input* secara default, *user* juga bisa mengganti secara manual. Setelah *user* menekan tombol enkripsi, maka teks akan diubah ke dalam bentuk kode enkripsi.
d. *User* menekan tombol kirim untuk mengirimkan pesan tersebut.

Tahap pengiriman SMS dapat digambarkan dengan diagram alur pada gambar 3.6.



Gambar 3.6 *Flowchart tahap pengiriman SMS.*

3.7.4 Tahap Penerimaan SMS

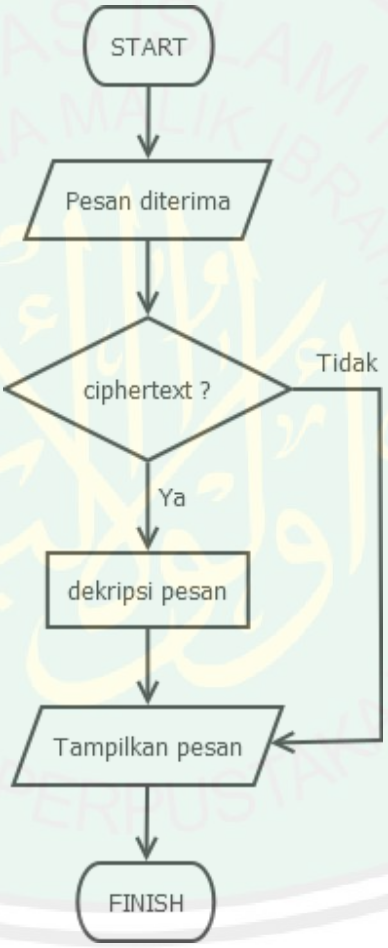
Proses-proses yang dilakukan pada tahapan penerimaan SMS adalah:

a. SMS telah diterima oleh handphone tujuan.

b. Setelah SMS diterima, akan ada notifikasi SMS telah diterima dan masuk ke dalam *Inbox default device* dan pesan masuk aplikasi.

c. Jika SMS yang masuk berupa *ciphertext*, maka untuk membaca kode tersebut harus di-dekripsi menjadi *plaintext*. Proses perubahan ini disebut *decoding*. Setelah mengalami proses *decoding*, kode tersebut baru dapat terbaca.

Tahap penerimaan SMS dapat digambarkan dengan diagram alir pada gambar 3.7.

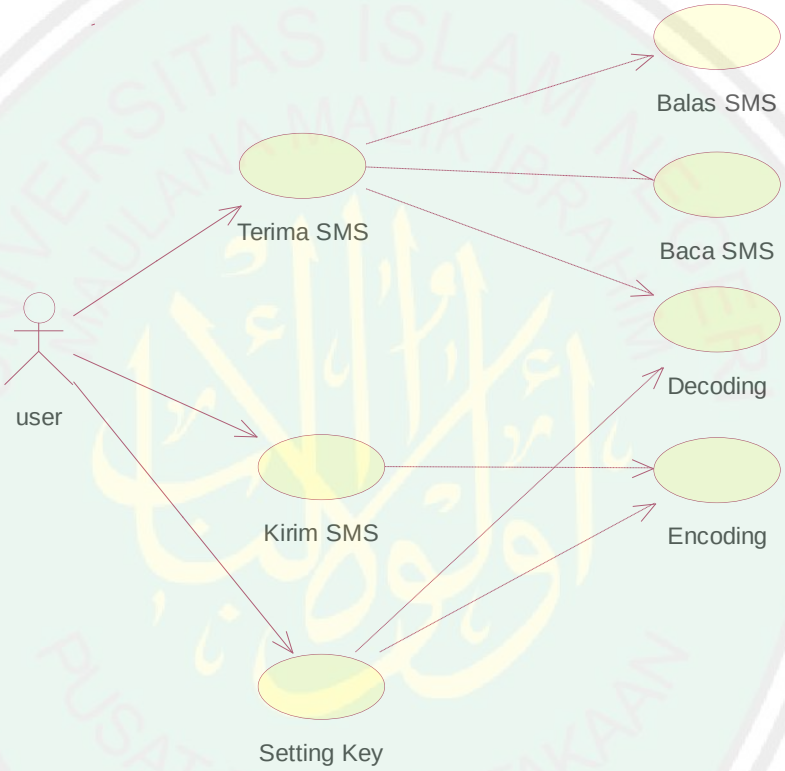


Gambar 3.7 *Flowchart tahap penerimaan SMS.*

3.8 Analisis *Use Case*

Langkah pertama yang harus dilakukan untuk mengetahui kebutuhan *user* adalah memodelkan sistem dengan menggunakan *use case diagram*. *Use case diagram* sendiri adalah gambaran mengenai sistem yang digunakan untuk

memodelkan proses dan data dari sistem. Dengan *use case diagram* ini dapat diketahui proses yang terjadi pada sistem aplikasi enkripsi SMS. Komponen-komponen yang terdapat pada sebuah *use case diagram* aplikasi enkripsi sms, sebagai berikut:



Gambar 3.8 *Use Case Diagram aplikasi enkripsi SMS.*

Dari gambar 3.8, maka dekripsi dari masing-masing *Use case* dapat dijelaskan sebagai berikut:

a. *Use case* : Terima SMS

   Actor : *User*

   Dekripsi : *Use case* ini digunakan untuk terima pesan yang dikirimkan oleh pengguna lain.

b. *Use case* : Balas SMS

Actor : *User*

Dekripsi : *Use case* ini digunakan untuk membalas pesan SMS yang telah diterima, baik berupa pesan terenkripsi maupun tidak.

c. *Use case* : Baca SMS

Actor : *User*

Dekripsi : *Use case* ini digunakan untuk membaca SMS yang telah diterima yang dikirimkan oleh pengguna lain, baik berupa pesan yang ter-enkripsi maupun tidak.

d. *Use case* : Decoding

Actor : *User*

Dekripsi : *Use case* ini berfungsi untuk mengubah pesan yang ter-enkripsi, dari bentuk *ciphertext* ke *plaintext*.

e. *Use case* : Kirim SMS

Actor : *User*

Dekripsi : *Use case* ini digunakan untuk mengirimkan pesan, baik pesan yang akan di-enkripsi maupun tidak.

f. *Use case* : Encoding

Actor : *User*

Dekripsi : *Use case* ini berfungsi untuk meng-enkripsi pesan, pesan tersebut akan diubah ke dalam bentuk *ciphertext* dengan menggunakan metode *Vigenere Cipher* dan *Base64*.

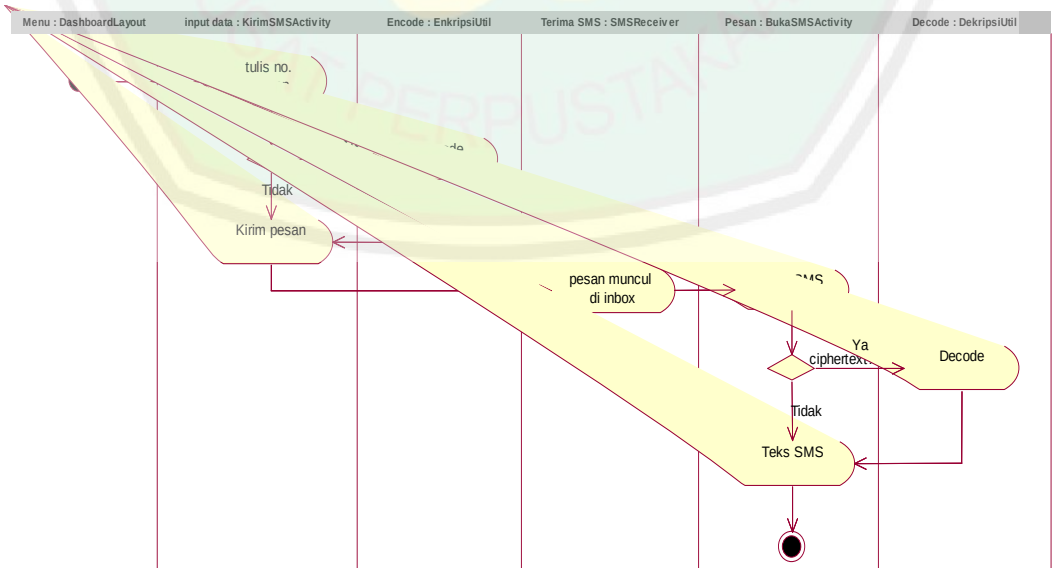
g. *Use case* : *Setting Key*

Actor : *User*

Dekripsi : *Use case* ini bergunakan untuk mengubah kunci yang digunakan untuk proses enkripsi dan dekripsi.

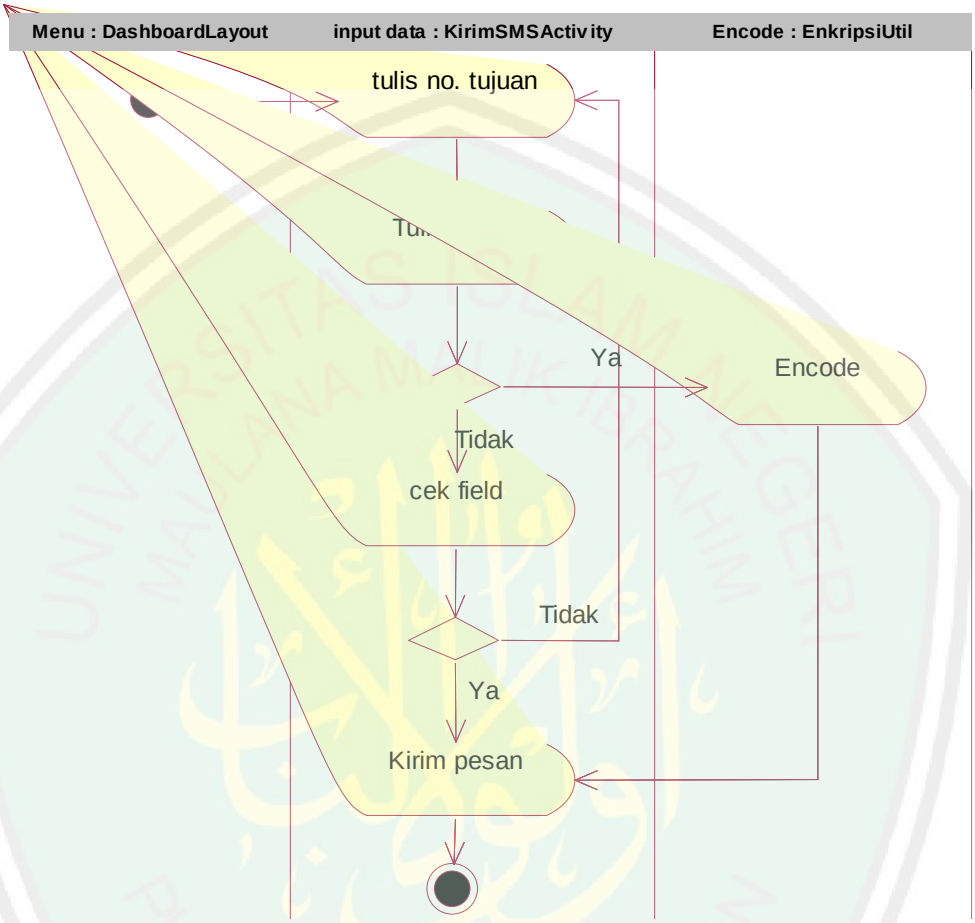
3.9 Analisis *Activity Diagram*

Activity diagram merupakan bentuk *flow diagram* yang memodelkan alur kerja sebuah proses dan urutan aktivitas dalam sebuah proses. Diagram ini mirip dengan *flowchart* karena dapat memodelkan sebuah alur kerja dari satu aktifitas ke aktifitas lainnya. *Activity diagram* juga berguna untuk menggambarkan perilaku parallel atau bagaimana perilaku dalam berbagai *use case* berinterkasi. Berikut *activity diagram* secara keseluruhan proses yang terjadi dalam sistem aplikasi enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64* pada gambar 3.9.



Gambar 3.9 *Activity diagram Aplikasi enkripsi SMS.*

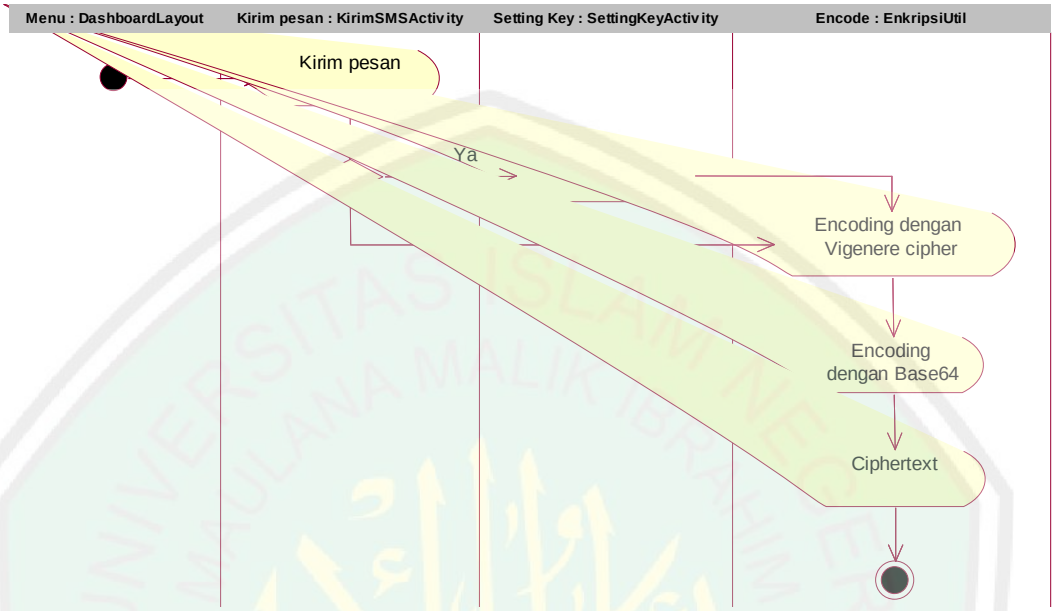
3.9.1 *Activity Diagram* Kirim SMS



Gambar 3.10 *Activity Diagram kirim SMS.*

Pada gambar 3.10, *Activity diagram* kirim SMS dimulai pada saat *user* memilih menu tulis SMS kemudian mengetikan no. tujuan da isi pesan. Setelah *user* mengisi semua *field*, maka pesan bisa di-enkripsi apabila tidak maka *user* bisa mengirimkan pesan tersebut. Jika *field* ada yang kosong, maka akan muncul peringatan untuk mengisi *field* yang kosong.

