JURNAL TEKNIK KOMPUTER AMIK BSI

IMPLEMENTASI APLIKASI ENKRIPSI SHORT MESSAGE SERVICE (SMS) BERBASIS ANDROID

Aries Gumilar Pratama[1], Anton[2], Firmansyah[3]

Abstract*—Android smartphones are very popular today because it is rich in features, ranging from multimedia, gaming applications also practically very varied, but behind a myriad of features android smartphone still has some of the same functionality as a regular phone, and it arguably can not be separated because it is basic functions of a mobile phone, one of which is a feature of the Short Message Service (SMS) which serves to send and receive short messages. function default SMS application is still frequently used, in this thesis the author makes SMS encryption application, it is intended that the contents of confidential messages can be delivered safely without having to worry about the content of the message is known by others. Methods for securing the contents of the message that is using encryption methods. Encryption is a process of converting an original message (plaintext) into a special code that can not be read and unintelligible (ciphertext), the message is the one that will be sent by SMS to the sender using the key, and to be able to read or restore contents of the messages that have been encrypted need a decryption process, the process is carried out by the SMS recipient by entering the same key with a key that is used by the sender. This method is also called Symmetric Algorithms, the algorithm uses the same key for both encryption and decryption process.

***Intisari*—** Smartphone Android dewasa ini sangat populer karena kaya akan fitur, mulai dari multimedia, game juga aplikasi yang bisa dibilang sangat bervariatif, namun dibalik segudang fitur smartphone android masih memiliki beberapa fungsi yang sama dengan ponsel biasa, dan hal ini bisa dibilang tidak bisa terlepas karena merupakan fungsi dasar dari sebuah ponsel, salah satunya adalah fitur Short Message Service (SMS) yang berfungsi untuk mengirim dan menerima pesan singkat. fungsi aplikasi SMS bawaan ini masih sering digunakan, pada tugas akhir ini penulis membuat aplikasi SMS enkripsi, hal ini dimaksudkan agar isi pesan yang bersifat rahasia dapat dikirim dengan aman tanpa harus khawatir isi pesan tersebut diketahui oleh orang lain. Metode untuk mengamankan isi pesan tersebut yaitu menggunakan metode enkripsi. Enkripsi merupakan suatu proses mengubah suatu pesan asli (plaintext) menjadi sebuah kode-kode khusus yang tidak bisa terbaca dan tidak dapat dimengerti (ciphertext), pesan ini lah yang nantinya akan dikirim oleh si pengirim SMS dengan menggunakan kunci, dan untuk dapat membaca atau mengembalikan isi pesan yang telah dienkripsi tersebut diperlukan suatu proses dekripsi, proses ini dilakukan oleh si penerima SMS dengan cara memasukan kunci yang sama dengan kunci yang digunakan oleh si pengirim. Metode ini disebut juga Symmetric Algorithms, yaitu algoritma menggunakan kunci yang sama baik untuk proses enkripsi maupun untuk proses dekripsi.

Kata Kunci: Android, Enkripsi, Dekripsi, SMS

[1,2,3] *Program Studi Teknik Komputer AMIK BSI Jakarta, Jln. RS. Fatmawati No. 24 Jakarta Selatan DKI Jakarta telp: (021) 31908575 fax:021-31908565; email: ariespratama96@bsi.ac.id; anton@bsi.ac.id; firmansyah.fmy@bsi.ac.id*

I. PENDAHULUAN

Handphone, sekarang ini bisa dibilang sudah merupakan kebutuhan yang wajib, dimana hampir setiap orang memilikinya. Handphone bukan lagi berperan sebagai alat komunikasi saja, sekarang ini sebuah handphone sudah melebihi fungsi dasarnya. Berbagai macam fitur telah ditanamkan didalamnya, seperti pengolah gambar dan video, pengolah dokumen dan lain sebagainya. Hal ini tak lepas dari penggunaan sistem operasi pada handphone. Layaknya pada komputer, handphone pun dapat di instal berbagai macam aplikasi yang diinginkan.

Android sebagai sistem operasi berbasis linux yang dapat digunakan di berbagai perangkat mobile. Android memiliki tujuan utama untuk memajukan inovasi perangkat mobile agar pengguna mampu mengeksplorasi kemampuan dan menambah pengalaman lebih dibandingkan dengan platform mobile lainnya. Hingga saat ini Android terus berkembang, baik secara sistem maupun aplikasinya. Smartphone android dewasa ini sangat populer dikarenakan begitu banyak fitur yang tersedia didalamnya.

Akan tetapi dibalik pesatnya perkembangan smartphone android ini tetap ada fitur-fitur dasar yang masih sering digunakan oleh para penggunanya, salah satunya adalah Short Message Service (SMS) yang berfungsi untuk mengirim dan menerima pesan, walaupun sudah banyak aplikasi chating yang relatif lebih cepat dan praktis, fungsi aplikasi SMS bawaan ini masih sering digunakan, pada tugas akhir ini penulis membuat aplikasi SMS enkripsi, hal ini dimaksudkan agar isi pesan yang bersifat rahasia dapat dikirim dengan aman tanpa harus khawatir isi pesan tersebut diketahui oleh orang lain. Metode untuk mengamankan isi pesan tersebut, yaitu menggunakan metode enkripsi.

