Pembangunan Aplikasi *Client* SMS dengan Enkripsi Menggunakan Algoritma Twofish pada Telepon Selular Android

Yudhistira[#1], Rinaldi Munir [*2]

Teknik Informatika, Institut Teknologi Bandung
Jalan Ganesha No.10 Bandung Indonesia 40135
[1]yudhistira16@gmail.com
[2]rinaldi@informatika.org

Abstrak— **SMS merupakan sarana komunikasi yang masih banyak digunakan oleh masyarakat di dunia. Sampai saat ini, SMS tetap menjadi sumber pendapatan utama penyedia layanan selular. Banyak pesan rahasia yang masih dikirimkan dengan menggunakan SMS. Namun, di dalam ponsel Android belum ada sistem pengamanan SMS yang memungkinkan pengguna untuk mengirimkan pesan rahasia dengan aman.**
Salah satu cara mengamankan SMS adalah dengan mengenkripsi SMS sebelum SMS tersebut dikirim. Algoritma enkripsi yang digunakan adalah algoritma Twofish. Algoritma ini dipilih karena kekuatannya dan kemampuannya untuk diimplementasikan dalam perangkat dengan komputasi minimal. Algoritma ini diimplementasikan dalam sebuah *client* SMS baru pada ponsel bersistem operasi Android.
Beberapa masalah yang dihadapi dalam pembuatan aplikasi ini adalah bagaimana cara mengirim dan menerima SMS, dan bagaimana cara untuk mengimplementasikan algoritma Twofish pada *client* SMS tersebut. Implementasi aplikasi ini dilakukan dengan menggunakan bahasa pemrograman Java.
Hasil pengujian membuktikan bahwa *client* baru yang dipasangi enkripsi dengan algoritma Twofish ini bisa mengamankan pesan tanpa manghambat kinerja ponsel. *Client* SMS ini bisa mengirimkan dan menerima pesan yang terenkripsi dengan algoritma Twofish. Apabila pengguna ingin membaca pesan, pengguna dapat mendekripsinya sendiri. Apabila pengguna keluar dari program, pesan akan langsung terenkripsi kembali.

Kata kunci— **Kata kunci : SMS, enkripsi, Android, Twofish, telepon selular.**

I. PENDAHULUAN

Seiring dengan perkembangan zaman, telepon seluler (ponsel) kini sudah berkembang menjadi sebuah alat yang bisa melakukan segala hal. Ponsel kini bisa dipakai untuk menjelajah internet, mengecek surat elektronik (*e-mail*), bermain game, mencatat agenda harian, sampai membaca buku. Bahkan, kini tersedia aplikasi pesan instan (*instant messaging*) melalui internet, seperti WhatsApp, BBM, Line, dan lain sebagainya. Singkatnya, ponsel kini telah berevolusi menjadi sebuah telepon pintar (*smartphone*).

Banyaknya fitur yang ada pada smartphone terutama *instant messaging* tidak menjadikan SMS menjadi fitur yang terlupakan. Menurut Portio Research, diperkirakan akan ada 4 triliun SMS yang dikirim dari Asia Pasifik pada tahun 2014 [13]. Dari sudut pandang penyedia layanan, SMS tetap akan menjadi penyumbang aset terbesar. Bahkan, prospek keuntungan SMS pada tahun 2014 masih lebih besar bila dibandingkan dengan prospek keuntungan MMS, *mobile e-mail*, *VoIP*, dan internet dijumlahkan sekaligus.

Ada banyak sistem operasi *mobile* yang disediakan oleh para pengembang ponsel. Beberapa di antaranya adalah Android, BlackBerryOS, iOS, dan Symbian. Dari keempat sistem operasi yang telah disebutkan, Android merupakan sistem operasi yang paling populer [4]. Dari 227 negara yang terrekam datanya oleh StatCounter, ada 135 negara yang pasarnya dikuasai oleh Android [4].

