

**APLIKASI ENKRIPSI SMS PADA *SMARTPHONE* ANDROID
MENGUNAKAN METODE *VIGENERE CIPHER*
DENGAN MODIFIKASI PERSAMAAN *FIBONACCI***

Skripsi

Untuk Memenuhi Sebagian Persyaratan

Mencapai Derajat Sarjana S-1

Program Studi Teknik Informatika



Disusun oleh

Nur Avesina Mustari

08650095

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2013



Universitas Islam Negeri Sunan Kalijaga

FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/556/2013

Skripsi/Tugas Akhir dengan judul : Aplikasi Enkripsi SMS Pada Smartphone Android
Menggunakan Metode Vigenere Cipher Dengan Modifikasi
Persamaan Fibonnaci

Yang dipersiapkan dan disusun oleh :
Nama : Nur Avesina Mustari
NIM : 08650095
Telah dimunaqasyahkan pada : Jum'at, 8 Februari 2013
Nilai Munaqasyah : A -
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

Bambang Sugiantoro, M.T
NIP. 19751024 200912 1 002

Penguji I

Sumarsono, M.Kom
NIP.19710209 200501 1 003

Penguji II

Nurochman, M.Kom
NIP. 19801223 200901 1 007

Yogyakarta, 14 Februari 2013
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan



Prof. Drs. H. Akh. Minhaji, M.A, Ph.D
NIP. 19580919 198603 1 002



SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Permohonan

Lamp : -

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

Assalamu 'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Nur Avesina Mustari

NIM : 08650095

Judul Skripsi : Aplikasi Enkripsi SMS pada Smartphone Android Menggunakan Metode *Vigenere Chiper* dengan Modifikasi Persamaan Fibonnaci

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Teknik Informatika.

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu 'alaikum wr. wb.

Yogyakarta, 28 Januari 2013

Pembimbing

Bambang Sugiantoro, M.T

NIP. 19751024 200912 1 002

PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini :

Nama : Nur Avesina Mustari
Nim : 08650095
Program Studi : Teknik Informatika
Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi dengan judul **APLIKASI ENKRIPSI SMS PADA SMARTPHONE ANDROID MENGGUNAKAN METODE VIGENERE CHIPER DENGAN MODIFIKASI PERSAMAAN FIBONNACI** tidak terdapat pada karya yang pernah diajukan untuk memperoleh gelar sarjana di suatu Perguruan Tinggi, dan sepengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 1 Februari 2013

Yang Menyatakan



Nur Avesina Mustari
NIM : 08650095

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi yang berjudul “Aplikasi Enkripsi SMS pada *Smartphone* Android Menggunakan Metode *Vigenere Cipher* dengan Modifikasi Persamaan *Fibonacci*”. Tak lupa pula sholawat serta salam semoga tetap tercurahkan kepada jujungan kita baginda Nabi Besar Muhammad SAW yang telah membawa kita dari zaman jahiliyah menuju zaman yang penuh dengan ilmu pengetahuan.

Penulisan skripsi ini tidak lepas dari bantuan dan dukungan serta bimbingan berbagai pihak. Dalam kesempatan ini penulis mengucapkan terima kasih dan penghargaan sebesar-besarnya kepada :

1. Prof.Dr. H. Musa Asy'arie, selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Drs Akh. Minhaji, M.A., Ph.D , selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga.
3. Bapak Agus Mulyanto, M.Kom. selaku Ketua Program Studi Teknik Informatika UIN Sunan Kalijaga Yogyakarta yang telah memberikan semangat, dorongan dan motivasi hingga terselesaikannya skripsi ini.
4. Bapak Bambang Sugiantoro, M.T selaku dosen pembimbing yang memberikan masukan, arahan dan bimbingan selama proses pelaksanaan dan penyelesaian skripsi.

5. Seluruh dosen Program Studi Teknik Informatika UIN Sunan Kalijaga, terima kasih atas kerjasama dan bantuannya.
6. Ibundaku tersayang Munawarotun, yang dengan penuh kesabarannya memberikan semangat, bimbingan, motivasi dan doa tiada henti. Semoga Allah SWT memberikan balasan pahala yang berlimpah atas kasih sayang bunda dan semoga penulis diberikan oleh-NYA kesempatan untuk membalas pengorbanan dan kasih sayang bunda.
7. Ayahanda tercinta Sartono, yang telah memberikan bimbingan , motivasi, inspirasi dan arahan sehingga penulis dapat menyelesaikan skripsi ini.
8. Mbak Fiska, “terima kasih sudah menjadi kakak yang baik”.
9. Terima kasih keluarga besar Muchtarom dan Kasan Syamsuri, atas motivasi dan dukungannya.
10. Terima Kasih untuk Arif, Ali, Ipul, Wawan, Mulyono, Kosim, Pengku, Syarif, Paijo, Ustad Heri yang telah membantu dan “mengganggunya”
11. Terima Kasih teman-teman satu kost Sanggar Tiban, atas dukungan dan yang selalu mengingatkan kegiatan yang lain selain skripsi
12. Serta semua teman-teman Teknik Informatika angkatan 2008 dan juga beberapa pihak yang telah membantu dan mendukung terselesaikannya skripsi ini. Penulis ucapkan terima kasih banyak.

Teriring do'a semoga amal baik ini mendapatkan balasan yang setimpal dari Allah SWT., Amin. Akhirnya, meski penulis telah berusaha sekuat tenaga untuk tidak melakukan kesalahan, namun sebagai manusia yang tak luput dari berbagai kekurangan dan kesalahan, skripsi ini masih kurang adanya. Oleh karena

itu, kritik dan saran dari pembaca sangat kami harapkan. Semoga skripsi ini dapat bermanfaat bagi semua pihak yang membacanya.

Yogyakarta, 14 Februari 2013

Nur Avesina Mustari

HALAMAN PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Ibu, Bapak dan Mbakyu, terima kasih untuk doa dan kasih sayang yang tiada henti

Keluarga besar Muchtarom dan Kasan Syamsuri

Semua Teman-teman Teknik Informatika Angkatan 2008

MOTTO

“Allah akan mengangkat orang-orang yang beriman di antara kamu dan orang-orang yang diberi ilmu beberapa derajat”

QS Al-Mujaadalah:21

Banyak kegagalan dalam hidup ini dikarenakan orang tidak menyadari betapa dekatnya mereka dengan keberhasilan saat mereka menyerah

-Thomas Alva Edison

Try not become a man of success, but rather try to become a man of value

-Albert Einstein

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
SURAT PERSETUJUAN SKRIPSI / TUGAS AKHIR	iii
PERNYATAAN KEASLIAN SKRIPSI.....	iv
KATA PENGANTAR	v
HALAMAN PERSEMBAHAN.....	viii
MOTTO.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xv
DAFTAR GAMBAR	xvi
DAFTAR MODUL	xviii
DAFTAR LAMPIRAN.....	xix
INTISARI.....	xx
<i>ABSTRACT</i>	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Batasan Penelitian	5
1.5 Manfaat Penelitian.....	5

1.6 Keaslian Penelitian.....	5
BAB II LANDASAN TEORI.....	7
2.1 Tinjauan Pustaka.....	7
2.2 Landasan Teori.....	9
2.2.1 <i>Short Message Service (SMS)</i>	9
2.2.1.1 <i>SMS Center</i>	10
2.2.1.2 <i>Arsitektur Jaringan SMS</i>	11
2.2.2 <i>Smartphone</i>	15
2.2.3 <i>Sistem Operasi Android</i>	16
2.2.3.1 <i>Pengenalan Sistem Operasi Android</i>	17
2.2.3.2 <i>Sejarah Sistem Operasi Android</i>	18
2.2.3.3 <i>Arsitektur Android</i>	23
2.2.3.4 <i>Fundamental Aplikasi</i>	27
2.2.4 <i>SQLite</i>	29
2.2.4.1 <i>Fitur SQLite</i>	29
2.2.5 <i>Aritmatika Modulus</i>	31
2.2.6 <i>byte</i>	31
2.2.7 <i>Vigenere Cipher</i>	31
2.2.7.1 <i>Sejarah Vigenere Cipher</i>	32
2.2.7.2 <i>Konsep Dasar</i>	32
2.2.7.3 <i>Kekuatan</i>	34
2.2.7.4 <i>Kelemahan</i>	34
2.2.8 <i>Metode Kasiski</i>	35

2.2.9 <i>ONE-TIME PAD</i>	37
2.2.9.1 Kekuatan	37
2.2.9.2 Kelemahan	38
2.2.10 <i>Fibonacci</i>	39
2.2.10.1 Sejarah <i>Fibonacci</i>	39
2.2.10.2 Baris <i>Fibonacci</i> dan Bilangan Keemasan.....	41
2.2.10.3 Barisan Dua Langkah <i>Fibonacci</i>	43
2.2.10.4 <i>Fibonacci</i> dalam Komputer	45
2.2.11 <i>Vigènere Cipher</i> Dengan Modifikasi <i>Fibonacci</i>	47
2.2.12 <i>UML (Unified Modelling Language)</i>	48
2.2.12.1 <i>Use Case Diagram</i>	49
2.2.12.2 <i>Activity Diagram</i>	50
2.2.12.3 <i>Sequence Diagram</i>	52
2.2.12.4 <i>Class Diagram</i>	53
BAB III METODE PENGEMBANGAN SISTEM	56
3.1 Analisis Kebutuhan Sistem	56
3.1.1 Pengumpulan Data.....	56
3.1.2 Kebutuhan Pengembangan Sistem	57
3.2 Desain / Perancangan Sistem (<i>Design</i>)	58
3.3 Implementasi Sistem	58
3.4 Pengujian Sistem	58
BAB IV ANALISIS DAN PERANCANGAN	60
4.1 Analisis Sistem	60

4.2 Analisis Masalah	60
4.3 Sistem Usulan.....	61
4.4 Analisis Kebutuhan Non-Fungsional	62
4.4.1 Analisis Metode Enkripsi dan Persamaan <i>Fibonacci</i>	62
4.4.2 Analisis Kebutuhan Perangkat Lunak	72
4.4.3 Analisis Kebutuhan Perangkat Keras	72
4.5 Analisis Kebutuhan Fungsional.....	73
4.5.1 Desains Tabel	73
4.5.2 Desains Antarmuka	80
4.5.2.1 Desain Antarmuka Daftar <i>Conversations</i>	81
4.5.2.2 Desains Antarmuka <i>New Conversations</i>	82
4.5.2.3 Desains Antarmuka <i>Contact Key</i>	83
4.5.2.4 Desains Antarmuka Daftar SMS.....	84
4.5.3 <i>Usecase</i>	85
4.5.4 <i>Activity Diagram</i>	86
4.5.4.1 <i>Activity Diagram New Conversations</i>	86
4.5.4.2 <i>Activity Diagram</i> Pilih SMS	87
4.5.5 <i>Class Diagram</i>	88
4.5.6 <i>Sequence Diagram</i>	89
BAB V IMPLEMENTASI DAN PENGUJIAN	95
5.1 Implementasi	95
5.1.1 Implementasi Bahasa Pemrograman <i>Java</i>	95
5.1.2 Implementasi Pengambilan Data SMS.....	97

5.1.3 Implementasi <i>Database</i> Kunci	98
5.1.4 Implementasi <i>Vigenere Cipher</i>	98
5.1.5 Implementasi Modifikasi <i>Fibonacci</i>	99
5.1.6 Implementasi Antarmuka Aplikasi.....	101
5.1.6.1 Antarmuka Awal	101
5.1.6.2 Antarmuka <i>New Conversations</i>	101
5.1.6.3 Antarmuka SMS.....	104
5.1.6.4 Antarmuka <i>List Contact Key</i>	105
5.1.7 Implementasi Enkripsi SMS.....	106
5.1.8 Implementasi Batasan Inputan	110
5.2 Pengujian.....	111
5.2.1 Pengujian <i>Alpha</i>	111
5.2.2 Pengujian <i>Beta</i>	112
BAB VI HASIL DAN PEMBAHASAN	114
6.1 Hasil Pengujian Sistem.....	114
6.2 Hasil Pembahasan Pengujian <i>Alpha</i>	115
6.3 Hasil Pembahasan Pengujian <i>Beta</i>	115
BAB VII PENUTUP	117
7.1 Kesimpulan.....	117
7.2 Saran.....	117
DAFTAR PUSTAKA	118
LAMPIRAN.....	120

DAFTAR TABEL

Tabel 2.1	Daftar Perbandingan Penelitian	8
Tabel 2.2	Notasi / Simbol <i>Use Case</i>	50
Tabel 2.3	Notasi / Simbol <i>Activity Diagram</i>	51
Tabel 2.4	Notasi / Simbol <i>Sequence Diagram</i>	53
Tabel 4.1	Kode ASCII Desimal	64
Tabel 4.2	Kode ASCII Desimal Pesan.....	65
Tabel 4.3	Nilai <i>Un</i> pada <i>Fibonacci</i>	68
Tabel 4.4	Kode ASCII Kunci Baru.....	69
Tabel 4.5	Hasil Penjumlahan ASCII desimal Pesan dengan Kuncinya	70
Tabel 4.6	Hasil ASCII decimal ke <i>Character</i>	72
Tabel 4.7	<i>sms</i>	75
Tabel 4.8	<i>threads</i>	76
Tabel 4.9	<i>raw_contact</i>	78
Tabel 4.10	<i>data</i>	79
Tabel 4.11	Tabkey	80
Tabel 5.1	Kuisoner Pengujian <i>Alpha</i>	111
Tabel 5.2	Kuisoner Pengujian <i>Beta</i>	112
Tabel 6.1	Tabel Daftar Responden Pengujian <i>Beta</i>	114
Tabel 6.2	Tabel Daftar Responden Pengujian <i>Alpha</i>	115
Tabel 6.3	Tabel Hasil Pengujian <i>Beta</i>	115