Enkripsi merupakan suatu proses mengubah suatu pesan asli (plaintext) menjadi sebuah kode-kode khusus yang tidak bisa terbaca dan tidak dapat dimengerti (ciphertext), untuk dapat membaca atau mengembalikan isi pesan yang telah dienkripsi tersebut diperlukan suatu proses dekripsi. Salah satu metode enkripsi yang umum digunakan yaitu algoritma menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi (Symmetric Algorithms).

Maksud dari penelitian ini adalah :

1. Membuat aplikasi untuk mengamankan isi pesan yang bersifat privasi dari pihak-pihak yang tidak diinginkan.
2. Memberikan pemahaman tentang pentingnya mengamankan data pribadi
3. Meningkatkan pemahaman tentang struktur dan sistem kerja dalam pengembangan aplikasi pada sistem operasi Android.

II. KAJIAN LITERATUR

a. Android
"Android merupakan sebuah sistem operasi berbasis linux yang didesain khusus untuk perangkat bergerak seperti smartphone atau tablet" [5].
b. SMS (Short Message Service)
"Suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telpon selular" [7]. "Short Message Service (SMS) adalah protokol layanan pertukaran pesan text singkat (sebanyak 160 karakter per pesan) antar telepon" [1]
c. Kriptografi
"Kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan ditransfer dari suatu tempat ke tempat lain, isi dari pesan tersebut kemungkinan dapat disadap oleh pihak lain. Untuk menjaga keamanan pesan, pesan tersebut di-*scramble*/diacak atau diubah menjadi kode yang tidak dapat dimengerti oleh orang lain" [5].
d. Enkripsi
"Enkripsi merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli disebut plaintext (teks biasa), yang diubah menjadi kode-kode yang tidak dimengerti (ciphertext)" [5].
e. Dekripsi
"Dekripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk aslinya" [5].
f. Ciphertext
"Merupakan suatu pesan yang telah melalui proses enkripsi. Pesan ini tidak dapat dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti)" [5].
g. Plaintext
"Merupakan pesan yang ditulis atau diketik yang memiliki makna. Teks asli inilah yang diproses menggunakan algoritma kriptografi untuk menjadi ciphertext. Plaintext ini juga sering disebut cleartext (teks biasa)" [5].
h. Flowchart
"Bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Flowchart digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi dalam membuat suatu algoritma." [6].
i. Diagram HIPO (Hierarchy Input Proses Output)
paket yang berisi sebuah set diagram secara grafis menjelaskan fungsi sebuah sistem dari tingkat umum ke tingkat khusus" [3].

III. METODE PENELITIAN

Dalam membuat aplikasi dan untuk memudahkan pengumpulan data-data yang diperlukan dalam penelitian ini, maka penulis menggunakan metode penelitian sebagai berikut:

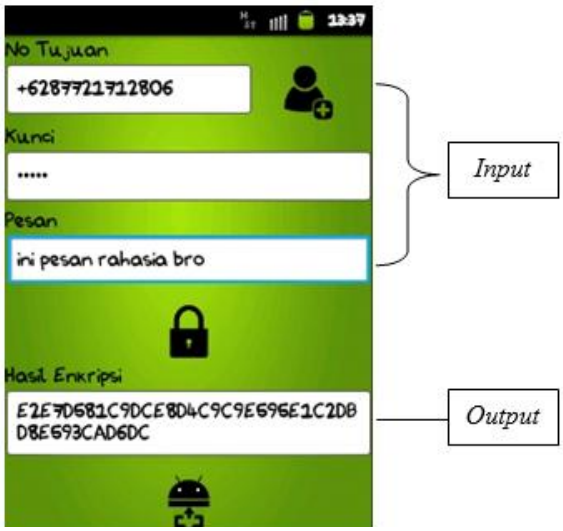
a. Observasi
Teknik pengumpulan data dengan mengadakan penelitian dan melihat secara langsung aplikasi yang terkait dengan pengamanan data untuk melihat sejauh mana teknik yang digunakan
b. Eksperimen
Melakukan beberapa kali uji coba terhadap aplikasi agar sesuai dengan yang diharapkan.
c. Studi Pustaka
Penulis mencari dan mengumpulkan data-data dengan cara mencari sumber referensi dari buku-buku yang terdapat diperpustakaan serta internet yang berhubungan dengan penelitian enkripsi SMS berbasis android.

IV. HASIL DAN PEMBAHASAN

Dari segi keamanan fitur SMS ini bukan merupakan jalur yang aman untuk bertukar informasi karena SMS yang kita kirim tidak langsung sampai ke penerima, akan tetapi harus melalui SMSC yang berfungsi mencatat komunikasi antara pengirim dan penerima SMS, aplikasi ini dimaksudkan agar isi pesan yang bersifat rahasia dapat dikirim dengan aman tanpa harus khawatir isi pesan tersebut diketahui oleh orang lain. Enkripsi merupakan suatu proses mengubah suatu pesan asli (plaintext) menjadi sebuah kode-kode khusus yang tidak bisa terbaca dan tidak dapat dimengerti (ciphertext), untuk dapat membaca atau mengembalikan isi pesan yang telah dienkripsi tersebut diperlukan suatu proses dekripsi. Salah satu metode enkripsi yang umum digunakan yaitu algoritma menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi, atau disebut juga algoritma simetris.

