ENKRIPSI SMS (SHORT MESSAGE SERVICE) PADA TELEPON SELULAR BERBASIS ANDROID DENGAN METODE RC6

Oleh:

Defni, Indri Rahmayun
Dosen Jurusan Teknologi Informasi Politeknik Negeri Padang

SMS (Short Message Service) is a service to send a sort message that many used in telephone, but in service of a SMS provider have infrequent method to make data of it be more secure, this making service of SMS have so many security gap that making data possibly lost or will be a third person who can access data of SMS. Android is a lateset mobile operating system from Google, android make programmer possibly develop the application easly to use in smartphone and tablet. Cybercryp SMS is an application that designed for android system that will use to secure data that use service of SMS. RC6 algorithm for encryption and decryption will be use in service of sender and receiver message from Short Message Service, that will use a private key and symmetric password for make data will be more secure and security gab can be resolved.
Key Word : SMS (Short Message Servive), Encryption, Decryption, Android.

1.1 Latar Belakang

Perangkat *mobile* saat ini yang disebut dengan *smartphone* memiliki fitur dari teknologi terbaru untuk menjalankan berbagai fungsi layaknya sebuah komputer biasanya, menjadi alasan utama kenapa teknologi *smartphone* menjadi lebih diminati dan menyebabkan produsen *smartphone* menjadi sanggat kompetitif di pasaran.

Adapun beberapa sistem yang saat ini populer digunakan pada perangkat *smartphone* seperti IOS, BlackBerry, Windows Phone dan juga Android. Dari beberapa sistem operasi *smartphone* tersebut, android menjadi yang paling diunggulkan oleh para pengguna dan juga produsen *smartphone* karna fiturnya yang sangat menarik, semenjak perkembanganya pada tahun 2005 dan dirilis pertama kali pada 2008 android sudah memiliki banyak *user* yang tersebar dari seluruh dunia, dengan sebab itu android yang merupakan salah satu sistem operasi *open source* memiliki banyak peminat sehingga memunculkan juga banyak pengembang (*Developer*), sehingga android mendukung perkembangan yang cepat, karena seperti *open source* lainya android membuka code sumbernya secara gratis untuk dikembangkan oleh para *developer*.

Security atau keamanan dari sebuah perangkat android adalah salah satu keunggulan dari perangkat ini, seperti yang diketahui bahwasanya perangkat android memiliki keamanan yang dirancang demi kenyamanan dari pengguna, namun beberapa layanan pada fitur-fitur tertentu yang sama sekali tidak memiliki metode pengamanan pada data yang disimpannya, salah satunya adalah layanan SMS (*Short Massage Service*) yang pada perangkat android secara *default* sama sekali tidak memiliki metoda pengamanan. Sehingga dibutuhkan untuk mengatasinnya software aplikasi pihak ketiga dalam mengamankan *service* SMS tersebut.

Pengamanan atau *security* dari layanan penyedia SMS juga sangat perlu diperhatikan karena pengunaanya yang sangat dominan pada sebuah perangkat telepon dan juga memiliki celah keamanan yang besar yang mungkin menyebabkan datanya diketahui oleh pihak yang tidak dikehendaki, seperti pihak operator memang terkadang menjanjikan kepada pelanggan bahwasanya keamanan akan dijamin dan tidak akan terjadi penyadapan data oleh pihak yang tidak diberhak, namun pada kenyataannya data yang dikirimkan melalui *service* SMS bisa saja akan tersadap atau diketahui dengan gampang saat data SMS tersimpan pada *data center* sebuah *provider*.

Oleh karena itu sebaiknya *user* terlebih dahulu menggunakan aplikasi yang bisa membuat data terenkripsi sebelum dikirimkan melalui *service* SMS sehingga *provider* hanya akan meneruskan pesan yang

sudah dienkripsi pada perangkat *user* pengirim.

1.2 Tujuan

Adapun tujuannya adalah sebagai berikut :

1. Mengimplementasikan enkripsi data pada pengiriman pesan teks pada layanan SMS dengan metode RC6.
2. Membangun aplikasi yang mampu bekerja pada perangkat android, sebagai aplikasi pihak ketiga yang mampu melakukan enkripsi dan dekripsi data text pada layanan SMS sebelum dan sesudah dikirimkan.

1.3 Batasan Masalah

Adapun batasan masalahnya adalah :

1. Aplikasi akan dikembangkan untuk android dengan versi minimal android 2.2 (froyo) dengan level API 8.
2. Aplikasi dibangun dengan target android versi 4.2 (jelly bean) dengan level API 18.
3. Proses enkripsi dan dekripsi pesan hanya digunakan untuk data text.
4. Dalam proses pengiriman pesan mengunakan jaringan *provider* SIM Card pada perangkat android, dan bagi perangkat android dengan fitur dual SIM Card, maka aplikasi akan mendeteksi pengunaan SIM Card pada slot SIM 1 (slot utama).
5. Implementasi program pada *smartphone* android.