Meskipun Andoid merupakan sistem operasi open source yang ditunjang dengan keamanan yang lebih baik dibandingkan dengan sistem operasi berbayar, sistem operasi ini belum memiliki sistem enkripsi SMS di dalam aplikasi SMSnya. Ada dua macam algoritma yang biasa dipakai untuk melakukan enkripsi. Algoritma tersebut adalah algoritma kunci simetri dan algoritma kunci publik. Dari kedua algoritma tersebut, algoritma yang lebih cocok untuk diimplementasikan dalam perangkat dengan komputasi terbatas seperti ponsel Android, adalah algoritma kunci simetri. Algoritma kunci simetri memiliki banyak macam. Salah satu algoritma kunci simetri terkuat yang ada adalah algoritma Twofish. Menurut Scheiner (1997), algoritma Twofish merupakan algoritma yang sangat cepat dan sangat cocok untuk diimplementasikan pada perangkat keras yang memiliki komputasi minimal [8].

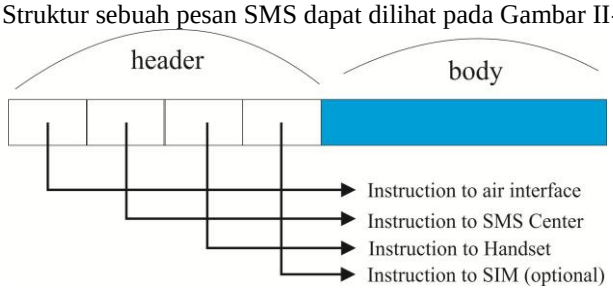
Minimnya keamanan pesan di ponsel Android, rentannya algoritma-algoritma yang ada, dan tersedianya algoritma yang cocok untuk perangkat dengan komputasi minimum, membuat celah untuk pengembangan client SMS yang lebih aman dengan menggunakan algoritma enkripsi yang kuat. Oleh karena itu, dipilihlah Android sebagai lingkungan pengembangan untuk pembangunan client SMS dengan menggunakan salah satu algoritma kunci simetri, yaitu algoritma Twofish.

II. *SHORT MESSAGE SERVICE*

Short Message Service atau SMS merupakan metode berkomunikasi melalui jaringan selular dalam bentuk teks. SMS bekerja pada sistem tanpa kabel. Ada beberapa jaringan

yang bisa dipakai untuk mengirim SMS. Beberapa jaringan yang terkenal adalah GSM, TDMA, CDMA, GPRS, EDGE, WCDMA, dan UMTS [1]. Dari jaringan-jaringan tersebut, yang paling populer di dunia adalah GSM (Global System for Mobile Communication) [5].

Struktur sebuah pesan SMS dapat dilihat pada Gambar II-1.



Gambar 1 Struktur sebuah pesan SMS [1]

Dari gambar 1, terlihat bahwa pesan SMS pada dasarnya terdiri atas *message body* dan headernya. *Header* SMS terdiri atas instruksi-instruksi yang ditujukan kepada SMSC, SIM, maupun kepada ponsel itu sendiri. Sedangkan *message body* adalah konten utama dari SMS berupa pesan yang ditulis oleh pengirim. Sebuah pesan SMS berukuran maksimal 160 karakter [5]. Di mana setiap karakter memiliki panjang 7 bit.

Menurut Clements (2013), SMS memiliki empat buah komponen utama [1]. Komponen-komponen tersebut adalah:

1. *Cell Tower*

Cell Tower bertanggung jawab untuk mentransmisikan suara dan data (SMS) antara ponsel dan MSC (*Mobile Switching Center*). Semua transmisi dikendalikan oleh *cell tower*.

2. *Mobile Switching Center* (MSC)

MSC adalah sebuah saklar yang dikontrol oleh komputer yang berfungsi untuk mengatur operasi jaringan yang berjalan secara otomatis. MSC secara otomatis mengontrol panggilan telepon dan dan merutekannya ke ponsel yang tepat pada sebuah *service area*. MSC dihubungkan ke *base station* oleh channel gelombang mikro, dan dihubungkan ke PSTN (Public Telephone Network) melalui sambungan telepon.

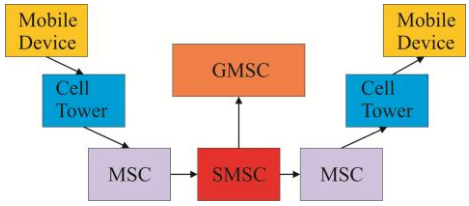
3. SMSC (SMS *Center*)

Ketika mengirim SMS, SMS akan disimpan sementara di SMSC. SMSC bertindak sebagai sebuah tempat penyimpanan, dan pem-forward SMS. Sama halnya dengan MSC, SMSC menjamin SMS akan sampai pada pengguna. SMS disimpan di jaringan sampai ponsel penerima tersedia di jaringan. Hal ini membuat pengguna dapat menerima atau mentransmisikan SMS kapanpun.

