BAB III

ANALISA DAN PERANCANGAN

III.1. Analisa Sistem

Pada tahapan analisis dan perancangan ini bertujuan menganalisa kebutuhan pengembangan aplikasi media pembelajaran enkripsi dengan algoritma *RC4*. *Input* yang diproses dalam aplikasi yang dirancang berupa *file* citra atau *file* gambar sehingga akan terjadi pengkodean yang dilakukan oleh sistem sehingga memungkinkan keamanan data *file* tersebut. Pada perancangan hanya memberikan informasi tentang *file* gambar, bukanlah *file* lainnya seperti *file* data teks atau *file* video. Algoritma *RC4* adalah algoritma yang bertujuan untuk mengenkripsi *file-file* yang asli kedalam bentuk yang tidak umum, Metode yang digunakan pada sistem ini adalah dengan metode *dictionary*. Sedangkan tahapan dalam sistem adal ada 2 (dua) tahapan. Tahap pertama yaitu proses ekripsi atau pengkodean terhadap *file* yang diinput serta proses dekripsi atau pengembalian *file* yang diinput ke bentuk semula. Desain dan implementasi ini meliputi desain data, deskripsi sistem, desain proses dan implementasi desain dan semua yang diperlukan dalam aplikasi enkripsi yang dirancang.

III.1.1. Analisa Input

Dalam sistem media pembelajaran pengenkripsian *file* gambar yang akan di implementasikan dalam aplikasi adalah menggunakan algloritma *RC4*. Dengan membaca tiap karakter yang dimasukan dari *file* yang dimasukan lalu diproses hingga hingga membentuk suatu tampilan yang tidak dapat dibaca. Dalam proses

yang dikembangkan hanya menampilkan proses kerja algoritma *RC4* sebagai media pembelajaran agar pengguna dapat mengetahui bagaimana proses dari sebuah algoritma *RC4.*

III.1.2. Analisa Proses

Permasalahan yang dibahas adalah membuat suatu pengkodean dengan menggunakan algoritma *RC4*. Masalah enkripsi data dengan algoritma *RC4* muncul ketika proses enkripsi dalam sistem sedang. Pembahasan masalah lebih ditekankan pada proses indeks kerja algoritma *RC4*.

Algoritma *RC4* mengenkripsi dengan mengombinasikannya dengan *plainteks* dengan menggunakan *bit-wise* Xor (*Exclusive-or*). RC4 menggunakan panjang kunci dari 1 sampai 256 *byte* yang digunakan untuk menginisialisasikan tabel sepanjang 256 *byte*. Tabel ini digunakan untuk generasi yang berikut dari *pseudo random* yang menggunakan XOR dengan *plaintext* untuk menghasilkan *ciphertext*. Masing - masing elemen dalam tabel saling ditukarkan minimal sekali. Proses dekripsinya dilakukan dengan cara yang sama (karena Xor merupakan fungsi simetrik). Untuk menghasilkan *keystream, cipher* menggunakan *state* internal yang meliputi dua bagian :

1. Tahap *key scheduling* dimana *state automaton* diberi nilai awal berdasar kan kunci enkripsi.

   State yang diberi nilai awal berupa *array* yang merepresentasikan suatu permutasi dengan 256 elemen, jadi hasil dari algoritma KSA adalah permutasi awal. *Array* yang mempunyai 256 elemen ini (dengan indeks 0 sampai dengan 255) dinamakan S. Berikut adalah algoritma KSA dalam

bentuk *pseudo-code* dimana *key* adalah kunci enkripsi dan *keylength* adalah besar kunci enkripsi dalam *bytes* (untuk kunci 128 bit, *keylength* = 16).

```
for i = 0 to 255
    S [i] := i
    j := 0
for i = 0 to 255
    j := (j + S[i] + key  [I mod keylenght] ) mod 256
    swap (S[i], S[j])
```

2. Tahap *pseudo-random generation* dimana *state automaton* beroperasi dan *output*nya menghasilkan *keystream*. Setiap putaran, bagian *keystream* sebesar 1 *byte* (dengan nilai antara 0 sampai dengan 255) dioutput oleh PRGA berdasarkan *state* S. Berikut adalah algoritma PRGA dalam bentuk *pseudo-code*:

```
i := 0
j := 0
loop
    i :=  ( i + 1 ) mod 256
    j := ( j + S[i] ) mod 256
    swap ( S[i], S[j] )
    output S[ (S[i] + S[j]) mod 256]
```

Setelah terbentuk *keystream*, kemudian *keystream* tersebut dimasukkan dalam operasi XOR dengan *plaintext* yang ada, dengan sebelumnya pesan dipotong-potong terlebih dahulu menjadi *byte-byte*.

