Pengaman Pengiriman Pesan Via SMS dengan Algoritma RSA Berbasis Android

Andi Riski Alvianto dan Darmaji
Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: darmaji@matematika.its.ac.id

Abstrak*—Saat ini penggunaan Smartphone sangat digemari oleh banyak kalangan, khususnya Smartphone berbasis Android. Mulai dari kalangan masyarakat bawah hingga kalangan atas, semua sudah tidak asing lagi. Salah satu fitur Smartphone yang juga banyak diminati oleh masyarakat adalah SMS. SMS adalah salah satu fitur yang banyak digunakan dan dimanfaatkan oleh masyarakat, namun ada satu masalah yang sering dihadapi oleh masyarakat dalam menggunakan SMS yaitu masalah keamanan pesan SMS. Kriptografi merupakan salah satu solusi yang dapat dimanfaatkan dan dikembangkan dalam menyelesaikan permasalahan keamanan pesan SMS. Dengan melakukan enkripsi pada pesan SMS, maka keamanan pesan SMS akan lebih terjaga dan aman. Kriptografi memiliki banyak teknik dalam melakukan pengenkripsian pesan SMS dan salah satu yang memiliki tingkat keamanan yang tinggi dan tingkat kesulitan dalam pemecahannya adalah algoritma RSA. Pada tugas akhir ini, penulis mengembangkan sebuah aplikasi pada smartphone berbasis Android untuk memodifikasi pesan SMS menjadi ciphertext agar isi informasi dari SMS tersebut tidak diketahui oleh orang lain. Untuk pengiriman SMS sistem mengenkripsi pesan menjadi ciphertext kemudian mengirimkan ke nomor tujuan. Untuk penerimaan SMS sistem mendekripsi ciphertext menjadi plaintext menggunakan kunci yang diinputkan oleh penerima kemudian menampilkan plaintext kepada penerima. Metode yang digunakan dalam mengenkripsi dan mendekripsi pesan adalah algoritma RSA dan implementasinya menggunakan Eclipse IDE.

***Kata Kunci*—Android, kriptografi, RSA, SMS**

I. PENDAHULUAN

SMARTPHONE berkembang pesat saat ini, dunia terasa semakin sempit. Sekarang ini telepon selular (ponsel) seperti menjadi kebutuhan pribadi yang harus terpenuhi untuk bisa saling berkomunikasi antar orang.

Jika membicarakan tentang *smartphone,* maka Android lebih dipilih dibandingkan dengan platform lainnya dikarenakan pasarnya sangat besar. Berbeda dengan iOS yang cenderung eksklusif, Android bisa diaplikasikan pada ponsel apapun, mulai dari ponsel murah hingga yang paling mahal. Inilah yang membuat pasar aplikasi berbasis Android lebih luas [1].

Mobile phone Android dapat digunakan untuk berbagai aktivitas karena memuat banyak fitur dan aplikasi. Salah satu fitur yang digunakan agar dapat saling berkomunikasi dan untuk mengirimkan pesan adalah dengan menggunakan SMS (*Short Message Service*).

SMS merupakan salah satu sarana komunikasi antara semua orang di dunia. SMS sangat populer di Eropa, Asia dan Australia. Teknologi ini banyak dipilih karena selain praktis, biaya yang harus dikeluarkan untuk mengirimkan satu SMS cukup terjangkau (di Indonesia, tergantung operatornya sebuah SMS berkisar antara Rp. 250,- sampai Rp. 350,- saja). Karena faktor inilah, SMS telah banyak digunakan. Namun pesan yang dikirimkan pada orang yang diinginkan belum tentu terjaga kerahasiaannya.

Kasus penyadapan telah ada sekitar 100 tahun yang lalu. Salah satu contoh kasus penyadapan yang terkenal yaitu perkara yang dilaporkan pada tahun 1867 oleh sebuah makelar saham Wall Street bekerjasama dengan Western Union untuk melakukan penyadapan ke operator telegrap yang dikirim ke koran yang ada di Timur Tengah kemudian pesan telegrap tersebut diganti dengan yang palsu.