2. Landasan Teori

2.1. Java

Java memiliki cara kerja yang unik dibandingkan dengan bahasa perograman lainya yaitu bahasa perograman java bekerja mengunakan *interptreter* dan juga *compiler* dalam proses pembuatan program, *Interpreter* java dikenal sebagai perograman *bytecode* yaitu dengan cara kerja mengubah paket class pada java dengan extensi .java menjadi .class, hal ini dikenal sebagai class *bytecode,* yaitunya class yang dihasilkan agar program dapat dijalankan pada semua jenis perangkat dan juga *platform*, sehingga program java cukup ditulis sekali namun mampu bekerja pada jenis lingkungan yang berbeda.

2.2 Android

Salah satu sistem operasi yang banyak digunakan saat ini adalah android. Hal ini didukung dengan didukungnya beberapa vendor besar, seperti samsung, htc, motorola, LG yang menggunakan sistem operasi ini dalam berbagai gadget yang mereka produksi. Sehingga menjadikan android lebih cepat populer dibandingkan dengan sistem operasi Smartphone lainnya (Edy winarno dan Ali Zaki, 2011:1).

2.1.1. *Developer* Android

Pada perangkat android bekerja dengan aturan yang simpel yaitunya membiarkan *user* mengelola aplikasi yang akan digunakan dengan bebas, dalam pengunaan sistem android pengelolaan aplikasi pemanasangan maupun pencopotan aplikasi dapat menggunakan aplikasi PlayStore yang ada ataupun Amazon Playstore, dengan menggunakan akun google yang sudah terdaftar untuk dapat menggunakan Google PlayStore tersebut. Kita dapat saja menginstall program bawaan dari ponsel dan kemudian mengantinya dengan program pihak ketiga untuk dijalankan pada perangkat yang *user* android miliki, dengan begitu pengunaan program yang datang dari luar perangkat dapat saja langsung digunakan untuk mengantikan dari program bawaan dari ponsel atau perangkat android, sehingga dengan inilah kenapa android dapat dikatakan sistem yang bebas, di google play sendiri pengguna dapat menjelajahi appplikasi yang sudah di upload pada market untuk dapat di pasang pada perangkat pengguna, googleplay sendiri akan memfilter aplikasi mana saja yang *compatible* dengan perangkat yang dimiliki masing-masing pengguna.

Java *Platform* sampai dengan saat ini memiliki 3 profil yaitunya adalah, java *ME* (*Micro Edition*) yaitunya adalah sitem java yang mampu bekerja pada *Embedded System* seperti *Java Card* dan *Heandphone*. Java *SE* (*Standart Edition*) yaitunya adalah versi java yang memiliki kinerja yang ditujukan membangun aplikasi yang mampu bekerja pada *pc, server* yang bersifat *stand alone,* kemudian java *EE* (*Enterprise Edition*) yaitunya adalah java yang ditujukan untuk membuat aplikasi *Enterprise* seperti *Web Aplication* (Servet) dan *Enterprice Java Bean*.

2.1.2. Fitur Perangkat Lunak Android

Dalam perangkat lunak android yang paling menonjol adalah tidak diberikannya akses root pada perangkat android untuk mengakses partisi yang ada pada android seperti pada partisi /sistem. Dikarena untuk mencegah adanya perubahan pada partisi yang hanya bersifat *read-only* dan kemudian juga tidak diinginkanya kesalahan pengembangan pada android dan penyebaran virus dengan membuka langsung akses root tersebut, akses tersebut dapat di dapatkan dengan metoda tertentu pada perangkat android.

2.1.3. Privasi dan Keamanan Pada Android

Pada perangkat android bekerja pada media *SandBox* yang diciptakan Google, dimana pada saat setiap *user* ingin menginstall aplikasi pada market maka akan muncul beberapa perizinan yang sebelumnya harus disetujui oleh *user* sebelum menginstall aplikasi tersebut pada perangkatnya.

2.1.4. Fitur Perangkat Keras Android

Pemahaman pada perangkat keras android dibutuhkan untuk mengetahui bagaimana cara kerja dari sebuah perangkat android dalam mensuport kinerja aplikasi yang diinstall dengan fasilitas *hardware* yang dikembangkan oleh pengembang ponsel android. Ada beberapa perangkat android yang harus dikenali :

1. Touchscreen

Perangkat Android memiliki fitur layar sentuh (*touchscreen*) yang memberikan beberapa kemungkinanan bagi pengguna untuk berinteraksi dengan aplikasi dengan menggunakan jari. Pengguna dapat melakukan *swipe*, *flip*, *drag*, dan *pinch* untuk zoom. Android juga mendukung *multitouch* yang berarti keseluruhan layar dapat disentuh dengan satu atau lebih jari pada saat yang bersamaan.