III.1.3. Analisa Output

Dari hasil analisa *input* dan analisa proses pada akhirnya akan menghasilkan *output*/hasil keluaran yang diterima pengguna, dari setiap bentuk *file* yang dimasukkan yang telah dienkripsi dengan menggunakan algoritma *RC4* akan diubah kedalam bentuk yang tidak dapat dikenali dan hanya akan dapat dilihat jika hasil yang sudah dieknripsi tersebut dikembalikan kebentuk semula dengan proses dekripsi.

III.2. Strategi Pemecahan Masalah

Untuk membangun aplikasi enkripsi dan dekripsi *file* gambar sesuai penggunaan algoritma *RC4* sebagai media pembelajaran. Beberapa strategi pemecahan masalah dalam perancangan adalah sebagai berikut :

1 *Input* dan *Output* merupakan sebuah teks masukan yang dapat diproses oleh aplikasi.

2 Proses enkripsi dan dekripsi pada uji coba hanya dilakukan pada tiap *file* gambar atau *file* citra.

3 *Interface* memunggunakan tampilan yang disajikan dalam membaca aplikasi yagn telah dienkripsi.

III.3. Perancangan Sistem

Pada perancangan aplikasi menjelaskan mengenai rancangan dan hal-hal yang dikerjakan serta fitur-fitur yang akan dipakai pada aplikasi tersebut. Hal ini bertujuan untuk menjelaskan tahapan-tahapan yang dikerjakan, prosedur

penggunaan, disain tampilan, serta spesifikasi sistem dari segi perangkat lunak maupun perangkas keras yang digunakan dalam proses perancangan.

III.3.1. Analisa Kebutuhan fungsional

Dalam kebutuhan fungsional adalah jenis kebutuhan yang berisi untuk melengkapi perancangan. Kebutuhan fungsional juga berisi informasi-informasi apa saja yang harus ada dan dihasilkan. Berikut kebutuhan fungsional yang terdapat pada rancangan aplikasi yang dibangun :

1. Mengimplementasikan penggunaan bahasa pemrograman *java* dalam membuat aplikasi media pembelajaran algoritma *RC4*.
2. Aplikasi dapat menggambarkan penerapan algoritma *RC4* sebagai media pembelajaran.
3. *Input* dan *output* berupa *file* gambar yang dapat diproses dengan algoritma *RC4.*

III.3.2. Analisa Kebutuhan Nonfungsional

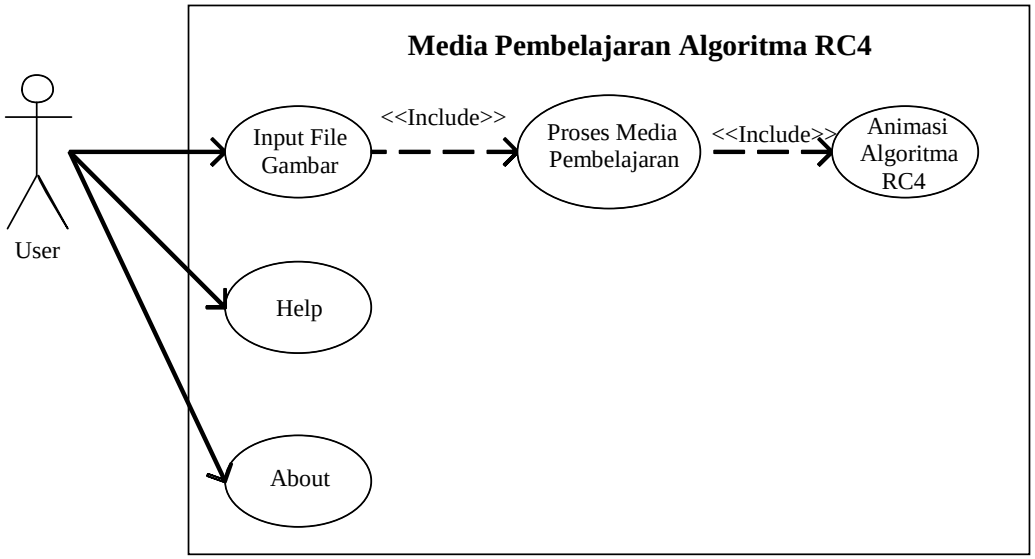
Dalam perancangan aplikasi media pembelajaran algoritma *RC4*, beberapa perangkat yang penulis gunakan agar aplikasi berjalan baik, yaitu sebagai berikut :

1. Perangkat Lunak (*Software*)
    a. *Operating System Windows Seven*.
    b. *Java* sebagai bahasa program yang digunakan serta *Netbean* sebagi bentuk pengkodean .