Berbagai alat komunikasi yang ada saat ini belum tentu aman untuk digunakan, karena belum ada standar keamanan yang dapat digunakan oleh alat-alat tersebut. Untuk itulah akan dibuat suatu perangkat lunak berbasis Android yang berfungsi untuk enkripsi (*encryption*) dan dekripsi (*decryption*) pesan yang telah dibuat tersebut. Proses enkripsi sendiri adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). Untuk melakukan pengamanan pesan khususnya via SMS dengan menggunakan teknik kriptografi dengan menggunakan algoritma RSA.

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma RSA yang dibuat oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1976. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan prima yang relatif besar. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan prima yang besar belum ditemukan algoritma yang berhasil memecahkan, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Dalam tugas akhir ini, seorang pengirim pesan (*sender*) menuliskan pesan yang akan dikirimkan pada penerima pesan (*receiver*), setelah pesan ditulis pengirim akan melakukan proses enkripsi pesan yang akan dikirim agar pesan tersandikan dan tidak dapat terbaca. Aplikasi ini akan menampilkan kunci publik, kunci privat, dan nilai modulus serta pesan yang telah terenkripsi Kemudian pengirim melakukan proses pengiriman pesan yang telah terenkripsi

beserta kunci privat dan nilai modulusnya, setelah terkirim maka pengirim akan menerima laporan pengirimannya. Dari pihak penerima akan menerima pesan yang terenkripsi beserta kuncinya dalam kotak masuk. Penerima akan melakukan proses dekripsi pesan yang tersandikan setelah memilihnya dan menuliskan kuncinya. Setelah proses dekripsi dijalankan maka akan ditampilkan pesan asli yang dikirimkan oleh pengirim pesan agar dapat dibaca.

II. URAIAN PENELITIAN

a. Masalah Keamanan

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Salah satu hal penting dalam komunikasi menggunakan komputer dan dalam jaringan komputer untuk menjamin keamanan pesan, data atau pun informasi adalah enkripsi. Enkripsi dapat diartikan sebagai sebuah proses yang dilakukan untuk mengubah pesan asli menjadi pesan yang tersandikan. Sebuah *cipher* adalah sebuah algoritma untuk menampilkan enkripsi dan kebalikannya dekripsi. Informasi yang asli disebut sebagai *plaintext*, dan bentuk yang sudah dienkripsi disebut sebagai *ciphertext*. Pesan *ciphertext* berisi seluruh informasi dari pesan *plaintext*, tetapi tidak dalam format yang dapat dibaca oleh manusia ataupun komputer tanpa menggunakan mekanisme yang tepat untuk melakukan dekripsi.

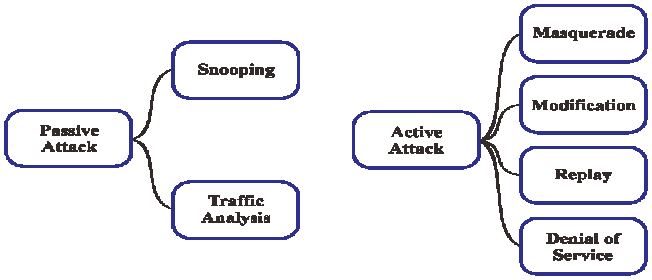
b. Aspek-Aspek Keamanan

Keamanan data dan informasi memiliki beberapa aspek penting, antara lain :

a. Authentification
b. Integrity
c. Non-repudiation
d. Authority
e. Confidentiality
f. Availability [2]

c. Serangan Keamanan

Secara umum serangan pada sistem keamanan dapat dikategorikan menjadi 2 jenis yaitu serangan pasif (*passive attack*) dan serangan aktif (*active attack*) seperti gambar 1 dibawah ini :



Gambar 1 Serangan terhadap keamanan [3]

d. Kriptografi

Kriptografi berasal dari bahasa yunani, menurut bahasa dibagi menjadi dua kata yaitu kripto yang berarti rahasia dan graphia yang berarti tulisan. Menurut terminologinya kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari satu tempat ketempat lain [4].

e. Sistem Kriptografi RSA

Pada tahun 1977, Ronald L. Rivest, Adi Shamir, dan Leonard M. Adleman merumuskan algoritma praktis yang mengimplementasikan sistem kriptografi kunci publik yang disebut dengan sistem kriptografi RSA. Sepasang kunci yang dipakai pada kedua proses ini adalah kunci publik (*e,n*) sebagai kunci enkripsi dan kunci privat *d* sebagai sebagai kunci dekripsi dimana *e*, *d* dan *n* adalah bilangan bulat positif. Algoritma RSA adalah sebuah *block cipher algorithm* (algoritma yang bekerja per blok data) yang mengelompokkan plaintext menjadi blok-blok terlebih dahulu sebelum dilakukan enkripsi hingga menjadi ciphertext.

