BAB III

ANALISIS DAN PERANCANGAN

3.1. Metode Penelitian

Metode penelitian yang akan digunakan dalam penelitian ini antara lain adalah sebagai berikut.

a. Studi Literatur

Melakukan studi mengenai teori dan konsep yang berkaitan dengan pokok bahasan penelitian, seperti teori mengenai algoritma ElGamal, teori mengenai algoritma LZW dan berbagai konsep pendukung lainnya. Referensi yang digunakan dapat berupa artikel, buku, jurnal ilmiah, dan lain-lain.

b. Analisis dan Perancangan Aplikasi

Melakukan analisis dan perancangan awal terhadap aplikasi yang akan dibangun, meliputi perancangan alur aplikasi dalam bentuk *flowchart* dan rancangan *user interface*.

c. Pembangunan Aplikasi

Melakukan pembangunan aplikasi dengan mengimplementasikan rancangan dan metode yang telah didefinisikan sebelumnya dengan menggunakan bahasa pemrograman yang telah ditentukan.

d. Uji Coba dan Evaluasi

Melakukan uji coba terhadap aplikasi disertai dengan evaluasi dari hasil yang didapatkan.

3.2. Analisis Aplikasi

Pada penelitian ini, dibangun sebuah Windows *desktop application* untuk mengamankan dokumen digital. Aplikasi dibuat dengan menggunakan Microsoft Visual Studio 2010 untuk komputer 32 bit dan bahasa pemrograman C#. Algoritma yang diimplementasikan dalam pembangunan aplikasi ini adalah algoritma kriptografi ElGamal dan algoritma kompresi LZW didasarkan kepada seluruh paparan yang telah dijelaskan sebelumnya.

Dokumen-dokumen digital yang dapat digunakan dalam aplikasi ini (*input*) hanya yang ber-ekstensi *text file* (.txt), *portable document format* (.pdf), .doc, .docx, .xls, .xlsx, .ppt, dan .pptx. Berdasarkan paparan pada bab sebelumnya (terlepas dari isi masing-masing dokumen) maka dokumen-dokumen digital dengan ekstensi yang disebutkan dapat digunakan dalam aplikasi ini. Dokumen dengan format Open XML dimungkinkan memiliki beban data yang lebih ringan dibandingkan dengan dokumen dengan format *default*.

3.3. Perancangan Aplikasi

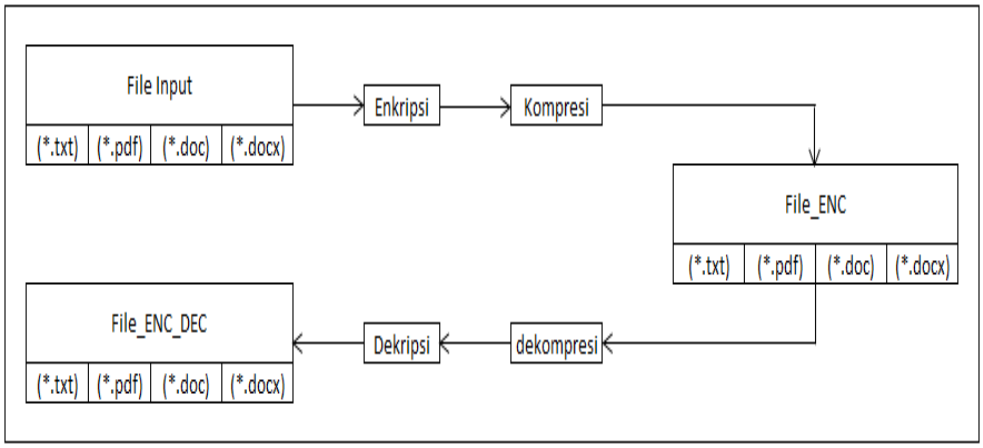
Pada subbab ini dipaparkan hasil perancangan dalam dua bagian, yaitu Pemodelan Sistem dan Perancangan Antarmuka.

3.3.1. Pemodelan Sistem

Pemodelan sistem dibagi menjadi Gambaran Umum Aplikasi dan *System Flow*.

