Penerapan Algoritma Vigenere Cipher pada Aplikasi SMS Android

Andi Kurniawan Dwi P. / 13508028
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If18028@students.if.itb.ac.id

***Abstrak*—Perkembangan teknologi di bidang komunikasi semakin tahun semakin maju. Salah satu hasil perkembangan teknologi di bidang komunikasi adalah layanan SMS (*Short Message Service)*. Semua jenis telepon seluler (*Hand Phone)* pasti mempunyai layanan komunikasi ini. Sebagian besar orang lebih sering menggunakan layanan SMS daripada layanan telepon karena biaya nya yang tergolong murah dan mudah digunakan. Namun, sayangnya pesan yang dikirimkan melalui SMS tidak dapat dijamin integritas dan keamanannya. Seseorang yang mengirimkan pesan yang bersifat personal atau rahasia tentunya menginginkan pesan tersebut hanya dapat dibaca oleh orang yang diinginkannya. Oleh karena itu dibutuhkan suatu sistem keamanan yang mampu menjaga kerahasiaan pesan yang dikirimkan melalui SMS. Dalam hal ini, ilmu kriptografi dapat diimplementasikan dalam membangun sistem keamanan. Pesan yang akan dikirimkan melalui SMS sebaiknya dilakukan enkripsi terlebih dahulu dengan algoritma tertentu, misalnya menggunakan Vigenere Cipher. Kemudian ketika pesan tersebut tersampaikan kepada penerima, penerima harus melakukan dekripsi dengan kunci yang sama agar pesan yang diterimanya dapat terbaca maknanya. Orang lain yang ingin membaca isi pesan tersebut tidak akan bisa apabila tidak memiliki kunci yang sama. Aplikasi kemanan pada layanan SMS ini dapat diterapkan pada platform Android. Telepon seluler dengan platform Android dipilih karena Android sedang berada di puncak penjualan telepon seluler, sehingga pengguna Android dapat dikatakan banyak.**

***Kata Kunci*— Android, Dekripsi, Enkripsi, Kriptografi, SMS, Vigenere Cipher.**

I. PENDAHULUAN

Beberapa tahun ini perkembangan teknologi komunikasi semakin pesat. Telepon seluler merupakan salah satu hasil dari perkembangan terknologi komunikasi. Telepon seluler mempermudah orang untuk berkomunikasi satu sama lain. Dengan adanya teknologi ini dunia terasa sempit karena seseorang dapat berkomunikasi dengan orang lain yang jaraknya jauh. Di dalam telepon seluler ini ada beberapa fungsi komunikasi yang dapat digunakan antara lain telepon, *video call*, SMS, MMS, *chatting*, internet, dan lain-lain. Di antara layanan komunikasi tersebut, layanan SMS yang menjadi komunikasi favorit karena sudah dipastikan semua telepon seluler memiliki layanan ini dan yang paling penting adalah biayanya yang tergolong murah.

Sayangnya SMS tidak menjamin integritas dan keamanan pesan yang disampaikan. Pesan yang bersifat personal atau rahasia tidak dijamin sampai ke penerima tanpa dicuri informasinya oleh orang lain. Ada beberapa risiko yang dapat mengancam keamanan pesan pada layanan SMS antara lain SMS *spoofing*, SMS *snooping*, dan SMS *interception*.

SMS *spoofing* merupakan pengiriman sms di mana nomor pengirim yang tertera bukanlah nomer pengirim yang sebenarnya. Mekanisme SMS *spoofing* ini dimungkinkan karena lemahnya proteksi koneksi SMSC-gateway. Penyusup dapat merekam login dan pasword dari pesan yang berasal dari SMS gateway menuju SMSC. Walaupun tak terlalu mudah namun ini dapat dilakukan dalam beberapa kasus.Dalam hal ini penyusup mengatur sebuah gateway palsu yang berlaku seperti gateway sesungguhnya. Gateway palsu ini dapat mengirim semua jenis pesan pendek kepada user MS melalui SMSC. Pada teknik spoofing ini pesan dikirim dengan memanipulasi nomor MSISDN asal (originate) pada field yang disediakan sehingga pesan akan tampak datang dari nomor pengirim lainnya. Kemungkinan spoofing yang lain adalah dengan membuat simulator SMSC yang berlaku seperti SMSC asli. Dengan cara ini gateway akan kebanjiran pesan, sebagai contoh aplikasi bank menggunakan gateway dapat dengan mudah diperoleh informasi account bahkan dapat digunakan untuk transaksi bank tanpa proses authorisasi.