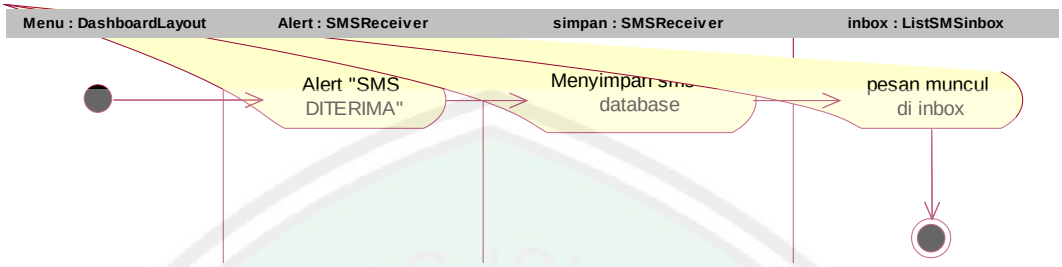
3.9.2 *Activity Diagram Encoding* SMS



Gambar 3.11 *Activity Diagram encoding SMS.*

Pada gambar 3.11, *Activity diagram encoding* diproses ketika *user* sudah mengisi semua *field*. Pada proses encoding, pesan akan di-enkripsi dengan metode *Vigenere Cipher*, dimana setiap karakter akan disubtitusi dengan kunci. Kemudian hasil *ciphertext* tersebut akan di-enkripsi lagi dengan *Base64*, setiap karakter akan di ubah ke dalam bentuk kode ASCII dan diubah ke bentuk 8 bit, kemudian akan di ubah menjadi 6 bit dan dikembalikan lagi ke kode ASCII kemudian di kembalikan ke bentuk *string*.

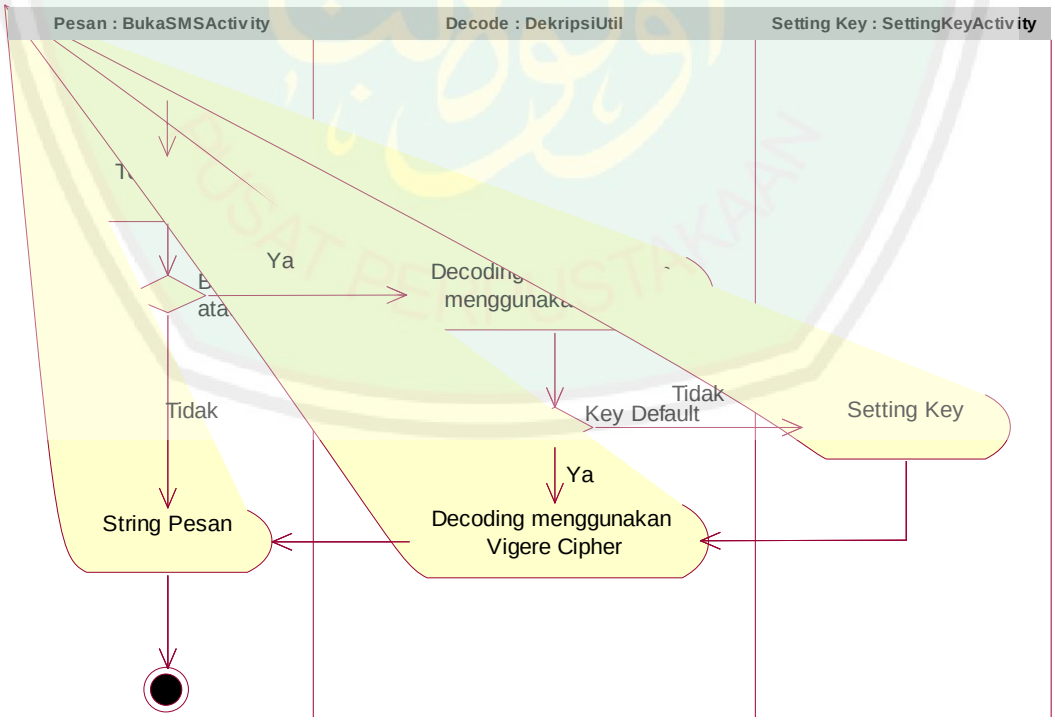
3.9.3 *Activity Diagram* Terima SMS



Gambar 3.12 *Activity Diagram terima SMS.*

Ketika ada pesan masuk, *smartphone* penerima akan mengluarkan sebuah peringatan pesan masuk. Ketika pesan masuk, aplikasi akan menyimpannya ke dalam *database* aplikasi yang kemudian akan muncul di *inbox* atau pesan masuk.

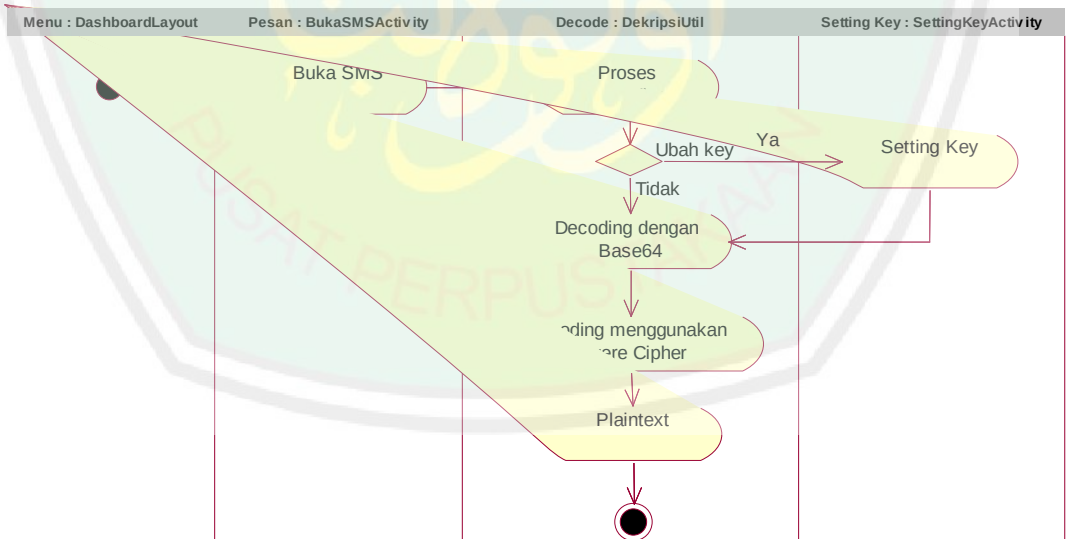
3.9.4 *Activity Diagram* Baca SMS



Gambar 3.13 *Activity diagram baca SMS.*

Jika pesan yang diterima berupa *ciphertext*, maka *user* memilih tombol dekrip untuk dekripsi pesan tersebut. Pada proses dekripsi pesan, teks SMS akan di-dekripsi dengan *Base64* yang kemudian akan di-dekripsi dengan *Vigenere Cipher*. Pada proses *decoding* menggunakan *Vigenere Cipher* harus menyesuaikan dengan kunci yang digunakan oleh pengirim, jika yang digunakan tidak sama maka proses tersebut tetap berjalan namun hasil dari proses dekripsi tidak sesuai dengan teks SMS yang dikirimkan, untuk mengatasi permasalahan tersebut, kunci yang digunakan untuk proses dekripsi bisa diubah sesuai kunci yang dipakai oleh pengirim. Setelah proses tersebut selesai, teks tersebut di ubah kembali ke dalam bentuk string dan ditampilkan di *field* pesan.

3.9.5 *Activity Diagram Decoding* SMS



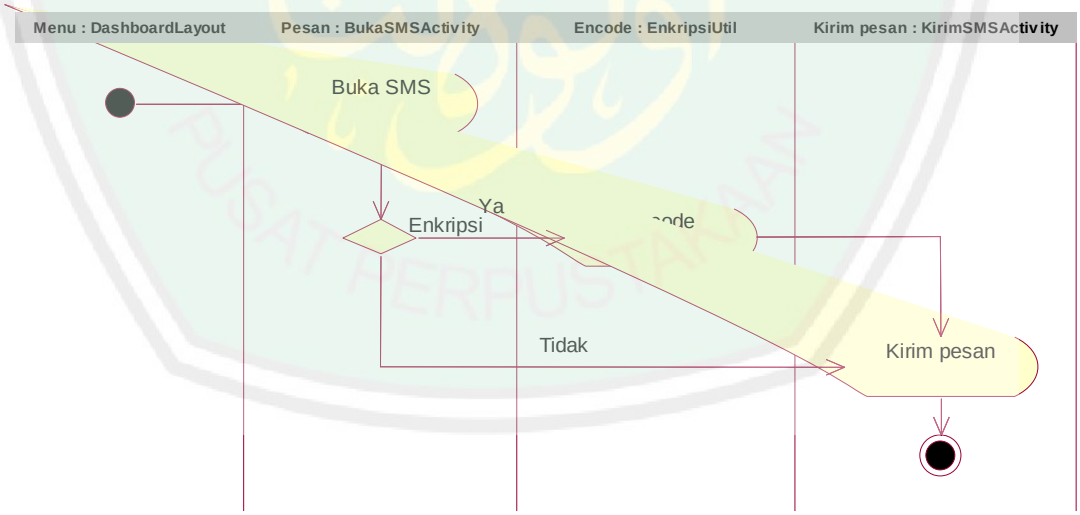
Gambar 3.14 *Activity diagram decoding SMS*.

Proses ini dilakukan apabila pesan yang diterima berupa *ciphertext*. Proses *decoding* ini untuk merubah kembali pesan yang ter-enkripsi ke bentuk

pesan asli (*plaintext*). Sebelum proses dilakukan, perlu adanya pengecekan kunci karena berperan penting dalam proses ini. Proses dekripsi *Base64* adalah pengubahan setiap karakter pada *ciphertext* akan diubah kedalam kode *ASCII* yang kemudian diubah ke bentuk 8 bit, dan diubah ke bentuk 6 bit kemudian di ubah ke dalam kode *ASCII* setelah itu dikembalikan ke bentuk *string*.

Setelah proses decoding *Base64* selesai, maka akan dilanjutkan ke proses berikutnya dengan menggunakan metode *Vigenere cipher*. Hasil dari proses *decoding* pertama akan diubah ke dalam kode *Vigenere Cipher* kemudian akan disubtitusi dengan kunci, setalah itu akan diubah bentuknya dari *ciphertext* ke *plaintext*. Bentuk *plaintext* inilah yang akan ditampilkan ke *field* pesan.

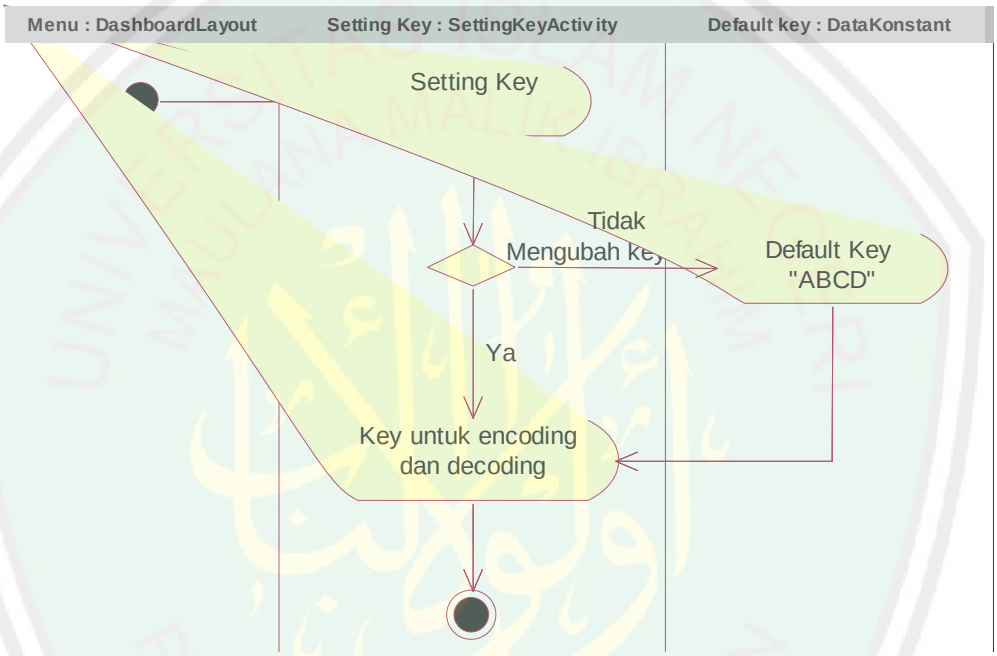
3.9.6 *Activity Diagram* Balas SMS



Gambar 3.15 *Activity diagram balas SMS.*

Ketika *user* buka pesan SMS dari *user* lain, maka *user* tersebut dapat membalas pesan SMS tersebut. Pada aplikasi enkripsi SMS, *user* dapat memilih apakah pesan tersebut ingin di-enkripsi atau tidak.