2. Perangkat Keras (*Hardware*)
    a. Komputer yang setara dengan *Intel pentium Dual Core*.
    b. *Mouse, keyboard,* dan *Monitor.*

III.3.3. *Use Case Diagram*

Pada *Use case* diagram menggambarkan aktor yang menggunakan aplikasi dan perilaku pengguna, dapat dilihat pada gambar III.1 berikut.



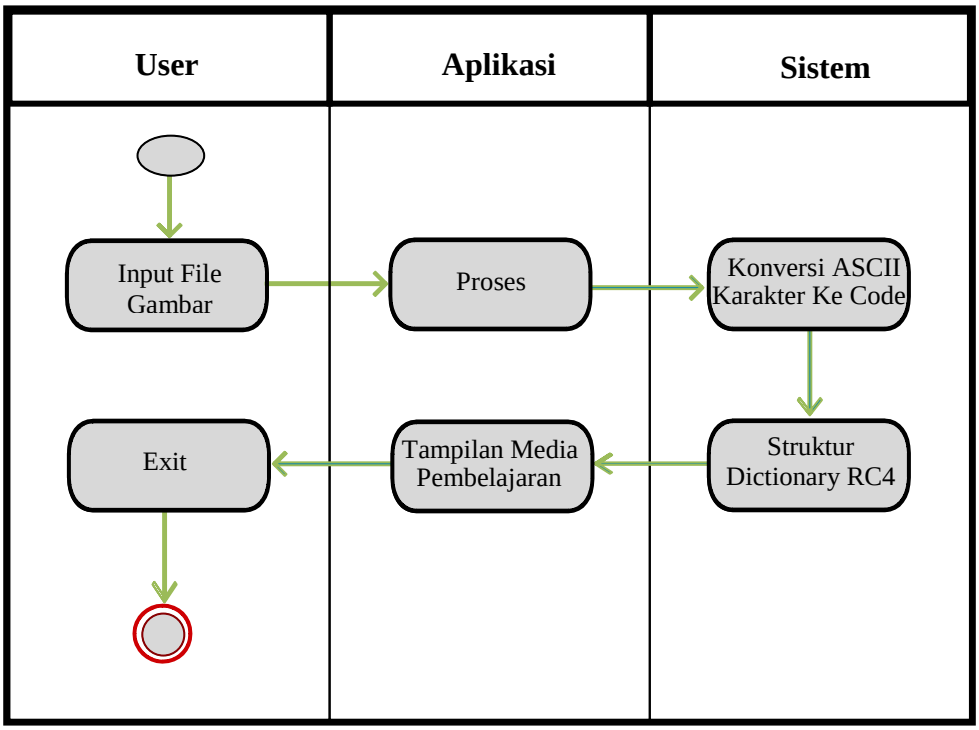
Gambar III.1. *Use Case Diagram* Pengguna Aplikasi

Kegiatan aktor atau pengguna pada aplikasi enkripsi *file* gambar, pengguna dapat memilih melakukan proses enkripsi *file* gambar atau dekripsi *file* gambar. Dari proses akan menampilkan media animasi 2D yang menjelaskan bagaimana proses kerja algoritma *RC4*.

III.3.4. *Activity Diagram* Proses Enkripsi

Pada *Activity diagram* menggambarkan berbagai aliran aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal.