2. *GPS*

Sistem operasi Android mendukung *GPS* yang memungkinkan *developer* untuk mengakses lokasi pengguna. Contoh aplikasi yang memanfaatkan *GPS* adalah Aplikasi Peta (Map) yang menunjukkan lokasi pengguna dan memberikan petunjuk untuk menuju suatu lokasi.

3. Accelerometer

Android mendukung Accelerometer, yaitu perangkat yang digunakan untuk mengukur percepatan. Accelerometer dapat memberitahukan apabila suatu Perangkat Android bergerak, atau terguncang, atau berbalik arah posisinya.

4. SD Card

Android memiliki fitur yang memungkinkan pengguna atau aplikasi untuk mengakses (menyimpan atau membuka) file pada SD Card. SD Card merupakan media penyimpanan medium yang digunakan Perangkat Android dan beberapa perangkat mobile lain non Android sebagai media penyimpanan.

2.1.5. Arsitektur Android

Menurut Nazruddin Safaat H (2012:7) arsetektur yang ada pada *platform* android adalah sebagai berikut :

1. *Application* dan *Widget*
Application dan *widget* ini adalah *layer* dimana *user* berhubungan dengan aplikasi saja. Dimana *user* hanya akan menanggani interaksi dengan aplikasi dan juga widget.

2. *Application Frameworks*
Android adalah *open development platform* yaitu *android* menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif.

Pengembang bebas mengakses perangkat keras, akses informasi *resources,* menjalankan *services background*, mengatur *alarm*, dan menambahkan status *notification,* dan sebagainya. Pengembang memiliki akes penuh menuju API *Frameworks* seperti yang dilakukan oleh aplikasi kategori inti.

Arsitektur aplikasi dirancang supaya pengembang dengan mudah menggunakan kembali komponen yang sudah digunakan *(reuse)*.Sehingga bisa disimpulkan *application frameworks* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan di sistem operasi *android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa SMS dan panggilan telepon.

3. Android *Runtime*

Layer yang membuat aplikasi *android* dapat dijalankan dimana dalam prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine* merupakan mesin yang membentuk dasar kerangka aplikasi *android*. Didalam *android runtime* dibagi menjadi dua bagian yaitu :

1. *Core Libraries*: aplikasi *android* dibangun dalam bahasa *java*, sementara *dalvik* sebagai *virtual* mesinnya bukan *virtual* mesin *java*, sehingga diperlukan sebuah *Libraries* yang berfungsi untuk menterjemahkan bahasa *java*/*c* yang ditangani oleh *core Libraries*.
2. *Dalvik Virtual Machine*: virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

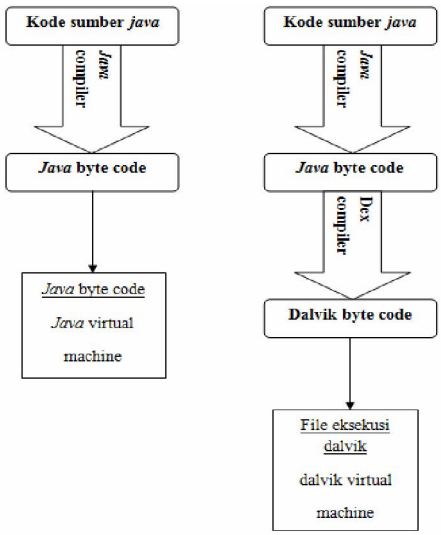
2.1.6. Dasar Pemrograman Android

Pada bahasa *java* programer meng-*compile* menggunakan *java compiler* menjadi *java bytecode*, dan sebuah *java virtual machine* akan menjalankan *java bytecode* tersebut.

Sedangkan pada pemrograman *android* ada sedikit yang berbeda, *programer* meng-*compile* menggunakan *java compiler* yang sama. Namun kemudian perlu di-*compile* ulang menggunakan *dalvik compiler* sehingga menjadi *dalvik bytecode*. Dan *dalvikbyte* code ini kemudian dieksekusi dalam *dalvik virtual machine* (Edy Winarno dan Ali Zaki, 2011:45).

Berikut gambaran secara umum yang membedakan proses pembuatan aplikasi pada *java* dan *android* :

-



Gambar 2.7 : proses *compile* program pada android

2.2. Kriptografi

Pengamanan pesan, data, atau informasi selain bertujuan untuk meningkatkan keamanan, juga berfungsi untuk:

1. Melindungi pesan, data, atau informasi agar tidak dapat dibaca oleh orang yang tidak berhak.
2. Mencegah agar orang – orang yang tidak berhak menyisipkan atau menghapus pesan, data, atau informasi.

