BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisa Masalah

Salah satu dukungan yang ada pada perangkat komputer, pengguna dapat membuat *folder* untuk menyimpan file yang diinginkan. Semua pengguna komputer,baik berupa pc ataupun laptop menggunakan layanan ini, dikarenakan fungsi dari folder itu sendiri. Namun tingkat keamanan pada folder masih belum terjamin dalam pengamanan file yang ada dalam folder tersebut. Hal ini yang cenderung menimbulkan bahaya bagi pengguna yang memiliki file-file 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. Saat ini sudah banyak berkembang algoritma kriptografi yang mendukung untuk mengamankan suatu pesan atau informasi yang ada dari orang atau pihak yang tidak berhak untuk mengakses data atau informasi tersebut.

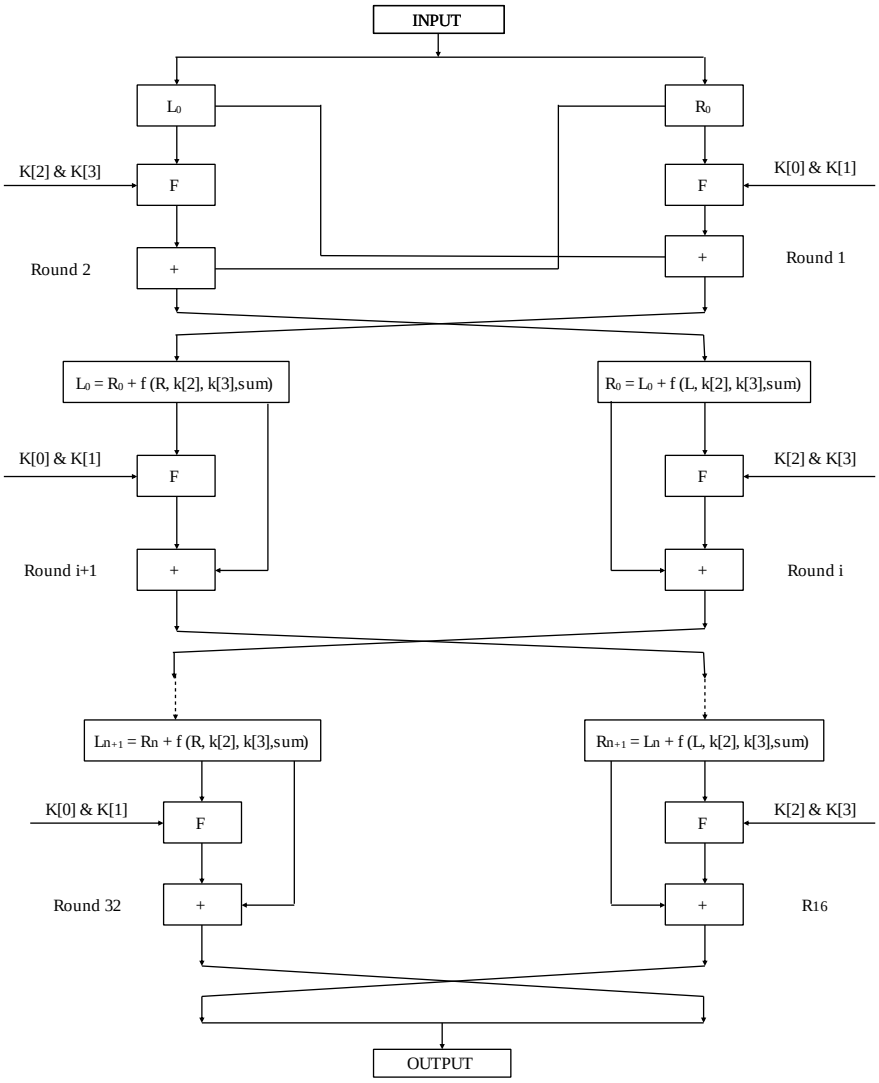
Salah satu teknik kriptografi adalah menggunakan algoritma *Tiny Encryption Algorithm* (*TEA*). *TEA* adalah algoritma *block cipher* yang diciptakan oleh *David J. Wheeler* dan *Roger M. Needham* dari *Cambridge University* tahun 1994. Hal yang paling menonjol dari *TEA* adalah kesederhanaan implementasi,

ketiadaan *S-Box* maupun *P-Box* dan kecepatan yang tinggi. Penulis mencoba untuk membangun sebuah aplikasi yang dapat mengamankan *folder* dengan algoritma kriptografi *Tiny Encryption Algorithm* (*TEA*).