DAFTAR GAMBAR

Gambar 2.1	Jaringan Arsitektur SMS	15
Gambar 2.2	Arsitektur Android.....	23
Gambar 2.3	Diagram <i>Rabbit Fibonacci</i>	40
Gambar 4.1	<i>Flowchart</i> Proses <i>Vigenere</i> dengan Modifikasi <i>Fibonacci</i>	63
Gambar 4.2	Daftar Tabel Database SMS	74
Gambar 4.3	<i>Contacts Provider table structure</i>	77
Gambar 4.4	Desains Tampilan Daftar <i>Conversations</i>	81
Gambar 4.5	Desain Antarmuka Menu Keamanan SMS.....	82
Gambar 4.6	Desains Antarmuka <i>New Conversations</i>	82
Gambar 4.7	Desains Antar Muka Daftar Kontak	83
Gambar 4.8	Desain Antarmuka Daftar SMS.....	84
Gambar 4.9	<i>Usecase Diagram</i>	85
Gambar 4.10	<i>Activity Diagram</i> SMS Baru.....	86
Gambar 4.11	<i>Activity Diagram</i> SMS Pilih SMS	87
Gambar 4.12	<i>Class Diagram</i>	88
Gambar 4.13	<i>Sequence Diagram</i> Mode <i>SECURE</i>	90
Gambar 4.14	<i>Sequence Diagram</i> Mode Not <i>SECURE</i>	91
Gambar 4.15	<i>Sequence Diagram</i> TAB 1.....	92
Gambar 4.16	<i>Sequence Diagram</i> TAB 2.....	93
Gambar 4.17	<i>Sequence Diagram</i> Menerima Pesan	94

Gambar 5.1	Implementasi <i>Java</i>	96
Gambar 5.2	Antarmuka Awal	101
Gambar 5.3	Notifikasi Keamanan	102
Gambar 5.4	Antarmuka <i>New Conversations</i>	102
Gambar 5.5	Antarmuka <i>Contact</i>	103
Gambar 5.6	Antarmuka <i>New Conversations</i>	103
Gambar 5.7	Antarmuka <i>Contact key</i>	104
Gambar 5.8	Antarmuka SMS	104
Gambar 5.9	Antarmuka <i>List Contact key</i>	105
Gambar 5.10	Antarmuka <i>add key</i>	106
Gambar 5.11	Antarmuka Ketik SMS	106
Gambar 5.12	Antarmuka <i>Sending Message</i>	107
Gambar 5.13	Antarmuka SMS Terenkripsi	108
Gambar 5.14	Antarmuka SMS Masuk	109
Gambar 5.15	LogCat	110

DAFTAR MODUL

Modul 5.1	<i>Script Vigenere</i>	99
Modul 5.2	<i>Script Fibonacci</i>	100
Modul 5.3	<i>Script Kirim Pesan</i>	107
Modul 5.4	<i>Script Receive SMS</i>	109

DAFTAR LAMPIRAN

LAMPIRAN A	<i>SOURCE CODE</i> APLIKASI ENKRIPSI SMS.....	1
LAMPIRAN B	DAFTAR PENGUJI <i>BETA</i>	54
LAMPIRAN C	QUISSONNER	56
LAMPIRAN D	<i>CURRICULUM VITAE</i>	72

**APLIKASI ENKRIPSI SMS PADA SMARTPHONE ANDROID
MENGUNAKAN METODE *VIGENERE CIPHER*
DENGAN MODIFIKASI PERSAMAAN FIBONNACI**

**Nur Avesina Mustari
NIM. 08650095**

INTISARI

Smartphone berbasis Android memiliki fitur pengiriman data berupa pesan singkat melalui *Short Message Service* (SMS). Namun fasilitas SMS tidaklah aman sehingga perlu dilakukan kriptografi dan salah satu metode kriptografi adalah *Vigènere chiper*, namun *Vigènere cipher* relatif mudah untuk dipecahkan dengan metode Kasiski oleh karena itu dibutuhkan *unbreakable chipper* menggunakan prinsip *ONE-TIME PAD*, maka *Vigènere cipher* menggunakan deretan karakter kunci yang “acak” sepanjang plainteks dengan memanfaatkan sifat bilangan *Fibonacci* untuk memenuhi prinsip *ONE-TIME PAD*. Sehingga penelitian bertujuan membuat Aplikasi *Smartphone* berbasis Android menggunakan metode *Vigènere cipher* dengan modifikasi persamaan *Fibonacci*.

Aplikasi dikembangkan dengan metode *System Development Life Cycle* (SDLC) model *waterfall*. menggunakan metode kriptografi *Vigenerere Cipher* dan kunci menggunakan prinsip *ONE-TIME PAD* dengan modifikasi persamaan *Fibonacci* pada *Smartphone* Android versi 4.0 atau yang lebih tinggi. Aplikasi Enkripsi SMS tersebut dapat meng-enkripsikan SMS keluar, mendekripsikan SMS masuk, dan menampilkannya dalam bentuk *Conversations*, dan juga menyimpan data Kunci untuk tiap data Buku *Telephone* pada *Smartphone* berbasis Android

Berdasarkan data hasil pengujian *Alpha* terdapat kesesuaian antara perancangan dengan hasil Aplikasi sedangkan data hasil pengujian *Beta* menunjukkan bahwa responden sangat setuju 22.22%, setuju 74,44%, tidak setuju 3.33%, dan yang mengatakan sangat tidak setuju 0%. Berdasarkan hasil pengujian tersebut, dapat disimpulkan Aplikasi Enkripsi SMS sebanyak 74.44 % setuju dapat digunakan.

Kata kunci: Android, *Short Message Service* (SMS), *ONE-TIME PAD* ,*Vigenerere Cipher*, *Fibonnaci*,

APPLICATION SMS ENCRYPTION ON ANDROID SMARTPHONE USING VIGENERE CIPHER'S METHOD WITH MODIFICATIONS OF THE EQUATION FIBONACCI

Nur Avesina Mustari
NIM. 08650095

ABSTRACT

Android-based smartphones have features such as short message data transmission through the Short Message Service (SMS). But SMS is not secure facility so we need cryptography and cryptographic methods are one of the Vigenere cipher, Vigenere cipher but relatively easy to solve with this method therefore needs Kasiski unbreakable chipper uses the principle of ONE-TIME PAD, the Vigenere cipher key using the sequence of characters "random" throughout the plaintext by exploiting properties of Fibonacci numbers to satisfy the principle of ONE-TIME PAD. So the research aims to create applications based on Android Smartphone using Vigenere cipher with a modified Fibonacci equation.

Applications developed with the method of System Development Life Cycle (SDLC) model waterfall. Vigenere Cipher using cryptographic methods and keys using the principle of ONE-TIME PAD with modifications equation Fibonacci on Android Smartphone version 4.0 or higher. SMS Application Encryption can encrypt outgoing SMS, incoming SMS decrypt and display it in the form of Conversations, and also stores data for each Key Contact data on Android based Smartphones

Based on the Alpha test results are the results of the fit between the design of applications while data Beta test results showed that 22.22% of respondents strongly agree, agree 74.44%, 3.33% disagree, and strongly disagree that said 0%. Based on the test results, we can conclude SMS Encryption Applications 74.44% as agreed can be used.

Keywords: Android, *Short Message Service (SMS)*, *ONE-TIME PAD*, *Vigenere Cipher*, *Fibonacci*.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Smartphone atau telepon pintar merupakan telepon yang bekerja menggunakan seluruh perangkat lunak sistem operasi yang menyediakan hubungan standar dan mendasar bagi pengembang aplikasi (Wikipedia,2012), walaupun tidak ada *standart* yang jelas yang membedakan antara sebuah ponsel (telepon) biasa dengan sebuah ponsel pintar, tetapi perkembangan sebuah ponsel saat ini mengarah pada bagaimana menjadikan sebuah ponsel sebagai perangkat komputer yang berukuran kecil, sehingga sebuah *smartphone* tentunya memerlukan sistem operasi seperti apa didefinisikan diatas bahwa sistem operasi tersebut digunakan sebagai penyedia hubungan *standart* dan mendasar bagi pengembang aplikasi.

Sistem operasi yang ada saat ini antara lain Symbian, Bada OS, Palm, Windows Phone, BlackBerry OS, iOS, Android OS akan tetapi perkembangan Smartphone dan Tablet PC berbasis Android berkembang dengan sangat pesat, hal ini terbukti saat ini hampir semua vendor-vendor *smartphone* sudah memproduksi smartphone berbasis Android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, WebstationCamangi, Dell, Nexus, SciPhone, WayteQ, Sony

Ericsson, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi *vendor smartphone* di dunia yang memproduksi *smartphone* (Nazarudin,2012).

Smartphone berbasis Android dengan segala fasilitas mutakhir dan salah satu fasilitas yang disediakan *smartphone* adalah untuk melakukan pengiriman data berupa pesan singkat melalui *Short Message Service* (SMS). Namun dengan fasilitas SMS yang ada, timbul pertanyaan mengenai keamanan informasi jika seseorang ingin mengirimkan suatu informasi rahasia melalui fasilitas SMS. Survei juga menemukan bahwa di saat pengguna menyadari bahwa mereka mendapatkan kepuasan dan produktivitas ketika mereka diperkenankan menggunakan *Smartphone* pilihan mereka di kantor, mereka tidak benar-benar memahami secara penuh besarnya ancaman keamanan yang dihasilkan. Bahkan, 78 persen responden mengira bahwa dengan memperkenankan karyawan menggunakan *smartphone* pilihan mereka tidak memiliki dampak atau hanya berdampak kecil terhadap penurunan tingkat keamanan jaringan dan informasi perusahaan (Darric,2012).

Di luar negeri pemanfaatan SMS untuk mengirim pesan rahasia telah lebih dulu dikembangkan. Misalnya di Inggris sebuah perusahaan operator telepon selular, *staellium UK*, mengeluarkan layanan bernama "*stealth text*" yang dapat digunakan untuk mengirim pesan dengan aman, yaitu dengan cara menghapus pesan secara otomatis segera setelah 40 detik pesan dibaca atau yang dikenal dengan nama *self-destruct text message*. Ada juga

pengamanan sms dengan menggunakan kriptografi sms yang memanfaatkan kunci untuk mendekripsikan sms yang telah di enkripsi.

Salah satu metode kriptografi *Vigènere cipher*, namun *Vigènere cipher* relatif mudah untuk dipecahkan dengan kriptanalisis, yakni dengan memanfaatkan perulangan huruf ataupun perulangan pasangan huruf. Perulangan huruf ini mungkin untuk membangkitkan perulangan pada cipherteks. Teknik ini digunakan pada metode Kasiski yang biasa dipakai untuk memecahkan *Vigènere cipher*. Karena itu, *Vigènere cipher* biasa merupakan *breakable cipher* (*chiper* yang dapat dipecahkan).

Untuk membuat *unbreakable cipher* (*chiper* yang tidak dapat dipecahkan), hal yang harus dilakukan adalah menggunakan kunci yang benar-benar acak dan panjang kunci sama dengan panjang plainteks. Salah satu algoritma kriptografi yang tidak dapat dipecahkan adalah *ONE-TIME PAD*, yang menggunakan deretan karakter kunci yang dibangkitkan secara acak. Sayangnya, algoritma ini tidak efisien karena bermasalah saat menyimpan dan mendistribusikan kunci yang sangat panjang. Dengan memanfaatkan sifat algoritma untuk *unbreakable chiper*, seperti pada *ONE-TIME PAD*, *Vigènere chiper* bisa dikembangkan menjadi lebih susah untuk dipecahkan. Caranya adalah dengan membangkitkan deretan karakter kunci yang “acak” sepanjang *plaintext* dengan memanfaatkan sifat bilangan *Fibonacci* terhadap kunci masukan dari pengguna. Kunci masukan ini pun tidak perlu sangat panjang, sehingga memudahkan untuk disimpan dan didistribusikan.

Oleh karena itu penulis akan membangun Aplikasi Enkripsi dan Dekripsi SMS pada *Smartphone* berbasis Android menggunakan metode *Vigènere chipper* dengan modifikasi *Fibonacci*, sehingga pemilik *Smartphone* yang berbasis Android dapat melakukan pertukaran data (SMS) dengan lebih aman dan nyaman tanpa mempengaruhi pengiriman pesan biasa (tanpa pengamanan enkripsi).

1.2 Rumusan Masalah

Pada umumnya setiap pengguna ponsel ketika berkomunikasi melalui pesan teks (SMS) dengan mudah dan terlindung dari penyalahgunaan dan penyadapan, maka dapat dibuat suatu rumusan masalah yaitu bagaimana merancang Aplikasi enkripsi dan dekripsi pesan teks (SMS) menggunakan metode *Vigènere Chiper* dengan modifikasi *Fibonacci* untuk membuat unbreakable seperti pada *ONE-TIME PAD* pada *Smartphone* berbasis Android dengan menggunakan bahasa pemrograman *Java* Android.