Ancaman SMS lainnya adalah SMS *snooping*. SMS *snooping* lebih sering terjadi karena kelalaian pengguna telepon seluler. Contohnya ketika seseorang meminjamkan telepon selulernya pada orang lain untuk menggunakan telepon selulernya. Pada saat itu orang tersebut dapat dengan sengaja atau tidak membuka isi pesan yang ada pada *inbox* SMS. Pesan yang bersifat personal atau rahasia dapat dibaca dengan mudah oleh orang lain melalui cara ini.

Celah keamanan terbesar pada layanan komunikasi SMS adalah pada saat SMS tersebut sedang dikirim melalui jaringan SMS tersebut. SMS bekerja pada jaringan nirkabel yang memungkinkan terjadinya

pencurian isi pesan SMS ketika dalam proses transmisi dari pengirim ke penerima. Kasus ini disebut SMS *interception.*

Dibutuhkan sebuah sistem keamanan pada layanan SMS yang mampu menjaga integritas dan keamanan isi pesan untuk menutupi celah keamanan SMS (terutama untuk SMS *snooping* dan SMS *interception*). Agar isi pesan hanya dapat dibaca maknanya oleh pengirim dan penerima, isi pesan sebelum dikirim melalui SMS harus dienkripsi terlebih dahulu dengan algoritma kriptografi, misalnya Vigenere Cipher. Penerima dapat membaca makna dari pesan tersebut dengan melakukan dekripsi isi pesan tersebut menggunakan kunci yang sama yang digunakan oleh pengirim. Apabila ada orang lain yang mencuri isi pesan tersebut, orang tersebut tidak akan mampu membaca makna pesan tersebut. Pesan yang dicurinya tidak akan memiliki makna karena dalam kondisi terenkripsi. Dengan adanya sistem keamanan ini isi pesan yang bersifat personal atau rahasia dapat tersampaikan secara aman.

II. DASAR TEORI

A. SMS (Short Message Service)

SMS (*Short Message Service*) merupakan sebuah layanan komunikasi yang ada pada telepon seluler untuk mengirim dan menerima pesan-pesan pendek. SMS pertama kali dikenalkan pada tanggal 3 Desember 1982. SMS pertama di dunia dikirimkan menggunakan jaringan GSM milik operator telepon bernama Vodafone. SMS pertama ini dikirimkan oleh ahli bernama Neil Papwort kepada Richard Jarvis menggunakan komputer.

SMS dihantarkan pada *channel signal* GSM (*Global System for Mobile Communication*) dengan spesifikasi teknis ETSI. SMS diaktifkan oleh ETSI dan dijalankan di *scope* 3GPP. SMS juga digunakan pada teknnologi GPRS dan CDMA. SMS menjamin pengiriman pesan oleh jaringan, jika terjadi kegagalan pesan akan disimpan dahulu di jaringan dan akan dikirmkan lagi ketika jaringan sudah stabil.

B. Platform Android

Android adalah sistem operasi untuk telepon seluler berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membangun aplikasi yang dapat dijalankan di bermacam telepon seluler. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru dalam teknologi telepon seluler. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Aliance menyatakan mendukung pengembangan standar terbuka pada telepon seluler.

Fitur yang tersedia di Android antara lain:

- *Framework* aplikasi yang mendukung penggantian komponen dan *reusable*
- *Dalvik virtual machine*
- *Integrated browser*
- *Grafik* berdasarkan OpenGL
- *SQLite* untuk penyimpanan data
- *Multimedia support*
- Linkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

C. Kriptografi

Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita. Selain pengertian tersebut terdapat pula pengertian ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tidak semua aspek keamanan informasi ditangani oleh kriptografi.