III.2. Algoritma TEA (*Tiny Encription Algorithm*)

Tiny Encription Algorithm (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David Wheeler dan Roger Needham dari Computer Laboratory, Cambridge University, England pada bulan November 1994. Algoritma ini merupakan algoritma penyandian *block cipher* yang dirancang untuk penggunaan memory yang seminimal mungkin dengan kecepatan proses yang maksimal.

Sistem penyandian TEA menggunakan proses *feistel network* dengan menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR. Hal ini dimaksudkan untuk menciptakan sifat non-linearitas. Pergeseran dua arah (ke kiri dan ke kanan) menyebabkan semua bit kunci dan data bercampur secara berulang ulang. Struktur penyandian TEA terlihat pada gambar III.1 sebagai berikut:



Gambar III.1. Algoritma TEA

TEA memproses 64-bit input sekali waktu dan menghasilkan 64-bit output. TEA menyimpan 64-bit input kedalam L_0 dan R_0 masing masing 32-bit, sedangkan 128-bit kunci disimpan kedalam k[0], k[1], k[2], dan k[3] yang masing masing berisi 32-bit. Diharapkan teknik ini cukup dapat mencegah penggunaan teknik *exshautive search* (teknik pencarian solusi secara solusi *brute force* atau percobaan terhadap semua kunci yang mungkin, untuk masalah yang melibatkan pencarian elemen dengan sifat khusus, biasanya di antara objek-objek

kombinatorik seperti permutasi, kombinasi, atau himpunan bagian dari sebuah himpunan) secara efektif. Hasil outputnya akan disimpan dalam L_{16} dan $R_{16.}$

Bilangan delta berasal dari *golden number*, digunakan delta=$(\sqrt{5}\ -1)2^{31}$. Suatu bilangan delta ganda yang berbeda digunakan dalam setiap roundnya sehingga tidak ada bit dari perkalian yang tidak berubah secara teratur. Berbeda dengan struktur *feistel* yang semula hanya mengoperasikan satu sisi yaitu sisi sebelah kanan dengan sebuah fungsi F, pada algoritma TEA kedua sisi dioperasikan dengan sebuah fungsi yang sama yaitu fungsi F (L_0 = L_0 + (((R0<<4)+k[0])^ R0+sum^((R0>>5)+k[1])) dan R_0 = R_0 + (((L0<<4)+k[2]^ L0+sum^((L0>>5)+k[3])) untuk satu round TEA.

III.3. Analisis Proses Enkripsi Algoritma TEA

Untuk melakukan enkripsi, proses diawali dengan input-bit teks terang sebanyak 64-bit. Kemudian 64-bit teks terang tersebut dibagi menjadi dua bagian, yaitu sisi kiri (L0) sebanyak 32-bit dan sisi kanan (R0) sebanyak 32-bit. Setiap bagian teks terang akan dioperasikan sendiri-sendiri. R0 (z) akan digeser kekiri sebanyak empat (4) kali dan ditambahkan dengan kunci k[0]. Sementara itu z ditambah dengan sum (delta) yang merupakan konstanta. Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Kemudian di-XOR-kan dengan hasil penambahan antara z yang digeser kekanan sebanyak lima (5) kali dengan kunci k[1]. Hasil tersebut kemudian ditambahkan dengan L0 (y) yang akan menjadi R1. Untuk lebih memahami dapat dilihat pada blok diagram proses enkripsi algoritma TEA berikut ini.

Sisi sebelah kiri akan mengalami proses yang sama dengan sisi sebelah kanan. L0 (y) akan digeser kekiri sebanyak empat (4) kali dan ditambahkan dengan kunci k[2]. Sementara itu Y ditambah dengan sum (delta). Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Kemudian di-XOR-kan dengan hasil penambahan antara Y yang digeser kekanan sebanyak lima (5) kali dengan kunci k[3]. Hasil tersebut kemudian ditambahkan dengan R0 (Z) yang akan menjadi L1.