1. Spesifikasi bentuk masukkan

Pada bentuk masukkan ini pengirim SMS melakukan input nomor tujuan, kunci dan isi pesan lalu mengenkripsinya sebelum akhirnya SMS tersebut dikirimkan.



Sumber : Hasil Penelitian (2014)

Gambar 1. Tampilan user interface bentuk masukan Tulis Pesan

a. *User* (pengirim SMS) melakukan input nomor tujuan yang akan dikirim SMS, untuk *input* nomor tujuan itu sendiri bisa dilakukan dengan dua cara, yaitu dengan cara input langsung pada *text box* atau bisa juga dengan mencari nomor pada kontak yang tersedia di

smartphone tersebut dengan cara menekan tombol kontak disamping *text box*.

b. Setelah *user* melakukan *input* nomor tujuan lalu *user* melakukan input kunci, *input* kunci ini diperlukan untuk mengenkripsi isi pesan nantinya. Berdasarkan gambar diatas kunci yang diinput adalah "susah" namun tidak ditampilkan dalam bentuk teks aslinya akan tetapi diganti dengan karakter "•••••", hal ini bertujuan untuk menjaga keamanan kunci.
c. Input pesan, ini adalah isi pesan yang nantinya akan dienkripsi (*plaintext*).
d. Setelah *user* (pengirim) melakukan input kunci dan isi pesan, tekan tombol enkripsi untuk mengenkripsi pesan, maka proses enkripsi akan dilakukan, proses enkripsi dilakukan dengan cara menjumlahkan isi pesan (*plaintext*) dengan kunci yang sebelumnya telah diinput.

Berikut cara penghitungan enkripsi.

Kunci : susah
Plaintext : ini pesan rahasia bro

Langkah pertama adalah membalik *plaintext* lalu ambil nilai ASCII dari setiap karakter kunci dan *Plaintext*

Kunci : susah
Plaintext : orb aisahar nasep ini

Tabel 1: Hasil Enkripsi

Kunci	s	u	s	a	h
ASCII	115	117	115	97	104

Plaintext	o	r	b	(spasi)	a	i	s
ASCII	111	114	98	32	97	105	115
Plaintext	a	h	a	r	(spasi)	n	a
ASCII	97	104	97	114	32	110	97
Plaintext	s	e	p	(spasi)	i	n	i
ASCII	115	101	112	32	105	110	105

Sumber : Hasil Penelitian (2014)

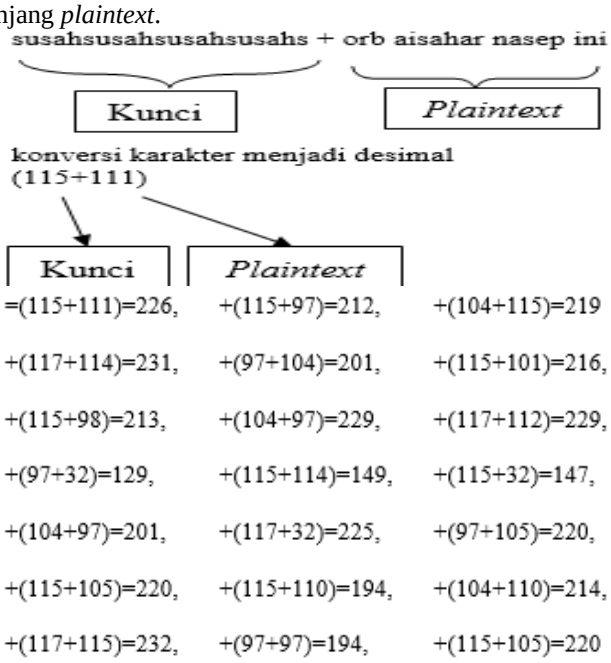
Lalu jumlahkan nilai ASCII kunci dengan nilai ASCII *plaintext* dengan ketentuan berikut:

a) Jika panjang kunci dan panjang *plaintext* sama maka langsung lakukan penjumlahan.
b) Jika panjang kunci lebih dari panjang *plaintext* maka tambahkan *null* pada pesan sampai panjang *plaintext* tersebut sama dengan panjang kunci.
c) Jika panjang kunci kurang dari panjang *plaintext* maka ulangi kunci sampai panjang kunci tersebut sama dengan panjang *plaintext.*

Setelah nilai proses penjumlahan nilai ASCII selesai maka hasilnya akan dirubah kedalam bentuk heksadesimal dan ditampilkan pada *text box* hasil.

Penghitungan untuk mendapatkan hasil enkripsi adalah sebagai berikut: Enkripsi = kunci + *plaintext*

Karena panjang kunci kurang dari panjang *plaintext* maka ulangi kunci sampai panjang kunci tersebut sama dengan panjang *plaintext*.