4. *Gateway Mobile Switching Center* (GMSC)

SMSC berkomunikasi dengan jaringan TCP/IP melalui GMSC. GMSC merupakan sebuah MSC yang dapat menerima SMS dari SMSC.

Hubungan antara keempat komponen SMS dapat dilihat pada Gambar II-2:



Gambar 2 Hubungan antara keempat komponen SMS [1]

Dari gambar di atas, jelas terlihat sebelum mencapai penerima, SMS terlebih dahulu disimpan di SMSC. Administrator SMSC tentu saja bisa membaca pesan yang disimpan di SMSC. Oleh karena itu, diperlukan sebuah metode untuk mengenkripsi SMS demi mengamankan pesan dari resiko disadap saat berada di SMSC.

III. Algoritma Kriptografi Twofish

Dalam dunia kriptografi modern, ada dua macam algoritma kriptografi yang bisa digunakan untuk mengenkripsi SMS. Algoritma pertama adalah algoritma kunci simetri. Algoritma kunci simetri adalah algoritma yang memiliki kunci enkripsi dan kunci dekripsi yang sama. Algoritma kedua adalah algoritma kunci publik. Algoritma ini membutuhkan dua buah kunci yang berbeda. Satu kunci berguna untuk melakukan enkripsi, sedangkan kunci lainnya berguna untuk melakukan dekripsi.

Algoritma kunci publik memiliki kompleksitas yang sangat besar karena algoritma ini melibatkan komputasi dengan angka yang besar. Oleh karena itu, algoritma ini kurang cocok untuk dijadikan enkripsi SMS. Sebaliknya, algoritma kunci simetri memiliki komputasi yang relatif lebih sederhana dibandingkan dengan algoritma kunci publik. Oleh karena itu, untuk enkripsi SMS, algoritma kunci simetri lebih sesuai untuk dipilih.

Salah satu algoritma kunci simetri yang dikenal kuat adalah algoritma Twofish. Algoritma ini ditemukan oleh lima orang ilmuwan, yaitu Bruce Scheiner, John Kelsey, Doug Whiting, David Wagner, dan Chris Hall. Algoritma ini merupakan salah satu dari lima finalis dalam kompetisi untuk menentukan *Advanced Encryption Standard* (AES) bersama dengan algoritma MARS, RC6, Serpent, dan Rijndael [10]. Walaupun pada akhirnya Twofish tidak dinobatkan menjadi AES, namun algoritma ini dinilai cukup aman oleh para juri, sehingga bisa berada di posisi ke-3 dalam *voting* pemilihan AES [12]. Adapun peringkat pertama diduduki oleh algoritma Rijndael, dan peringkat kedua diduduki oleh algoritma Serpent [12].

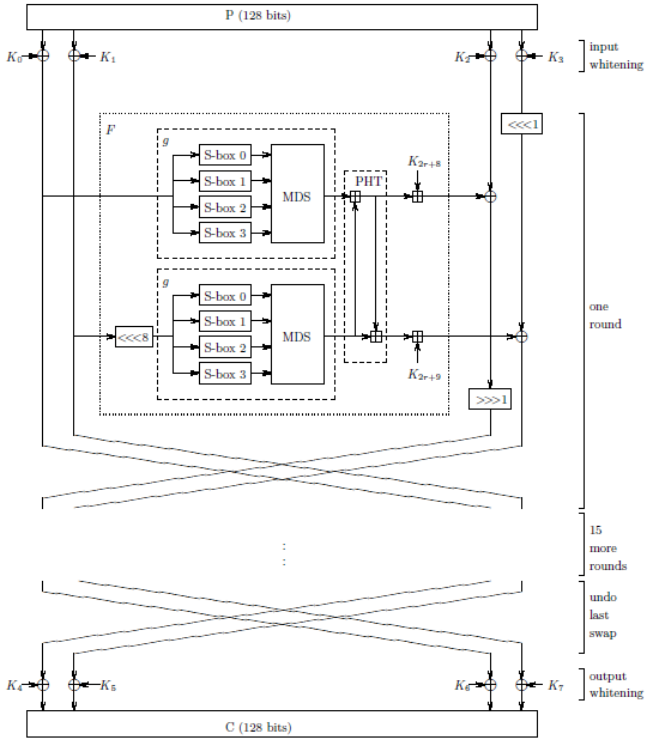
Menurut Scheiner (1997), algoritma Twofish merupakan algoritma yang sangat cepat untuk dijalankan pada perangkat yang memiliki komputasi minimum, contohnya pada *smartcard* [8]. Oleh karena itu, algoritma ini sangat cocok untuk dipakai dalam ponsel, khususnya ponsel Android.