III.4. Analisis Proses Deskripsi Algoritma TEA

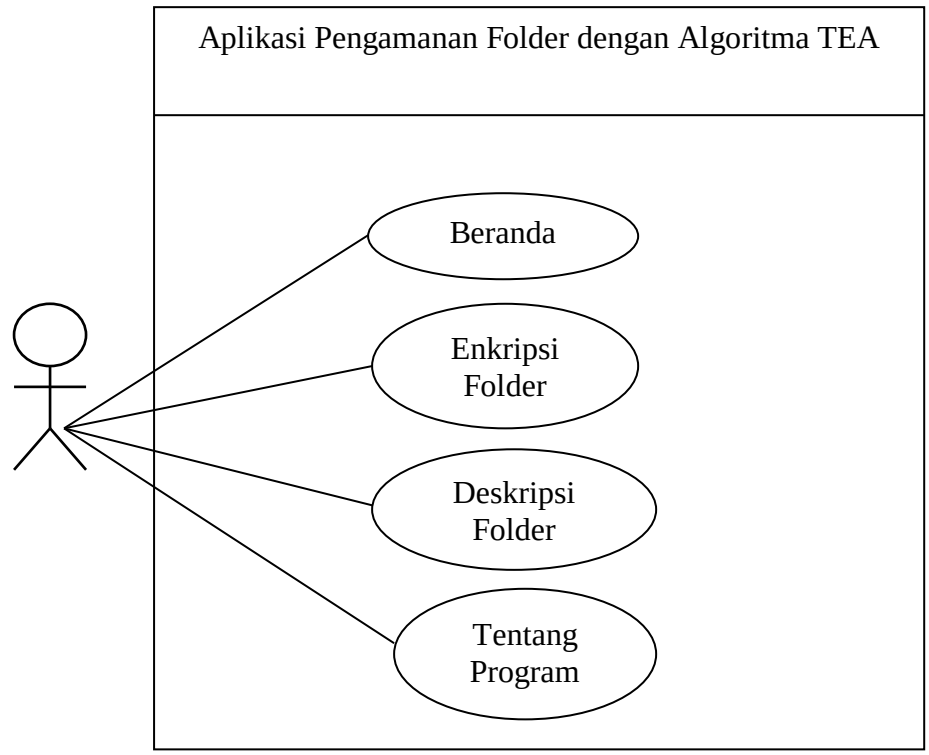
Untuk proses deskripsi pada algoritma TEA sama halnya dengan proses enkripsinya. Hanya saja terjadi perbedaan pada penjadwalan kuncinya yaitu pada proses enkripsi untuk cipher R yang mengalami pergeseran bit ke kiri sebanyak 4 bit digunakan kunci k[0] pada proses deskripsi digunakan kunci k[1], untuk cipher R yang mengalami pergeseran ke kanan sebanyak 5 bit menggunakan kunci [1] pada proses deskripsi menggunakan kunci k[0]. Begitu juga halnya dengan cipher L, pada proses enkripsi untuk cipher L yang mengalami pergeseran ke kiri sebanyak 4 bit menggunakan kunci k[2] pada yang mengalami Pergeseran proses deskripsi digunakan kunci k[3]. Untuk cipher L yang mengalami pergeseran kekanan sebanyak 5 bit digunakan kunci k[3] pada proses deskripsi digunakan kunci k[2].

III.5. Desain Sistem Baru

Desain sistem baru manggunakan bahasa pemodelan UML yang terdiri dari *Use Case Diagram, Activity Diagram,* dan *Sequence Diagram.*