3.9.7 *Activity Diagram* Atur Kunci

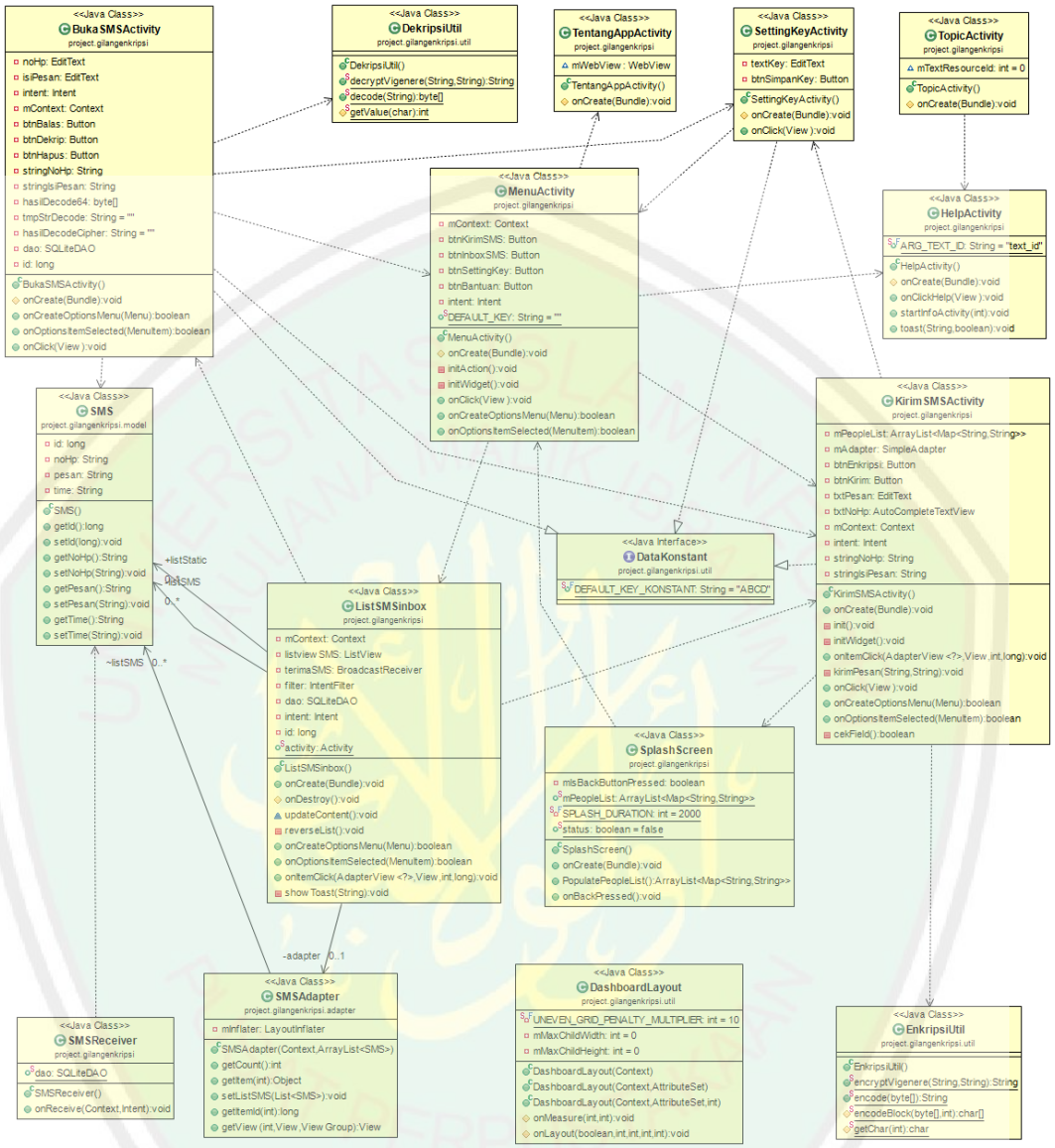


Gambar 3.16 *Activity diagram atur kunci*

Pada proses ini adalah, mengubah *key* atau kunci yang akan digunakan dalam proses *encoding* dan *decoding* pada *Vigenere Cipher*.

3.10 Analisis *Class Diagram*

Class Diagram merupakan visualisasi struktur kelas-kelas dari suatu sistem. Dalam *class diagram* diperlihatkan hubungan antar kelas dan penjelasan mengenai tiap-tiap kelas. *Class diagram* berperan untuk menangkap struktur dari semua kelas membentuk arsitektur sistem.



Gambar 3.17 *Class diagram Aplikasi enkripsi SMS.*

Pada gambar 3.17, terdapat enam belas *class* yang digunakan, antara lain:

a. *Class MenuActivity*

Class MenuActivity digunakan untuk menampilkan menu aplikasi enkripsi SMS yang berisi menu-menu, antara lain Tulis Pesan, Pesan Masuk, Atur Kunci, dan Bantuan.

b. *Class KirimSMSActivity*

Class KirimSMSActivity digunakan untuk mengirimkan pesan, dalam *class* ini terdapat *field* no. tujuan, pesan, tombol enkrip dan tombol kirim. Pada *class* ini terdapat *menu item* yang digunakan untuk mengatur kunci yang berfungsi untuk proses enkripsi.

c. *Class ListSMSinbox*

Class ListSMSinbox digunakan untuk mengatur pesan yang tersimpan pada *database*. *Class* ini juga mengatur aksi rekasi yang berhubungan dengan SMS yang masuk tersebut. Dalam class ini terdapat *menu item* untuk membuat pesan baru.

d. *Class BukaSMSActivity*

Class BukaSMSActivity digunakan untuk membuka pesan pada *inbox*, disini terdapat fitur dekripsi pesan dan balas pesan, terdapat *menu item* untuk mengatur kunci yang digunakan untuk proses dekripsi.

e. *Class SettingKeyActivity*

Class SettingKeyActivity digunakan untuk mengatur kunci yang digunakan untuk proses enkripsi dan dekripsi.

f. *Class TentangAppActivity*

Class TentangAppActivity digunakan untuk menampilkan biografi singkat pembuat aplikasi enkripsi SMS menggunakan metode *Vigenere Cipher* dan *Base64*.

g. *Class HelpActivity*

Class HelpActivity digunakan untuk menampilkan form bantuan yang berisi tentang penjelasan tentang fitur-fitur pada aplikasi.

h. *Class TopicActivity*

Class TopicActivity berhubungan dengan *Class HelpActivity* digunakan untuk menampilkan detail dari penjelasan fitur-fitur pada aplikasi.

i. *Class SplashScreen*

Class SplashScreen digunakan untuk menampilkan *splash screen* dan membaca data pada kontak telepon seluler.

j. *Class SMSReceiver*

Class SMSReceiver digunakan untuk menangkap dan menyimpan pesan yang diterima oleh perangkat *mobile* yang sudah terpasang aplikasi enkripsi SMS.

k. *Class SMSAdapter*

Class SMSAdapter digunakan untuk mengatur pesan yang diterima oleh perangkat *mobile*. Dimana class ini berhubungan dengan *class List*SMS*Activity* dan *class* SMS.

l. *Class SMS*

Class SMS digunakan untuk mendeklarisasikan variable no. pesan, *time* dan pesan, serta sebagai kelas database.

m. *Class DashboardLayout*

Class DashboardLayout digunakan untuk mengatur tampilan dari menu Aplikasi enkripsi SMS.

n. *Class DataKonstant*

Class DataKonstant digunakan untuk mengatur default kunci yang digunakan pada proses enkripsi dan dekripsi.

o. *Class EnkripsiUtil*

Class EnkripsiUtil berfungsi untuk proses enkripsi, dimana terdapat metode *Vigener cipher* dan *Base64*.

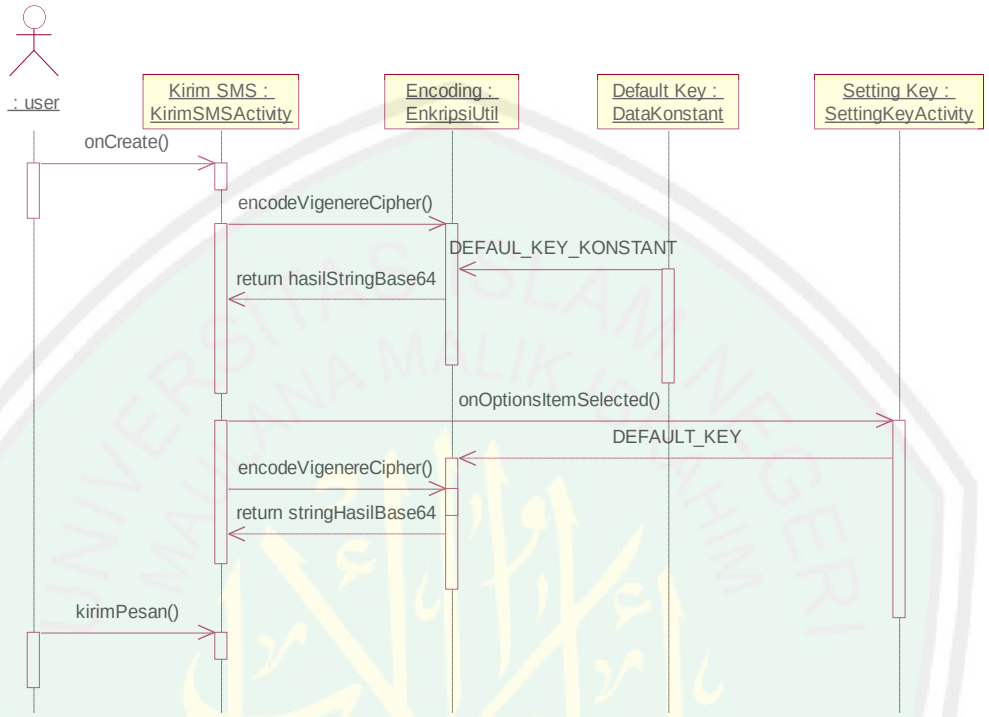
p. *Class DekripsiUtil*

Class DekripsiUtil berfungsi untuk proses dekripsi, dimana terdapat metode *Vigenere Cipher* dan *Base64*.

3.11 Analisis *Sequence Diagram*

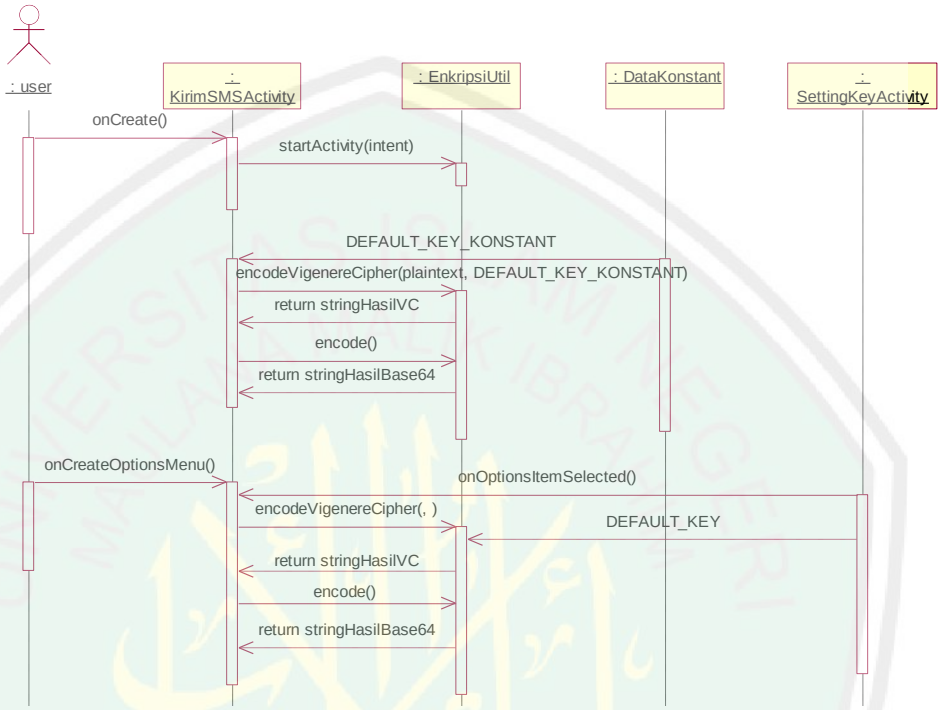
Sequence diagram menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Dalam *sequence diagram* diperlihatkan tahap demi tahap apa saja yang seharusnya terjadi untuk menghasilkan sesuatu di dalam *Use case*. Pada sistem aplikasi enkripsi SMS, terdapat tujuh *sequence diagram*, antara lain:

3.11.1 *Sequence Diagram* Kirim SMS



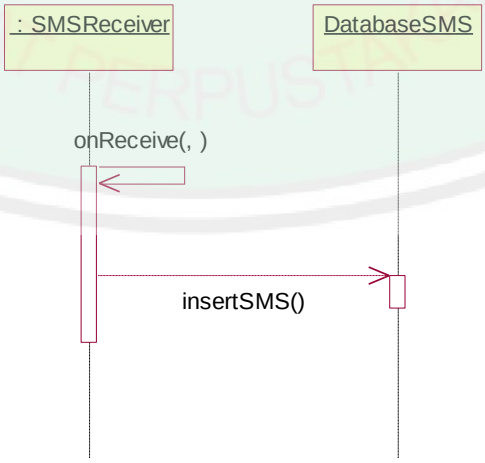
Gambar 3.18 *Sequence diagram kirim SMS.*

3.11.2 *Sequence Diagram Encoding* SMS



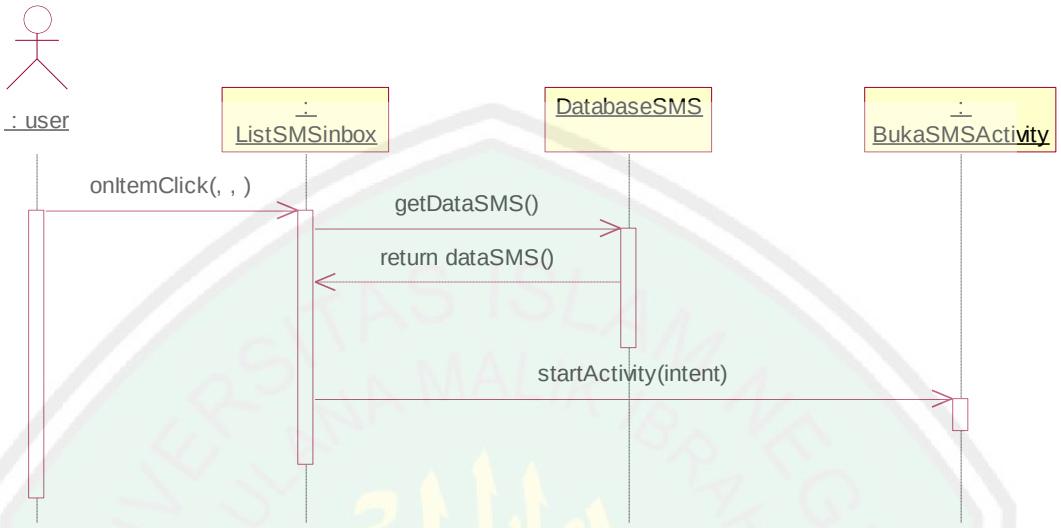
Gambar 3.19 *Sequence diagram encoding SMS.*