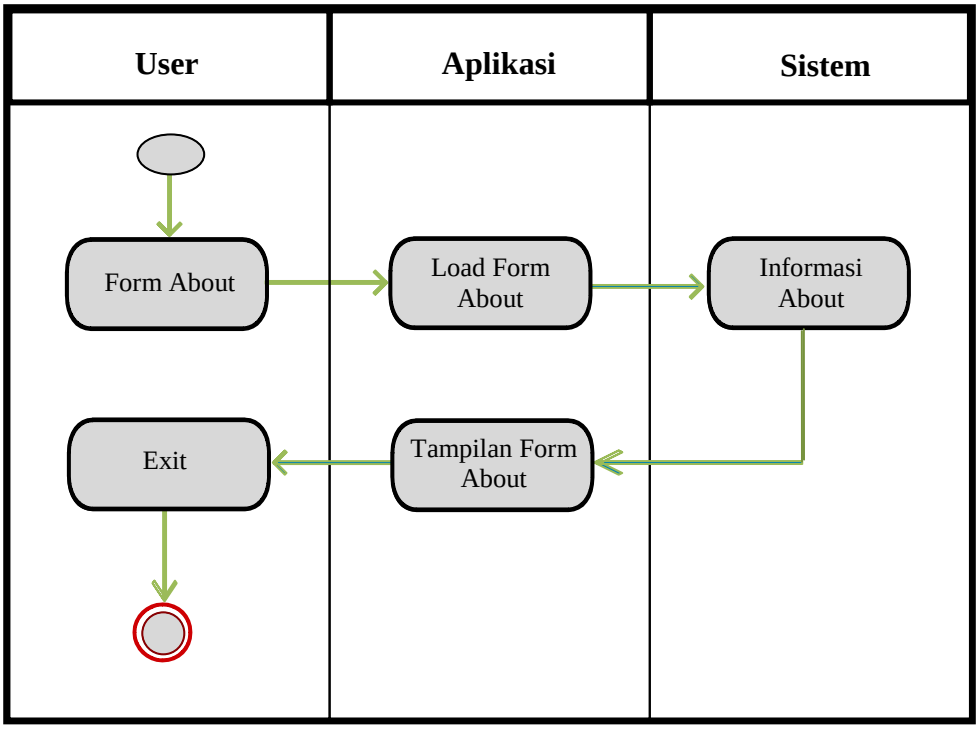
Adapun rancangan diagram aktivitas untuk proses enkripsi dari aplikasi yang dirancang adalah sebagai berikut.



Gambar III.2. *Activity Diagram* Proses Media

III.3.5. *Activity Diagram* Proses *Form About*

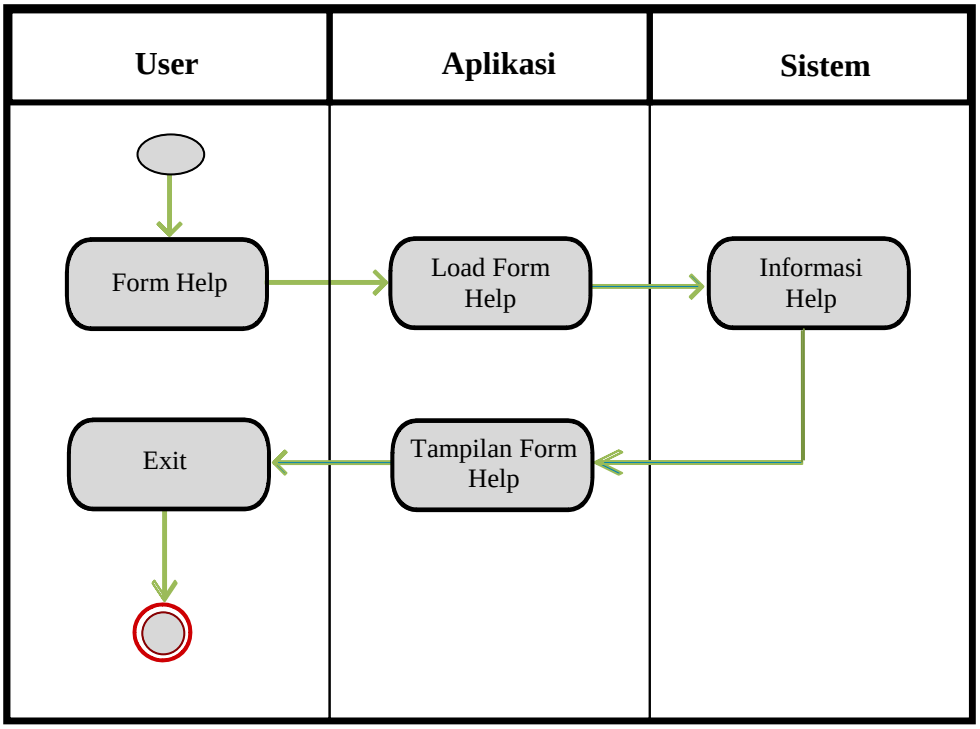
Dalam *Activity diagram* menjelaskan berbagai aktivitas dalam sistem yang sedang dirancang, tentang kegiatan dari masing-masing alir aktivitas berawal. Adapun rancangan untuk menampilkan *form about* dari aplikasi yang dirancang terdapat pada gambar III.3 berikut.