Konversi desimal menjadi heksadesimal

=226=E2, +231=E7, +213=D5, +129=81, +201=C9, +220=DC, +232=E8, +212=D4, +201=C9, +201=C9, +229=E5, +149=95, +225=E1, +194=C2, +219=DB, +216=D8, +229=E5, +147=93, +202=CA, +214=D6, +220=DC

Maka hasil akhir dari proses enkripsi (*ciphertext*) yang didapat adalah:

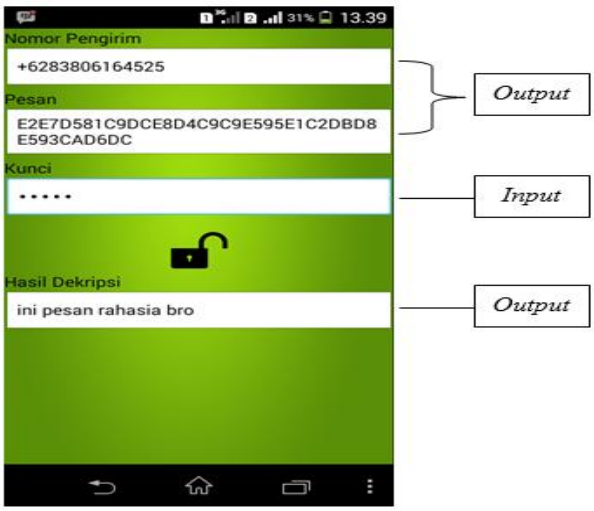
E2E7D581C9DCE8D4C9C9E595E1C2DBD8E593CAD6DC

Nilai inilah yang akan muncul pada *text box* hasil enkripsi sebagai output.

Setelah proses enkripsi selesai dilakukan dan hasil enkripsi telah muncul, maka sekarang *user* (pengirim) dapat langsung mengirimkan pesan hasil enkripsi dengan cara menekan tombol kirim pesan.

2. Spesifikasi bentuk keluaran

Pada bentuk keluaran ini pesan sebelumnya telah dikirim oleh pengirim akan diterima oleh penerima, setelah membuka pesan yang telah diterima maka nomor pengirim dan isi pesan akan tampil, lalu penerima melakukan input kunci yang sama dengan kunci yang diinput oleh pengeirim SMS maka hasil dekripsi akan tampil seperti gambar.2 berikut ini.



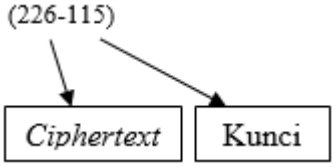
Sumber : Hasil Penelitian (2014)

Gambar 2. Tampilan *user interface* bentuk keluaran Baca Pesan

Adapun penjelasannya sebagai berikut:

1. Setelah *user* (penerima) menerima SMS yang sebelumnya telah dikirim oleh *user* (pengirim), maka akan langsung tampil *output* nomor pengirim dan isi pesan yang masih dalam bentuk heksadesimal (*chipertext*)
2. *User* (penerima) SMS melakukan *input* kunci, untuk dapat melakukan proses dekripsi atau membaca isi pesan asli (*plaintext*) dari pesan yang masih dalam bentuk heksadesimal (*ciphertext*) dibutuhkan kunci yang sama seperti saat *user* (pengirim) melakukan enkripsi.
3. Seteleh *user* (penerima) melakukan input kunci, tekan tombol dekripsi untuk mendekripsi pesan, maka proses dekripsi akan dilakukan, proses dekripsi ini adalah kebalikan dari proses enkripsi. Berikut cara penghitungan dekripsi.

Ciphertext : E2E7D581C9DCE8D4C9C9E595E1C2DBD8E593CAD6DC
Kunci : susah
Langkah pertama adalah memisahkan nilai *ciphertext* menjadi masing-masing dua digit heksadesimal
E2+E7+D5+81+C9+DC+E8+D4+C9+C9+E5+95+E1+C2+DB+D8+E5+93+CA+D6+DC
Konversi setiap dua digit heksadesimal tersebut menjadi bilangan desimal
226+231+213+129+201+220+232+212+201+201+229+149+225+194+219+216+229+147+202+214+220
Dekripsi = *ciphertext* – kunci



Lakukan pengurangan dengan kunci
= (226-115)=111, +(231-117)=114, +(213-115)=98, +(129-97)=32, +(201-104)=97, +(220-115)=105, +(232-117)=115, +(212-115)=97, +(201-97)=104, +(201-104)=97, +(229-115)=114, +(149-117)=32, +(225-115)=110, +(194-97)=97, +(219-104)=115, +(216-115)=101, +(229-117)=112, +(147-115)=32, +(202-97)=105, +(214-104)=110, +(220-115)=105

Konversi bilangan desimal tesebut menjadi karakter
=111=o, +114=r, +98=b, +32=(spasi), +97=a, +105=i, +115=s, +97=a, +104=h, +97=a, +114=r, +32=(spasi), +110=n, +97=a, +115=s, +101=e, +112=p, +32=(spasi), +105=i, +110=n, +105=i

