BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Tahap analisis dan perancangan merupakan tahap sistematis untuk menyesuaikan kegunaan dan tujuan pada aplikasi. Tahap awal pada tahap analisis dimulai dari analisis masalah, menganalisis sistem dengan menganalisis aplikasi sejenis, analisis fungsional dan non-fungsional. Sedangkan untuk tahap perancangan dimulai dengan melakukan perancangan sistem yang mencakup perancangan antar muka dan perancangan struktur menu yang nantinya akan digunakan untuk diterapkan pada aplikasi.

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponenya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, kebutuhan serta hambatan yang terjadi. Menentukan kebutuhan yang sesuai dengan kebutuhan-kebutuhan yang diharapkan, untuk melakukan analisis sistem dilakukan beberapa analisis yaitu :

a. Analisis Masalah
b. Analisis Sistem yang Berjalan
c. Analisis Aplikasi Kriptografi yang Dibangun
d. Analisis Metode / Metode RSA – CRT
e. Analisis Kebutuhan Non-Fungsional
f. Analisis Kebutuhan Fungsional

3.1.1 Analisis Masalah

Analisis masalah adalah tahap penjabaran masalah yang ada sebelum aplikasi ini dibangun dan bertujuan untuk membantu pembangunan aplikasi kriptografi ini. Berikut adalah penjabaran dari masalah – masalah yang ada antara lain sebagai berikut :

1. Belum terdapat pembatasan hak akses untuk melihat citra digital hasil rancangan dari pihak house the house. Hal ini mengakibatkan orang lain yang tidak memiliki kepentingan dapat melihat dan menyalahgunakan citra digital tersebut.
2. Sebuah gambar asli yang diserahkan tentu bisa dilihat berkali-kali dan ini mengakibatkan pihak klien dapat menyalahgunakan gambar tersebut untuk kepentingan pribadi seperti menggunakan desain tersebut tanpa kontrak kerjasama lebih lanjut dengan pihak perusahaan.

3.1.2 Analisis Sistem yang Berjalan

Berdasarkan hasil pengamatan dan wawancara langsung didapatkan prosedur yang dilakukan pihak perusahaan dalam proses pengajuan atau penanganan proyek yang diminta oleh pihak klien. Secara garis besar terdapat beberapa tahapan yang dilakukan seperti berikut :

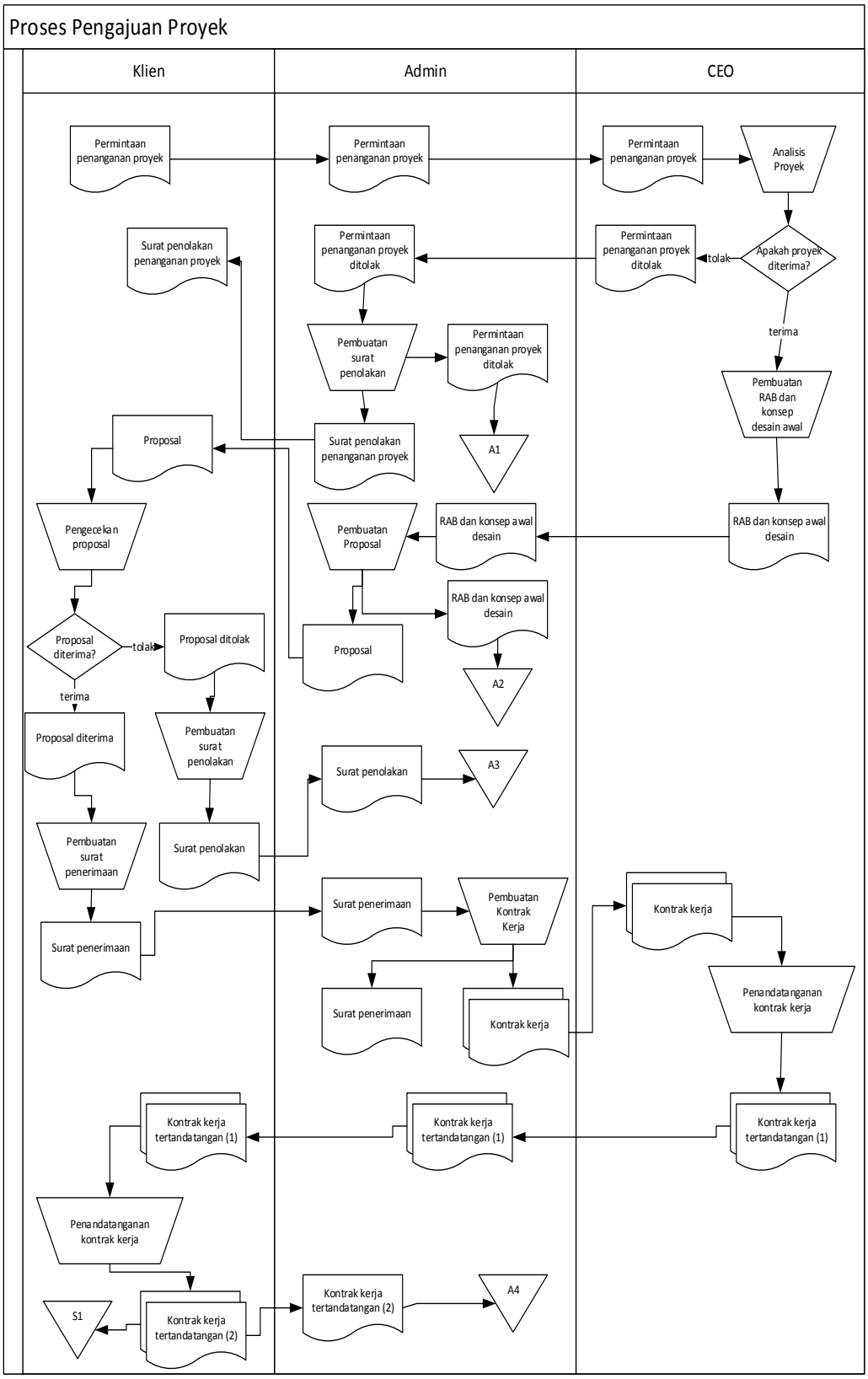
1. Pihak klien meminta perusahaan untuk menangani proyek yang mereka miliki. Untuk hal ini bisa dilakukan tanpa dokumen namun pihak perusahaan meminta dokumen, minimal memo sebagai bukti untuk CEO dan dokumentasi perusahaan.
2. Memo yang telah sampai pada admin akan diperiksa oleh CEO untuk dirapatkan dengan tim yang bersangkutan. Selanjutnya CEO akan memberikan keputusan apakah akan mengambil atau menolak proyek.
3. Jika proyek ditolak maka admin akan memberikan surat penolakan dengan disertai alasan pihak perusahaan. Sedangkan jika proyek diterima maka CEO akan memberikan rancangan awal dan rancangan anggaran biaya kepada admin untuk dibuatkan dalam bentuk proposal.
4. Proposal yang telah dibuat akan disampaikan kepada pihak klien. Pihak klien akan memberikan keputusan apakah setuju dengan rancangan dan anggaran biaya yang diajukan. Jika klien tidak setuju maka mereka akan memberikan surat penolakan kepada pihak perusahaan. Sedangkan jika mereka setuju maka pihak klien akan memberikan surat persetujuan kepada perusahaan

5. Surat persetujuan atau penerimaan dari pihak klien terhadap proposal pihak perusahaan yang diterima admin akan disimpan. Selanjutnya admin akan membuatkan kontrak kerja.
6. Kontrak kerja yang telah dibuat admin akan ditandatangi oleh kedua pihak dan kontrak tersebut adalah landasan perusahaan dalam bekerja. Kontrak kerja tersebut harus disimpan dengan baik oleh kedua pihak karena itu sebagai bukti jika nanti suatu hari proyek tidak berjalan sesuai rencana.
7. Setelah kontrak kerja ditandatangani kedua pihak maka proyek dapat langsung berjalan.

Permasalahan yang terjadi dalam sistem yang berjalan pada perusahaan konsultan *Art and Design House The House* terdapat pada poin 3 dan 4. Untuk lebih jelasnya kita uraikan kembali permasalah yang terjadi pada proses berikut :

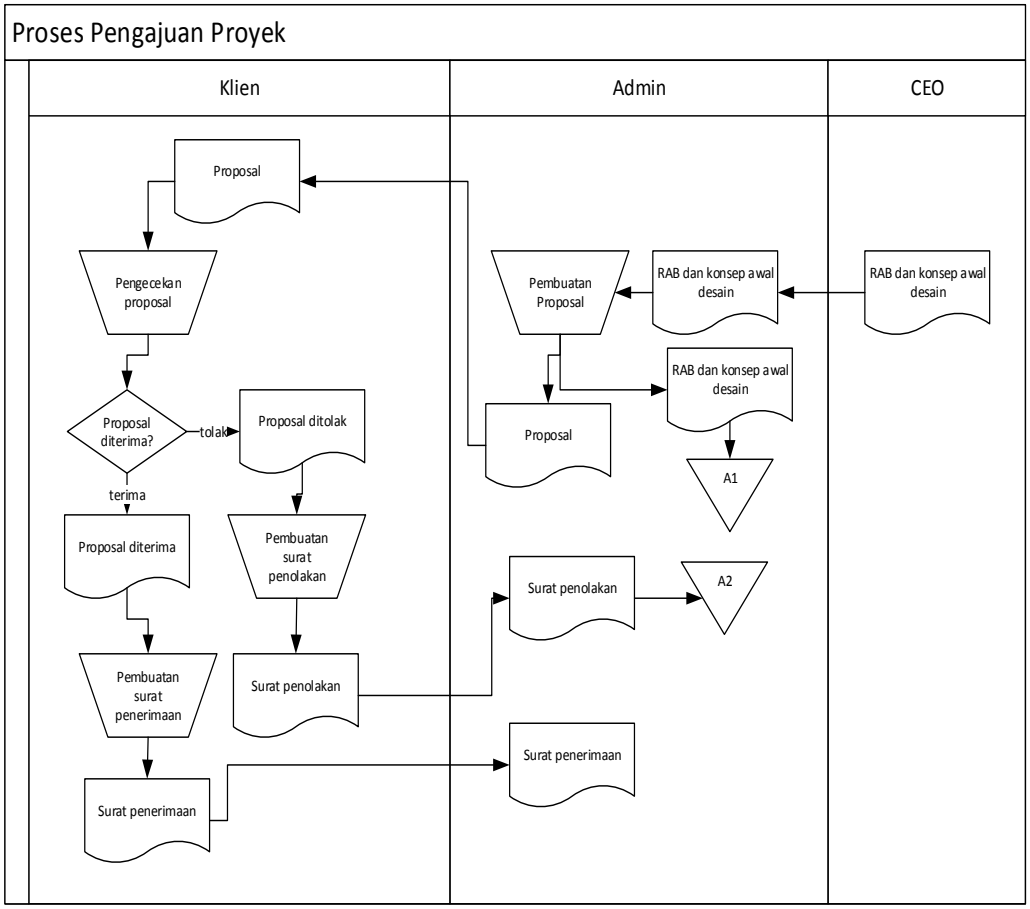
1. Pihak perusahaan memberikan proposal pembangunan kepada pihak klien yang didalamnya terdapat anggaran biaya, waktu dan konsep serta lampiran berupa desain gambar.
2. Pihak klien akan memberikan keputusan apakah setuju dengan rancangan dan anggaran biaya yang diajukan. Jika klien tidak setuju maka mereka akan memberikan surat penolakan kepada pihak perusahaan namun proposal dan contoh desain tidak dikembalikan kepada pihak perusahaan. Sedangkan jika mereka setuju maka pihak klien akan memberikan surat persetujuan kepada perusahaan dan akan dibuatkan kontrak kerja dengan klien.

