BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Secara umum sistem yang akan dibangun pada penelitian adalah aplikasi pengamanan data dokumen untuk mengingatkan tingkat keamanan dan kerahassiaan sebuah dokumen agar terhindar dari gangguan orang yang lain yang tidak berhak. Ada banyak cara untuk melakukan pengamanan dokumen beberapa diantaranya yaitu dengan memberikan kata sandi untuk dokumen atau dengan mengubah konten dari dokumen misalnya mengubah teks yang ada didokumen menjadi sulit dimengerti atau tidak terbaca.

3.2. Hasil Analisis Sistem

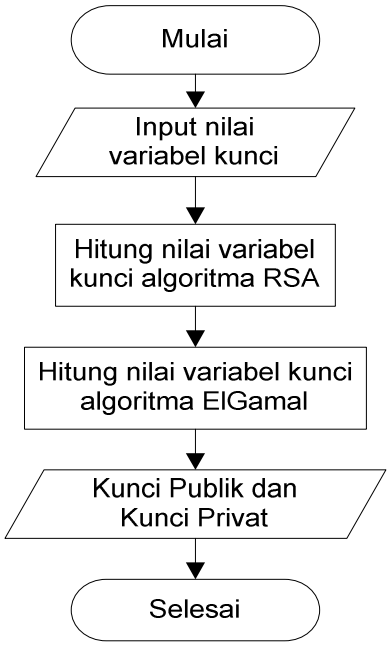
Adapun hasil analisis yang diperoleh dari penelitian ini yaitu

1. Melakukan pengamanan dokumen dengan mengubah isi dari dokumen sehingga dokumen tersebut tidak dapat dimengerti oleh orang lain yang tidak berhak.
2. Untuk dapat membaca isi dari dokumen tersebut terlebih dahulu dilakukan dekripsi untuk mengembalikan isi dari dokumen yang telah dirubah menjadi seperti semula.
3. Algoritma yang digunakan untuk mengubah isi dokumen dan mengembalikannya yaitu algoritma kriptografi asimetris RSA dan algoritma asimetris ELGAMAL. Untuk kelebihan dan kekurangan algoritma RSA dapat dilihat pada poin 2.2.2. Sedangkan kelebihan algoritma ElGamal adalah meskipun mempunyai nilai plaintext yang sama namun mempunyai hasil enkripsi (ciphertext) yang berbeda (dengan kepastian yang dekat) namun kekurangan dari algoritma ini adalah hasil enkripsinya mempunyai panjang 2 kali lipat dari data aslinya.

Langkah-langkah yang dilakukan dalam pengamanan dokumen adalah pembangkitan kunci privat dan kunci publik, melakukan proses enkripsi dokumen dan melakukan proses dekripsi dokumen.

1. Membangkitkan kunci privat dan kunci publik

   Langkah-langkah dalam pembangkitan kunci dapat dilihat pada gambar 3.1

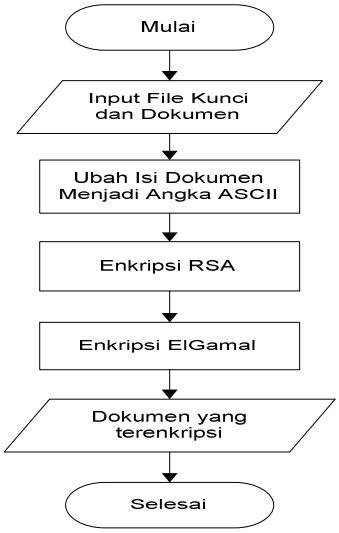


Gambar 3.1 Flowchart Langkah Pembangkitan Kunci

Secara umum proses untuk melakukan pembentukan kunci publik dan kunci privat telah digambarkan pada *flowchart* (gambar 3.1) dengan penjelasan sebagai berikut : memasukan nilai-nilai variabel kunci yang dibutuhkan dalam proses pembangkitan kunci. Kemudian dilakukan perhitungan dari variabel-variabel pembentukan kunci RSA dan dilanjutkan dengan melakukan perhitungan dari variabel-variabel pembentukan kunci ElGamal sehingga diperoleh kunci publik dan kunci privat.

2. Langkah melakukan enkripsi dokumen

   Langkah-langkah dalam melakukan enkripsi dokumen dijelaskan dalam diagram alir pada gambar 3.2



Gambar 3.2 Flowchart Enkripsi Dokumen

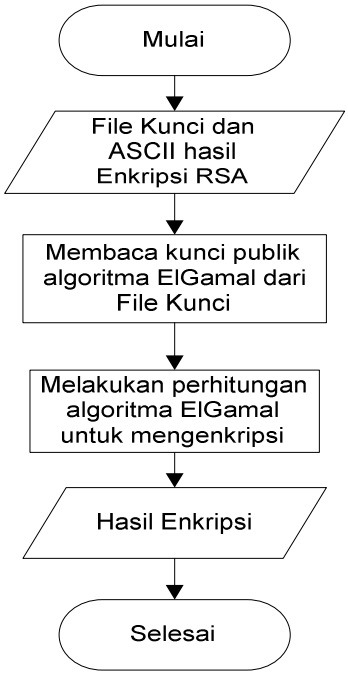
Yang pertama dilakukan dalam proses enkripsi ini adalah memasukkan dokumen dan file kunci yang telah dibuat sebelumnya, kemudian mengubah isi dokumen menjadi angka kode ASCII sesuai dengan masing-masing karakternya. Kemudian dilakukan proses enkripsi menggunakan algoritma *RSA* yang kemudian hasil yang diperoleh pada proses enkripsi RSA dienkripsi ulang dengan menggunakan algoritma *ElGamal.* Sehingga memperoleh dokumen yang telah terenkripsi hasil dari enkripsi kedua metode tersebut.

Adapun proses-proses yang terjadi dalam proses enkripsi RSA pada gambar 3.2 dijelaskan pada diagram alir (Gambar 3.3) berikut

Gambar 3.3 Flowchart Proses Enkripsi Dokumen Dengan RSA

Dalam proses enkripsi dokumen dengan algoritma RSA ini dilakukan pembacaan/pencarian kunci publik algoritma RSA yang tersimpan dalam file kunci yang kemudian kunci publik tersebut digunakan dalam perhitungan algoritma RSA untuk mengenkripsi data yang menghasilkan data yang terenkripsi atau data yang sudah tidak sama dengan data semula.