Maka hasil dari konversi diatas adalah:
orb(spasi)aisahar(spasi)nasep(spasi)ini

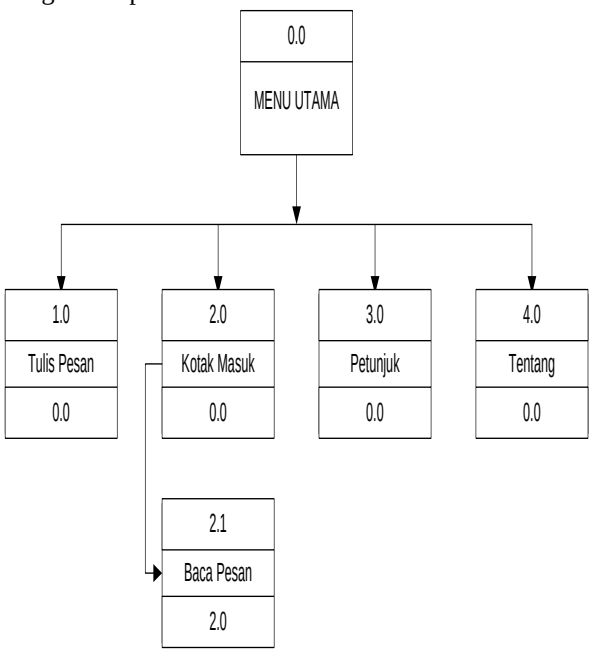
Balik hasil dekripsi tersebut dan hilangkan nilai *null* (jika ada)
ini(spasi)pesan(spasi)rahasia(spasi)bro
maka didapatlah hasil dekripsi berikut:
ini pesan rahasia bro

Sumber : Hasil Penelitian (2014)

Gambar 3. Hasil Dekripsi

Nilai inilah yang akan muncul pada *text box* hasil dekripsi sebagai output.

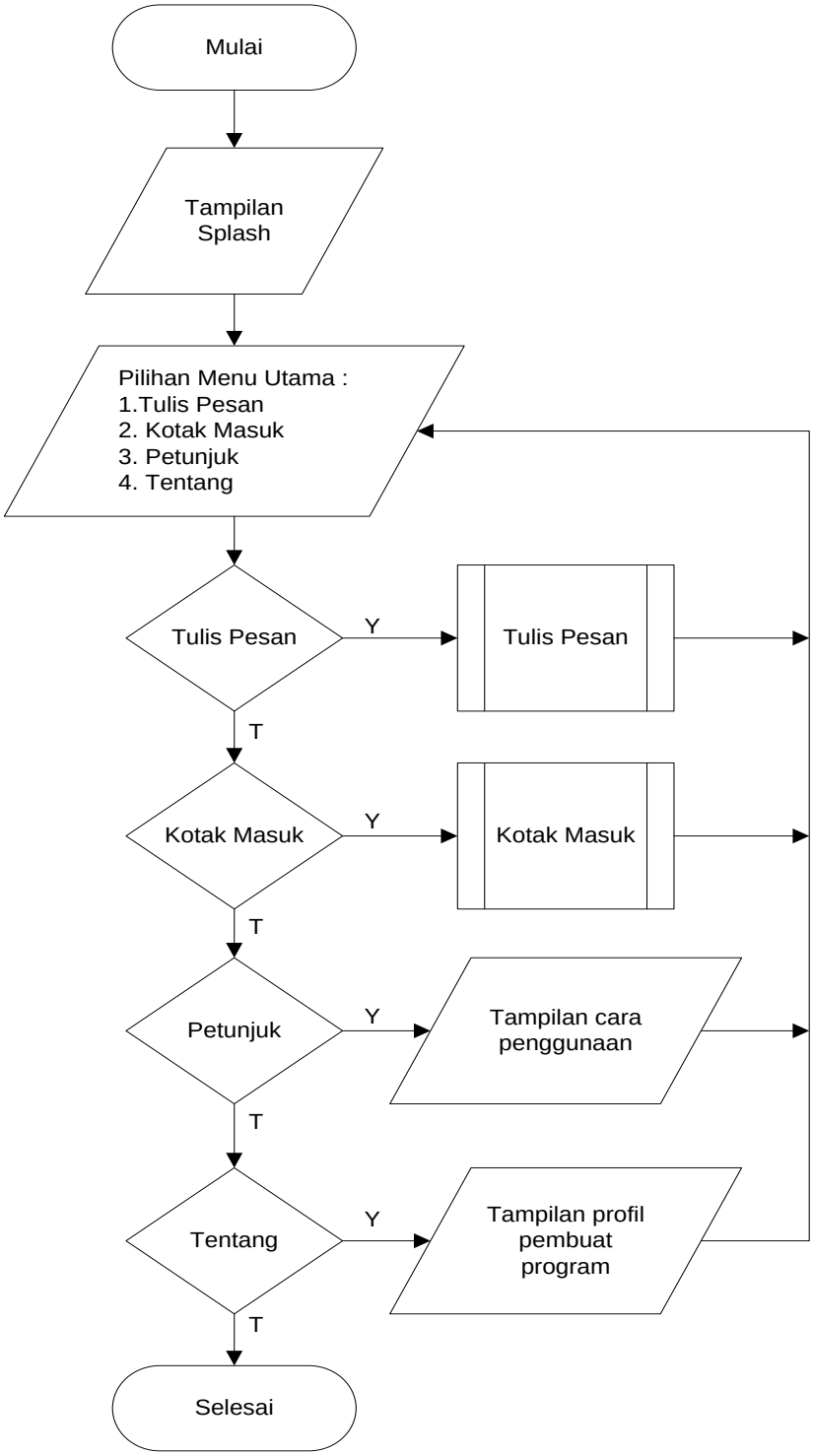
3. Diagram Hipo



Sumber : Hasil Penelitian (2014)