Gambar III.3. ***Activity Diagram Form About***

III.3.6. *Activity Diagram* Proses *Form* Help

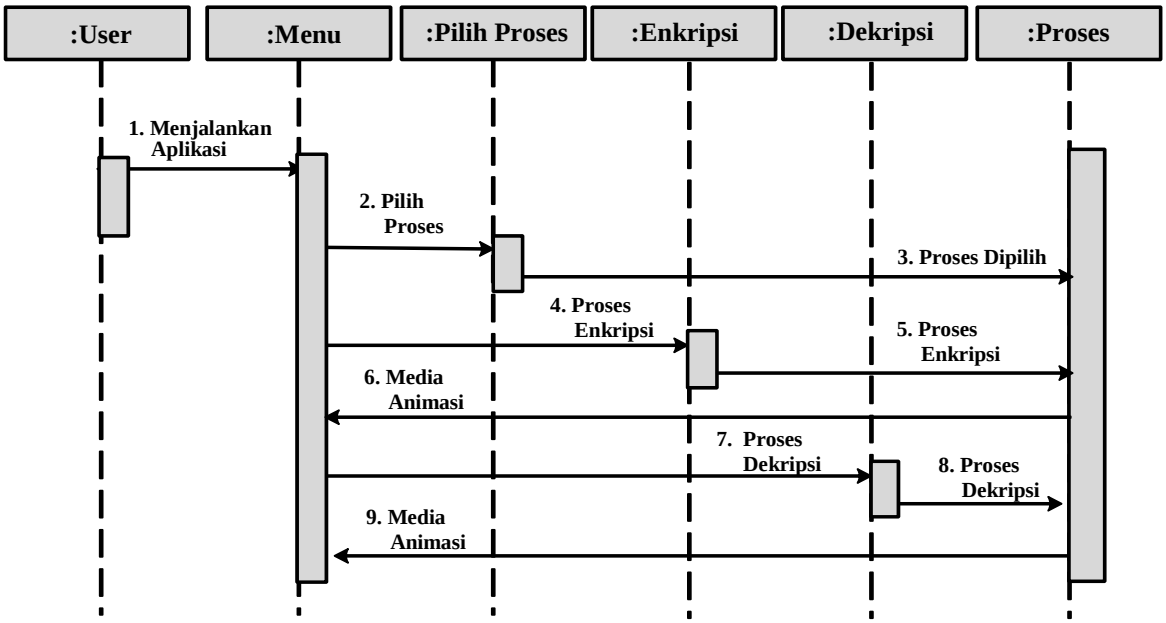
Pada *Activity diagram* ini menjelaskan tentang berbagai aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal. Adapun rancangan diagram aktivitas untuk menampilkan *form help* atau bantuan dari aplikasi yang dirancang terdapat pada gambar III.4. berikut ini :



Gambar III.4. *Activity Diagram Form* Help

III.3.7. *Sequence* Diagram Proses Enkripsi

Pada *Sequence* diagram proses enkripsi ini menggambarkan kegiatan dari skenario penggunaan aplikasi, *sequence* diagram memilih proses yang terjadi dengan memilih proses enkripsi pada media pembelajaran dapat dilihat pada gambar III.5 berikut.