A. Gambaran Umum Aplikasi

Aplikasi yang dibangun merupakan Windows *desktop application*. Seluruh aktivitas dikendalikan oleh *user*, karena *user* langsung terhubung dengan aplikasi tanpa ada perantara apapun. Gambaran umum aplikasi dapat digambarkan seperti pada Gambar 3.1.



Gambar 3.1. Gambaran Umum Aplikasi

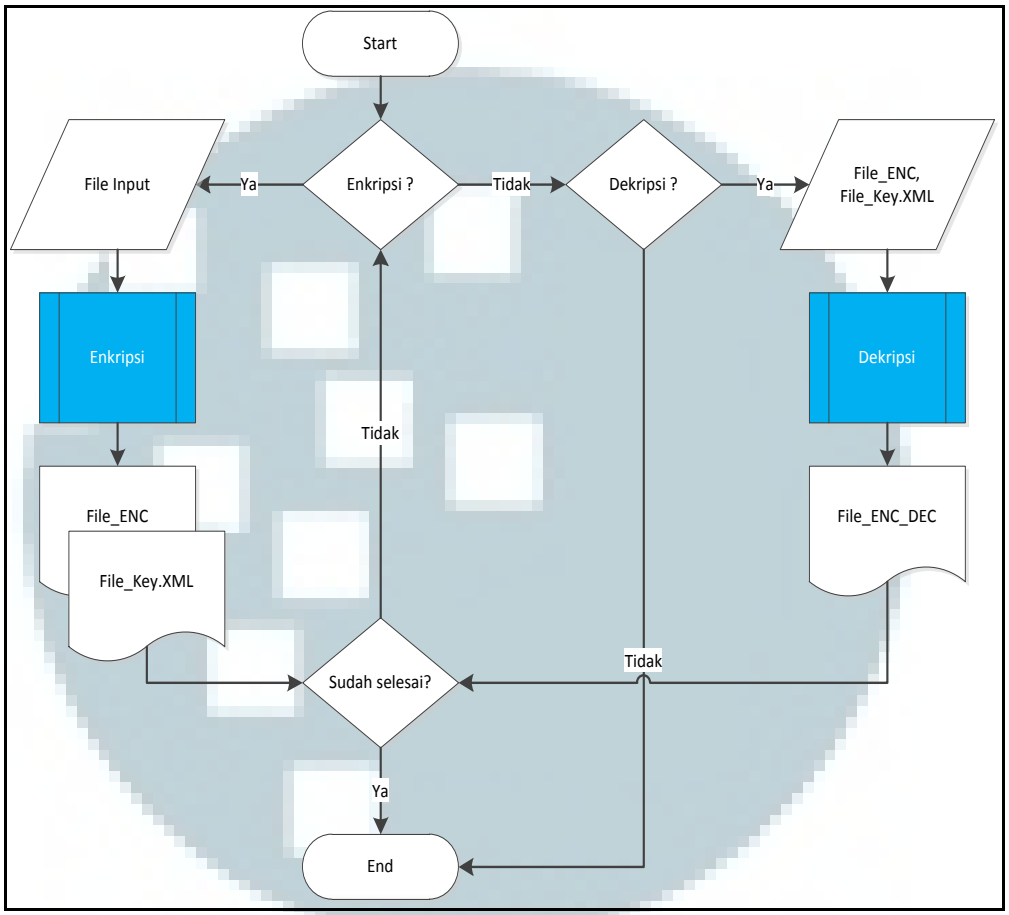
B. System Flow

Bagian ini berisi bagan yang memiliki arus dan digunakan untuk menggambarkan langkah-langkah yang dilalui oleh sistem dalam menyelesaikan suatu masalah.

B.1. System Flow Aplikasi

Aplikasi akan menampilkan dua pilihan proses, Enkripsi dan Dekripsi. Jika ingin melakukan proses enkripsi, *user* dapat memilih *file* yang ingin di enkripsi (*file*_input). Setelah enkripsi selesai dilakukan, *user* akan mendapatkan *file*_ENC dan *file*_Key.XML. Jika ingin melakukan proses dekripsi, *user* dapat memilih *file* yang sudah ter-enkripsi (*file*_ENC) dan *flie* kunci yang harus digunakan (*file*_Key.XML). Setelah dekripsi selesai dilakukan, *user* akan mendapatkan