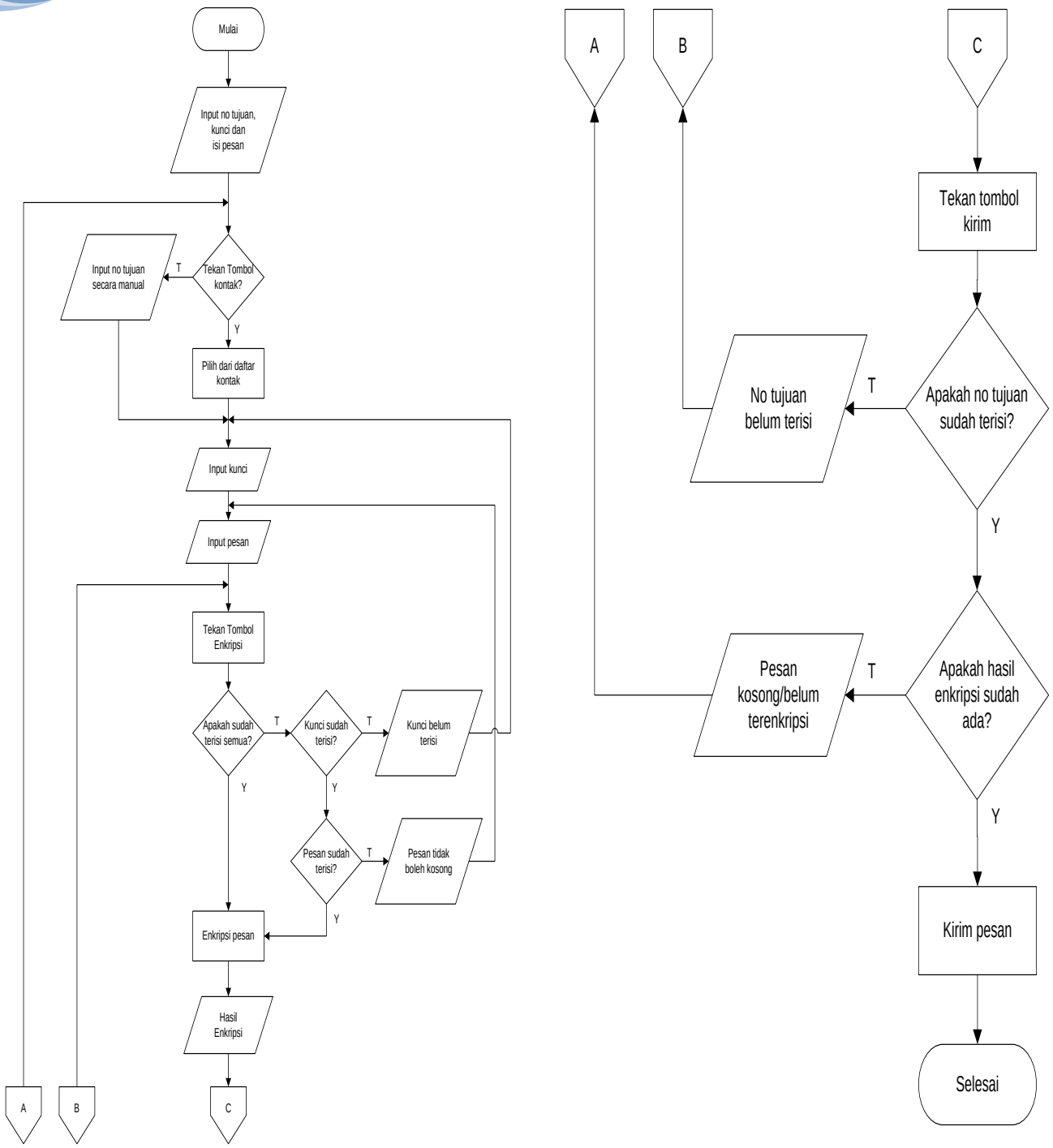
Gambar 4. Diagram HIPO

4. Flowchart



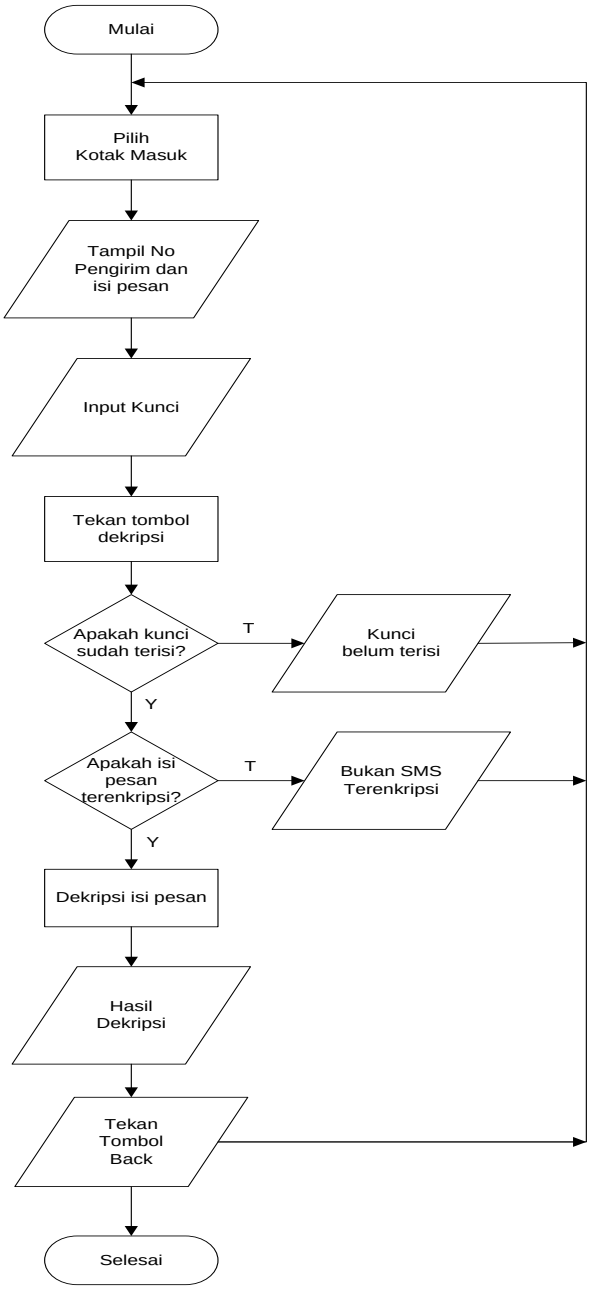
Sumber : Hasil Penelitian (2014)

Gambar 4. Flowchart Menu Utama



Sumber : Hasil Penelitian (2014)

Gambar 5. Flowchart Tulis Pesan



Sumber : Hasil Penelitian (2014)

Gambar 6. Flowchart Kotak Masuk

3. Code Generation

Code Generation berisikan script listing program yang digunakan dalam pembuatan aplikasi enkripsi SMS berbasis Android, adapun script programnya diantaranya sebagai berikut :

A. Listing Class SMS.java

```
public class Sms extends Activity {
        Button
var_tulispesan,var_bacasms,var_about,var_help;

        public void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
          setContentView(R.layout.xmenu);
          var_tulispesan=(Button)findViewById(R.id.btn
TulisPesan);
          var_bacasms=(Button)findViewById(R.id.btnBac
aSMS);
          var_about=(Button)findViewById(R.id.btnAbout
);
          var_help=(Button)findViewById(R.id.btnHelp);
          var_tulispesan.setOnClickListener(
        new Button.OnClickListener(){
                public void onClick(View v) {
                        tulispesan();
                }
            });
```

B. Listing Tulis pesan

```
setContentView(R.layout.xtulis_pesan);
var_kunci=(EditText)  findViewById(R.id.txtKunci);
var_pesan=(EditText)  findViewById(R.id.txtPesan);
