BAB III

ANALISIS DAN PERANCANGAN

III.1. Analisis Sistem

Tahapan analisis dan perancangan ini bertujuan menganalisa kebutuhan pengembangan aplikasi media pembelajaran enkripsi dengan algoritma *Triple* DES. Dalam perancangan yang dilakukan dalam penelitian ini adalah pengamanan dalam pengiriman email. Keamanan adalah aspek yang paling menantang di internet dan jaringan aplikasi. Internet dan jaringan aplikasi yang berkembang sangat cepat, sehingga pentingnya dan nilai data yang dipertukarkan melalui internet atau jenis media lainnya meningkat. Oleh karena itu pencarian solusi terbaik untuk menawarkan perlindungan yang diperlukan terhadap serangan penyusup data bersama dengan menyediakan layanan ini dalam waktu adalah salah satu mata pelajaran yang paling menarik dalam masyarakat keamanan terkait. Kriptografi adalah salah satu kategori utama keamanan komputer yang mengkonversi informasi dari bentuk normal ke dalam bentuk terbaca. Aplikasi yang nantinya dirancang berupa berupa pengiriman *email* sehingga proses yang terjadi dalam pengiriman email terjamin keamanannya.. *Triple* DES (*Triple Data Encryption Standard*) merupakan salah satu algoritma simetris pada kriptografi yang digunakan untuk mengamankan data dengan cara menyandikan data. Proses yang dilakukan dalam penyandian datanya, yaitu proses enkripsi dan proses dekripsi. Desain dan implementasi ini meliputi desain data, deskripsi sistem, desain proses dan implementasi desain dan semua yang diperlukan dalam aplikasi

enkripsi yang dirancang akan dilakukan pada tahapan berikutnya sehingga prerancangan mencapai target dan tujuan yang telah ditentukan sebelumnya.

III.1.1. Analisa Input

Sistem media pembelajaran yang akan di implementasikan dalam aplikasi adalah menggunakan algloritma *Triple* DES. Pada algoritma *Triple* DES, menggunakan 3 kunci yang panjangnya 168 bit atau masing-masing panjangnya 56 bit. Dengan membaca tiap karakter yang dimasukan dari *file* yang dimasukan lalu diproses hingga membentuk suatu tampilan yang tidak dapat dibaca. Dalam proses yang dikembangkan hanya menampilkan proses kerja algoritma *Triple DES* sebagai media pembelajaran agar pengguna dapat mengetahui bagaimana proses dari sebuah algoritma *Triple DES* yang melakukan proses terhadap email yang dikirim.

III.1.2. Analisa Proses

Pembahasan masalah lebih ditekankan pada proses indeks kerja algoritma *Triple* DES yang melakukan pengkodean terhadap pengiriman email. *Triple* DES (*Triple Data Encryption Standard*) merupakan suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama, hanya pada *Triple* DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali. *Triple* DES memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari DES). Pada algoritma *Triple* DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES.

Tahap pertama, plainteks yang diinputkan dioperasikan dengan kunci eksternal pertama (K1) dan melakukan proses enkripsi dengan menggunakan algoritma DES. Sehingga menghasilkan pra-cipherteks pertama. Tahap kedua, pra-cipherteks pertama yang dihasilkan pada tahap pertama, kemudian dioperasikan dengan kunci eksternal kedua (K2) dan melakukan proses enkripsi atau proses dekripsi (tergantung cara pengenkripsian yang digunakan) dengan menggunakan algoritma DES. Sehingga menghasilkan *prs-cipherteks* kedua. Tahap terakhir, pra-cipherteks kedua yang dihasilkan pada tahap kedua, dioperasikan dengan kunci eksternal ketiga (K3) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan *cipher teks* (C). Proses Perhitungan Triple DES 64 bit dari blok input yang dienkripsi adalah subjek pertama dari permutasi yang disebut permutasi dengan inisial IP. Perhatikan tabel III.1 permutasi inisial IP.

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Input yang mengalami permutasi mempunyai bit 58 dari input bit pertamanya, bit 50 sebagai bit kedua dan bit ke 7 sebagai bit terakhir. Blok input yang mengalami permutasi kemudian menjadi input pada perhitungan dan tergantung pada kunci kompleks. Output perhitungan ini, disebut preoutput dan output ini akan diteruskan pada permutasi berikutnya yang merupakan kebalikan dari permutasi inisial.

Perhatikan tabel II.2 kebalikan dari permutasi inisial IP yaitu IP –1 .

IP –1

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Output dari algoritma di atas mempunyai bit 40 dari blok preoutput sebagai bit pertamanya, bit 8 sebagai bit kedua sampai bit 25 sebagai bit terakhir. Perhitungan yang menggunakan blok input dikenakan permutasi sebagai inputnya untuk menghasilkan blok preoutput. Tetapi untuk pertukaran blok akhir, dari 16 iterasi dari kalkulasi yang dijelaskan di bawah ini merupakan fungsi cipher f yang

mengoperasikan 2 blok, yaitu salah satu dari 32 bit dan salah satu dari 48 bit. Kalkulasi tersebut akan menghasilkan blok

sepanjang 32 bit. 64 bit dari blok input terdiri dari 32 bit blok L dan diikuti oleh 32 bit blok R. Input blok ini didefinisikan sebagai LR. K menjadi input blok dari 48 bit yang dipilih dari 64 bit kunci. Kemudian output L'R' dari iterasi dengan input LR menghasilkan persamaan berikutini :

L' = R

R' = L(+)f(R,K)

Dimana (+) merupakan penambahan bit demi bit kemudian dibagi 2. Input iterasi pertama dari perhitungan tadi adalah blok input yang mengalami permutasi. L'R' adalah output dari iterasi ke 16, kemudian R'L' adalah blok preoutput. Pada masing-masing iterasi sebuah blok yang berbeda, K merupakan kunci bit yang dipilih dari 64 kunci yang ditunjukkan oleh KEY. Dengan notasi di atas, kita bisa menjelaskan iterasi menjadi lebih rinci. KS menjadi sebuah fungsi yang menggunakan bilangan bulat n dengan jangkauan dari bilangan 1 sampai bilangan 16 dan blok 64 bit KEY sebagai input serta hasilnya sebagai output blok 48 bit Kn, di mana bisa dilihat pada persamaan berikut ini :

Kn = KS (n, KEY)

Dengan Kn ditentukan oleh bit dalam posisi bit yang berbeda dengan KEY.