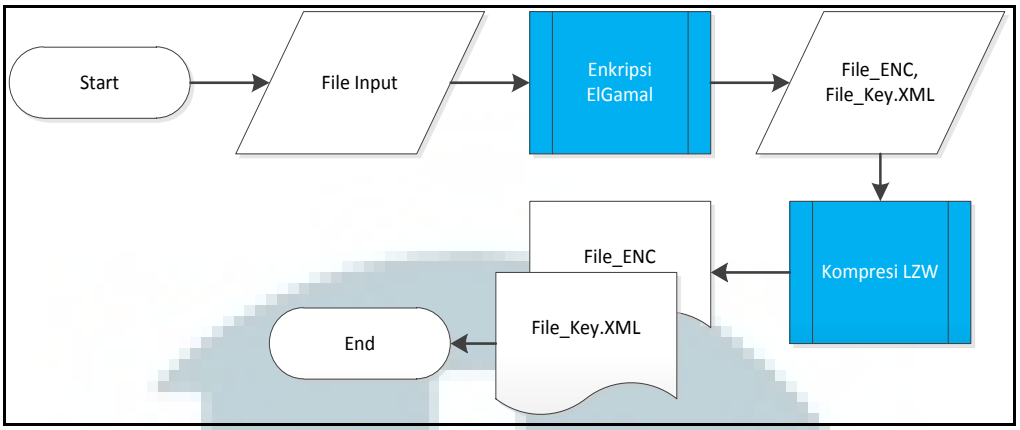
*file*_ENC_DEC. *User* dapat keluar dari aplikasi setiap saat atau mengulang kembali setiap proses. Gambar 3.2. menampilkan *system flow* aplikasi.



Gambar 3.2. *System Flow* Aplikasi

B.2. System Flow Subproses Enkripsi

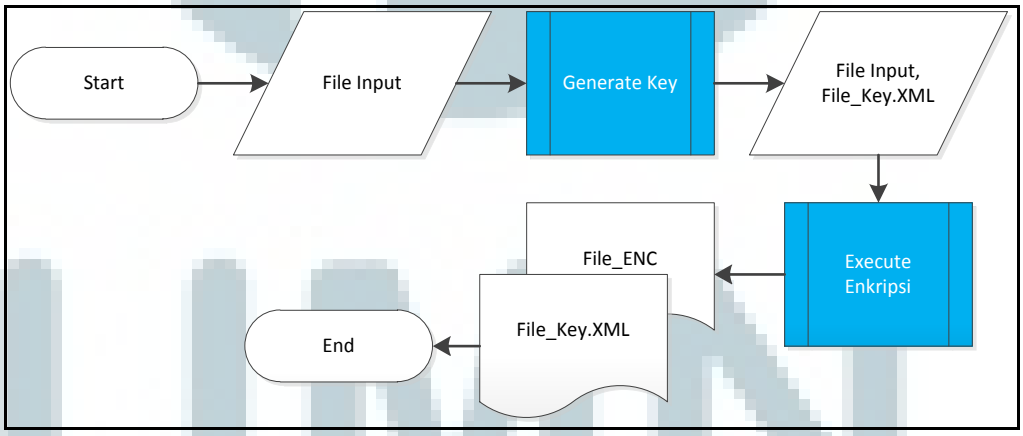
Pada subproses enkripsi, *user* harus memilih *file* yang ingin di enkripsi. Pertama-tama aplikasi akan melakukan proses enkripsi pada *file*_input sehingga menghasilkan *file*_ENC dan *file*_Key.XML. Kemudian aplikasi akan langsung melakukan kompresi. Setelah proses kompresi selesai dilakukan, *user* akan mendapatkan *file*_ENC dan *file*_Key.XML. Gambar 3.3. menampilkan *system flow* subproses enkripsi.



Gambar 3.3. *System Flow* subproses enkripsi

B.3. System Flow Subproses Enkripsi ElGamal

Pada subproses enkripsi ElGamal, hal yang pertama kali dilakukan adalah *generate key*. Pada tahap ini *file*_input belum dienkripsi. Aplikasi akan menghasilkan *file*_input dan *file*_Key.XML, kemudian akan menjalankan *execute* enkripsi langsung pada *file*_input. Hasilnya, *file*_input akan ter-enkripsi (*file*_ENC). Gambar 3.4. menampilkan *system flow* subproses enkripsi ElGamal.