f. Sistem Operasi Android

Pemakaian sistem operasi Android pada smartphone pada saat ini banyak digunakan oleh perusahaan penghasil telepon seluler. Karena keunggulannya sebagai software yang memakai basis kode komputer yang bisa diunduh oleh pengguna smartphone tanpa membayar biaya aplikasi tersebut. Diyakini smartphone yang menggunakan sistem operasi Android akan lebih murah harganya dibandingkan smartphone yang menggunakan sistem operasi yang berbayar.

g. Short Message Service (SMS)

SMS adalah suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat nirkabel, yaitu perangkat komunikasi telepon selular, dalam hal ini perangkat nirkabel yang digunakan adalah telepon selular. Salah satu kelebihan dari SMS adalah biaya yang murah.

III. ANALISA SISTEM

a. Komparasi Penelitan Sebelumnya

Pada penelitian sebelumnya algoritma RSA digunakan terkait tentang pengamanan pesan via email. Pengamanan pesan yang dilakukan untuk mengenkripsi email agar terjaga kerahasiaannya saat sampai ke penerima pesan yang dituju. Dari penelitian tersebut penulis mengembangkan algoritma RSA untuk pengamanan pesan via SMS berbasis Android [5].

b. Studi Literatur

Pada tahap ini dilakukan studi yang berkaitan dengan permasalahan yang ada. Studi literatur dilakukan terhadap jurnal-jurnal ilmiah, tugas akhir, dan buku-buku yang berhubungan dengan kriptografi, Algoritma RSA, pemrograman Android dan *Short Message Service* (SMS).

c. Analisa Kebutuhan

Kebutuhan perangkat lunak pada penelitian ini adalah yang memiliki kemampuan sebagai berikut :

a. Dapat melakukan proses komputasi dengan cepat, termasuk penanganan *input* dan *output*.
b. Memiliki tingkat ketelitian yang tinggi.

c. Memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak, khususnya yang bersistem operasi Android seperti mengambil kontak dan SMS masuk dari bawaan Android serta mengirim dan menerima SMS.

Dari kebutuhan tersebut *Eclipse IDE* dapat memenuhi semua kriteria, sehingga dapat digunakan untuk membangun sistem pada penelitian ini.

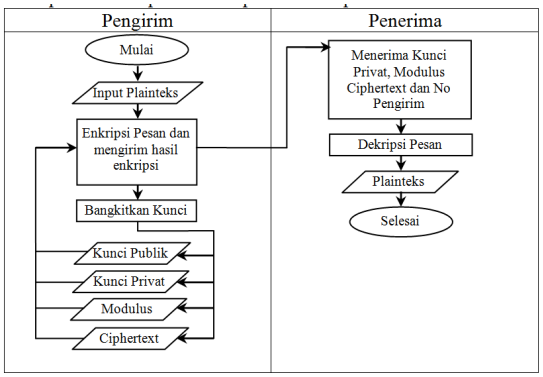
d. Analisa Perancangan Perangkat Lunak

Perangkat lunak pengaman SMS dalam tugas akhir ini dibuat menurut algoritma RSA, mulai dari pembentukan kunci, enkripsi dan dekripsi. Serangan pematahan algoritma RSA pada program pengamanan data ini mungkin dilakukan dengan memanfaatkan persamaan $e.d = 1 \bmod \phi(n)$ atau $e.d = k.\phi(n)+1$. Jika seorang *hacker* masuk sebagai *user* dan mendapat kunci privat d maka *hacker* tersebut dapat menghitung nilai $\phi(n)$ sehingga secara logika nilai kunci privat *user* lainnya bisa dihitung dari nilai kunci publik masing-masing user.

IV. PERANCANGAN PERANGKAT LUNAK

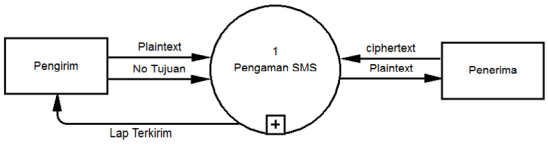
a. Perancangan Flowchart Perangkat Lunak

Dalam penelitian ini *flowchart* dibuat untuk mengetahui langkah-langkah apa saja yang harus diterapkan, agar sistem yang dibuat dapat menghasilkan keluaran yang sesuai dengan harapan berupa hasil dari proses enkripsi dan dekripsi seperti gambar 2 berikut.



