BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Pada saat ini, penggunaan perangkat mobile sudah menjadi trend di masyarakat dunia. Dalam suatu media penyimpanan, terdapat suatu data penting atau rahasia yang tidak semua orang boleh mengetahuinya. Data-data penting yang hanya boleh diketahui oleh pemiliknya saja antara lain dokumen-dokumen pribadi, akun email, akun jejaring sosial, akun kartu kredit, akun internet banking. Apalagi saat proses pengiriman file melalui media internet maupun saat perangkat mobile itu hilang, membuat pemiliknya sangat riskan kehilangan data-data pentingnya.

Oleh karena itu, dalam penelitian ini akan coba dibuat sebuah aplikasi pengamanan data berupa dokumen dengan menggunakan metode algoritma *Stream Cipher* untuk mengenkripsi data yang berjalan pada sistem operasi Android sehingga pemilik merasa aman untuk menyimpan datanya.

Adapun resiko permasalahan yang muncul dari hal diatas adalah :

1. Banyaknya berkas *file* penting yang belum ada pengamananya.
2. Masih sedikit aplikasi pengamanan berbasis mobile dengan metode *Stream Cipher*.
3. Keamanan data *file* yang beresiko mengalami tindak pencurian, penyadapan, dan manipulasi data yang akan merugikan pihak tertentu dikemudian hari.

III.1.1. Evaluasi

Berdasarkan analisa diatas penulis memberikan suatu solusi atau strategi pemecahan masalah yang diharapkan dapat mengatasi masalah ada. Adapun pemecahan masalah yang diusulkan oleh penulis untuk mengantisipasi permasalahan diatas adalah merancang aplikasi enkripsi dan dekripsi *file* berbasis *android* dengan keamanan kriptografi *Stream Cipher*. Aplikasi enkripsi dan dekripsi *file* dengan menggunakan metode algoritma *Stream Cipher* berbasis *android* ini adalah salah satu sistem yang diyakini mampu memberikan konstribusi positif bagi penggunanya.

III.2. Penerapan Metode Kriptografi *Stream Cipher*

Stream Cipher adalah algoritma kriptografi yang mengenkripsi *Plaintext* menjadi *Ciphertext* bit per bit (B*yte* per *Byte*). (Arham A., 2014).

III.2.1. Rumus Kriptografi *Stream Cipher*

Adapun cara enkripsi dan dekripsi pada kriptografi *Stream Cipher* adalah :

Proses Enkripsi :

1. Tentukan *Plaintext* yang akan dienkripsi.
2. Ubah karakter kedalam bentuk bilangan biner dengan menggunakan bantuan tabel bilangan biner.
3. Buatlah suatu kunci (*Keystream*) untuk mengenkripsi *Plaintext,* jumlah karakter *Keystream* harus sama dengan jumlah karakter *Plaintext*.

4. Enkripsi *Plaintext* dengan menggabungkan *Keystream* dan *Plaintext* dengan operasi *Bitwise Exclusive*-OR (XOR) pada tiap bit bilangan biner.
5. Ubah hasil perhitungan *Ciphertext* kedalam bentuk hexadesimal dengan menggunakan bantuan tabel bilangan biner.

Proses Dekripsi :

1. Ubah bilangan hexadesimal kedalam bentuk bilangan biner dengan menggunakan bantuan tabel bilangan biner.
2. Masukkan suatu kunci (*Keystream*) dari proses enkripsi *Plaintext.*
3. Dekripsi *Ciphertext* dengan menggabungkan *Keystream* dan *Ciphertext* dengan operasi *Bitwise Exclusive*-OR (XOR) pada tiap bit bilangan biner.
4. Ubah hasil perhitugan *Ciphertext* kedalam bentuk karakter dengan menggunakan bantuan tabel bilangan biner.

Rumus perhitungan operasi XOR dapat dilakukan dengan 2 cara, yaitu dengan perhitungan langsung menggunakan operasi XOR dan menggunakan penjumlahan modulus 2 dan tetap akan menghasilkan jumlah yang sama.

Adapun uraian rumus dari kedua penjumlahan XOR adalah sebagai berikut :

Rumus Operasi XOR :

$\mathbf{0 \oplus 0 = 0}$ .....(3)

$\mathbf{1 \oplus 0 = 1}$ .....(4)

$\mathbf{0 \oplus 1 = 1}$ .....(5)

$\mathbf{1 \oplus 1 = 0}$ .....(6)

Rumus Operasi Modulus 2 :

(0 + 0) mod 2 = 0 .....(7)

