BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisa Masalah

Keamanan atau sering dikatakan kriptografi, kriptografi menjadi salah satu pilihan dalam pengamanan untuk menghindari jika terjadinya ancaman yang dilakukan oleh orang yang tidak bertanggung jawab. Saat ini, keamanan dalam suatu aplikasi sangat dibutuhkan oleh banyak orang termasuk keamanan SMS pada perangkat *Mobile Phone Andoid.*

Sms merupakan pesan tulisan yang ingin disampaikan atau dikirim ke orang yang dituju dengan menggunakan *mobile phone*. Salah satu dukungan yang ada pada perangkat seluler, pengguna dapat menggunakan pesan *SMS* atau dalam bahasa Indonesia adalah pesan singkat. Banyak pengguna telepon seluler yang menggunakan layanan ini, dikarenakan biaya yang cukup murah dan hanya menggunakan teks dalam berkomunikasi. Namun tingkat keamanan pada layanan *SMS* tidak ada tetapi masih belum terjamin dalam pengamanan pesan teks terkirim atau diterima. Hal ini yang cenderung menimbulkan bahaya bagi pengguna yang memiliki pesan-pesan pribadi penting, sehingga dapat disalah gunakan oleh pihak yang tidak bertanggung jawab. Di dalam keamanan komputer dikenal sebuah teknik kriptografi, yang difungsikan untuk penyandian pesan. Berdasarkan kepentingan dan kerahasiaan sebuah pesan diperlukannya sebuah cara untuk mengamankan suatu pesan atau informasi dengan menggunakan teknik kriptografi.

III.2. Spesifikasi Perangkat

Dalam perancangan aplikasi pengamanan SMS untuk perangkat *mobile android* ini penulis membutuhkan beberapa perangkat agar aplikasi berjalan dengan baik dan sesuai dengan yang diharapkan, yaitu sebagai berikut :

1. Perangkat Lunak (*Software*)
    a. *Operating System*, OS yang digunakan dalam perancangan dan tes untuk adalah *Windows* 7 dan OS *Android*pada perangkat *mobile*.
    b. JDK *Java 1.7*, sebagai bahasa program dan *compiler Java*.
    c. *Eclipse*, sebagai *editor source code Java.*
2. Perangkat Keras (*Hardware*)
    a. Komputer yang setara *Core i3*.
    b. *Smartphone Android* dengan OS 4.1 atau diatasnya.
    c. *Mouse*, *keyboard*, dan *Monitor.*

III.2.1. Teknik Pemecahan Masalah

Adapun teknik pemecahan masalah tentang perancangan aplikasi keamanan sms yang dibuat terdiri dari beberapa poin yaitu sebagai berikut:

1. Untuk langkah awal analisa terhadap perancangan yang akan dibangun terutama tentang keamanan sms yang menggunakan perangkat *mobile phone android.*
2. Memilih atau Menentukan perangkat yang dibutuhkan dalam membangun aplikasi seperti perangkat keras maupun perangkat lunak.
3. Merancang sistem yang nantinya akan di implementasikan pada aplikasi yang

akan dibangun.

4. Berikutnya merupakan proses uji coba terhadap *input*an, proses ataupun *output* aplikasi, apakah sudah sesuai dengan perancangan yang telah direncanakan sebelumnya dan mencapai tujuan yang diharapkan.