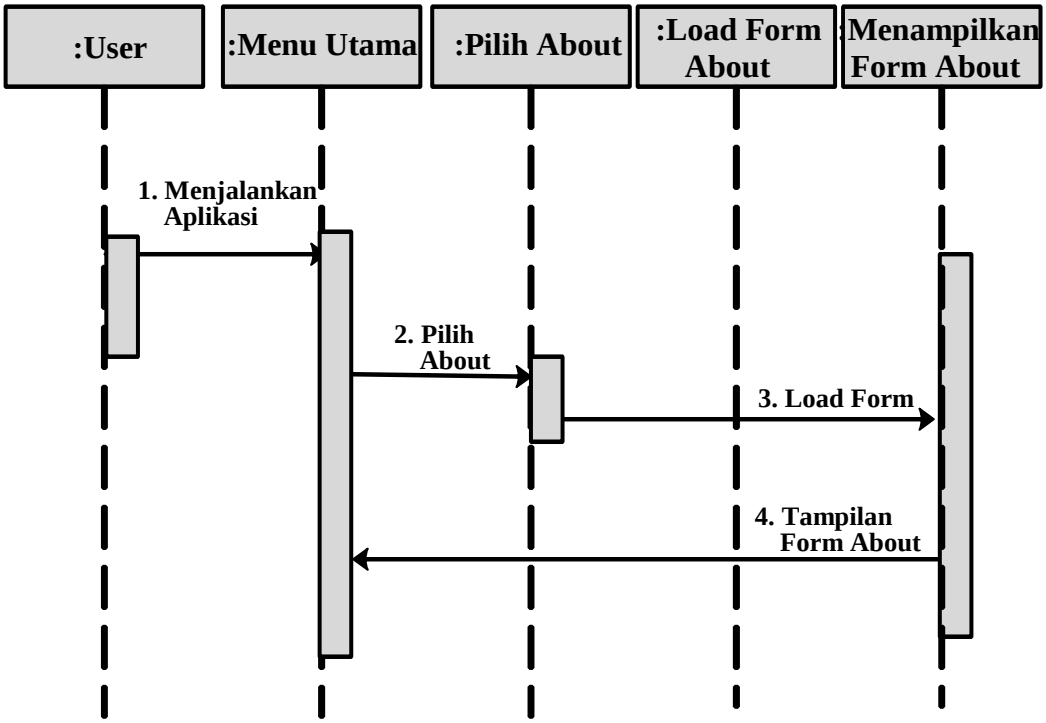


Gambar III.5. *Sequence Diagram* Proses Enkripsi

Untuk lebih jelasnya gambar *sequence* diagram proses enkripsi yang terdapat pada gambar III.5 diatas menjelaskan bahwa setelah *user* atau pengguna memulai menjalankan aplikasi sehingga terdapat menu utama dari sistem dengan lanjut berinteraksi melalui pilihan proses yang ada pada menu utama, pengguna memilih proses yang disediakan yaitu proses enkripsi dan serta proses dekripsi. Setelah pilihan proses ditentukan oleh pengguna kembali pada menu utama. Proses enkripsi maupun dekripsi hanya dapat dilakukan setelah pengguna memasukan *file input* yang ingin diproses. Selanjutnya proses algoritma *RC4* dapat dilakukan dengan menghasilkan tampilan media yang berfungsi sebagai pembelajaran.

III.3.8. *Sequence* Diagram Proses *Form About*

Pada rancangan *Sequence* diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *squence* diagram memilih proses *form about* pada media pembelajaran yang dapat dilihat pada gambar III.6 berikut.

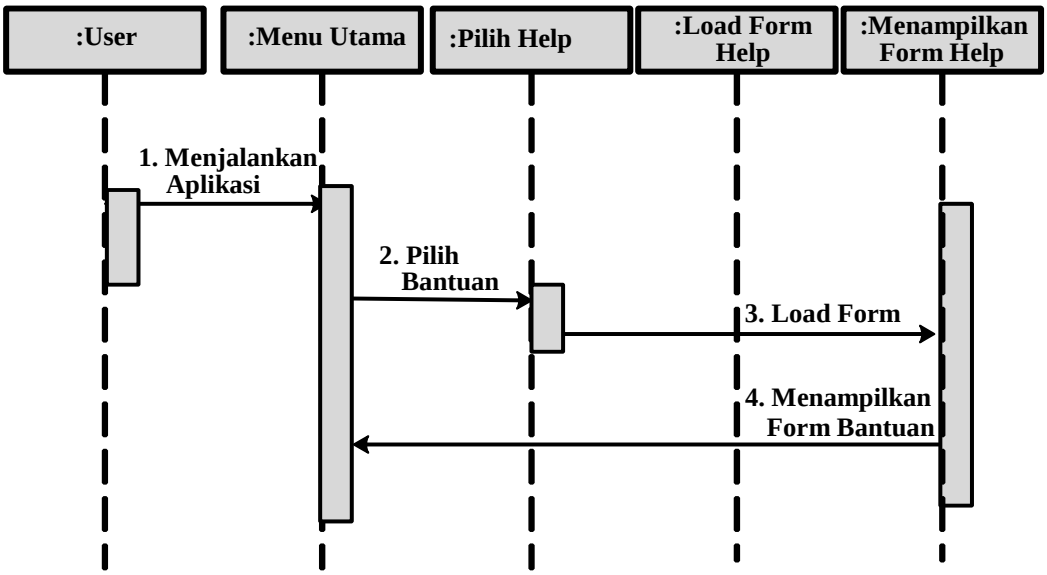


Gambar III.6. *Sequence Diagram* Proses *Form About*

Pada gambar diatas menjelaskan bagaimanan proses aktifitas yang terjadi saat user atau pengguna menjalankan aplikasi dan melanjutkan aktifitas dengan memilih form about atau bantuan sehingga menampilkan *form about* pada sistem.

III.3.9.*Sequence* Diagram Proses *Form Help*

Sequence diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *squence* diagram memilih proses *form* help pada media pembelajaran dapat dilihat pada gambar III.7 berikut.