III.5.1. Use Case Diagram

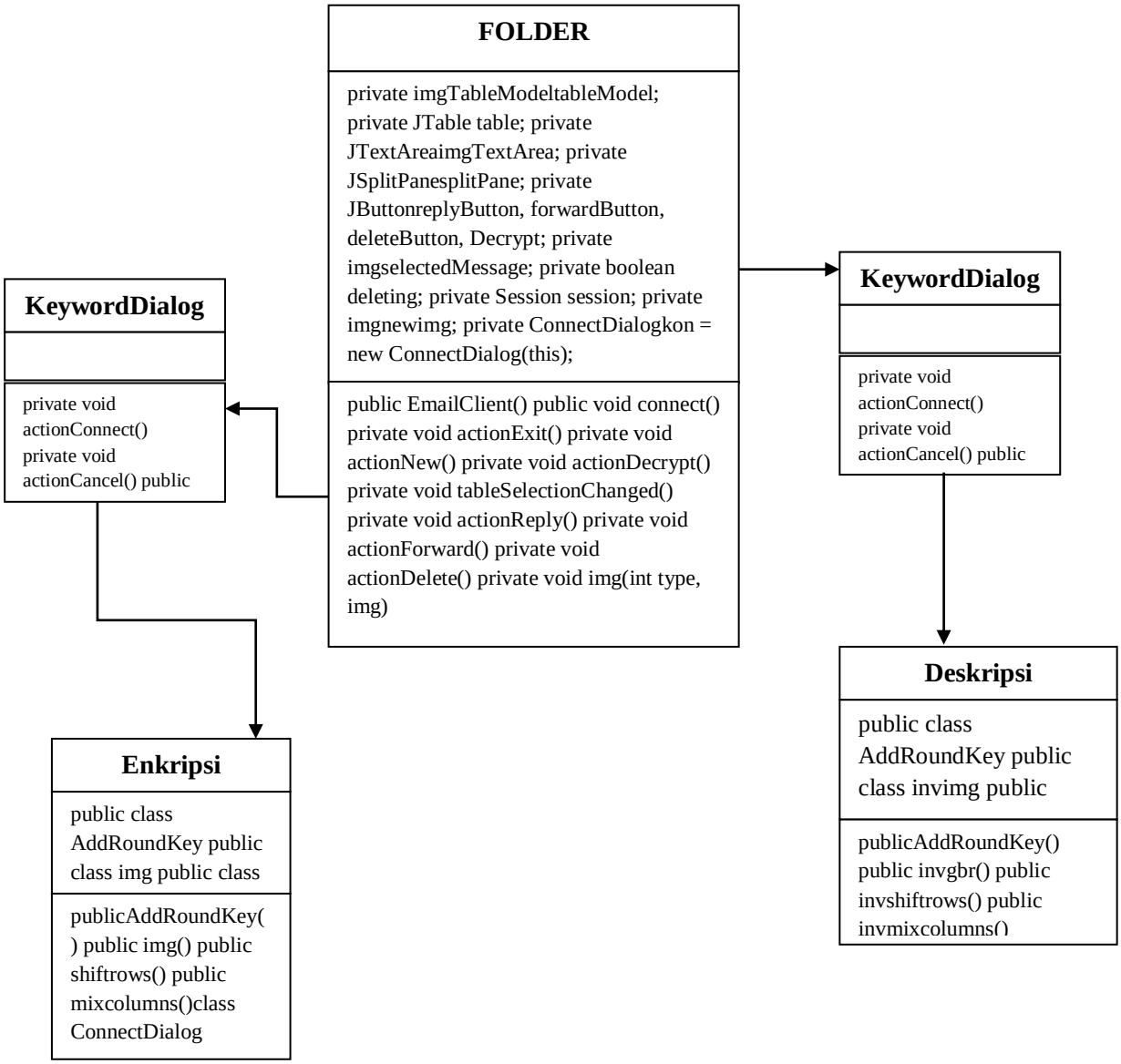
Secara garis besar, proses sistem yang akan dirancang digambarkan dengan use case diagram yang terdapat pada gambar III.2. :



Gambar III.2. Use Case Diagram Aplikasi Pengamanan Folder dengan Metode TEA

III.5.2. Class Diagram

Rancangan kelas-kelas yang akan digunakan pada system yang akan diancang dapat dilihat pada gambar berikut ini :



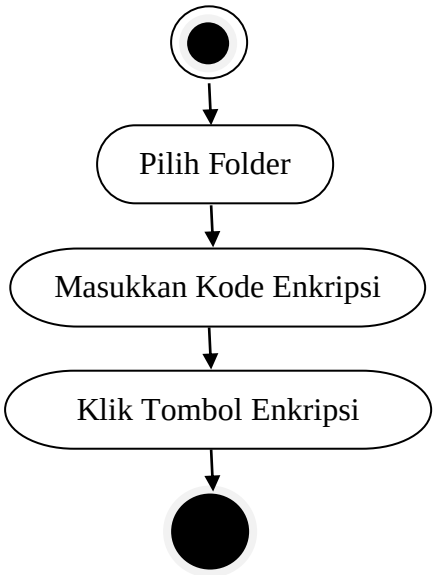
Gambar III.3. *Class* Diagram Aplikasi Pengamanan Folder Dengan Metode TEA (Tiny Encription Algorithm)