1.3 Batasan Penelitian

Agar penelitian dapat lebih terfokus maka penelitian ini diberikan batasan antara lain:

1. Inputan berupa pesan sms baik berupa teks baik berupa simbol *emoticon* yang biasa digunakan dalam layanan SMS
2. Menggunakan tiga buah *database* diantaranya dua *database default* dari Android yaitu *database* SMS dan *Contact*, dan satu *database* yang dibuat penulis dengan menggunakan *SQLite*

3. Target development Aplikasi Android adalah Android versi 4.0 atau *targetsdk* adalah API versi 14, dengan *minsdk* API 11
4. Aplikasi yang dibuat hanya menggunakan data buku telepon yang tersimpan di *phonestorage* Android
5. Aplikasi hanya dapat mengirimkan pesan melalui nomor *default* untuk *device* Android *dual* SIM

1.4 Tujuan Penelitian

Tujuan penulis melakukan penelitian ini adalah untuk membangun aplikasi *Smartphone* berbasis Android berupa aplikasi enkripsi SMS menggunakan metode *Vigènere Chiper* dengan pengacakan kunci dengan modifikasi persamaan *Fibonacci*.

1.5 Manfaat Penelitian

Dengan dilakukannya penelitian ini, harapan yang ingin dicapai oleh penulis adalah memberikan pengamanan terhadap layanan SMS (*Short Message Service*) yang ada pada *smartphone* berbasis Android sehingga pesan atau informasi yang terdapat didalamnya dapat tersampaikan kepada pihak yang berhak.

1.6 Keaslian penelitian

Sepengetahuan penulis, penelitian tentang Enkripsi SMS pada *smartphone* berbasis Android menggunakan metode *Vigènere* di UIN Sunan Kalijaga belum ada, penelitian sebelumnya terdapat penelitian enkripsi SMS pada perangkat ponsel yang mendukung J2ME dengan judul *Aplikasi Enkripsi SMS Pada Telephone Seluler Dengan Metode Base 64* (Nadziroh,

2012), dan metode enkripsi *Vigènere* pernah digunakan pada penelitian yang berjudul *Implementasi Stegonografi dan Algoritma Enskripsi Vigènere Cipher pada Media Plain Text* (Tsani, 2012)

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan penulis pada Aplikasi Enkripsi SMS menggunakan Metode *Vigenere Cipher* dengan modifikasi *Fibonnaci* maka dapat diambil kesimpulan yaitu penelitian ini berhasil mengembangkan Aplikasi Enkripsi tersebut pada *Smartphone* berbasis Android dengan menggunakan bahasa pemrograman Java, dengan database SQLite dan mengakses database SMS dari device *Smartphone Android* dan menghubungkan database Contact dengan database Contact key

7.2 Saran

Penelitian yang dilakukan tidak terlepas dari kekurangan dan kelemahan. Oleh karena itu, untuk kebaikan pengembangn sistem lebih lanjut, maka perlu diperhatikan beberapa hal, diantaranya:

1. Aplikasi lebih cepat dalam menampilkan data SMS
2. Aplikasi diberikan fasilitas dalam melakukan tukar-menukar kunci dengan lebih mudah dan aman
3. Aplikasi dapat menghapus satu *Conversations* penuh

Akhirnya dengan segala keterbatasan hasil penelitian ini, penulis tetap berharap bahwa penelitian ini akan memberikan gagasan baru bagi pembaca untuk mengembangkan lebih lanjut.

DAFTAR PUSTAKA

- Aljufri, Fatimah. 2012. "Sistem Pemandu Pencarian Masjid Terdekat Berbasis Lokasi Di Atas Platform Android". Yogyakarta: Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
- Dhocostantine. 2010. *Cara Kerja SMS*. <http://dhoconstatine.wordpress.com/2010/05/12/cara-kerja-sms-short-message-service-2/>
- Hor, Darric. "Survei: Karyawan Tak Paham Tantangan Keamanan Perangkat Mobile" http://www.infogate.com/viewstory/2011/09/26/survei_karyawan_tak_paham_tantangan_keamanan_perangkat_mobile (diakses tanggal 20 Desember 2012).
- Kristian N, Bayu. 2010. "Aplikasi Enkripsi SMS Pada Telepon Selular Berbasis J2ME Dengan Metode Vigenere Cipher". Semarang: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Diponegoro.
- Munir, Rinaldi. 2004. *Matematika Diskrit*. Bandung: Informatika
- Nadziroh, Ulfah. 2012. "Aplikasi Enkripsi SMS Pada Telephone Seluler Dengan Metode Base 64". Yogyakarta: Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga
- Naughton, Patrick. 2002. *Java Handbook Konsep Dasar Pemrograman Java*. Yogyakarta: Penerbit Andi
- Pressman, K.S. 2008. *Rekayasa Perangkat Lunak*. Andi : Yogyakarta.
- Pujastuti, Puji. 2011. "Enkripsi SMS Menggunakan Advanced Encryption Standard Pada J2ME dengan Bantuan Bouncy Castle Cryptography API". Yogyakarta: Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer Amikom
- Rorie, Vincentius W, Ferdinandes, Boyke dan Stephanie. 2011 "Analisis Dan Perancangan Aplikasi Komposer Musik Berbasis Jaringan Ad Hoc Pada Ponsel Dengan Sistem Operasi Android". Jakarta: Jurusan Teknik Informatika Universitas Bina Nusantara.
- Safaat, Nazarudin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Penerbit Informatika.

- Sugianto, Anggriawan. , _____, Vigenere Cipher dengan Modifikasi Fibonacci,[pdf],(http://www.scribd.com/document_downloads/direct/39161913?extension=pdf&ft=1356587300<=1356590910&uahk=phUZu3FUDo/u/OMwObBv2VyThPU diakses tanggal 27 Desember 2012).
- Tsani, Fahmi Aulia. 2012. "Implementasi Stegonografi dan Algoritma Enskripsi Vigenere Chipere pada Media Plain Text", Yogyakarta: Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
- Wahono, Romi Satria; Dharwiyanti, Sri, 2003, Pengantar Unified Modeling Language (UML) , [pdf], (http://risdawati.staff.gunadarma.ac.id/Downloads/files/31599/Modul_UML.pdf diakses 27 Desember 2012).
- Wekiardi, Hasni. 2008. *Analisa Performasi pengiriman Short Message Service (SMS) untuk Pelanggan Prabayar pada Jaringan GSM PT Indosat*, <http://www.lontar.ui.ac.id>.
- Wikipedia . "Telepon Cerdas " . http://id.wikipedia.org/wiki/Telepon_cerdas (diakses tanggal 20 Desember 2012).
- Wikipedia . "Layanan pesan singkat" . http://id.wikipedia.org/wiki/Layanan_pesan_singkat (diakses tanggal 21 Desember 2012).
- Wikipedia . "SQLite" . <http://id.wikipedia.org/wiki/SQLite> (diakses tanggal 21 Desember 2012).
- Wikipedia . "Bilangan Fibonacci" . http://id.wikipedia.org/wiki/Bilangan_Fibonacci (diakses tanggal 21 Desember 2012).
- Wikipedia . "Fibonacci Number" . http://en.wikipedia.org/wiki/Fibonacci_number (diakses tanggal 21 Desember 2012).

LAMPIRAN A

SOURCE CODE APLIKASI ENKRIPSI SMS

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
  xmlns:android="http://schemas.android.com/apk/res/android"
    package="aves.enkripsi.sms"
    android:versionCode="1"
    android:versionName="1.0.0">
  <uses-sdk
    android:minSdkVersion="13"
    android:targetSdkVersion="14"></uses-sdk>
  <uses-permission android:name="android.permission.SEND_SMS"/>
  <uses-permission android:name="android.permission.VIBRATE" />
  <uses-permission
  android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission
  android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission
  android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"
  />
  <uses-permission
  android:name="android.permission.READ_PHONE_STATE" />
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.RECEIVE_SMS" />
  <uses-permission android:name="android.permission.RECORD_AUDIO" />
  <uses-permission
  android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
  <uses-permission android:name="android.permission.READ_CONTACTS"
  />
  <uses-permission
  android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
  <uses-permission
  android:name="android.permission.ACCESS_NETWORK_STATE" />
  <uses-permission android:name="android.permission.GET_ACCOUNTS" />
  <uses-permission
  android:name="android.permission.BROADCAST_STICKY" />
  <uses-permission android:name="android.permission.READ_SMS" />
  <uses-permission
  android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"
  />
  <uses-permission android:name="android.permission.VIBRATE" />
  <uses-permission android:name="android.permission.WRITE_SMS"/>
  <application android:label="@string/app_name"
    android:icon="@drawable/icon_sms">
    <activity android:name=".MainActivity"
      android:label="@string/app_name">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"
        />
        <category
        android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>

```



```

        android:name=".SMSActivity"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name=".key.KeyListActivity"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name=".key.KeyActivity"
        android:theme="@android:style/Theme.Dialog"
        android:icon="@android:drawable/ic_lock_idle_lock"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name=".SMSNewActivity"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:name=".data.ContactDB"
        android:label="@string/app_name"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>

```

```

        <activity
            android:label="@string/app_name"
android:name=".SendSMS"
            android:launchMode="singleTop"
            android:configChanges="keyboardHidden|orientation">
        <intent-filter>
            <action android:name="android.intent.action.VIEW"
/>
        <category
android:name="android.intent.category.DEFAULT" />
        </intent-filter>
    </activity>
    <activity
        android:label="@string/app_name" android:name=".SMS"
        android:launchMode="singleTop"
        android:configChanges="keyboardHidden|orientation">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"
/>
    </intent-filter>
</activity>
<receiver android:name=".SmsReceiver">
    <intent-filter>
        <action android:name=
            "android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
</application>
</manifest>

```

MainActivity.java

```

package aves.enkripsi.sms;

import java.util.ArrayList;
import java.util.HashMap;

import android.app.ActionBar.Tab;
import android.app.ActionBar;
import android.app.Activity;
import android.app.Fragment;
import android.app.FragmentTransaction;
import android.app.ListFragment;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast; import aves.enkripsi.sms.fragment.*;
import aves.enkripsi.sms.key.KeyListActivity;

public class MainActivity extends Activity{

```

```

    public static final ArrayList<HashMap<String, String>>
SMSThread = new ArrayList<HashMap<String, String>>();
    public static Context c;
    Button btsms;
    ListFragment PlayerFragment;
    ListFragment StationsFragment;
    ActionBar.Tab PlayerTab;
    ActionBar.Tab StationsTab;
    ActionBar actionbar;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.maintab);
        btsms = (Button)findViewById(R.id.composesms);
        c=getApplicationContext();
        BulidActionTab();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.mainmenu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        if(item.getItemId()==R.id.composesms){
            Intent in = new Intent(getApplicationContext(),
SMSNewActivity.class);
            in.putExtra("msg", "");
            startActivity(in);
        }else if(item.getItemId()==R.id.listkey){
            Intent i = new Intent(getApplicationContext(),
KeyListActivity.class);
            i.putExtra("pick", false);
            startActivity(i);
        }else{
        }
        return false;
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putInt("tab",
getActionBar().getSelectedNavigationIndex());
    }

```

```

    public void BulidActionTab() {
        actionBar = getActionBar();

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        PlayerTab =
        actionBar.newTab().setIcon(R.drawable.device_access_not_secure);
        StationsTab =
        actionBar.newTab().setIcon(R.drawable.device_access_secure);

        PlayerFragment = new AFragment();
        StationsFragment = new BFragment();

        PlayerTab.setTabListener(new
MyTabsListener(PlayerFragment));
        StationsTab.setTabListener(new
MyTabsListener(StationsFragment));
        actionBar.addTab(PlayerTab);
        actionBar.addTab(StationsTab);
    }
}

class MyTabsListener implements ActionBar.TabListener {
    public Fragment fragment;
    ViewPager mViewPager;

    public MyTabsListener(Fragment fragment) {
        this.fragment = fragment;
    }

    public void onTabReselected(Tab tab, FragmentTransaction ft)
{
        Toast.makeText(MainActivity.c, "Reselected!",
Toast.LENGTH_LONG).show();
    }

    public void onTabSelected(Tab tab, FragmentTransaction ft) {
        ft.replace(R.id.fragment_container, fragment);
    }

    public void onTabUnselected(Tab tab, FragmentTransaction ft)
{
        ft.remove(fragment);
    }
}

```

SMS.java

```

package aves.enkripsi.sms;

import java.util.ArrayList;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ListActivity;
import android.app.PendingIntent;
import android.app.ProgressDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.os.Handler;
import android.telephony.SmsManager;
import android.util.Log;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import aves.enkripsi.sms.data.SmsDB;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.KeyDB;
import aves.enkripsi.sms.key.SmsVigenere;

public class SMS extends ListActivity {
    ProgressDialog progDial;
    public static final int PICK_CONTACT_not_SECURE = 1;
    public static final int PICK_CONTACT_SECURE = 2;
    SmsManager SMSMANAGER;

    String phone=null;
    String name;

    TextView et_msg;
    TextView et_phn;
    TextView msg_count;
    ImageButton bt_add;
    Button bt_send;
    LinearLayout New;
    LinearLayout Old;

    SmsDB sms;
    KeyDB kdb=null;
    SmsVigenere SV;
    ContactDB contact;

    String mode;
    String p,m;

    String NewSMS="1";
    String openThread="2";