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

- Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/ mengupas informasi yang telah disandi.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
- Autentikasi, adalah berhubungan dengan identifikasi/ pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
- Non-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/ terciptanya suatu informasi oleh yang mengirimkan/ membuat.

D. Vigenere Cipher

Vigenere Cipher adalah salah satu jenis kriptografi kalsi yang pada dasarnya melakukan subtitusi cipher abjad majemuk (*polyalphabetic substitution).* Metode ini pertama kali dipublikasikan oleh seorang diplomat (sekaligus esorang kriptologis) Prancis, Blaise de Vigenere pada abad ke-16, tepatnya pada tahun 1586. Sebenarnya Giovan Batista Belaso telah menggambarkan algoritma ini pertama kali pada tahun 1553 seperti ditulis

di dalam buku *La Cifra del Sig*. metode Vigenere Cipher ini berhasil dipecahkan oleh matematikawan Inggris Charles Babbage dan Kasiski pada pertengahan abad 19. Vigenere Cipher ini digunakan oleh tentara konfiderasi pada perang sipil Amerika. Perang sipil akhirnya berhasil dihentikan setelah Vigenere Cipher berhasil dipecahkan.

Di metode kriptografi klasik Caesar Cipher, setiap huruf alpahabet akan disubtitusi sepanjang 3 huruf sesudah huruf teresbut. Contoh, huruf A akan diganti dengan hurf D, B akan diganti dengan huruf E, Y akan diganti dengan huruf B dengan metode Caesar Cipher. Vigenere Cipher ini menerapkan prinsip Caesar Cipher dalam metode enkripsinya.

Untuk memudahkan dalam proses enkripsi, maka dapat digunakan alat bantu berupa bujur sangkar Vigenere.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 1. Bujur Sangkar Vigenere

Baris pada gambar 1 menyatakan huruf plainteks yang akan dienkripsi dan kolom menyatakan huruf kunci enkripsi. Perpotongan antara baris dan kolom menyatakan huruf yang sudah ternekripsi atau diistilahkan dengan cipherteks.

Jika panjang kunci lebih pendek daripada panjang plainteks, maka kunci akan diulang secra periodic.

Contoh:

Kunci	: sony
Plainteks	: THIS PLAINTEXT
Kunci pada saat enkripsi	: sony sonysonys
Cipherteks	: LVVQ HZNGFHRVL

Pada dasarnya, setiap enkripsi hurfu adalah Caesar Cipher dengan kunci yang berbeda-beda.

c(‘T’) = (‘T’+’s’) mod 26 = L

c(‘H’) = (‘H’+’o’) mod 26 = V, dst.

Jadi dengan Vigenere Cipher, huruf yang sama pada palainteks tidak selalu dienkripsi menjadi huruf cipherteks yang sama pula. Contoh huruf plainteks T dapat dienkripsi menjadi L atau J, huruf cipherteks V dapat merepresentasikan huruf plainteks H, I, dan X. Hal ini merupakan karakterisitik dari cipher abjad majemuk di mana setiap huruf plainteks dapat memiliki kemungkinan banyak huruf plainteks. Hal ini berbeda dengan cipher substitusi sederhana di mana setiap huruf cipherteks selalu menggantikan huruf plainteks tertentu.

Vigenere Cipher yang akan dipakai pada aplikasi ini adalah Vigenere Cipher extended di mana enkripsi tidak hanya untuk huruf alphabet tetapi termsauk juga karakter-karakter ASCII. Jadi batas pengekripsian tidak terbatas untuk 26 karakter tetapi mencapai 256 karakter.

III. Analisis dan Implementasi Vigenere Cipher Pada Aplikasi SMS Android

A. Analisis Kebutuhan

Aplikasi KriptoSMS ini digunakan untuk mengirim dan menerima pesan melalui SMS. Pesan yang akan dikirimkan melalui SMS terlebih dahulu dienkripsi dengan menggunakan Vigenere Cipher extended, dari proses enkripsi ini akan diperoleh cipherteks. Cipherteks inilah yang akan dikirimkan ke penerima melalui SMS. Untuk dapat membaca isi makna dari pesan tersebut penerima harus mendekripsi cipherteks dengan kunci yang sama.