III.5.3. Activity Diagram

Diagam aktivitas menggambarkan suatu urutan proses yang terjadi pada sistem dari dimulainya aktivitas hingga aktivitas berhenti. Diagram aktivitas hamper mirip dengan diagram *flowchart*. Diagram aktivitas merupakan salah satu cara untuk memodelkan hal yang terjadi dalam suatu *use-case*. Berikut *activity* diagram yang ditunjukkan pada gambar dibawah ini :

1. Activity Diagram Enkripsi Folder

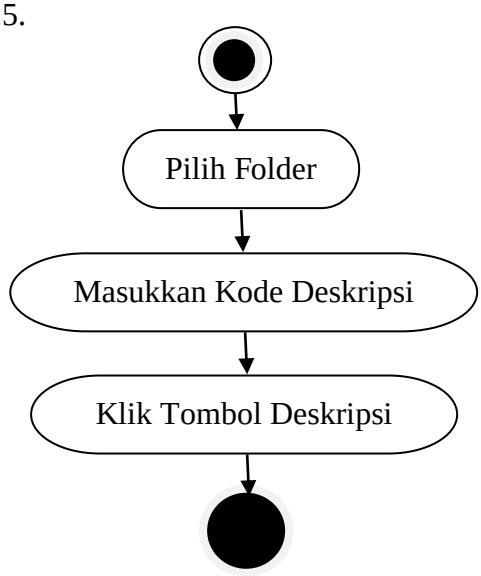
Pada *activity* diagram enkripsi *folder* menjelaskan bahwa informasi atau data enkripsi *folder*. Adapun *activity* diagram enkripsi *folder* dapat dilihat pada gambar III.4.



Gambar III.4. *Activity* Diagram Enkripsi *Folder*

2. Activity Diagram Deskripsi Folder

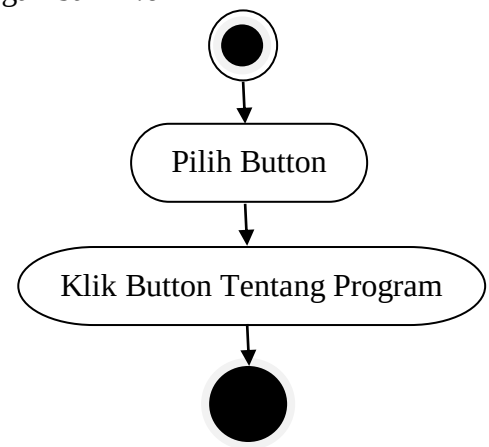
Pada *activity* diagram deskripsi *folder* menjelaskan bahwa informasi atau data deskripsi *folder*. Adapun activity diagram enkripsi folde dapat dilihat pada gambar III.5.



Gambar III.5. *Activity* Diagram Deskripsi *Folder*

3. Activity Diagram Melihat Tentang Program

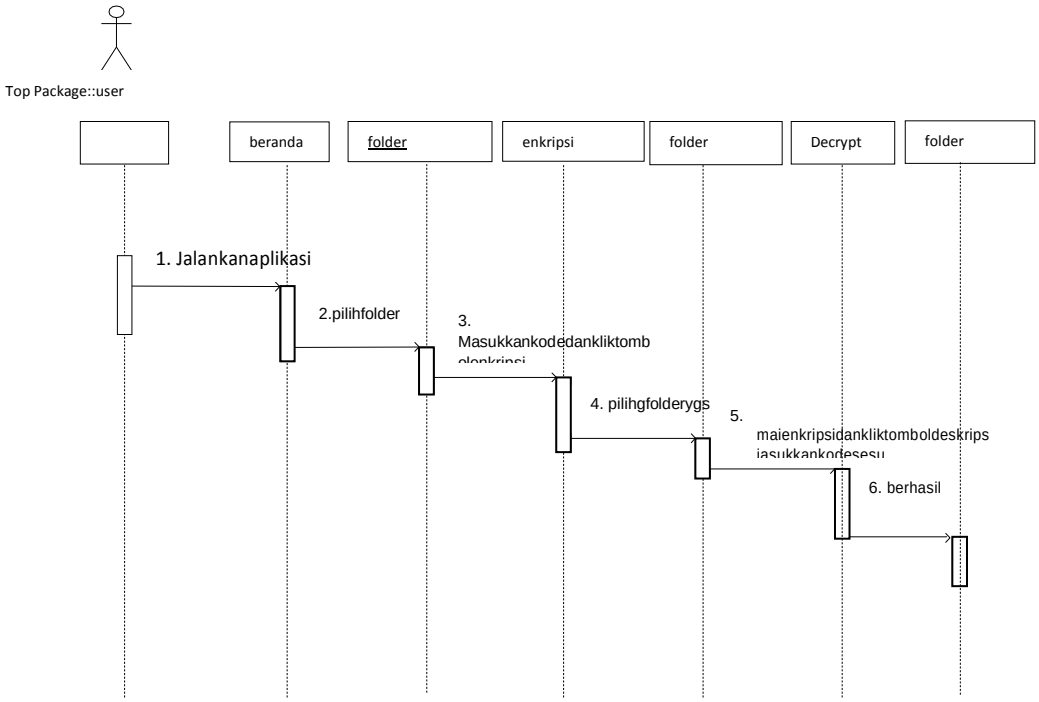
Pada activity diagram ini menjelaskan tentang informasi atau data diri pembuat program. Adapaun activity diagram tentang program dapat dilihat pada gambar III.6