Informasi dibagi menjadi dua bagian yaitu informasi yang bersifat umum dan informasi yang bersifat pribadi. Pada informasi yang bersifat pribadi maksudnya informasi yang terkandung hanya untuk satu orang sedangkan informasi yang bersifat umum artinya dapat diketahui orang banyak. Adapun

perjalanan informasi tersebut tidak luput dari ganguan – ganguan pihak yang tidak berhak. Salah satu ilmu untuk menjaga keamanan dan kerahasiaan data atau informasi yaitu kriptografi.

Ada banyak model dan metodelogi enkripsi, salah satunya adalah enkripsi dengan *Riverst Code* 6 (*RC6*). Model ini merupakan salah satu kandidat *Advenced Encryption Standart* (*AES*) yang diajukan *RSA Security Laboratories* kepada NIST. Algoritma ini adalah pengembangan dari algoritma sebelumnya adalah *RC*5 dan telah memenuhi kriteria dari NIST.

Setiap kriptografi atau cryptosystem yang baik memiliki karakteristik sebagai berikut:

1. Keamanan sistem terletak pada kerahasiaan kunci dan bukan pada kerahasiaan algoritma yang digunakan.
2. *Cryptosystem* yang baik memiliki ruang kunci (*keysspace*) yang besar.
3. *Cryptosystem* yang baik akan menghasilkan *chipertext* yang terlihat acak dalam seluruh tes static yang dilakukan.
4. *Cryptosystem* yang baik mampu menahan seluruh serangan yang telah dikenal sebelumnya.

2.4.1. Aspek – Aspek Keamanan

Kriptografi tidak hanya memberikan kerahasiaan dalam telekomunikasi, namun juga memiliki sejumlah aspek, yaitu :

1. *Authentication*. Penerima pesan dapat memastikan keaslian pengirimnya. Penyerang tidak dapat perpura-pura sebagai orang lain.
2. *Integrity*. Penerima harus dapat memeriksa apakah pesan telah dimodifikasi ditengah dijalan atau tidak. Seorang penyusup seharusnya tidak dapat memasukan tambahan ke dalam pesan, mengurangi atau mengubah pesan selama data berada di perjalanan.
3. *Nonrepudiation*. Pengirim seharusnya tidak dapat mengelak bahwa dialah pengirim pesan yang sesungguhnya. Tanpa kriptografi, seseorang dapat mengelak bahwa dia yang mengirim email yang sesungguhnya.
4. *Authority*. Informasi yang berada pada sistem jaringan seharusnya hanya dapat dimodifikasi oleh pihak yang berwenang.

2.4.2Algoritma Kriptografi *RC*6

Algoritma enkripsi RC6 adalah suatu algoritma yang menggunakan kunci *private*, mampu bekerja dengan panjang kunci yang beragam dan mengunakan prinsip *intered chiper*.

Algoritma RC6 merupakan salah satu kandidat *Advanced Encryption Standard* (*AES*) yang diajukan oleh *RSA Security Laboratories* kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu *RC*5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. *RC*6 adalah algoritma yang menggunakan ukuran blok hingga 128 bit, dengan ukuran kunci yang diguankan bervariasi antara 128, 192 dan 256 bit.

Algoritma *RC*6 dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai *RC*6-w/r/b. Parameter w merupakan ukuran kata dalam satuan bit, parameter r merupakan bilangan bukan negatif yang menunjukan banyaknya iterasi selama proses enkripsi dan parameter b menunjukan ukuran kunci enkripsi dalam byte. Setelah algoritma ini masuk dalam kandidat *AES*, maka ditetapkan bahwa nilai w = 32, r=20 dan b bervariasi antara 16, 24 dan 32 byte.

1.5.1. Algoritma Enkripsi Kriptografi *RC*6

Karena RC6 memecah blok 128 bit menjadi 4 buah blok 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. Byte yang

pertama dari *plaintext* atau *ciphertext* ditempatkan pada *byte* A, sedangkan *byte* yang terakhirnya ditempatkan pada *byte* D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.

2.4.3 Algoritma Dekripsi Kriptografi *RC*6

Proses dekripsi ciphertext pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses *whitening*, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses whiteneing setelah iterasi terakhir diterapkan sebelum iteasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada *whitening* setelah iterasi terkahir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik. Berikut ini adalah algoritma dekripsi *RC*6.

2.4.4 Penjadwalan Kunci

Pengguna memasukkan sebuah kunci yang besarnya b byte, dimana $0 \leq b \leq 255$. byte kunci ini kemudian ditempatkan dalam array c w-bit words L[0]…L[c-1]. Byte pertama kunci akan ditempatkan sebagai pada L[0], byte kedua pada L[1], dan seterusnya. (Catatan, bila b=0 maka c=1 dan L[0]=0). Masing-masing nilai kata w-bit akan dibangkitkan pada penambahan kunci round 2r+4 dan akan ditempatkan pada array S[0,…,2r+3].