3.11.3 *Sequence Diagram* Terima SMS



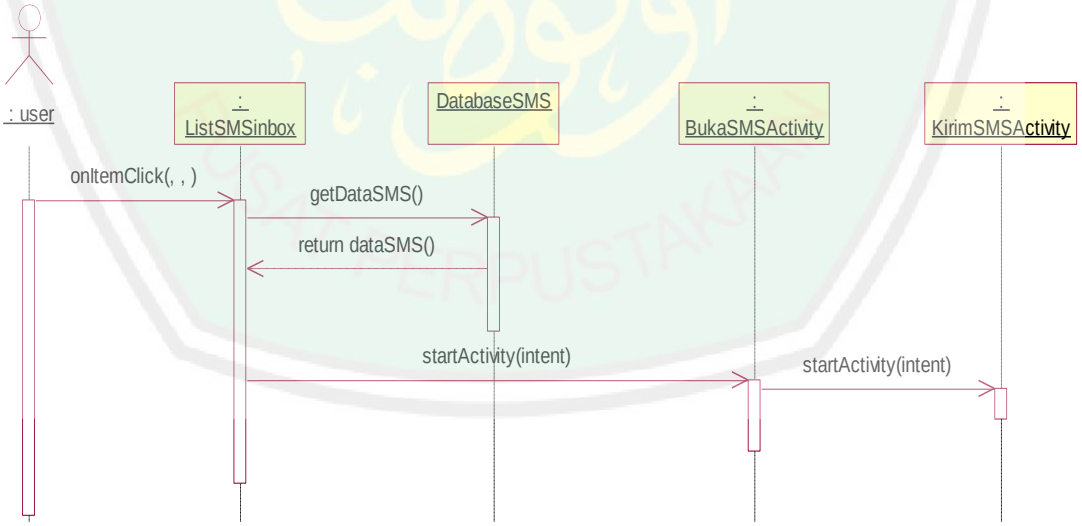
Gambar 3.20 *Sequence diagram terima SMS.*

3.11.4 *Sequence Diagram* Baca SMS



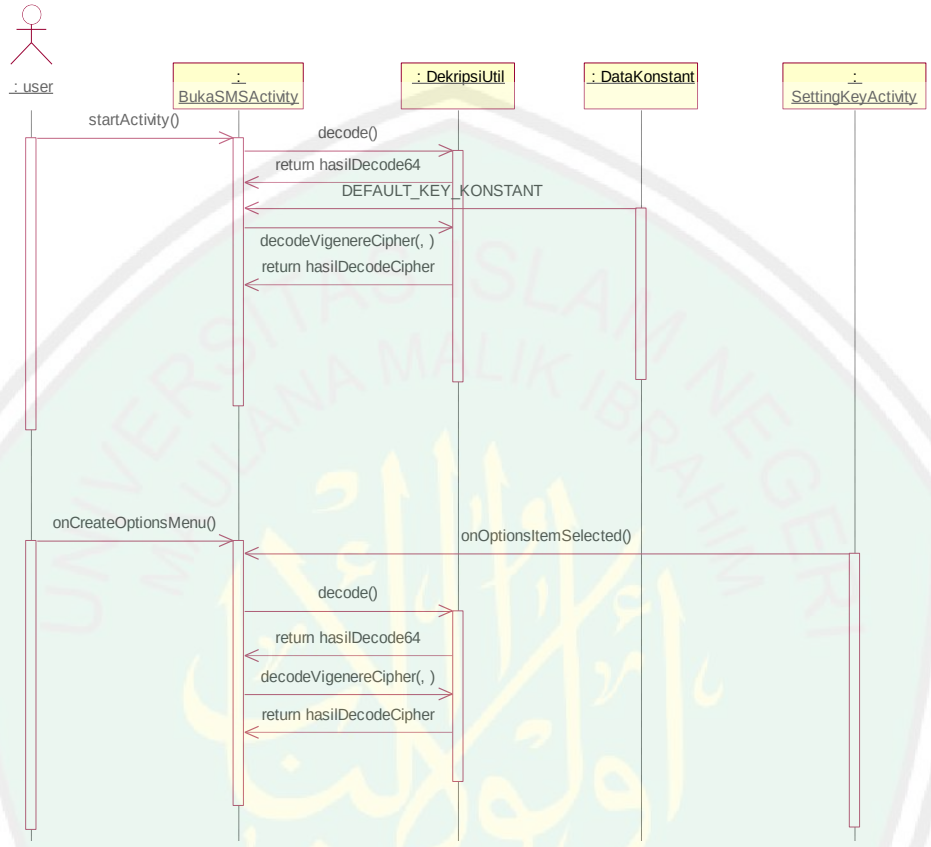
Gambar 3.21 *Sequence diagram baca SMS.*

3.11.5 *Sequence Diagram* Balas SMS



Gambar 3.22 *Sequence diagram balas SMS.*

3.11.6 *Sequence Diagram Decoding* SMS



Gambar 3.23 *Sequence diagram decoding SMS.*

3.12 Perancangan Data

Perancangan data dilakukan untuk mendesain dan merencanakan data-data apa saja yang akan dipakai dan diolah dalam proses enkripsi SMS beserta dengan tipe dari masing-masing data tersebut.

3.12.1 Perancangan Data *Input*

Data *input* adalah data-data yang dimasukkan untuk diproses. Pada tabel 3.1 merupakan perancangan data *input* yang nantinya akan digunakan dalam Aplikasi enkripsi SMS.

Tabel 3.1 *Perancangan data input.*

No.	Nama Data	Tipe Data	Keterangan
1	isiPesan	String	Pesan yang diketikkan oleh *user*. Masing-masing karakter pesan ini nantinya akan di-enkripsi dengan metode *Vigenere Cipher* dan *Base64*, dan sebaliknya.
2	txtKey	String	Karakter ini yang kemudian dijadikan kunci untuk proses encoding dan decoding dalam *Vigenere Cipher*.

3.12.2 Perancangan Data Proses

Tabel 3.2 *Perancangan data proses.*

No.	Nama Data	Tipe Data	Keterangan
1	isiPesan	String	Pesan yang akan di proses menggunakan metode *Vigenere Cipher*. Setiap karakter pada pesan ini akan di konversi ke bentuk ASCII kemudian akan di olah dengan kunci. Setelah itu akan di ubah ke bentuk *plaintext*. Dan sebalik untuk proses dekripsi
2	txtKey	String	Berfungsi untuk proses enkripsi dan dekripsi pesan.

3.12.3 Perancangan Data *Output*

Data *output* adalah data jadi yang telah selesai mengalami proses pengolahan. Pada tabel 3.3 merupakan perancangan data *output* yang akan digunakan dalam Aplikasi enkripsi SMS.

Tabel 3.3 *Perancangan data output.*

No	Nama Data	Tipe Data	Keterangan
1	isiPesan	String	Pesan yang akan di proses menggunakan metode *Vigenere Cipher*. Setiap karakter pada pesan ini akan di konversi ke bentuk ASCII kemudian akan di olah dengan kunci. Setelah itu akan di ubah ke bentuk *plaintext*. Dan sebalik untuk proses dekripsi

3.13 Perancangan Desain Database

Dalam pembuatan program ini dibutuhkan desain *database* untuk menyimpan data yang akan digunakan dalam proses sistem aplikasi enkripsi SMS. Desain *database* ini menjelaskan tabel dan *field* yang digunakan. Berikut adalah tabel yang diguakan untuk proses sistem aplikasi enkripsi SMS.

3.13.1 Tabel SMS

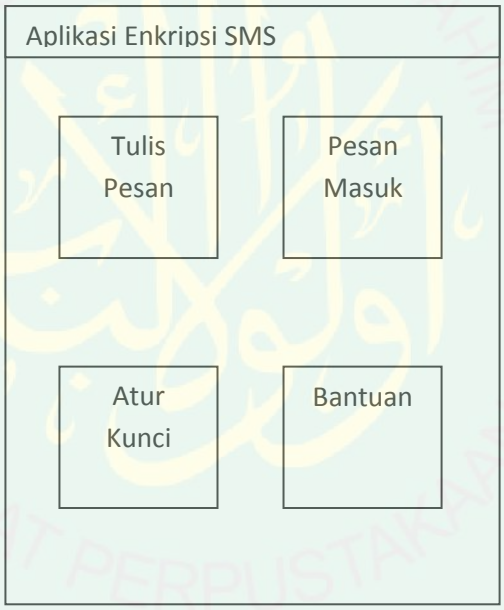
Tabel SMS adalah tabel data database yang digunakan untuk menyimpan data-data SMS, secara rinci pada tabel 3.4.

Tabel 3.4 *Perancangan database SMS.*

No.	Nama Field	Tipe Data	Keterangan
1	Id	Varchar	Menyimpan kode sebagai primary key dan unik untuk kode pesan masuk
2	noHp	Varchar	Menyimpan data no hp
3	Pesan	Varchar	Menyimpan data pesan SMS
4	Time	Varchar	Menyimpan waktu pesan SMS diterima.

3.14 Perancangan *Interface*

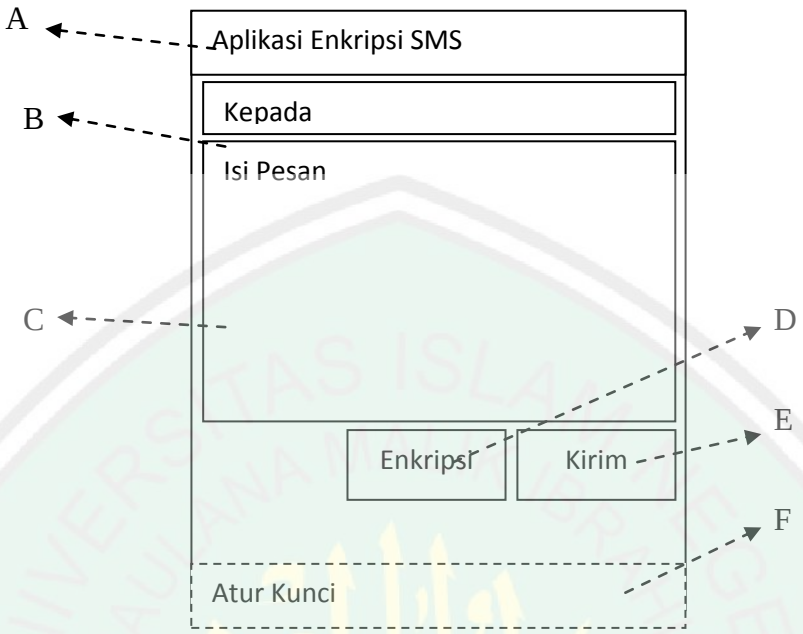
Dalam mendesain sebuah sistem, salah satu hal yang perlu diperhatikan adalah rancangan tersebut harus dapat memudahkan *user* dalam menggunakan sistem aplikasi yang dibuat, sehingga perlu diperhatikan dalam mengatur letak *button*, *textfield*, *menu*, ataupun komponen visual yang lain sehingga tidak membingugkan *user* dalam pemakaian. Berikut adalah perancangan menu utama sistem aplikasi enkripsi SMS:



Gambar 3.24 *Perancangan tampilan menu utama aplikasi enkripsi SMS*

3.14.1 Tulis Pesan

Menu Tulis Pesan digunakan untuk menulis pesan baru, enkripsi dan mengirimkan ke nomor tujuan. Berikut adalah perancangan menu Tulis Pesan:



Gambar 3.25 *Perancangan tampilan tulis pesan.*

Keterangan:

A. Nama Aplikasi

B. *Field* no. tujuan, berisi nomor tujuan pesan tersebut akan dikirimkan.

C. *Field* Pesan, berisi pesan yang akan dikirimkan.

D. *Button* Enkripsi, digunakan untuk enkripsi pesan yang ada di dalam *field* pesan.

E. *Button* Kirim, digunakan untuk mengirimkan pesan.

F. *Menu Item* Atur Kunci, digunakan untuk mengatur kunci.

3.14.2 Pesan Masuk

Menu Pesan Masuk digunakan untuk menampilkan pesan yang masuk yang dikirimkan oleh *user* lain. Tampilan di dalam *menu pesan masuk* adalah sebagai berikut :

Gambar 3.26 *Perancangan tampilan pesan masuk*

Keterangan :

A. Nama Aplikasi

B. No. pengirim pesan

C. Waktu penerimaan pesan

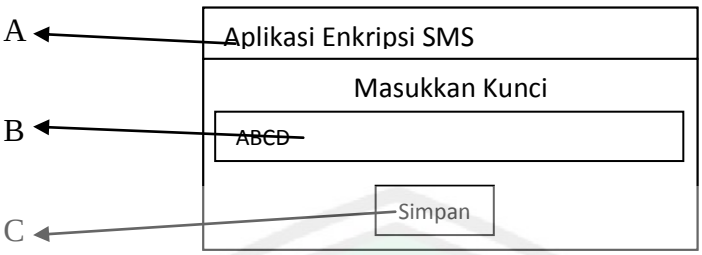
D. Isi pesan yang diterima

E. *List* pesan SMS

F. Menu Item

3.14.3 Atur Kunci

Menu Atur Kunci digunakan untuk mengatur kunci yang digunakan untuk proses enkripsi dan dekripsi metode *Vigenere Cipher*. Tampilan menu atur kunci adalah sebagai berikut:



Gambar 3.27 *Perancangan tampilan atur kunci*

Keterangan

A. Nama Aplikasi

B. *Field* pengisian kunci, jika *field* dibiarkan kosong maka kunci akan *default* dengan kunci "ABCD"

C. *Button* Simpan, digunakan untuk menyimpan kunci.

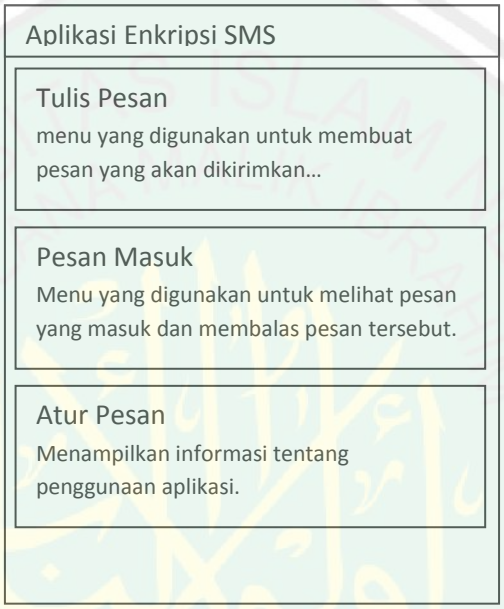
3.14.4 Info

Menu Info digunakan untuk menampilkan informasi singkat tentang aplikasi dan pembuat aplikasi enkripsi SMS.