Twofish merupakan sebuah algoritma block cipher 128-bit yang dapat menerima kunci sampai sepanjang 256 bit.

Gambar 3 menunjukkan skema dari block cipher Twofish.

Plainteks yang akan dienkripsi di-split menjadi 4 buah sub-plainteks berukuran 32-bit. Pada langkah whitening awal, keempat sub-plainteks ini di-XOR dengan 4 kata kunci. Hal ini diulangi sebanyak 16 kali, sesuai dengan banyaknya round pada algoritma ini. Pada masing-masing round, dua sub-kunci di sebelah kiri digunakan sebagai input untuk fungsi g (salah satunya dirotasi 8 bit terlebih dahulu). Fungsi g terdiri atas S-box yang bergantung pada kunci dengan lebar 4-byte, dan satu langkah yang berdasar pada matriks MDS. Hasil dari dua buah fungsi g dikombinasikan dengan menggunakan *Pseudo-*

Hadamard Transform (PHT), dan dua kata kunci ditambahkan ke sini. Dua hasil di atas kemudian di-XOR ke sub-kunci yang ada di sebelah kanan (salah satu dirotasi ke kiri 1 bit terlebih dahulu). Bagian kiri dan kanan kemudian ditukar untuk round berikutnya. Setelah berlangsung selama 16 round, pertukaran bagian kiri dan kanan yang terakhir dibalik, dan 4 sub-plainteks di XOR dengan 4 kunci lagi untuk menghasilkan cipherteks [9].



Gambar 3 : Skema Block Cipher Twofish [9]

Berikut adalah komponen-komponen yang membentuk algoritma Twofish.

1. Jaringan Feistel

Jaringan Feistel adalah sebuah metode untuk mentransformasikan sebuah fungsi (biasanya disebut fungsi F) ke dalam sebuah permutasi [9]. Skema ini ditemukan oleh Horst Feistel. Skema ini merupakan basis dari kebanyakan algoritma block cipher.

2. S-Box

S-box adalah sebuah tabel substitusi yang digunakan pada kebanyakan block cipher [9]. S-box memiliki ukuran masukan dan keluaran yang bervariasi. Ukuran ini dapat diatur secara random ataupun menggunakan algoritma tertentu. S-Box digunakan pada kebanyakan algoritma enkripsi. Algoritma Twofish menggunakan empat buah S-box 8 x 8. S-box ini dibentuk dengan menggunakan dua buah permutasi 8 x 8 dan bagian dari kunci.

3. Matriks MDS

Maximum Distance Separable (MDS) adalah pemetaan linear dari elemen a ke elemen b, kemudian menghasilkan vektor gabungan dari elemen a+b¬, dengan properti nilai minimum dari elemen bukan nol pada tiap vektor bukan nol paling sedikit b+1 [11]. Matriks MDS berguna untuk membangun block untuk cipher karena menjamin beberapa tingkat difusi. Jika satu dari elemen masukan berubah, setiap elemen keluaran harus berubah. Twofish menggunakan sebuah matriks MDS berukuran 4 x 4.

4. Pseudo-Hadamard Transforms

Pseudo-Hadmard Transform (PHT) adalah sebuah operasi pencampuran sederhana yang dijalankan secara cepat pada sebuah perangkat lunak tertentu [9]. Jika diberikan dua buah masukan, a dan b, PHT 32-bit didefinisikan sebagai:

a’ = a + b mod 232

b’ = a + 2b mod 232 (1)

Twofish menggunakan PHT 32-bit untuk mencampurkan keluaran dari dua buah fungsi pararel 32-bit g.