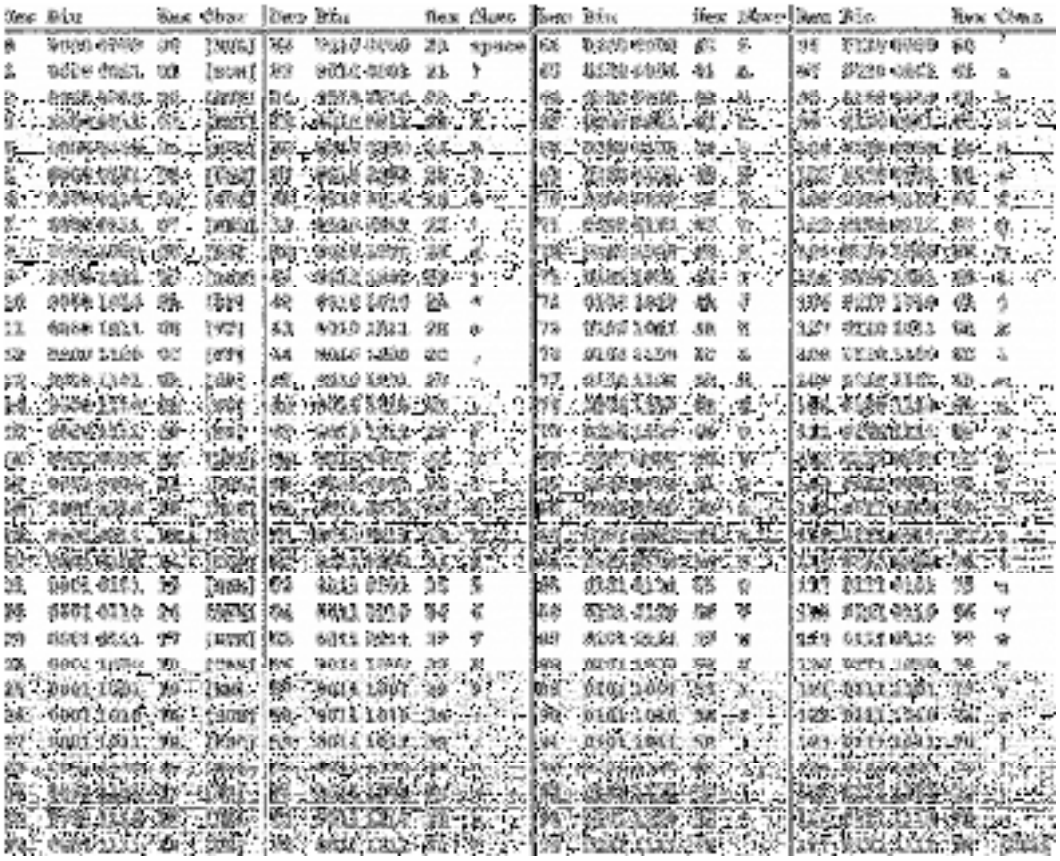
(1 + 0) mod 2 = 1 .....(8)

(0 + 1) mod 2 = 1 .....(9)

(1 + 1) mod 2 = 0 .....(10)

Barisan bit-bit kunci dihasilkan dari sebuah pembangkit yang dinamakan pembangkit arus-kunci (K*eystream Generator*). Semakin acak *Keystream* semakin kuat keamanan dari *Stream Cipher*. *Keystream* di-XOR-kan dengan *Ciphertext* untuk menghasilkan *Plaintext* semula dengan persamaan diatas.

Untuk mempermudah mengkonversi karakter menjadi biner dan biner menjadi desimal, maka dibuatlah tabel bilangan biner pada tabel III.1 sebagai berikut :



Tabel III.1. Tabel Bilangan Biner

III.2.2. Contoh Studi Kasus

Dalam perancangan aplikasi ini, Penulis memberikan uraian perhitungan pengenkripsian dan pendekripsian beberapa *File* berkas. Contoh perhitungan algoritma kriptografi *Stream Cipher* dengan ketentuan dokumen.ekstensi yaitu sebagai berikut :

1. Proses enkripsi dan dekripsi *File* berekstensi .docx

   Plaintext (Pi) = Test1.docx.

   Keystream (Ki) = R (Ulangi R sebanyak jumlah karakter *Plaintext* yaitu 10 kali).

   Ciphertext (Ci) = ?