Adapun proses-proses yang terjadi dalam proses enkripsi ElGamal pada gambar 3.2 dapat dilihat pada diagram alir (Gambar 3.4) berikut

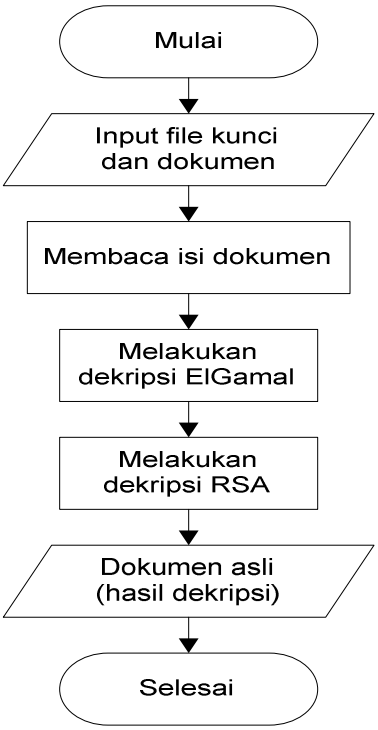


Gambar 3.4 Flowchart Proses Enkripsi dokumen dengan ElGamal

Dalam proses enkripsi dokumen dengan algoritma ElGamal ini dilakukan dengan melakukan pembacaan atau pencarian kunci publik algoritma ElGamal dalam file kunci yang kemudian digunakan dalam melakukan perhitungan enkripsi menggunakan algoritma ElGamal terhadap hasil dari pengenkripsian dokumen menggunakan algoritma RSA. Dan hasil yang diperoleh adalah dokumen atau data yang terenkripsi ulang oleh algoritma ElGamal.

3. Langkah melakukan dekripsi dokumen

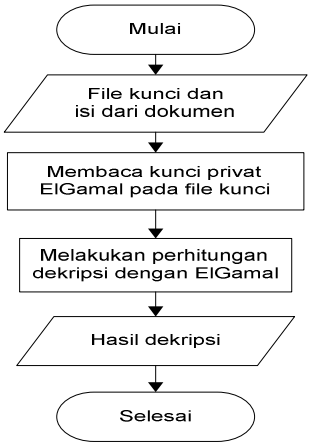
Setelah dokumen tersebut dienkripsi maka isi dari dokumen tersebut menjadi tidak terbaca atau sulit dibaca maka dilakukan proses dekripsi atau pengembalian data atau dokumen menjadi data atau dokumen yang sebenarnya sehingga data atau dokumen tersebut dapat dibaca kembali. Adapun diagram alir untuk proses dekripsi atau pengembalian data menjadi yang sebenarnya dapat dilihat pada gambar 3.5 berikut



Gambar 3.5 Flowchart Proses Dekripsi Dokumen

Dalam proses dekripsi ini, terlebih dahulu menginputkan dokumen yang telah dienkripsi dan file kunci. Selanjutnya dilakukan proses dekripsi menggunakan algoritma *ElGamal* dan hasil pendekripsian tersebut kemudian didekripsi kembali menggunakan algoritma *RSA* sehingga diperoleh isi sebenarnya dari dokumen yang terenkripsi tersebut.

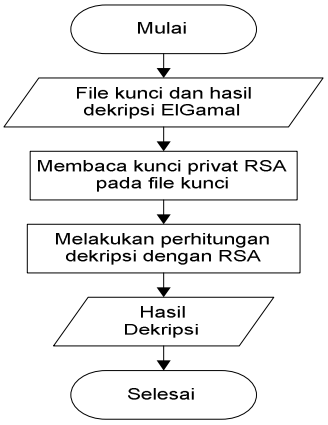
Adapun proses-proses yang terjadi dalam proses dekripsi ElGamal pada gambar 3.5 dapat dilihat pada diagram alir (Gambar 3.6) berikut



Gambar 3.6 Flowchart Proses Dekripsi Dokumen dengan ElGamal

Dalam proses dekripsi dokumen dengan algoritma ElGamal ini dilakukan pembacaan atau pencarian kunci privat algoritma ElGamal yang terdapat pada file kunci yang kemudian digunakan untuk melakukan perhitungan dekripsi terhadap isi dari dokumen yang terenkripsi yang nantinya akan menghasilkan data yang telah terdekripsi dengan algoritma ElGamal.

Adapun proses-proses yang terjadi dalam proses dekripsi RSA pada gambar 3.6 dapat dilihat pada diagram alir (Gambar 3.7) berikut



Gambar 3.7 Flowchart Proses Dekripsi Dokumen Dengan RSA

Dalam proses dekripsi dokumen dengan algoritma RSA ini dilakukan pembacaan atau pencarian kunci privat algoritma RSA yang terdapat dalam file kunci yang kemudian digunakan untuk melakukan perhitungan dekripsi menggunakan algoritma RSA terhadap hasil dari dekripsi yang menggunakan

algoritma ElGamal. Sehingga akan menghasilkan dokumen atau data yang sebenarnya.

3.3. Representasi Data

Sebagai contoh, pengguna mempunyai sebuah dokumen dan pengguna ingin mengenkripsi dokumen tersebut agar tidak diketahui oleh orang lain. Dan melakukan proses dekripsi untuk mengetahui kembali isi dari dokumen tersebut. Proses-proses yang dilakukan adalah proses pembentukan/pembangkitan kunci yang kemudian dilakukan proses pengenkripsian dokumen dan pendekripsian dokumen.

3.3.1. Proses Pembangkitan Kunci

Langkah-langkah dalam pembangkitan kunci algoritma *RSA* telah dijelaskan pada poin (2.2.3. Pembangkitan Kunci). Misal jika pengguna memilih bilangan prima $p = 17$ dan bilangan prima $q = 11$ maka diperoleh $n = 187$ yang didapat dari perhitungan $n = p \times q$ dan $\phi(n) = 160$ yang diperoleh dari perhitungan $\phi(n) = (p - 1) \times (q - 1)$. Kemudian dilakukan perhitungan untuk menentukan kunci publik dan kunci privat dengan persamaan $e \times d = 1 \bmod \phi(n)$. Jika pengguna memilih nilai e = 7 maka diperoleh d = 23. Sehingga diperoleh kunci publik {e, n} yaitu {7, 187} dan kunci privat {d, n} yaitu {23, 187} untuk kunci algoritma *RSA*.