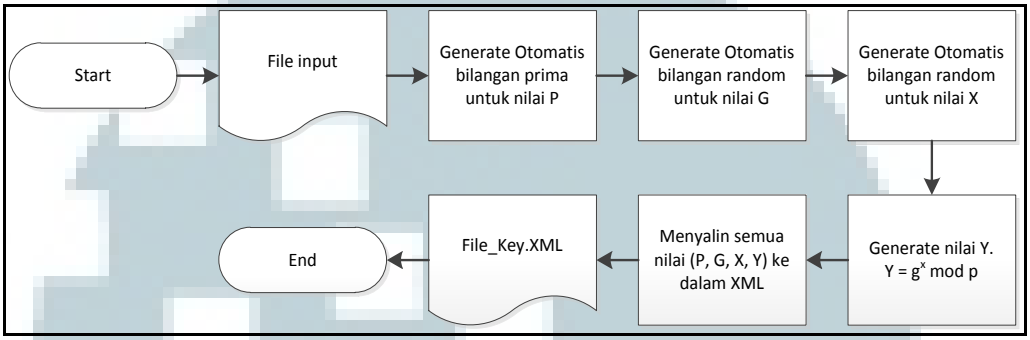


Gambar 3.4. *System Flow* subproses enkripsi ElGamal.

B.4. System Flow Subproses Generate Key

Generate key adalah subproses terpenting, karena setiap dokumen membutuhkan kunci untuk melakukan enkripsi dan dekripsi. Pertama-tama,

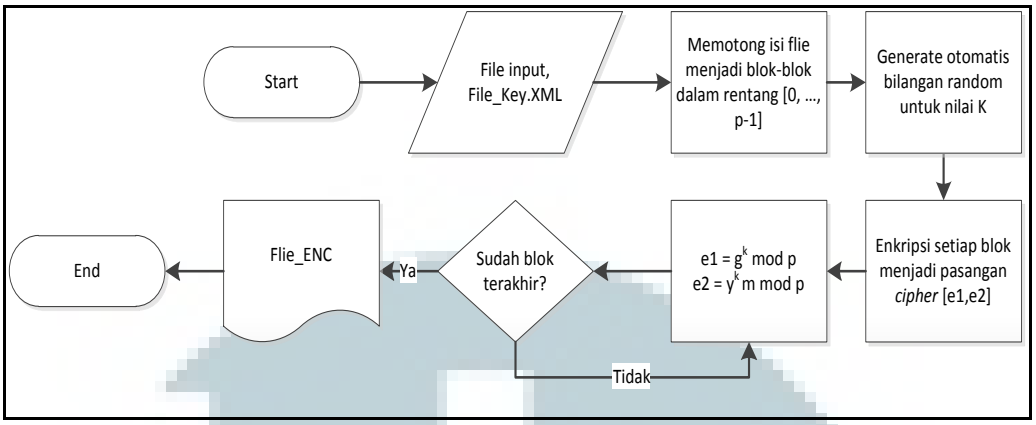
aplikasi akan men-*generate* sebuah bilangan prima P. Kemudian aplikasi kembali akan men-*generate* dua buah bilangan acak, yaitu G dan X. Kemudian aplikasi akan menghitung nilai Y dengan rumus $Y = G^X \bmod P$. Gambar 3.5. menampilkan *system flow* subproses *generate key*.