Konstanta P32 = B7E15163 dan Q32 = 9E3779B9 (dalam satuan heksadesimal) adalah "konstanta ajaib"yang digunakan dalam penjadwalan kunci pada RC6. nilai P32 diperoleh dari perluasan bilangan biner e-2, dimana e adalah sebuah fungsi logaritma. Sedangkan nilai Q32 diperoleh dari perluasan bilangan biner ∅-1, dimana ∅ dapat dikatakan sebagai "golden ratio" (rasio emas).

2.4.5 Keamanan Algoritma *RC*6

Dalam algoritma enkripsi, panjang kunci yang biasanya dalam ukuran bit, juga menentukan kekuatan dari enkripsi. Kunci yang lebih panjang biasanya lebih aman daripada kunci yang pendek. Jadi enkripsi dengan menggunakan kunci 128-bit lebih sukar dipecahkan dengan algoritma enkripsi yang sama tetapi memiliki kunci 56-bit. Semakin panjang sebuah kunci, semakin besar *keyspace* yang harus dijalani untuk mencari kunci dengan cara brute force attack atau coba-coba karena *keyspace* yang harus dilihat merupakan pangkat bilangan dari 2. Jadi kunci 128-bit memiliki *keyspace* 2128, sedangkan kunci 56-bit memiliki keyspace 256. Artinya semakin lama kunci baru bisa ditemukan.

Pada intinya, keamanan suatu pesan tidak tergantung pada sulitnya algoritma tetapi pada kunci yang digunakan. Pada *RC*6 dengan adanya fungsi f(x)=x(2x+1) yang diikuti pergeseran lima bit kekiri dapat memberi tingkat keamanan data yang tinggi. Adanya avalanche effect juga memberikan cukup kesulitan kepada kriptanalis untuk melakukan serangan.

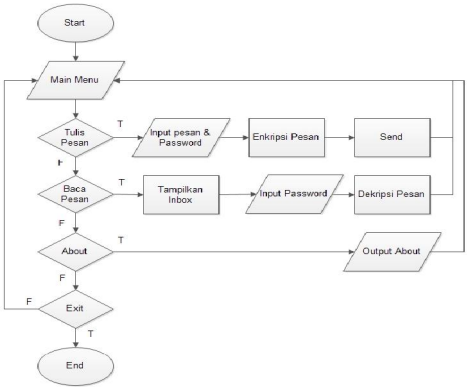
Kekurangan umum dari algoritma yang berbentuk simetris atau kunci pribadi adalah pada kunci itu sendiri. Kelemahan ini timbul jika terdapat banyak orang yang ingin saling berkomunikasi, karena setiap pasangan maupun file enkripsi mempunyai *key* berbeda yang harus disepakati sehingga *key* tiap orang maupun file harus menghafal banyak kunci dan menggunakannya dengan tepat.

3. **Perancangan**.

Aplikasi yang akan akan memanfaatkan sistem operasi *smartphone* yang memiliki *platform* android yang mampu untuk dikembangkan dengan tool developer android, dengan megunakan algoritma kunci simetris (*symectric key*) pada proses enkripsinya. Aplikasi akan memiliki fungsi pengiriman dan juga menerima pesan. Pengguna akan berinteraksi dengan perangkat lunak melalui *user interace* yang memuat konten pesan masuk dan pesan yang akan dikirim dengan cara *user* memasukan input langsung pesan dan password dan akan mendapatkan hasil enkripsi dan juga dekripsi dari pesan, dan aplikasi akan mengirimkan melalui service SMS. Aplikasi ini berjalan pada sistem operasi *android* versi 2.2 *(froyo)* ke atas yang mempunyai *API (Application Programing Interface)* minimum level 8.