Gambar 2. *Flowchart* Perangkat Lunak

b. Perancangan Data Context Diagram

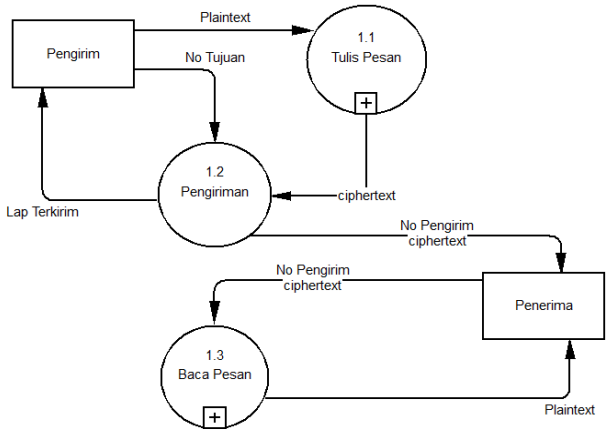


Gambar 3 DCD Aplikasi Pengaman SMS

Pada gambar 3 menjelaskan bahwa pengirim memasukkan plaintext dan nomor tujuan kepada sistem. Sistem memberikan keluaran berupa laporan pesan terkirim kepada pengirim. Penerima memasukkan kunci kepada sistem dan sistem memberikan keluaran berupa plaintext kepada penerima.

c. Perancangan Data Flow Diagram

c.1 DFD Level 1



Gambar 4. DFD level 1

d. Perancangan Proses dari Perangkat Lunak

d.1 Penggunaan Kelas BigInteger

Jika presisi bilangan bulat (*Long*) dan *floating point* sudah tidak memadai, dapat menggunakan kelas *BigInteger* untuk dapat melakukan operasi aritmatika untuk bilangan bulat yang sangat besar sehingga kelas ini sering dipakai dalam aplikasi kriptografi yang melibatkan ratusan bahkan ribuan bit. Pada algoritma RSA, terdapat beberapa besaran yang digunakan. Besaran-besaran tersebut merupakan variabel atau objek dari kelas *BigInteger*. Karena dalam kelas *BigInteger* sudah mempunyai metode-metode khusus dalam kriptografi maka dapat mempermudah dalam proses perancangan.

d.2 Pembuatan Generator Bilangan Prima

Algoritma RSA memerlukan dua bilangan prima *p* dan *q* untuk pembentukan kunci. Oleh karena itu diperlukan sebuah program untuk menentukan bilangan prima. Dalam tugas akhir ini penulis menggunakan angka acak / random yang digunakan untuk memudahkan user agar tidak perlu menginput *p* dan *q* secara manual, tetapi secara otomatis.

d.3 Pembentukan Kunci Publik dan Kunci Privat dengan Algoritma RSA

Setelah proses pembentukan bilangan prima dilakukan maka dipilih dua buah bilangan prima yang cukup besar dengan selisih keduanya relatif kecil. Hitung modulus n dari perkalian *p* dan *q*. Selanjutnya *p* dan *q* akan terus digunakan sebagai bilangan prima pembentuk kunci publik dan kunci privat. Dengan kata lain, proses pemilihan dua bilangan prima sebagai pembentuk kunci publik dan kunci privat hanya dilakukan sekali saja namun tidak menutup kemungkinan suatu saat nilai *p* dan *q* harus diubah demi pembaharuan sistem pengaman data pada aplikasi ini. Dengan p dan q yang sama maka modulus n yang digunakan pada setiap pembentukan *e* pada kunci publik dan kunci privat *d* berikut adalah sama.

d.4 Proses Enkripsi dengan Algoritma RSA

Inti dari proses enkripsi adalah perhitungan $c = m^e \bmod n$ dimana c, m dan n dalam bilangan bulat positif dan e yang telah direpresentasikan dalam biner. Proses pengubahan tiap

karakter pesan dari bentuk text kedalam bentuk bilangan bulat positif sangat diperlukan.

