Implementasi Algoritma Blowfish dalam Layanan Pesan Singkat pada *Platform* Android

Sonny Theo Tumbur (13510027)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13510027@std.stei.itb.ac.id

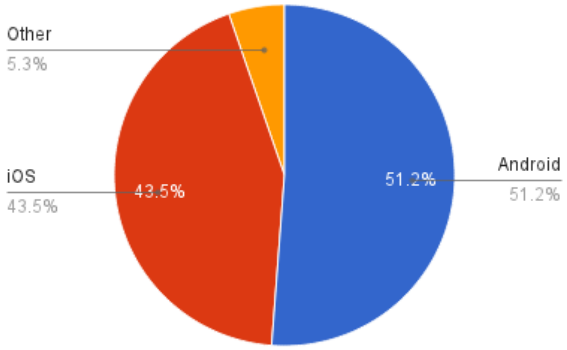
***Abstract*—Pengguna layanan pesan singkat terus bertumbuh, sedangkan pengembangan aplikasi lebih cenderung mengarah ke aplikasi *instant messaging*. Karena itu, dibutuhkan suatu pengembangan lanjut terkait layanan pesan singkat khususnya dalam bidang keamanan pesan. Makalah ini membahas implementasi algoritma Blowfish dalam layanan pesan singkat (*Short Message Service*) pada *platform* Android. *Platform* Android dipilih dengan alasan *platform* yang relatif lebih *reliable* dan banyak digunakan.**

***Index Terms*—Blowfish, Android, *Encryption*, BroadcastReceiver.**

I. PENDAHULUAN

Layanan pesan singkat (*short message service*) merupakan salah satu mekanisme pengiriman pesan yang banyak digunakan oleh berbagai kalangan seperti kalangan keluarga, organisasi bisnis, dsb. Situs www.factbrowser.com (29 Maret 2014, 02:42:00) menyatakan bahwa 92% penduduk Amerika Serikat dan 96% penduduk Inggris yang memiliki *Smartphone* kerap kali menggunakan fitur layanan pesan singkat ini. Semakin maraknya penggunaan *instant messaging* secara tidak langsung dapat menyebabkan kurang berkembangnya pengembangan di bidang layanan pesan singkat khususnya fitur keamanan.

Mekanisme enkripsi-dekripsi masih merupakan pilihan utama dalam melakukan pengamanan pesan. Teknik lainnya yang dapat digunakan mencakup steganografi, dll.



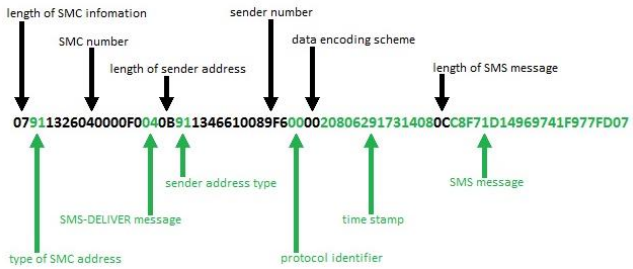
Gambar 1. Data pengguna Smarphone. (Kantar US Smartphone Sales).

Berdasarkan data pengguna *Smartphone* pada berbagai perangkat dan *platform* (Gambar 1), dapat dilihat bahwa *platform* Android merupakan *platform* dengan persentase pengguna terbanyak. Selisih 7,7% dari pengguna merupakan angkat yang cukup besar untuk menjadi perbedaan penggunaan kedua *platform* raksasa ini.

II. TELAAH PUSTAKA

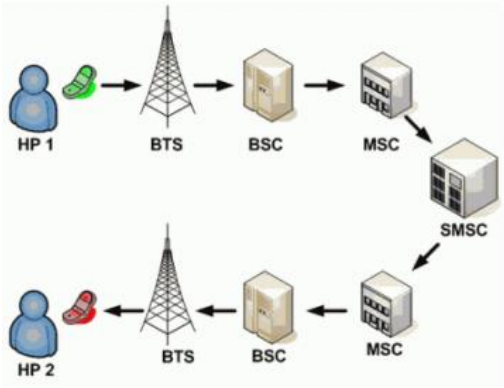
Layanan pesan singkat merupakan sebuah layanan pengiriman pesan yang didasarkan pada jaringan suara (*voice network*) yang berdiri di atas teknologi jaringan seperti GSM, CDMA, maupun TDMA. Layanan ini memungkinkan pengirimkan pesan dalam bentuk teks dengan panjang maksimal 160 karakter. Untuk beberapa alfabet selain alfabet Latin, seperti Chinese ataupun Arabic, ukuran pesan teks maksimal adalah 70 karakter. Ukuran 160 karakter ini pertama kali ditentukan oleh Friedhelm Hillbrand yang telah melakukan pengamatan mengenai jumlah karakter tipikal yang digunakan dalam kalimat pada umumnya.