    Loading l;
    ProgressDialog dlg;

```

```

    public Handler handler;
    public BroadcastReceiver smsReceiver;
    public static final String
SMS_SENT_ACTION="aves.enkripsi.sms.SMS_SENT";
    public static final String
SMS_ERROR_ACTION="aves.enkripsi.sms.SMS_ERROR";

    public void onCreate(Bundle b){
        super.onCreate(b);
        setContentView(R.layout.sms);
        sms = new SmsDB(this);
        kdb= new KeyDB(this);
        l = new Loading(this);
        SV =new SmsVigenere();
        contact= new ContactDB(this);
        ///contact = new getContact(this);

        SMSMANAGER = SmsManager.getDefault();
    }

    protected void sendSMS(String phoneNumber,String
phoneNumber, String status, String key){

        String SENT = "SMS_SENT";
        String DELIVERED = "SMS_DELIVERED";
        this.p=phoneNumber;
        this.m=phoneNumber;

        if(status.equalsIgnoreCase("lock")){
            phoneNumber=SV.smsEnkrip(phoneMessage, key);
        }
        ///this.m=phoneNumber;

        registerReceiver(new BroadcastReceiver(){
            String help;
            @Override
            public void onReceive(Context arg0, Intent arg1)
{
                switch (getResultCode())
                {
                    case Activity.RESULT_OK:

Toast.makeText(getApplicationContext(), "SMS sent",

                    Toast.LENGTH_SHORT).show();

                    et_msg.setText("");
                    onRefresh(p,m);
                    progDial.dismiss();
                    p="";
                    m="";

                    break;

                    case
SmsManager.RESULT_ERROR_GENERIC_FAILURE:

Toast.makeText(getApplicationContext(), "Generic failure",

```

```

        Toast.LENGTH_SHORT).show();
        help="Generic failure
cause";
        onFailedSend(p,m,"Generic failure",help);
        progDial.dismiss();
        break;
        case
SmsManager.RESULT_ERROR_NO_SERVICE:
Toast.makeText(getApplicationContext(),"No service",
        Toast.LENGTH_SHORT).show();
        help="Failed because
service is currently unavailable";
        onFailedSend(p,m,"No
service",help);
        progDial.dismiss();
        break;
        case SmsManager.RESULT_ERROR_NULL_PDU:
Toast.makeText(getApplicationContext(),"Null PDU",
        Toast.LENGTH_SHORT).show();
        help="Failed because no
pdu provided";
        onFailedSend(p,m,"Null
PDU",help);
        progDial.dismiss();
        break;
        case
SmsManager.RESULT_ERROR_RADIO_OFF:
Toast.makeText(getApplicationContext(),"Radio off",
        Toast.LENGTH_SHORT).show();
        help="Failed because
radio was explicitly turned off";
        onFailedSend(p,m,"Radio
off",help);
        progDial.dismiss();
        break;
    }
}

}, new IntentFilter(SENT));

//---when the SMS has been delivered---
registerReceiver(new BroadcastReceiver() {
    @SuppressWarnings("NewApi")
    @Override
    public void onReceive(Context arg0, Intent arg1)
{
        switch (getResultCode())

```

```

        {
            case Activity.RESULT_OK:
Toast.makeText(getApplicationContext(), "SMS delivered",
    Toast.LENGTH_SHORT).show();

                break;
            case Activity.RESULT_CANCELED:
Toast.makeText(getApplicationContext(), "SMS not delivered",
    Toast.LENGTH_SHORT).show();

                break;
        }
    }, new IntentFilter(DELIVERED));

    ArrayList<String> messages =
SMSMANAGER.divideMessage(phoneMessage);
    ArrayList<PendingIntent> listOfIntentsSENT = new
ArrayList<PendingIntent>();
    ArrayList<PendingIntent> listOfIntentsDELIVERY = new
ArrayList<PendingIntent>();

    for (int i=0; i < messages.size(); i++){
        Intent sentIntent = new Intent(SENT);
        Intent delevaryIntent=new Intent(DELIVERED);
        PendingIntent pis =
PendingIntent.getBroadcast(this, 0, sentIntent,
PendingIntent.FLAG_CANCEL_CURRENT);
        PendingIntent pid =
PendingIntent.getBroadcast(this, 0, delevaryIntent,
PendingIntent.FLAG_CANCEL_CURRENT);
        listOfIntentsSENT.add(pis);
        listOfIntentsDELIVERY.add(pid);
    }
    Log.i("SENT SMS", phoneNumber+":"+phoneMessage);
    SMSMANAGER.sendMultipartTextMessage(phoneNumber, null,
messages, listOfIntentsSENT,listOfIntentsDELIVERY);
}

@Override
public void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    kdb.close();
}

public void onRefresh(String p, String m){
}

```



```

    public void onFailedSend(String p, String m, String error,
String help){

    }

    public void putData(String id){

    }

    public ProgressDialog getProgressDialog() {
        // TODO Auto-generated method stub
        return dlg;
    }

}

```

SMSActivity.java

```

package aves.enkripsi.sms;

import java.sql.Date;
import java.util.ArrayList;
import java.util.HashMap;

import android.annotation.SuppressLint;

import android.app.AlertDialog.Builder;
import android.app.AlertDialog;
import android.app.Dialog;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;

import android.content.ClipboardManager;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.ClipData.Item;
import android.database.Cursor;

import android.os.Bundle;

import android.text.InputFilter;
import android.text.SpannableStringBuilder;
import android.text.Spanned;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;

import android.widget.AdapterView;

```

```

import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

@SuppressLint("ShowToast")
public class SMSActivity extends SMS{
    public static final ArrayList<HashMap<String, String>> SMS =
new ArrayList<HashMap<String, String>>();
    final CharSequence[] items = {"Copy message text", "Forward
Message", "Delete Message"};
    public String thread_id;
    String status;
    Item menu_enkripsi;
    ListAdapter adapterinbox;
    public static final int DELETE_ALERT=1;
    public static final int CHANGE_STATUS_ALERT=2;
    MenuItem menuItem;
    String id_contact;

    public void onCreate(Bundle savedInstanceState) {
        //-----inisiiasi layout-----
-//
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sms);

        progDial=1.onThreadRun("Opening
Conversation.....");

        New=(LinearLayout) findViewById(R.id.New);
        Old=(LinearLayout) findViewById(R.id.Old);
        New.setVisibility(LinearLayout.INVISIBLE);
        Old.setVisibility(LinearLayout.VISIBLE);

        phone=null;

        et_msg = (EditText) findViewById(R.id.inputBody);
        bt_send = (Button) findViewById(R.id.SendButton);
        msg_count =
(TextView) findViewById(R.id.count_sms);

        Bundle b = getIntent().getExtras();
        thread_id = b.getString("thread_id");
        name = b.getString("name");
        et_msg.setText(b.getString("msg").toString());
        String type=b.getString("type");
        if (type.equalsIgnoreCase("unread")) {
            sms.setReadThread(thread_id);
        }
        setTitle(name);

```

```

        phone=sms.getNumber (thread_id);

        id_contact=contact.getContactIDFromNumber (sms.getNumber (thead_id));

        bt_send.setOnClickListener (new
View.OnClickListener () {

            public void onClick (View v) {

                if (et_msg.getText ().toString ().length ()==0) {
                    Toast.makeText (getBaseContext (),
                "You Not Entry Any Text Message", Toast.LENGTH_SHORT);
                } else {
                    if (phone==null && et_phn.length () !=0
) {

                        phone=et_phn.getText ().toString ();
                    }
                    String key=kdb.getKey (id_contact);
                    sendSMS (phone,
                et_msg.getText ().toString (), status, key);
                    progDial=1.onThreadRun ("Sending
                message.....");
            }
        }
    });
}

@Override
public boolean onCreateOptionsMenu (Menu menu) {
    MenuInflater inflater = getMenuInflater ();
    inflater.inflate (R.menu.smsmenu, menu);
    menuItem= menu.findItem (R.id.enkripsi);
    if (kdb.cekStatus (id_contact)) {

        menu.findItem (R.id.enkripsi).setIcon (R.drawable.device_access_secure);

        status="lock";
        InputFilter filter = new InputFilter () {
            @Override
            public CharSequence
filter (CharSequence source, int start, int end,
                Spanned dest, int
dstart, int dend) {

                if (source instanceof
                SpannableStringBuilder) {
                    SpannableStringBuilder
                sourceAsSpannableBuilder = (SpannableStringBuilder) source;
                    for (int i = end - 1; i >= start;
                i--) {

                        char currentChar =
                source.charAt (i);

```

```

                int ascii = currentChar;
                if (ascii > 127) {
sourceAsSpannableBuilder.delete(i, i+1);

Toast.makeText(getApplicationContext(), "Invalid non-Ascii
Character", Toast.LENGTH_SHORT).show();
                }
            }
            return source;
        } else {
            StringBuilder
filteredStringBuilder = new StringBuilder();
            for (int i = 0; i < end; i++) {
                char currentChar =
source.charAt(i);

                int ascii = currentChar;
                if (ascii <= 127 && ascii >= 0)
{
filteredStringBuilder.append(currentChar);
                }
            }
            return
filteredStringBuilder.toString();
        }
    }
};
    et_msg.setFilters(new InputFilter[]{filter});
} else {
    menu.findItem(R.id.enkripsi).setIcon(R.drawable.device_access_not_secure);
        status="unlock";
    }
    return super.onCreateOptionsMenu(menu);
}

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.enkripsi: {
            if(status=="lock") {
                showDialog(CHANGE_STATUS_ALERT);
            } else {
                if(kdb.cekkey(id_contact)) {

item.setIcon(R.drawable.device_access_secure);
                status="lock";

                kdb.setStatus(id_contact, kdb.LOCK);
                } else {
                    Toast.makeText(this, "You don't
have any key to enkription this message You must set your key in
menu setting", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
}

```

```

        }
        }break;
    }
    return false;
}

public void putData(String thread_id){
    SMS.clear();
    Cursor cursor = sms.GetInbox(thread_id);
    while (cursor.moveToNext()){
        String
message=cursor.getString(cursor.getColumnIndex("body"));
        String id=cursor.getString(0);
        Long dt=Long.valueOf(cursor.getString(4));
        Date dateFromSms = new Date(dt);
        String date= dateFromSms.toLocaleString();
        HashMap<String, String> map = new
HashMap<String, String>();

        if(cursor.getString(cursor.getColumnIndex("type")).toString().equals
equalsIgnoreCase("1")){
            map.put("msginbox",message);
            map.put("type", "1");
        }else
        if(cursor.getString(cursor.getColumnIndex("type")).toString().equals
equalsIgnoreCase("2")){
            map.put("msginbox",message);
            map.put("type", "2");
        }else{
            et_msg.setText(message);
            sms.deletesms(id);
        }
        map.put("date", date);
        map.put("id", id);
        SMS.add(map);
    }
    cursor.close();
}

public void adapter_listview() {
    setListAdapter(adapterinbox);
    ListView lv = getListView();

    lv.setOnItemClickListener(new OnItemClickListener()
{
        public boolean onItemClick(AdapterView<?> arg0, View
view,int pos, long id) {
            makeDialog(view);
            return true;
        }
    });
}

public void onRefresh(String phone, String message){

```

```

        sms.putSmsToDatabase(phone, message);
        putData(thread_id);
        this.adapter_listview();
    }

    public void onResume() {
        super.onResume();
        putData(thread_id);
        adapterinbox = new ModifListSMS(this,
SMS,R.layout.listsms,new String[] {"id","msginbox","type","date"},
new int[] {R.id.id,R.id.inbox,R.id.type,R.id.date},name);
        this.adapter_listview();
        progDial.dismiss();
    }

    public void onFailedSend(String phone, String message,
String error, String help){
        NotificationManager notificationManager =
(NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE
);
        Notification.Builder builder = new
Notification.Builder(this);

        Intent intent = new Intent(getApplicationContext(),
SMSActivity.class);
        intent.putExtra("thread_id", thread_id);
        intent.putExtra("name", name);
        intent.putExtra("msg", message);

        PendingIntent pendingIntent =
PendingIntent.getActivity(getApplicationContext(), 0, intent, 0);
        builder
            .setSmallIcon(android.R.drawable.ic_dialog_alert)
            .setTicker(error)
            .setContentTitle(error)
            .setContentText(help)
            .setLights(0xFFFF0000, 500, 500) //setLights
(int argb, int onMs, int offMs)
            .setContentIntent(pendingIntent)
            .setAutoCancel(true);
        Notification notification =
builder.getNotification();

notificationManager.notify(android.R.drawable.ic_dialog_alert,
notification);
    }

    public void makeDialog(final View v){
        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        final String _id = ((TextView)
v.findViewById(R.id.id)).getText().toString();
        final String message =
((TextView)v.findViewById(R.id.inbox)).getText().toString();
        builder.setItems(items, new
DialogInterface.OnClickListener() {

```

```

        public void onClick(DialogInterface dialog, int item)
    {
        if(item==0){
            ClipboardManager c = (ClipboardManager)
getSystemService(CLIPBOARD_SERVICE);
            String whatyouaresearching =
message.substring(message.indexOf(":")+1,message.length());
            c.setText(whatyouaresearching);