5. Whitening

Whitening merupakan teknik untuk melakukan XOR pada material kunci sebelum round pertama dan setelah round terakhir. Metode ini mempersulit para penyerang untuk membongkar algoritma Twofish. Twofish melakukan XOR kepada 128bit subkunci sebelum jaringan Feistel pertama, dan XOR terhadap 128bit yang lain setelah Feistel terakhir.

6. Fungsi F

Fungsi F adalah permutasi yang bergantung pada kunci pada nilai 64-bit. Fungsi ini memerlukan tiga argumen, dua input sub-plainteks Ro dan R1, dan urutan round r yang digunakan untuk memilih subkunci yang tepat. R0 dioper ke fungsi g, yang menghasilkan T0. R1 kemudian dirotasikan ke kiri sebanyak 8 bit kemudian dioper melalui fungsi g untuk menghasilkan T1. Hasil dari T0 dan T1 kemudian dikombinasikan di dalam PHT dan dua subplainteks ditambahkan [9].

T0 = g(R0)

T1 = g(ROL(R1,8))

F0 = (T0 + T1 + K2r+8) mod 232

F1 = (T0 + 2T1 + K2r+9) mod 232 (2)

F0 dan F1 adalah hasil dari F.

7. Fungsi g

Fungsi g membentuk core dari Twofish. Plainteks masukan X di-split menjadi 4 byte. Masing-masing byte diproses melalui S-box masing-masing [9]. Tiap S-box adalah permutasi 8-bit: membutuhkan masukan masukan 8 bit dan keluaran 8 bit. Empat hasil tersebut diinterpretasikan sebagai komponen vektor dengan panjang 4 GF(28), dan dikalikan oleh matriks MDS 4x4. Berikut adalah matriks MDS:

$$\mathrm{MDS} = \begin{pmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5B \end{pmatrix} \quad (3)$$

8. Penjadwalan Kunci

Penjadwalan kunci harus menyediakan 40 kata dari K0 sampai K39, dan 4 buah S-box yang digunakan pada fungsi g. Twofish didefinisikan untuk panjang kunci 128, 192, dan 256. Kunci dengan panjang kurang dari 256 dapat digunakan dengan cara mem-padding kunci dengan angka 0 sampai memenuhi panjang kunci yang dibutuhkan [9]. Pada gambar II-3, byte plainteks yang berjumlah 16 (p0, ... p15) dipisah menjadi 4 kata (p0, p1, p2, p3) yang masing-masing berukuran 32 bit. Pada

tahap *whitening*, kata-kata tersebut di-XOR-kan dengan 4 kata kunci yang diperlebar :

$R0,1 = Pi \oplus Ki , i = 0,...,3$ (4)

Tiap putaran, dua kata pertama digunakan sebagai masukan terhadap fungsi F, yang juga mengambil nomor putaran sebagai masukan. Kata ketiga di-XOR-kan dengan keluaran dari F dan dirotasi ke kanan 1 bit. Kata keempat dirotasikan ke kiri 1 bit dan di XOR kan dengan kata keluaran kedua dari F. Terakhir, kedua potongan ditukar [11].

$(Fr,0,Fr,1) = F(Rr,0,Rr,1,r)$

$Rr+1,0 = ROR(Rr,2 \oplus Fr,0,1)$

$Rr+1,1 = ROL(Rr,3,1) \oplus Fr,1)$

$Rr+1,2 = Rr,0$

$Rr+1,3 = Rr,1$ (5)

Untuk r = 0,...,15 dan dimana ROR dan ROL adalah fungsi untuk rotasi argumen pertama ke kiri atau ke kanan. Kemudian, tahap whitening akhir membatalkan penukaran untuk putaran akhir, dan melakukan XOR terhadap data dengan menggunakan 4 kata dari kunci.

$Ci = R16,(i+2) \bmod 4 \oplus Ki+4 \; i = 0,...,3$ (6)

Empat kata dari cipherteks kemudian ditulis sebagai 16 byte, c0, ... , c15.