Sedangkan langkah-langkah pembangkitan kunci algoritma *ElGamal* telah dijelaskan pada poin (2.3.2 Pembentukan Kunci). Misal jika pengguna memilih bilangan prima besar $p = 191$, nilai $g = 3$ dan nilai $x = 5$ kemudian hitung nilai y dengan pesamaan 2.3 sehingga diperoleh $y = 52$. Dari hasil perhitungan tersebut diperoleh kunci publik {p, g, y} yaitu {191, 3, 52} dan kunci privat {p, x} yaitu {191, 5}. Hasil dari pembangkitan kunci yang telah dilakukan lebih jelasnya dapat lihat pada tabel 3.1 berikut

Tabel 3.1 Hasil Pembangkitan Kunci

JENIS KUNCI	ALGORITMA RSA	ALGORITMA ELGAMAL
Kunci Publik	{7, 187}	{191, 3, 52}
Kunci Privat	{23, 187}	{191, 5}

3.3.2. Proses Enkripsi Dokumen

Proses enkripsi dapat dilakukan dengan menggunakan kunci publik. Sebagai contohnya pengguna menggunakan kunci publik yang telah dibuat (Tabel 3.1). Pertama yang dilakukan dalam proses enkripsi adalah mengubah isi dari dokumen menjadi blok-blok yang kemudian setiap blok diubah menjadi angka-angka kode ASCII. Misal jika isi dokumen terdapat karakter "Abc" maka isi dokumen tersebut dibagi menjadi 3 blok yaitu "A", "b", dan "c" yang kemudian diubah menjadi angka ASCII. Untuk Karakter "A" mempunyai kode ASCII "65" sedangkan karakter "b" kode ASCII-nya adalah "98" dan untuk "c" adalah "99". Untuk mengetahui kode-kode ASCII selengkapnya dapat dilihat pada gambar 2.6 dan gambar 2.7

Misalnya jika isi dari dokumen adalah "Dokumen Ini Sangat Penting" maka akan diubah menjadi blok-blok dan kemudian karakter tiap-tiap blok tersebut diubah menjadi angka-angka ASCII. Angka-angka ASCII tersebut kemudian disebut sebagai Plainteks (M). Hasilnya akan tampak seperti pada tabel 3.2 berikut.

Tabel 3.2 Hasil Mengubah Karakter Menjadi Angka ASCII

Karakter	ASCII (M)	Karakter	ASCII (M)
D	68	P	80
o	111	e	101
k	107	n	110
u	117	t	116
m	109	i	105
e	101	n	110
n	110	g	103

<spasi>	32		
I	73		
n	110		
i	105		
<spasi>	32		

Selanjutnya melakukan perhitungan enkripsi menggunakan algoritma *RSA* dan kunci publik algoritma RSA. Kunci publik algoritma *RSA* yang telah dibuat adalah {7, 187} (tabel 3.1) dan dilakukan perhitungan untuk mencari ciphertext C menggunakan persamaan 2.1 untuk masing-masing blok. Blok pertama adalah karakter "D" dengan kode ASCII "68" sehingga nilai M = 68

$C = M^e \bmod n$

$= 68^7 \bmod 187$

$= ((68 \bmod 187)^4 \times (68 \bmod 187)^3) \bmod 187$

$= ((21381376 \bmod 187) \times (314432 \bmod 187)) \bmod 187$

$= (170 \times 85) \bmod 187$

$= 14450 \bmod 187 =$ **51**

Jadi hasil enkripsi dengan algortima *RSA* untuk kode ASCII "68" adalah 51. Untuk hasil lengkapnya dapat dilihat pada tabel 3.3 berikut

Tabel 3.3 Hasil Enkripsi algoritma *RSA*

Karakter	**ASCII (M)**	**$C = M^e \bmod n$**
D	68	51
o	111	155
k	107	112
u	117	127
m	109	131
e	101	84
n	110	66
<spasi>	32	76
I	73	61

n	110	66
i	105	96
<spasi>	32	76
P	80	75
e	101	84
n	110	66
t	116	74
i	105	96
n	110	66
g	103	137

Selanjutnya melakukan enkripsi ulang menggunakan algoritma *ElGamal* yang telah dijelaskan pada poin (2.3.3 Proses Enkripsi) terhadap hasil enkripsi yang dilakukan dengan menggunakan algoritma RSA dan hasil yang diperoleh adalah 2 blok karakter yaitu (a, b). Kunci publik ElGamal yang telah dibuat adalah {191, 3, 52} (dapat dilihat pada Tabel 3.1) dan blok-blok hasil enkripsi algoritma *RSA* disebut sebagai plaintext (M). Untuk blok pertama adalah 51 jadi diperoleh M = 51. Dan langkah selanjutnya adalah menentukan nilai k, nilai k disini merupakan bilangan integer acak dengan ketentuan $1 \leq k \leq p - 2$. Misalnya jika nilai acak k = 7 maka hitung nilai a dengan menggunakan persamaan 2.4

$a = g^k \bmod p$

$= 3^7 \bmod 191$

$= 2187 \bmod 191 =$ **86**

dan hitung nilai b dengan Persamaan 2.5

$b = y^k \, M \bmod p$

$= 52^7 \times 51 \bmod 191$

$= ((52 \bmod 191)^4 \times (52 \bmod 191)^3 \times (51 \bmod 191)) \bmod 191$

$= ((7311616 \bmod 191) \times (140608 \bmod 191) \times (51 \bmod 191)) \bmod 191$

= (136 × 32 × 51) mod 191

= 221952 mod 191 = **10**

Sehingga hasil enkripsi yang diperoleh adalah (86, 10). Untuk hasil lengkapnya dapat dilihat pada tabel 3.4 berikut

Tabel 3.4 Hasil Enkripsi Algoritma *ElGamal*

ASCII (M)	**K (Acak)**	**$a = g^k \bmod p$**	**$b = y^k M \bmod p$**	**Hasil (ciphertext)**
51	7	86	10	(86, 10)
155	4	81	70	(81, 70)
112	6	156	88	(156, 88)
127	10	30	119	(30, 119)
131	5	52	82	(52, 82)
84	4	81	155	(81, 155)
66	8	67	55	(67, 55)
76	4	81	22	(81, 22)
61	6	156	7	(156, 7)
66	9	10	186	(10, 186)
96	5	52	98	(52, 98)
76	7	86	131	(86, 131)
75	5	52	184	(52, 184)
84	3	27	14	(27, 14)
66	10	30	122	(30, 122)
74	7	86	22	(86, 22)
96	11	90	77	(90, 77)
66	5	52	139	(52, 139)
137	6	156	94	(156, 94)

Menurut tabel 3.4 diatas, tampak bahwa dengan menggunakan bilangan integer acak yang berbeda akan menghasilkan cipherteks yang berbeda pula dan

blok yang dihasilkan juga membengkak dua kali lipat dari blok sebelumnya. Namun saat di dekripsi akan memperoleh plainteks yang sama dan blok yang didapat juga sama seperti sebelum dienkripsi. Hasil enkripsi diatas jika digabungkan akan menjadi "86 10 81 70 156 88 30 119 52 82 81 155 67 55 81 22 156 7 10 186 52 98 86 131 52 184 27 14 30 122 86 22 90 77 52 139 156 94".