KS disebut kunci schedule karena blok K digunakan dalam iterasi ke-n (persamaan 1) dan blok Kn ditentukan oleh persamaan 2. Karena sebelumnya blok input dipermutasikan dengan LR, akhirnya L0 dan R0 berubah menjadi L dan R, sedangkan Ln dan Rn berubah menjadi L' dan R' (persamaan 1). Selanjutnya L

dan R berubah menjadi Ln-1 dan Rn-1. K adalah Kn, yaitu ketika n dalam jangkauan bilangan 1 sampai bilangan 16. Perhatikan persamaan berikut ini :

Ln = Rn-1

Rn n = Ln-1 (+) f(Rn-1, Kn)

Blok *preoutput* dari persamaan di atas adalah R16L16.

Adapun algoritmanya Enkripsi adalah sebagai berikut :

- Ambillah blok data sebanyak 64 bit tersebut. Apabila kita dalam mengambil blok data kurang dari 64 bit, maka perlu adanya penambahan supaya dalam penggunaannya sesuai dengan jumlah datanya.
- Bentuklah permutasi awal (Initial Permutation, IP) pada blok data 64 bit tadi dengan memperhatikan permutasi berikut ini. Initial Permutation

  58 50 42 34 26 18 10 2

  60 52 44 36 28 20 12 4

  62 54 46 38 30 22 14 6

  64 56 48 40 32 24 16 8

  57 49 41 33 25 17 9 1

  59 51 43 35 27 19 11 3

  61 53 45 37 29 21 13 5

  63 55 47 39 31 23 15 7

- Blok data tersebut dan dibagi menjadi 2 bagian, yaitu 32 bit pertama disebut L[0] dan 32 bit kedua disebut R[0].
- Ke 16 sub kunci dioperasikan dengan blok data, dimulai dengan j=1 dan terbagi menjadi cara-cara berikut ini :

- R[j – 1] dikembangkan menjadi 48 bit menurut fungsi pemilihan ekspansi berikut :

Expansion (E)

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

- Kemudian langkah berikutnya adalah : E (R[j – 1]) di XOR dengan K[j].
- Hasil E(R[j -1]) XOR K[j] dipecah menjadi delapan blok 6-bit. Kelompok bit 1 – 6 disebut B[1], bit 7 – 12 disebut B[2], dan seterusnya bit 43-48 disebut B[8].
- Jumlah bit dikurangi dengan penukaran nilai-nilai yang ada dalam table S untuk setiap B[j]. Dimulai dengan j = 1, setiap nilai dalam tabel S memiliki 4 bit.

Adapun langkah-langkah dalam tahap ini adalah sebagai berikut :

1. Ambil bit ke 1 dan ke 6 dari B[j] bersama-sama menjadi nilai 2 bit, misalkan m, yang menunjukkan baris dalam tabel S[j]. Ambil bit ke 2 hingga 5 dari B[j] sebagai nilai 4 bit, misalkan n, yang menunjukkan kolom dalam S[j].
2. Hasil proses ini adalah S[j][m][n] untuk setiap B[j] sehingga iterasi yang diperlukan sebanyak 8 kali. Hasil ini sering disebut juga substitution box. Nantinya akan ada substitution box sebanyak 8 buah iterasi. Perhatikan masing-masing tabel III.3 berikut ini.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Substitution Box 1 (S[1])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Substitution Box 2 (S[2])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	2	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Substitution Box 3 (S[3])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Substitution Box 4 (S[4])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	14	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Substitution Box 5 (S[5])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	15	10	11	14	1	7	6	0	8	13	12

Substitution Box 6 (S[6])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Substitution Box 7 (S[7])

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Substitution Box 8 (S[8])

Permutasi dilakukan kembali pada kombinasi hasil substitusi di atas S[1][m1][n1] sampai dengan S[8][m2][n2] dengan memperhatikan keterangan berikut ini :

Permutation P

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

- Hasil permutasi kemudian di XOR dengan L[j-1], selanjutnya hasil ini menjadi R[j]. Perhatikan rumus berikut ini :

R[i]=L[i -1] XOR P(S[1](B[1]) …. S[8](B[8])

B[j] merupakan blok 6 bit hasil kombinasi R(R[i – 1]) XOR

K[i]. Fungsi ini bisa ditulis sebagai berikut :

R[i]=L[i -1] XOR f (R[i -1], K[i])

L[i]=R[i-1]

- Ulangi kembali ke langkah paling atas hingga K[16]
- Permutasi akhir dilakukan kembali dengan tabel permutasi yang merupakan invers dari permutasi awal. Perhatikan permutasi berikut ini :

Final Permutation (IP**-1)

40 8 48 16 56 24 64 32

39 7 47 15 55 23 63 31

38 6 46 14 54 22 62 30

37 5 45 13 53 21 61 29

36 4 44 12 52 20 60 28

35 3 43 11 51 19 59 27

34 2 42 10 50 18 58 26

33 41 9 49 17 57 25

III.1.3. Analisa Output

Pada hasil analisa *input* dan analisa proses pada akhirnya akan menghasilkan *output*/hasil keluaran yang nantianya akan diterima pengguna. Dari setiap pengiriman email yang dilakukan dan akan dienkripsi dengan menggunakan algoritma *Triple* DES dan diubah kedalam bentuk yang tidak dapat

dikenali dan hanya akan dapat dilihat jika hasil yang sudah dieknripsi tersebut dikembalikan kebentuk semula dengan proses dekripsi pada aplikasi dengan mengisi *password* yang digunakan pada tahapan pengiriman sebelumnya.