Algoritma 8.2 Enkripsi RSA
Input $K_{publik} = (e,n)$, $m \in Z_n$
Output $c = m^e \bmod n$.

d.5 Proses Dekripsi dengan Algoritma RSA

Dalam proses dekripsi yang menjadi intinya adalah perhitungan $m = c^d \bmod n$ dimana c, m dan n dalam bilangan bulat positif dan d yang telah direpresentasikan dalam bentuk biner. Proses yang digunakan dalam proses deskripsi adalah mengembalikan nilai m yang semula dalam bentuk bilangan bulat (ciphertext) menjadi m ke bentuk text (plaintext). Dengan kata lain proses ini adalah invers dari proses enkripsi.

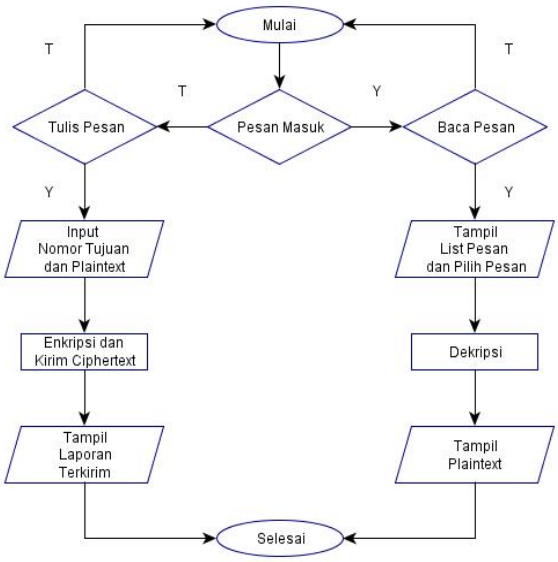
Algoritma 8.2 Dekripsi RSA
Input $K_{privat} = d$, $K_{publik} = (e,n)$, $c \in Z_n$
Output $m = c^d \bmod n$.

d.6 Perancangan Proses Pengiriman dan Penerimaan Pesan

Apabila akan mengirim pesan yang pertama kali dilakukan adalah menulis pesan yang ingin disandikan, kemudian setelah melakukan proses enkripsi pengirim akan melakukan pengiriman pesan yang telah disandikan serta mengirimkan kunci D dan kunci N. Pada saat adanya pesan sandi yang masuk, maka terdapat list-list pesan dalam bentuk ListView atau Array. Kemudian setelah pesan yang diterima masuk kedalam Inbox maka langkah selanjutnya akan dilakukan proses dekripsi untuk menampilkan pesan yang tersandikan oleh proses enkripsi yang kemudian dijadikan pesan asli setelah memasukkan kunci-kunci yang diperlukan.

e. Perancangan Tampilan Perangkat Lunak

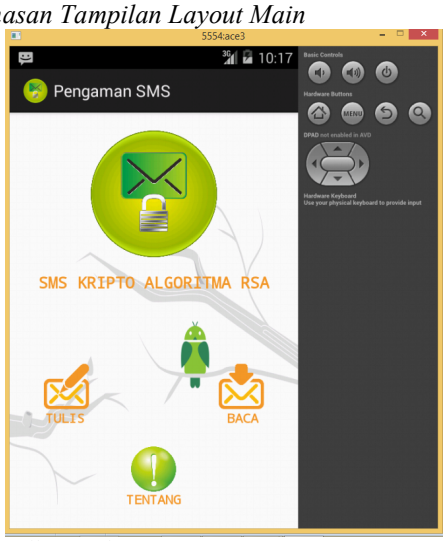
Pada perancangan tampilan ini akan dibahas satu persatu tampilan yang akan dibuat untuk mendukung perancangan proses yang telah dibahas sebelumnya. Diagram alurnya dapat dilihat pada gambar 5 berikut.