3.3.3. Proses Dekripsi Dokumen

Proses dekripsi ini dilakukan untuk mengembalikan data yang terenkripsi menjadi data yang sebenarnya. Untuk melakukan proses dekripsi dibutuhan kunci privat yang dibuat bersama dengan pembuatan kunci publik yang digunakan untuk mengenkripsi data atau dokumen.

Langkah-langkah proses dekripsi yang dilakukan dalam aplikasi pengamanan data ini yaitu misalnya pengguna mempunyai data yang terenkripsi berupa "**86 10 81 70 156 88 30 119 52 82 81 155 67 55 81 22 156 7 10 186 52 98 86 131 52 184 27 14 30 122 86 22 90 77 52 139 156 94**" yang merupakan hasil enkripsi pada poin (3.2.5.2 Melakukan proses enkripsi) dan mempunyai kunci privat {23, 187} untuk kunci algoritma *RSA* dan kunci privat {131, 5} untuk kunci algoritma *ElGamal* (Tabel 3.1).

Langkah pertama yang dilakukan dalam proses dekripsi di aplikasi pengamanan dokumen ini adalah mengubah data menjadi blok-blok. Dalam proses enkripsi, data di enkripsi menggunakan algoritma *RSA* terlebih dahulu kemudian dienrkripsi ulang menggunakan algoritma *ElGamal*. Jadi untuk proses dekripsinya adalah dengan melakukan proses sebaliknya dari proses enkripsi yaitu melakukan proses dekripsi menggunakan algoritma *ElGamal* terlebih dahulu kemudian dilakukan dekripsi ulang menggunakan algoritma *RSA*.

Proses dekripsi menggunakan algoritma elgamal membutuhkan sebuah kunci privat yaitu {5, 191} dan 2 blok data sebagai contoh 2 blok pertama adalah "86" dan "10" jadi diketahui nilai a = 86 dan nilai b = 10. Kemudian dilakukan perhitungan menggunakan persamaan 2.6 untuk menentukan nilai $(a^x)^{-1}$

$(a^x)^{-1}$ $= a^{p-1-x} \bmod p$

$= 86^{191-1-5} \bmod 191$

$= 91^{185} \bmod 191 =$ **177**

Dan kemudian dihitung menggunakan persamaan 2.7 untuk mendapatkan nilai M atau plaintext.

$M = b \times (a^x)^{-1} \bmod p$

$= 10 \times 177 \bmod 191$

$= 1770 \bmod 191 =$ **51**

Jadi hasil dekripsi untuk 2 blok pertama adalah **51**. Untuk hasil selengkapnya dapat dilihat pada tabel 3.5 berikut

Tabel 3.5 Hasil dekripsi algoritma *ElGamal*

a	**b**	**$(a^x)^{-1} = a^{p-1-x} \bmod p$**	**$M = b \times (a^x)^{-1} \bmod p$**
86	10	177	51
81	70	125	155
156	88	36	112
30	119	107	127
52	82	153	131
81	155	125	84
67	55	154	66
81	22	125	76
156	7	36	61
10	186	25	66
52	98	153	96
86	131	177	76
52	184	153	75
27	14	6	84
30	122	107	66
86	22	177	74

90	77	160	96
52	139	153	66
156	94	36	137

Berdasarkan tabel 3.5 diatas, hasil yang diperoleh setelah proses dekripsi dengan menggunakan algoritma *ElGamal* sama dengan hasil enkripsi dengan menggunakan algoritma *RSA* (tabel 3.3). selanjutnya dilakukan dekripsi ulang menggunakan algoritma *RSA*. Langkah awal yang dilakukan adalah menentukan blok yang akan didekripsi, blok pertama yang akan di dekripsi adalah karakter "3" dengan kode ASCII "**51**" sehingga C = 51 sedangkan kunci privat yang digunakan untuk mendekripsi adalah kunci privat algoritma *RSA* yaitu {23, 187} (Tabel 3.1). Kemudian hitung dengan menggunakan persamaan 2.2 untuk mendapatkan plaintext (M) atau data yang sebenarnya.

$M = C^d \bmod n$

$= 51^{23} \bmod 187$

$= \mathbf{68}$

Hasil yang diperoleh dari perhitungan diatas adalah "68" yang merupakan kode ASCII dari karakter "**D**". Untuk hasil selengkapnya dapat dilihat pada tabel 3.6 berikut

Tabel 3.6 Hasil Dekripsi Algoritma *RSA*

ASCII (C)	**$M = C^d \bmod n$**	**Karakter**
51	68	D
155	111	o
112	107	k
127	117	u
131	109	m
84	101	e
66	110	n

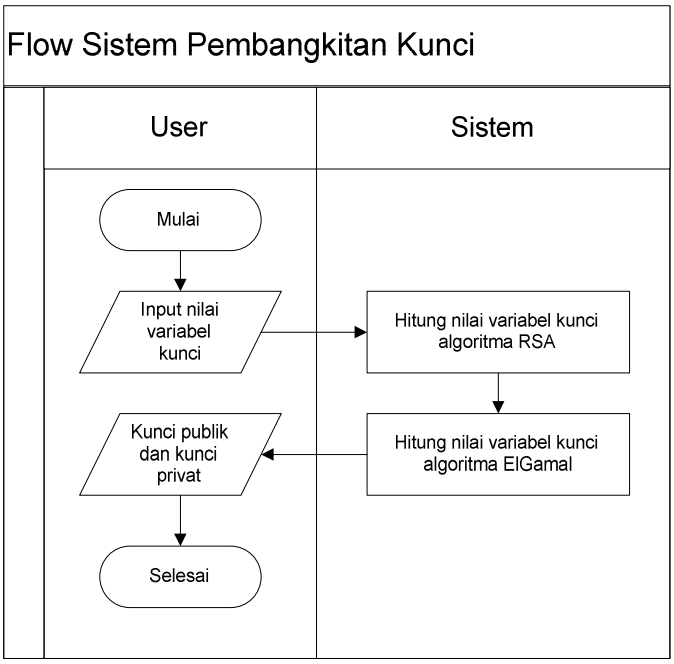
76	32	<spasi>
61	73	I
66	110	n
96	105	i
76	32	<spasi>
75	80	P
84	101	e
66	110	n
74	116	t
96	105	i
66	110	n
137	103	g