IV. Sistem Operasi Android

Android merupakan sistem operasi yang berbasiskan Linux kernel, dan dirancang untuk perangkat *mobile touchscreen* seperti *smartphone* dan komputer tablet. Android merupakan sistem operasi *open source*. Ada empat keuntungan dari sistem operasi *open source* [2]. Keuntungan pertama, sistem operasi ini gratis. Kedua, semua orang bebas memodifikasi sistem yang ada. Ketiga, pengguna tidak harus menggunakan perangkat lunak berbayar yang hanya bisa bersinergi dengan perangkat lunak berbayar yang lain dari perusahaan yang sama. Keempat, banyaknya orang yang ikut mengembangkan sistem membuat sistem operasi *open source* selalu diperbaharui, dan sistem keamannnya pun lebih baik.

Android merupakan sistem operasi yang paling populer [4]. Dari 227 negara yang terrekam datanya oleh StatCounter, ada 135 negara yang pasarnya dikuasai oleh Android [4].

Android menyediakan developing tools tersendiri untuk para pengembang aplikasi. Pengembangan aplikasi Android menggunakan bahasa pemrograman Java.

V. Review Penelitian Terkait

1. SMS Encryption using AES Algorithm on Android

Rayarikar, dkk. (2012) dalam publikasinya membuat sebuah *client* SMS untuk Android dengan penambahan fitur enkripsi dengan menggunakan AES (*Advanced Encryption Standard*) [7]. *Client* ini aman dari serangan brute-force, statistical, dan pattern analysis karena AES memang merupakan algoritma yang sangat kuat. AES memiliki kemiripan dengan algoritma Twofish. Kedua algoritma ini sama-sama menggunakan S-box, dan proses dalam algoritmanya pun sama-sama berantai. Namun, AES memiliki keunggulan dari segi memori yang dipakai. AES memakan memori lebih sedikit daripada Twofish [11]. Namun demikian, Twofish memiliki keunggulan dari segi jarangnya algoritma ini digunakan.

2. *Cryptography On Android Message Application Using Look Up Table And Dynamic Key* (CAMA)

Madhwani, dkk. (2012) dalam publikasinya menawarkan enkripsi SMS dengan menggunakan acuan sebuah tabel dan kunci dinamis [3]. Sama halnya dengan AES, algoritma ini merupakan algoritma kunci simetri. Hasil dari penelitian ini adalah sebuah algoritma yang lebih hemat memori *storage* karena tabel acuannya tidak disimpan melainkan di-*generate*, lebih mudah diimplementasikan, lebih ringan terhadap prosesor, lebih *confidential*, namun memiliki tingkat keamanan di bawah algoritma-algoritma lain yang sudah ada.

3. Securing SMS using Cryptography

Rangarajan, dkk. dalam publikasinya menawarkan enkripsi SMS dengan menggunakan *Elliptic Curve Cryptography* (ECC) [6]. Berbeda dengan Rayikar, dkk. yang menggunakan AES yang merupakan algoritma kunci simetri, Rangarajan memilih untuk menggunakan algoritma kunci publik. Rangarajan menggunakan ECC karena ECC memiliki kekuatan di permasalahan logaritma diskrit. Selain itu, ECC juga tidak memerlukan komputasi yang terlampau kompleks, sehingga sangat cocok untuk SMS. Dalam publikasinya, Rangarajan juga menawarkan solusi untuk pertukaran kunci. Namun, algoritma ini kurang cocok digunakan karena membutuhkan komputasi yang besar.

VI. Proses Enkripsi Dengan Menggunakan ALgoritma Twofish

Pada Gambar 1, bisa dilihat bahwa SMS memiliki dua buah bagian penting. Bagian pertama adalah header SMS, sedangkan bagian kedua adalah body SMS. Proses enkripsi SMS dilakukan langsung pada body SMS. Enkripsi tidak bisa dilakukan pada header, karena header SMS berisi instruksi-instruksi yang berhubungan dengan operator, sehingga isi dari header tidak boleh diganggu.

Dalam mengimplementasikan algoritma Twofish dalam SMS, langkah-langkahnya adalah sebagai berikut:

1. SMS ditulis, kemudian kunci dimasukkan.
2. SMS dan kunci dikonversi ke dalam bentuk byte.
3. Apabila kunci belum mencapai 128-bit, 192-bit, atau 256-byte, maka tambahkan padding pada byte SMS.
4. Setelah itu dilakukan enkripsi dengan algoritma Twofish.
5. SMS dikirim ke penerima dalam bentuk terenkripsi.
6. Penerima memasukkan kunci.
7. Setelah itu dilakukan dekripsi dengan algoritma Twofish.
8. Penerima menerima pesan dalam bentuk plainteks asli.