Prosedur yang dijelaskan diatas adalah prosedur formal yang berjalan di konsultan *Art and Design House The House.* Untuk lebih jelasnya dari semua prosedur yang berjalan diatas dapat terlihat dari flowmap proses pengajuan proyek seperti berikut :



Bagan 3.1 Flowmap Pengajuan Proyek

Flowmap diatas adalah proses lengkap penanganan proyek dari pihak perusahaan terhadap klien. Adapun untuk melihat permasalah yang terjadi adalah pada proses berikut :

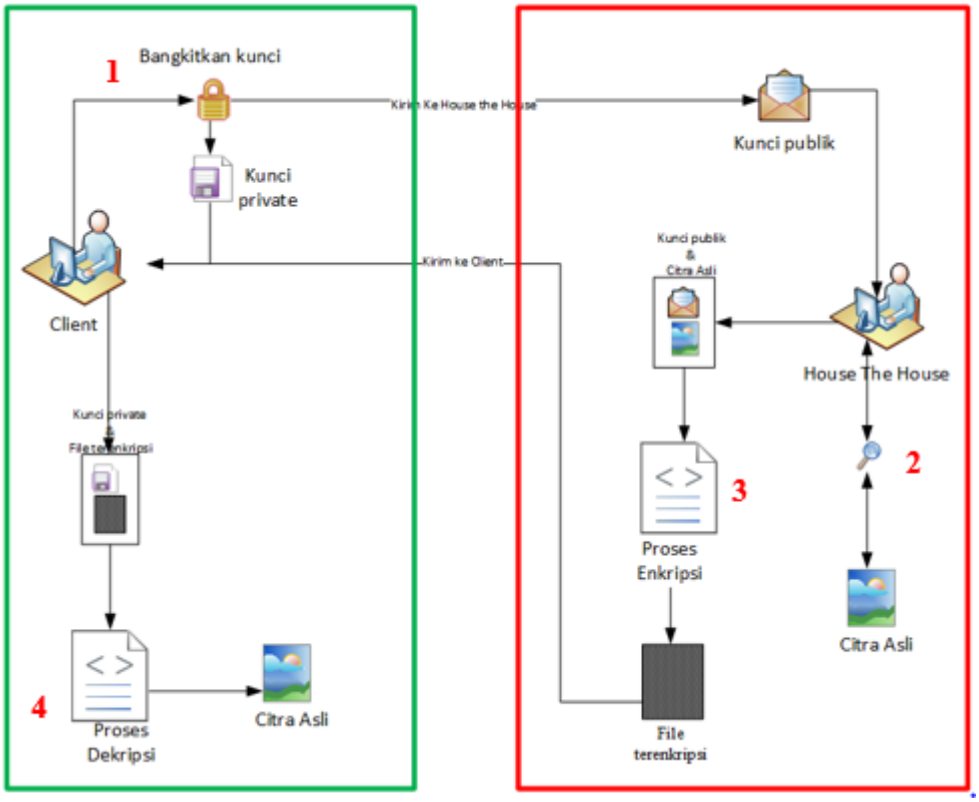


Bagan 3.2 Flowmap permasalahan pada sistem yang berjalan

3.1.3 Analisis Aplikasi yang Akan Dibangun

Aplikasi yang dibangun adalah sebuah aplikasi yang dapat melakukan kriptografi terhadap citra. Proses kriptografi ini dilakukan agar citra digital tidak bisa dilihat menggunakan aplikasi pembuka foto. Hal ini bisa dilakukan dengan asumsi bahwa setiap foto yang akan telah dikenakan algoritma kriptografi akan mengalami pertambahan nilai sehingga citra tidak dapat dibaca oleh aplikasi pembuka foto biasa. Selain itu juga akan diberikan penyisipan nilai pada citra digital sehingga citra digital hanya dapat dibuka beberapa kali saja.

Untuk lebih jelasnya proses yang akan terjadi dalam aplikasi yang akan dibangun adalah sebagai berikut :



Bagan 3.3 Proses kriptografi terhadap citra

Hal yang dilakukan bagan 3.3 adalah interaksi antara pihak *House The House* dan pihak *Client.* Proses pertama yang dilakukan adalah proses diluar sistem yaitu saat pihak klien meminta sebuah desain maka pihak perusahaan akan mengirimkan sebuah aplikasi dekripsi yang dilengkapi dengan menu pembangkit kunci. Selanjutnya adalah proses yang termasuk dalam cakupan aplikasi :

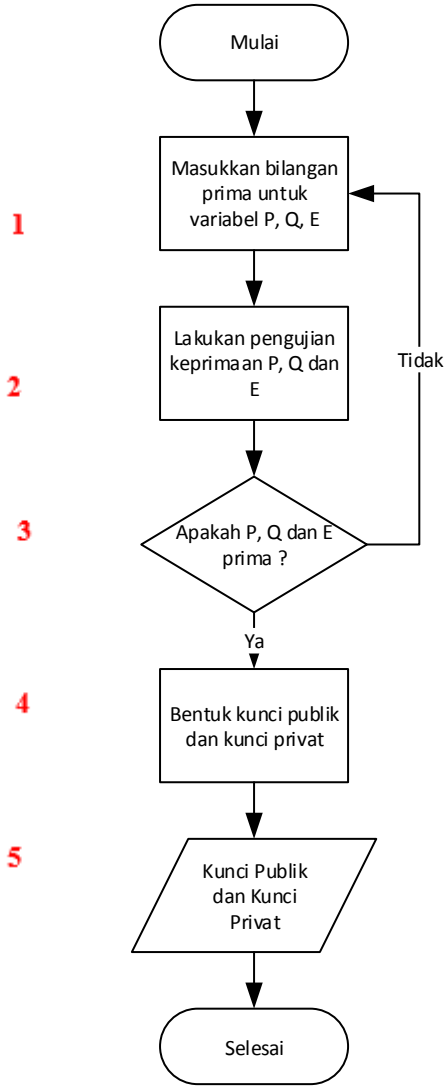
a. Pertama pihak klien membangkitkan kunci privat dan kunci publik. Kunci privat disimpan pihak klien dan kunci privat harus dijaga dengan baik karena kunci ini akan digunakan untuk proses dekripsi. Jika kunci ini berhasil didapatkan oleh pihak lain maka citra akan mudah didekripsi sehingga rancangan bisa dilihat oleh pihak yang tidak berkepentingan.
b. Kunci publik yang telah dibangkitkan ditahap pertama akan dikirimkan ke pihak perusahaan. Kunci ini berguna untuk melakukan proses enkripsi citra. Pengiriman ini bisa dilakukan melalui jaringan lain diluar aplikasi. Kunci ini sifatnya aman karena jika kunci publik berhasil diambil maka pihak lain tidak dapat melakukan proses dekripsi karena proses dekripsi hanya menggunakan kunci privat.
c. Tahapan ketiga adalah ketika kunci publik telah didapatkan oleh pihak perusahaan maka pihak perusahaan akan melakukan pencarian gambar yang akan dikirimkan kepada klien.
d. Setelah gambar dan kunci siap maka keduanya akan dimasukkan kedalam aplikasi enkripsi. Pada proses ini akan gambar/citra akan dienkripsi dengan menyisipkan nilai pembatas dekripsi pada salah satu indeks *array byte*. Pada proses enkripsi ini akan menghasilkan suatu file dengan format .jpg.
e. Ketika proses enkripsi telah dilakukan dan menghasilkan file terenkripsi maka selanjutnya pihak perusahaan akan mengirimkan file terenkripsi tersebut kepada pihak perusahaan.
f. File terenkripsi yang telah sampai pada pihak perusahaan akan digunakan pada proses dekripsi. Dengan menggunakan kunci privat yang telah disimpan sebelumnya dan file terenkripsi tersebut maka proses dekripsi dapat dilakukan. Proses dekripsi ini akan mengembalikan file terenkripsi menjadi file citra semula. Pada proses enkripsi ini terdapat hal penting yaitu

proses pengecekan batas dekripsi. Pengecekan ini akan menentukan file tersebut masih bisa dienkripsi ataukah tidak. Jika proses ini telah dilakukan maka akan menghasilkan file citra aslinya dan tahapan proses aplikasi ini pun selesai.

3.1.3.1 Analisis Pembangkit Kunci

Proses pembangkit kunci dilakukan terlebih dahulu oleh klien. Proses ini akan menghasilkan kunci privat yang digunakan untuk proses dekripsi dan kunci publik yang akan digunakan pihak perusahaan untuk proses enkripsi. Proses ini mewakili model proses kriptografi pada bagan 3.3 pada tahapan 1.

Berikut flowchat yang akan menggambarkan proses pembangkit kunci dari sebuah metode RSA – CRT :



Bagan 3.4 Flowchart Pembangkit Kunci

3.1.3.1.1 Pengujian Bilangan Prima

Dalam aplikasi ini diperlukan pengujian nilai bilangan prima, hal ini untuk memenuhi tahap 2 dan 3 *flowchart* 3.4. Berikut untuk melakukan pengujian bilangan prima :

Tabel 3.1 Algoritma pengujian bilangan prima

	Algoritma Uji Bilangan Prima
1	BP ← Bilangan Prima.
2	**If** BP <= 1 **Then**
3	**Output** (“Bukan Bilangan Prima”)
4	**Else**
5	**For** i=2 to BP-1 **do**
6	**If** (BP mod i = 0) **then**
7	**Output** (“Bukan Bilangan Prima”)
8	**Else**
9	**Output** (“Bilangan Prima”)
10	**Endif**
11	**Endfor**
12	**Endif**

3.1.3.1.2 Proses Pembangkit Kunci

Flowchart 3.4 pada proses 1, 4 dan 5 dijelaskan dalam bentuk algortima berikut :

Tabel 3.2 Algoritma Pembangkit Kunci RSA - CRT

Tahap	**Algoritma** Pembangkit Kunci RSA – CRT
1	Bangkitkan bilangan prima besar *p* dan *q*.
2	$n \leftarrow p * q$
3	$\phi(n) \leftarrow (p - 1) * (q - 1)$
4	$e \leftarrow Z_{\phi(n)}$ dengan $\gcd(e, \phi(n)) = 1$
5	$d \leftarrow e^{-1}$ pada $Z_{\phi(n)}$
6	$dP \leftarrow d \bmod (p - 1)$

7	dQ ← d mod (q – 1)
8	qInv = q^{-1} pada Z_p
	K_{publik} = (e, n), K_{privat} = (dP, dQ, qInv, p, q)

Uraian untuk proses pembakit kunci RSA – CRT adalah sebagai berikut :

1. Membangkitkan bilangan prima besar untuk nilai p dan q (tahap 1)

Bilangan prima merupakan bilangan asli (bilangan yang lebih besar sama dengan nol, b >= 0) yang lebih besar dari 1 (p > 1) memiliki faktor pembagi yaitu 1 dan bilangan itu sendiri. Contoh bilangan prima adalah {2, 3, 5, 7, ...}. Satu (1) bukan merupakan bilangan prima karena menurut pengertian bilangan prima, bilangan prima adalah bilangan bulat lebih dari 1 [2]. Nilai p dan q tidak boleh sama ($p \neq q$)

Pada proses pembangkit kunci ini diperlukan bilangan prima karena hanya bilangan prima yang dapat memenuhi prinsip sistem kongruen pada eksponensial modular. Selain itu perkalian prima besar yang akan dilakukan pada langkah 2 akan menentukan kekuatan dari proses enkripsi yang akan dilakukan.

Sebagai contoh untuk menggambarkan proses analisis ini akan kita tentukan nilai p dan q sebagai berikut p = 17 dan q = 31.

2. Perkalian p dan q untuk mendapatkan nilai n (tahap 2)

Nilai n ini akan menjadi salah satu variabel untuk melakukan proses enkripsi. Semakin besar nilai p dan q maka akan menghasilkan nilai n yang semakin besar. Nilai n yang besar maka akan semakin sulit bagi peretas untuk menentukan nilai p dan q. Nilai p dan q sendiri selain untuk menentukan nilai n tapi juga merupakan sumber bagi pembentukan nilai d yang merupakan unsur pembentuk kunci privat.

Sebagai contoh untuk menggambarkan proses analisis ini akan kita tentukan nilai p dan q yang telah ditetapkan pada proses pertama akan kita lakukan perkalian.

n = p x q

= 17 x 31

= 527

3. Mencari nilai ϕn (tahap 3)

Mencari nilai ϕn sangat penting, hal ini karena nilai ϕn akan digunakan untuk menentukan nilai e yang akan dicari pada proses selanjutnya.

Dengan nilai yang telah ditetapkan pada contoh di proses sebelumny maka nilai ϕn dapat kita tetapkan sebagai berikut :

ϕn = (p – 1) x (q – 1)

= (17 – 1) x (31 -1)

= 16 x 30

= 480

4. Menentukan nilai e (tahap 4)

Nilai e merupakan pasangan kunci publik n (e, n). Nilai e ini juga yang akan menentukan kekuatan dari sebuah *chipertext*. Untuk mencari nilai e akan digunakan prinsip faktor persekutuan terbesar. Hal ini untuk mencari nilai prima relatif.

Faktor persekutuan terbesar adalah elemen terbesar pada himpunan *divisor* (himpunan bilangan integer yang membagi habis sebuah bilangan integer) dua bilangan integer [2]. Misalnya 18 memiliki *divisor* {1, 2, 3, 6, 9,18} dan 12 memiliki *divisor* {1, 2, 3, 4, 6, 12} maka himpunan *divisor* bersamanya adalah {1, 2, 3, 6} dan yang terbesar adalah 6. Dinotasikan sebagai gcd(18, 12) = 6.