            Toast.makeText(getApplicationContext(), "Text Copied",
Toast.LENGTH_SHORT).show();
        }else if(item==1){
            Intent in = new
Intent(getApplicationContext(), SMSNewActivity.class);
            String whatyouaresearching =
message.substring(message.indexOf(":")+1,message.length());
            in.putExtra("msg", whatyouaresearching);
            startActivity(in);
        }else{
            Bundle b = getIntent().getExtras();
            b.putString("id", _id);
            showDialog(DELETE_ALERT,b);
            ///sms.deletesms(_id);
        }
    }
    }).show();
}

@Override
protected Dialog onCreateDialog(int id,final Bundle b) {
    switch (id) {
        case DELETE_ALERT:{
            Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("Delete");

            builder.setIcon(android.R.drawable.ic_dialog_alert);
            builder.setMessage("Delete message ?");
            builder.setCancelable(true);
            builder.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface
dialog, int which) {

                    String id=b.getString("id");
                    sms.deletesms(id);
                    onResume();
                }
            });
            builder.setNegativeButton("No", new
DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface
dialog, int which) {

                    dialog.dismiss();

```

```

        }
    });
    AlertDialog dialog = builder.create();
    dialog.show();
}break;
case CHANGE_STATUS_ALERT:{
    Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Change Security");

    builder.setIcon(android.R.drawable.ic_dialog_alert);
    builder.setMessage("Are you Sure ? May your next
message is not secure");
    builder.setCancelable(true);
    builder.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface
dialog, int which) {

            menuItem.setIcon(R.drawable.device_access_not_secure);

            kdb.setStatus(id_contact, kdb.UNLOCK);
                                status="unlock";
        }
    });
    builder.setNegativeButton("No", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface
dialog, int which) {

            dialog.dismiss();
        }
    });
    AlertDialog dialog = builder.create();
    dialog.show();
}
}
return super.onCreateDialog(id);
}
}

```

SMSNewActivity.java

```

package aves.enkripsi.sms;
import java.util.ArrayList;
import java.util.List;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.app.AlertDialog.Builder;
import android.app.Dialog;

```



```

import android.content.ContentResolver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;

import android.provider.BaseColumns;
import android.provider.ContactsContract;

import android.text.InputFilter;
import android.text.SpannableStringBuilder;
import android.text.Spanned;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;

import android.widget.Toast;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.key.KeyListActivity;

public class SMSNewActivity extends SMS{
    public static final int DIALOG_SECURE=0;
    public static final int SECURE=0;
    public static final int not_SECURE=1;
    private static final int PICK_CONTACT = 0;
    int i=0;
    List<String> list = new ArrayList<String>();
    int mode;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.sms);
        setTitle("New SMS");
        getActionBar().setDisplayHomeAsUpEnabled(true);

        ////getActionBar().setIcon(android.R.drawable.ic_media_previous);

        New=(LinearLayout) findViewById(R.id.New);
        Old=(LinearLayout) findViewById(R.id.Old);
        New.setVisibility(LinearLayout.VISIBLE);
        Old.setVisibility(LinearLayout.INVISIBLE);

        et_msg = (EditText) findViewById(R.id.inputBody);
        bt_send = (Button) findViewById(R.id.SendButton);
        bt_add = (ImageButton) findViewById(R.id.bt_add);
        et_phn = (EditText) findViewById(R.id.et_phn);

        Bundle b = getIntent().getExtras();
        if(b.getString("msg").toString().length()>0){

```

```

        et_msg.setText(b.getString("msg").toString());
    }

    bt_add.setTag("ADD");
    bt_add.setOnClickListener(new
View.OnClickListener() {
    public void onClick(View v) {
        if(bt_add.getTag().toString()=="ADD") {
            if(mode==SECURE) {
                //Intent in = new
Intent(Intent.ACTION_PICK,ContactsContract.Contacts.CONTENT_URI);
                Intent in = new
Intent(SMSNewActivity.this, KeyListActivity.class);
                in.putExtra("pick", true);
                startActivityForResult(in,
PICK_CONTACT_SECURE);
            } else if(mode==not_SECURE) {
                Intent in = new
Intent(Intent.ACTION_PICK,ContactsContract.Contacts.CONTENT_URI);
                startActivityForResult(in,
PICK_CONTACT);
            } else{
            }
        } else{
            et_phn.setText("");
            phone=null;
            et_phn.setEnabled(true);

            bt_add.setImageResource(R.drawable.social_add_person);
            bt_add.setTag("ADD");
        }
    }
});

    bt_send.setOnClickListener(new
View.OnClickListener() {

    public void onClick(View v) {

        if(et_msg.getText().toString().length()==0) {
            Toast.makeText(getApplicationContext(),
"You Not Entry Any Text Message", Toast.LENGTH_SHORT).show();
        } else{
            if(phone==null && et_phn.length()!=0
){

                phone=et_phn.getText().toString();
            }
            String
id_contact=contact.getContactIDFromNumber(phone);
            String key=kdb.getKey(id_contact);
            String status;
            if(mode==SECURE) {

```

```

        status="lock";
    }else{
        status="unlock";
    }
    sendSMS (phone,
et_msg.getText().toString(),status,key);
    progDial=1.onThreadRun ("Sending
message.....");
    }
    });
    showDialog(DIALOG_SECURE);
}

@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case DIALOG_SECURE:
            Builder builder = new AlertDialog.Builder(this);
builder.setIcon(android.R.drawable.ic_dialog_alert);
            builder.setTitle("Choose your SMS mode or cancel
?");
            builder.setCancelable(false);
            builder.setPositiveButton("SECURE", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialog, int which) {
                    setTitle("NEW SMS SECURE");
                    et_phn.setEnabled(false);
                    et_phn.setFocusable(false);
                    mode=SECURE;
                    InputFilter filter = new
InputFilter() {
                        @Override
                        public CharSequence
filter(CharSequence source, int start, int end,
Spanned dest, int
dstart, int dend) {
                            if (source instanceof
SpannableStringBuilder) {
                                SpannableStringBuilder
sourceAsSpannableBuilder = (SpannableStringBuilder)source;
                                for (int i = end - 1; i >=
start; i--) {
                                    char currentChar =
source.charAt(i);
                                    int ascii = currentChar;
                                    if (ascii > 127) {
                                        sourceAsSpannableBuilder.delete(i, i+1);

```

```

Toast.makeText(getApplicationContext(), "Invalid non-Ascii
Character", Toast.LENGTH_SHORT).show();
    }
    }
    return source;
} else {
    StringBuilder
filteredStringBuilder = new StringBuilder();
    for (int i = 0; i < end; i++) {
        char currentChar =
source.charAt(i);
        int ascii = currentChar;
        if (ascii <= 127 &&
ascii>=0) {
filteredStringBuilder.append(currentChar);
        }
    }
    return
filteredStringBuilder.toString();
}
}
};
et_msg.setFilters(new
InputFilter[]{filter});
}
});

builder.setNeutralButton("NOT SECURE", new
DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface
dialog, int which) {
        setTitle("NEW SMS not SECURE");
        mode=not_SECURE;
        et_msg.setHint("Write your
message");
    }
});
builder.setNegativeButton("CANCEL", new
DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface
dialog, int which) {
        finish();
    }
});
AlertDialog dialog = builder.create();
dialog.show();
}
return super.onCreateDialog(id);

```

```

    }

    public void onRefresh(String phone, String message){
        sms.putSmsToDatabase(phone, message);
        String thread_id=sms.getThread_id(phone);
        name=contact.findNameByAddress(phone);
        Intent in = new Intent(getApplicationContext(),
SMSActivity.class);
        in.putExtra("thread_id", thread_id);
        in.putExtra("name", name);
        in.putExtra("msg", "");
        in.putExtra("type", "read");
        finish();
        startActivity(in);
    }

    public void ReturnContact(String nama,String string){
        if(!nama.isEmpty()){
            this.phone=string;
            et_phn.setText(nama+"("+string+")");
            et_phn.setEnabled(false);
            bt_add.setTag("X");

            bt_add.setImageResource(R.drawable.content_backspace);
        }
    }

    protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        list.clear();
        String nama="";
        if (requestCode == PICK_CONTACT && resultCode ==
Activity.RESULT_OK) {
            Uri contactData = data.getData();
            ContentResolver cr = getContentResolver();
            Cursor cur = managedQuery(contactData, null,
null, null, null);
            if (cur.getCount() > 0) {
                while (cur.moveToNext()) {
                    String id = cur.getString(cur

.getColumnIndex(BaseColumns._ID));
                    if (Integer

.parseInt(cur.getString(cur

.getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER)
) > 0) {

                        Cursor pCur = cr

.query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
null,

ContactsContract.CommonDataKinds.Phone.CONTACT_ID

```

```

+ " = ?", new String[] { id },
                                null);

                                while (pCur.moveToNext()) {
                                    String nomorHp = pCur

                                .getString(pCur

                                .getColumnIndex(ContactsContract.CommonDataKinds.Phone.DATA)
                                );
                                    nama =
                                pCur.getString(pCur.getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
                                    list.add(nomorHp);
                                }
                                makeDialogContact(nama, list);
                                pCur.close();
                            }
                        }
                    }else if(requestCode == PICK_CONTACT_SECURE &&
resultCode == Activity.RESULT_OK) {
                        super.onActivityResult(requestCode, resultCode,
data);

                        Bundle bundle = data.getExtras();
                        String id = bundle.getString("id");
                        Toast.makeText(getApplicationContext(), id,
Toast.LENGTH_SHORT).show();
                        contact = new ContactDB(this);

                        makeDialogContact(contact.findNameByID(id), contact.LoadDataKey(id));
                    }

                public void makeDialogContact(final String nama,
List<String> ls){
                    final CharSequence[] items = ls.toArray(new
CharSequence[ls.size()]);
                    if(ls.size()!=1){
                        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
                        builder.setItems(items, new
DialogInterface.OnClickListener() {
                            public void onClick(DialogInterface
dialog, int item) {

                                ReturnContact(nama, items[item].toString());
                            }
                        })
                        .show();
                    }else{
                        ReturnContact(nama, items[0].toString());
                    }
                }
            }
        }
    }
}

```

```

    }

    public void onFailedSend(String phone, String message,
String error, String help){
        NotificationManager notificationManager =
(NotificationManager) getSystemService (Context.NOTIFICATION_SERVICE
);
        Notification.Builder builder = new
Notification.Builder(this);

        Intent intent = new Intent(getApplicationConteXt(),
SMSNewActivity.class);
        intent.putExtra("msg", message);

        PendingIntent pendingIntent =
PendingIntent.getActivity(getApplicationConteXt(), 0, intent, 0);
        builder
            .setSmallIcon(android.R.drawable.ic_dialog_alert)
            .setTicker(error)
            .setContentTitle(error)
            .setContentText(help)
            .setLights(0xFFFF0000, 500, 500) //setLights
(int argb, int onMs, int offMs)
            .setContentIntent(pendingIntent)
            .setAutoCancel(true);
        Notification notification =
builder.getNotification();

notificationManager.notify(android.R.drawable.ic_dialog_alert,
notification);
    }
}

```

SMSRecieve.java

```

package aves.enkripsi.sms;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.SmsDB;
import aves.enkripsi.sms.data.KeyDB;
import aves.enkripsi.sms.key.SmsVigenere;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.util.Log;
import android.widget.Toast;

public class SmsReceiver extends BroadcastReceiver
{
    SmsDB s;
}

```

```

SmsVigenere v;
KeyDB kdb;
ContactDB cdb;

@Override
public void onReceive(Context context, Intent intent)
{
    //---get the SMS message passed in---
    Bundle bundle = intent.getExtras();
    SmsMessage[] msgs = null;
    String str = "";
    String pesan= "";
    if (bundle != null)
    {
        //---retrieve the SMS message received---
        Object[] pdus = (Object[]) bundle.get("pdus");
        msgs = new SmsMessage[pdus.length];
        for (int i=0; i<msgs.length; i++){
            msgs[i] =
SmsMessage.createFromPdu((byte[])pdus[i]);
            str += "SMS[] from " +
msgs[i].getOriginatingAddress();
            str += " :";
            str += msgs[i].getMessageBody().toString();
            str += "\n";
            pesan
+=msgs[i].getDisplayMessageBody().toString();
            Log.i("RECIEVE SMS",
msgs[i].getOriginatingAddress()+":"+msgs[i].getDisplayMessageBody(
));

        }

buildNotification(context,msgs[0].getOriginatingAddress(),pesan);
        Toast.makeText(context, str,
Toast.LENGTH_SHORT).show();

    }
}

private void buildNotification(Context context,String
number, String message){
    s=new SmsDB(context);
    kdb=new KeyDB(context);
    cdb=new ContactDB(context);
    v=new SmsVigenere();
    String id_contact=cdb.getContactIDFromNumber(number);

    if(v.enkripsiornot(message)){
        if(kdb.cekkey(id_contact)){
            String _id=s.getID(number);
            String psn=s.getMessage(number);
            String key=kdb.getkey(id_contact);
            message=v.smsDekrip(psn,key);

```



```

        s.Updatesms(_id, message);
    }
    NotificationManager notificationManager
    =
(NotificationManager) context.getSystemService(Context.NOTIFICATION
_SERVICE);
    Notification.Builder builder = new
Notification.Builder(context);

    Intent intent = new Intent(context, MainActivity.class);