Hasil yang diperoleh dari dekripsi adalah isi asli dari dokumen yang telah dienkripsi yaitu "**Dokumen Ini Penting**" yang masing-masing mempunyai kode ASCII "**68 111 107 117 109 101 110 32 73 110 105 32 80 101 110 116 105 110 103**"

3.4. Perancangan Sistem

Dalam sistem terdapat beberapa proses yaitu proses pembangkitan kunci yang akan menghasilkan kunci publik dan kunci privat yang digunakan untuk mengenkinripsi dan mendekripsi dokumen. Proses selanjutnya adalah proses enkripsi dokumen yaitu proses untuk mengubah isi dokumen menjadi sulit dimengerti oleh orang yang tidak berhak. Dan proses dekripsi yaitu proses untuk mengembalikan dokumen menjadi seperti sebelumnya atau aslinya.

Gambaran bagaimana proses yang terjadi dalam pembangkitan kunci publik dan kunci privat dapat dilihat pada gambar 3.2

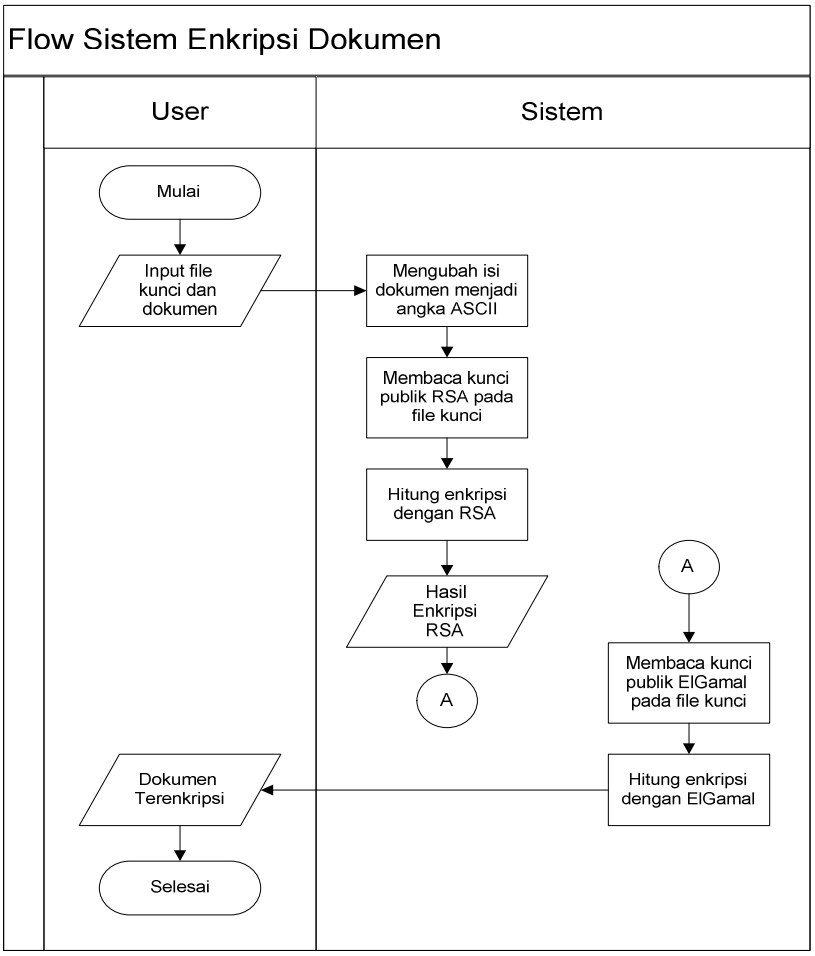


Gambar 3.8 Flowchart Sistem Untuk Pembangkitan Kunci

Pada gambar 3.8 flowchart sistem diatas terdapat dua kolom, yaitu kolom pengguna dan kolom sistem.

1. Kolom pengguna, menginputkan variabel-variabel yang dibutuhkan untuk membangkitkan kunci sehingga pengguna menerima dua pasang kunci yaitu kunci publik dan kunci privat.
2. Kolom sistem, variabel yang diinputkan pengguna akan dianalisis dan dilakukan perhitungan untuk bisa mendapatkan sepasang kunci yang digunakan dalam proses enkripsi dan dekripsi.

Adapun gambaran bagaimana proses yang terjadi dalam enkripsi dokumen dapat dilihat pada gambar 3.6 berikut



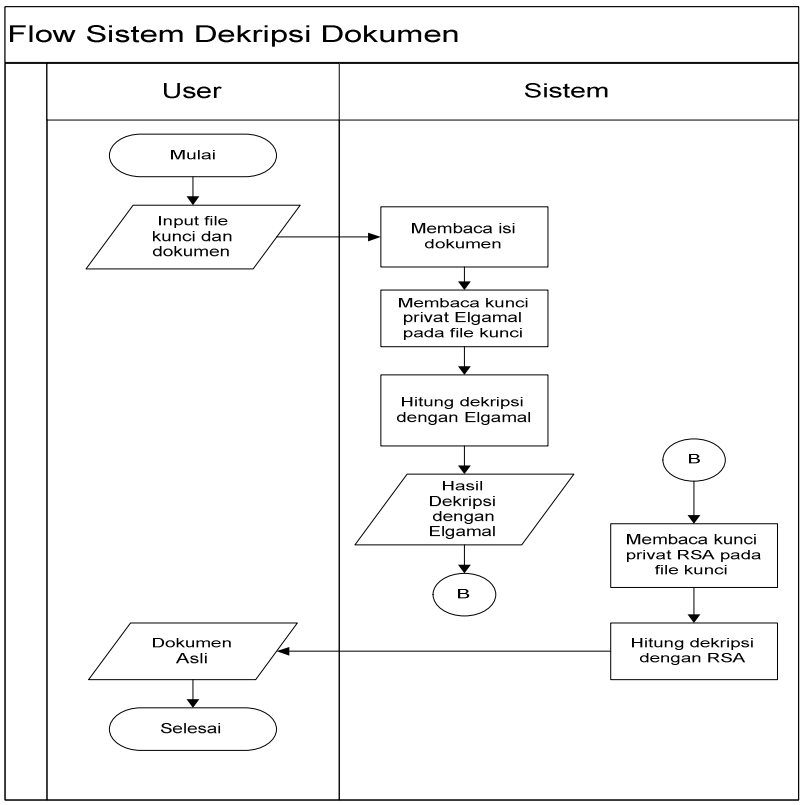
Gambar 3.9 Flowchart Sistem Untuk Enkripsi Dokumen

Pada gambar 3.9 flowchart sistem diatas terdapat dua kolom yaitu kolom pengguna dan kolom sistem.