Dalam tahapan ke-4 algoritma pembangkit kunci RSA – CRT dituliskan bahwa "$e \leftarrow Z_{\phi(n)}$ dengan gcd(ϕ(n), e) = 1". Hal ini berarti bahwa untuk setiap nilai e dan ϕ(n) akan memiliki faktor persekutuan terbesar 1. Untuk kondisi seperti gcd(ϕ(n), e) =1 biasa disebut juga dengan "Prima Relatif".

Untuk memenuhi kondisi tahapan tersebut maka bisa gunakan cara sederhana yaitu menggunakan bilangan prima untuk mengisi nilai *e*. Bilangan prima yang digunakan haruslah memiliki nilai yang bukan merupakan divisor dari nilai ϕ(n). Sebagai contoh :

Diberikan suatu nilai untuk P = 17 dan Q = 31. Maka kita bisa saja mengambil nilai prima sembarang untuk nilai e. Jika P =17 dan Q = 31 maka nilai ϕ(n) sebagai berikut :

$\phi(n) = (p - 1) \times (q - 1)$

$= (17 - 1) \times (31 - 1)$

$= 16 \times 30$

$= 480$

Maka langkah selanjutnya adalah menentukan nilai e, nilai e akan diberikan adalah bilangan prima yang bukan merupakan *divisor* dari 480. Penggunaan bilangan prima ini tidaklah mutlak karena bisa saja kita menggunakan bukan bilangan prima selama memenuhi syarat gcd(ϕ(n), e) = 1. Sebagai contoh bisa saja kita menggunakan nilai 77. Bisa kita lihat nilai gcd(480, 77) dengan algoritma Euclid yang bersifat iteratif [2] seperti berikut ini :

Tabel 3.3 Algoritma Euclid Iteratif

	Algoritma Euclid Iteratif
1	A ← e
2	B ← ϕ(n)
3	**While** B > 0 **do**
4	Q ← A/B
5	R ← A – Q * B
6	A ← B
7	B ← R
8	**Endwhile**
9	**Return** A

Tabel 3.4 Contoh gcd (480, 77)

A	**B**	**Q = A / B**	**R = A mod B**
480	77	6	18
77	18	4	5
18	5	3	3
5	3	1	2
3	2	1	1

A	B	Q = A / B	R = A mod B
2	1	2	0
1	0		

Dari hasil diatas terlihat bahwa gcd (480, 77) = 1 hal ini sesuai dengan syarat gcd(ϕ(n), e) = 1. Namun penggunaan bilangan acak ini akan memberikan beban tersendiri terhadap memori karena harus melakukan pengujian terhadap nilai "*prima relatif*" yang belum tentu bernilai benar. Sedangkan dengan menggunakan bilangan prima kita hanya akan melakukan pengujian nilai dengan cara membagi nilai $\phi(n)$ dengan *e*. Hal ini bisa dilakukan karena bilangan prima hanya memiliki faktor pembagi bilangan itu sendiri dan 1, yang berarti bahwa yang mungkin membagi bilangan $\phi(n)$ hanya nilai *e* itu sendiri. Sebagai contoh kita akan menggunakan nilai *e* = 73 dengan nilai $\phi(n)$ tetap 480. Berikut terlihat di tabel berikut :

Tabel 3.5 Contoh gcd (480, 73)

A	B	Q = A / B	R = A mod B
480	73	6	42
73	42	1	31
42	31	1	11
31	11	2	9
11	9	1	2
9	2	4	1
2	1	2	0
1	0		

Dari tabel diatas terlihat bahwa nilai untuk gcd (480, 73) = 1. Hal ini membuktikan bahwa nilai prima dapat digunakan untuk nilai dari *e*. Sehingga nilai *e* dalam aplikasi ini hanya bisa dimasukkan dengan nilai prima.

Nilai e dapat sembarang dengan syarat bahwa nilai e hanya memiliki faktor persekutuan terbesar dengan ϕn yaitu 1. Untuk mempercepat proses yang ada maka dapat kita asumsikan bahwa nilai e merupakan bilangan prima yang tidak habis membagi nilai ϕn. Dengan hal tersebut maka kita tentukan nilai e = 73. Dengan demikian sampe tahap ke- 4 ini telah kita dapatkan kunci publik yaitu

K_{publik} = (e, n)

= (73, 527)

5. Menentukan nilai d (tahap 5)

Nilai d digunakan untuk proses dekripsi . Dalam algortima RSA – CRT nilai d akan diuraikan ke beberapa variabel kunci yaitu dP, dQ dan qInv. Nilai d bisa dipecah ke beberapa variabel tersebut karena menggunakan sistem kongruen modular.

Untuk mencari nilai d digunakan teorema invers. Invers digunakan pada proses pembangkit kunci dan dekripsi. Dalam pembangkit kunci invers digunakan untuk mencari nilai *d* dan *qInv.* Seperti yang terlihat pada algoritma pembangkit kunci yaitu "$d = e^{-1}$ *pada* $Z_{\phi(n)}$" dan "$qInv = q^{-1}$ *pada* Z_p".

Untuk operasi penjumlahan modulo nilai identitas adalah 0. Sehingga untuk setiap $a \in Z_m$ nilai invers *a* (-*a*) adalah *m* – *a*. Sebagai contoh pada Z_7, -2 adalah 7 – 2 = 5 dan 5 + 2 ≡ 0 (mod 7) [2]. Notasi yang digunakan untuk merepresentasikan invers perkalian bilangan a adalah a^{-1}.

Tidak semua elemen pada Z_m memiliki invers perkalian. Secara matematika dapat dibuktikan elemen yang memiliki invers perkalian hanya elemen yang merupakan bilangan prima relatif terhadap *m*, yaitu untuk $a \in Z_m$ berlaku *gcd(ϕ(n), e).*

Salah satu cara untuk mencari nilai invers adalah dengan menggunakan algoritma *extend euclid* yang dapat menemukan nilai *s* dan *t* yang memenuhi persamaan "*s x m + t x a = gcd(m, a)"*. Karena gcd(*m*, *a*) = 1 dan *s* x *m* ≡ 0 (mod *m*) maka persamaan tersebut menjadi *t* x *a* = 1 atau *t* x *a* ≡ 1 (mod *m*) sehingga $t = a^{-1}$. Sehingga dapat digambarkan algortima *extend euclid* sebagai berikut :

Tabel 3.6 Algoritma Extend Euclid

	Algoritma Extend Euclid
1	A ← e ; B ← b;
2	T1 ← 0 ; T2 ← 1;
3	**While** B > 0 **do**

```
4     Q ← A/B
5     R ← A – Q * B
6     A ← B ; B ← R
7     T ← T1 – Q * T2;
8     T1 ← T2 ; T2 ← T
9   Endwhile
10  Return t ← T1
```

Berikut contoh untuk mencari nilai *d* dengan nilai *e* = 73 dan $\phi(n)$ = 480 menggunakan algoritma *extend euclid* :

Tabel 3.7 Contoh mencari nilai invers (e^{-1})

A	B	Q	R	T1	T2	T
480	73	6	42	0	1	-6
73	42	1	31	1	-6	7
42	31	1	11	-6	7	-13
31	11	2	9	7	-13	33
11	9	1	2	-13	33	-46
9	2	4	1	33	-46	217
2	1	2	0	-46	217	-480
1	0			**217**	-480	

Dari hasil perhitungan diatas didapatkan nilai T1 = 217 dimana *t* = T1 maka *t* = 217. Nilai *t* merupakan nilai yang didapatkan dari 73^{-1} pada Z_{480} sehingga dapat disimpulkan bahwa nilai *d* = *t* yang berarti *d* = 217. Untuk nilai *qInv* juga dilakukan dengan cara yang sama dengan mencari nilai *d*.

Dengan teorema invers didapatkan nilai d yaitu d = 217.

6. Mencari nilai dP dan dQ (tahap 6 dan 7)

Nilai dP dan dQ merupakan bagian kunci yang digunakan untuk proses dekripsi (kunci privat). Untuk mencari nilai dP dan dQ cukup mengikuti rumus yang ada pada algoritma. Berikut adalah proses mencari dP dan dQ jika nilai yang digunakan adalah nilai yang telah ditetapkan pada contoh sebelumnya :

dP = d mod (p - 1)

= 217 mod (17 – 1)

= 9

dQ = d mod (q – 1)

= 217 mod (31 – 1)

= 7

7. Mencari nilai qInv (tahap 8)

Nilai qInv sama dengan dP dan dQ karena merupakan bagian dari kunci private. Nilai ini akan digunakan pada proses dekripsi. Untuk mencari nilai qInv kita menggunakan teorema invers seperti yang ada pada pencarian nilai d. Dengan nilai yang telah ditetapkan pada contoh sebelumnya maka akan kita cari nila qInv sebagai berikut :

qInv = q^{-1} pada Z_p

Dengan rumus yang digunakan maka akan kita dapatkan nilai qInv = 11. Sehingga sampai tahapan ini kita telah mendapatkan kunci privat K_{privat} = (dP, dQ, qInv, P, Q) = (9, 7, 11, 17, 31).

Semua tahapan proses pembangkit kunci diatas menghasilkan kunci sebagai berikut :

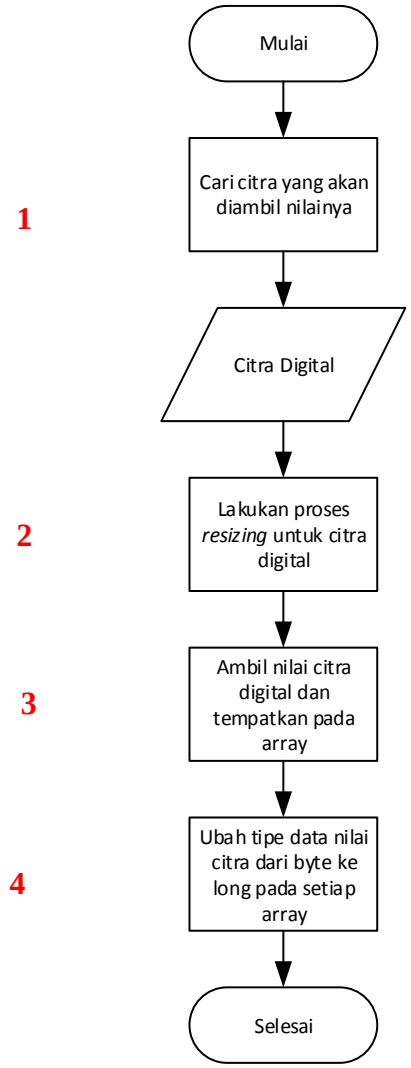
Kunci publik : (73, 527)

Kunci privat : (9, 7, 11, 17, 31)

3.1.3.2 Analisis Citra Digital

Sebelum proses enkripsi dilakukan maka pihak perusahaan harus menyediakan terlebih dahulu citra yang akan dilakukan enkripsi. Proses ini merupakan penggambaran proses yang terjadi pada bagan 3.3 tahap 2.

Citra ini dapat berformat .jpg dan .bmp dengan pembatasan ukuran yaitu ukuran terbesar lebar dan tinggi yaitu 1037 x 384 piksel. Berikut merupakan proses pengolahan citra sehingga citra siap digunakan pada proses dekripsi :



Bagan 3.5 Flowchat pengambilan nilai citra digital

Sebagai gambaran akan digunakan sebuah citra dengan ukuran 360 x 273 pixel seperti berikut :



Bagan 3.6 Citra digital ukuran 360 x 273

Proses ke 1 dan 4 pada bagan 3.5 dilakukan pada proses komputasi. Proses ini lebih dititik beratkan pada proses implementasi yaitu pengkodean.

Proses ke 2 pada bagan 3.5 untuk melakukan fungsi *resizing* terhadap citra digital digunakan algoritma berikut ini :

Tabel 3.8 Algortima Resizing citra digital

	Algoritma Resizing citra digital
1	LebarAsli ← lebargambar
2	TinggiAsli ← tinggigambar
3	Persenlebar ← 1037 / LebarAsli
4	Persentinggi ← 384 / TinggiAsli
5	**If** (persentinggi < persenlebar) **then**
6	Persen ← persentinggi
7	**Else**
8	Persen ← persenlebar
9	**endif**
10	Lebarbaru ← LebarAsli * Persen
11	Tinggibaru ← TinggiAsli * Persen

Proses ke 3 pada bagan 3.5 untuk mengambil nilai dari suatu digital dibutuhkan fungsi khusus pada pemrograman yang digunakan. Hal ini dikarenakan hanya proses pengambilan nilai digital suatu citra digital hanya bisa dilakukan dengan proses komputasi. Proses komputasi ini sangat bergantung aturan setiap bahasa pemrograman yang digunakan.