Struktur pesan yang digunakan dalam layanan pesan singkat dijelaskan dalam *Protocol Description Unit* (PDU). Deskripsi ini mencakup panjang pesan, penanda waktu, pengirim pesan, target penerima pesan, berserta isi dari pesan itu sendiri. *String* PDU terdiri dari *hexadecimal-octet* dan semi *decimal-octet*. *Hexadecimal* merupakan nilai numerik dengan basis 16, sedangkan nilai *decimal* menggunakan basis 10. Sedangkan untuk merepresentasikan isi dari pesan tersebut, digunakan 7-bit GSM alfabet. Berikut ini adalah ilustrasi dari representasi data dalam layanan pesan singkat.



Gambar 2. Representasi data dalam layanan pesan singkat.

Umumnya layanan pesan singkat dikenal berada pada telepon seluler, namum sesungguhnya layanan ini juga dapat diakses melalui perangkat lainnya seperti *personal computer* (PC), *tablet*, maupun *laptop* selama perangkat tersebut dapat menggunakan kartu SIM. Berikut ini adalah ilustrasi mekanisme pengiriman pesan melalui layanan pesan singkat. (Rayarikar, Upadhyay, Pimpale, 2012).



Gambar 3. Transmisi layanan pesan singkat

BTS (*Base Transceiver Station*) merupakan komponen yang menghubungkan *User Equipment* (UE) dengan jaringan. UE mencakup telepon seluler, *handset*, WLL *phone*, perangkat WiFi, perangkat WiMAX, dll. MSC (*Mobile Switching Center*) merupakan simpul yang penting untuk layanan GSM maupun CDMA. Komponen ini berfungsi untuk melakukan *routing* (pengaturan rute) untuk tiap layanan seperti layanan suara, fax, maupun layanan pesan singkat ini. Komponen ini jugalah yang melakukan penanganan terhadap akun untuk setiap kartu SIM baik terkait jumlah pulsa yang tersedia untuk mengakses layanan maupun masa aktif / tenggan akun yang bersangkutan. Komponen berikutnya adalah SMSC (*Short Message Service Center*). Komponen ini bertugas untuk melakukan pencarian terkait alamat target dari pesan dan juga melakukan pengiriman pesan tersebut ke target yang dituju. Di samping itu, SMSC juga berperan sebagai tempat penyimpanan pesan sementara pada kondisi dimana target berada dalam kondisi tidak aktif. Tentu saja SMSC hanya dapat menyimpan pesan tersebut dalam waktu tertentu dikarenakan keterbatasan sumber daya. Dalam proses pengiriman ini, SMSC juga banyak melakukan komunikasi dengan pengirim pesan singkat contohnya terkait notifikasi berhasilnya pengiriman pesan. (Rayarikar, Upadhyay, Pimpale, 2012).

Algoritma Blowfish merupakan salah satu *block cipher* yang memiliki kunci simetrik. Algoritma ini menggunakan blok dengan ukuran 64-bit dan kunci minimal sepanjang 32 bit dan maksimal 448 bit. Operasi yang banyak digunakan antara lain operasi *exclusive*-or / xor yang dilambangkan dengan notasi $\oplus$. Suatu *subkey* (upakunci) sebanyak 18 juga dibutuhkan dalam menyelesaikan algoritma ini. Gambar 4 menjelaskan skema algoritma Blowfish untuk mengenkripsi blok pesan *x*. (Meyers, Desoki, 2008).