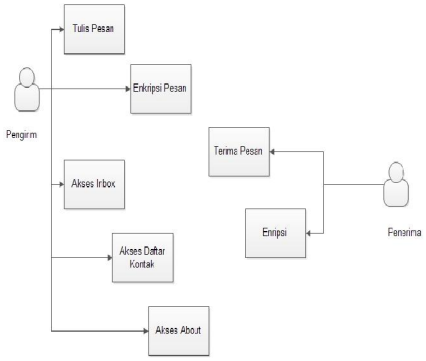
3.1 *Flowchart* Diagram

Pada aplikasi ini *user* akan menjalankan sebuah *user* interface yang menampilkan menu applikasi dalam beberapa tahapan untuk membuat sebuah enkripsi SMS. Pada halaman awal atau *main menu* dari program *user* hanya akan melihat empat buah tombol yaitu tulis pesan, baca pesan, about, dan exit. Pada tombol tulis pesan akan berfungsi untuk menampilkan *layer* baru yang berguna menulis *plaintext* dan applikasi akan menjalankan fungsi untuk menghasilkan sebuah *chipertext* yang akan dikirim ke penerima SMS, kemudian pada tombol baca pesan, akan mengaktifkan fungsi untuk melihat kotak masuk dari pesan yang terdapat pada *inbox* smartphone *user*, pada tombol about akan menampilkan versi dari aplikasi dan beberapa dekripsi mengenai aplikasi, dan juga terdapat tombol exit yang akan mengizinkan *user* untuk keluar dari program.



Gambar 3.1 : Gambar flowchart program

Pada aplikasi memiliki alur dalam pengunaanya, adapun *usecase* dari aplikasi adalah sebagai berikut :



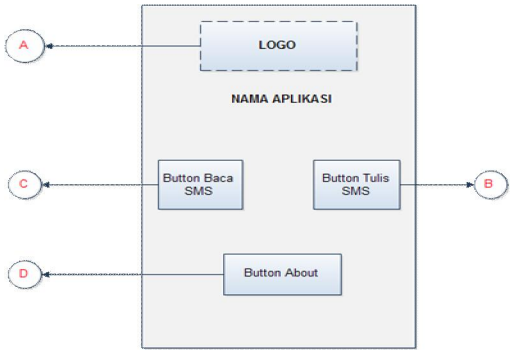
Gambar 3.2 : *Usecase* aplikasi enkripsi SMS

3.2 Rancangan Aplikasi

1. Perancangan Tampilan
   Perancangan Tampilan terdiri dari beberapa menu yaitu :

a. Rancangan Menu Awal

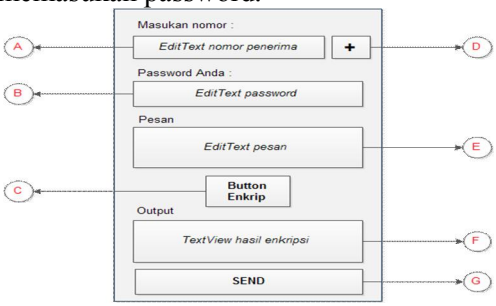
Rancangan pada menu awal terdiri dari tampilan yang terdiri dari beberapa *button,* yaitunya *button write* SMS (tulis pesan), *button read message* (baca pesan), *button about*, dan juga *button exit*.



Gambar 3.2.1 : Layout tampilan awal program

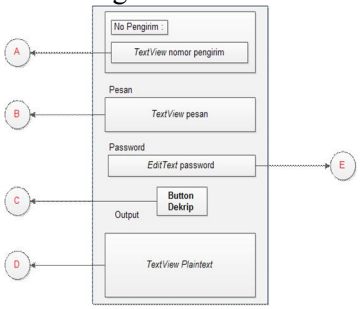
b. Rancangan Menu Tulis SMS

Pada menu ini menampilkan *user interface* untuk melakukan enkripsi data pada pesan yang akan dikirimkan oleh *user*. Akan terdapat beberapa tombol pada menu ini. Yaitunya terdapat tombol enkripsi SMS, tambah kontak, dan juga adanya *edit text* untuk memasukan pesan dan kemudian *text view* untuk melihat *chipertext*, dan juga *edit text* untuk memasukan password.



Gambar 3.2.2 : Layout tampilan Tulis SMS

c. Rancangan Menu Baca SMS



Gambar 3.2.3 : Layout menu Baca SMS

4. Implementasi dan Pengujian

Program enkripsi SMS pada *smartphone* android dirancang dengan tujuan meminimalkan celah keamanan yang terdapat pada jalur komunikasi yang menggunakan layanan SMS, program bekerja dengan melakukan proses enkripsi dan dekripsi untuk tipe data text.

4.1. Implementasi

Aplikasi yang dirancang kali ini akan berjalan pada *smartphone* yang memiliki sistem :

1. Menggunakan sistem android minimal android 2.2 (Froyo) dan *compatible* dengan versi android 2.2 keatas dan Memiliki API minimal versi 8.