III.2.2. Penerapan Algoritma *Run Length Encoding* Dan *Knapsack*

Algoritma *Knapsack* merupakan bagian dari kriptografi asimetri yang mana kunci enkripsinya sama dengan kunci dekripsinya. Di samping masalah keamanan SMS, masalah ukuran dari sebuah SMS juga menjadi pertimbangan. SMS yang berukuran besar dapat dimampatkan dengan melakukan proses kompresi. Algoritma *Run Length Encoding* (RLE) merupakan algoritma yang mengecilkan ukuran *file* teks, jika teks tersebut mengalami banyak perulangan karakter. Kombinasi algoritma *Knapsack* dan RLE dapat menjamin *file* Teks tidak dapat dilihat oleh pengguna yang tidak berhak dan dapat menjamin SMS dapat disimpan dalam media *file* yang berkapasitas rendah. Pada algoritma *Knapsack* akan terjadi penambahan ukuran *file* teks, contoh sederhana adalah 9 bytes, kemudian setelah dilakukan proses enkripsi ukuran *file* teks menjadi 37 *bytes*. Pada algoritma kompresi RLE terjadi pengurangan ukuran sebagai contoh kasus cipherteks (pesan yang disamarkan) yang awalnya berukuran 37 *bytes* setelah dilakukan proses kompresi ukurannya menjadi 7 *bytes*. Proses kombinasi dapat dilakukan sebaliknya, yaitu kompresi teks terlebih dahulu kemudian enkripsi teks tersebut, masih dengan plainteks yang sama berukuran 9 *bytes*, setelah dilakukan kompresi ukuran *file* menjadi 2 *bytes*. Kemudian dilakukan enkripsi ukuran *file*

menjadi 9 *bytes*. Karena itu pengunaan kombinasi enkripsi dan kompresi data lebih baik karena *file* menjadi lebih kecil dibandingkan kombinasi kompresi dan enkripsi data. *Plainteks* yang memiliki banyak perulangan karakter akan terkompresi dengan baik.

a. Cara perhitungan Algoritma *Knapsack* :

Algoritma ini didasarkan pada persoalan *1/0 Knapsack Problem* yang berbunyi:

Diberikan bobot *knapsack* adalah *M*. Diketahui *n* buah objek

yang masing-masing bobotnya adalah w_1, w_2, …, w_n. Tentukan nilai b_i sedemikian sehingga

$M = b_1w_1 + b_2w_2 + ... + b_nw_n$ (1)

yang dalam hal ini, b_i bernilai 0 atau 1. Jika $b_i = 1$, berarti objek *i* dimasukkan ke dalam *knapsack*, sebaliknya jika $b_i = 0$, objek *i* tidak dimasukkan.

Ide dasar dari algoritma *knapsack* adalah mengkodekan pesan sebagai rangkaian solusi dari dari persoalan *knapsack*. Setiap bobot w_i di dalam persoalan *knapsack* merupakan kunci rahasia, sedangkan bit-bit plainteks menyatakan b_i.

Contoh : Misalkan $n = 7$ dan $w_1 = 1$, $w_2 = 5$, $w_3 = 6$,

$w_4 = 11$, $w_5 = 14$, $w_6 = 20$ dan $w_7 = 28$

Contoh Teks : R I F A I

Plainteks: 1010010,1101001,1100110,1100001,1101001

Plainteks dibagi menjadi blok yang panjangnya *n*, kemudian setiap bit di dalam blok dikalikan dengan w_i yang berkorepsonden sesuai dengan persamaan (1):

Blok plainteks ke-1 : 1010010

Knapsack : 1, 5, 6, 11, 14, 20,28

Kriptogram : (1 ´ 1) + (1 ´ 6) + (1 ´ 20) = 27

Blok plainteks ke-2 : 1101001

Knapsack : 1, 5, 6, 11, 14, 20,28

Kriptogram : (1 ´ 1) + (1 ´ 5) + (1 ´ 11) + (1 ’ 28) = 45

Blok plainteks ke-3 : 1100110

Knapsack : 1, 5, 6, 11, 14, 20,28

Kriptogram : (1 ’ 1) + (1 ‘ 5) + (1 ‘ 14) + (1 ‘ 20) = 40

Blok plainteks ke-4 : 1100001

Knapsack : 1, 5, 6, 11, 14, 20, 28

Kriptogram : (1 ´ 1) + (1 ´ 5) + (1 ‘ 28) = 34

Blok plainteks ke-5 : 1101001

Knapsack : 1, 5, 6, 11, 14, 20, 28

Kriptogram : (1 ´ 1) + (1 ´ 5) + (1 ´ 11) + (1’28) = 45

Jadi, cipherteks yang dihasilkan: 27 45 40 34 45

b. Metode *Run Length Encoding*

Adalah suatu proses kompresi dengan memindahkan pengulangan byte yang sama berturut-turut atau secara terus menerus.