Algorithm 1 Blowfish Encryption

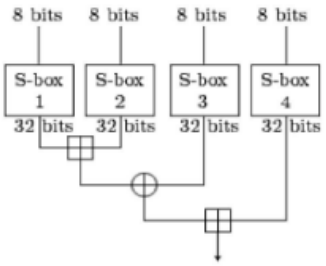
Divide x into two 32-bit halves: x_L and x_R
for $i = 1, 2, ..., 16$ **do**
  $x_L = x_L \oplus P_i$
  $x_R = F(x_L) \oplus x_R$
  Swap x_L and x_R
end for
Swap x_L and x_R (undo the last swap)
$x_R = x_R \oplus P_{17}$
$x_L = x_L \oplus P_{18}$
Recombine x_L and x_R

Gambar 4. Skema Enkripsi Algoritma Blowfish. (Meyers, Desoki, 2008).

Fungsi F dalam skema enkripsi blowfish tersebut menggunakan kotak substitusi / S-box, yang mana terdapat empat kotak dengan masing-masing kotak berisi 256 32-bit entri. Misalnya blok x_l dibagi menjadi blok 8-bit a, b, c, d, maka dapat dirumuskan:

$$F(X_l) = ((S_{1,a} + S_{2,b} \bmod 2^{32}) \oplus S_{3,c}) + S_{4,d} \bmod 2^{32}$$

Ilustrasi mengenai fungsi tersebut berada pada gambar 5.



Gambar 5. Fungsi F pada Blowfish.

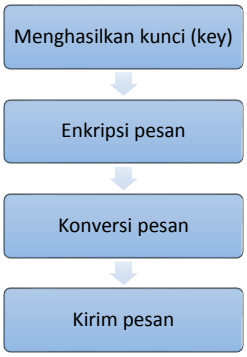
III. PERANCANGAN

Perancangan mencakup penentuan daftar kebutuhan fungsional, proses aliran data, perancangan antarmuka.

Kebutuhan fungsional yang akan diimplementasikan dalam aplikasi mencakup tiga hal, antara lain:

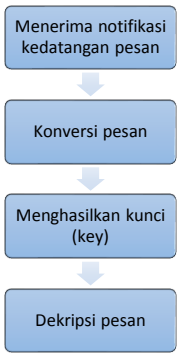
1. pengguna dapat melakukan enkripsi pesan teks sebelum dikirimkan
2. pengguna dapat mengirimkan pesan terenkripsi menuju target pengguna
3. pengguna dapat melakukan dekripsi pesan.

Proses dalam aplikasi dapat dibagi ke dalam dua tahapan yaitu tahap pengiriman pesan dan tahap penerimaan pesan. Berikut adalah ilustrasi mengenai tahap pengiriman pesan:



Pertama, pengguna memasukan kunci yang berukuran 32-448 bit sesuai dengan spesifikasi pada algoritma Blowfish. Kunci inilah yang akan digunakan untuk melakukan enkripsi dan dekripsi pesan. Selanjutnya pesan akan dienkripsi dengan melakukan operasi pada struktur jaringan Feistel pada Blowfish dengan menggunakan kunci dan upakunci yang dihasilkan. Setelah itu, pesan akan dikonversi ke dalam representasi *hexadecimal* sebelum dikirimkan. Selanjutnya, pesan dikirimkan dalam bentuk *string hexadecimal* menuju alamat tujuan.

Tahap-tahap yang ada pada proses penerimaan pesan tidak jauh berbeda dengan proses pengiriman pesan. Berikut adalah ilustrasinya:



Pertama, aplikasi akan mendapatkan notifikasi dari sistem operasi Android mengenai kedatangan pesan dalam bentuk layanan pesan singkat (SMS). Selanjutnya, aplikasi akan melakukan konversi pesan dari representasi *string hexadecimal* ke dalam representasi *array of byte*. Setelah konversi dilakukan, aplikasi akan meminta pengguna untuk memasukan kunci (*key*) yang akan digunakan untuk melakukan dekripsi progam. Berikutnya, aplikasi akan melakukan dekripsi pesan.

Secara umum, kelas-kelas yang akan diimplementasikan mencakup:

1. *EncryptionActivity*
2. *DecryptionActivity*
3. *SmsBroadcastReceiver*

EncryptionActivity memasuki status aktif ketika pengguna ingin melakukan enkripsi dan pengiriman pesan. *DecryptionActivity* akan memasuki status aktif saat *SmsBroadcastReceiver* menerima broadcast terkait adanya pesan yang tiba. Tahap pengiriman dan penerimaan pesan akan memiliki rancangan antarmuka yang berbeda: EncryptionView.xml dan DecryptionView.xml.