III.2. Strategi Pemecahan Masalah

Untuk membangun aplikasi enkripsi dan dekripsi pada pengiriman email dengan penerapan algoritma *Triple* DES Beberapa strategi pemecahan masalah dalam perancangan adalah sebagai berikut :

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

2 Dalam proses enkripsi dan dekripsi pada uji coba hanya dilakukan pada tiap-tiap pengiriman email.

3 *Interface* memunggunakan tampilan yang disajikan dalam membaca aplikasi yagn telah dienkripsi maupun tampilan dalam menginputkan email yang akan dikirim.

III.3. Perancangan Sistem

Perancangan implementasi aplikasi ini 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

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

1. Mengimplementasikan penggunaan bahasa pemrograman *java* dalam membuat aplikasi media pembelajaran algoritma *Triple* DES.
2. Aplikasi dapat menggambarkan penerapan algoritma *Triple* DES pada pengiriman email.
3. *Input* berupa *file* yang akan dikirim melalui email dan *output* berupa *file* telah diproses dengan algoritma *Triple* DES.

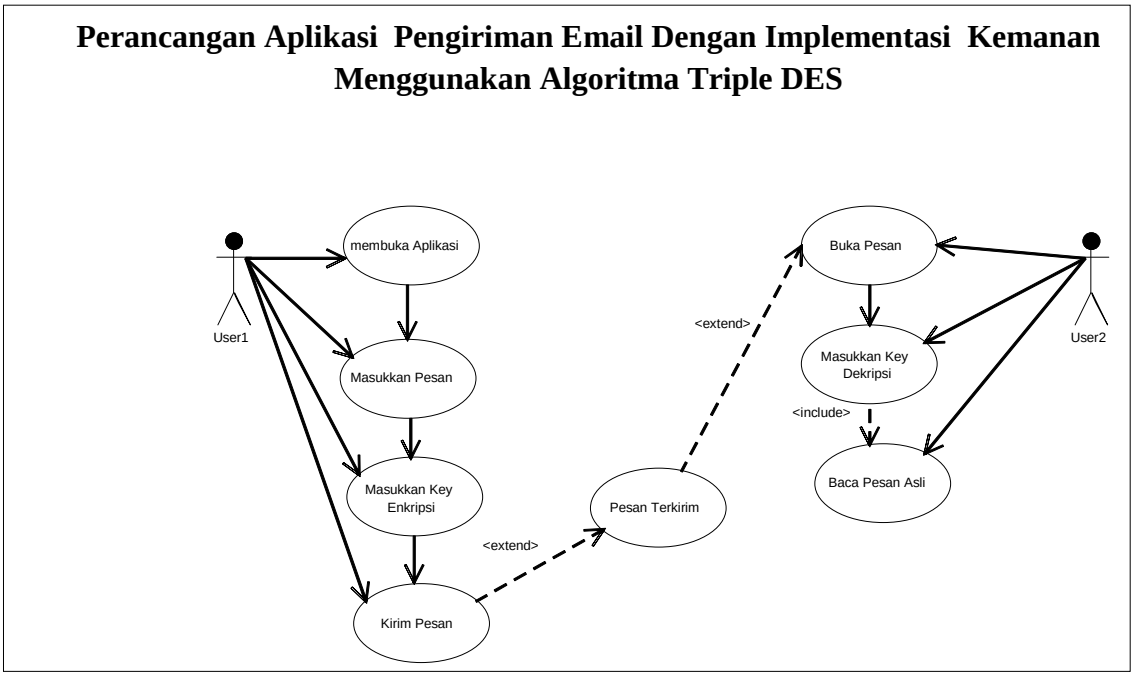
III.3.2. Analisa Kebutuhan Nonfungsional

Untuk mencapai tujuan yang tekah ditetapkan sebelumnya dalam perancangan aplikasi pengamanan dalam pengiriman email dengan menggunakan algoritma *Triple* DES ini, beberapa perangkat yang penulis gunakan agar aplikasi berjalan baik, yaitu sebagai berikut :

1. Perangkat Keras (*Hardware*)
    a. Komputer yang setara dengan *Intel pentium Dual Core*.
    b. *Mouse*, *keyboard*, dan *Monitor.*
2. Perangkat Lunak (*Software*)
    a. *Operating System Windows Seven*.
    b. *Java* sebagai bahasa program yang digunakan serta *Netbean* sebagi bentuk pengkodean.

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.