Contoh byte : 27,45,40,34,45

Untuk mempersingkat penulisan dari sebuah code ,maka dengan melakukan kompresi hasil dari contoh diatas setelah dikompresi menjadi (27,1) (45,2) (40,1) (34,1).

c. *Superincreasing Knapsack* (*Private Key***)**

Jika senarai bobot disebut barisan *superincreasing*, maka kita dapat membentuk *superincreasing knapsack*. Barisan *superincreasing* adalah suatu barisan di mana setiap nilai di dalam barisan lebih besar daripada jumlah semua nilai sebelumnya. Misalnya {1, 3, 7, 15, 31, 63} adalah barisan *superincreasing.*

Contoh : { 1, 3, 7, 15, 31, 63 }

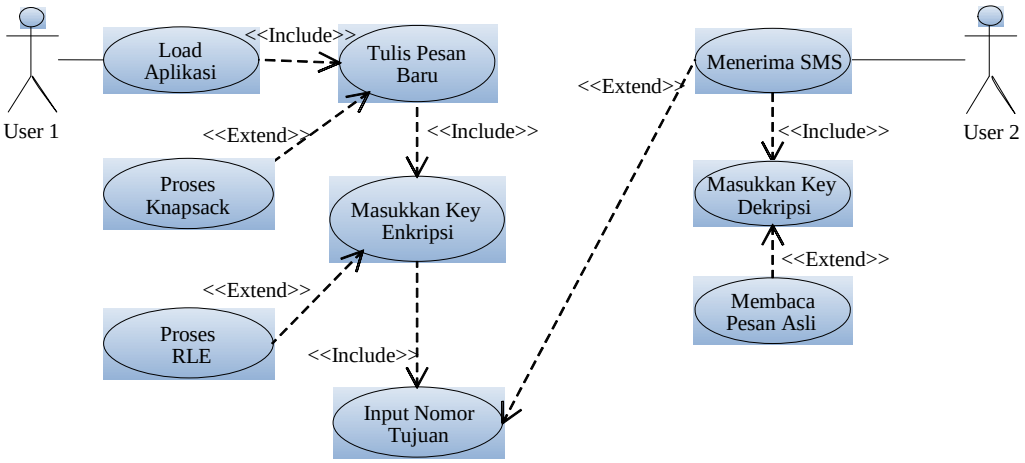
Diuraikan menjadi (1+1) = 2+1 = 3, (3+3)= 6+1 = 7, (7+7) = 14+1 = 15, (15+15) = 30+1 = 31, (31+31) = 62+1 = 63.

III.3. Desain Sistem

Pada proses perancangan ini akan dijelaskan mengenai beberapa rancangan aplikasi yang akan dikerjakan yang menggunakan perangkat *android* yaitu sebagai berikut:

III.3.1. *Use Case* Diagram

Use case diagram berfungsi untuk menggambarkan kegiatan aktor atau pengguna aplikasi, adapun *use case* diagram aplikasi yang dirancang dapat dilihat pada gambar III.1 berikut.