   Jawab :

   Rumus Enkripsi : $Ci = Pi \oplus Ki$

   Pi = 01010100 01100101 01110011 01110100 00110001 00101110 01100100 01101111 01100011 01111000 (Test1.docx)

   Ki = 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 (RRRRRRRRRR)

   Ci = 00000110 00110111 00100001 00100110 01100011 01111100 00110110 00111101 00110001 00101010 (06372126637C363D312A)

Rumus Dekripsi : Pi = Ci ⊕ Ki

Ci = 00000110 00110111 00100001 00100110 01100011 01111100 00110110 00111101 00110001 00101010 (06372126637C363D312A)

Ki = 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 01010010 (RRRRRRRRRR)

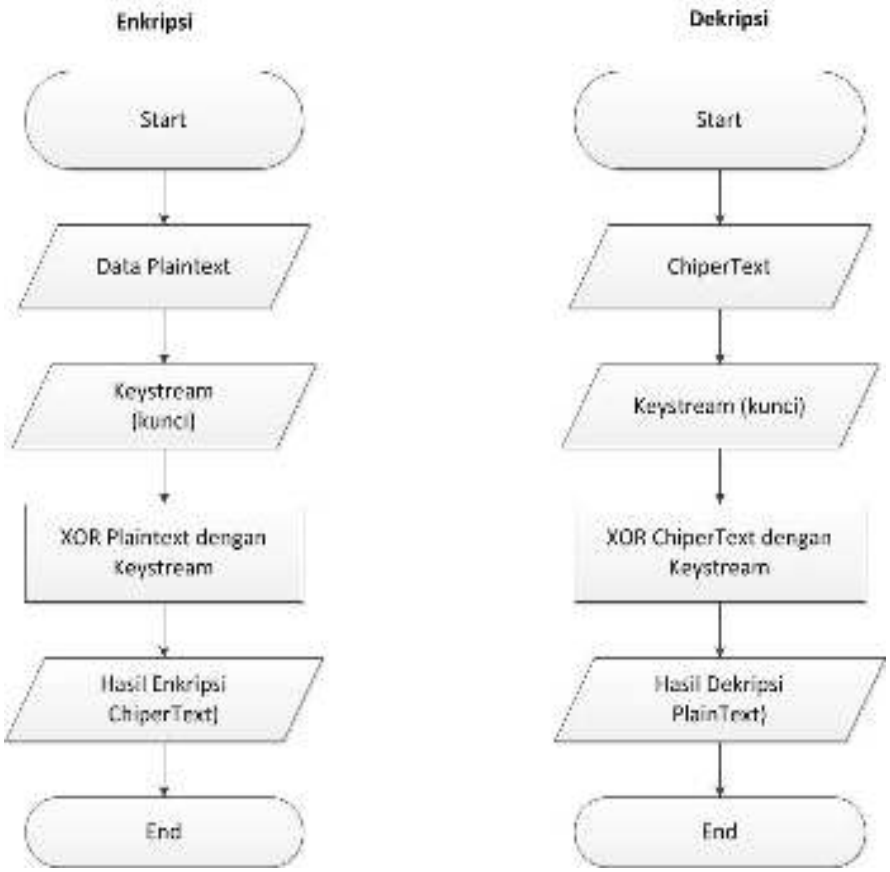
Pi = 01010100 01100101 01110011 01110100 00110001 00101110 01100100 01101111 01100011 01111000 (Test1.docx)

III.3. Desain Sistem

Untuk Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Use Case* Diagram, *Activity* Diagram, *Class* Diagram, dan *Sequence* Diagram.

III.3.1. *FlowChart Algoritma Stream Cipher*

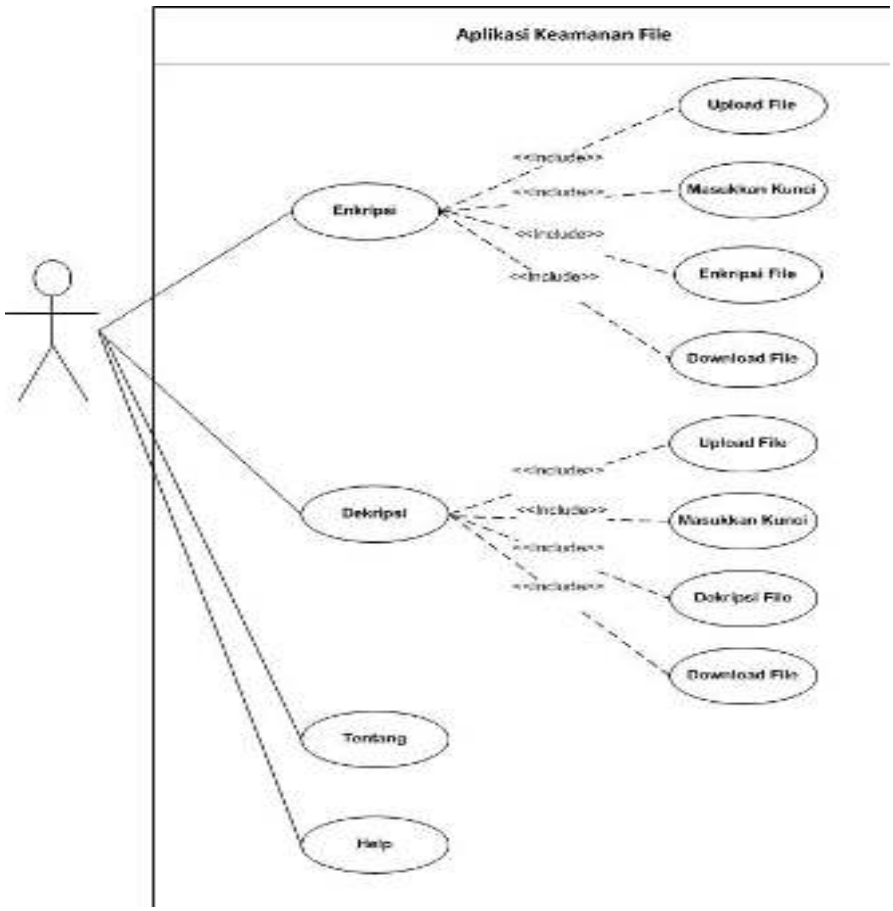
Flowchart suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program.Dapat dilihat proses yang menggambarkan algoritma *stream cipher* pada saat enkripsi dan dekripsi pada Gambar III.1 sebagai berikut .