1. Kolom pengguna, menginputkan kunci publik dan dokumen yang akan dienkripsi. Kemudian pengguna akan mendapatkan dokumen yang telah terenkripsi.
2. Kolom sistem, dokumen inputan pengguna akan dianalisis dan isi dari dokumen akan diubah menjadi angka-angka sesuai dengan kode ASCII dari masing-masing karakter kemudian dilakukan proses pencarian atau pembacaan kunci publik yang terdapat pada file kunci dan digunakan untuk melakukan enkripsi sehingga isi sebenarnya dari dokumen tersebut

tidak dapat diketahui sebelum dilakukan proses pendekripsian untuk dokumen tersebut.

Gambaran bagaimana proses yang terjadi dalam dekripsi dokumen dapat dilihat pada gambar 3.10

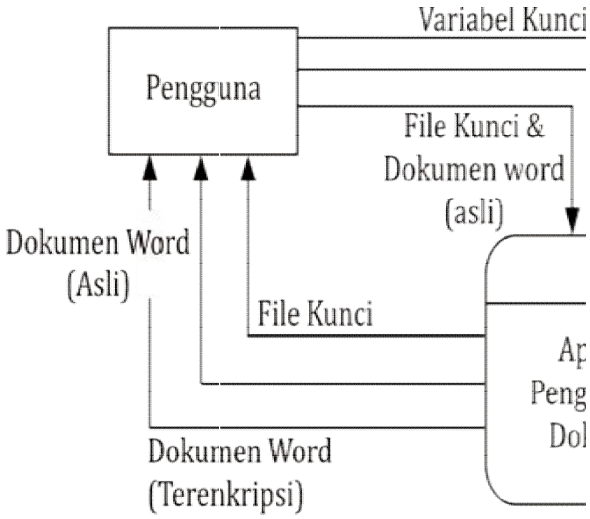


Gambar 3.10 Flowchart sistem untuk dekripsi

Sama halnya dengan flowchart sistem untuk enkripsi (Gambar 3.9), flowchart sistem untuk proses enkripsi (Gambar 3.10) juga terdapat dua kolom yaitu kolom pengguna dan kolom sistem.

1. Kolom pengguna, menginputkan kunci privat dan dokumen yang telah terenkripsi. Kemudian pengguna akan mendapatkan file dokumen yang telah didekripsi.
2. Kolom sistem, dokumen dan file kunci yang diinputkan pengguna akan dianalisis dan dilakukan proses dekripsi sehingga menjadi dokumen yang sebenarnya (sebelum dienkripsi).

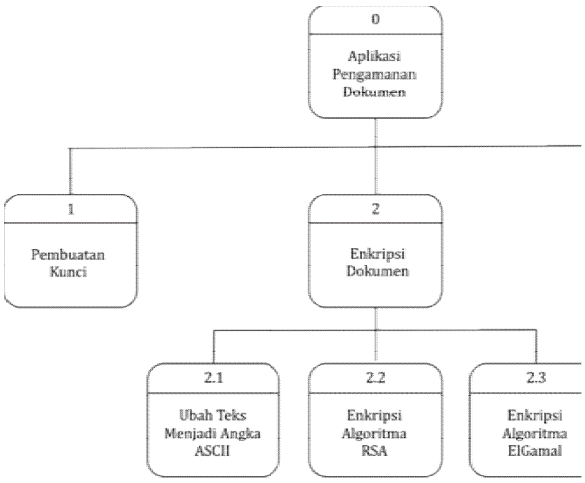
3.4.1. Diagram Konteks



Gambar 3.11 Diagram Konteks Aplikasi Pengamanan Dokumen

Diagram konteks aplikasi pengamanan dokumen menggunakan algoritma kriptografi asimetris *RSA* dan *ElGamal* pada gambar 3.11 hanya melibatkan 1 external entity yaitu pengguna.

3.4.2. Diagram Berjenjang



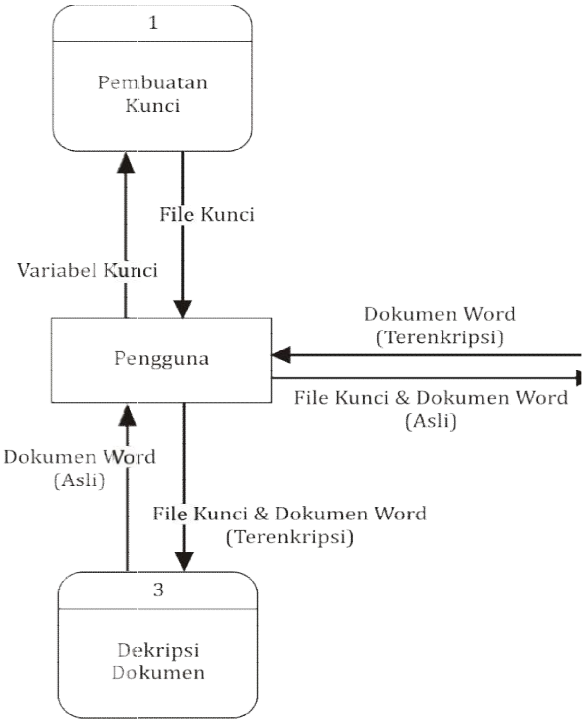
Gambar 3.12 Diagram Berjenjang Aplikasi Pengamanan Dokumen

Adapun keterangan gambar 3.12 secara rinci ada sebagai berikut

1. Top level : aplikasi pengamanan dokumen

2. Level 1 : 1. Pembuatan Kunci
   2 Enkripsi Dokumen
   3 Dekripsi Dokumen
3. Level 2 : 2.1 Ubah Teks Menjadi Angka ASCII
   2.2 Enkripsi Algoritma RSA
   2.3 Enkripsi Algoritma Elgamal
   3.3 Ubah Teks Menjadi Blok Angka
   3.2 Dekripsi Algoritma ElGamal
   3.3 Dekripsi Algoritma RSA