    PendingIntent pendingIntent =
PendingIntent.getActivity(context, 0, intent, 0);
    builder
    .setSmallIcon(R.drawable.social_chat)
    .setContentTitle(""+cdb.findNameByAddress(number))
    .setContentText(""+message)
    .setTicker("New SMS from "+cdb.findNameByAddress(number))
    .setLights(0xFFFF0000, 500, 500) //setLights
(int argb, int onMs, int offMs)
    .setContentIntent(pendingIntent)
    .setAutoCancel(true);

    Notification notification = builder.getNotification();
    notification.flags|= Notification.FLAG_ONGOING_EVENT;

    notificationManager.notify(R.drawable.social_chat,
notification);
    }
}

```

MainFragment.java

```

package aves.enkripsi.sms.fragment;

import java.util.ArrayList;
import java.util.HashMap;

import android.app.AlertDialog;
import android.app.ListFragment;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemLongClickListener;
import android.widget.ImageView;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.TextView;
import aves.enkripsi.sms.Loading;

```

```

import aves.enkripsi.sms.R;
import aves.enkripsi.sms.SMSActivity;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.SmsDB;
import aves.enkripsi.sms.data.KeyDB;

public class MainFragment extends ListFragment {
    public static final ArrayList<HashMap<String, String>>
    SMSthread = new ArrayList<HashMap<String, String>>();
    final CharSequence[] items = {"View Conversations", "Delete
Conversations"};
    public static int NOT_SECURE=0;
    public static int SECURE=1;

    ListAdapter adapterthread;
    public String thread_id;
    SmsDB sms;
    KeyDB key;
    ContactDB contact;

    ProgressDialog progDial;
    Loading l;
    TextView et_key;
    String phnkey;
    ListView liv;

    public void adapter_listview(ListView lv) {
        setListAdapter(adapterthread);
        lv.setOnItemLongClickListener(new
OnItemLongClickListener() {
            public boolean onItemClick(AdapterView<?> arg0,
View view, int pos, long id) {
                makeDialog(view);
                return true;
            }
        });
    }

    public void makeDialog(final View view) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
        builder.setItems(items, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item)
            {
                if(item==0){
                    String thread_id = ((TextView)
view.findViewById(R.id.thread_id)).getText().toString();
                    String name =
((TextView)view.findViewById(R.id.name)).getText().toString();
                    ImageView
m=(ImageView)view.findViewById(R.id.icon);

```

```

SMSActivity.class);
        Intent in = new Intent(getActivity(),
        in.putExtra("thread_id", thread_id);
        in.putExtra("name", name);
        in.putExtra("msg", "");
        if(m.getTag().toString()=="unread"){
            in.putExtra("type", "unread");
            m.setTag("read");
        }else{
            in.putExtra("type", "read");
        }
        startActivity(in);
    }else{
        Log.e("error delete conversation :",
        "hahah");
        Cursor c=sms.GetInbox(thread_id);
        ProgressDialog p = new
Loading(getActivity()).onThreadRun("Deleting...");
        p.show();
        while(c.moveToNext()){
            sms.deletesms(c.getString(0));
        }
        c.close();
        p.dismiss();
    }
    }
    }).show();
}

@Override
public void onItemClick(ListView parent, View view, int
position, long id) {

        String thread_id = ((TextView)
view.findViewById(R.id.thread_id)).getText().toString();
        String name =
        ((TextView)view.findViewById(R.id.name)).getText().toString();
        ImageView
m=(ImageView)view.findViewById(R.id.icon);
        Intent in;
        String msg="";

        in = new Intent(getActivity(),
SMSActivity.class);
        in.putExtra("thread_id", thread_id);
        in.putExtra("name", name);
        in.putExtra("msg", msg);

        if(m.getTag().toString()=="unread"){
            in.putExtra("type", "unread");
        }else{
            in.putExtra("type", "read");
        }

        startActivity(in);
    }
}

```

```

    @Override
    public void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        key.close();
    }
}

```

AFragment.java

```

package aves.enkripsi.sms.fragment;

import java.util.HashMap;

import android.database.Cursor;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;
import aves.enkripsi.sms.Loading;
import aves.enkripsi.sms.R;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.SmsDB;
import aves.enkripsi.sms.data.KeyDB;

public class AFragment extends MainFragment {
    ListView lv;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.main, container,
false);
        sms=new SmsDB(getActivity());
        l=new Loading(getActivity());
        key=new KeyDB(getActivity());
        contact=new ContactDB(getActivity());
        adapterthread = new ModifListThread(getActivity(),
SMSThread,R.layout.listthread,new String[]
{"name", "lastsms", "count", "id"}, new int[] {R.id.name
,R.id.lastsms,R.id.count,R.id.thread_id});
        lv=(ListView)view.findViewById(android.R.id.list);

        return view;
    }

    public void onResume() {
        super.onResume();
    }
}

```

```

        putData();
        adapter_listview(lv);
    }

    public void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        key.close();
    }

    public void putData() {
        SMSthread.clear();
        Cursor c=sms.GetConversation();
        while(c.moveToNext()){
            String
id_contact=contact.getContactIDFromNumber(sms.getNumber(c.getStrin
g(0)));
                if(!key.cekStatus(id_contact)){
                    String
name=contact.findNameByAddress(sms.getNumber(c.getString(0)));
                    String
count=c.getString(c.getColumnIndexOrThrow("msg_count"));
                    String lastsms = c.getString(2);
                    String id = c.getString(0);
                    HashMap<String, String> map = new
HashMap<String, String>();
                    map.put("name", name);
                    map.put("lastsms", lastsms);
                    map.put("count", count);
                    map.put("id", id);
                    SMSthread.add(map);
                }
            }
        c.close();
    }
}

```

BFragment.java

```

package aves.enkripsi.sms.fragment;

import java.util.HashMap;

import android.database.Cursor;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListView;
import aves.enkripsi.sms.Loading;
import aves.enkripsi.sms.R;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.SmsDB;
import aves.enkripsi.sms.data.KeyDB;

```

```

public class AFragment extends MainFragment {
    ListView lv;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view=inflater.inflate(R.layout.main, container,
false);
        sms=new SmsDB(getActivity());
        l=new Loading(getActivity());
        key=new KeyDB(getActivity());
        contact=new ContactDB(getActivity());
        adapterthread = new ModifListThread(getActivity(),
SMSThread,R.layout.listthread,new String[]
{"name", "lastsms", "count", "id"}, new int[] {R.id.name
,R.id.lastsms,R.id.count,R.id.thread_id});
        lv=(ListView)view.findViewById(android.R.id.list);

        return view;
    }

    public void onResume() {
        super.onResume();
        putData();
        adapter_listview(lv);
    }

    public void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        key.close();
    }

    public void putData() {
        SMSThread.clear();
        Cursor c=sms.GetConversation();
        while(c.moveToNext()) {
            String
id_contact=contact.getContactIDFromNumber(sms.getNumber(c.getStrin
g(0)));
            if(!key.cekStatus(id_contact)) {
                String
name=contact.findNameByAddress(sms.getNumber(c.getString(0)));
                String
count=c.getString(c.getColumnIndexOrThrow("msg_count"));
                String lastsms = c.getString(2);
                String id = c.getString(0);
                HashMap<String, String> map = new
HashMap<String, String>();
                map.put("name", name);
                map.put("lastsms", lastsms);
                map.put("count", count);
                map.put("id", id);
                SMSThread.add(map);
            }
        }
    }
}

```

```

        }
        c.close();
    }
}

```

ContactDB.java

```

package aves.enkripsi.sms.data;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.provider.ContactsContract.PhoneLookup;
import android.util.Log;

public class ContactDB extends Activity{
    String nama;
    String nomor;
    String id;
    Context context;

    public ContactDB(Context context){
        this.context=context;
    }

    public String getNama(){
        if(nama==null){
            return "";
        }
        return this.nama;
    }

    public String getNomer(){
        if(nomor==null){
            return "";
        }
        return this.nomor;
    }

    public String findNameByAddress(String addr)
    {
        Uri myPerson =
        Uri.withAppendedPath(ContactsContract.CommonDataKinds.Phone.CONTENT_FILTER_URI,

```

```

        Uri.encode(addr));
        String[] projection = new String[] {
ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME };
        Cursor cursor =
context.getContentResolver().query(myPerson,
        projection, null, null, null);
        if (cursor.moveToFirst()) {
            String
name=cursor.getString(cursor.getColumnIndex(ContactsContract.Common
nDataKinds.Phone.DISPLAY_NAME));
            Log.e("", "Found contact name");
            cursor.close();
            return name;
        }

        cursor.close();
        Log.e("", "Not Found contact name");
        return addr;
    }

    public String findNameByID(String id){
        String nama="This Contact is not available you must
deleted";
        Cursor phoneCur =
context.getContentResolver().query(ContactsContract.CommonDataKind
s.Phone.CONTENT_URI, null,
            ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String[] { id }, null);

        if(phoneCur.moveToFirst())
        {
            nama =
phoneCur.getString(phoneCur.getColumnIndex(ContactsContract.Common
DataKinds.Phone.DISPLAY_NAME));

        }else{
            KeyDB k=new KeyDB(context);
            k.deletekey(id);
        }
        phoneCur.close();
        return nama;
    }

    public String getContactIDFromNumber(String contactNumber)
    {
        contactNumber = Uri.encode(contactNumber);
        int phoneContactID = new Random().nextInt();
        Cursor contactLookupCursor =
context.getContentResolver().query(Uri.withAppendedPath(PhoneLooku
p.CONTENT_FILTER_URI,Uri.encode(contactNumber)),new String[]
{PhoneLookup.DISPLAY_NAME, PhoneLookup._ID}, null, null, null);
        while (contactLookupCursor.moveToNext()) {

```



```

        phoneContactID =
contactLookupCursor.getInt (contactLookupCursor.getColumnIndexOrThrow (PhoneLookup._ID));
    }
    contactLookupCursor.close ();
    return String.valueOf (phoneContactID);
}

public String getNumberbyID (String id) {
    String nomor = "";
    Cursor phoneCur =
context.getContentResolver ().query (ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String [] { id }, null);

    if (phoneCur.moveToFirst ())
    {
        nomor =
phoneCur.getString (phoneCur.getColumnIndex (ContactsContract.CommonDataKinds.Phone.NUMBER));
    }
    phoneCur.close ();
    return nomor;
}

public String getID () {
    if (id == null) {
        return "";
    }
    return this.id;
}

public List <String> LoadDataKey (String id) {
    String name = "";
    String no = "";
    List <String> listItems = new ArrayList <String> ();
    ContentResolver contact_resolver =
context.getContentResolver ();

    Cursor phoneCur =
contact_resolver.query (ContactsContract.CommonDataKinds.Phone.CONTENT_URI, null,
ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String [] { id }, null);

    while (phoneCur.moveToNext ())
    {
        name =
phoneCur.getString (phoneCur.getColumnIndex (ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
        no =
phoneCur.getString (phoneCur.getColumnIndex (ContactsContract.CommonDataKinds.Phone.NUMBER));
    }
}

```

```

        listItems.add(no);
    }

    this.nama=name;
    this.id=id;
    phoneCur.close();
    return listItems;
}
}

```

KeyDB.java

```

package aves.enkripsi.sms.data;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

import android.app.Activity;
import android.content.ContentResolver;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.provider.ContactsContract;
import android.provider.ContactsContract.PhoneLookup;
import android.util.Log;

public class ContactDB extends Activity{
    String nama;
    String nomor;
    String id;
    Context context;

    public ContactDB(Context context){
        this.context=context;
    }

    public String getNama(){
        if(nama==null){
            return "";
        }
        return this.nama;
    }

    public String getNomer(){
        if(nomor==null){
            return "";
        }
        return this.nomor;
    }

    public String findNameByAddress(String addr)

```

```

    {
        Uri myPerson =
Uri.withAppendedPath(ContactsContract.CommonDataKinds.Phone.CONTENT_
T_FILTER_URI,
                Uri.encode(addr));
        String[] projection = new String[] {
ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME };
        Cursor cursor =
context.getContentResolver().query(myPerson,
                projection, null, null, null);
        if (cursor.moveToFirst()) {
            String
name=cursor.getString(cursor.getColumnIndex(ContactsContract.Commo
nDataKinds.Phone.DISPLAY_NAME));
            Log.e("", "Found contact name");
            cursor.close();
            return name;
        }

        cursor.close();
        Log.e("", "Not Found contact name");
        return addr;
    }

    public String findNameByID(String id){
        String nama="This Contact is not available you must
deleted";
        Cursor phoneCur =
context.getContentResolver().query(ContactsContract.CommonDataKind
s.Phone.CONTENT_URI, null,
                ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String[] { id }, null);

        if(phoneCur.moveToFirst())
        {
            nama =
phoneCur.getString(phoneCur.getColumnIndex(ContactsContract.Common
DataKinds.Phone.DISPLAY_NAME));
        }else{
            KeyDB k=new KeyDB(context);
            k.deletekey(id);
        }
        phoneCur.close();
        return nama;
    }

    public String getContactIDFromNumber(String contactNumber)
    {
        contactNumber = Uri.encode(contactNumber);
        int phoneContactID = new Random().nextInt();
        Cursor contactLookupCursor =
context.getContentResolver().query(Uri.withAppendedPath(PhoneLooku

```