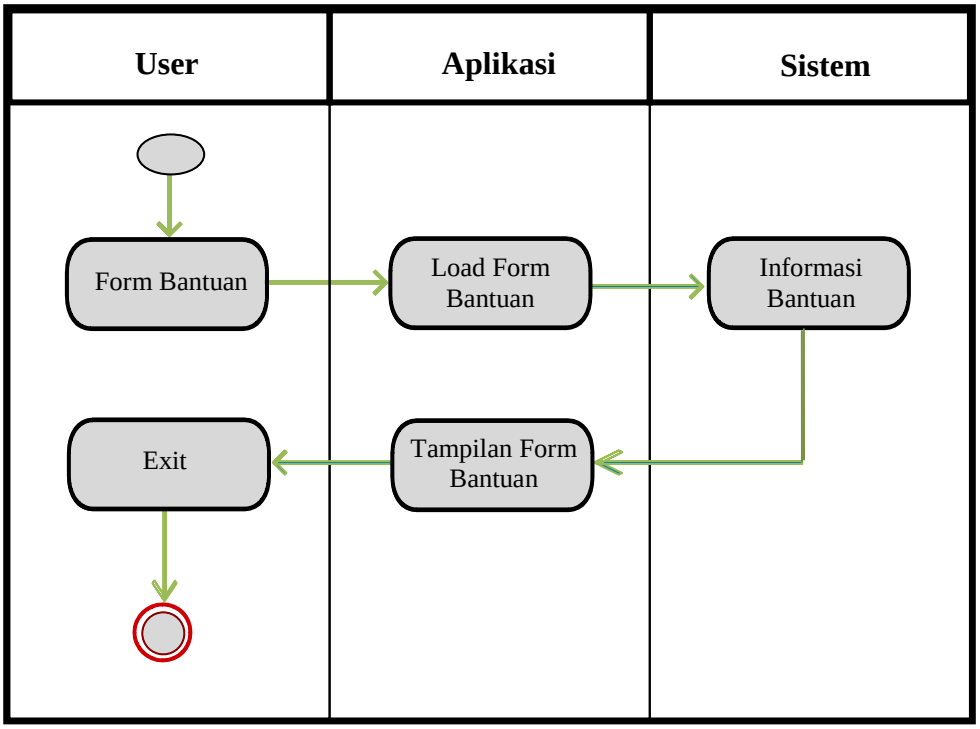


Gambar III.1. *Use Case Diagram* Pengguna Aplikasi

Kegiatan aktor atau pengguna pada aplikasi enkripsi *file* yang dikirm, pengguna dapat memilih melakukan pengiriman email dan user juga melakukan q penerimaan *email*.

III.3.4. *Activity Diagram* Proses *Form* Help

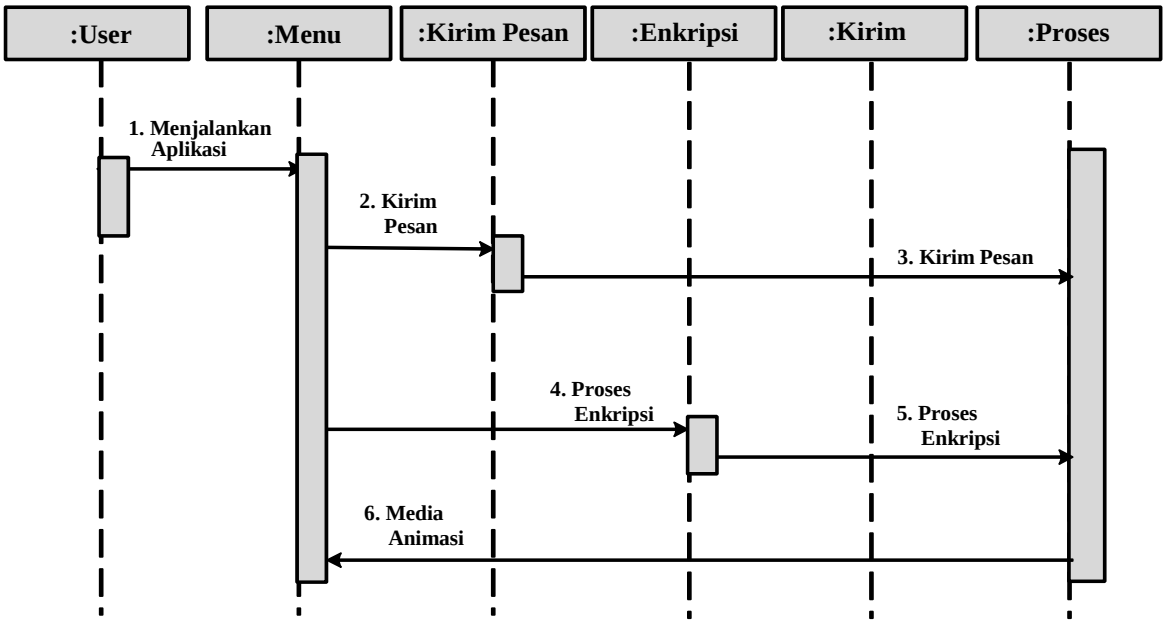
Pada *Activity diagram* ini menjelaskan tentang berbagai aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir aktivitas berawal. Adapun rancangan diagram aktivitas untuk menampilkan *form help* atau bantuan dari aplikasi yang dirancang terdapat pada gambar III.2. berikut ini :



Gambar III.2. *Activity Diagram Form* Bantuan

III.3.5. *Sequence* Diagram Proses Pengiriman *Email*

Pada *Sequence* diagram adalah pengiriman atau 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.3 berikut.