Berdasarkan analisis yang dilakukan untuk bahasa C# dapat digunakan fungsi *Image.Save(Stream, Format)*. Fungsi ini akan menyimpan semua nilai dari citra digital yang berisi header hingga semua piksel yang terdapat didalam image tersebut. Semua nilai akan disimpan dalam bentuk stream atau antrian sementara didalam memori yang selanjutnya kita konversi ke bentuk array.

Dari hasil analisis yang dilakukan untuk citra pada bagan 3.6 didapatkan 53.292 nilai yang tersebar dalam array 1 dimensi. Penggunaan array 1 dimensi sangat sesuai dengan data yang ada. Hal ini melihat nilai yang didapatkan tidak sesuai jika bibentuk ke dalam array 2 dimensi.

Jika array 2 dimensi merujuk pada *square tesselation* maka kita membutuhkan array dengan kolom dan baris yang sama. Hal ini berarti bahwa dibutuhkan nilai kuadrat sehingga ukuran kolom dan baris sama. Contoh bahwa terdapat ukuran matriks 2 dimensi berukuran 8 x 8 atau 8 kuadrat maka akan bisa ditampung data sebanyak 64 data. Berlaku sebaliknya yaitu jika kita mempunyai 100 data maka kita memerlukan kolom x baris yaitu $\sqrt{100}$ atau matriks berukuran 10 x 10.

Perhitungan kolom x baris pada matriks dua dimensi untuk data citra digital adalah sebagai berikut :

Nilai = 53.292 data

Kolom = baris

Kolom = $\sqrt{53.292}$

= 230,85

≈ 231

Untuk banyak kolom tidak bisa menggunakan bilangan pecah sehingga dibulatkan ke atas menjadi 231. Dari hasil ini maka didapatkan ukuran matriks 231 x 231. Karena melalui proses pembulatan maka tentu akan menyisakan tempat untuk nilai yang banyak. Berikut perhitungan dari array yang tersisa :

Sedangkan yang dibutuhkan sebanyak 53.292. Hal ini berarti ada 53361 – 53929 = 69 array tidak terpakai. Hal ini berarti bahwa jika menggunakan tipe data long yang memiliki kapasitas pemesanan memori sebanyak 4 byte maka akan ada 276 byte dari memori yang tidak digunakan.

Berdasarkan analisis dan perhitungan maka yang paling sesuai adalah dengan menyimpan nilai pada array satu dimensi. Dan penyimpanan ini pun tidak

Kolom x baris = 231 x 231 = 53361 array yang terbentuk

melanggar aturan mengingat *square tesselation* hanya untuk memberikan gambaran nilai dari suatu file citra.

Dari 53292 data yang didapatkan akan kita gunakan 10 data terakhir dari citra untuk menggambarkan proses enkripsi dan dekripsi. Berikut adalah nilai yang diambil :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
		217
	53292	← array ke

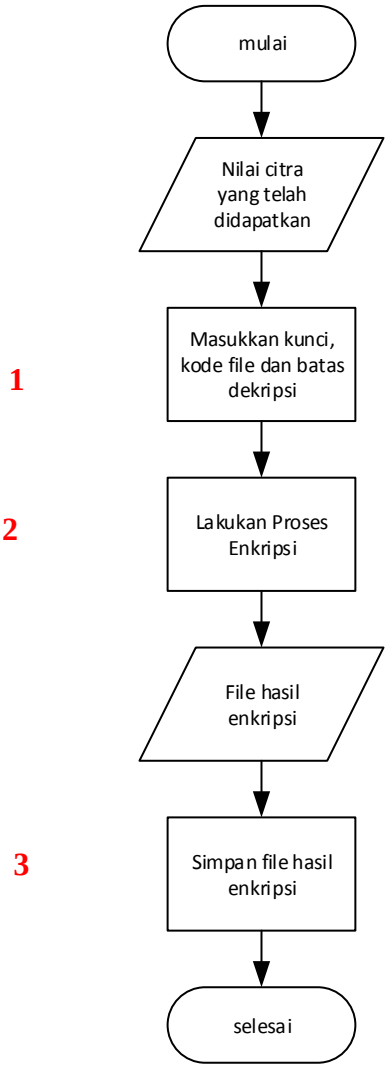
Bagan 3.7 Nilai citra bagan 3.6

3.1.3.3 Analisis Proses Enkripsi

Proses enkripsi ini mewakili proses ke 3 pada bagan 3.3. Proses ini dilakukan oleh pihak perusahaan untuk menyandikan gambar sehingga hasilnya gambar tidak bisa dibuka dengan aplikasi pembuka citra digital.

Proses yang dilakukan untuk melakukan enkripsi dengan metode RSA – CRT ini sama dengan melakukan enkripsi pada RSA biasa. Hal ini karena teorema CRT hanya digunakan pada proses dekripsi yang berakibat pada proses pembangkit kunci juga.

Secara umum proses enkripsi yang terjadi dapat terlihat seperti pada flowchart berikut ini :



Bagan 3.8 Flowchart proses enkripsi

Proses 1 dan 2 tergambar pada algoritma enkripsi RSA –CRT. Sedangkan untuk proses 3 lebih dikhususkan pada tahapan implementasi karena proses penyimpanan hanya bisa dilakukan melalui proses komputasi. Untuk algortima pada proses enkripsi cukup sederhana. Hal ini bisa terlihat pada tabel 3.9 berikut :

Tabel 3.9 Algoritma Enkripsi RSA - CRT

	Algoritma Enkripsi RSA – CRT
1	**Input** : $K_{publik} = (e, n)$, $P \in Z_n$
2	**Output** : $C \in Z_n$
3	$C = P^e \bmod n$

Dari algoritma diatas terlihat bahwa P merupakan masukan untuk dilakukan proses enkripsi. Dalam kasus kriptografi pada citra ini nilai P merupakan masukan berupa nilai *byte array* yang didapatkan dari suatu file citra.

Sebagai contoh kita ambil gambar berekstensi .jpg berukuran 360 x 273 sebagai berikut :



Bagan 3.9 Citra jpg 360 x 273 yang akan dienkripsi

Selanjutnya adalah kita ambil nilai dari citra digital tersebut yang akan kita tampilkan dalam bentuk sampel yang digambarkan dalam bentuk array satu dimensi. Dan dalam hal ini akan kita ambil 10 nilai terakhir dari citra tersebut :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
	53292	217 ← Array ke

Bagan 3.10 Nilai gambar bagan 3.11

Selanjutnya nilai tersebut yang akan kita lakukan enkripsi. Dari proses pembentukan kunci telah kita dapatkan nilai *e* dan *n* yang merupakan bahan untuk melakukan enkripsi. Nilai e = 73 dan nilai n = 527 akan kita masukkan kedalam algoritma kriptografi dengan sample nilai dari tabel 3.11.

Dalam sistem enkripsi RSA dilakukan dengan menggunakan sistem eksponensial modular ($C = P^e \bmod n$). Pada persamaan tersebut memungkinkan proses yang memakan waktu dan memori yang besar ketika nilai *P* yang dipangkatkan dengan nilai yang *e* besar. Untuk mempercepat proses pada sistem eksponensial modular tersebut dilakukan dengan menggunakan algoritma *square and multiply* [2].

Algoritma *square and multiply* mengasumsikan eksponen *e* dalam bentuk biner sehingga dapat direpresentasikan sebagai berikut :

$$e = \sum_{i=0}^{l-1} x_i 2^i$$

Dengan *l* adalah panjang *x* dalam biner. Dengan x_i bernilai 0 atau 1, $0 \leq l \leq l - 1$. Adapun untuk algortima *square and multiply* adalah sebagai berikut :

Tabel 3.10 Algoritma square and multiply

	Algoritma Square and Multiply
1	**Input** : C, e, n
2	**Output** : $y = C^e \bmod n$
3	y = 1
4	**For** i = l – 1 **downto** 0 **do**
5	$y = y^2 \bmod n$
6	**If** e_i = 1 **then**
7	y = (y x C) mod n
8	**Endif**
9	**Endfor**
10	**Return** y

Untuk membuktikan dan memberikan gambaran lebih jelas dari algoritma diatas maka diberikan contoh dengan mencari *y* dari $55^{22} \bmod 7$ sebagai berikut :

Representasi biner 22 adalah 10110 dengan *e* = 5 sehingga perhitungan eksponensial $5^{22} \bmod 7$ adalah :

Tabel 3.11 Contoh eksponensial modular

i	x_i	z
4	1	$1^2 \times 5 \equiv 5 \bmod 7$
3	0	$5^2 \equiv 4 \bmod 7$
2	1	$4^2 \times 5 \equiv 3 \bmod 7$
1	1	$3^2 \times 5 \equiv 3 \bmod 7$
0	0	$3^2 \equiv 2 \bmod 7$

Oleh karena itu terlihat bahwa $5^{22} \equiv 2 \bmod 7 = 2$.

Dengan menggunakan algoritma enkripsi dan memakai prinsip eksponensial modular dalam proses perhitungannya didapatkan nilai citra yang terenkripsi sebagai berikut :

Array ke →	53282	205
		320
		175
		1
		205
		320
		175
		112
		391
	53292	217 ← Array ke

Bagan 3.11 Nilai citra terenkripsi

Dari nilai diatas sangat terlihat perubahan nilai yang sangat signifikan yang melampui batas tipe data *byte* yaitu 255. Sehingga jika kita akan melakukan proses enkripsi sebaiknya ubah tipe data menjadi tipe data *long*. Selain itu juga seperti yang telah dibahas pada analisis citra digital bahwa dengan nilai yang melebihi 255 kita tidak akan bisa melihat citra tetapi yang kita bisa lihat hanya sebuah file yang berisi nilai saja dan file tersebut akan kita beri ekstensi .jpg.

Pada proses ini pula akan ditambahkan dua buah indeks array untuk menampung kode file dan batas dekripsi dari citra terenkripsi tersebut. Kedua nilai tersebut digunakan untuk membatasi proses dekripsi sehingga proses dekripsi hanya bisa dilakukan sebatas yang diinginkan pihak perusahaan.

Kode file digunakan untuk menamakan file yang menyimpan riwayat batas dekripsi. File ini akan disimpan ditempat dimana aplikasi ini berada dan dibentuk dalam bentuk *hidden file* agar mengurangi resiko dihapus oleh pengguna.

Kita misalkan bahwa dengan kode file HTH dan batas dekripsi 2 kali dekripsi akan kita masukkan ke dalam aaray yang baru ditambahkan maka array baru yang terbentuk adalah sebagai berikut :

Array ke →	53282	205		
		320		
		175		
		1		
		205		
		320		
		175		
		112		
		391		
		217	53292	← Array ke
Penyisipan kode file	HTH →	HTH	53293	
Penyisipan pembatas dekripsi	2 →	2	53294	

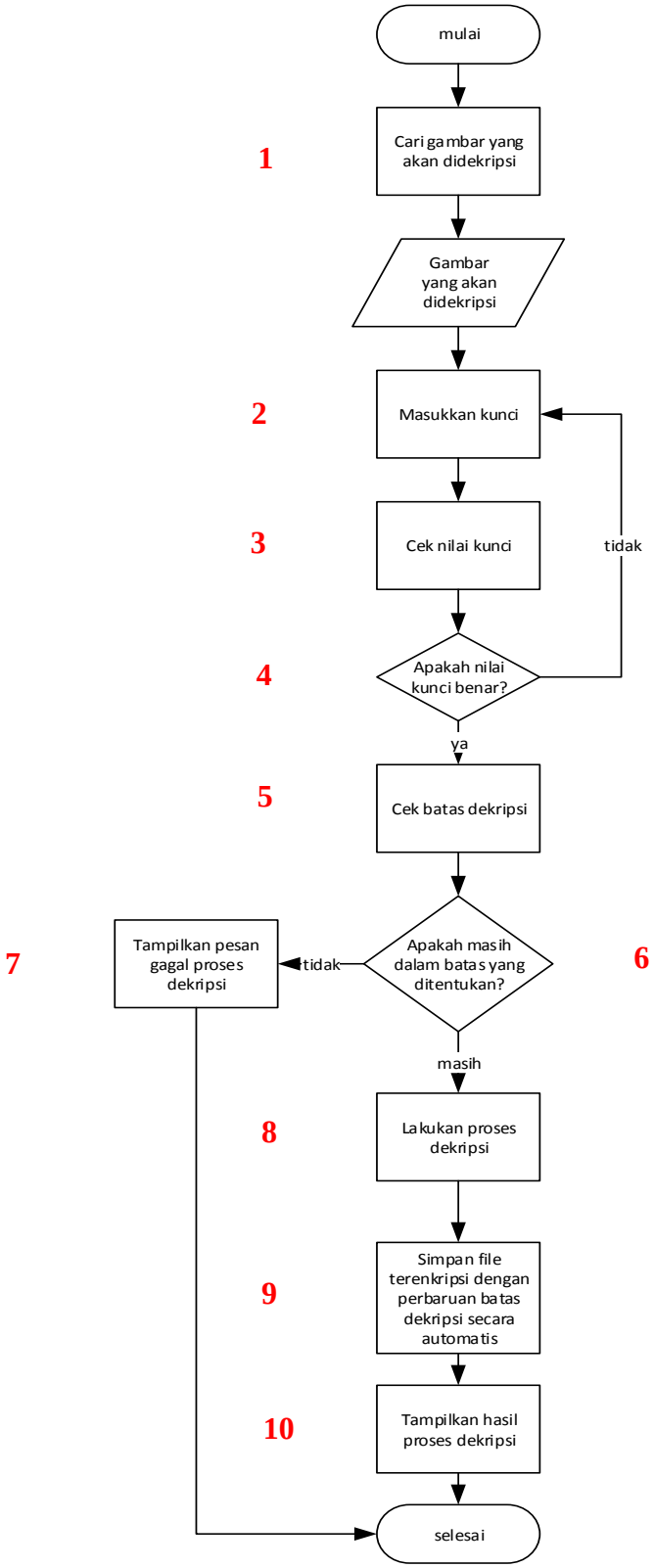
Bagan 3.12 Nilai citra terenkripsi setelah penyisipan kodefile dan batas dekripsi