IV. IMPLEMENTASI

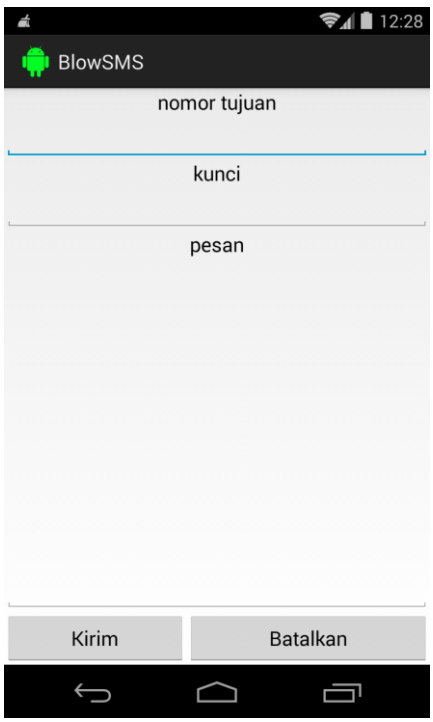
Dalam menghasilkan kunci, aplikasi menggunakan dua parameter masukan (*input*) yaitu kunci masukan pengguna dalam representasi byte dan *string* yang berisi algoritma yang digunakan. Berikut adalah cuplikan kode terkait pembentukan kunci.

```
Key key = new SecretKeySpec(secretKeyString.getBytes(), "Blowfish");
```

Dalam sistem operasi Android, aplikasi harus mengatur mekanisme tertentu untuk bisa menerima wewenang (privilege) untuk diberikan notifikasi pada saat SMS *datang* dan untuk melakukan pengiriman SMS. Pengaturan tersebut dilakukan dengan menambahkan dua izin pada berkas (*file*) AndroidManifest.xml. Kedua izin tersebut mencakup SEND_SMS dan RECEIVE_SMS. Berikut ini adalah cuplikan kode terkait permission.

```
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
```

Implementasi antarmuka yang dilakukan mencakup halaman pengiriman pesan dan halaman penerimaan pesan. Gambar 5 dan Gambar 6 menunjukan antarmuka untuk proses pengiriman pesan dan penerimaan pesan oleh pengguna.

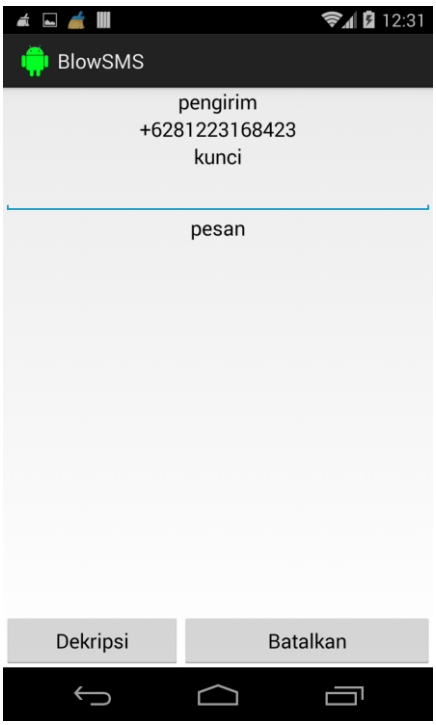


Gambar 6. Antarmuka Pengiriman Pesan.

Pada mode penerimaan pesan, aplikasi telah

menyediakan SmsBroadcastReceiver yang berfungsi untuk menerima notifikasi mengenai kedatangan pesan. BroadcastReceiver ini terlebih dahulu didaftarkan ke dalam sistem agar ketika pesan singkat datang, sistem operasi dapat langsung melakukan broadcast ke seluruh receiver yang telah didaftarkan sebelumnya. Mekanisme pendaftaran receiver dilakukan dengan menambahkan baris tertentu pada berkas AndroidManifest.xml seperti berikut ini.

```
<receiver android:name="android.encdecsms.SmsBroadCastReceiver">
    <intent-filter android:priority="1000">
        <action android:name="android.provider.Telephony.SMS_RECEIVED'
    </intent-filter>
</receiver>
```



Gambar 7. Antarmuka Penerima Pesan.