Dalam membangun aplikasi KriptoSMS ini diperlukan batasan yang jelas agar aplikasi yang dibangun tidak keluar dari rencana awal. Beberapa kebutuhan sistem yang akan didefinisikan antara lain:

- Memiliki kemampuan untuk mengirimkan dan menerima pesan melalui SMS
- Memiliki kemampuan untuk mengenkripsi pesan
- Memiliki kemampuan untuk mendekripsi pesan
- Memiliki kemampuan untuk membaca pesan masuk yang ada di dalam *inbox*

Karena aplikasi ini tergolong aplikasi sederhana maka tidak dibutuhkan suatu kondisi yang rumit. Berikut adalah gambaran sistem secara umum:

- Pengirim akan mengirim pesan melalui layanan SMS
- Pesan ini akan dienkripsi terlebih dahulu sebelum dikirim
- Pesan ini akan dikirim berupa pesan teks (SMS)
- Pesan ini nantinya akan diterima oleh penerima pesan dalam keadaan terenkripsi
- Karena pesan yang diterima dalam keadaan terenkripsi maka harus ada pendekripsi pesan supaya pesan yang diterima memiliki makna.

Cara kerja sistem ini akan dibagi ke dalam beberapa proses utama. Proses ini dibagi menjadi empat tahapan yaitu enkripsi pesan, pengiriman pesan, pembacaan pesan, dan dekripsi pesan. Untuk lebih jelasnya dapat dilihat pada gambar 2.

Gambar 2. Alur Proses

B. Komponen yang Digunakan

Implementasi aplikasi KriptoSMS pada makalah ini dilakukan pada perangkat lunak Eclipse Helios dengan bahasa Java dan dibangun pada platform Android 2.2.

C. Implementasi

Implementasi aplikasi KriptoSMS ini dibagi menjadi 6 kelas yaitu kelas Global, KriptoSMS, SMSCryptor, SMSSender, SMSReader, dan SMSDecrypt.

1. Kelas Global

   Kelas ini hanya berfungsi untuk menyimpan variabel global. Terdiri dari 2 variabel static yang bertipe String. Variabel static ini menyimpan String nomor pengirim SMS dan isi pesan yang akan dihasilkan pada SMSReader untuk diproses pada SMSDecrypt.

2. Kelas KriptoSMS

   Pada kelas ini diimplementasikan *interface* untuk tampilan menu utama. *Interface* pada kelas ini dapat dilihat pada gambar 3. Pada menu utama ini terdapat 2 pilihan menu yaitu Kirim SMS dan Baca SMS. Ketika tombol Kirim SMS ditekan maka akan memanggil *interface* yang ada pada kelas SMSSender dan apabila tombol Baca SMS ditekan maka akan memanggil *interface* yang ada pada kelas SMSReader.

Gambar 3. *Interface* Menu Utama

3. Kelas SMSCryptor

   Di dalam kelas ini terdapat 2 prosedur static yaitu Enkripsi dan Dekripsi. Kedua prosedur ini dibuat dengan metode Vigenere Cipher extended. Berikut adalah psudocode untuk kedua prosedur tersebut:

```
public static String Enkripsi256AutoKey(String plaintext, String key) {
    String s = "";
    int lengthPlainText = plaintext.length();
    int lengthKey = key.length();
    int j = 0;
    for (int i = 0; i < lengthPlainText; i++) {
        if (j >= lengthKey) {
            key = plaintext;
            lengthKey = lengthPlainText;
            j = 0;
        }
        s += (char) (((int) plaintext.charAt(i) + key.charAt(j)) % 256);
        j++;
    }
    return s;
}
```