Nilai pembatas dekripsi ini disimpan dalam file citra itu sendiri dan file terpisah (file yang muncul akibat proses dekripsi). Pada file citra nilai tersebut akan terus terkurangi yang selanjutnya akan disimpan kembali (*rewrite*) di file citra terenkripsi sehingga file terenkripsi mempunyai nilai pembatas dekripsi yang baru.

Selain itu juga untuk mengatasi penggunaan file citra terenkripsi yang belum di-*rewrite* setelah nilai dekripsi habis maka dibuatlah file bayangan yang berfungsi untuk melakukan penyimpanan nilai dekripsi dari file tersebut.

3.1.3.4 Analisis Proses Dekripsi

Proses dekripsi merupakan proses membalikan nilai yang sudah terenkripsi dari suatu *chipertext* ke bentuk semula dalam bentuk *plaintext* dalam kasus ini akan diubah ke citra semula. Alur proses dekripsi dapat terlihat pada flowchart berikut ini :



Bagan 3.13 Flowchart proses dekripsi

Adapun algoritma dari proses dekripsi dapat terlihat sebagai berikut :

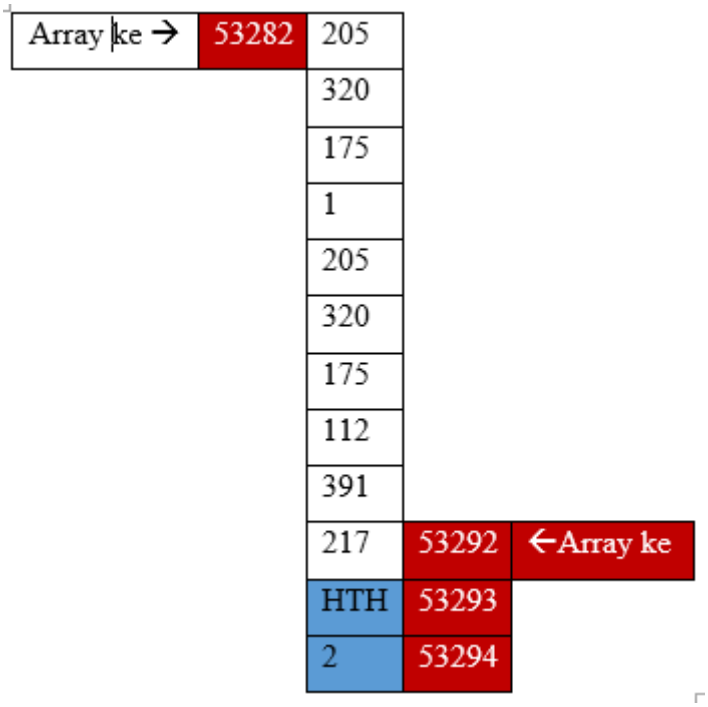
Tabel 3.12 Algoritma Proses Dekripsi RSA - CRT

	Algoritma Dekripsi
	Input : $C = P^e \bmod n$, $K_{privat} = (dP, dQ$
	Output : P
1	$m_1 = C^{dP} \bmod p$
2	$m_2 = C^{dQ} \bmod q$
3	$h = qInv\ (m_1 - m_2) \bmod p$
4	$P = m_2 + h.q$

Terlihat bahwa untuk mencari nilai m_1 dan m_2 digunakan eksponensial modular seperti yang telah dibahas pada proses emkripsi. Untuk membuktikan algoritma tersebut kita akan membuktikan dengan mendekripsikan nilai C = 2942 yang merupakan hasil enkripsi dari nilai asli P = 777. Kunci privat yang digunakan adalah K_{privat} = (dP, dQ, qInv, p, q) = (5, 17, 76, 83, 71). Berikut rincian proses dekripsi yang dilakukan :

a. $m_1 = C^{dP} \bmod p = 2942^5 \bmod 83 = 30$
b. $m_2 = C^{dQ} \bmod q = 2942^{17} \bmod 71 = 67$
c. $h = qInv * (m_1 - m_2) \bmod p = 76 * (30 - 67) \bmod 83 = 10$
d. $P = m_2 + (h * q) = 67 + (10 * 71) = 777$

Dengan proses yang sama dengan yang ada dicontoh maka akan kita lakukan pula terhadap file terenkripsi yang dijelaskan pada proses enkripsi tabel 3.13. Hal ini dilakukan untuk memberikan gambaran nyata proses dekripsi yang akan dilakukan. Berikut merupakan gambaran dari proses dekripsi yang akan dilakukan :



Bagan 3.14 citra terenkripsi yang akan dilakukan proses dekripsi

$K_{private}$ = (9, 7, 11, 17, 31)

Dengan nilai P = 205 pada larik ke 53282 maka akan terlihat proses berikut

a. $m_1 = C^{dP} \bmod p = 205^9 \bmod 17 = 1$

b. $m_2 = C^{dQ} \bmod q = 205^7 \bmod 31 = 7$

c. $h = qInv * (m_1 - m_2) \bmod p = 11 * (1 - 7) \bmod 17 = 2$

d. $P = m_2 + (h * q) = 7 + ((2) * 31) = 69$

Terlihat bahwa nilai P hasil dekripsi adalah 69 dan sesuai nilai asli citra pada array citra asli pada larik 53282 (bagan 3.17). Lakukan hal yang sama pada nilai citra terenkripsi lainnya. Sehingga menghasilkan nilai hasil dekripsi sebagai berikut :

Array ke →	53282	69
		20
		80
		1
		69
		20
		80
		7
		255
	217	53292 ←Array ke

Bagan 3.15 Nilai citra setelah proses dekripsi

Pada proses ini juga kurangi nilai batas dekripsi sehingga nilai citra terenkripsi yang baru sebagai berikut :

Array ke →	53282	205	
		320	
		175	
		1	
		205	
		320	
		175	
		112	
		391	
		217	53292 ← Array ke
		HTH	53293
Penyisipan pembatas dekripsi yang baru	2 – 1 →	1	53294

Bagan 3.16 Nilai citra terenkripsi yang baru

3.1.3.5 Deskripsi Aplikasi yang Akan Dibangun

Aplikasi kriptografi pada citra ini merupakan aplikasi yang dapat digunakan untuk melakukan penyamaran gambar pada citra digital sehingga gambar tidak dapat dibuka pada aplikasi pembuka citra digital. Selain itu juga aplikasi ini dapat melakukan pembatasan bagi klien untuk melihat gambar yang ada.

Aplikasi ini akan diberikan pihak perusahaan pada klien dengan beberapa cara yaitu dengan cara transfer menggunakan media penyimpanan seperti flashdisk, fasilitas berbagi file seperti dropbox dan melalui pengiriman pesan seperti email.

Dengan konsep aplikasi desktop yang dapat berjalan disemua komputer bersistem operasi windows maka ditetapkan beberapa hal sebagai berikut :

1. Ukuran jendela aplikasi

   Aplikasi ini berukuran 1280 x 600 piksel dengan mode *default* yaitu maksimum tampilan layar dan dapat diperkecil maupun dilakukan minimimasi.

2. Alat input

   Alat input yang digunakan yaitu keyboard dan penunjuk pointer seperti *mouse* dan *touchpad.*

3. Citra yang diolah

   Untuk citra yang ditampilkan berukuran maksimum lebar atau tinggi yaitu 1080 piksel.

4. Warna jendela aplikasi

   Warna jendela aplikasi yang digunakan adalah sebagai berikut :

Putih	Abu-abu	Biru Muda	Biru

Bagan 3.17 Warna jendela aplikasi

5. Huruf

   Huruf yang digunakan adalah jenis *Times New Roman*

3.1.4 Analisis Spesifikasi Kebutuhan Perangkat Lunak (SKPL)

Analisis spesifikasi kebutuhan perangkat lunak merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini berisi deskripsi dari kebutuhan perangkat lunak yang akan dibangun baik kebutuhan non fungsional maupun kebutuhan fungsional.

3.1.4.1 Kebutuhan Non Fungsional

Analisis spesifikasi kebutuhan non fungsional berisi batasan-batasan dari layanan-layanan dan fungsi-fungsi dari sebuah sistem, seperti batasan waktu, batasan dari pengembangan proses, serta batasan pengguna.

Tabel 3.13 Kebutuhan non fungsional

Nomor	Spesifikasi Kebutuhan Non Fungsional
SKPL-NF-001	Sistem ini dibangun berbasis desktop.
SKPL-NF-002	Sistem ini bisa digunakan dilaptop dengan spesifikasi rendah

3.1.4.2 Kebutuhan Fungsional

Analisis spesifikasi kebutuhan fungsional berisi pernyataan dari layanan sistem yang harus disediakan, bagaimana sistem harus berjalan terhadap input tertentu dan bagaimana sistem harus berperilaku dalam situasi tertentu. Analisis spesifikasi kebutuhan fungsional menjelaskan fungsionalitas dari sistem serta menjelaskan layanan dari sistem secara detail.

Tabel 3.14 Kebutuhan Fungsional

User Recuirement	
Nomor	**Spesifikasi Kebutuhan Fungsional**
SKPL-F-001	Klien dapat merasa aman bahwa gambar hanya dilihat oleh pihak klien
SKPL-F-002	Perusahaan dapat memberikan batasan klien untuk melihat gambarnya
System Recuirement	
Nomor	**Spesifikasi Kebutuhan Fungsional**
SKPL-F1-001	Klien dapat membuat kunci publik dan kunci kunci privat
SKPL-F1-002	Gambar dienkripsikan oleh aplikasi dengan menggunakan kunci publik
SKPL-F1-003	Gambar hanya bisa didekripsikan oleh klien dengan kunci privat
SKPL-F2-006	Ditambahkan nilai pembatas dekripsi pada file terenkripsi

3.1.5 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional dilakukan untuk mengetahui spesifikasi kebutuhan untuk aplikasi. Spesifikasi kebutuhan melibatkan analisis perangkat keras / *hardware*, analisis perangkat lunak / *software*, analisis pengguna / *user*.

3.1.5.1 Analisis Kebutuhan Perangkat Keras

Berdasarkan hasil pengamatan yang ada didapatkan beberapa data perangkat keras yang tersedia di perusahaan *Art and Design House The House* adalah sebagai berikut:

Tabel 3.15 Ketersediaan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	6 Buah Komputer	Hardisk 500 GB
		Prosesor 2,5 GB
		VGA 4 GB
		Memory RAM 2 GB
		Monitor LCD dengan resolusi 1024x768
		Keyboard Standard port USB
		Mouse Standard port USB
		Lan Card 100Mbps
2	Switch	8 Chanel / 100 Mbps

Sedangkan untuk kebutuhan aplikasi yang akan dibangun bisa dilihat seperti yang tertera pada analisis sistem berikut ini :

Tabel 3.16 Minimum spesifikasi perangkat keras

No	Perangkat Keras	Spesifikasi
1	Komputer	Hardisk 256 GB
		Prosesor 1 GB
		VGA 512 GB
		Memory RAM 512 GB

No	Perangkat Keras	Spesifikasi
		Monitor LCD dengan resolusi 1024x768
		Keyboard Standard port USB
		Mouse Standard port USB

Berdasar hasil analisis yang dilakukan maka dapat disimpulkan bahwa perangkat keras yang tersedia di perusahaan *Art and Design House The House* sudah sangat memadai untuk menjalankan aplikasi kriptografi yang akan dibangun dan sepesifikasi ini pun berlaku untuk setiap klien dari perusahaan ini. Dengan spesifikasi ini diharapkan dengan menggunakan netbook aplikasi ini pun bisa dijalankan sehingga tidak akan mempersulit pihak perusahaan dan klien dalam mengoperasikannya.