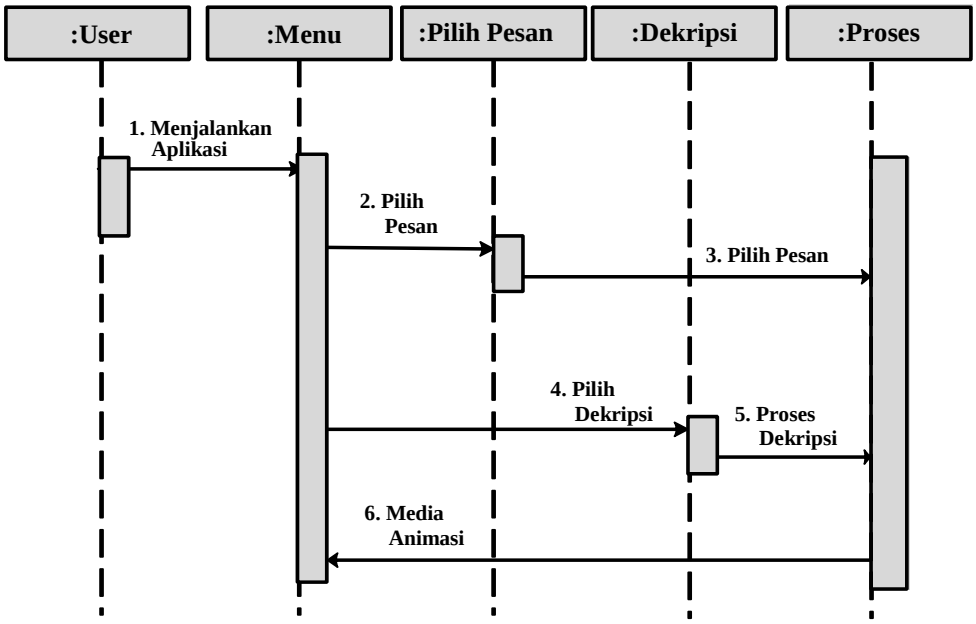


Gambar III.3. *Sequence Diagram* Proses Pengiriman Email

Untuk lebih jelasnya gambar *sequence* diagram proses pengiriman email yang dilakukan oleh user dengan enkripsi yang terdapat pada gambar III.4 diatas menjelaskan bahwa *user* atau pengguna memulai menjalankan aplikasi sehingga terdapat menu utama dari sistem dengan lanjut berinteraksi melalui kirim pesan yang ada pada menu utama. Setelah memilih kirim pesan ditentukan oleh pengguna kembali pada menu utama.

III.3.6. *Sequence* Diagram Proses Penerimaan *Email*

Sequence diagram proses penerimaan *email* ini menggambarkan kegiatan dari skenario penggunaan aplikasi, *sequence* diagram memilih proses yang terjadi dengan memilih proses enkripsi untuk membaca atau mengembalikan *file* yang diterima yang telah dienkripsi pada pengiriman sebelumnya.

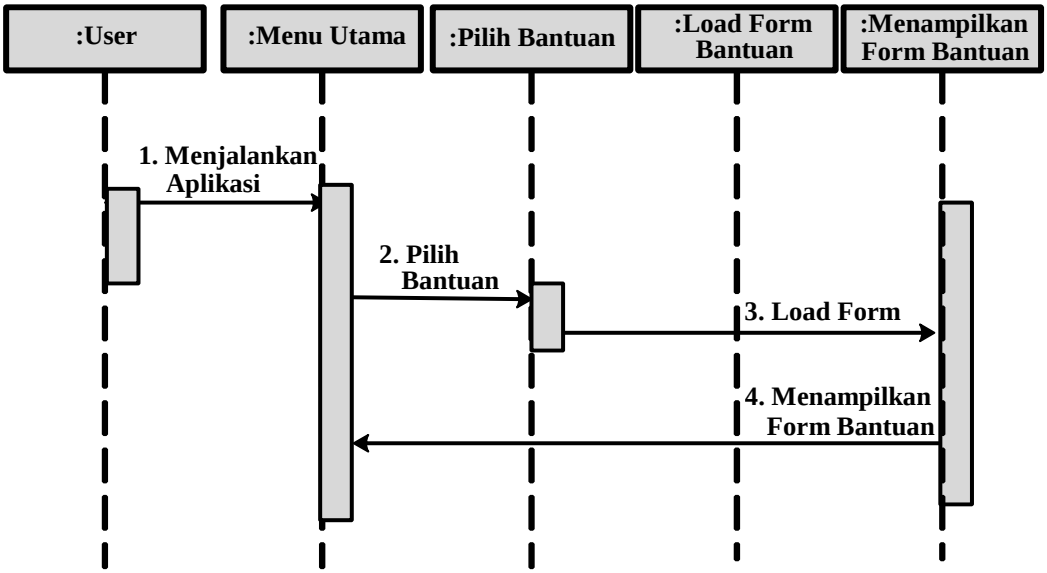


Gambar III.4. *Sequence Diagram* Proses Penerimaan Email

Untuk lebih jelasnya gambar *sequence* diagram proses pengembalian pesan email yang diterima akan dikembalikan pada bentuk asli dari file dilakukan yang terdapat pada gambar III.4 diatas menjelaskan bahwa *user* atau pengguna memulai menjalankan aplikasi sehingga terdapat menu utama dari sistem dengan lanjut berinteraksi dengan memilih proses dan mendekripsi pesan email.

III.3.7. *Sequence* Diagram Proses *Form* Bantuan

Sequence diagram menggambarkan kegiatan dari skenario penggunaan aplikasi, *squence* diagram memilih proses *form* help pada media pembelajaran dapat dilihat pada gambar III.5 berikut.