3.4.3. DFD Level 1 Aplikasi Pengamanan Dokumen



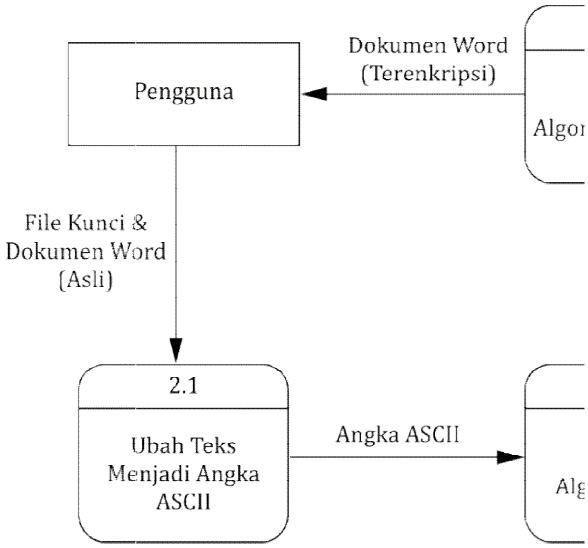
Gambar 3.13 DFD Level 1 Aplikasi Pengamanan Dokumen

Data flow diagram Level 0 aplikasi pengamanan dokumen seperti terlihat pada gambar 3.13 menunjukkan 3 (tiga) proses, yang pertama adalah proses pembangkitan kunci yang digunakan untuk membuat sepasang kunci yaitu kunci publik dan kunci privat yang nantinya akan digunakan untuk melakukan proses enkripsi dan dekripsi. Kedua adalah proses enkripsi yaitu proses dimana

melakukan perubahan isi dokumen yang sebenarnya menjadi isi dokumen yang bukan sebenarnya (berupa angka-angka). Ketiga adalah proses dekripsi yang merupakan kebalikan dari proses enkripsi yaitu mengembalikan isi dokumen yang telah dienkripsi menjadi isi dokumen yang sebenarnya.

3.4.4. DFD Level 2 Proses Enkripsi Dokumen

Gambar 3.14 berikut adalah data flow diagram level 2 proses enkripsi dokumen yang menunjukkan adanya 3 proses dalam melakukan proses enkripsi. Yang pertama adalah proses konversi isi dokumen menjadi blok-blok angka ASCII. Yang kedua adalah proses enkripsi menggunakan algoritma *RSA*. Dan yang ketiga adalah proses enkripsi ulang menggunakan algoritma *ElGamal* untuk hasil dari enkripsi algoritma *RSA*.

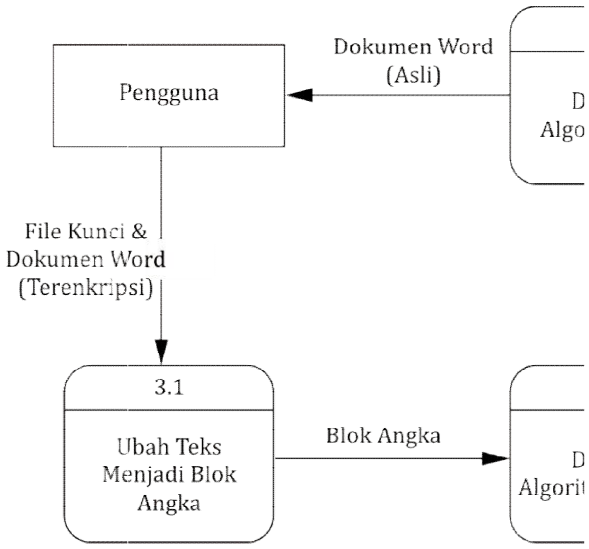


Gambar 3.14 DFD Level 2 Proses Enkripsi Dokumen

3.4.5. DFD Level 1 Proses Dekripsi Dokumen

Data flow diagram level 2 Proses Dekripsi seperti terlihat pada gambar 3.15 menunjukkan adanya 3 proses dalam melakukan proses dekripsi. Sama halnya dengan proses enkripsi, proses pertama adalah proses konversi isi teks dokumen menjadi blok-blok angka ASCII. Proses selanjutnya adalah proses dekripsi menggunakan algoritma *ElGamal* dan yang ketiga adalah hasil dari pendekripsian isi dokumen yang menggunakan algoritma *ElGamal* dilakukan

pendeskripsian ulang menggunakan algoritma *RSA* sehingga dapat menghasilkan isi dokuman yang sebenarnya.



Gambar 3.15 DFD Level 2 Proses Dekripsi Dokumen

3.5. Spesifikasi Kebutuhan Perangkat

Dalam pembuatan aplikasi pengamanan data dokumen menggunakan algoritma kriptografi *RSA* dan *ElGamal* dibutuhkan spesifikasi perangkat keras dan perangkat lunak sebagai berikut :

3.5.1. Kebutuhan Perangkat Keras

Perangkat keras merupakan komponen fisik yang membentuk sistem komputer, serta peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Adapun perangkat keras yang dibutuhkan untuk menjalankan aplikasi ini yaitu :

1. Processor minimum Pentium 4
2. Memory minimum SDRAM 1 GB
3. Hardisk dengan kapasitas penyimpanan minimum 30GB
4. Monitor
5. Keyboard
6. Mouse

3.5.2. Kebutuhan Perangkat Lunak

Perangkat lunak (sofware) adalah program-program yang digunakan untuk menjalankan atau mendukung sistem perangkat keras (*Hardware*). Adapun perangkat lunak yang dibutuhkan untuk menjalankan aplikasi sistem ini yaitu :

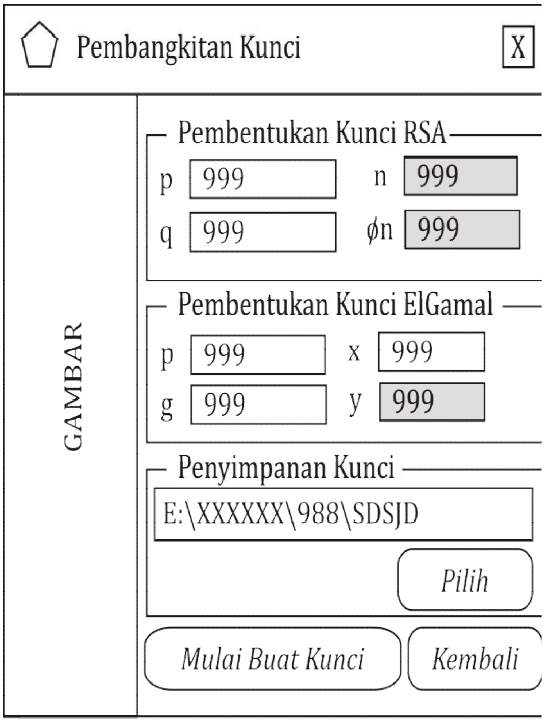
1. Sistem operasi windows
2. Microsoft Office 2007
3. Visual Basic 6.0