3.1.5.2 Analisis Perangkat Lunak

Berdasarkan pengamatan lapangan yang dilakukan didapatkan bahwa sudah terdapat beberapa perangkat lunak yang dimiliki sebagai berikut :

Tabel 3.17 Ketersediaan perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows 7 dan Windows 8.1

Sedangkan berdasar analisis perangkat lunak digunakan untuk menjalankan aplikasi ini maka yang dibutuhkan adalah sebagai berikut :

Tabel 3.18 Minimum spesifikasi perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Microsoft Windows Xp

Berdasar hasil analisis yang dilakukan maka dapat disimpulkan bahwa perangkat lunak yang tersedia di perusahaan *Art and Design House The House* sudah sangat memadai untuk menjalankan aplikasi kriptografi yang akan dibangun

dan sepesifikasi ini pun berlaku untuk setiap klien dari perusahaan ini. Dengan spesifikasi perangkat lunak yang sederhana ini diharapkan tidak mempersulit pihak perusahaan dan klien dalam menggunakan aplikasi ini.

3.1.5.3 Analisis Pengguna

Berdasarkan pengamatan lapangan yang dilakukan didapatkan bahwa sudah terdapat beberapa data pengguna yang dimiliki sebagai berikut :

Tabel 3.19 Tabel ketersediaan pengguna

Pengguna	Tugas	Kemampuan
CEO	1. Bertanggung Jawab terhadap jalannya organisasi 2. Mengatur Pegawai 3. Memantau semua aktifitas proyek	1. Mampu mengatur tugas pegawai 2. Mampu menggunakan komputer 3. Mampu membuat laporan 4. Mampu menganalisis proyek yang diambil
Chief Finance Officer and Accounting	1. Mengatur keuangan perusahaan dari segi pemasukan, pengeluaran, penggajian dan lain – lain. 2. Sebagai sekretaris perusahaan yang mengatur jadwal kegiatan seperti *meeting* dengan pihak luar 3. Mencatat pinjaman dan tagihan peminjam	1. Mampu menggunakan Komputer 2. Mampu mengatur jadwal dan mempersiapkan semua dokumen yang dibutuhkan
Chief Marketing and Account Director	1. Mencari klien untuk menggunakan jasa perusahaan	1. Mampu menggunakan komputer 2. Cakap dalam berbicara

Pengguna	Tugas	Kemampuan
	2. Menentukan strategi pemasaran produk 3. Memberikan respon terhadap masukan klien	3. Mampu mengatasi masalah nasabah
Vp Built Environment	1. Membuat rancang bangunan dan desain interior	1. Mampu menggunakan Komputer 2. Mampu mendesain bangunan dan ruangan
Vp Identity and Branding	1. Membuat rancangan suatu logo mapun desain *branding* perusahaan	1. Mampu menggunakan Komputer 2. Mampu mendesain logo dan identitas suatu perusahaan
Vp Cultural Strategy	1. Membuat penelitian tentang desain yang dihubungkan dengan kultur 2. Membuat sosialisasi dan pameran mengenai desain dan kultur	1. Mampu menggunakan Komputer 2. Mampu membuat ide penelitian dan sosialisasi terhadap hasil penelitian

Analisis pengguna bertujuan untuk mengetahui siapa saja yang bisa menjalankan aplikasi ini. Hal ini agar aplikasi dapat berjalan seperti yang diharapkan. Berikut analisis pengguna :

Tabel 3.20 Minimum Spesifikasi Pengguna

Pengguna	Kebijakan	Tingkat Keterampilan (Minimal)
CEO / *Vp Built Environment* / *Vp*	Melakukan proses enkripsi terhadap gambar	a. Mengusai komputer

Pengguna	Kebijakan	Tingkat Keterampilan (Minimal)
Identity and Branding / *Vp Cultural Strategy*		b. Mengerti penggunaan aplikasi
Klien (pihak diluar perusahaan)	1. Membangkitkan kunci publik dan kunci privat 2. Melakukan dekripsi terhadap citra terenkripsi	a. Mengusai komputer b. Mengerti penggunaan aplikasi

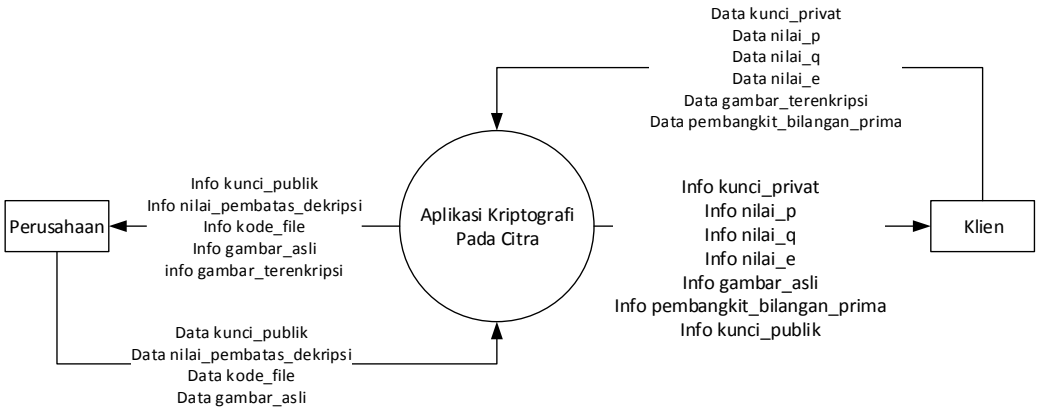
Berdasarkan analisi yang dilakukan maka didapatkan beberapa kesimpulan yaitu pengguna yang terdapat di *Art and Design House The House* telah memenuhi syarat semua. Sehingga tidak perlu untuk mencari tenaga kerja yang baru lagi. Selain itu juga untuk pihak klien tidak ada spesifikasi khusus, sedangkan untuk menguasai penggunaan aplikasi ini akan disertakan tutorial untuk menggunakan aplikasi ini.

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional bertujuan untuk menganalisis proses yang akan diterapkan dalam sistem dan menjelaskan kebutuhan yang diperlukan agar aplikasi dapat berjalan dengan baik.

3.1.6.1 Diagram Konteks

Diagram konteks berfungsi untuk menggambarkan hubungan antara entitas luar, masukkan dan keluaran sistem, yang direpresentasikan dengan lingakaran tunggal yang memiliki keseluruhan sistem. Diagram Konteks dari aplikasi kriptografi dengan metode RSA - CRT di *Art and Design House The House* dapat dilihat sebagai berikut :



Bagan 3.18 Diagram konteks

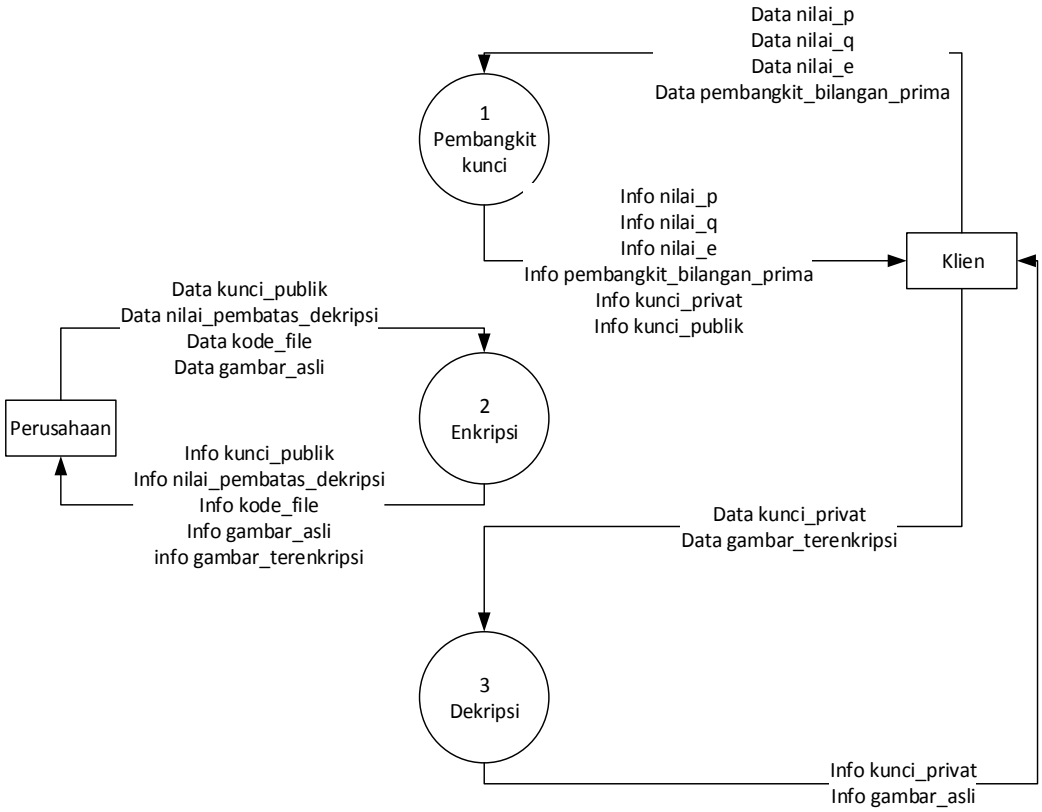
Terlihat bahwa untuk menjalankan aplikasi ini dibutuhkan dua entitas penting yaitu klien dan perusahaan. Meskipun tidak menutup kemungkinan pada kenyataannya aplikasi ini hanya digunakan oleh perusahaan untuk kepentingan pengamanan penyimpanan gambar di perusahaan itu sendiri.

3.1.6.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) memodelkan proses beserta aliran data pada setiap proses yang terjadi dalam sistem dan DFD merupakan *breakdown* dari diagram konteks.

3.1.6.2.1 DFD Level 1

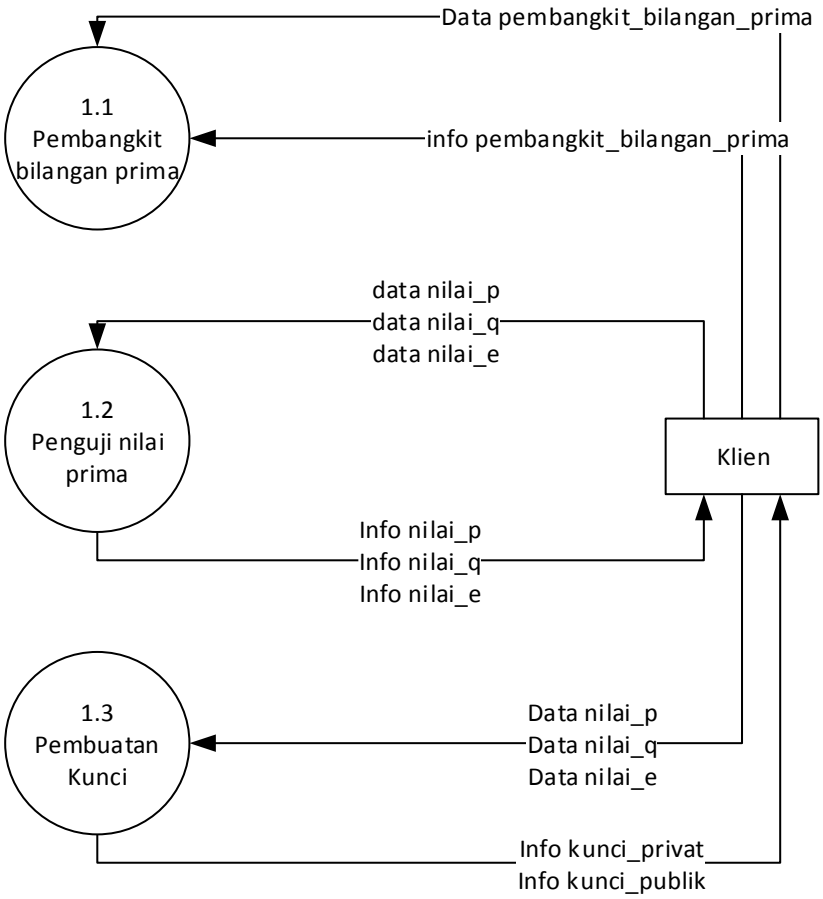
DFD level 1 digunakan untuk memperjelas proses yang ada pada diagram konteks. Berikut gambar dari DFD level 1 :