Gambar 3.28 *Perancangan tampilan info.*

3.14.5 Bantuan

Menu Bantuan digunakan untuk menampilkan informasi penggunaan aplikasi. Tampilan menu bantuan sebagai berikut:



Gambar 3.29 *Perancangan tampilan bantuan*

3.15 Kuesioner Uji Kelayakan Produk

Kuesioner dilakukan untuk menghitung hasil survei kelayakan aplikasi. Survei ditujukan kepada pengguna dengan mengisi pernyataan sesuai dengan pengamatan pada aplikasi. Pengukuran kuesioner ini menggunakan skala *Likert*, dimana digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang. Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item-item instrument yang dapat berupa pernyataan atau pertanyaan.(Sugiyono, 2010) Berikut pernyataan pada kuesioner:

1. Ketertarikan dengan Aplikasi
2. Tampilan Aplikasi
3. Kesesuaian desain warna
4. Kelengkapan fitur
5. Kemudahan penggunaan fitur
6. Penggunaan bahasa
7. Manfaat Aplikasi
8. Pengembangan Aplikasi
9. Publikasi Aplikasi pada masyarakat
10. Tanggapan pengguna secara keseluruhan

Adapun nilai dari pernyataan diasumsikan dengan huruf, maka keterangan sebagai berikut:

SB	= Sangat Baik	diberi skor	5
B	= Baik	diberi skor	4
C	= Cukup	diberi skor	3
K	= Buruk	diberi skor	2
SK	= Sangat Buruk	diberi skor	1

Setelah hasil dari kuesioner dianalisis, maka akan dicari tingkat kelayakan produk yang ditentukan dari rata-rata prosentase jawaban responden, berikut tingkatan kelayakan produk:

Sangat layak = 81% - 100%

Layak = 61% - 80%

Biasa = 41% - 60%

Tidak layak = 21% - 40%

Sangat tidak layak = 1% - 20%

Berdasarkan tingkatan tersebut, maka akan diperoleh hasil yang nanti akan digunakan dalam menetapkan tingkat kelayakan produk dari hasil kuesioner.

BAB IV

IMPLEMENTASI HASIL DAN PEMBAHASAN

4.1 Ruang Lingkup Implementasi

Implementasi merupakan proses transformasi representasi rancangan ke bahasa pemrograman yang dapat dimengerti oleh komputer. Pada bab ini akan dibahas hal-hal yang berkaitan dengan implementasi sistem enkripsi SMS.

Lingkungan implementasi yang akan dipaparkan meliputi lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Ruang Lingkup Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan aplikasi enkripsi SMS pada *smartphone* menggunakan metode *Vigenere Cipher* dan *Base64* ini adalah sebagai berikut:

a. Laptop Acer 4730z
   1. Processor Intel(R) Pentium(R) Dual CPU T3200 2.00 GHz
   2. RAM 1.93 GB
   3. Harddisk dengan kapasitas 160 GB
   4. LCD Monitor 14.1” inch

b. Samsung Galaxy Gio s5660

4.1.2 Ruang Lingkup Perangkat Lunak

Adapun perangkat lunak yang digunakan untuk pengembangan aplikasi ini antara lain:

a. Sistem Operasi Windows 7
b. Eclipse
c. SDK Android

Perangkat lunak dibangun dengan menggunakan bahasa pemrograman JAVA dengan berbasis Android. Hasil implementasi akan diujikan pada *mobile device* Samsung Galaxy Gio S5660.

Gambar 4.1 *Samsung Galaxy Gio S5660*

4.2 Implementasi Sistem

Pada subbab ini akan memaparkan implementasi sistem berdasarkan rancangan program. Rancangan yang telah dibangun akan diimplementasikan ke

dalam bentuk *source code* dalam bahasa pemrograman Java berbasis Android. Berikut adalah paparan implementasi dari perangkat lunak yang telah dibangun.

Dalam pembuatan aplikasi ini telah dijelaskan dalam Al-Qur'an pada Surat Al-Quraisy ayat 4, sebagai berikut :

ٱلَّذِىٓ أَطْعَمَهُم مِّن جُوعٍ وَءَامَنَهُم مِّنْ خَوْفٍۢ ﴿٤﴾

"Yang telah memberi makan kepada mereka untuk menghilangkan lapar dan mengamankan mereka dari ketakutan." (QS. Al-Quraisy : 4).

Dari ayat tersebut menerangkan secara jelas bahwa Allah lah yang memberi kita nikmat berupa makanan sehingga hilanglah segala bentuk kelaparan dan hanya Allah lah yang memberi keamanan dari segala ketakutan. Maha Suci Allah yang maha mengaruniakan dan menjamin keamanan bagi manusia.

Apabila dijelaskan maka sesungguhnya Allah telah memberikan manusia segala apa yang telah mereka butuhkan secara gratis untuk memenuhi apa yang menjadi kebutuhan mereka. Dan selain daripada itu Allah jugalah yang telah memberikan rasa aman dari sesuatu yang jahat (tidak bertanggungjawab) sehingga daripada itu manusia dapat hidup dan berkembang. Untuk itu, keseluruhan proses dalam pengerjaan aplikasi enkripsi SMS pada *smartphone* Android menggunakan metode *Vigenere Cipher* dan *Base64* guna mengamankan SMS, semuanya merupakan pemberian dari Allah semata dan hanya kepada Allah semua akan kembali.

Akan tetapi kebanyakan manusia itu tidak mengetahui tentang pemberian keamanan ini, Allah SWT berfirman, *"Demi masa. Sesungguhnya manusia itu benar-benar berada dalam kerugian."* (QS. Al-'Ashr : 1-2), maka diperlukan beberapa program yang harus diimplementasikan baik dari segi perangkat keras maupun perangkat lunak komputer.

Al-Qur'an Surat Al-'Ashr ayat 1-2 menjelaskan bahwa manusia memang benar-benar berada dalam kerugian apabila tidak memanfaatkan waktu yang telah diberikan oleh Allah SWT secara optimal untuk mengerjakan perbuatan-perbuatan baik. Dan untuk memenuhi panggilan Allah tersebut maka penulis membuat aplikasi enkripsi SMS ini guna mengembangkan rasa aman dalam menggunakan layanan SMS, sehingga dapat terhindar dari tindakan kriminalitas.

4.2.1 Tahap Enkripsi

Dalam proses enkripsi teks atau pesan yang akan dikirimkan mengalami dua kali tahap enkripsi dengan menggunakan *Vigenere Cipher* dan *Base64*. Berikut penjelasan proses enkripsi beserta *pseudocode*-nya.

a. Proses enkripsi pada *Vigenere Cipher*

Pada proses enkripsi *Vigenere Cipher*, dimana *plaintext* dan *key* atau kunci akan diproses pada `String encryptVigenere(String text, String key)`, didalam proses ini *plaintext* dan *key* akan diubah menjadi kode ASCII kemudian akan dikurangi dengan 65 sehingga menghasilkan kode Caesar. Setelah proses itu selesai maka *plaintext* dan *key* akan diolah dengan `((basicPlaintextValue + basicKeyValue) % 26)` sehingga

menghasilkan *ciphertext* dari *Vigenere Cipher* dan nilai dari *ciphertext* akan ditambahkan kembali dengan 65 sehingga kembali ke bentuk kode ASCII dan diubah kembali ke bentuk string.

```
public static String encryptVigenere(String text, String
key){
            String ciphertext = "";
            int kIndex = 0;

            // mengubah plaintext dan key ke bentuk kapital
            text = text.toUpperCase();
            key = key.toUpperCase();

            // menghitung jumlah karakter plaintext
            for (int ptextIndex = 0; ptextIndex <
            text.length(); ptextIndex++) {

            // mengubah setiap karakter ke dalam bentuk
            // ASCII
            char pChar = text.charAt(ptextIndex);

            int asciiValue = (int) pChar;

            // jika terdapat spasi maka hasil proses sama
            if (pChar == ' ') {
                  ciphertext += pChar;
                  continue;
            }

            // jika karakter sama dengan kode ASCII 65-90
            // maka proses akan berlanjut
            if (asciiValue < 65 || asciiValue > 90) {
                  ciphertext += pChar;
                  continue;
            }

            // ASCII dari plaintext akan dikurangi 65
            // diubah ke bentuk Caesar cipher
            int basicPlaintextValue = ((int) pChar) - 65;

            // mengubah karakter key menjadi ASCII
            char kChar = key.charAt(kIndex);


            // ASCII dari key dikurangi 65 diubah ke bentuk
            // caesar code
            int basicKeyValue = ((int) kChar) - 65;

            // menjumlahkan plaintext dan key kemudian mod
            // 26
            int cText = ((basicPlaintextValue +
            basicKeyValue) % 26);
```

```
            // setiap karakter hasil penjumlahan akan
            // ditambah dgn 65, untuk mengembalikan ke
            // bentuk ASCII
            char cChar = (char) (cText + 65);
                  ciphertext += cChar;
                  kIndex++;

            if (kIndex == key.length())
                  kIndex = 0;
            }

            return ciphertext;
```

b. Proses enkripsi pada *Base64*

Pada proses ini, *ciphertext* yang dihasilkan oleh *Vigenere Cipher* akan diubah pada proses `String encode(byte[] raw)` dimana setiap karakter dari *ciphertext* dikelompokkan menjadi 3 karakter untuk setiap kelompok. Kemudian setiap karakter akan diubah kedalam bentuk kode ASCII dan diproses menjadi biner, dalam bentuk 8 bit diubah ke bentuk 6 bit pada proses `encodeBlock(byte[] raw, int offset)`, setelah itu hasil pembentukan 6 bit akan diubah ke dalam bentuk kode *Base64*. Jika pada proses pengelompokan tersebut tidak sesuai (karakter kurang dari 3) maka akan menghasilkan *padding* dan hasilnya akan diubah ke dalam bentuk '='.

```
// proses encode base64
public static String encode(byte[] raw) {

        StringBuffer encoded = new StringBuffer();

        //memilah ciphertext menjadi 3 karakter
        for (int i = 0; i < raw.length; i += 3) {
              encoded.append(encodeBlock(raw, i));
        }
        //mengembalikan ke nilai string
        return encoded.toString();
  }
```

```
//proses pengubahan ke bentuk biner
protected static char[] encodeBlock(byte[] raw, int
      offset) {

      int block = 0;
      int slack = raw.length - offset - 1;
      int end = (slack >= 2) ? 2 : slack;

      //pengubahan karakter ke bentuk biner 8 bit
      for (int i = 0; i <= end; i++) {
            byte b = raw[offset + i];
            int neuter = (b < 0) ? b + 256 : b;
            block += neuter << (8 * (2 - i));
      }

      char[] base64 = new char[4];
      for (int i = 0; i < 4; i++) {

      //pengubahan dari 8 bit ke bentuk 6 bit
      int sixbit = (block >>> (6 * (3 - i))) & 0x3f;

      //pengubahan hasil dari sixbit ke dalam base64 code
      base64[i] = getChar(sixbit);
}


      //proses pengubahan padding null ke bentuk '='
      if (slack < 1)
      base64[2] = '=';
      if (slack < 2)
      base64[3] = '=';

      return base64;
}

//proses pembentukan base64 code
protected static char getChar(int sixBit) {
      if (sixBit >= 0 && sixBit <= 25)
      return (char) ('A' + sixBit);
      if (sixBit >= 26 && sixBit <= 51)
      return (char) ('a' + (sixBit - 26));
      if (sixBit >= 52 && sixBit <= 61)
      return (char) ('0' + (sixBit - 52));
      if (sixBit == 62)
      return '+';
      if (sixBit == 63)
      return '/';
      return '?';
}
```