Gambar 5 *Flowchart* Aplikasi Pengaman SMS

V.IMPLEMENTASI

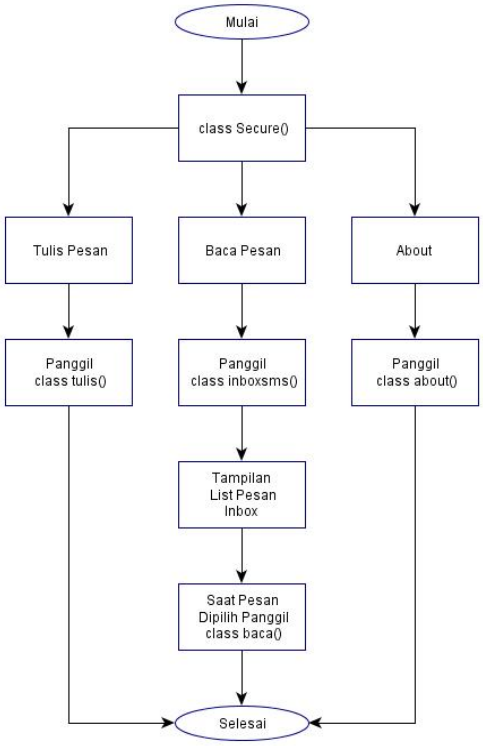
a. Pembahasan Tampilan Layout Main



Gambar 6 Tampilan Awal

Gambar 6 ini dijalankan dengan emulator Android yaitu *Eclipse IDE* dan merupakan awal mula dalam menjalanan proses aplikasi. Pada tampilan awal aplikasi mempunyai tiga buah pilihan yaitu "Tulis Pesan", "Baca Pesan" dan "About".

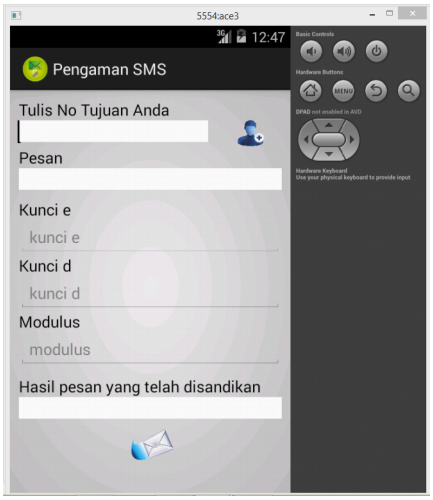
Untuk proses dengan alur *flowchart* dapat digambarkan pada gambar 7 sebagai berikut :



Gambar 7 *Flowchart* Pemanggilan *Class*

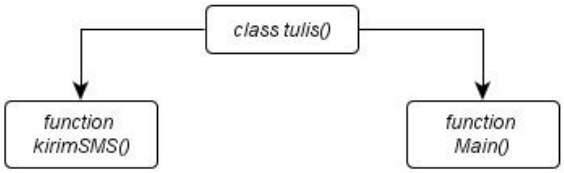
b. Pembahasan Tampilan Layout Tulis Pesan

Dalam tahap ini, disaat pengguna memilih menu "Tulis Pesan" maka tampilan awal dari aplikasi akan memanggil *class tulis()*. Tampilan dari Tulis Pesan adalah seperti gambar 8 berikut.



Gambar 8. Tampilan tulis pesan

Pengirim dapat memasukkan nomor telepon secara manual dan terdapat juga tombol *Add Contact* yang berfungsi mengambil kontak atau nomor telepon secara otomatis ke kontak bawaan Android. *Button* kirim pesan akan berfungsi untuk mengenkripsi pesan yang telah dimasukkan oleh pengirim pesan dan sekaligus mengirimkan pesan yang sudah disandikan yang kemudian akan ditampilkan di *textfield* hasil. Dalam tampilan ini, *class* yang digunakan adalah *class tulis()* yang mempunyai beberapa *function* yaitu seperti pada gambar 9 berikut ini.



Gambar 9. *Function* pada *class tulis()*

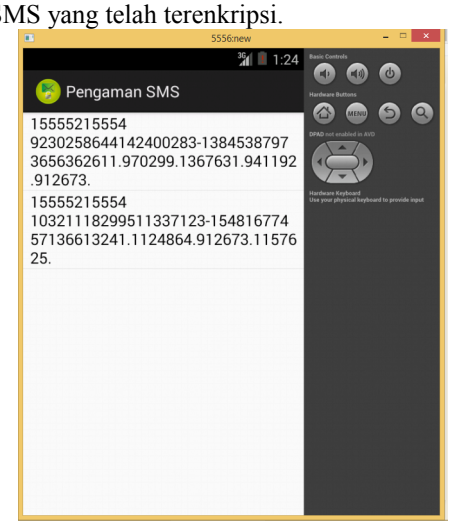
Dalam *function kirim SMS()* mempunyai fungsi yang bertugas untuk mengirimkan pesan yang telah dienkripsi dan nomor telepon pengirim kepada penerima pesan, sementara untuk kunci dikirimkan secara terpisah yaitu dengan menekan tombol kirim kunci.