2. Perangkat memiliki layanan pengiriman SMS dengan jalur provider pada SIM Card yang aktif.

Pengembangan aplikasi membutuhkan beberapa aplikasi pendukung atau *software requirement* untuk membangun aplikasi android. Beberapa *software* yang dibutuhkan tersebut adalah :

1. JDK (*Java Development Kit*)
2. Eclipse
3. Android SDK
4. ADT (*Android Development Tools*)

Langkah-langkah yang dilakukan untuk instalasi aplikasi

1. Installasi *Software Requirement*

Langkah-langkah yang dilakukan dalam software requirement android adalah:

a. Installasi JDK (*Java Development Kit*)

JDK dibutuhkan sebagai interpreter yang akan digunakan pada pemrograman android

b. Installasi Eclipse

Eclipse akan digunakan sebagai editor pemrograman android dalam pengembangan aplikasi enkripsi SMS, pada tahapan installasi eclipse yang perlu dilakukan adalah menginstall ADT dan SDK sebagai tool yang dibutuhkan untuk membangun program android.

c. Persiapan Pembuatan Aplikasi

Setelah installasi eclipse selesai digunakan maka pembuatan project android bisa

langsung dilakukan, Langkahnya adalah sebagai berikut :

1. Pembuatan Project

Langkah awal dalam membangun aplikasi android yaitunya dengan membuat project android aplication project untuk memulai melakukan penulisan program. langkah-langkahnya seperti berikut inn:

1. Jalankan Eclipse.exe dan klik file → new → Project dan pilih "android application project".
2. Berikan nama aplikasi android yang akan dibanggun dan juga tentukan target sistem android yang akan digunakan.
3. Lanjutkan konfigurasi android pada settingan awal aplikasi sampai selesai dengan memberikan nama activity dan file java.

2. Mempersiapkan Konten Resource String

Konten String.xml adalah salah satu file xml yang disimpan di dalam folder "android_package/res/value" pada workspace aplikasi, string,xml berfungsi sebagai penyimpanan bagi data dengan tipe data String yang akan digunakan pada aplikasi android nantinya seperti label dan juga nama aplikasi.

3. Mempersiapkan Konten Resource Drawable

Dalam proses pengembangan aplikasi diperlukan beberapa file multimedia dalam format gambar untuk digunakan pada tampilan *user interface* nantinya, file tersebut akan memiliki tempat penyimpanan pada *workspace* aplikasi yaitunya folder "android_pakage / res / drawable", dalam hal ini program android mengunakan format gambar .png untuk file gambarnya dan .xml untuk namambahkan script yang diperlukan.

4. Pembuatan Layout XML dan Kode Program Java

Untuk merancang program ada beberapa tahapan pembuatan layout sebagai user interface, dan penambahan class pemrograman java sebagai penyimpan script yang dijalankan pada aplikasi nantinya. Adapun layout dan fungsi pada kode program pada aplikasi yaitu :

a. Pembuatan XML Layout Dan Kode Program Java Menu Utama

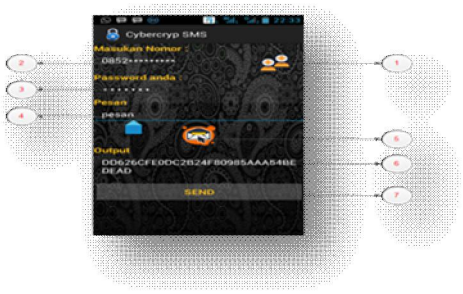
Tampilan utama aplikasi berfungsi untuk menampilkan tombol pilihan dari beberapa fungsi aplikasi yang akan dipilih oleh user, tampilan dirancang agar mudah dipahami oleh user.

Pada layout main.xml akan diatas dipanggil melalui file java CyberSMS.java yang akan menjadikan main.xml sebagai *user interface* pertama yang dilihat oleh user, adapun kode programnya dalam pemanggilan layout yaitu sebagai berikut :