```

p.CONTENT_FILTER_URI,Uri.encode(contactNumber)),new String[]
{PhoneLookup.DISPLAY_NAME, PhoneLookup._ID}, null, null, null);
        while (contactLookupCursor.moveToNext()) {
            phoneContactID =
contactLookupCursor.getInt (contactLookupCursor.getColumnIndexOrThr
ow(PhoneLookup._ID));
        }
        contactLookupCursor.close();
        return String.valueOf(phoneContactID);
    }

    public String getNumberbyID(String id){
        String nomor="";
        Cursor phoneCur =
context.getContentResolver().query(ContactsContract.CommonDataKind
s.Phone.CONTENT_URI, null,
        ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String[] { id }, null);

        if (phoneCur.moveToFirst())
        {
            nomor =
phoneCur.getString (phoneCur.getColumnIndex (ContactsContract.Common
DataKinds.Phone.NUMBER));

        }
        phoneCur.close();
        return nomor;
    }

    public String getID() {
        if(id==null){
            return "";
        }
        return this.id;
    }

    public List<String> LoadDataKey(String id) {
        String name = "";
        String no = "";
        List<String> listItems = new ArrayList<String>();
        ContentResolver contact_resolver =
context.getContentResolver();

        Cursor phoneCur =
contact_resolver.query(ContactsContract.CommonDataKinds.Phone.CONT
ENT_URI, null,
        ContactsContract.CommonDataKinds.Phone.CONTACT_ID
+ " = ?", new String[] { id }, null);

        while (phoneCur.moveToNext())
        {
            name =
phoneCur.getString (phoneCur.getColumnIndex (ContactsContract.Common
DataKinds.Phone.DISPLAY_NAME));

```

```

        no =
phoneCur.getString(phoneCur.getColumnIndex(ContactsContract.Common
DataKinds.Phone.NUMBER));
        listItems.add(no);
    }

    this.nama=name;
    this.id=id;
    phoneCur.close();
    return listItems;
}
}

```

SMSDB.java

```

package aves.enkripsi.sms.data;

import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.util.Log;
import aves.enkripsi.sms.Loading;

public class SmsDB{
    Uri CON = Uri.parse("content://sms/conversations/");
    Uri SMS = Uri.parse("content://sms/");
    Uri INBOX = Uri.parse("content://sms/inbox/");

    static Context context;

    public SmsDB (Context c)
    {
        context =c;
    }

    public Cursor GetInbox(String thread_id){
        Cursor inbox = context.getContentResolver().query(SMS,
null,"thread_id="+thread_id, null,"date");
        return inbox;
    }

    public boolean cekMessageUnread(String thread_id){
        Cursor c=context.getContentResolver().query(INBOX,
null, "read = 0 AND thread_id="+thread_id, null, null);
        if(c.getCount()>0){
            c.close();
            return true;
        }
        c.close();
        return false;
    }
}

```

```

public void setReadThread(String thread_id){
    Cursor c=GetInbox(thread_id);
    while (c.moveToNext()) {

if(Boolean.valueOf(c.getString(c.getColumnIndex("read")))){

        }else{

setRead(c.getString(c.getColumnIndex("_id")));
        }
    }
    c.close();
}

protected void setRead(String _id) {
    ContentValues values = new ContentValues();
    values.put("read", true);
    context.getContentResolver().update(INBOX, values,
    "_id="+_id, null);
}

public Cursor GetConversation(){
    Cursor conversations =
context.getContentResolver().query(CON, null, null, null, "date
desc");
    return conversations;
}

public void deletesms(String _id){
    try{
        context.getContentResolver().delete(SMS,
    "_id=?", new String[] {_id});
    } catch(Exception e)
    {
        Log.e("error delete sms :", e.toString());
    }
}

public void deleteconversation(String thread_id){
    try{
        Log.e("error delete conversation :", "hahah");
        Cursor c=GetInbox(thread_id);
        ProgressDialog p = new
Loading(context).onThreadRun("Deleting...");
        while(c.moveToNext()){

deletesms(c.getString(c.getColumnIndex("_id")));
        }
        c.close();
        p.dismiss();
    }
    catch (Exception e)
    {
        Log.e("error delete conversation :", e.toString());
}
}

```

```

    }
}

public void InsertSent(String phone, String message){
    ContentValues values = new ContentValues();
    values.put("address", phone);
    values.put("body", message);

    context.getContentResolver().insert(Uri.parse("content://sms
/sent"), values);
}

public void Updatesms(String _id, String message){
    ContentValues values = new ContentValues();
    values.put("body", message);
    context.getContentResolver().update(SMS,
values, "_id="+_id, null);
}

public String getID(String number){
    String id=null;
    Cursor c=context.getContentResolver().query(INBOX,
null, "address=?", new String[]{number}, "DATE DESC");
    if(c.moveToFirst()){
        id=c.getString(c.getColumnIndex("_id"));
        c.close();
        return id;
    }
    c.close();
    return id;
}

public String getNumber(String thread_id){
    String number="";
    Cursor c=context.getContentResolver().query(SMS,
null, "thread_id="+thread_id, null, null);
    if(c.moveToNext()){
        number=c.getString(c.getColumnIndex("address"));
        c.close();
    }
    c.close();
    return number;
}

public void putSmsToDatabase(String no,String sms)
{
    if(!no.isEmpty()&&!sms.isEmpty()){
        ContentValues values = new ContentValues();
        values.put("address", no);
        values.put("body", sms);

        context.getContentResolver().insert(Uri.parse("content://sms
/sent"), values );
    }
}

```

```

    }
}

    public String getThread_id(String number) {
        Cursor cursor =
context.getContentResolver().query(SMS, null, "address=?", new
String[]{number}, null);
        String thread_id="";
        if(cursor.moveToFirst()){

thread_id=cursor.getString(cursor.getColumnIndex("thread_id"
));
            cursor.close();
            return thread_id;
        }
        cursor.close();
        return thread_id;
    }

    public String getDraftFromThread(String thread_id) {
        String message="";
        Cursor c=GetInbox(thread_id);
        if(c.moveToLast()){
            c.moveToLast();

            if(c.getString(c.getColumnIndex("type")).toString().equalsIgnoreCase("3")) {

                message=c.getString(c.getColumnIndex("body"));
                context.getContentResolver().delete(SMS,
"_id="+c.getString(0).toString(), null);

            }
        }
        c.close();
        return message;
    }

    public Boolean cekTypeDraft(String thread_id) {
        Cursor c=GetInbox(thread_id);
        if(c.moveToLast()){
            c.moveToLast();

            if(c.getString(c.getColumnIndex("type")).toString().equalsIgnoreCase("3") &&
c.getString(c.getColumnIndex("address")).toString().length()==0) {
                return true;
            }
        }
        c.close();
        return false;
    }

    public String getMessage(String number) {
        String msg=null;

```



```

        Cursor c=context.getContentResolver().query(INBOX,
null, "address=?", new String[]{number}, "DATE DESC");
        if(c.moveToFirst()){
            msg=c.getString(c.getColumnIndex("body"));
            c.close();
            return msg;
        }
        c.close();
        return msg;
    }
}

```

SMSVigenere.java

```

package aves.enkripsi.sms.key;

public class SmsVigenere {
    private static final String HEAD_KEY="(O-#)";
    FibonacciGenerated fb=new FibonacciGenerated();

    public String smsEnkrip(String message, String key){
        key=new String(fb.generatedKey(key.getBytes(),
message.getBytes()));
        message= new String(enkrip(message.getBytes(),
key.getBytes()));

        return HEAD_KEY+message;
    }

    public String smsDekrip(String message, String key){
        key=new String(fb.generatedKey(key.getBytes(),
message.getBytes()));

        message=message.substring(message.indexOf(HEAD_KEY)+HEAD_KEY
.length(),message.length());
        message=new String(dekrip(message.getBytes(),
key.getBytes()));
        return message;
    }

    public boolean enkripsiornot(String msg){
        if(msg.indexOf(HEAD_KEY)==0){
            return true;
        }

        return false;
    }

    protected byte[] enkrip(byte[] input,byte[] kunci){
        byte[] output = new byte[input.length];
        byte buffer;
        byte bufferKey;
        for(int i=0;i<input.length;i++){
            buffer=input[i];

```

```

        bufferKey=kunci[i%kunci.length];
        buffer=(byte)((buffer+bufferKey));
        while(buffer<0){
            buffer=(byte)(128+buffer);
        }
        output[i]=buffer;
    }
    return output;
}

protected byte[] dekrip(byte[] input, byte[] kunci){
    byte [] output= new byte[input.length];
    byte buffer,bufferKey;
    for(int i=0;i<input.length;i++){
        buffer=input[i];
        bufferKey=kunci[i%kunci.length];
        buffer=(byte)((buffer-bufferKey));
        while(buffer<0){
            buffer=(byte)(128+buffer);
        }
        output[i]=buffer;
    }
    return output;
}
}

```

FibonnaciGenerated.java

```

package aves.enkripsi.sms.key;

import android.util.Log;

public class FibonnaciGenerated {
    public int getm(byte [] key){
        int m=0;
        int panjang=key.length;
        byte b;
        int jumlah=0;
        for(int i=0;i<panjang;i++){
            b=key[i];
            jumlah=jumlah+b-65;
        }
        m=1+(jumlah%(panjang-1));
        return m;
    }

    public byte[] generatedKey(byte [] key, byte [] message){
        int keyL=key.length;
        int msgL=message.length;
        byte[] kunci =message;
        Log.e("error", "nang kene");

        if(keyL>msgL){
            msgL=keyL;
            kunci=key;
        }
    }
}

```

```

    }

    int m=getm(key);

    for(int i=0;i<keyL;i++){
        kunci[i]=key[i];
    }

    for(int i=keyL;i<msgL;i++){
        kunci[i]=(byte) ((kunci[i-keyL]+kunci[i-keyL+m]) %
26);
        kunci[i]=(byte) (kunci[i] + 65);
    }
    return kunci;
}
}

```

KeyActivity.java

```

package aves.enkripsi.sms.key;

import java.util.ArrayList;
import java.util.List;

import android.annotation.SuppressLint;
import android.app.ActionBar.LayoutParams;
import android.app.Activity;
import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.content.ContentResolver;

import android.content.DialogInterface;
import android.content.Intent;

import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.BaseColumns;
import android.provider.ContactsContract;
import android.text.InputFilter;
import android.text.Spanned;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.KeyDB;
import aves.enkripsi.sms.*;

```

```

@SuppressLint("DefaultLocale")
public class KeyActivity extends Activity {
    public static final int PICK_CONTACT = 1;
    public static final int EDIT = 1;
    public static final int ADD = 0;

    List<String> list = new ArrayList<String>();

    Button bt_save;
    ImageButton bt_add;
    EditText et_key;
    EditText et_phn;
    String phone;
    String id_contact;
    Loading l;
    ProgressDialog progDial;

    int mode;

    KeyDB kdb;
    ContactDB contact;

    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        requestWindowFeature(Window.FEATURE_LEFT_ICON);
        setContentView(R.layout.saveeditkey);

        bt_save = (Button) findViewById(R.id.bt_keysave);
        bt_add =
        (ImageButton) findViewById(R.id.bt_addcontactkey);
        et_key = (EditText) findViewById(R.id.et_key);
        et_phn = (EditText) findViewById(R.id.et_phnkey);

        l=new Loading(this);
        kdb=new KeyDB(this);
        contact=new ContactDB(this);

        android.view.WindowManager.LayoutParams params =
        getWindow().getAttributes();
        params.width = LayoutParams.FILL_PARENT;

        getWindow().setFeatureDrawableResource(Window.FEATURE_LEFT_ICON,
        android.R.drawable.ic_lock_idle_lock);

        getWindow().setBackgroundDrawableResource(android.R.color.background_dark);

        getWindow().setAttributes((android.view.WindowManager.LayoutParams
        ) params);
        bt_save.setText("SAVE");

        Bundle b = getIntent().getExtras();

```

```

        if(b.getString("action").toString().equalsIgnoreCase("edit")
    ){
        mode=EDIT;
        setTitle("Edit Key
"+b.getString("nama").toString());
        id_contact=b.getString("contact_id").toString();
        contact.loadDataKey(id_contact);
        et_key.setText(b.getString("key").toString());
    }else{
        mode=ADD;
        setTitle("Add Key");
    }

    bt_add.setTag("ADD");
    bt_add.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {

            if(bt_add.getTag().toString().equalsIgnoreCase("ADD")){
                Intent in = new
Intent(Intent.ACTION_PICK, ContactsContract.Contacts.CONTENT_URI);
                startActivityForResult(in,
PICK_CONTACT);
            }else{
                et_phn.setText("");
                phone="";
                bt_add.setTag("ADD");
            }

            bt_add.setImageResource(R.drawable.social_add_person);
        }
    });

    bt_save.setOnClickListener(new OnClickListener() {
        @SuppressWarnings("DefaultLocale")
        @Override
        public void onClick(View v) {
            if(et_key.getText().toString().length()>0
&& id_contact.length()>0 && !id_contact.isEmpty() &&
id_contact!=null){
                if(mode==ADD){

                    kdb.insert(id_contact,et_key.getText().toString().toUpperCase
e());
                }else if(mode==EDIT) {

                    kdb.replace(id_contact,et_key.getText().toString().toUpperCase
se());
                }else{

                    Toast.makeText(getApplicationContext(), "You can't leave
blank :"+id_contact, Toast.LENGTH_SHORT).show();
                }
            }else{

```



```

        null);
        if(pCur.moveToFirst()) {
            String nomorHp = pCur

                .getString(pCur

                .getColumnIndex(ContactsContract.CommonDataKinds.Phone.DATA)
            );