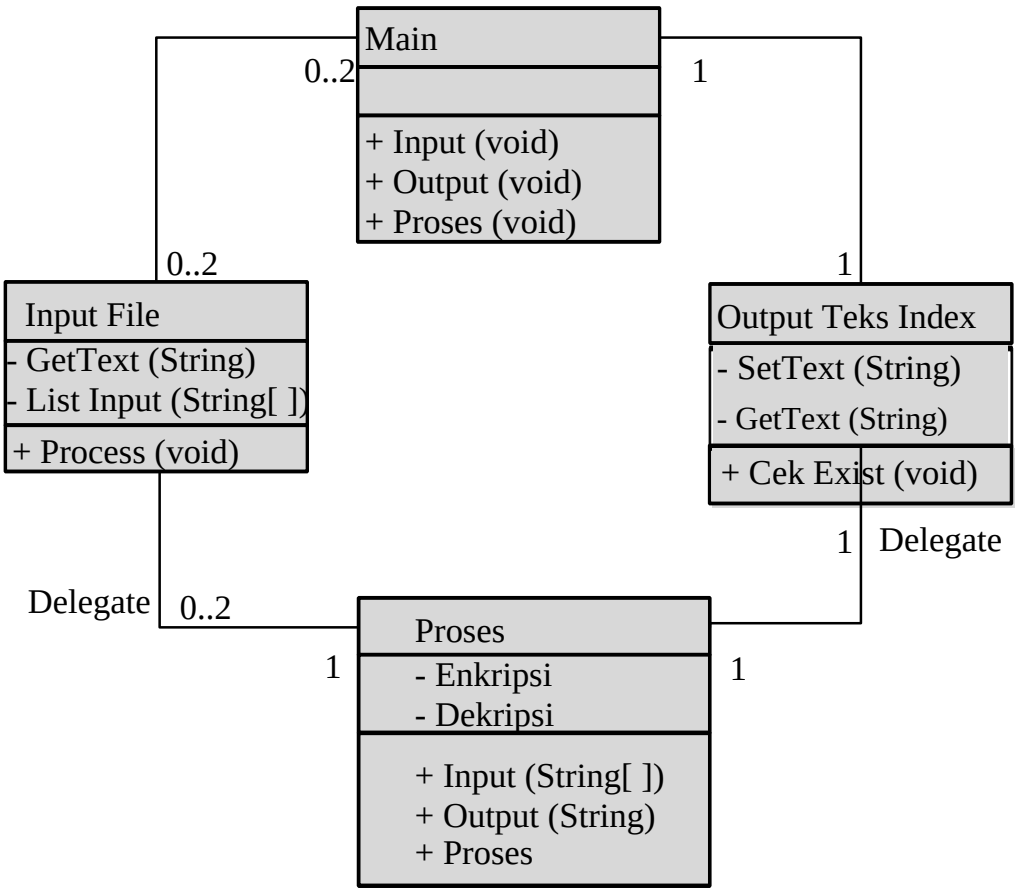


Gambar III.5. *Sequence Diagram* Proses *Form* Bantuan

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

III.3.8. *Class* Diagram

Pada *Class* diagram perancangan aplikasi ini, dapat dilihat pada gambar III.6 berikut.



Gambar III.6. *Class* Diagram Sistem Perancangan Aplikasi

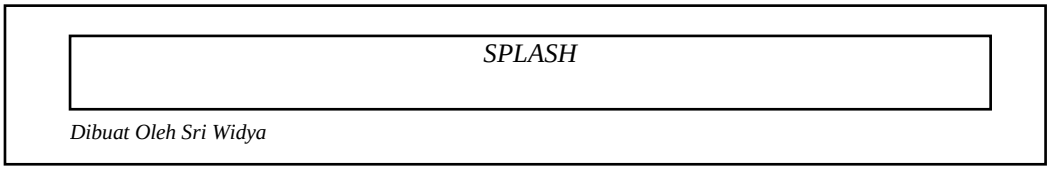
Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* diagram juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana 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 Splash*

Rancangan *form splash* ini merupakan tampilan pembuka saat menjalankan aplikasi, yang dapat dilihat pada gambar III.7.

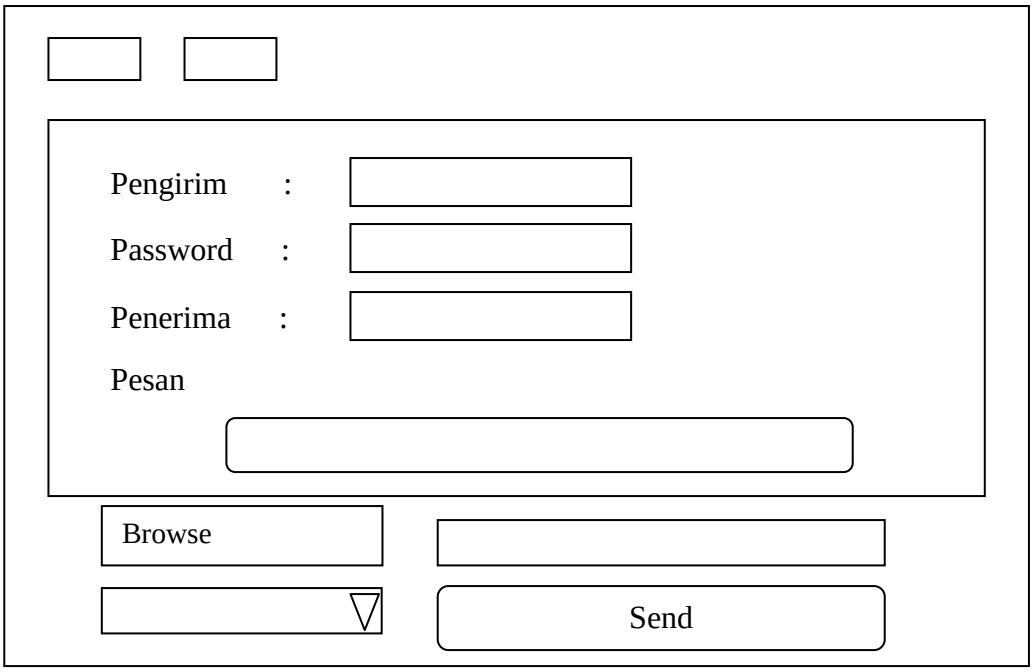


Gambar III.7. Tampilan *Form Splash*

form ini menampilkan pembuka aplikasi sebelum *form* utama, *form* ini akan membuka *form* utama, tampil pada layar dan akan ditutup setelahnya, *form* ini hanya tampil ketika program aplikasi baru dijalankan.