Bagan 3.19 DFD Level 1

3.1.6.2.2 DFD Level 2 Proses 1

DFD Level 2 proses 1 merupakan proses pembangkit kunci dimana terdapat 3 proses lainnya seperti pada gambar dibawah ini :



Bagan 3.20 DFD Level 2 Proses 1

3.1.6.3 Spesifikasi Proses

Spesifikasi proses adalah penjabaran dari setiap proses yang ada dalam DFD yang dibuat. Berikut pemaparan lebih lengkap :

Tabel 3.21 Spesifikasi Proses

No Urut	Proses	Keterangan
1	No Proses	1.1
	Nama Proses	Pembangkit bilangan prima

No Urut	Proses	Keterangan
	Sumber	Klien
	Input	Data pembangkit_bilangan_prima
	Output	Info pembangkit_bilangan_prima
	Tujuan	Klien
	Logika Proses	a. Masukkan banyak bilangan prima yang akan dibangkitkan (*n*) b. Lakukan perulangan sebanyak 1 sampai 1000000 kali c. Jika bilangan prima yang dibangkitkan sama dengan *n* maka akhiri perulangan d. Jika bilangan prima tidak sama dengan *n* maka simpan bilangan prima dan simpan jumlah bilangan prima yang telah dibangkitkan.
2	No Proses	1.2
	Nama Proses	Penguji nilai prima
	Sumber	Klien
	Input	Data nilai_p, Data nilai_q, Data nilai_e
	Output	Info nilai_p, Info nilai_q, Info nilai_e
	Tujuan	Klien
	Logika Proses	a. Masukkan masing-masing Data nilai_p, Data nilai_q, Data nilai_e ke dalam algoritma pengujian nilai prima b. Jika masing-masing Data nilai_p, Data nilai_q, Data nilai_e merupakan bilangan prima maka data disimpan dalam suatu variabel c. Jika masing-masing Data nilai_p, Data nilai_q, Data nilai_e bukan merupakan

No Urut	Proses	Keterangan
		bilangan prima maka tampilkan pesan bukan bilangan prima
3	No Proses	1.3
	Nama Proses	Pembuatan kunci
	Sumber	Klien
	Input	Data nilai_p, Data nilai_q, Data nilai_e
	Output	Info kunci_privat, Info kunci_publik
	Tujuan	Klien
	Logika Proses	a. Cari nilai *n* dengan mengalikan *p* dan *q* b. Berturut – turut untuk nilai *e* dan *n* akan menghasilkan output kunci publik (info kunci_publik) c. Cari nilai *d* dengan menggunakan algortima *extend euclid* d. Cari nilai *dP* dan *dQ* seperti pada algoritma pembangkit kunci e. Cari nilai *qInv* dengan menggunakan algoritma *extend euclid* f. Berturut – turut untuk nilai *dP, dQ, qInv, p, q* menghasilkan output kunci privat (info kunci_privat)
4	No Proses	2
	Nama Proses	Enkripsi
	Sumber	Perusahaan
	Input	Data kunci_publik, Data nilai_pembatas_dekripsi, Data kode_file, Data gambar_asli

No Urut	Proses	Keterangan
	Output	Info kunci_publik, Info nilai_pembatas_dekripsi, Info kode_file, Info gambar_asli, info gambar_terenkripsi
	Tujuan	Perusahaan
	Logika Proses	a. Masukkan gambar asli, kunci publik, kode file, nilai pembatas dekripsi b. Lakukan pengecekan apakah semua masukan telah terisi c. Jika tidak tampilkan pesan masukan belum terisi d. Jika terisi semua maka lanjutkan ke proses berikutnya e. Ambil semua nilai *byte* dari sebuah gambar f. Konversikan tipe data nilai *byte* tersebut ke nilai integer g. Lakukan enkripsi terhadap nilai *byte* yang telah terkonversi dengan menggunakan algoritma *square and multiply* sehingga menghasilkan nilai terenkripsi yang bertipe data long (*C*) h. Simpan semua nilai *C* ke array string i. Tambahkan dua field array j. Sisipkan kode file ke total array – 1 k. Sisipkan nilai pembatas dekripsi ke total array l. Simpan semua nilai array ke file yang berekstensi .jpg
5	No Proses	3
	Nama Proses	Dekripsi

No Urut	Proses	Keterangan
	Sumber	Klien
	Input	Data kunci_privat, Data gambar_terenkripsi
	Output	Data kunci_privat, Data gambar_asli
	Tujuan	Klien
	Logika Proses	a. Masukkan semua nilai Data kunci_privat, Data gambar_terenkripsi b. Lakukan pengecekan terhadap kunci, jika kunci salah maka tampilkan pesan kunci salah c. Jika benar maka lanjutkan proses dekripsi d. Lakukan pengecekan terhadap nilai pembatas dekripsi e. Jika nilai pembatas dekripsi > 0 maka lanjutkan proses dekripsi dengan mengurangi nilai pembatas pada gambar dan buat file histori dengan nama file adalah kode file serta simpan file histori yang memuat nilai pembatas dekripsi di folder aplikasi (.exe) f. Jika nilai pembatas pada gambar dan histori = 0 maka tampilkan pesan kesalahan dan berhentikan proses dekripsi

3.1.6.4 Kamus Data DFD

Kamus data (*data dictionary*) merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem. Kamus data dibuat berdasarkan arus data yang ada pada diagram aliran data. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir pada sistem secara lengkap. Berikut merupakan kamus data dari DFD diatas :

Tabel 3.22 Kamus data DFD

1	Nama	Data nilai_p
	Digunakan Pada	1. Proses pembangkit kunci – Proses 1 2. Proses penguji bilangan prima – Proses 1.2 3. Proses pembuat kunci – Proses 1.3
	Deskripsi	Data ini bernilai prima dan digunakan untuk membuat kunci privat dan kunci publik
	Struktur Data	1. Nilai_p
	Nilai_p	Bilangan prima [0...n | n > 0]
2	Nama	Data nilai_q
	Digunakan Pada	1. Proses pembangkit kunci – Proses 1 2. Proses penguji bilangan prima – Proses 1.2 3. Proses pembuat kunci – Proses 1.3
	Deskripsi	Data ini bernilai prima dan digunakan untuk membuat kunci privat dan kunci publik
	Struktur Data	1. Nilai_q
	Nilai_q	Bilangan prima [0...n | n > 0]
3	Nama	Data nilai_e
	Digunakan Pada	1. Proses pembangkit kunci – Proses 1 2. Proses penguji bilangan prima – Proses 1.2 3. Proses pembuat kunci – Proses 1.3
	Deskripsi	Data ini bernilai prima dan digunakan untuk membuat kunci privat dan kunci publik
	Struktur Data	1. Nilai_e
	Nilai_e	Bilangan prima [0...n | n > 0]
4	Nama	Data pembangkit_bilangan_prima
	Digunakan Pada	1. Proses pembangkit kunci – Proses 1

		2. Proses pembangkit bilangan prima – proses 1.1
	Deskripsi	Data ini untuk menentukan banyak bilangan prima yang akan dibangkitkan
	Struktur Data	1. Banyak_bilangan_prima
	Banyak_bilangan_prima	[0 ... n | n > 0]
5	Nama	Data Kunci Privat
	Digunakan Pada	1. Pembuat kunci – Proses 1.3 2. Dekripsi – Proses 3
	Deskripsi	Berisi tentang kunci privat
	Struktur Data	1. Nilai_dP 2. Nilai_dQ 3. Nilai_qInv 4. Nilai_p 5. Nilai_q
	Nilai_dP	[0 ... n | n > 0]
	Nilai_dQ	[0 ... n | n > 0]
	Nilai_qInv	[0 ... n | n > 0]
	Nilai_p	[0 ... n | n > 0]
	Nilai_q	[0 ... n | n > 0]
6	Nama	Data kunci_publik
	Digunakan Pada	1. Pembuat kunci – Proses 1.3 2. Enkripsi – Proses 2
	Deskripsi	Berisi tentang kunci publik
	Struktur Data	1. Nilai_e 2. Nilai_n
	Nilai_e	[0 ... n | n > 0]
	Nilai_n	[0 ... n | n > 0]
7	Nama	Data pembatas_nilai_dekripsi
	Digunakan Pada	1. Enkripsi – Proses 2

	Deskripsi	Berisi nilai pembatas untuk melakukan dekripsi
	Struktur Data	1. Pembatas_dekripsi
	Pembatas_dekripsi	[0 ... n | n > 0]
8	Nama	Data kode_file
	Digunakan Pada	1. Enkripsi – proses 2
	Deskripsi	Berisi sebagai identitas dari suatu foto
	Struktur Data	1. Kode_file
	Kode_file	{[A-Z|a-z|0-9]}
9	Nama	Data gambar_asli
	Digunakan Pada	Enkripsi – proses 2 Dekripsi – proses 3
	Deskripsi	Berisi tentang nilai dari suatu foto
	Struktur Data	Array_gambar_asli
	Array_gambar_asli	Byte[]
10	Nama	Data gambar_terenkripsi
	Digunakan Pada	Enkripsi – proses 2 Dekripsi – proses 3
	Deskripsi	Berisi tentang nilai dari suatu foto
	Struktur Data	1. Array_gambar_terenkripsi 2. Nilai_pembatas_dekripsi 3. Kode_file
	Array_gambar_asli	Byte[]
	Nilai_pembatas_dekripsi	[0 ... n | n > 0]
	Kode_file	{[A-Z|a-z|0-9]}

3.2 Perancangan Sistem

Perancangan sistem merupakan tahap dimana perencanaan, penggambaran serta pembuatan aplikasi atau sistem dari penggabungan beberapa elemen menjadi

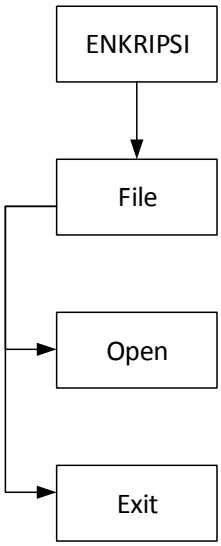
satu bagian yang utuh. Pada tahap ini akan dipaparkan perancangan terhadap struktur menu, antarmuka, pesan dan jaringan semantik.

3.2.1 Perancangan Struktur Menu

Struktur menu merupakan sebuah bentuk umum rancangan program yang bertujuan untuk membantu pengguna dalam menyelesaikan sebuah tugas. Struktur menu untuk aplikasi kriptografi ini meliputi perancangan menu enkripsi dan perancangan menu dekripsi.