            String name =
pCur.getString(pCur.getColumnIndex(ContactsContract.Contacts.DISPL
AY_NAME));

            if(!nomorHp.isEmpty()){

                ReturnContact(name, id);

            }

        }

        pCur.close();

    }

}

    public void makeDialogContact(final String nama,
List<String> ls){
        final CharSequence[] items = ls.toArray(new
CharSequence[ls.size()]);
        if(ls.size()!=1){
            AlertDialog.Builder builder = new
AlertDialog.Builder(this);
            builder.setItems(items, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface
dialog, int item) {

                    ReturnContact(nama,items[item].toString());

                }
            })
            .show();
        }else{
            ReturnContact(nama,items[0].toString());
        }

    }

    private void ReturnContact(String nama, String id_contact) {
        et_phn.setText(nama);
        this.id_contact=id_contact;
        bt_add.setTag("X");
        bt_add.setImageResource(R.drawable.content_backspace);
    }

}

```

KeyListActivity.java

```

package aves.enkripsi.sms.key;

import java.util.ArrayList;
import java.util.HashMap;

import android.app.AlertDialog;
import android.app.AlertDialog.Builder;
import android.app.Dialog;
import android.app.ListActivity;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.AdapterView.OnItemClickListener;
import aves.enkripsi.sms.R;
import aves.enkripsi.sms.data.ContactDB;
import aves.enkripsi.sms.data.KeyDB;

public class KeyListActivity extends ListActivity{
    public static final ArrayList<HashMap<String, String>> KEY =
new ArrayList<HashMap<String, String>>();
    final CharSequence[] items = {"View Details", "Edit Contact
Key", "Delete Contact Key"};
    ListAdapter adapterkey;
    KeyDB kdb;
    ContactDB ckb;
    Button bt_addkey;
    public static final int DELETE_ALERT=0;

    public void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.key);

        kdb=new KeyDB(this);
        ckb=new ContactDB(this);

        bt_addkey=(Button) findViewById(R.id.addkey);

        bt_addkey.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Intent i = new
Intent(getApplicationContext(), KeyActivity.class);
                i.putExtra("action", "add");

```



```

        startActivity(i);
    }
    });
}

public void onResume() {
    super.onResume();
    putData();
    adapterkey = new
ModifListKEY(this, KEY, R.layout.listkey, new
String[]{"id_contact", "nama", "key"}, new int[]
{R.id.id_contact, R.id.keynama, R.id.nilai_key});
    this.adapter_listview();
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    kdb.close();
}

public void putData() {
    KEY.clear();
    Cursor cursor = kdb.getAll();
    while (cursor.moveToNext()) {
        String id_contact=cursor.getString(0);
        String nama=ckb.findNameByID(id_contact);
        String key=cursor.getString(1);
        HashMap<String, String> map = new
HashMap<String, String>();
        map.put("id_contact", id_contact);
        map.put("nama", nama);
        map.put("key", key);
        KEY.add(map);
    }
    cursor.close();
}

public void adapter_listview() {
    setListAdapter(adapterkey);
    ListView lv=getListView();
    lv.setOnItemClickListener(new OnItemClickListener() {

        @Override
        public void onItemClick(AdapterView<?> arg0,
View arg1, int arg2,
                long arg3) {
            Bundle bundle = getIntent().getExtras();
            if(bundle.getBoolean("pick")){
                TextView
id=(TextView)arg1.findViewById(R.id.id_contact);
                Intent i = new Intent();

                i.putExtra("id", id.getText().toString());
                setResult(RESULT_OK, i);
            }
        }
    });
}

```

```

        finish();
    }
}

});
lv.setOnItemClickListener(new
OnItemClickListener() {
    public boolean onItemClick(AdapterView<?>
arg0, View view,int pos, long id) {
        makeDialog(view);
        return true;
    }
});
}

    public void makeDialog(final View v){
        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        final String _id = ((TextView)
v.findViewById(R.id.id_contact)).getText().toString();
        final String nama =
((TextView)v.findViewById(R.id.keynama)).getText().toString();
        final String key
=((TextView)v.findViewById(R.id.nilai_key)).getText().toString();
        builder.setItems(items, new
DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item)
{
                if(item==1){
                    Intent i = new
Intent(getApplicationContext(), KeyActivity.class);
                    i.putExtra("action", "edit");
                    i.putExtra("contact_id", _id);
                    i.putExtra("nama", nama);
                    i.putExtra("key", key);
                    startActivity(i);
                }else if(item==2){
                    Bundle args = new Bundle();
                    args.putString("id", _id);
                    showDialog(DELETE_ALERT, args);
                }else if(item==0){
                    Intent i = new Intent(Intent.ACTION_VIEW,
Uri.withAppendedPath(ContactsContract.Contacts.CONTENT_URI,
""+_id));
                    startActivity(i);
                }else{
                }
            }
        }).show();
    }
}
@Override
protected Dialog onCreateDialog(int id,final Bundle b) {
    switch (id) {
        case DELETE_ALERT:
            Builder builder = new AlertDialog.Builder(this);

```

```
        builder.setTitle("Delete");
        builder.setIcon(android.R.drawable.ic_dialog_alert);
        builder.setMessage("Delete contact key?");
        builder.setCancelable(true);
        builder.setPositiveButton("Yes", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int
which) {

                String id=b.getString("id");
                kdb.deletekey(id);
                onResume();
            }
        });
        builder.setNegativeButton("No", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int
which) {

                dialog.dismiss();
            }
        });
        AlertDialog dialog = builder.create();
        dialog.show();
    }
    return super.onCreateDialog(id);
}
}
```

LAMPIRAN B

DAFTAR PENGUJI BETA

DAFTAR PENGUJI BETA

No	Nama	Pekerjaan	Instansi	Vendor	Type Smartphone	No Telp
1	Ali Alhadi	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia mini	085747500070
2	Arif Setiawan	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Hisense	Eg909	085729074620
3	Saifullah A P	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia U	0857127332338
4	Faisal Rohman	Mahasiswa	Prodi HI, Fak. Syariah, UIN Sunan Kalijaga	Sony	Xperia tipo	085643417224
5	Mulyono	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia U	085764321 993
6	Muh Setiawan	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Samsung	Galaxy Mini	08995108875
7	Husni Tamrin	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia U	085747463677
8	M. Alfian Jauhari	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia ray	085729333493
9	Fiska N H	Wiraswasta	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	Sony	Xperia x8	085643396569
10	Fathan Tri K	Mahasiswa	Prodi TIF, Fak. Saintek,UIN Sunan Kalijaga	LG	Optimus	08995453190

LAMPIRAN C

QUISSONNER

PENGUJIAN ALPHA APLIKASI ENKRIPSI SMS

Nama : Syaifullah

Pendidikan : T. Informatika UIN SUKA

No.	Pertanyaan	Ya	Tidak
1.	Apakah sistem sudah sesuai dengan Activity Diagram ?	√	
2.	Apakah sistem sudah sesuai dengan Usecase Diagram ?	√	
3.	Apakah sistem sudah sesuai dengan batasan masalah ?	√	
4.	Apakah enkripsi sudah berjalan sesuai metode yang digunakan ?	√	
5.	Apakah sistem dapat mengirim dan menerima pesan dengan benar baik pesan terenkripsi maupun tidak ?	√	

PENGUJIAN ALPHA APLIKASI ENKRIPSI SMS

Nama : Mulyono

Pendidikan : T. Informatika UIN SUKA

No.	Pertanyaan	Ya	Tidak
1.	Apakah sistem sudah sesuai dengan Activity Diagram ?	√	
2.	Apakah sistem sudah sesuai dengan Usecase Diagram ?	√	
3.	Apakah sistem sudah sesuai dengan batasan masalah ?	√	
4.	Apakah enkripsi sudah berjalan sesuai metode yang digunakan ?	√	
5.	Apakah sistem dapat mengirim dan menerima pesan dengan benar baik pesan terenkripsi maupun tidak ?	√	

PENGUJIAN ALPHA APLIKASI ENKRIPSI SMS

Nama : Arif Setiawan

Pendidikan : T. Informatika UIN SUKA

No.	Pertanyaan	Ya	Tidak
1.	Apakah sistem sudah sesuai dengan Activity Diagram ?	√	
2.	Apakah sistem sudah sesuai dengan Usecase Diagram ?	√	
3.	Apakah sistem sudah sesuai dengan batasan masalah ?	√	
4.	Apakah enkripsi sudah berjalan sesuai metode yang digunakan ?	√	
5.	Apakah sistem dapat mengirim dan menerima pesan dengan benar baik pesan terenkripsi maupun tidak ?	√	

PENGUJIAN ALPHA APLIKASI ENKRIPSI SMS

Nama : Ali Alhadi

Pendidikan : T. Informatika UIN SUKA

No.	Pertanyaan	Ya	Tidak
1.	Apakah sistem sudah sesuai dengan Activity Diagram ?	√	
2.	Apakah sistem sudah sesuai dengan Usecase Diagram ?	√	
3.	Apakah sistem sudah sesuai dengan batasan masalah ?	√	
4.	Apakah enkripsi sudah berjalan sesuai metode yang digunakan ?	√	
5.	Apakah sistem dapat mengirim dan menerima pesan dengan benar baik pesan terenkripsi maupun tidak ?	√	

PENGUJIAN ALPHA APLIKASI ENKRIPSI SMS

Nama : Muh. Setiawan

Pendidikan : T. Informatika UIN SUKA

No.	Pertanyaan	Ya	Tidak
1.	Apakah sistem sudah sesuai dengan Activity Diagram ?	√	
2.	Apakah sistem sudah sesuai dengan Usecase Diagram ?	√	
3.	Apakah sistem sudah sesuai dengan batasan masalah ?	√	
4.	Apakah enkripsi sudah berjalan sesuai metode yang digunakan ?	√	
5.	Apakah sistem dapat mengirim dan menerima pesan dengan benar baik pesan terenkripsi maupun tidak ?	√	

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Muh. Setiawan

Pekerjaan : Mahasiswa

Type HP : Galaxy mini

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)			√	
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik	√			
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim	√			
5.	Sistem dapat mendekripsikan pesan yang masuk	√			
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk	√			
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk	√			
8.	Hasil dekripsi sesuai dengan pesan asli	√			
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Fiska Nuraida Hanifah

Pekerjaan : Wiraswasta

Type HP : Sony Xperia x8

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)			√	
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Alfian Jauhari

Pekerjaan : Mahasiswa

Type HP : Sony Xperia ray

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)	√			
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk	√			
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Faisal Rohman

Pekerjaan : Mahasiswa

Type HP : Sony Xperia tipo

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik	√			
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli	√			
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci	√			

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Fathan Trikurniawan

Pekerjaan : Mahasiswa

Type HP : LG P705

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)			√	
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk	√			
8.	Hasil dekripsi sesuai dengan pesan asli	√			
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Mulyono

Pekerjaan : Mahasiswa

Type HP : Sony Xperia U

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik	√			
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk	√			
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli	√			
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Syaifullah A P

Pekerjaan : Mahasiswa

Type HP : Sony Xperia U

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Ali Alhadi

Pekerjaan : Mahasiswa

Type HP : Sony Xperia mini

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Arif Setiawan

Pekerjaan : Wiraswasta

Type HP : Smartfren Andromax i

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik		√		
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk		√		
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk		√		
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk		√		
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci		√		

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

PENGUJIAN BETA APLIKASI ENKRIPSI SMS

Nama : Husni Tamrin

Pekerjaan : Mahasiswa

Type HP : Sony Xperia U

No.	Pertanyaan	SS	S	TS	STS
1.	Tampilan aplikasi tidak membingungkan (User-friendly)		√		
2.	Sistem dapat mengirimkan pesan dan menerima pesan dengan baik	√			
3.	Sistem dapat mengirimkan pesan terenkripsi maupun tidak		√		
4.	Sistem dapat mengenkripsikan pesan yang dikirim		√		
5.	Sistem dapat mendekripsikan pesan yang masuk	√			
6.	Ada notifikasi pesan ketika pesan terkirim dan pesan diterima dan ketika pesan masuk	√			
7.	Sistem dapat menyimpan dan menampilkan pesan keluar dan masuk	√			
8.	Hasil dekripsi sesuai dengan pesan asli		√		
9.	Sistem dapat mengambil data dari Phonebook Device ketika mengirim pesan atau menambahkan kunci	√			

Keterangan:

SS : Sangat Setuju

S : Setuju

TS : Tidak Setuju

STS : Sangat Tidak Setuju

LAMPIRAN D
CURRICULUM VITAE



Nama : Nur Avesina Mustari

Tempat, Tanggal Lahir : Kebumen, 10 April 1990

Jenis Kelamin : Laki-laki

Nama Ayah : Sartono S.Pd

Nama Ibu : Munawarotun S.Pd

Alamat : Perum Pepabri Boro Kulon RT. 02 RW .05 No.26
Kec. Banyuurip Kab. Purworejo Prov. Jawa Tengah
54171

No. HP : +6285729000936

Email : nuravesinamustari@gmail.com

Riwayat Pendidikan : 1996-2002 : SD 2 Popongan - Purworejo
2002-2005 : SMP Negeri 6 Purworejo – Purworejo
2005-2008 : SMA Negeri 1 Purworejo – Purworejo
2008-2013 : Program Studi Teknik
Informatika, Fakultas Sains dan
Teknologi, Universitas Islam Negeri Sunan Kalijaga
Yogyakarta