Gambar III.1. ***FlowChart Algoritma Stream Cipher***

III.3.2. *Use Case Diagram*

Dalam *Use Case Diagram* merupakan model diagram UML (*Unified Modelling Language*) yang digunakan untuk menggambarkan *Requirement* fungsional yang diharapkan dari sebuah sistem, proses sistem yang akan dirancang digambarkan terdapat pada Gambar III.2 sebagai berikut.



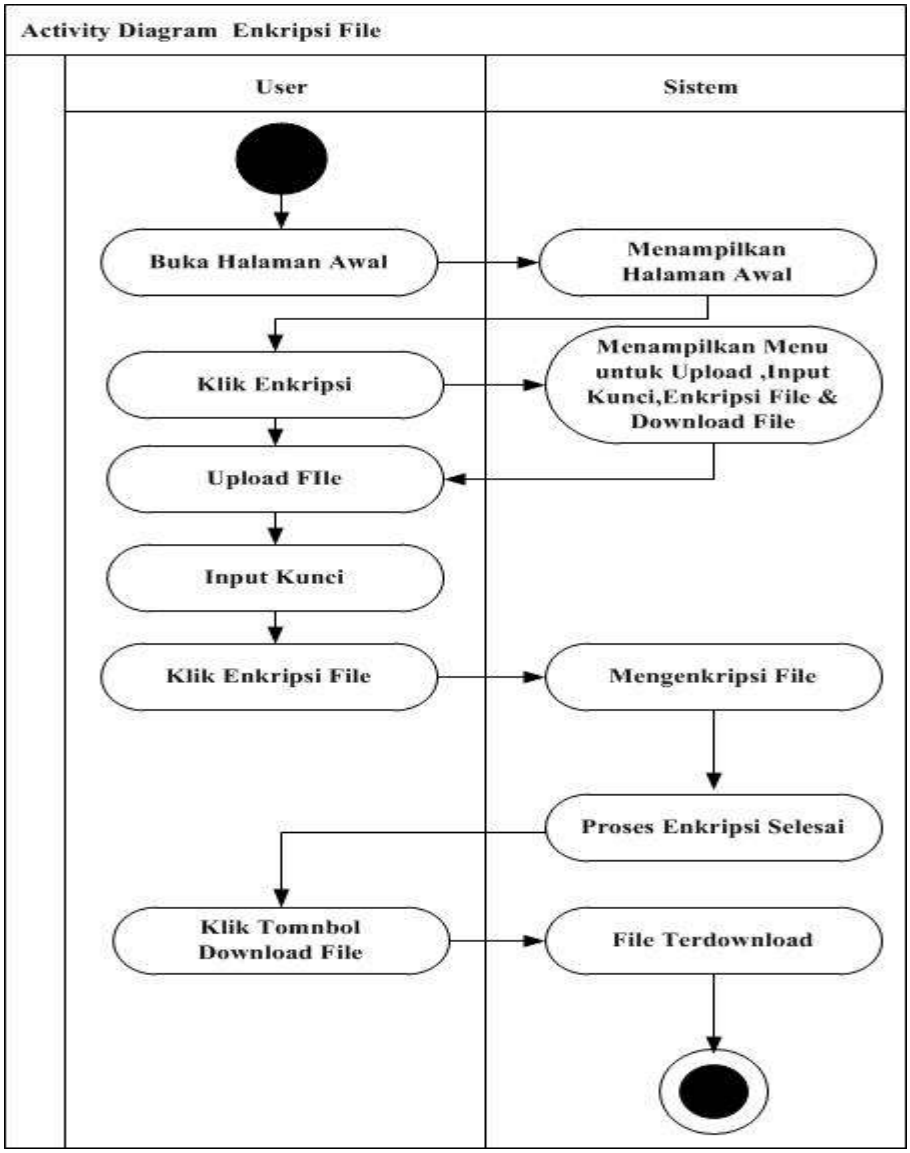
Gambar III.2. ***Use Case Diagram***

III.3.3. *Activity Diagram*

Proses yang telah digambarkan pada *Use Case Diagram* diatas dijabarkan dengan *Activity Diagram.* Adapun beberapa rangkaian *Activity Diagram* adalah sebagai berikut:

1. *Activity Diagram* Enkripsi *File*

Pada *Activity Diagram* ini, *User* menekan tombol enkripsi dan *Upload File* yang ingin dienkripsi. Adapun aktivitas yang dilakukan dapat dilihat pada Gambar III.3 sebagai berikut.