4.2.2 Tahap Dekripsi

Dalam proses dekripsi pesan, sama halnya dengan proses enkripsi pesan memiliki dua tahapan dekripsi, yaitu proses dekripsi Base64 dan Vigenere cipher. Berikut penjelasan proses dan pseudocodenya :

a. Proses dekripsi pada *Base64*

Pada proses dekripsi *Base64*, dimana pesan yang diterima akan diolah menjadi bentuk string. Proses dari dekripsi ini merupakan kebalikan proses dari enkripsi Base64. Dimana teks yang berbentuk string diubah ke dalam bentuk byte pada proses `byte[] decode(String base64)`, setelah itu teks tersebut akan diubah ke dalam bentuk 6 bit biner, kemudian akan dibentuk dalam 8 bit. Setelah proses ini pengubahan biner ke bentuk kode ASCII dan dikembalikan ke bentuk string.

```
//proses decode base64
public static byte[] decode(String base64) {
      int pad = 0;

      //mengecek panjang karakter dan padding
      for (int i = base64.length() - 1; base64.charAt(i)==
      '='; i--) pad++;

      int length = base64.length() * 6 / 8 - pad;

      byte[] raw = new byte[length];

      int rawIndex = 0;

      //proses pengembalian dari base64 code menjadi 6 bit
      for (int i = 0; i < base64.length(); i += 4) {
            int block = (getValue(base64.charAt(i)) << 18)
                  + (getValue(base64.charAt(i + 1)) << 12)
                  + (getValue(base64.charAt(i + 2)) << 6)
                  + (getValue(base64.charAt(i + 3)));
```

```
                    // proses pengembalian dari 6 bit
                    // menjadi 8 bit
                    for (int j = 0; j < 3 && rawIndex + j <
                    raw.length; j++)
                    raw[rawIndex + j] = (byte) ((block >> (8
                    * (2 - j))) & 0xff);
                    rawIndex += 3;
              }
              return raw;
        }
        //base64 code
        protected static int getValue(char c) {
              if (c >= 'A' && c <= 'Z')
                    return c - 'A';
              if (c >= 'a' && c <= 'z')
                    return c - 'a' + 26;
              if (c >= '0' && c <= '9')
                    return c - '0' + 52;
              if (c == '+')
                    return 62;
              if (c == '/')
                    return 63;
              if (c == '=')
                    return 0;
              return -1;
        }
```

b. Proses dekripsi pada *Vigenere Cipher*

Pada proses dekripsi *Vigenere Cipher*, hasil dari dekripsi *Base64* akan diolah menjadi *plaintext* (teks-asli). Pada proses `decryptVigenere(String ciphertext, String key)`, *ciphertext* dan *key* akan diubah ke dalam bentuk kode ASCII kemudian akan dikurangi 65 untuk mengubah ke dalam bentuk Caesar kode, setelah itu akan diproses dengan `((basicCiphertextValue - basicKeyValue) % 26)`, hasil dari penjumlahan tersebut akan diubah kedalam bentuk kode ASCII dengan menambahkan 65 yang kemudian akan dikembalikan dalam bentuk string dan menjadi *plaintext*.

```
public static String decryptVigenere(String ciphertext,
String key) {
	// mengubah ciphertext dan key ke bentuk kapital
	ciphertext = ciphertext.toUpperCase();
	key = key.toUpperCase();
	String plaintext = "";
	int keyIndex = 0;

	// hitung panjang karakter ciphertext dengan index
	for (int ctextIndex = 0; ctextIndex <
		ciphertext.length(); ctextIndex++) {
		// ubah setiap karakter ke bntuk ASCII
		char cChar = ciphertext.charAt(ctextIndex);
		int asciiVal = (int) cChar;

		// jika ada spasi maka proses berlanjut
		if (cChar == ' ') {
			plaintext += cChar;
			continue;
		}
	// jika karakter sama dengan kondisi maka proses
	// berlanjut
	if (asciiVal < 65 || asciiVal > 90) {
		plaintext += cChar;
		continue;
	}
	// mengurangi bentuk ASCII tiap karakter dengan 65
	int basicCiphertextValue = ((int) cChar) - 65;
	// mengubah key ke bentuk ASCII
	char kChar = key.charAt(keyIndex);
	// mengurangi bentuk ASCII tiap karakter dengan 65
	int basicKeyValue = ((int) kChar) - 65;
	// menjumlahkan dgn rumus (ciphertext-key)%26
	int pText = ((basicCiphertextValue - basicKeyValue)
		% 26);
	// jika nilai pText kurang dari 0, maka akan
	// ditambah dengan 26
	if (pText < 0) {
		pText = pText + 26;
	}
	// menambahkan setiap karakter dengan 65 dan
	// mengubah ke bentuk semula
	char pChar = (char) (pText + 65);
	plaintext += pChar;

	keyIndex++;
	if (keyIndex == key.length())
	keyIndex = 0;
}


return plaintext.toLowerCase();
```

4.3 Implementasi Antarmuka

Pada antarmuka yang dibangun terdapat beberapa form, yaitu form menu, form tulis pesan, form pesan masuk, form buka pesan, form atur kunci, form bantuan dan form info. Adapun penjabaran form aplikasi enkripsi SMS sebagai berikut:

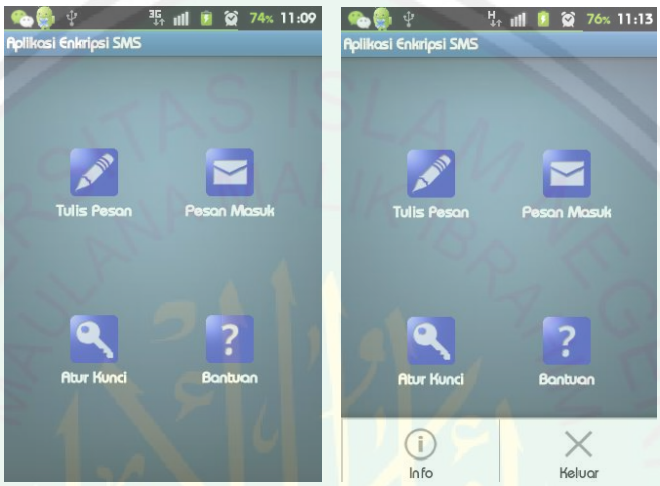
4.3.1 *Splash Screen*

Splash screen adalah tampilan berupa gambar atau teks yang akan muncul ketika aplikasi pertama kali dijalankan. Untuk menampilkannya harus diatur berapa lama waktu yang dibutuhkan untuk muncul di layar. Setelah waktu yang ditentukan maka layar akan menampilkan interface selanjutnya. Splashscreen ini menunjukkan nama dari aplikasi yang sedang dijalankan. Pada proses ini, terdapat proses pengambilan data telepon pada kontak manager *mobile devices*.

Gambar 4.2 *Splash screen aplikasi*

4.3.2 Form Menu

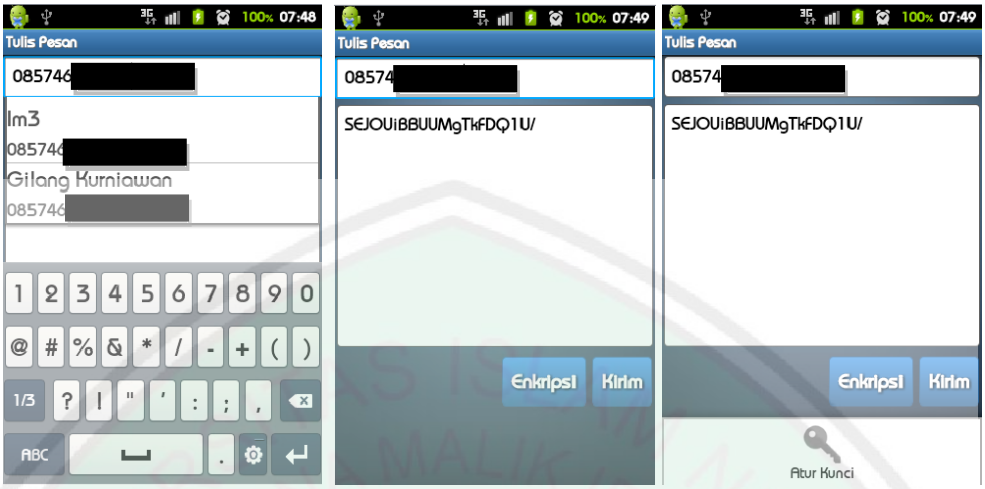
Pada form ini, berisi menu-menu aplikasi enkripsi SMS diantaranya, Tulis Pesan, Pesan Masuk, Atur Pesan, Bantuan dan Info.



Gambar 4.3 *Tampilan menu aplikasi enkripsi SMS*

4.3.3 Form Tulis Pesan

Pada form ini berisi kolom-kolom dan tombol yang digunakan untuk mengirim pesan, diantaranya kolom no. tujuan, kolom isi pesan, tombol enkripsi, tombol kirim dan menu atur kunci.



Gambar 4.4 *Tampilan tulis pesan*

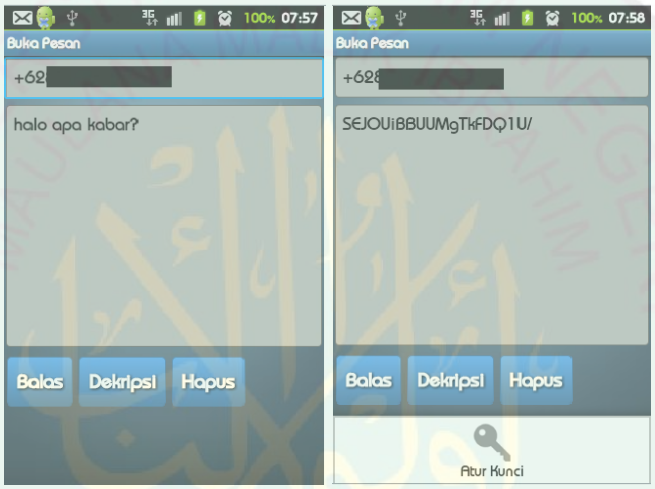
4.3.4 Form Pesan Masuk

Pada form Pesan Masuk berisi tentang pesan-pesan yang telah diterima, dimana menampilkan no. pengirim, waktu penerimaan pesan, dan isi pesan, serta berisi menu, yaitu menu Tulis pesan baru dan Hapus semua pesan.

Gambar 4.5 *Tampilan pesan masuk*

4.3.5 Form Buka Pesan

Pada form ini berisi kolom no. pengirim pesan, kolom pesan yang dikirimkan, tombol Balas, tombol Dekripsi, tombol Hapus dan menu Atur Kunci. Menu atur kunci ini digunakan untuk mengatur kunci yang akan digunakan pada saat akan men-dekripsi pesan yang disesuaikan dengan kunci pengirim pesan.



Gambar 4.6 *Tampilan buka pesan*

4.3.6 Form Atur Pesan

Pada form ini berisi kolom kunci dan tombol Simpan, dimana kunci ini akan digunakan pada saat proses enkripsi dan dekripsi pesan. Secara default kunci ini berisi "ABCD", kunci ini bisa disesuaikan dengan keinginan pengguna. Pengisian kunci ini hanya berkarakter text.

Gambar 4.7 *Tampilan atur pesan*

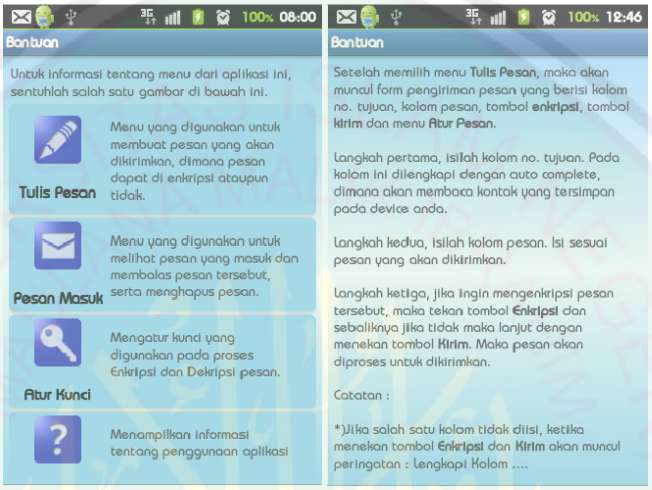
4.3.7 Form Info

Pada form Info berisi tentang informasi aplikasi Enkripsi SMS dan pembuat aplikasi.

Gambar 4.8 *Tampilan info*

4.3.8 Form Bantuan

Pada form ini berisi tentang informasi bantuan untuk menggunakan aplikasi enkripsis SMS.



Gambar 4.9 *Tampilan bantuan*

4.4 Evaluasi dan Analisis Hasil Pengujian

Setelah aplikasi sudah dibuat, maka aplikasi akan di test dengan beberapa pengujian, diantaranya pengujian enkripsi dan dekripsi pesan dan pengujian perangkat lunak. Dan mengevaluasi dan menganalisis hasil dari kuesioner uji kelayakan aplikasi.

4.4.1 Pengujian Enkripsi dan Dekripsi Pesan

Pengujian ini dilakukan untuk mengetahui kemampuan aplikasi ini dalam mengenkripsi dan mendekripsi pesan. Berikut tabel hasil dari pengujian:

Tabel 4.1 *Pengujian enkripsi pesan dan pengiriman pesan*

No	Isi pesan	Panjang Karater	Kunci	Hasil	Keterangan
1	Halo apa kabar?	15	ABCD	SEJOUiBBUUMgTkFDQ1U/	Enkripsi berjalan, pesan terkirim
2	Halo apa kabar?	15	1234	ODI+QiAxQTMgPjEzM0U/	Enkripsi berjalan, pesan terkirim
3	12345!@#$%	10	ABCD	MTIzNDUhQCMkJQ==	Enkripsi berjalan, pesan terkirim
4	Baik, kamu gimana kabarnya?	27	A1B2	QjFKPCwgSzFORiBHOU 4yTjEgTDJCMVM/WTE/	Enkripsi berjalan, pesan terkirim
5	Aplikasi enkripsi SMS adalah aplikasi yang digunakan untuk keamanan dalam menggunakan fitur layanan SMS sehingga data yang dikirim maupun yang diterima terjaga privasinya.	171	ABCD	QVFOTEtCVUwgRU9NV UlRVUwgU05VIEREQk 5ESCBCUk9JTENWS SBaQ1FHIEVLSlVPQ 05BTyBXUVRWTSBOR UJPRE5CUCBHQU1DU CBNRlBKR1ZQREtCUC BJSVVXVSBMQkFETk JQIFZNVCBVSEhKU EpHQiBGRFRCIEFET kggRkxLSlRMTSBOQ1h QVlAgQkFPSSBHSV VHVUlOQyBXRVNM REdCIFJVSVdDVklPQUQu	Enkripsi berjalan, tetapi pesan gagal dikirimkan karena panjang pesan melebihi 160 karakter

Berdasarkan pengujian enkripsi dan dekripsi pada tabel 4.1 dan 4.2, menyatakan bahwa kunci yang harus digunakan adalah berkarakter alphabet, dimana jika menggunakan kunci yang berkarakter itu, maka hasilnya tidak akan sesuai dengan pesan asli. Panjang karakter yang dikirim kurang dari 160 karakter. Jika melebihi dari 160 karakter, maka pengiriman pesan akan gagal.