VII. Proses Pengiriman Dan Penerimaan SMS Pada SIstem Operasi Android

Untuk mengakomodasi pengiriman dan penerimaan SMS, Android telah menyediakan SDK (*Software Development Kit*). SDK dari Android ini memiliki *library* SMSManager dan SMSMessage. Dalam *library* ini, terdapat prosedur dan fungsi

yang memungkinkan pengembang untuk membuat penerima SMS, dan juga mengirim SMS.

VIII. ANALISIS PERANGKAT LUNAK

Perangkat lunak yang dibangun memiliki kebutuhan fungsional dan non-fungsional sebagai berikut:

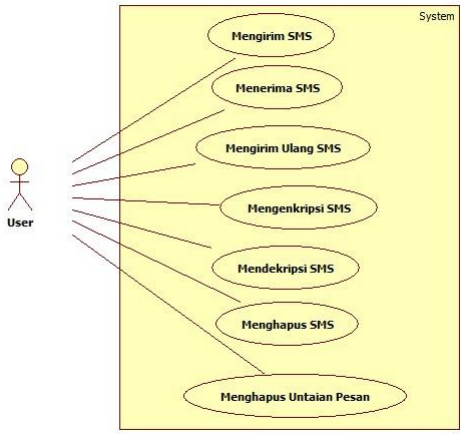
TABEL I
KEBUTUHAN FUNGSIONAL PERANGKAT LUNAK

No	SRS-ID	Deskripsi
1	SRS-F-1	Mampu mengirim SMS
2	SRS-F-2	Mampu menerima SMS
3	SRS-F-3	Mampu mengenkripsi SMS
4	SRS-F-4	Mampu mendekripsi SMS
5	SRS-F-5	Mampu mengirim ulang SMS
6	SRS-F-6	Mampu menghapus SMS
7	SRS-F-7	Mampu menghapus untaian SMS

TABEL 2
KEBUTUHAN NON-FUNGSIONAL PERANGKAT LUNAK

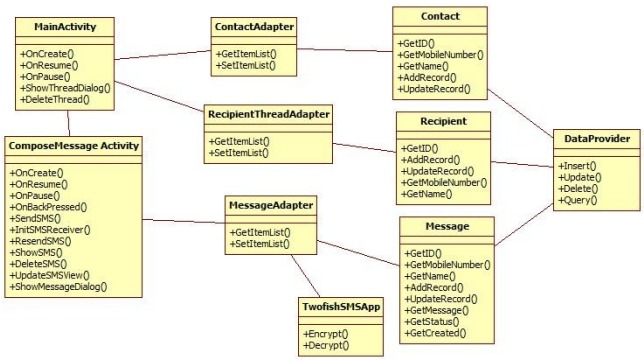
No	SRS-ID	Deskripsi
1	SRS-NF-1	Antarmuka mampu menampilkan untaian pesan
2	SRS-NF-2	Antarmuka mampu menampilkan kumpulan untaian pesan
3	SRS-NF-3	Antarmuka mampu menampilkan kecepatan enkripsi

Kebutuhan-kebutuhan di atas didefinisikan ke dalam sebuah diagram *use case.* Diagram *use case* untuk aplikasi ini dapat dilihat pada gambar 4.



Gambar 4 Diagram *Use Case* Aplikasi.

Dari *use case* yang ada, dirancang sebuah diagram kelas untuk aplikasi ini. Diagram kelas untuk aplikasi ini dapat dilihat pada gambar 5.



Gambar 5 Diagram Kelas Aplikasi

IX. IMPLEMENTASI DAN PENGUJIAN

Implementasi kelas pada aplikasi ini sesuai dengan yang telah dirancang sebelumnya. Algoritma Twofish diimplementasikan dengan menggunakan pustaka GNU Crypto.

Tampilan antarmuka dapat dilihat pada gambar 5.

Berikut adalah tujuan pengujiannya.