Gambar 4. Pseudocode Enkripsi Vigenere Cipher extended

```
public static String Dekripsi256AutoKey(String ciphertext, String key) {
    String s = "";
    int lengthCipherText = ciphertext.length();
    int lengthKey = key.length();
    int j = 0;
    int temp1;
    int a = 0;
    for (int i = 0; i < lengthCipherText; i++) {
        if (j >= lengthKey) {
            temp1 = (int) ciphertext.charAt(i) - (int) s.charAt(a);
            if (temp1 < 0) {
                temp1 += 256;
            }
            s += (char) (temp1 % 256);
```

```
                a++;
        } else {
                temp1 = (int) ciphertext.charAt(i) -
(int) key.charAt(j);
                if (temp1 < 0) {
                        temp1 += 256;
                }
                s += (char) (temp1 % 256);
        }
        j++;
    }
    return s;
}
```

Gambar 5. Pseudocode Dekripsi Vigenere Cipher extended

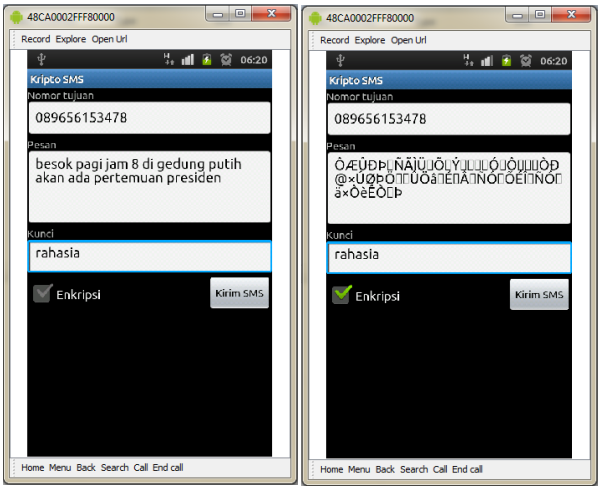
4. Kelas SMSSender

Kelas ini berfungsi untuk menampung input user berupa nomor tujuan, isi pesan, dan kunci. Kemudian isi pesan dienkripsi dengan kunci menggunakan prosedur static enkripsi yang ada pada kelas SMSCryptor. Setelah terbentuk plainteks maka pesan dikirimkan melalui SMS. Pada kelas ini terdapat prosedur utama untuk mengirim isi pesan melalui layanan SMS yaitu prosedur sendSMS. Pseudocode untuk prosedur ini dapat dilihat pada gambar 6.

```
   private void sendSMS(String phoneNumber, String
message)
   {
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";
    PendingIntent sentPI =
PendingIntent.getBroadcast(this, 0,
        new Intent(SENT), 0);
    PendingIntent deliveredPI =
PendingIntent.getBroadcast(this, 0,
        new Intent(DELIVERED), 0);

    //---when the SMS has been sent---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent
arg1) {
                switch (getResultCode())
                {
                        case
Activity.RESULT_OK:

    Toast.makeText(getBaseContext(), "SMS sent",
Toast.LENGTH_SHORT).show();
                                break;
                        case
SmsManager.RESULT_ERROR_GENERIC_FAILURE:

    Toast.makeText(getBaseContext(), "Generic failure",
Toast.LENGTH_SHORT).show();
                                break;
                        case
SmsManager.RESULT_ERROR_NO_SERVICE:

    Toast.makeText(getBaseContext(), "No service",
Toast.LENGTH_SHORT).show();
                                break;
                        case
SmsManager.RESULT_ERROR_NULL_PDU:

    Toast.makeText(getBaseContext(), "Null PDU",
Toast.LENGTH_SHORT).show();
                                break;
                        case
SmsManager.RESULT_ERROR_RADIO_OFF:

    Toast.makeText(getBaseContext(), "Radio off",
Toast.LENGTH_SHORT).show();
                                break;
                }
        }
    }, new IntentFilter(SENT));

    //---when the SMS has been delivered---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0,Intent
arg1) {
                switch (getResultCode())
                {
                        case
Activity.RESULT_OK:

    Toast.makeText(getBaseContext(), "SMS delivered",
Toast.LENGTH_SHORT).show();
                                break;
                        case
Activity.RESULT_CANCELED:

    Toast.makeText(getBaseContext(), "SMS not
delivered", Toast.LENGTH_SHORT).show();
                                break;
                }
        }
    }, new IntentFilter(DELIVERED));
    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message,
sentPI, deliveredPI);
   }
```