Tabel 4.2 *Pengujian dekripsi pesan*

No	Isi pesan	Kunci	Hasil	Keterangan
1	SEJOUiBBUUMgTkFDQ1U/	ABCD	Halo apa kabar?	Dekripsi berjalan
2	ODI+QiAxQTMgPjEzM0U/	1234	82>r 1p3 >133s?	Dekripsi berjalan, tetapi hasil tidak sesuai dengan pesan asli
3	MTIzNDUhQCMkJQ==	ABCD	12345!@#$%	Dekripsi berjalan
4	QjFKPCwgSzFORiBHOU 4yTjEgTDJCMVM/WTE/	A1B2	b1z<, j1cf w9m2c1 l2r1r?n1?	Dekripsi berjalan, tetapi hasil tidak sesuai dengan pesan asli
5	Halo apa kabar?	ABCD	-	Dekripsi pesan gagal

4.4.2 Pengujian Kekuatan Enkripsi Vigenere Cipher dengan Analisis Frekuensi

Pengujian ini dilakukan untuk mengetahui seberapa kuat metode yang digunakan dalam proses enkripsi, pengujian dilakukan pada metode Vigenere Cipher dimana akan memecahkan enkripsi dengan menggunakan metode Analisis Frekuensi.

Berikut langkah-langkah yang digunakan pada metode analisis frekuensi, yaitu:

a. Misalkan panjang kunci yang sudah dideduksi adalah *n*. Setiap huruf kelipatan ke-*n* pasti dienkripsi dengan huruf kunci yang sama. mengelompokkan setiap huruf ke-*n* bersama sama sehingga kriptanalis

memiliki *n* buah "pesan", yang telah dienkripsi menggunakan metode vigenere cipher.

b. Tiap-tiap pesan dari hasil langkah pertama, dapat dipecahkan dengan teknik analisis frekuensi.

c. Dari hasil langkah kedua, kriptanalis dapat menyusun huruf-huruf kunci. Atau, kriptanalis dapat menerka kata yang membantu untuk memecahkan *ciphertext*.

Contoh :

LJVBQ STENZ LQMED **LJV**MA MPKAU FAVAT **LJV**DA YYVNF JQLNP **LJV**HK VTRNF **LJV**CM LKETA **LJV**HU YJVSF KRFTT WEFUX VHZNP

Kriptogram yang berulang adalah **LJV**.

Jarak **LJV** ke-1 dengan **LJV** ke-2 = 15

Jarak **LJV** ke-2 dengan **LJV** ke-3 = 15

Jarak **LJV** ke-3 dengan **LJV** ke-4 = 15

Jarak **LJV** ke-4 dengan **LJV** ke-5 = 10

Jarak **LJV** ke-5 dengan **LJV** ke-6 = 10

Faktor pembagi 15 = {3, 5, 15}

Faktor pembagi 10 = {2, 5, 10}

Irisan kedua himpunan ini = 5. Jadi, panjang kunci diperkirakan = 5

Mengelompokkan "pesan" setiap kelipatan ke-5, dimulai dari huruf *ciphertext* pertama, kedua, dan seterusnya.

Tabel 4.3 *Pengelompokkan pesan setiap kelipatan ke-5*

Kelompok	Pesan	Huruf paling sering muncul
1	LSLLM FLYHL VLLLY KWV	L
2	JTQJP AJYQJ TJKJJ REH	J
3	VNMVK VVVLV RVEVV FFZ	V
4	BEEMA ADNNH NCTHS TUN	N
5	QZDAU TAFPK FMAUF TXP	A

Dalam Bahasa Inggris, 10 huruf yang paling sering muncul adalah E, T, A, O, I, N, S, H, R, dan D. Triplet yang paling sering muncul adalah THE. Karena **LJV** paling sering muncul di dalam ciphertext, maka dari 10 huruf tersebut semua memungkinkan kata 3-huruf dibentuk dan kata yang cocok untuk **LJV** adalah THE. Jadi, dapat diterka bahwa LJV mungkin adalah THE.

Setelah itu, memetakan huruf plaintext dengan ciphertext dan huruf-huruf kuncinya.

Tabel 4.4 *Pemetaan huruf plaintext dengan ciphertext dan kunci*

Kelompok	Huruf plaintext	Huruf ciphertext	Huruf Kunci
1	T	L	S (= 18)
2	H	J	C (= 2)
3	E	V	R(= 17)
4	N	N	A(= 0)
5	O	A	M(= 12)

Dengan menggunakan kunci “SCRAM” ciphertext berhasil didekripsi menjadi:

THEBE ARWEN TOVER THEMO UNTAI NYEAH

THEDO GWENT ROUND THEHY DRANT THECA

TINTO THEHI GHEST SPOTH ECOUL DFIND

Atau dalam kalimat yang lebih jelas adalah:

THE BEAR WENT OVER THE MOUNTAIN YEAH

THE DOG WENT ROUND THE HYDRANT THE CAT INTO

THE HIGHEST SPOT HE COULD FIND

Berdasarkan hasil pengujian tersebut, metode *Vigenere Cipher* masih memiliki kekurangan, karena dapat dipecahkan dengan menggunakan metode analisis data. (Kurniawan, 2012)

4.4.3 Pengujian Perangkat Lunak

Setelah program dapat bekerja dengan baik pada komputer dan *emulator*, maka selanjutnya program akan diuji coba pada *smartphone* yang sebenarnya. Hal ini bertujuan untuk mengetahui apakah program juga dapat bekerja pada *smartphone* yang sebenarnya. Selain itu juga untuk hasil tampilan aplikasi pada beberapa *mobile devices* yang memiliki perbedaan resolusi layar.

Aplikasi Enkripsi SMS telah diuji pada beberapa *mobile devices* yang berbasis Android, untuk lebih jelasnya pada tabel 4.3.

Tabel 4.5 *Pengujian aplikasi dan tampilan pada mobile devices*

No.	Tipe	Resolusi Layar	Versi Android	Keterangan
1.	Samsung Galaxy Gio	HVGA (320x480 px)	2.3.6	Berjalan dengan baik
2.	Samsung Galaxy Mini	QVGA (240x320 px)	2.3.6	Berjalan dengan baik
3.	Samsung Galaxy Mini 2	HVGA (320x480 px)	2.3.6	Berjalan dengan baik
4.	Samsung Galaxy Young	QVGA (320x480 px)	2.3.5	Berjalan dengan baik
5.	Samsung Galaxy Young Duos	QVGA (320x480 px)	2.3	Berjalan dengan baik
6.	Samsung Galaxy Ace Plus	HVGA (320x480 px)	2.3	Berjalan dengan baik
7.	Samsung Galaxy Wonder	WVGA (480x800 px)	2.3	Berjalan dengan baik
8.	Sony Xperia Go	HVGA (320x480 px)	2.3.5, 4.0.4	Berjalan dengan baik
9.	Sony Xperia J	FWVGA (480x854 px)	4.0.4	Berjalan dengan baik
10.	Sony Ericson WT19I Live With Walkman	HVGA (320x480 px)	4.0.4	Berjalan dengan baik
11.	HTC Evo 3D	qHD (540 x 960 px)	4.0.4	Berjalan dengan baik
12.	New Smartfren Andro Tab	WSVGA (1024x600 px)	4.0.4	Berjalan dengan baik, *interfaces* aplikasi kurang sesuai dengan resolusi *smartphone*

Berdasarkan pada tabel 4.3, hasil pengujian menunjukkan bahwa aplikasi enkripsi sms dapat berjalan pada beberapa *devices* yang memiliki resolusi layar QVGA, HVGA, WVGA, FWVGA, qHD dengan tampilan optimal. Namun pada *mobile devices* yang memiliki resolusi layar WSVGA masih mengalami masalah pada *icon* menu, dikarenakan perbedaan resolusi layar dengan *device* yang digunakan pada pengembangan aplikasi.

4.4.4 Pengujian Aplikasi dengan Penyadapan Pesan

Pengujian ini menggunakan salah satu aplikasi penyadap SMS pad android, yang bernama SMS Watcher Lite. pengujian ini menggunakan tiga buah handphone, dimana pada handphone 1 (A) sebagai pengirim, handphone 2 (B) sebagai penerima, dan handphone 3 (C) sebagai penyadap. Dan aplikasi tersebut di install pada handphone 2. berikut hasil uji coba dengan pengiriman pesan pertama dengan menggunakan aplikasi pesan default dan pengiriman pesan kedua menggunakan aplikasi enkripsi SMS.

Tabel 4.6 *Pengujian penyadapan pesan*

Pengujian ke-	HP 1	HP 2	HP 3
1	Halo	Halo	Halo
2	SEJOUg==	SEJOUg==	SEJOUg==

Berdasarkan pengujian penyadapan sms dengan menggunakan aplikasi SMS Watcher Lite, pesan yang dikirim dengan aplikasi enkripsi sms dengan pengenkripsian pesan masih dapat tersadap, tetapi pesan yang diterima oleh penyadap dalam keadaan terenkripsi, sehingga pesan tersebut masih terjaga keamanannya.

4.4.5 Hasil Kuesioner Uji Kelayakan Produk

Hasil perolehan perhitungan dari 50 koresponden, yang diambil dari masyarakat umum.

Tabel 4.7 *Rekapitulisasi hasil kuesioner*

No	Pernyataan	Jumlah Penilaian Responden				
		SB	B	C	K	SK
1	Ketertarikan dengan Aplikasi	8	31	11	-	-
2	Tampilan Aplikasi	1	27	21	1	-
3	Kesesuaian desain warna	3	23	20	4	-
4	Kelengkapan fitur	2	12	30	6	-
5	Kemudahan penggunaan fitur	12	22	14	2	-
6	Penggunaan bahasa	6	29	14	1	-
7	Manfaat Aplikasi	7	24	18	1	-
8	Pengembangan Aplikasi	15	34	1	-	-
9	Publikasi Aplikasi pada masyarakat	8	33	9	-	-
10	Tanggapan pengguna secara keseluruhan	2	37	11	-	-

Berdasarkan hasil rekapitulasi kuesioner pada tabel 4.4, maka diperoleh hasil untuk masing-masing pertanyaan.

Data interval tersebut, dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden dan menghitung prosentase total hasil jawaban dengan membagi total jawaban responden dengan total skor ideal (dengan nilai 250, jika semua jawaban mendapat nilai SB). Berdasarkan skor yang telah ditetapkan dapat dihitung, sebagai berikut:

1. Ketertarikkan dengan Aplikasi

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
8	5	**40**	31	4	**124**	11	3	**33**	0	2	**0**	0	1	**0**

2. Tampilan Aplikasi

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
1	5	**5**	27	4	**108**	21	3	**63**	1	2	**1**	0	1	**0**

3. Kesesuaian desain warna

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
3	5	**15**	23	4	**92**	20	3	**60**	4	2	**8**	0	1	**0**

4. Kelengkapan fitur

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
2	5	**40**	12	4	**124**	30	3	**33**	6	2	**12**	0	1	**0**

5. Kemudahan penggunaan fitur

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
12	5	**60**	22	4	**88**	14	3	**42**	2	2	**4**	0	1	**0**

6. Penggunaan bahasa

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
6	5	**30**	29	4	**116**	14	3	**42**	1	2	**2**	0	1	**0**

7. Manfaat Aplikasi

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
7	5	**35**	24	4	**96**	18	3	**54**	1	2	**2**	0	1	**0**

8. Pengembangan Aplikasi

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
15	5	**75**	34	4	**136**	1	3	**3**	0	2	**0**	0	1	**0**

9. Publikasi Aplikasi pada masyarakat

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
8	5	**40**	33	4	**132**	9	3	**27**	0	2	**0**	0	1	**0**

10. Tanggapan pengguna secara keseluruhan

SB			B			C			K			SK		
∑SB (t1)	Skor (t2)	Total (t1*t2)	∑B (t1)	Skor (t2)	Total (t1*t2)	∑c (t1)	Skor (t2)	Total (t1*t2)	∑K (t1)	Skor (t2)	Total (t1*t2)	∑SK (t1)	Skor (t2)	Total (t1*t2)
2	5	**10**	37	4	**148**	11	3	**33**	0	2	**0**	0	1	**0**

Data tersebut merupakan hasil dari perhitungan kuesioner dengan menggunakan skala *Likert*. Pada tabel 4.5 merupakan hasil perhitungan dengan skala *Likert*.

Tabel 4.8 *Hasil perhitungan kuesioner dengan skala Likert*

No	Pernyataan	Hasil Perhitungan					Prosentase
		SB	B	C	K	SK	
1	Ketertarikan dengan Aplikasi	40	124	33	0	0	78.8%
2	Tampilan Aplikasi	5	108	63	2	0	69.2%
3	Kesesuaian desain warna	15	92	60	8	-	70%
4	Kelengkapan fitur	10	48	90	12	-	64%
5	Kemudahan penggunaan fitur	60	88	42	4	-	77.6%
6	Penggunaan bahasa	30	116	42	2	-	76%
7	Manfaat Aplikasi	35	96	54	2	-	74.8%
8	Pengembangan Aplikasi	75	136	3	-	-	85.6%
9	Publikasi Aplikasi pada masyarakat	40	132	27	-	-	79.6%
10	Tanggapan pengguna secara keseluruhan	10	148	33	-	-	76.4%
Total hasil							**752**
Rata-rata							**75.2%**

Berdasarkan tabel 4.5, hasil tersebut akan dicari nilai rata-rata prosentase hasil kuesioner adalah 75.2%. Berdasarkan hasil dari analisis kuesioner uji kelayakan aplikasi dari jawaban 50 responden, 75.2% menyatakan bahwa aplikasi enkripsi SMS ini layak berdasarkan tingkatan uji kelayakan, sehingga dapat digunakan dan dipublikasikan untuk masyarakat umum. Diharapkan dapat membantu dalam mengamankan data SMS pengguna aplikasi enkripsi SMS. Dan terciptanya keamanan dalam menggunakan layanan SMS pada *mobile devices* berbasis Android.