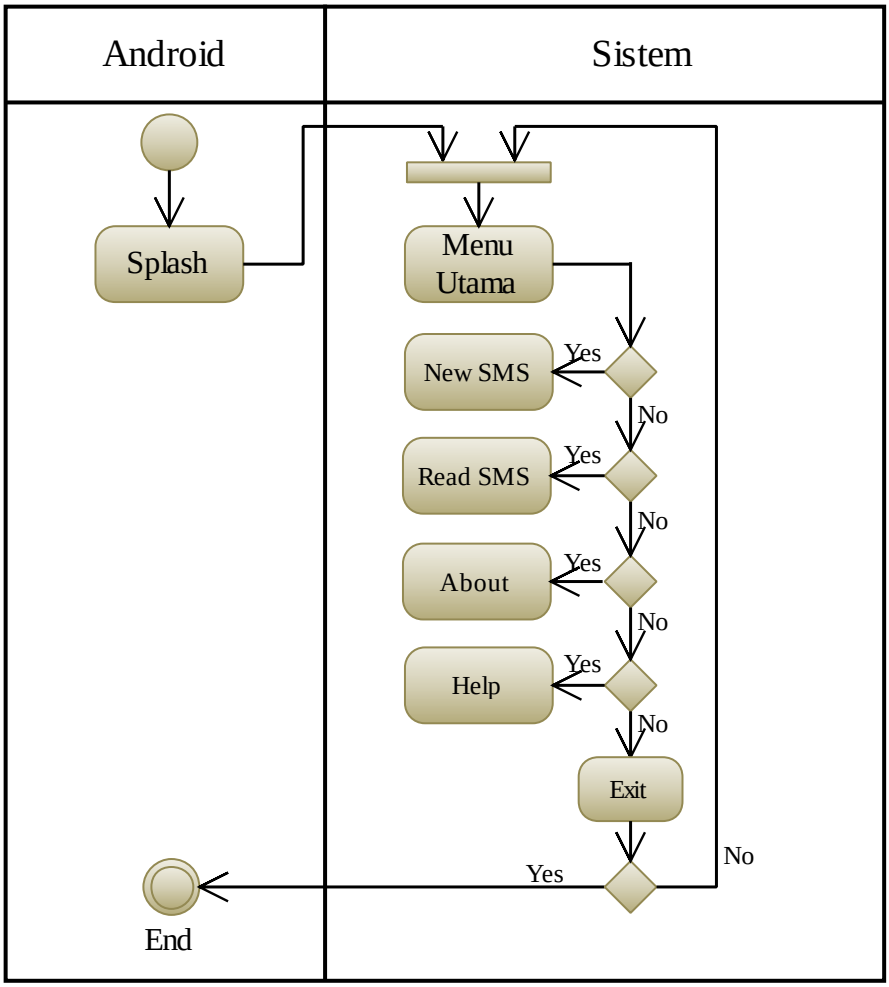
Gambar III.1. *Use Case* Diagram

Dari gambar *usecase* diagram diatas, pengguna memulai menjalankan aplikasi dan memilih menu SMS dan melakukan penulisan pesan, pada saat penulisan pesan hal berikutnya dilakukan proses *knapcask* sehingga ukuran teks akan semakin besar, lalu pengguna melakukan enkripsi dengan menginputkan *key* dan algoritma *Run Length Encoding* akan bekerja pada saat itu sehingga ukuran *file* teks akan mengalami penurunan dan melakukan pengiriman dengan memasukkan nomor tujuan target pengiriman. Pengguna atau penerima akan menerima pesan masuk dan melakukan dekripsi dengan menginputkan *key*. Proses dekripsi ini dilakukan untuk membaca pesan asli atau SMS yang masuk.

III.3.2. *Activity* Diagram

Pada *activity* diagram dibawah ini menggambarkan proses yang berjalan pada aplikasi *android* terdapat beberapa menu yang ditampilkan.Proses yang

berlangsung terjadi setelah pengguna menjalankan aplikasi, yang dapat dilihat pada gambar III.2 berikut.