Perlu diingat bahwa dalam menggunakan metode RSA sesuai algoritma pembentuk kunci publik dan kunci privat, proses penentuan **p** dan **q** haruslah bilangan prima, penulis menggunakan angka acak (*random*) yang digunakan untuk memudahkan pengguna agar tidak perlu menginput p dan q secara manual , tetapi secara otomatis.

c. Pembahasan Tampilan Layout Inbox

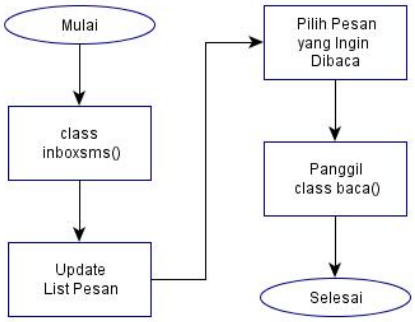
Tampilan ini mempunyai fungsi menampilkan list pesan yang ada di bawaan inbox Android yang nantinya akan digunakan untuk melakukan proses dekripsi pesan. Tampilan dari pilihan Inbox pada gambar 10.

Dalam menampilkan list pesan, pengguna sebelumnya memilih tombol Baca Pesan secara otomatis akan muncul kotak dialog List SMS yang masuk. Pada list SMS hanya menampilkan nomor telepon pengirim dan SMS yang dikirim dan untuk kuncinya dikirim berbeda. List SMS ini secara otomatis update setiap kali ada SMS masuk, baik SMS biasa ataupun SMS yang telah terenkripsi.



Gambar 10. Tampilan inbox

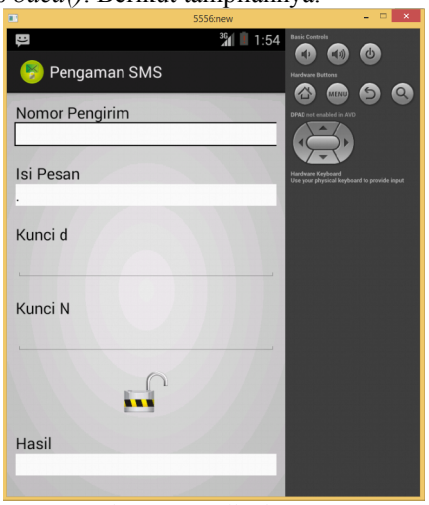
Proses alur dalam bentuk *flowchart* ditampilkan oleh gambar 11 berikut.



Gambar 11. *Flowchart* proses inbox

d. Pembahasan Tampilan Layout Baca Pesan

Proses ini hanya akan berjalan jika pengguna memilih pilihan “Baca Pesan” pada tampilan awal dan klik list pesan yang ingin dibaca. Class yang digunakan dalam tampilan ini adalah *class baca()*. Berikut tampilannya.

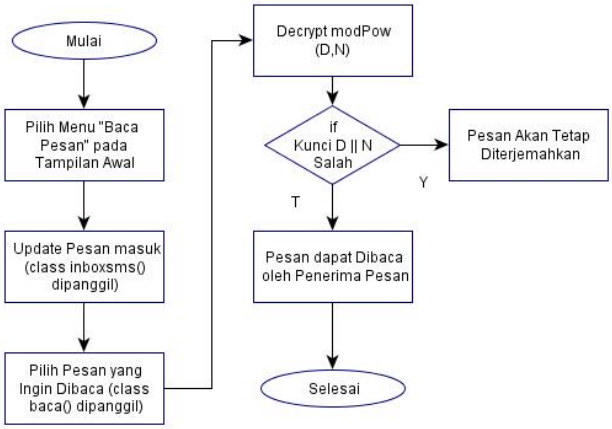


Gambar 12. Tampilan baca pesan

Pada tampilan baca pesan ini terdapat dua buat *textfield* yang harus diisi pengguna agar pesan yang disandikan dapat

dibaca oleh manusia, yang harus diisi pengguna adalah memasukkan kunci D dan kunci N untuk dapat melakukan proses deskripsi pesan. Untuk nomor pengirim dan isi pesan secara otomatis akan terisi ketika penerima pesan memilih pesan yang berada pada list SMS pada tampilan inbox pada aplikasi Pengaman SMS ini.