Gambar III.3. *Activity Diagram* Enkripsi

Keterangan :

a. *User* mengakses halaman awal, lalu sistem akan menampilkan halaman awal program

b. *User* klik tombol Enkripsi, sistem akan menampilkan halaman *Upload File*, *Input* kunci, *Enkripsi File* dan *Download File*

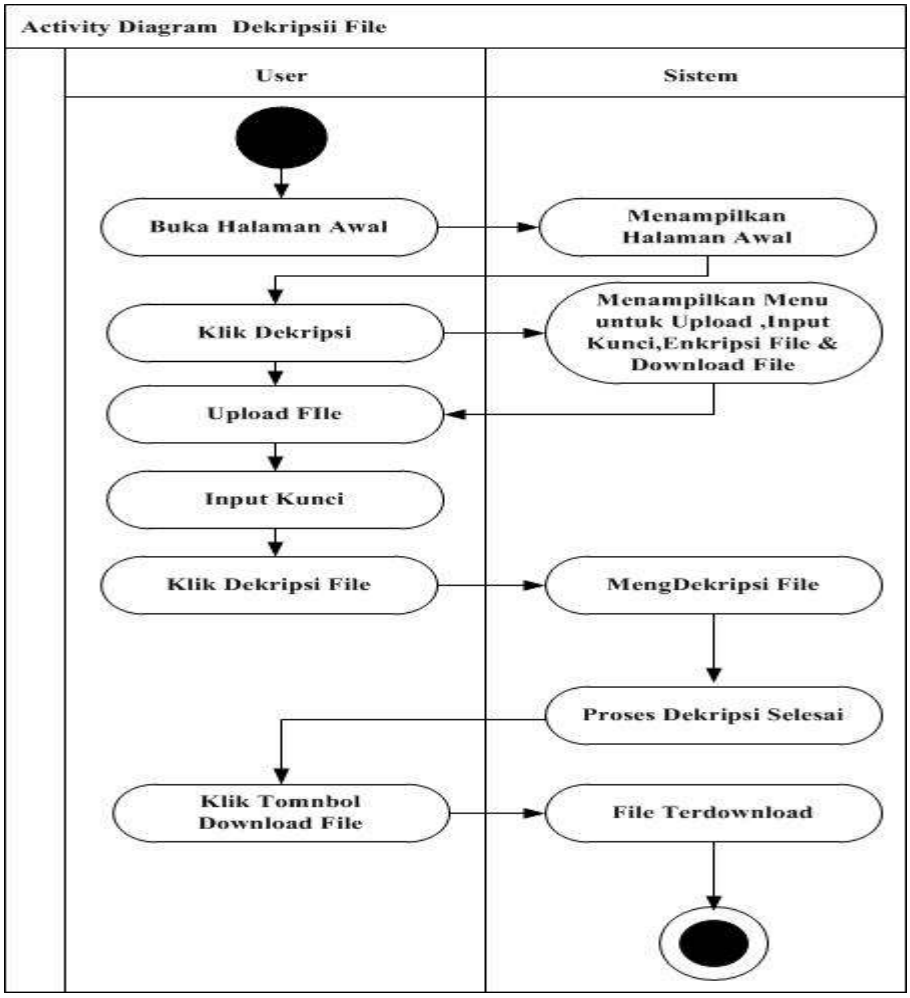
c. *User* meng-*Upload File* dan mengisi kolom kunci setelah itu klik *enkripsi file*, sistem akan mengenkripsi *File* tersebut dan proses enkripsi selesai.

d. *User* mengklik tombol *download file*, sistem akan mendownload *file* dan disimpan di perangkat..

2. *Activity Diagram* Dekripsi *File*

Pada *Activity Diagram* ini, *User* menekan tombol Dekripsi dan *Upload File* yang ingin didekripsi. Adapun aktivitas yang dilakukan dapat dilihat pada Gambar III.4 sebagai berikut.

.