Gambar 3.5. *System Flow* subproses *generate key*

B.5. System Flow Subproses Execute Enkripsi

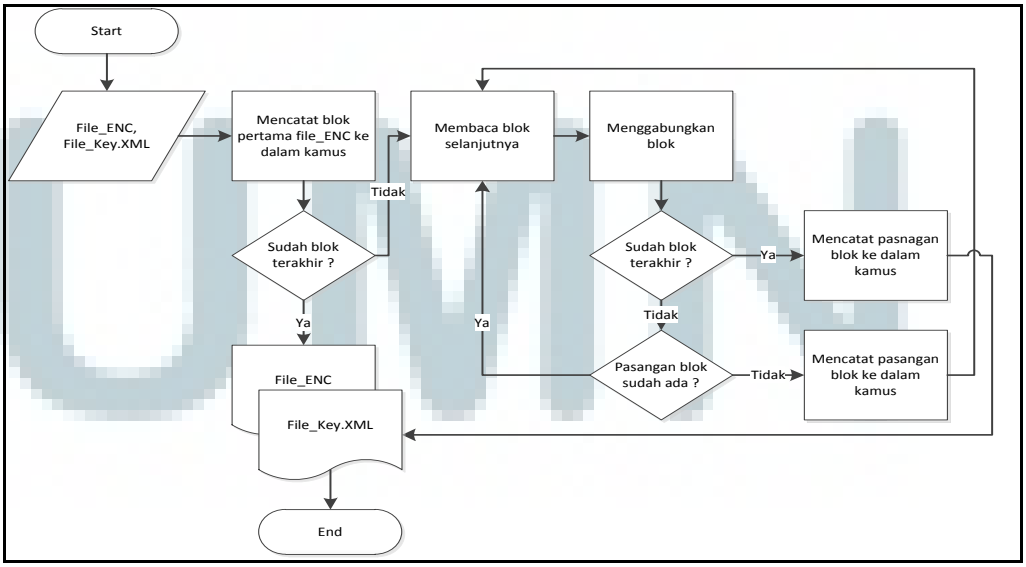
Dalam subproses *execute* enkripsi, aplikasi akan menggunakan *size* dari *file*_input dan data-data dari *file*_Key.XML. Pertama-tama aplikasi akan memotong isi file menjadi blok-blok dalam rentang [0, …, P-1]. Kemudian sistem akan men-*generate* sebuah bilangan acak K yang dibutuhkan untuk melakukan perhitungan enkripsi. *File*_input akan dienkripsi mulai dari blok pertama hingga blok terakhir. Setiap blok akan dienkripsi dengan rumus $e1 = G^K \bmod P$ dan $e2 = Y^K m \bmod P$. Pasangan e1 dan e2 akan menjadi blok baru di dalam *ciphertext*. Gambar 3.6. menampilkan *system flow* subproses *execute* enkripsi.



Gambar 3.6. *System Flow* subproses *execute* enkripsi

B.6. System Flow Subproses Kompresi LZW

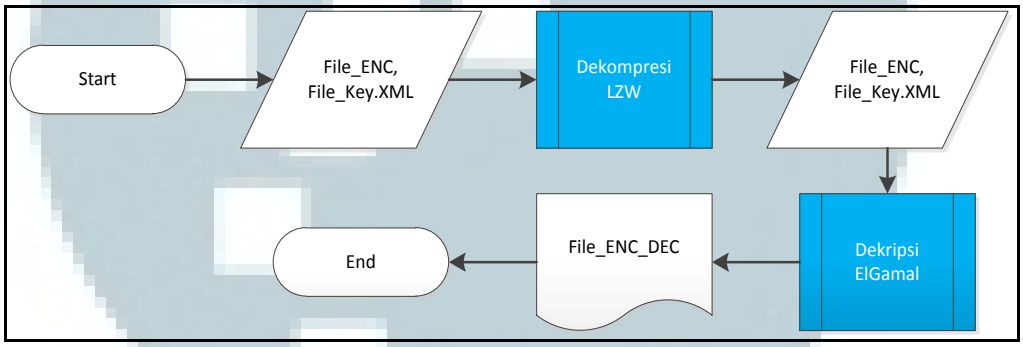
Pada subproses kompresi LZW, aplikasi akan menggunakan isi dari *file*_ENC yang sudak terenkripsi menjadi blok-blok. Pertama-tama aplikasi akan membaca blok pertama dan mencatatnya kedalam kamus. Kemudian aplikasi akan terus membaca hingga mencapai blok terakhir. Jika ditemukan pasangan blok baru pada saat membaca blok, maka aplikasi akan mencatatnya kedalam kamus. Pasangan blok yang sudah pernah dicatat ke dalam kamus tidak akan dicatat ulang. Gambar 3.7. menampilkan *system flow* subproses kompresi LZW.