III.4.2. Tampilan *Form* Utama

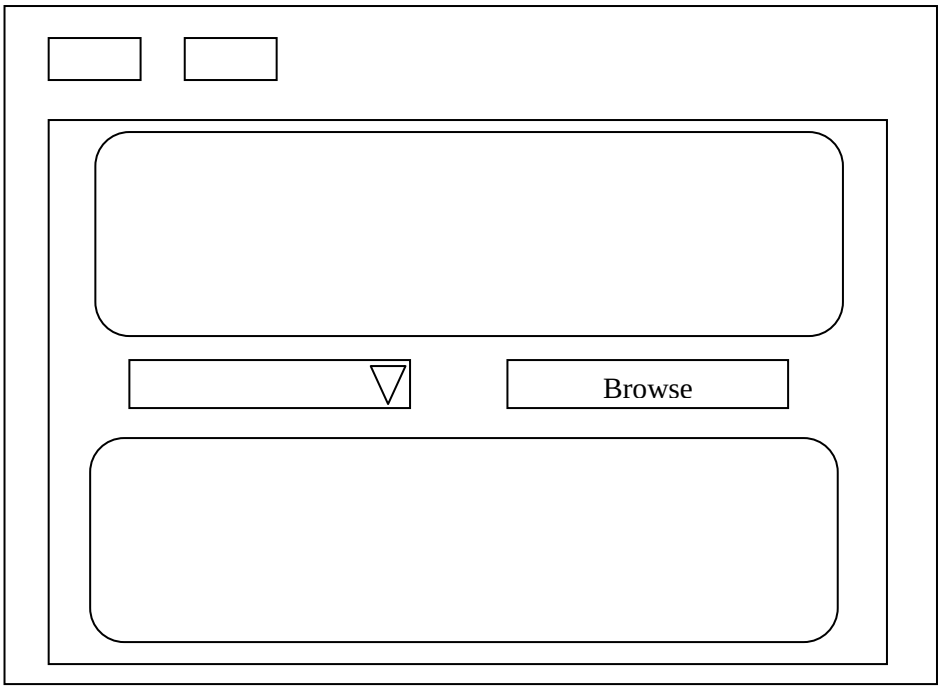
Tampilan *form* utama merupakan tampilan *form* yang fungsi sebagai media proses, didalamnya terdapat *field-field input* dan media tampilan algoritma *Triple DES*. Adapun rancangan tampilan *form* utama dapat dilihat pada gambar III.8.



Gambar III.8. Tampilan *Form* Utama

III.4.3. Tampilan *Form* Pengiriman Email Teks

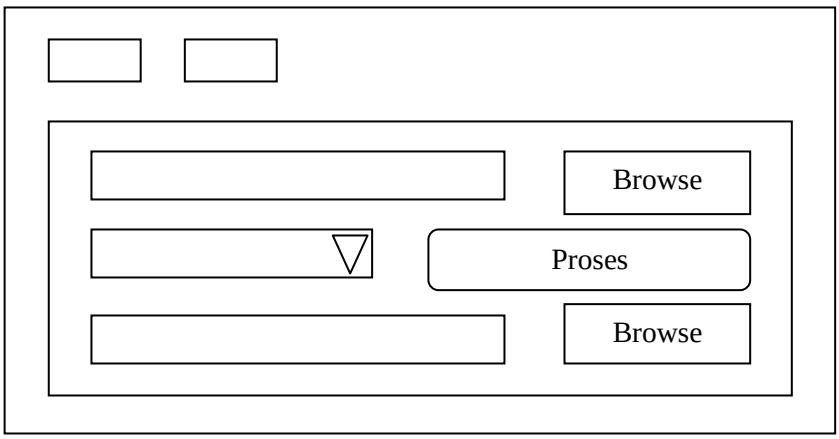
Tampilan *form* tentang aplikasi merupakan *form* yang memberikan tampilan dalam mengenkripsi *file* teks yang dikirim pada *email*, yang dapat dilihat pada gambar III.9 berikut.



Gambar III.9. Tampilan *Form* Pengiriman Teks

III.4.4. Tampilan *Form* Pengiriman *File*

Tampilan *form* bantuan adalah *form* pengiriman *file* yang berisi penjelasan mengenai cara mengenkripsi *file*, dapat dilihat pada gambar III.10 berikut.



Gambar III.10. Tampilan *Form* Pengiriman *File*

III.4.5. Tampilan *Form* Bantuan

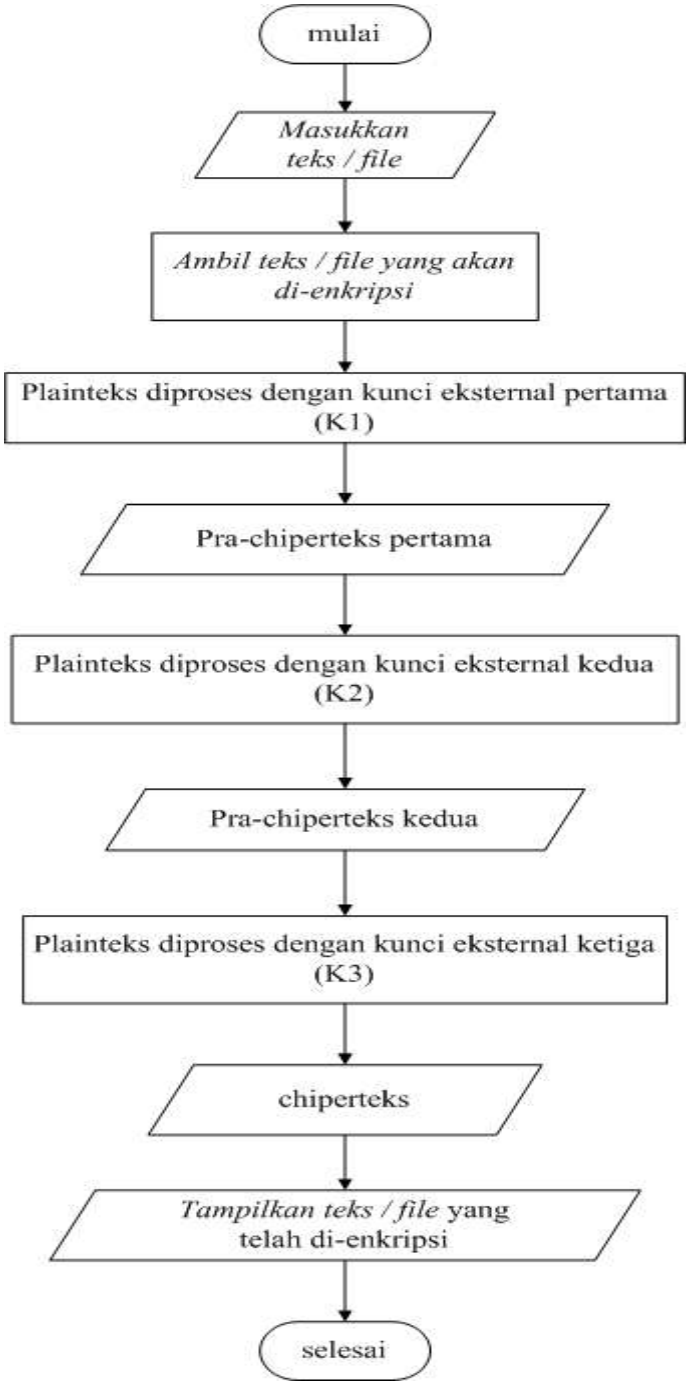
Tampilan *form* bantuan adalah *form* pengiriman *file* yang berisi penjelasan mengenai cara mengenkripsi *file*, dapat dilihat pada gambar III.11 berikut.