Gambar III.4. *Activity Diagram* Dekripsi

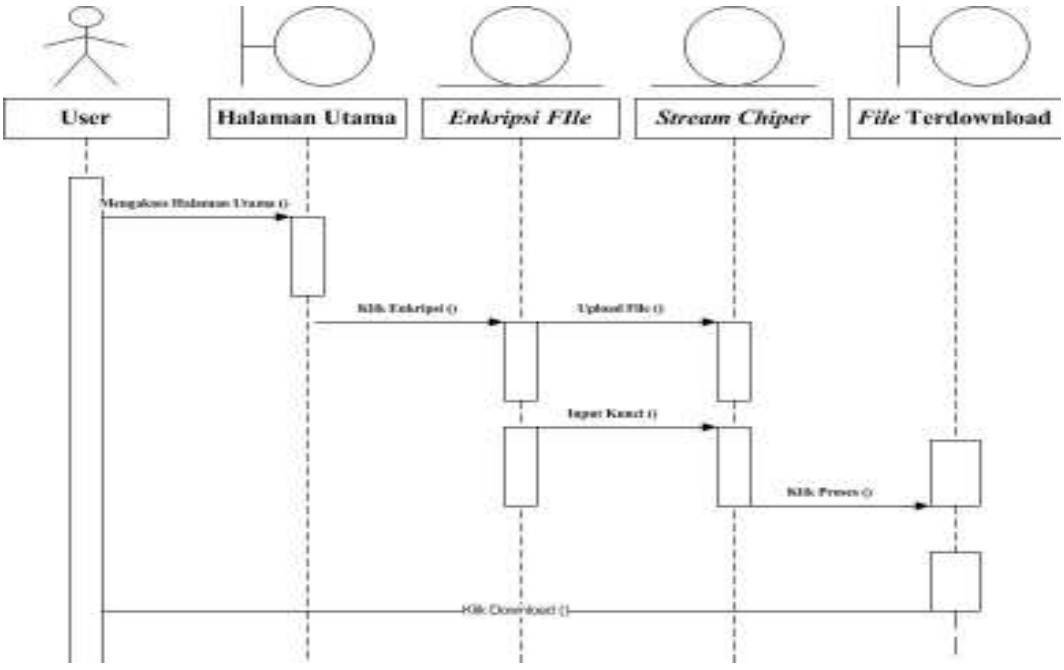
Keterangan :

a. *User* mengakses halaman awal, lalu sistem akan menampilkan halaman awal program
b. *User* klik tombol Dekripsi, sistem akan menampilkan halaman *Upload File*,*Input* kunci dan *Download File*.
c. *User* meng-*Upload File* dan mengisi kolom kunci setelah itu klik *Dekripsi File*, sistem akan mengDekripsi *File* tersebut dan proses dekripsi selesai..
d. *User* mengklik tombol *Download File*, otomatis *file* hasil dekripsi tersimpan di perangkat.

III.3.4. *Sequence Diagram*

1. *Sequence Diagram Enkripsi File*

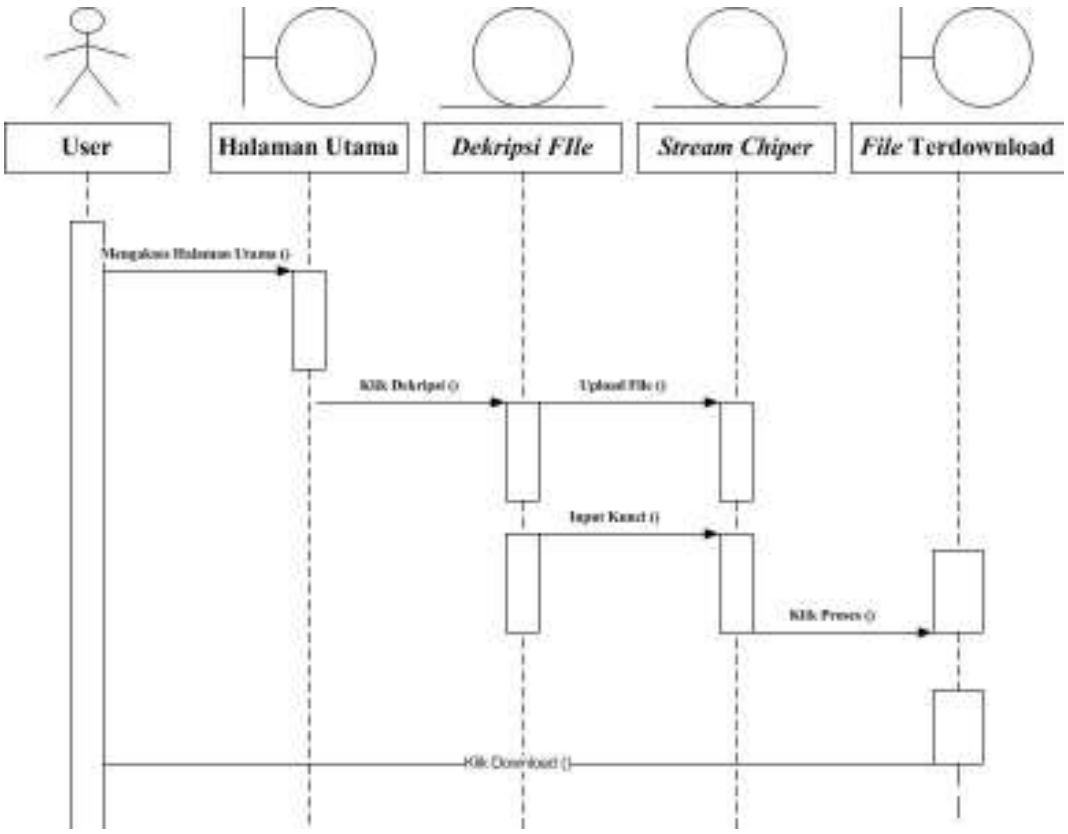
Serangkaian kerja proses enkripsi *file* terlihat seperti pada gambar III.5 berikut.