4.5 Hubungan Kriptografi dengan Perspektif Agama Islam

Kriptografi adalah seni untuk mengamankan dan merahasiakan informasi. Keamanan merupakan aspek yang penting dari ilmu kriptografi sehingga informasi yang dirahasiakan tetap aman dari orang-orang yang tidak diperkenankan untuk mengetahuinya. Dalam kriptografi terdapat proses-proses yang harus dilakukan secara berurutan, seperti algoritma kriptografi memiliki beberapa tahapan, yaitu enkripsi, dekripsi dan kunci.

Dalam pandangan agama Islam, mengerjakan suatu kewajiban pasti dilaksanakan secara teratur atau berurutan. Misalnya saja sewaktu melaksanakan sholat, wudhu bahkan ketika melakukan kegiatan sehari-hari dilakukan dengan berurutan. Sama halnya kriptografi dan ajaran agama Islam, melakukan sesuatu pasti ada urutannya, dari memulai sampai mengakhiri. Semua ada aturan yang harus dilakukan untuk mendapatkan hasil yang sesuai dengan keinginan.

Hasil yang diperoleh dari skripsi ini adalah perangkat lunak yang dibangun telah berhasil mengenkripsi pesan sesuai dengan aturan pada kriptografi. Sehingga dapat digunakan untuk mengamankan pesan. Hal ini sudah memenuhi keamanan yang juga menjadi aspek dari kriptografi. Kaitannya dengan keislaman tentang keamanan dan kesesuaian dengan urutan yang telah ditetapkan. Dan semua itu merupakan pemberian dari Allah SWT, dengan keamanan dan mengikuti aturan maka akan tercapai segala kemashlahatan dan kebaikan yang dibutuhkan oleh manusia.

BAB V

PENUTUP

5.1. Kesimpulan

Dari pendefinisian masalah serta analisis dan pembuatan aplikasi ini, dapat diambil kesimpulan bahwa:

1. Aplikasi enkripsi sms menggunakan metode *Vigenere Cipher* dan *Base64* berhasil diimplementasikan pada *mobile devices* yang berbasis Android.
2. Aplikasi enkripsi sms dapat diimplementasikan guna mengamankan data sms pengguna layanan sms.
3. Berdasarkan uji coba enkripsi dan dekripsi, kunci yang digunakan harus berkarakter *alphabet*. Jika kunci yang digunakan berkarakter selain itu, proses enkripsi tetap berjalan tetapi hasil dari dekripsi berbeda dengan teks aslinya.
4. Berdasarkan uji coba kekuatan enkripsi *Vigenere Cipher*, *ciphertext* dari *Vigenere Cipher* dapat dipecahkan dengan menggunakan metode analisis frekuensi.
5. Berdasarkan uji coba aplikasi dan tampilan aplikasi pada beberapa *mobile devices*, aplikasi dapat bekerja dan tampilan dapat optimal pada *devices* yang menggunakan resolusi layar QVGA, HVGA, WVGA, FWVGA, qHD.
6. Berdasarkan uji coba penyadapan pesan yang pengirimannya menggunakan Aplikasi Enkripsi SMS dengan pesan yang terenkripsi, pesan tersebut dapat tersadap tetapi keadaan pesan dalam keadaan terenkripsi, sehingga pesan tersebut masih terjaga keasliannya atau privasinya.

7. Berdasarkan hasil dari analisis kuesioner uji kelayakan aplikasi dari jawaban 50 responden, 75.2% menyatakan bahwa aplikasi enkripsi SMS ini layak digunakan dan dipublikasikan untuk masyarakat umum. Meskipun demikian ada beberapa kendala dalam pengembangan aplikasi enkripsi sms diantaranya adalah keterbatasan sumberdaya dan keragaman jenis *smartphone* yang digunakan.

5.2. Saran

Saran yang disampaikan sebagai pengembangan dari aplikasi ini untuk penelitian selanjutnya yaitu dapat di tambah dengan beberapa fitur lainnya, seperti pesan terkirim, filter pesan, pengiriman pesan yang melebihi 160 karakter, penggunaan kunci dengan menggunakan karakter *alphanumeric* dan lain-lain. Penggunaan metode lain yang lebih efektif juga bisa diteliti untuk mendapatkan keamanan yang lebih baik.

DAFTAR PUSTAKA

Ahmad, Mohammad A, Imad Fakhri Al Shaikhli, Hanady Mohammad Ahmad. 2012 *Protection of the Texts Using Base64 and MD5*. Journal of Advanced Computer Science and Technology Research 2

Ariyus, Dony. 2006. *Kriptografi: Keamanan Data dan Komunikasi*. Yogyakarta: Graha Ilmu.

Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta: Penerbit Andi.

Calhoun, David. 2011. *When to base64 encode images (and when not to)*. [*online*]. [*Accessed 11 Februari 2013*]. *Available from World Wide Web*: <http://davidbcalhoun.com/2011/when-to-base64-encode-images-and-when-not-to>

Dzulqarnain. 2009. *Islam adalah Penegak Keamanan*. [*online*]. [*Accessed 22 Januari 2013*]. *Available from World Wide Web*: <http://jihadbukankenistaan.com/jalan-petunjuk/islam-adalah-penegak-keamanan.html>

Fahmi, Husni, Haret Faidah. 2006. *Tutorial Kriptografi Klasik dan Penerapannya dalam Visual Basic .Net*. IlmuKomputer.com

Hukum Online. 2007. *Seputar Penyadapan SMS Wartawan Tempo*. [*online*]. [*Accessed 30 Januari 2013*]. *Available from World Wide Web*: <http://www.hukumonline.com/berita/baca/hol17609/seputar-kasus-penyadapan-wartawan-itempoi>

Kester, Quist-Aphetsi. 2012. *A cryptosystem based on Vigenère cipher with varying key*. International Journal of Advanced Research in Computer Engineering & Technology.

Kurniawan, Agus. 2008. *Konsep dan Implementasi Cryptography Dengan .NET*. Jakarta: Dian Rakyat.

Kurniawan, Rakhmat. 2012. *Blog Dosen Stimik Triguna Dharma : Computer Security & Cryptography*. [*online*]. [*Accessed 13 April 2013*]. *Available from World Wide Web*:< http://rakhmatkurniawan.trigunadharma.ac.id/wp-content/uploads/2012/06/Bab-6-Metode-Kasiski_.pdf>

Menezes, Alferd J, Paul C. van Oorschot, Scott A. Vanstone. 1996. *Handbook of Applied Cryptography*. UK: CRC Press.

Rathke, David. 2012. *Introduction to SMS and SMS Messaging Services*. Illinois State University. [*online*]. [*Accessed 20 Maret 2013*]. *Avaible from World Wide Web*: <http://www.itk.ilstu.edu/staff/drathke/277web/WebContent/reading/SMSoverview2.pdf>

Safaat H, Nazruddin. 2012. *Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.

Sugiyono. 2010. *Metode Penelitian Bisnis (Pendekatan Kuantatif, Kualitatif, dan R&D)*. Bandung: Alfabeta.

Syafaat, Nazruddin. 2010. *Android : Arsitektur Android*. [*online*]. [*Accessed 26 Maret 2013*]. *Available from World Wide Web*: <http://nazruddinsyafaat.blogspot.com/p/android-arsitekrur-android_23.html>

TribunNews. 2013. *Kapolresta Malang Bicara Tentang Penyadapan SMS Yuni Shara*. [*online*]. [*Accessed 12 April 2013*]. *Available from World Wide Web:*<http://www.tribunnews.com/2013/03/07/kapolresta-malang-bicara-tentang-penyadapan-sms-yuni-shara>

Yayasan Indonesia Membaca. 2009. *Al-Quran online Indonesia : Tafsir Depag RI*. [*online*]. [*Accessed 22 Januari 2013*]. *Available from World Wide Web*: <http://quran.bacalah.net/content/surat/index.php>

Lampiran

Kuesioner Uji Kelayakan Produk

Bapak/Ibu/Saudara yang terhormat,

Demi kepentingan peningkatan dan uji kelayakan aplikasi enkripsi sms, kami mohon bantuan untuk memberikan informasi. Semua keterangan dan jawaban yang diperoleh semata-mata hanya untuk kepentingan penelitian dan dijamin kerahasiaannya. Atas bantuan Bapak/Ibu/Saudara peneliti mengucapkan terima kasih.

Keterangan:

SB	= Sangat baik	K	= Buruk/Kurang
B	= Baik	SK	= Sangat buruk/kurang
C	= Cukup/Biasa		

Nama : ……..

No	Pernyataan	Jawaban				
		SB	S	C	K	SK
1	Apakah anda tertarik dengan aplikasi enkripsi SMS					
2	Bagaimanakah desain tampilan (User Interfaces) pada aplikasi enkripsi SMS					
3	Bagaimanakah kesesuaian desain warna pada aplikasi enkripsi SMS					
4	Bagaimanakah kelengkapan fitur yang terdapat pada aplikasi enkripsi SMS					
5	Bagaimanakah kemudahan penggunaan fitur-fitur pada aplikasi enkripsi SMS					
6	Bagaimana penggunaan bahasa pada aplikasi enkripsi SMS					
7	Apakah aplikasi enkripsi SMS bermanfaat dalam penggunaan sehari-hari					
8	Apakah aplikasi enkripsi SMS perlu dikembangkan lebih lanjut					
9	Apakah aplikasi enkripsi SMS ini layak dipublikasikan di masyarakat					
10	Secara keseluruhan, baik dari segi fitur maupun interface, bagaimanakah tanggapan anda terhadap aplikasi enkripsi SMS					

Hasil Kuesioner Uji Kelayakan Produk

Keterangan:

SB = Sangat baik
B = Baik
C = Biasa/Cukup
K = Buruk/Kurang
SK = Sangat buruk/kurang
1-10 = Item pernyataan dalam kuesioner

No	Responden	Jawaban Kuesioner									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	Fitriana Nelvi	B	B	B	B	B	B	B	SB	B	B
2	Dini	C	C	C	C	C	C	B	B	B	C
3	Nike P.	B	B	C	C	SB	C	B	B	B	B
4	Dwi Nilam Hadi	C	B	C	C	C	C	SB	SB	SB	B
5	Anik Oktavia P. S.	B	B	C	C	B	B	B	B	B	B
6	Aris Agustina	B	C	B	K	C	C	B	B	B	B
7	Septya Luckyana Putri	B	B	B	C	C	B	A	B	B	B
8	Indra Wahyudi	B	SB	C	C	K	B	B	SB	B	B
9	Ricky Ardhian Rukmana	SB	B	B	C	SB	B	C	B	B	B
10	Ahmad Hariadi	B	B	B	C	B	B	C	SB	SB	B
11	Whelly Yulianto	C	C	C	C	B	B	C	B	B	B
12	Zaki Mubarok	B	B	SB	C	B	B	SB	SB	B	B
13	Amadeuz Ezrafel	B	C	B	C	SB	B	B	SB	SB	B
14	Dewi Setyaningrum	SB	B	B	C	SB	SB	SB	SB	SB	B
15	Fahmi Alpha Yanitra	SB	B	B	C	SB	SB	B	SB	B	B
16	Danguna	B	C	K	C	B	B	C	B	SB	B
17	Fahmi Dzikrullah	B	C	B	C	C	C	C	B	C	B
18	Avalya Dyah	B	B	B	B	SB	B	B	B	B	B
19	Ahmad Farhan	B	B	B	C	SB	SB	B	B	C	B
20	Aziz	SB	B	C	C	B	B	C	B	B	B
21	Prima Oktavia W.	B	B	B	C	B	SB	B	B	B	B
22	Miftahul arifuddin	B	C	C	K	B	B	B	B	B	B
23	Habib Adullah	SB	B	B	B	B	B	B	B	B	B
24	Binti Mu'arofah	SB	C	C	C	B	B	B	SB	B	SB
25	Widi Tri Hananto	B	B	B	B	C	C	B	B	B	C
26	Amirullah Andi Bramantya	B	C	K	K	C	C	B	B	B	C
27	Faisal	B	B	C	B	B	B	B	SB	B	B
28	Zaki Alawi	C	C	C	B	K	K	K	B	C	C

29	Dede Pradana	SB	B	B	SB	SB	B	SB	SB	SB	SB
30	Rizal Budi P	B	C	C	B	SB	B	SB	B	B	B
31	Fajar Rizqi Saputra	B	C	C	C	C	SB	C	SB	C	C
32	Ardina Kartika Sari	B	B	B	B	SB	SB	SB	B	SB	B
33	Muhamad Hasbi Sahara	SB	B	SB	SB	SB	B	B	SB	B	B
34	Kurniawan D.	B	C	C	C	B	B	C	B	B	B
35	Angga Gusta	B	B	B	B	B	B	B	B	SB	B
36	NN	C	B	C	K	C	B	B	SB	B	B
37	Latifah Nur Baiti	B	B	SB	C	SB	C	C	SB	C	C
38	Nursyahira Nabilla	C	C	K	C	C	B	C	B	B	C
39	Gdg	C	B	C	C	C	C	C	C	C	C
40	Niyatul Muna	B	B	B	C	B	C	B	B	B	B
41	Anonymous 1	C	C	C	C	C	C	B	B	B	C
42	Anonymous 2	B	B	B	B	B	B	SB	B	B	B
43	Anonymous 3	B	B	B	C	B	B	C	B	C	B
44	Anonymous 4	C	C	C	C	B	B	B	B	B	B
45	Anonymous 5	B	C	B	K	B	B	C	B	B	B
46	Asep	C	C	K	C	B	C	C	B	B	B
47	Nanang	B	C	C	K	C	C	C	B	C	C
48	Edo	B	K	C	C	C	C	C	B	C	C
49	Anonymous 6	C	C	B	B	B	B	C	B	B	B
50	Anonymous 7	B	C	B	B	B	B	C	B	B	B
Total											
SB		8	1	3	2	12	6	7	15	8	2
B		31	27	23	12	22	29	24	34	33	37
C		11	21	20	30	14	14	18	1	9	11
K		0	1	4	6	2	1	1	0	0	0
SK		0	0	0	0	0	0	0	0	0	0