Gambar III.11. Tampilan *Form* Pengiriman *File*

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.

III.4.6. *Flowchart* Diagram

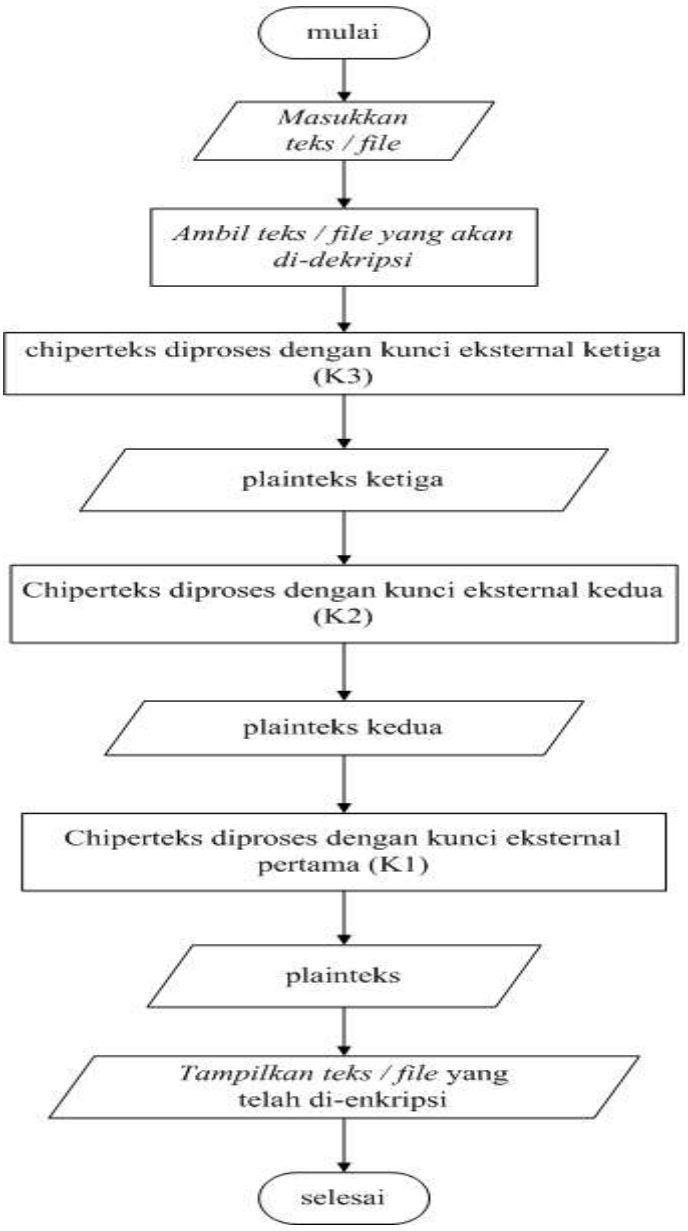
Dalam suatu perancangan perangkat lunak diperlukan suatu metode yang digunakan untuk pemecahan masalah terhadap rancangan. Penulis menggunakan *flowchart* untuk memudahkan pemahaman. Tujuan utama penggunaan *flowchart* adalah untuk merepresentasikan simbol-simbol yang standar sehingga memudahkan penulis untuk merancang perangkat lunak. Perangkat lunak yang dirancang dibagi atas dua *flowchart* proses kerja secara keseluruhan yang dapat dilihat pada Gambar III.12 sebagai proses kerja enkripsi dan dekripsi :



Gambar III.12. *Flowchart* Diagram Enkripsi

Adapun penjelasan dari *flowchart* Gambar III.12 di atas adalah Enkripsi teks /*file* plainteks *(P)* mula-mula diproses dengan kunci eksternal pertama

(K1),setelah mendapatkan hasil dari K1 lanjut keproses enkripsi dengan kunci K2 sampai ke penghitungan K3 dan hasil enkripsi akhir adalah chiperteks *(C).*



Gambar III.13. *Flowchart* Diagram Dekripsi

Pada *flowchart* Gambar III.13 dijelaskan bahwa untuk mendapatkan hasil dari proses dekripsi adalah kebalikan dari enkripsi yaitu Mula-mula kunci K3

digunakan untuk mendekripsi chiperteks *(C),* lalu hasilnya dienkripsi lagi dengan kunci K2 kemudian didekripsi lagi dengan kunci K1 dan hasil dekripsi terakhir adalah pesan semula plainteks *(P).*