Gambar III.5. *Sequence Diagram Enkripsi File*

2. *Sequence Diagram* Dekripsi *File*

Serangkaian kerja proses dekripsi *file* terlihat seperti pada gambar III.6 berikut.



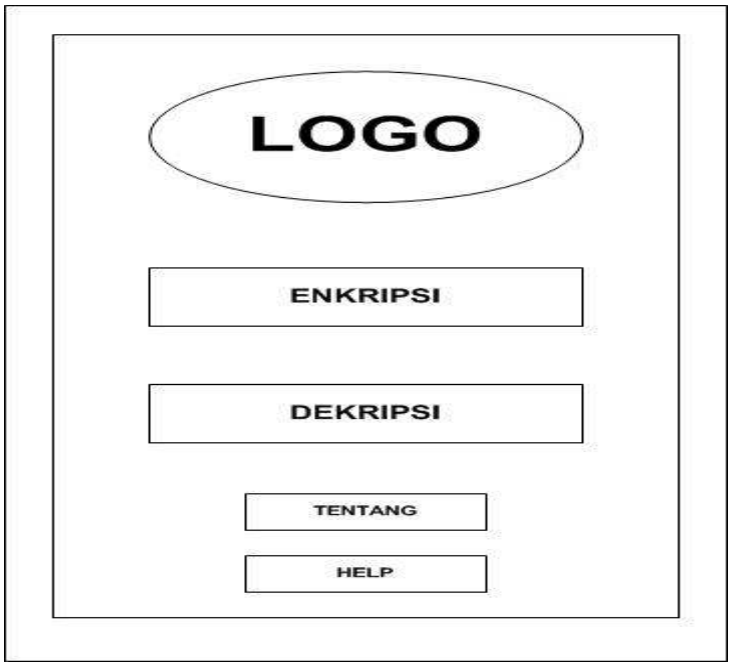
Gambar III.6. *Sequence Diagram Dekripsi File*

III.4. Desain *User Interface*

Tahap perancangan berikutnya yaitu desain sistem secara detail yang meliputi desain Halaman *Home* , Halaman *Enkripsi File* Dan Halaman *Dekripsi File*.

III.4.1. Desain Halaman *Home*

Desain Halaman *Home* adalah halaman yang pertama kali ditampilkan saat aplikasi dijalankan dapat dilihat pada gambar III.7 sebagai berikut.



Gambar III.7. Desain *Form HOME*

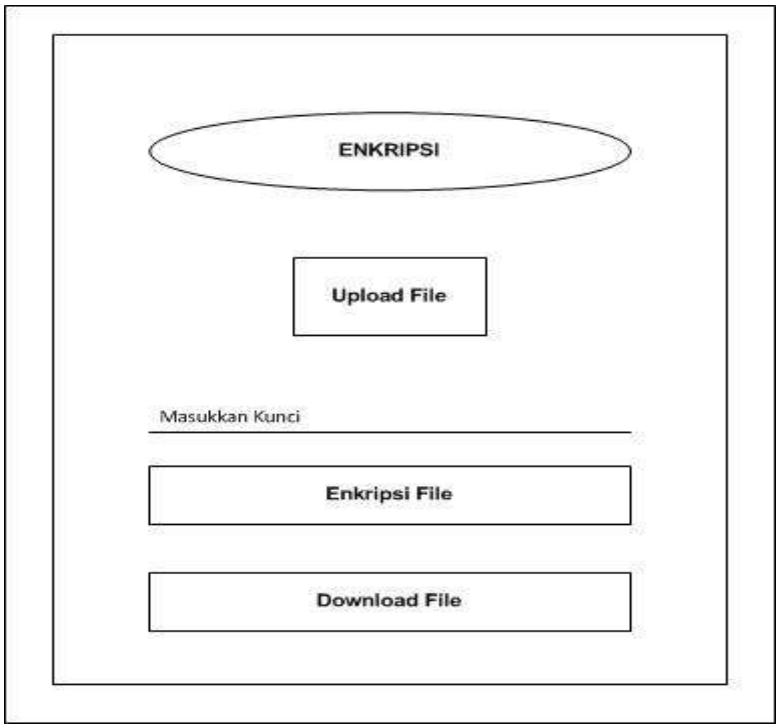
Keterangan:

a. Tombol *Enkripsi* adalah tombol untuk masuk kehalaman *Enkripsi* setelah user mengklik user akan ditampilkan ke halaman *enkripis untuk upload file*,input kunci dan *Download File*
b. Tombol *Dekripsi* adalah tombol untuk masuk kehalaman *Dekripsi* setelah user mengklik user akan ditampilkan ke halaman *dekripsi* untuk *upload file*,input kunci dan *Download File*
c. Tombol tentang adalah tombol informasi yang menyangkut aplikasi dan prmbuat aplikasi tersebut.

d. Tombol Help adalah tombol bantuan tata cara pemakian aplikasi yang benar dan tepat.

III.4.2. Desain Halaman *Enkripisi File*

Desain Halaman *Enkripisi File* berfungsi untuk *Upload Fil*e dan input kunci serta memproses *file* agar di enkripsi kemudian hasilnya dapat didownload halaman ini dapat dilihat pada gambar III.8 sebagai berikut.



Gambar III.8. Desain Halaman *Enkripsi File*

Keterangan:

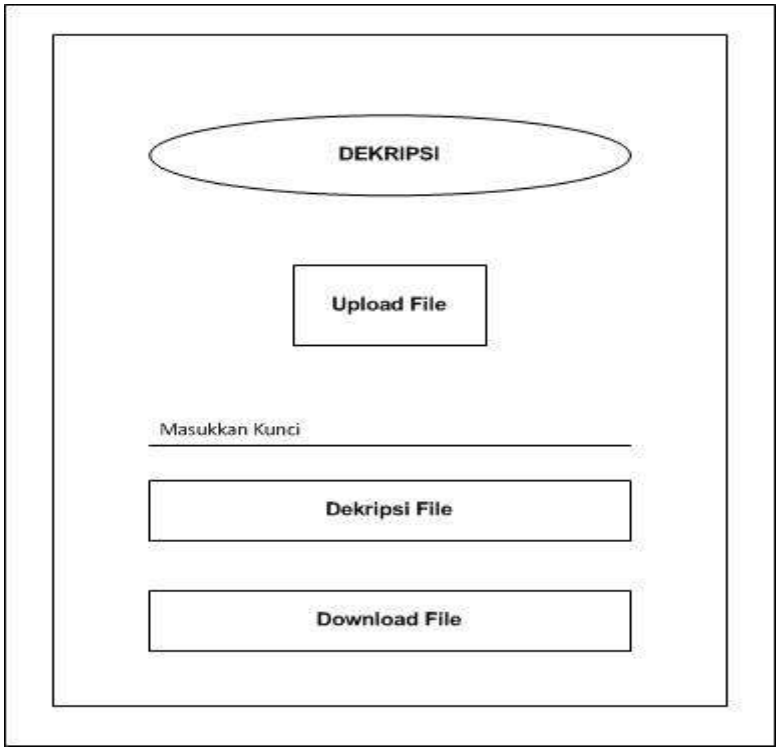
a. Tombol *Upload File* berfungsi untuk menupload file yang ingin kita enkripsi.

b. Setelah *Upload File* yang akan di *enkripsi* kita mengisi kolom kunci .

c. Tombol *Enkripsi File* untuk memproses *file* atau memulai *enkripsi* pada *file* yang telah kita *upload* tadi setelah proses enkripsi selesai.
d. Tombol *Download File* bisa di klik fungsi tombol *download File* adalah untuk mengunduh file yang sudah kita enkripsi tadi.

III.4.3. Desain Halaman *Dekripisi File*

Desain Halaman *Dekripisi File* berfungsi untuk *Upload Fil*e dan input kunci serta memproses *file* agar di dekripsi kemudian hasilnya dapat didownload halaman ini dapat dilihat pada gambar III.9 sebagai berikut.



Gambar III.9. Desain Halaman *Dekripsi File*

Keterangan:

a. Tombol *Upload File* berfungsi untuk menupload file yang ingin kita dekripsi.

b. Setelah *Upload File* yang akan di *dekripsi* kita mengisi kolom .

c. Tombol *Dekripsi File* untuk memproses *file* atau memulai *dekripsi* pada *file* yang telah kita *upload* tadi setelah proses dekripsi selesai..

d. Tombol *Download File* bisa di klik fungsi tombol *download* adalah untuk mengunduh file yang sudah kita dekripsi tadi.