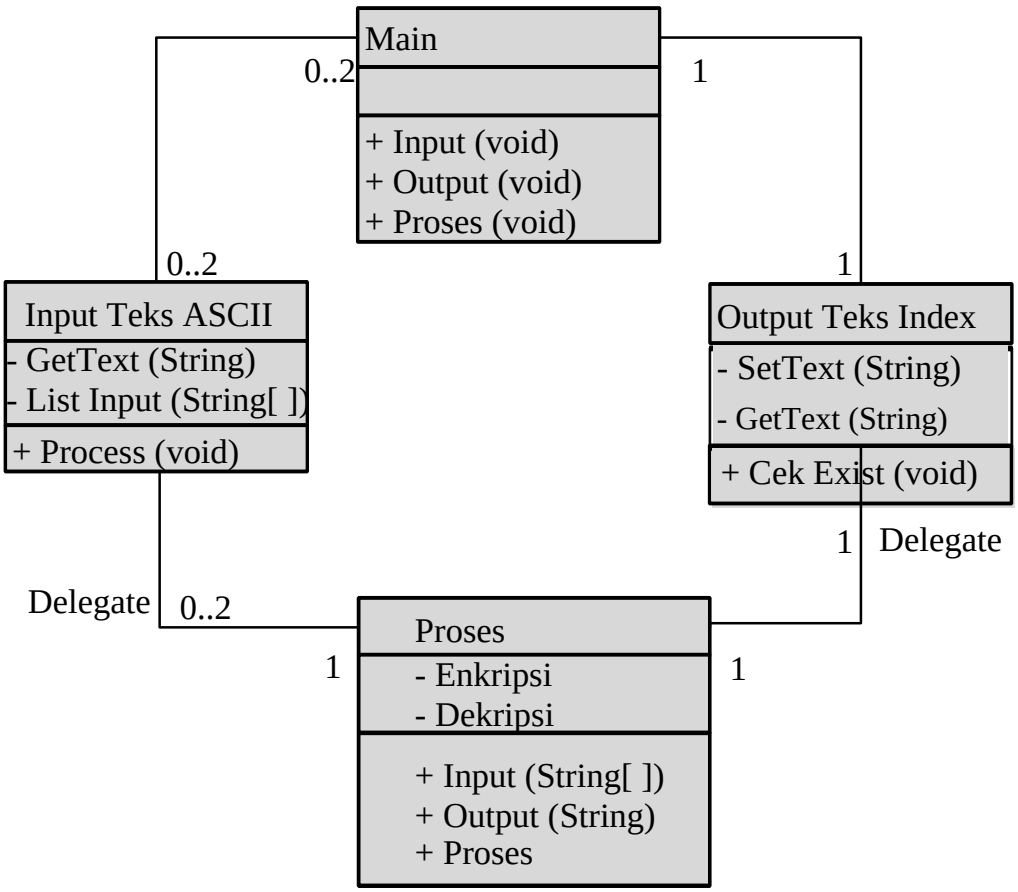


Gambar III.7. *Sequence Diagram* Proses *Form Help*

Pengguna berinteraksi melalui pilihan proses yang ada pada menu utama, dapat dilihat pada *squence diagram* diatas, pengguna memilih menu *form* help yang disediakan, sehingga menampilkan *form* bantuan yang berisi informasi tentang bantuan penggunaan dalam menjalankan aplikasi yang telah dirancang.

III.3.10. *Class* Diagram

Pada *Class* diagram perancangan aplikasi ini, dapat dilihat pada gambar III.8 berikut.