Gambar 3.7. *System Flow* subproses kompresi LZW

B.7. System Flow Subproses Dekripsi

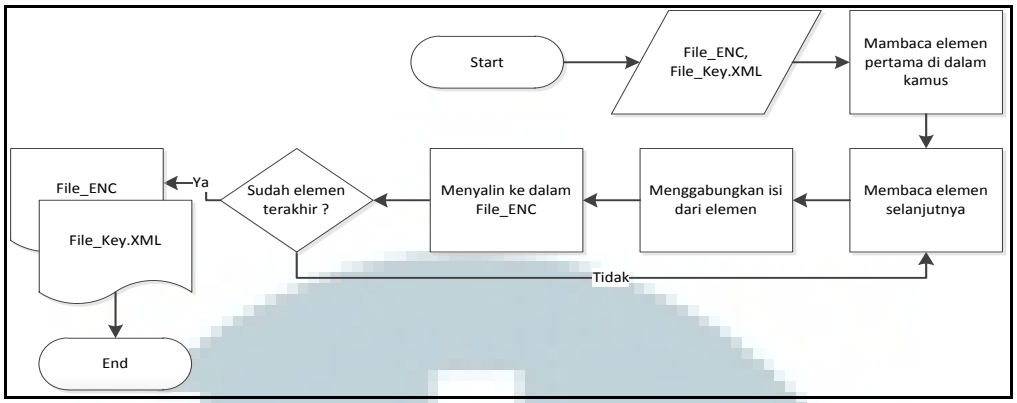
Pada subproses dekripsi, *user* harus memilih *file* yang ingin di dekripsi (*file*_ENC) dan kunci yang digunakan (*file*_Key.XML). Pertama-tama aplikasi akan melakukan dekompresi pada *file*_ENC. Setelah proses dkompresi selesai dilakukan, aplikasi akan langsung mendekripsi *file*_ENC dengan menggunakan *file*_Key.XML yang diberikan. Setelah proses dekripsi selesai, *user* akan mendapatkan flie_ENC_DEC. Gambar 3.8. menampilkan *system flow* subproses dekripsi.



Gambar 3.8. *System Flow* subproses dekripsi

B.8. System Flow Subproses Dekompresi LZW

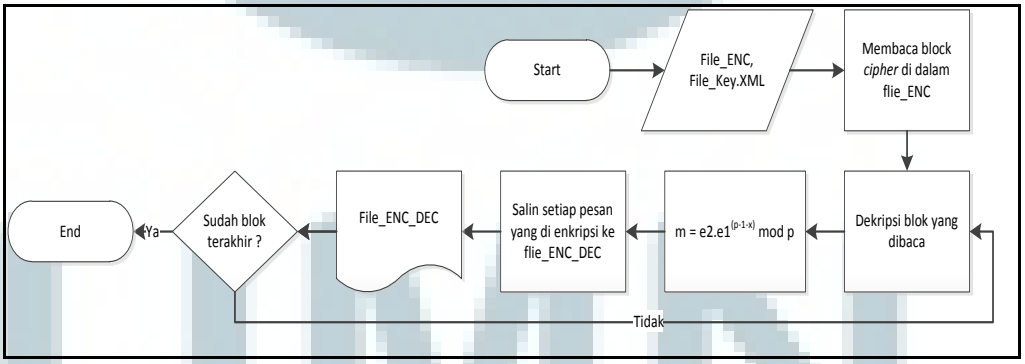
Pada subproses dekompresi LZW, aplikasi akan menggunakan *file*_ENC yang sudah terkompresi. Pertama-tama aplikasi akan membaca isi elemen di dalam kamus yang sudah dibuat pada proses kompresi. Aplikasi akan membaca isi kamus dari awal hingga akhir. Setiap elemen yang dibaca, isinya akan digabungkan dengan isi elemen yang dibaca sebelumnya. Seluruh hasil penggabungan isi elemen akan disimpan ke dalam *file*_ENC. Gambar 3.9. menampilkan *system flow* subproses dekompresi LZW.