var_hasil=(EditText)  findViewById(R.id.txtHasil);
var_NoTujuan=(EditText)findViewById(R.id.txtNoTu
juan);
var_enkripsi=(Button)findViewById(R.id.btnEnkrip
);
var_kirim=(Button)findViewById(R.id.btnKirim);
var_contact=(Button)  findViewById(R.id.contact);

var_hasil.setFocusable(false);
```

Listing Tombol YA

```
on(release){ loadMovieNum("loading.swf",0);}
```

Listing Tombol TIDAK

```
on (release)
{
   getURL("FSCommand:quit", "");
}
```

C. Listing Baca Pesan

```
public class baca_sms extends Activity {
private EditText var_kunci2,var_pesan2,var_hasil2,var_NoPengirim;
      private Button dekripsi;
      private String Skunci,Spesan,dekrip;

      public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
       setContentView(R.layout.xbaca_sms);

         Bundle extras=getIntent().getExtras();
         String no=extras.getString(Inbox.nosms);
         String isi=extras.getString(Inbox.isisms);

         var_kunci2=(EditText) findViewById(R.id.txtKunci2);
         var_pesan2=(EditText) findViewById(R.id.txtPesan2);
         var_hasil2=(EditText) findViewById(R.id.txtHasil2);
         var_NoPengirim=(EditText)findViewById(R.id.txtNoPengirim);
         dekripsi=(Button)findViewById(R.id.btnDekrip);
         var_NoPengirim.setFocusable(false);
         var_pesan2.setFocusable(false);
         var_hasil2.setFocusable(false);
         var_NoPengirim.setText(no);
         var_pesan2.setText(isi);
```