Gambar 6. Pseudocode SendSMS

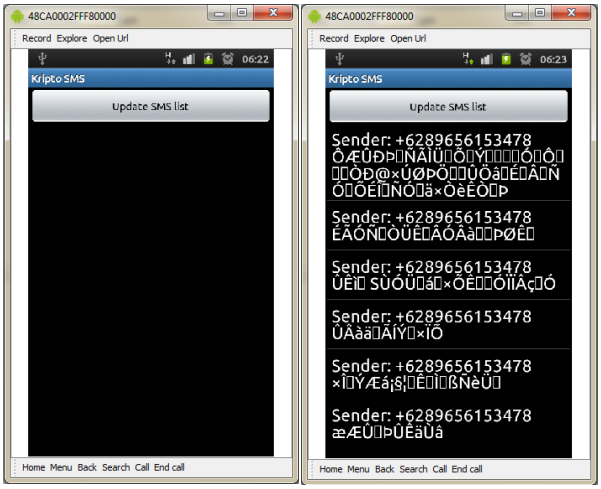
Interface yang ada pada kelas ini dapat dilihat pada gambar 7. Dapat dilihat terdapat 3 buah *textbox* yang menampung input pengguna. Input yang diterima antara lain nomor tujuan, isi pesan, dan kunci yang akan digunakan untuk mengenkripsi pesan. Secara *default* isi pesan ditampilkan dalam plainteks, namun ketika *checkbox* Enrkipsi dipilih maka isi pesan akan menjadi cipherteks. Pesan akan dikirim melalui layanan SMS apabila tombol Kirim SMS ditekan. Sistem akan memberikan notifikasi ketika sms terkirim dan diterima.



Gambar 7. Interface untuk menulis pesan

5. Kelas SMSReader

Kelas ini berfungsi membaca semua pesan yang disimpan di dalam inbox telepon seluler kemudian menampilkannya di dalam aplikasi KriptoSMS. Interface pada kelas ini dapat dilihat pada gambar 8.



Gambar 8. Interface Inbox KriptoSMS

Pertama kali masuk pada tampilan inbox KriptoSMS hanya terdapat tombol update SMS list. Ketika tombol ditekan maka sistem akan menampilkan pesan yang disimpan di inbox telepon seluler. Apabila salah satu isi pesan ditekan maka sistem akan memanggil *interface* untuk mendekripsi isi pesan.

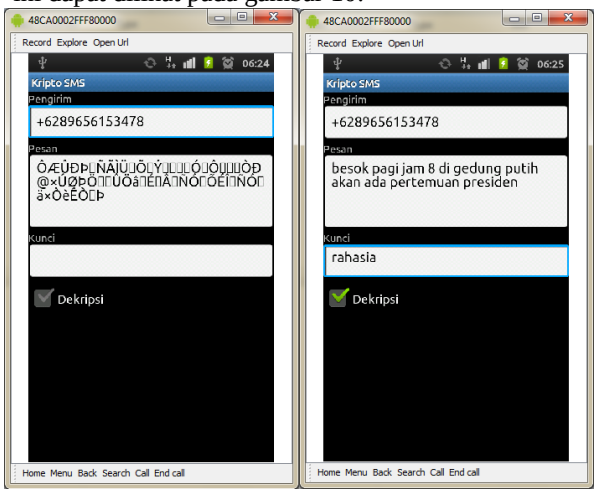
Di dalam kelas ini terdapat prosedur utama yang berfungsi untuk menampilkan list pesan yang ada di dalam inbox. Prosedur ini dapat dilihat pada gambar 9.

```
public void onClick( View v )
{
 ContentResolver contentResolver = getContentResolver();
 Cursor cursor = contentResolver.query( Uri.parse( "content://sms/inbox" ), null, null, null, null);
 int indexBody = cursor.getColumnIndex( "body");
 int indexAddr = cursor.getColumnIndex( "address" );
 if ( indexBody < 0 || !cursor.moveToFirst() ) return;
 smsList.clear();
 do
 {
  String str = "Sender: " + cursor.getString( indexAddr ) + "\n" + cursor.getString( indexBody );
  smsList.add( str );
 }
 while( cursor.moveToNext() );
 ListView smsListView = (ListView) findViewById( R.id.SMSList );
 smsListView.setAdapter( new ArrayAdapter<String>( this, android.R.layout.simple_list_item_1, smsList) );
 smsListView.setOnItemClickListener( this );
}
```