Gambar III.8. *Class* Diagram Sistem Perancangan Aplikasi

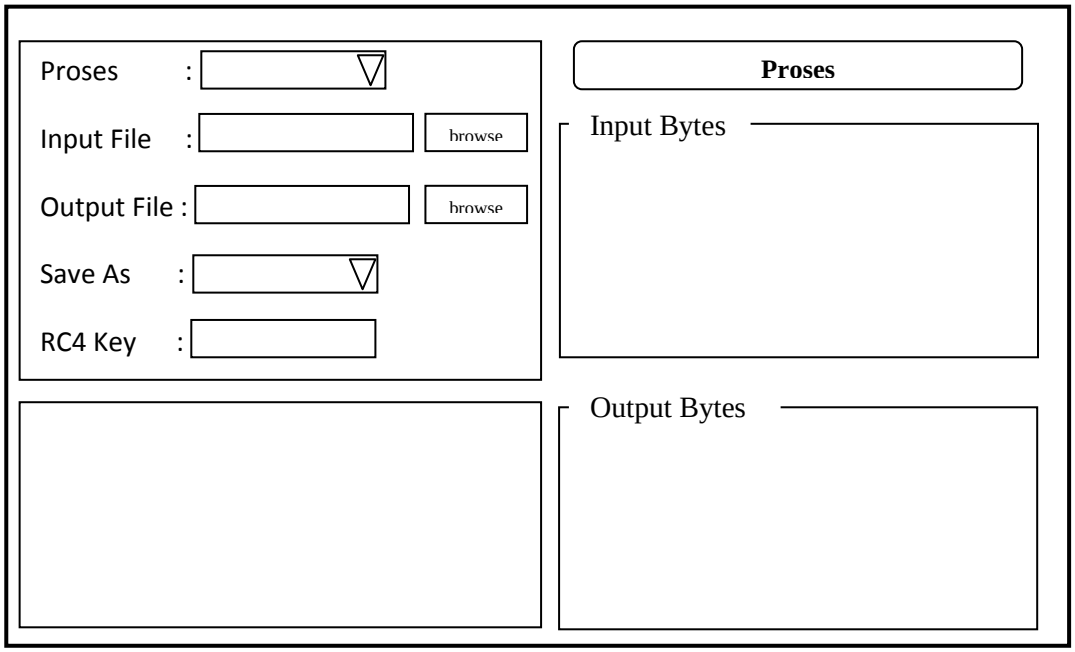
Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* diagram juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

III.4. Perancangan Tampilan

Pada perancangan tampilan aplikasi yang akan dibangun nantinya akan memiliki tampilan yang direncanakan. Adapun rancangan tampilan masing-masing halaman *form* tersebut dapat dijelaskan pada gambar berikut.

III.4.1. Tampilan *Form* Utama

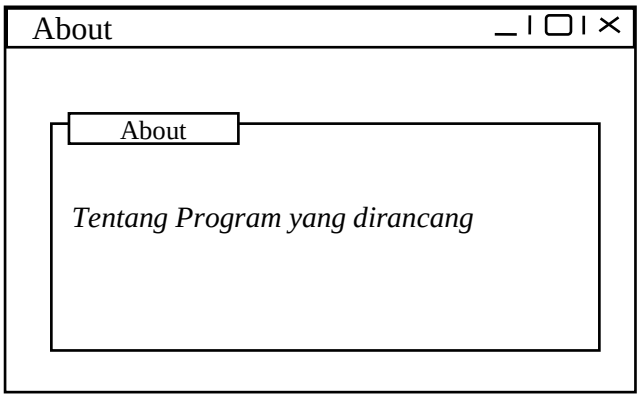
Tampilan *form* utama merupakan tampilan *form* yang fungsi sebagai media proses, didalamnya terdapat *field-field input* dan media tampilan algoritma *RC4*. Adapun rancangan tampilan *form* utama dapat dilihat pada gambar III.9.



Gambar III.9. Tampilan *Form* Utama

III.4.2. Tampilan *Form* Tentang Aplikasi

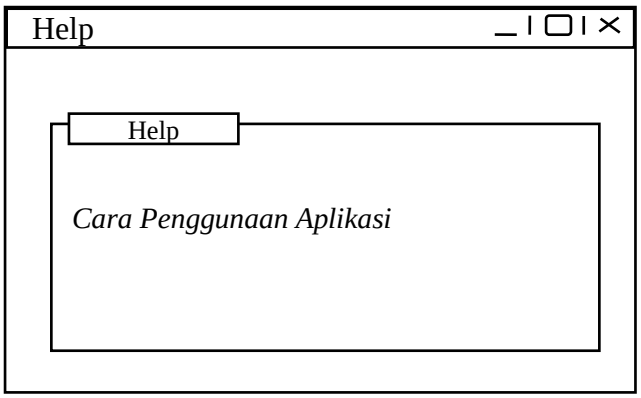
Tampilan *form* tentang aplikasi merupakan *form* yang memberikan sekilas mengenai tujuan perancangan aplikasi yang dibuat, yang dapat dilihat pada gambar III.11 berikut.



Gambar III.10. Tampilan *Form* Tentang Aplikasi

III.4.3. Tampilan *Form* Bantuan

Tampilan *form* bantuan adalah *form* yang berisi penjelasan mengenai cara penggunaan aplikasi, dapat dilihat pada gambar III.12 berikut.



Gambar III.11. Tampilan *Form* Bantuan

Informasi yang terdapat pada *form* bantuan adalah informasi cara menggunakan aplikasi yang telah dirancang dan dapat dijalankan, agar pengguna dapat dengan mudah memahami untuk menjalankannya.