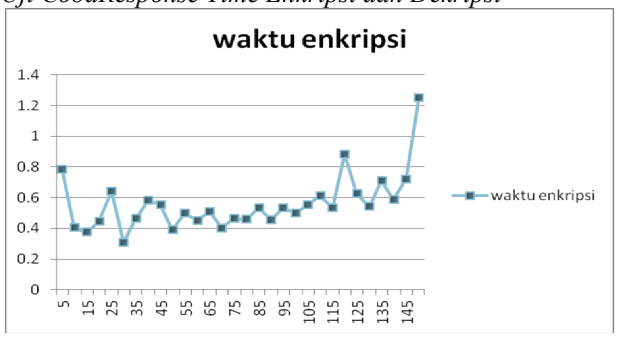
Berikut alur flowchart dalam proses dari tampilan baca pesan pada gambar 13 berikut.



Gambar 13 *Flowchart* proses tampilan baca pesan

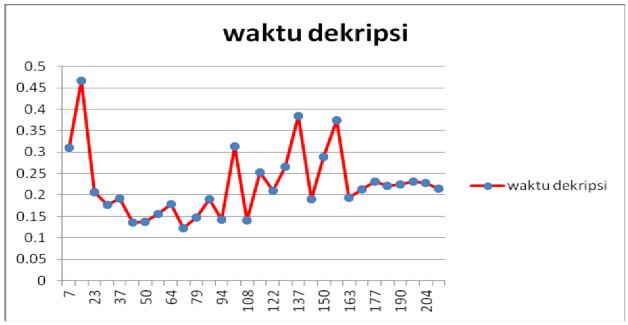
Sebelum melakukan proses dekripsi yang harus dilakukan adalah mengubah tipe data pada kunci dan pesan yang awalnya adalah bertipe string diubah ke tipe data BigInteger.

e. Uji CobaResponse Time Enkripsi dan Dekripsi



Gambar 14. Grafik *Response Time* enkripsi berdasarkan jumlah karakter pesan

Dari percobaan tersebut pada gambar 14 diperoleh *response time* rata-rata proses enkripsi adalah 0,014925 detik per karakter.



Gambar 15. Grafik Response Time dekripsi berdasarkan jumlah karakter pesan

Dari gambar 15 tersebut diperoleh *response time* rata-rata proses dekripsi adalah 0,004679 detik per karakter.

VI. SIMPULAN

1. Validitas pesan yang akan dikirim setelah melakukan proses enkripsi dan dekripsi adalah 100% sehingga untuk keamanan pengiriman pesan asli terjamin aman.
2. Membuktikan bahwa metode RSA tidak hanya cuma digunakan untuk mengamankan data dan *digital signature*, tetapi metode ini dapat diterapkan untuk proses pengiriman dan penerimaan pesan berbasis SMS.
3. Penyamaran pesan menjadi ciphertext pada saat proses enkripsi dengan Algoritma RSA dapat mencegah orang lain untuk mengetahui pesan asli yang dimaksud.
4. *Response time* rata-rata untuk proses enkripsi yaitu 14,925 milidetik per karakter, sedangakan untuk proses dekripsi rata-rata *response time*-nya adalah 4,679 milidetik per karakter.

DAFTAR PUSTAKA

[1] JubileeEnterprise. (2013). "Memahami Pemrograman Android secara cepat dan mudah". Cetakan Pertama. Elex Media Komputindo. Jakarta.
[2] Ariyus, D. (2008). "Pengantar Ilmu Kriptografi – Teori Analisis dan Implementasi". Edisi Pertama. Penerbit ANDI. Yogyakarta.
[3] Sadikin, R. (2012). "Kriptografi untuk Keamanan Jaringan". Edisi Pertama. Penerbit ANDI. Yogyakarta.
[4] Ariyus, D. (2006). "Kriptografi – Keamanan Data dan Komunikasi". Cetakan Pertama. GRAHA ILMU. Yogyakarta.
[5] Kastawan, I K. (2003). "Pembuatan Perangkat Lunak Pengaman Pengiriman Pesan Via Email dengan Algoritma RSA". Tugas Akhir Jurusan Matematika Institut Teknologi Sepuluh Nopember. Surabaya.