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
setContentView(R.layout.main);
```

b. Pembuatan XML Layout Dan Kode Program Java Tulis SMS

Layout tulis SMS berfungsi sebagai user interface untuk menjalankan fungsi input pesan, password dan juga melakukan proses enkripsi dan pengiriman pesan.



Gambar 4.1 Tampilan menu tulis sms

Keterangan :

1. *Button* kontak
2. *EditText* nomor kontak penerima
3. *EditText* password
4. *EditText* pesan (plaintext)
5. *Button* enkripsi

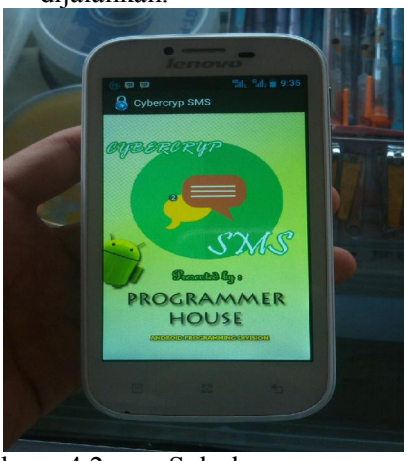
6. *TextView* chipertext
7. *Button* Send

4.3. Pengujian Program

Pengujian program akan mengunakan perangkat *smartphone* Lenovo Armani A706 dengan versi android 4.2 (Jelly bean). Adapun beberapa pengujian aplikasi yaitu :

1. Pengujian Splashscreen

Splashscreen akan tampil selama 3,5 pada saat aplikasi pertama dijalankan.



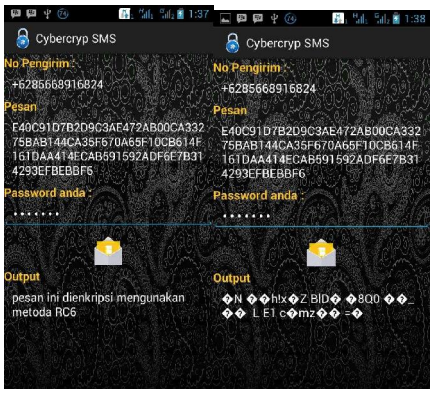
Gambar 4.2 : Splashscreen program berjalan

2. Pengujian Akses Inbox

Pada saat user akan memilih menu tulis pesan maka diperlukan nomor telepon dari penerima untuk menjadi tujuan dari pengiriman pesan, kontak dapat diinputkan langsung ataupun dapat diimport dari kontak pada memory perangkat android. Berikut adalah pengetesan saat tombol tambah kontak ditekan.

3. Pengujian Proses Cryptografi

Pengujian dilakukan pada SMS yang berisi chipertext di dekripsikan kembali dengan menggunakan password yang benar dan juga yang salah.



Gambar 4.4 : Tampilan saat password dekripsi pesan benar dan salah

PENUTUP

5.1. Kesimpulan

1. Aplikasi Cybercryp SMS adalah aplikasi yang berfungsi untuk mengamankan data text yang dikirim dari service SMS agar celah keamanan data pada layanan SMS dapat dikurangi.
2. Aplikasi Cybercryp SMS menggunakan enkripsi dengan kunci simetris RC6 dengan panjang password yang tidak dibatasi.
3. Aplikasi memiliki *user interface* yang menarik sehingga akan membantu user dalam menggunakan aplikasi ini karena aplikasi dirancang *user friendly*.
4. Aplikasi ini paketkan menjadi sebuah file APK yang merupakan paket aplikasi agar bisa diinstall ke perangkat *smartphone* atau tablet android untuk bisa digunakan mengirim ataupun membaca pesan Cybercryp SMS sehingga data yang tidak diinginkan untuk diketahui pihak lain bisa diminimalisir.
5. Aplikasi berjalan pada *smartphone* android dengan versi minimal 2.2 (Froyo), dengan API

minimal versi 8 dan competible dengan versi android terbaru.

5.2. Saran

1. Aplikasi bekerja untuk data text pada proses enkripsi dan dekripsi RC6 yang dijalankan pada saat mengirim ataupun menerima pesan, namun aplikasi masih belum bisa digunakan untuk mengarsipkan sebuah file ke dalam aplikasi yang juga bisa dienkripsi dan juga dikirimkan melalui aplikasi Cybercryp SMS ini. Fitur untuk mengarsipkan file sangat dianjurkan untuk digunakan pada pengembangan aplikasi ini selanjutnya.
2. Service pengiriman SMS pada aplikasi hanya bisa menggunakan layanan yang ada pada SIM Card yang terdeteksi sebagai SIM 1 atau SIM utama, sehingga pada perangkat android yang memiliki dual SIM Card atau lebih tidak dapat melakukan pemilihan kartu SIM untuk melakukan proses pengiriman pesan, sehingga sebaiknya fungsi pemilihan SIM disertakan pada pengembangan aplikasi ini selanjutnya.

Daftar Pustaka

Hermawan, Stephanus S.(2011). *Mudah Membuat Aplikasi Android.* Yogyakarta:ANDI.

H, Nazruddin Safaat. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Tablet*

PC Berbasis Android. Bandung: Informatika Bandung.

Huda,Arif Akbarul.(2012). *24 Jam Pintar Pemrograman Android*. Yogyakarta:ANDI.

Kadir, Abdul. (2003). *Dasar Pemrograman Java 2.* Yogyakarta: ANDI.

Raharjo, Budi dan Imam Heryanto.(2012). *Mudah Belajar Java Revisi Kedua.* Bandung:BI-OBSES.

Salahuddin, M dan Rosa A.S. (2007). Belajar *Pemrograman dengan bahasa C++ dan java* . Bandung: Informatika Bandung

Sutedjo, Budi dan Michael AN. (2004). *Algoritma dan Teknik Pemrograman.* Yogyakarta: ANDI.

Taufik, Andik. (2010). *Pemrograman Grafik dengan java.* Informatika Bandung: Bandung.

Winarno, Edi. (2011). *Membuat Sendiri Aplikasi Android untuk Pemula.* Jakarta : PT. Elex Media Komputindo.