1. Mengetahui apakah modul enkripsi dan dekripsi yang dibuat dapat dipergunakan untuk enkripsi pesan.
2. Mengetahui apakah program mampu mengirim pesan terenkripsi dan tidak terenkripsi.
3. Mengetahui apakah program mampu menerima pesan terenkripsi dengan lengkap, dan menerima pesan tidak terenkripsi.
4. Mengetahui apakah pesan terenkripsi masih terenkripsi ketika sampai di penerima.
5. Mengetahui apakah fungsi dasar aplikasi seperti hapus pesan, hapus untaian pesan, dan kirim ulang SMS bisa berjalan dengan baik.
6. Mengetahui apakah enkripsi bisa berjalan tanpa mempengaruhi kinerja ponsel.

Semua hasil pengujian memberikan nilai positif. Program bisa berjalan sesuai dengan yang telah direncanakan. Algoritma Twofish pun bisa berjalan dengan baik. Kecepatan enkripsi pun terbilang sangat baik.

Gambar 5 Halaman Utama Aplikasi

Gambar 6 SMS terkirim dan diterima; Pengukuran waktu Enkripsi

X. KESIMPULAN DAN SARAN

1. Kesimpulan

Implementasi Algoritma Twofish pada ponsel Android dilakukan dengan menggunakan bahasa pemrograman Java. GNU sudah menyediakan library untuk algoritma Twofish, sehingga implementasi algoritma Twofish tidak menjadi kendala dalam pengerjaan Tugas Akhir ini. Algoritma Twofish dapat berjalan dengan sangat cepat, sehingga tidak mempengaruhi komputasi ponsel. Ponsel tetap berjalan dengan baik dengan menggunakan aplikasi ini.

2. Saran

Untuk implementasi lebih lanjut, bisa dilakukan implementasi dengan algoritma kombinasi Twofish-AES. AES bisa diimplementasikan untuk mengamankan untaian pesan, sedangkan Twofish digunakan untuk mengamankan sebuah pesan. Selain itu, untuk membuktikan tingkat keamanan, bisa dilakukan penyadapan.

REFERENCES

[1] Clements, T. (2013, 2). SMS -- Short but Sweet. Diambil kembali dari Oracle: http://www.oracle.com/technetwork/systems/sms-155850.html

[2] Entrepreneur Handbook. (2013, 11 4). Open Source Software – The Advantages & Disadvantages. Diambil kembali dari entrepreneurhandbook: http://www.entrepreneurhandbook.co.uk/open-source-software/

[3] Madhwani, M., C.V., K., & George, J. P. (2012). Cryptography On Android Message Application Using Look.

[4] Mahapatra, L. (2013, 11 11). Android Vs. iOS: What's The Most Popular Mobile Operating System In Your Country? Diambil kembali dari ibtimes: http://www.ibtimes.com/android-vs-ios-whats-most-popular-mobile-operating-system-your-country-1464892

[5] Permana, R. W. (2007). Implementasi Algoritma RC6 Untuk Enkripsi SMS Pada Telepon Seluler.

[6] Rangarajan, S., Ram, N. S., & Krishna, N. V. (2013). Securing SMS using Cryptography.

[7] Rayarikar, R., Upadhyay, S., & Pimpale, P. (2012). SMS Encryption using AES Algorithm on Android.

[8] Scheiner, B. (1997). http://csrc.nist.gov/archive/aes/round1/conf1/twofish-slides.pdf.

[9] Scheiner, B., Kelsey, J., Whiting, D., Wagner, D., & Hall, C. (1998). Twofish : A 128-Bit Block Cipher.

[10] Schneier, B., & Whiting, D. (2000). A Performance Comparison of the Five AES Finalists.

[11] Setiawan, W. (2011). Analisa dan Perbandingan Algoritma Twofish dan Rijndael.

[12] Smid, M. E. (2000). Computer Security Research Center. Diambil kembali dari NIST Computer Security Division: http://csrc.nist.gov/archive/aes/round2/comments/20000523-msmid-2.pdf

[13] Whitfield, K. (2013, 08 09). 17 Incredible Facts about Mobile Messaging that you should know. Diambil kembali dari Portio Research: http://www.portioresearch.com/en/blog/2013/17-incredible-facts-about-mobile-messaging-that-you-should-know.aspx