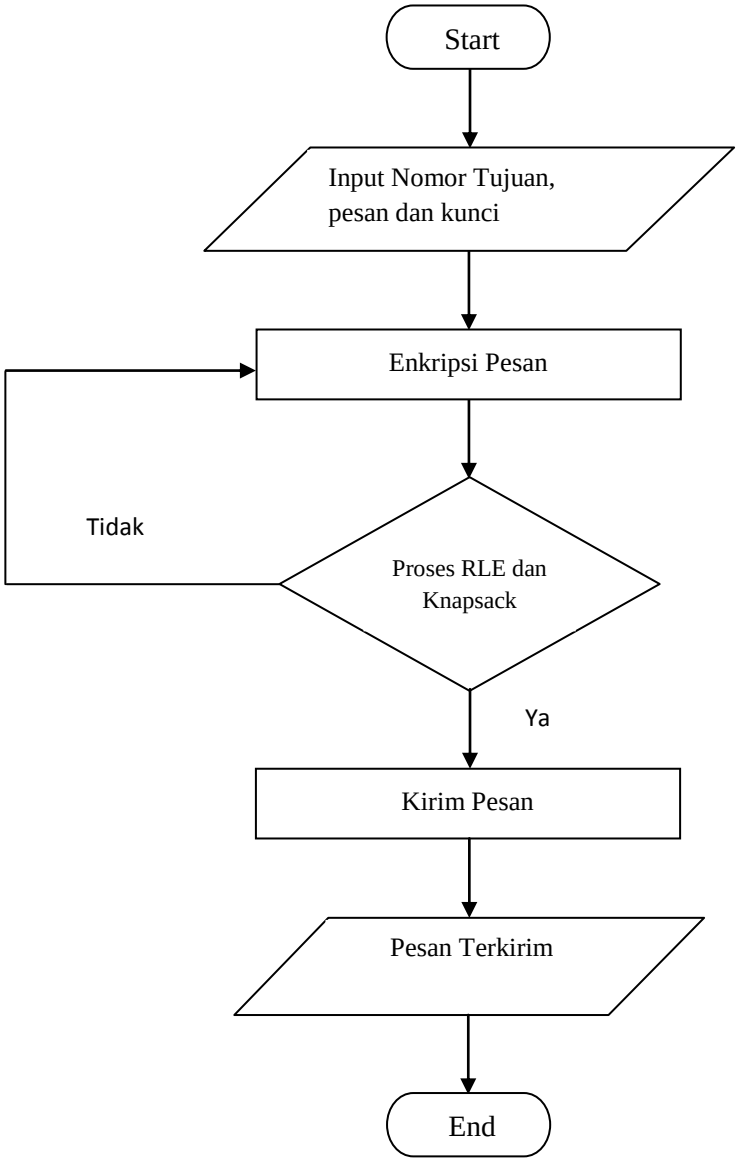


Gambar III.2. *Activity* Diagram *Android*

Dari gambar *Activty* diagram diatas, proses aplikasi merupakan tahapan yang disajikan terhadap cara kerja aplikasi keamanan sms ketika digunakan oleh pengguna.

III.3.3. *Flowchart*

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah.



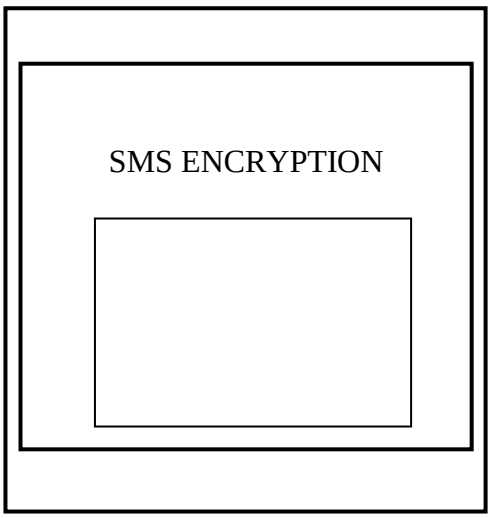
Gambar III.3 *Flowchart* Aplikasi SMS Enkripsi

III.4. Desain *Interface*

Pada rancangan aplikasi *android* terdiri dari bebarapa tampilan dan menu yang dapat digunakan, rancangan tampilan yang ada pada aplikasi *android* adalah sebagai berikut.

1. Rancangan *Splash*

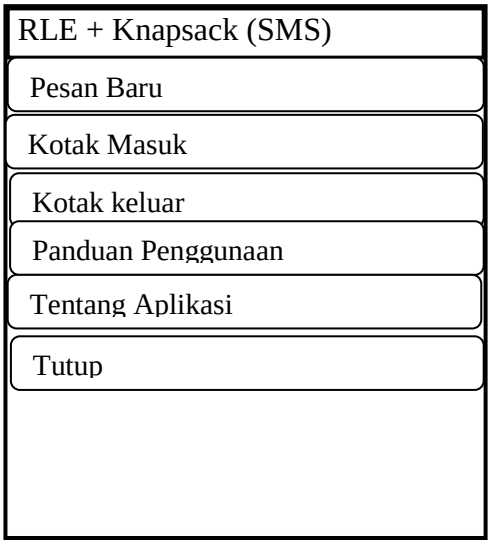
Rancangan layar *splash* merupakan rancangan awal pembuka aplikasi, Yang dapat dilihat pada gambar III.5.