Gambar III.6. *Activity* Diagram Melihat Tentang Program

III.5.3 Sequence Diagram

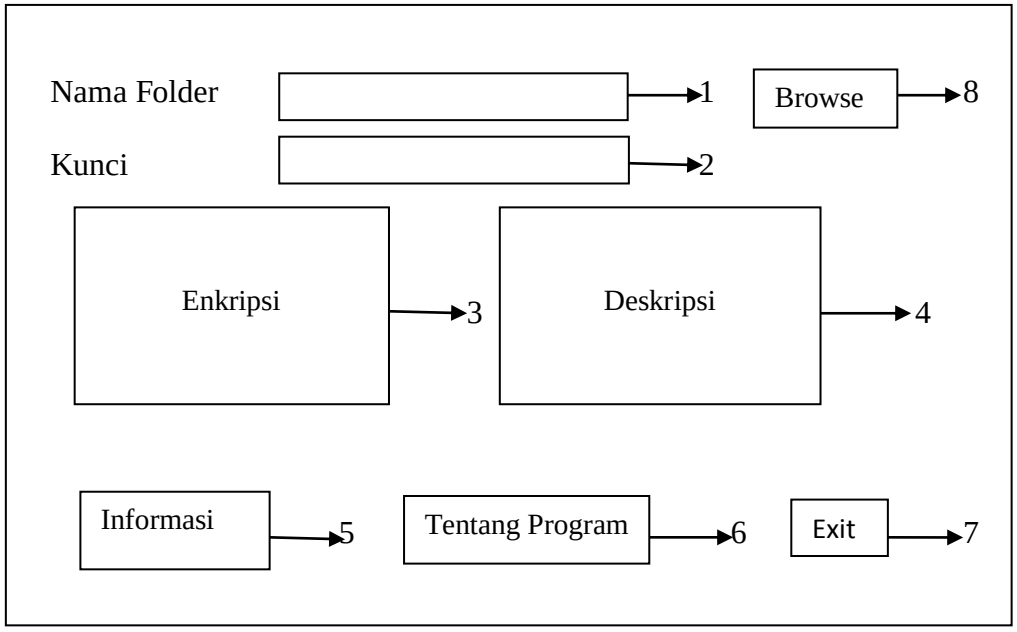
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dandimensi horizontal (objek-objek yang terkait). Serangkaian kegiatan saat terjadi *event* pada aplikasi ini dapat dilihat pada gambar III.7:



Gambar III.7. *Sequence* Diagram Aplikasi Pengamanan Folder Dengan Metode Tiny Encryption Algorithm (TEA)

III.6. Desain Tampilan Rancangan

Rancangan *form* enkripsi dan deskripsi berfungsi untuk menampilkan keseluruhan rancangan Aplikasi Pengamanan Folder Dengan Metode Tiny Encryption Algorithm, rancangan dapat dilihat pada gambar berikut :



Gambar III.8. Desain Tampilan Rancangan

Keterangan gambar :

1. Text Field : Untuk menampilkan hasil dari pencarian
2. Text Field : Untuk menampilkan password
3. Button Enkripsi : Untuk mengenkripsi hasil
4. Button Deskripsi : Untuk mendeskripsi hasil
5. Button Informasi : Untuk menampilkan informasi
6. Button Tentang Program : Untuk menampilkan tentang program
7. Button Exit : Untuk Keluar Dari Program
8. Button Browse : Untuk melakukan Pencarian