3.6. Desain Antarmuka

Interface adalah bagian yang menghubungkan antara aplikasi pengamanan data dokumen dengan user. *Interface* yang digunakan dalam sistem ini adalah sistem yang berbasis desktop dengan source code yang dipakai menggunakan visual basic 6.0. halaman yang akan dibuat adalah sebagai berikut :

3.6.1. Form Pembangkitan Kunci

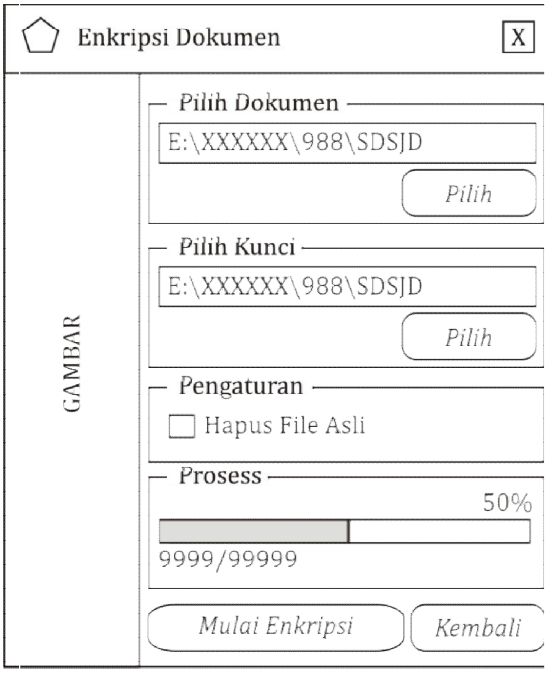
Menurut gambar 3.16 perancangan form pembangkitan kunci, dalam form tersebut terdapat beberapa textbox yang digunakan menginputkan data-data atau variabel-variabel yang diperlukan dalam melakukan pembentukan kunci. Nilai *p, q, n* dan ϕn pada GroupBox Pembentukan Kunci dimana nilai p dan q merupakan bilangan prima dan nilai p dan q tidak boleh sama sedangkan pada nilai n merupakan output dari perhitungan nilai $p \times q$ dan nilai n juga merupakan output dari perhitungan nilai $(p - 1) \times (q - 1)$. Pada GroupBox Pembentukan Kunci ElGamal terdapat inputan nilai *p, g* dan *x* dimana *p*_merupakan bilangan prima besar (disarankan bilangan prima yg lebih dari 100) sedangkan nilai *g* dan *x* harus lebih kecil dari nilai *p*. Untuk nilai *y* merupakan output hasil perhitungan dari $g^x \ mod \ p$. Dalam form juga terdapat tombol pilih untuk memilih tempat penyimpanan kunci yang akan dibuat dan lokasi yang dipilih akan ditampilkan pada Textbox yang terdapat pada GroupBox penyimpanan kunci. Tombol mulai buat kunci disini akan difungsikan untuk memulai pembuatan kunci dan tombol kembali digunakan untuk kembali ke form utama.



Gambar 3.16 Desain Form Pembangkitan Kunci

3.6.2. Form Enkripsi

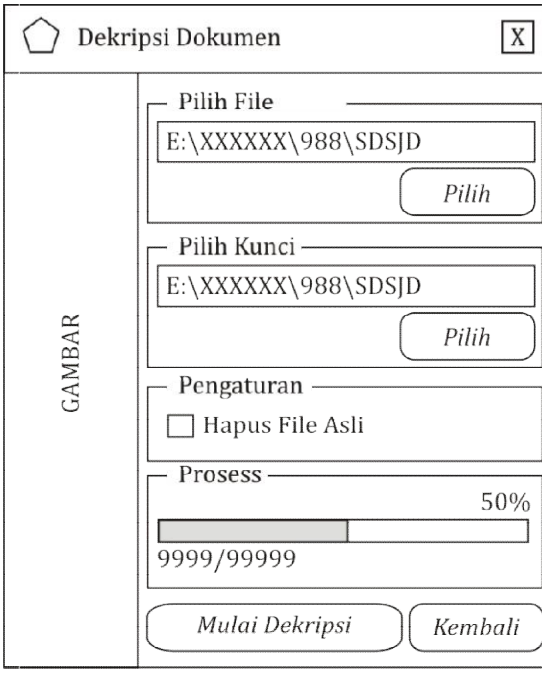
Dalam form ini tombol untuk memilih dokumen dan file kunci. Lokasi dokumen yg dipilih akan ditampilkan pada Textbox yang berada pada Groupbox Pilih dokumen sedangkan lokasi kunci yang dipilih berada pada GroupBox Pilih Kunci. Selain itu, disini juga disertakan pengaturan apakah file asli akan dihapus atau tidak. Form ini juga berisi progres dalam melakukan enkripsi data. Terdapat juga tombol mulai enkripsi untuk memulai proses enkripsi. Dan tombol kembali untuk kembali ke form utama namun tombol ini akan dinonaktifkan jika proses enkripsi sedang berlangsung. Untuk desain perancangan form enkripsi dapat dilihat pada gambar 3.17 berikut



Gambar 3.17 Desain Form Enkripsi

3.6.3. Form Dekripsi

Pada gambar 3.18 perancangan form dekripsi, yaitu form yang digunakan untuk mendekripsi dokumen. Dalam form ini berisi button untuk memilih dokumen, lokasi dan nama dokumen akan disimpan dalam textbox yang ada pada box pilih dokumen. Terdapat juga button untuk memilih sebuah file kunci privat, lokasi dan nama file akan disimpan dalam textbox yang terdapat pada box pilih kunci. Pada box pengaturan terdapat checkbox yang digunakan pilihan apakah file asli yang terenkripsi akan dihapus setelah proses dekripsi selesai atau tidak, dan juga terdapat progres untuk proses dekripsi. Button mulai dekripsi digunakan untuk memulai proses dekripsi sedangkan proses kembali digunakan untuk kembali ke halaman/form utama namun jika proses dekripsi sedang berjalan maka button ini tidak dapat digunakan atau dengan kata lain dinonaktifkan.



Gambar 3.18 Desain Form Dekripsi

3.7. Skenario Pengujian

Pengujian aplikasi pengamanan dokumen dengan menggunakan algoritma kriptografi asimetris RSA dan ElGamal ini dilakukan dengan cara menguji keakuratan dalam pengembalian isi dokumen yang telah dienkripsi menjadi isi dokumen yang sebenarnya.