Gambar III.3. Rancangan *Form Splash*

2. Rancangan Menu Utama

Rancangan menu adalah menu yang ada setelah pengguna masuk ke aplikasi. Yang dapat dilihat pada gambar III.6. berikut.



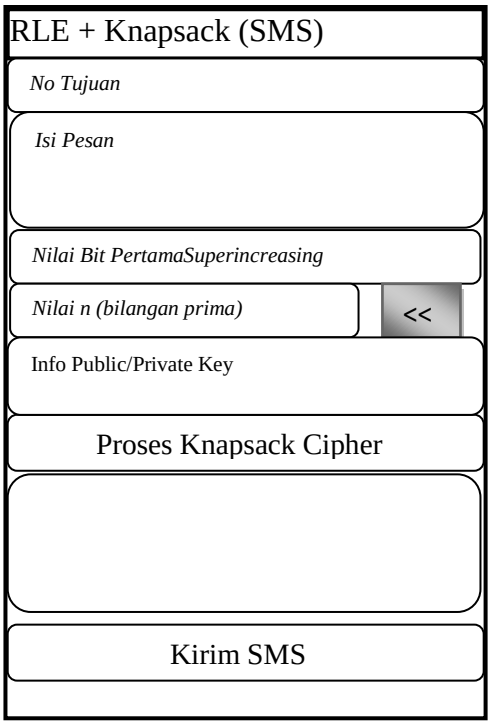
Gambar III.4. Rancangan Menu Utama

Pada gambar diatas terdapat beberapa menu yang dapat dijelaskan antara lain sebagai berikut :

a. *Menu Enkripsi*, merupakan menu rancangan untuk membuat sms baru yaitu dengan mengetikkan text / kata-kata melalui menu *new sms*.
b. *Inbox/Outbox*, merupakan menu yang digunakan untuk membuka pesan masuk dan membaca pesan masuk serta melihat pesan yang terlah terkirim.
c. Tentang Aplikasi, yang merupakan menu untuk menyajikan informasi mengenai *developer* pembuat program.
d. Panduan Penggunaan, merupakan menu untuk menyajikan inforamsi tetang cara penggunaan aplikasi.
e. Tutup, untuk pengguna menutup aplikasi.

3. Rancangan *Menu Enkripsi*

Form ini berfungsi untuk tampilan untuk mengenkripsi pesan. Yang dapat dilihat pada gambar III.7.berikut.



Gambar III.5. Rancangan *Menu Enkripsi*

4. Rancangan *Menu Dekripsi*

Form ini berfungsi untuk mendekripsikan hasil pesan yang telah dienkripsi menjadi pesan asli , di aplikasi keamanan sms. Yang dapat dilihat pada gambar III.8. berikut.

Gambar III.8. Rancangan Menu Dekripsi

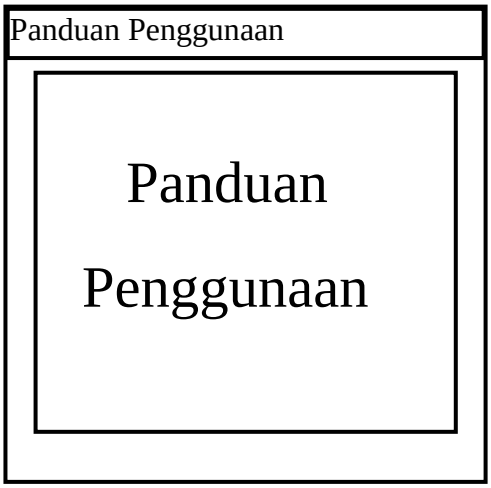
5. Rancangan *Form* Tentang Aplikasi

Form ini menampilkan informasi tentang penggunaan aplikasi *keamanan sms*, yang dapat dilihat pada gambar III.9. berikut.

Gambar III.9. Rancangan *Form* Tentang Aplikasi

6. Rancangan *Form* Panduan Penggunaan

Form ini berfungsi untuk menampilkan informasi tentang cara menggunakan aplikasi keamanan sms ini, dapat dilihat pada gambar III.10.berikut.



Gambar III.10. Rancangan *Form* Panduan Penggunaan