D. Listing Enkripsi

```
package com.enkripsisms;
public class enkripsi{
public String Enkripsi(String pesan, String kunci){
int i, j, panjangPesan, panjangKunci;
      String cipher = new String();
panjangPesan = pesan.length();
      panjangKunci = kunci.length();
      j = 0;
```

```
if (panjangPesan < panjangKunci){
        for (i = 0; i < (panjangKunci - panjangPesan); i++)
pesan = pesan + '\0';
        panjangPesan = pesan.length();
}
for (i = panjangPesan - 1; i >= 0; i--)
cipher = cipher + pesan.charAt(i);
pesan = new String();

for (i = 0; i < panjangPesan; i++){
int nilai = cipher.charAt(i) + kunci.charAt(j);
pesan = pesan + Integer.toHexString(nilai);

    if (j == (panjangKunci - 1))
    j = 0;
    else
    j++;
}
return pesan;
}
```

E. Listing Dekripsi

```
public String Dekripsi(String cipher, String
kunci){
int i, j = 0;
String pesan = new String();

for (i = 0; i < cipher.length(); i+=2){
int nilai = Integer.parseInt(cipher.substring(i,
i+2), 16);

nilai = nilai - kunci.charAt(j);
      pesan = pesan + (char) nilai;

            if (j == (kunci.length() - 1))
                j = 0;
             else
                j++;
}
cipher = new String();
      for (i = pesan.length() - 1; i >= 0; i--){
            if (pesan.charAt(i) != '\0')
            cipher = cipher + pesan.charAt(i);
           }
return cipher;
}
}
```

V. KESIMPULAN

Kesimpulan yang penulis dapatkan selama melakukan penelitian ini, adalah :

1. Aplikasi SMS *Security* ini dapat mengamankan isi pesan yang bersifat privasi agar *user* merasa aman dalam mengirim pesan dari pihak-pihak yang tidak berwenang.
2. Algoritma pengenkripsian aplikasi ini bisa dibilang masih sangat sederhana, sehingga relatif mudah ditebak jika dilakukan *cryptanalist*.
3. Tingkat keamanan pesan pada aplikasi ini tergantung pada kerumitan kunci yang digunakan untuk mengenkripsi isi pesan.

Pada bagian ini penulis memberikan saran-saran berdasarkan permasalahan serta kesimpulan yang penulis dapat, yaitu :

1. Tingkat keamanan enkripsi pesan aplikasi ini yang masih sangat sederhana, sehingga perlu ditingkatkan tingkat keamanannya dengan cara merubah algoritma enkripsinya menjadi lebih kompleks.
2. Untuk pengembangan aplikasi di masa yang akan datang bisa ditambahkan enkripsi file, seperti foto, dokumen, video dsb.
3. Membuat *database* sendiri yang khusus untuk aplikasi ini karena aplikasi SMS *Security* ini masih menggunakan *database* dari aplikasi SMS bawaan *smartphone*.

REFERENSI

[1] Alul. 2012. Tutorial Android Lifecycle Aplikasi Android. Diambil dari: http://www.omayib.com/2012/06/21/tutorial-android-lifecycle-aplikasi.html. (21 Juni 2012).

[2] Ariyus, Dony. Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi. Yogyakarta: Andi Offset. 2008.

[3] Hartono, Jogiyanto. Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis. Yogyakarta: Andi Offset. 2005.

[4] Huda, Arif. LiveCoding! 9 Aplikasi Android Buatan Sendiri. Yogyakarta: Andi Offset. 2013.

[5] Kusumo, Ario. Visual Basic .Net versi 2002 dan 2003. Jakarta: PT. Elex Media Komputindo. 2004.

[6] Ladjamudin. Analisis dan Desain Sistem Informasi. Yogyakarta: Graha Ilmu. 2005.

[7] Novia. Pengertian SMS *(Short Message Service)*. Diambil dari: http://www.rapendik.com/program/pengayaan-pembelajaran/petik/561-pengertian-sms-short-message-service.html. (13 Februari 2013).

[8] Prastyo, Didik. 150 Rahasia Pemrograman Java. Jakarta: PT. Elex Media Komputindo. 2007.

[9] Safaat, Nazruddin. Aplikasi Berbasis Android. Bandung: Informatika Bandung. 2013.

[10] Seralo. Android Version Comparison. Diambil dari: http://socialcompare.com/ en/comparison/android-versions-comparison. (27 April 2014)

[11] Stiawan, Deris. Sistem Keamanan Komputer. Jakarta: PT. Elex Media Komputindo. 2005.

Aries Gumilar Pratama. Tahun 2014 lulus dari Program Diploma Tiga (DIII) Program Studi Teknik Komputer AMIK BSI Jakarta.

Anton, M.Kom. Tahun 2002 lulus dari Program Strata Satu (S1) Program Studi Teknik Informatika STMIK MH. Thamrin Jakarta. Tahun 2009 lulus dari Program Strata Dua (S2) Program Studi Magister Komputer Universitas Budi Luhur Jakarta. Staf Akademik AMIK BSI Jakarta

Firmansyah, S.Kom. Tahun 2014 lulus dari Program Strata satu (S1) Program Studi Teknik Informatika STMIK Nusa Mandiri Jakarta.