Gambar 3.9. *System Flow* subproses dekompresi LZW

B.9. System Flow Subproses Dekripsi ElGamal

Pada subproses dekripsi ElGamal, aplikasi akan menggunakan *file*_ENC dan *file*_Key yang sudah ditentukan. Aplikasi memulai dekripsi dengan membaca blok per blok. Setiap blok pada yang dibaca dalam *file*_ENC akan didekripsi dengan rumus $m = e2.e1^{(P-1-X)} \bmod P$. Setiap blok yang dienkripsi, disalin ke dalam *file*_ENC_DEC. Gambar 3.10. menampilkan *system flow* subproses dekripsi ElGamal.



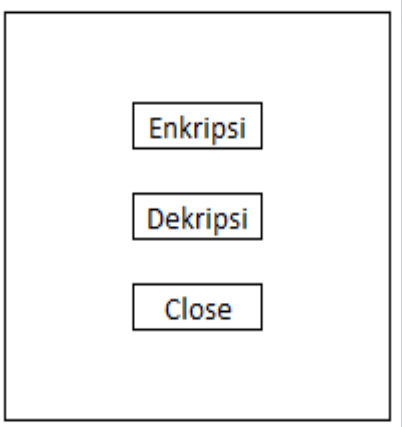
Gambar 3.10. *System Flow* subproses dekripsi ElGamal

3.3.2. Perancangan Antarmuka

Perancangan antarmuka ini dibuat untuk menjadi acuan pembangunan antarmuka aplikasi.

A. Halaman Utama

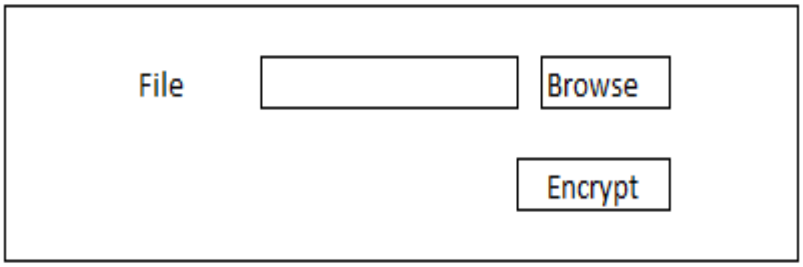
Halaman utama memiliki tiga buah komponen, yaitu tombol enkripsi, tombol dekripsi, dan tombol close. Tombol enkripsi untuk masuk ke halaman login. Tombol dekripsi untuk masuk ke halaman dekripsi. Tombol close untuk keluar dari aplikasi.



Gambar 3.11. Sketsa antarmuka halaman utama

B. Halaman Enkripsi

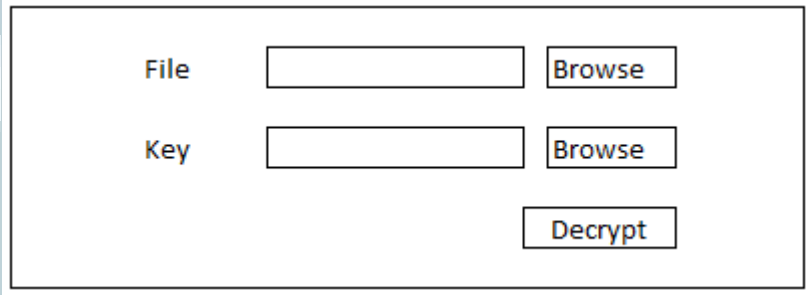
Halaman enkripsi adalah halaman dimana *user* dapat menjalankan fungsi enkripsi. Halaman enkripsi memiliki dua komponen, yaitu tombol *browse* dan tombol *encrypt*. Tombol *browse* disediakan untuk *user* memilih *file* yang ingin di enkripsi (*file*_input). Tombol *encrypt* disediakan untuk menjalankan fungsi enkripsi.



Gambar 3.12. Sketsa antarmuka halaman enkripsi

C. Halaman Dekripsi

Halaman dekripsi adalah halaman dimana *user* dapat menjalankan fungsi dekripsi. Halaman dekripsi memiliki tiga buah komponen, yaitu dua buah tombol *browse* dan tombol *decrypt*. Tombol *browse* disediakan masing-masing untuk *user* memilih *file* yang ingin di dekripsi (*file*_ENC) dan memilih *key* (*file*_Key.XML). Tombol *decrypt* disediakan untuk menjalankan fungsi dekripsi.



Gambar 3.13. Sketsa antarmuka halaman dekripsi