V. PENGUJIAN

Pengujian aplikasi dilakukan dengan mengirimkan pesan secara langsung. Berikut ini adalah contoh daftar masukan yang digunakan sebagai kasus uji:

1. nomor tujuan: 081223168423
2. kunci: *blowfish*
3. pesan: *saya belajar kriptografi*

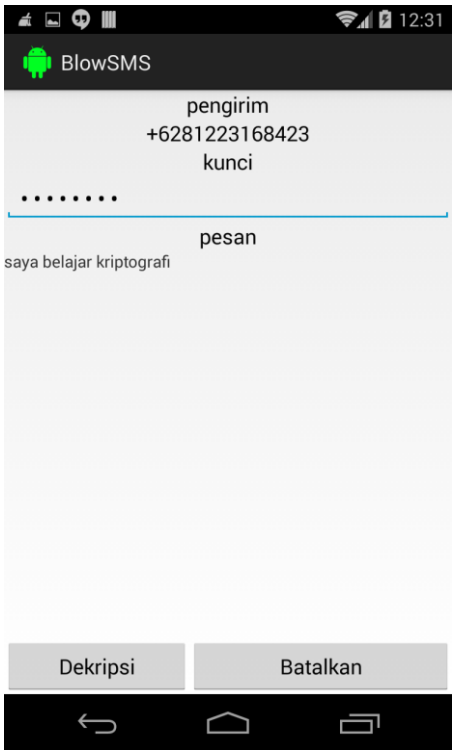
Pada contoh uji berikut, pengiriman pesan (SMS) ditujukan ke perangkat telepon seluler yang sama. Perangkat yang digunakan untuk pengujian ini memiliki spesifikasi sebagai berikut:

1. Perangkat Nexus 4 LG.
2. Sistem operasi Android 4.4.2 Kitkat.
3. Kernel Version 3.4.0-perf-g2cae413.

Berikut adalah cuplikan antarmuka dari kasus uji di atas:

Gambar 8. Kasus Uji Pengiriman Pesan.

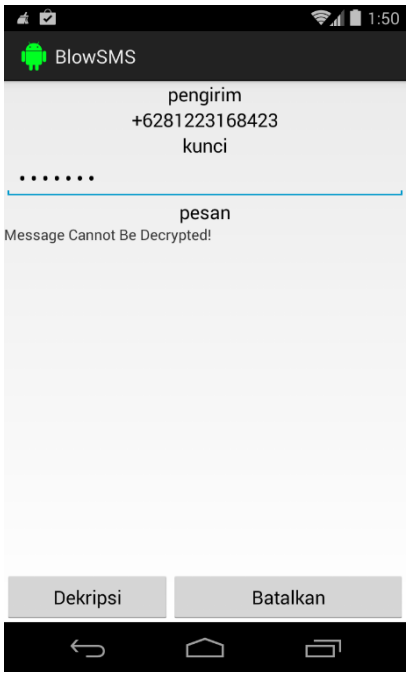
Pada kasus uji ini, aplikasi akan mulai melakukan pemrosesan pesan yang dimulai dengan pengolahan kunci setelah pengguna menekan kirim.



Gambar 9. Kasus uji penerimaan pesan.

Setelah pesan diterima, aplikasi akan menunggu terlebih dahulu pengguna memasukan kunci yang sesuai. Ketika pengguna telah memasukan kunci yang sesuai lalu menekan tombol dekripsi, aplikasi akan

menampilkan pesan dalam bentuk plainteks. Jika kunci yang dimasukan salah oleh pengguna, aplikasi akan menampilkan pesan bahwa pesan tidak dapat didekripsi. Berikut adalah ilustrasinya.



Gambar 10. Kasus uji kesalahan kunci.

VI. KESIMPULAN

Berikut adalah kesimpulan yang dapat dihasilkan dari percobaan dalam makalah ini.

1. Algoitma Kriptografi Blowfish dapat diimplementasikan di atas *platform* Android.
2. Algoritma Kriptografi Blowfish dapat digunakan untuk melakukan pengamanan pada layanan pesan singkat (SMS) pada sistem operasi Android.

REFERENSI

[1] Android Authority. 17 Maret 2014. *What is SMS and how does it work.* Retrieved from http://www.androidauthority.com/what-is-sms-280988.

[2] Meyers, Desoki. 2008. *An Implementation of Blowfish Cryptosystem*. University of Louisville.

[3] Rayarikar, Upadhyay, Pimpale. 2012. *SMS Encryption using AES Algorithm*. International Journal of Computer Applications.

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, 29 April 2010

ttd

Sonny Theo Tumbur
13510027