Gambar 9. Pseudocode untuk menampilkan list pesan

6. Kelas SMSDecrypt

Kelas ini berfungsi mendekripsi pesan yang ada di dalam inbox. Pertama-tama sistem meminta pengguna untuk memasukkan kunci yang akan digunakan untuk mendekripsi pesan. Kemudian ketika pengguna memilih checkbox Dekripsi maka isi pesan akan didekripsi dengan menggunakan prosedur dekripsi yang ada pada kelas SMSCryptor. *Interface* pada kelas ini dapat dilihat pada gambar 10.



Gambar 10. Interface Dekripsi pesan

IV. PENGUJIAN

Pengujian dilakukan pada telepon seluler Android Samsung Galaxy GT-i9003 di dalamnya menggunakan firmware Android 2.3. Pengujian terdiri dari 4 bagian utama yaitu pengujian enkripsi pesan, pengiriman SMS, pembacaan SMS, dan dekripsi pesan. Pengujian ini dilakukan dengan mengirimkan pesan pada nomor sendiri, sehingga dapat dibaca kembali isi pesan yang diterima.

Selama dilakukan pengujian dengan mengirim pesan terenkripsi secara berulang-ulang dengan isi pesan dan kunci yang berbeda-beda, tidak ditemukan masalah. Semua fungsi dapat berjalan dengan baik.

V. SIMPULAN DAN SARAN

A. Simpulan

Pesan yang bersifat personal atau rahasia tidak aman jika dikirimkan melalui aplikasi SMS biasa. Orang lain dapat dengan mudah mencuri informasi dari SMS tersebut dengan cara *snooping* maupun *interception*. Untuk mengatasi celah keamanan pada layanan SMS ini dibutuhkan aplikasi SMS yang mampu mengenkripsi dan mendekripsi isi pesan SMS, sehingga hanya orang yang memiliki kunci yang sama yang dapat membaca makna dari pesan.

KriptoSMS adalah aplikasi SMS yang dibangun untuk mengatasi masalah kemanan pada layanan SMS. Aplikasi ini mampu berjalan dengan baik pada platform Android. Aplikasi ini mempunyai fungsi menulis pesan, mengenkripsi pesan, mengirim pesan melalui SMS, membaca pesan yang ada pada inbox telepon seluler, dan mendekripsi pesan.

B. Saran

Untuk perbaikan dan pengembangan aplikasi KriptoSMS lebih lanjut disarankan sebagai berikut:

- Aplikasi dapat menyimpan SMS terkirim dan SMS yang belum dikirim
- Aplikasi menampilkan isi SMS masuk dan keluar dalam bentuk *thread* seperti aplikasi SMS bawaan Android
- Aplikasi didesign dengan interface yang menarik

DAFTAR PUSTAKA

[1] Rinaldi Munir, Situs Perkuliahan Kriptografi, http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2010-2011/kripto10-11.htm

[2] Nazruddin Safaat, "Android, Pemrograman Aplikasi Mobile Smartphone dan tablet PC Berbasis Android", Bandung: Penerbit Informatika, 2009.

[3] http://mobiforge.com/developing/story/sms-messaging-android

[4] http://www.apriorit.com/our-company/dev-blog/227-handle-sms-on-android

[5] http://gabohong.blogspot.com/2012/01/inilah-sejarah-sms-pertamakali-di-dunia.html

[6] http://octianaeni.blogspot.com/2011/11/pengenalan-android.html

[7] http://id.wikipedia.org/wiki/Kriptografi

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Maret 2012

Andi Kurniawan Dwi P.
13508028