3.2.1.1 Perancangan Menu Enkripsi

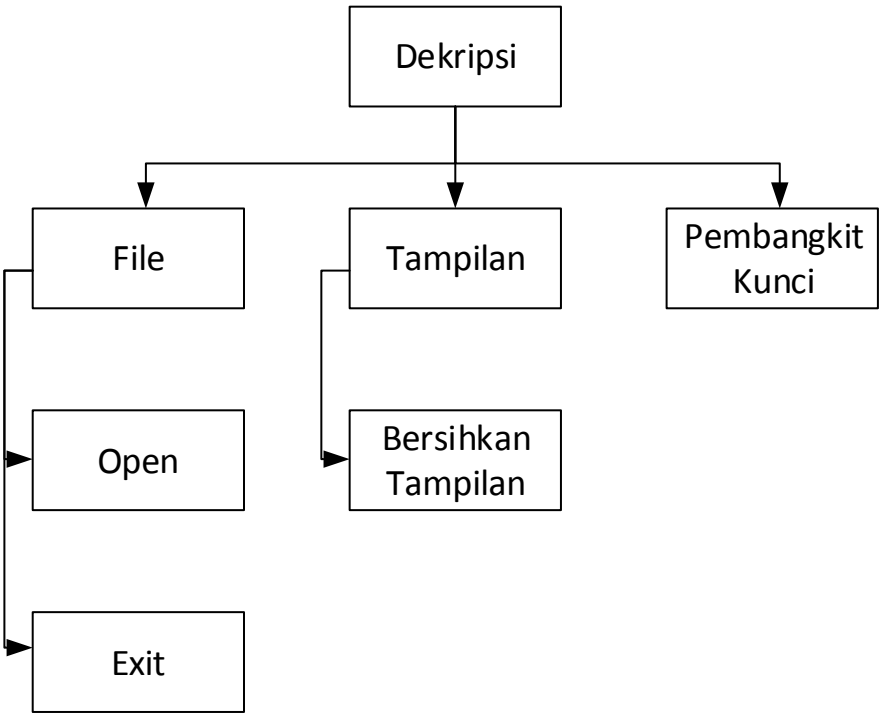
Struktur menu untuk enkripsi bertujuan untuk memudahkan pengguna dalam menggunakan aplikasi ini dan mengurangi penggunaan button pada aplikasi. Berikut struktur menu dari enkripsi :



Bagan 3.21 Perancangan menu enkripsi

3.2.1.2 Perancangan Menu Dekripsi

Struktur menu untuk dekripsi bertujuan untuk memudahkan pengguna dalam menggunakan aplikasi ini dan mengurangi penggunaan button pada aplikasi. Berikut struktur menu dari dekripsi:



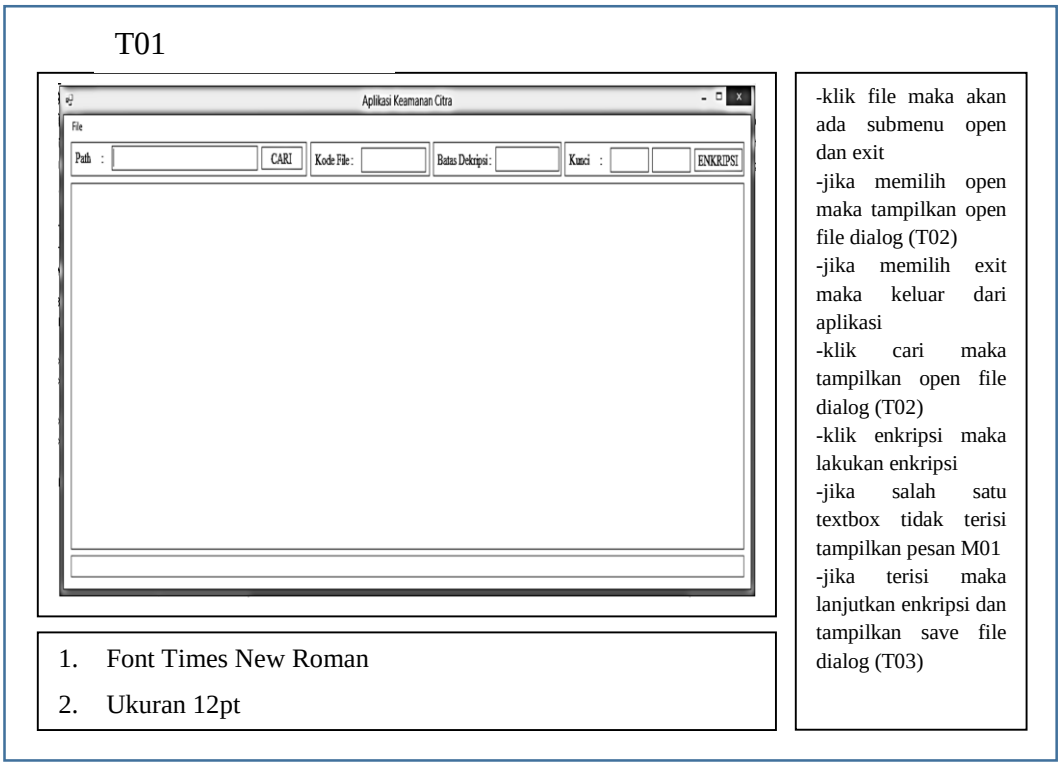
Bagan 3.22 Perancangan arsitektur menu dekripsi

3.2.2 Perancangan Antarmuka

Perancangan antarmuka merupakan sebuah sketsa penggambaran dari penggabungan elemen-elemen yang dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang dibangun. Berikut adalah gambar-gambar dari perancangan antarmuka yang digunakan pada pembangunan aplikasi kriptografi pada citra dengan menggunakan algoritma RSA – CRT di *art and design House The House* :

1. Perancangan antarmuka enkripsi

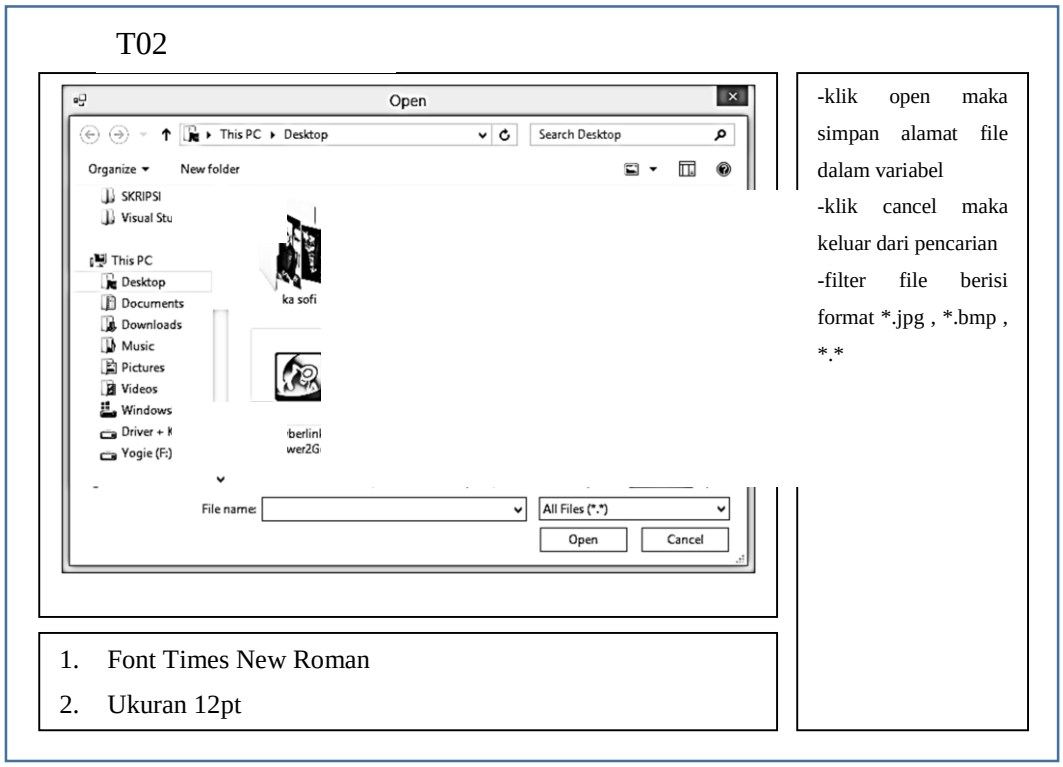
Halaman ini digunakan untuk melakukan proses enkripsi pada gambar. Berikut tampilannya :



Bagan 3.23 Perancangan antarmuka enkripsi

2. **Perancangan antarmuka open file dialog**

Perancangan antarmuka open file dialog sebenarnya sudah tersedia didalam GUI C# visual studio namun dimasukkan ke dalam rancangan agar terlihat lebih jelas gambaran open file dialog. Berikut tampilan open file dialog :

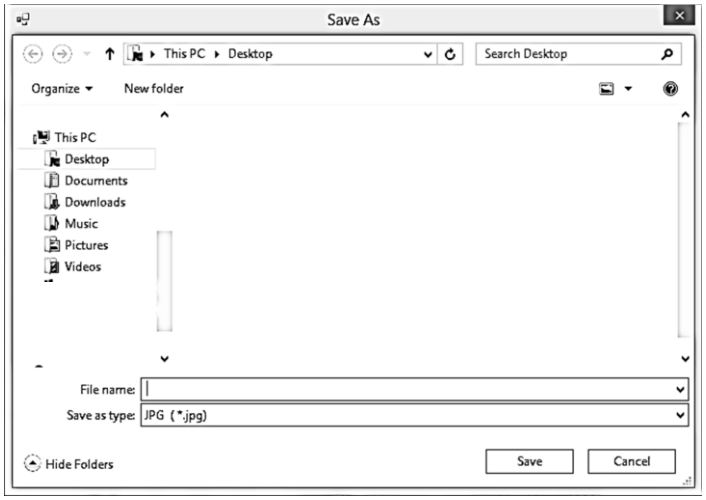


Bagan 3.24 Perancangan antarmuka open file dialog

3. **Perancangan antarmuka save file dialog**

Perancangan antarmuka save file dialog sebenarnya sudah tersedia juga didalam GUI C# visual studio namun dimasukkan ke dalam rancangan agar terlihat lebih jelas gambaran save file dialog. Berikut tampilan save file dialog :

T03



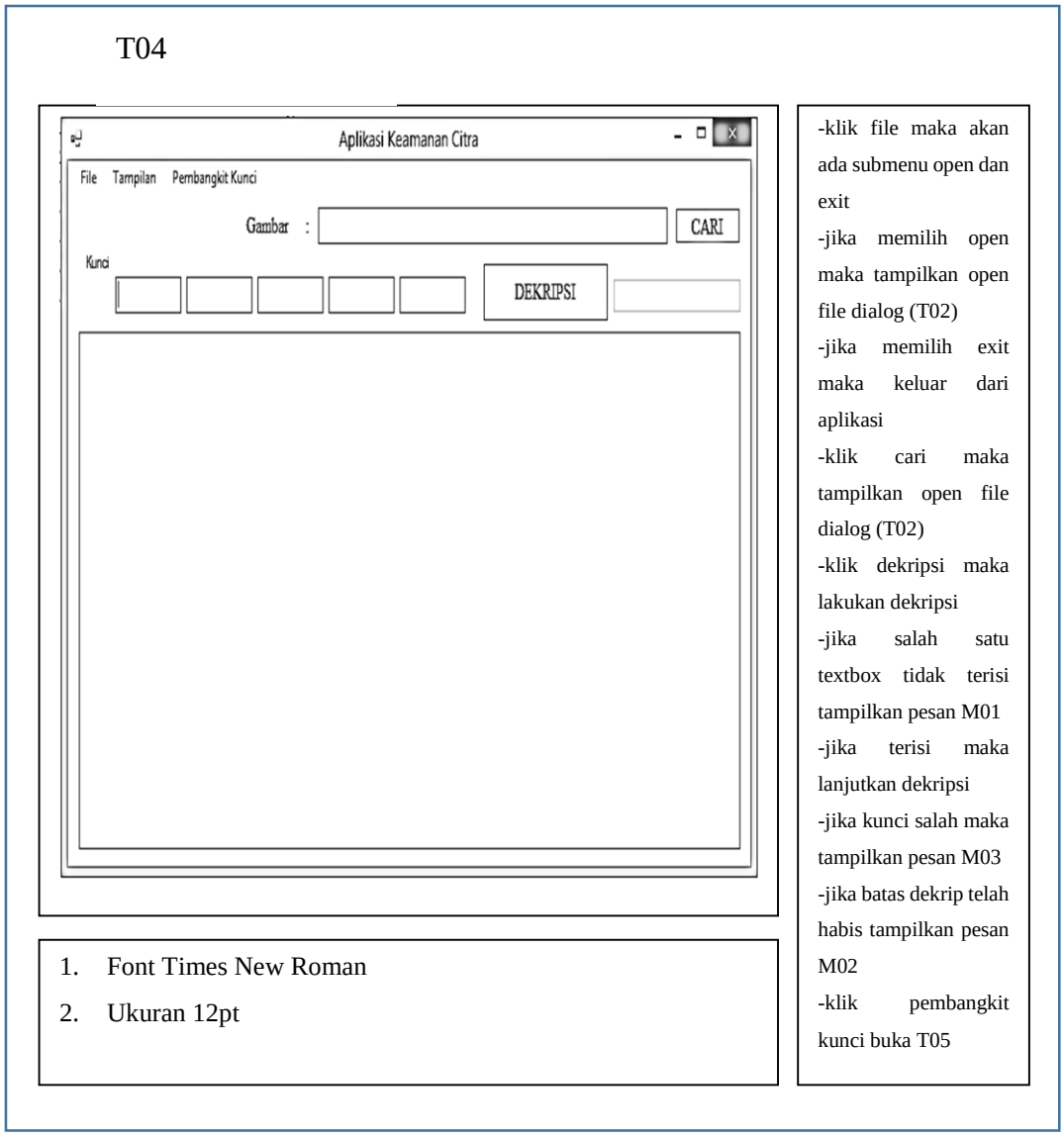
-klik save maka file akan tersimpan dengan format .jpg
-klik cancel maka keluar dari pencarian
-filter file berisi format *.jpg

1. Font Times New Roman
2. Ukuran 12pt

Bagan 3.25 Perancangan antarmuka save dialog

4. **Perancangan antarmuka dekripsi**

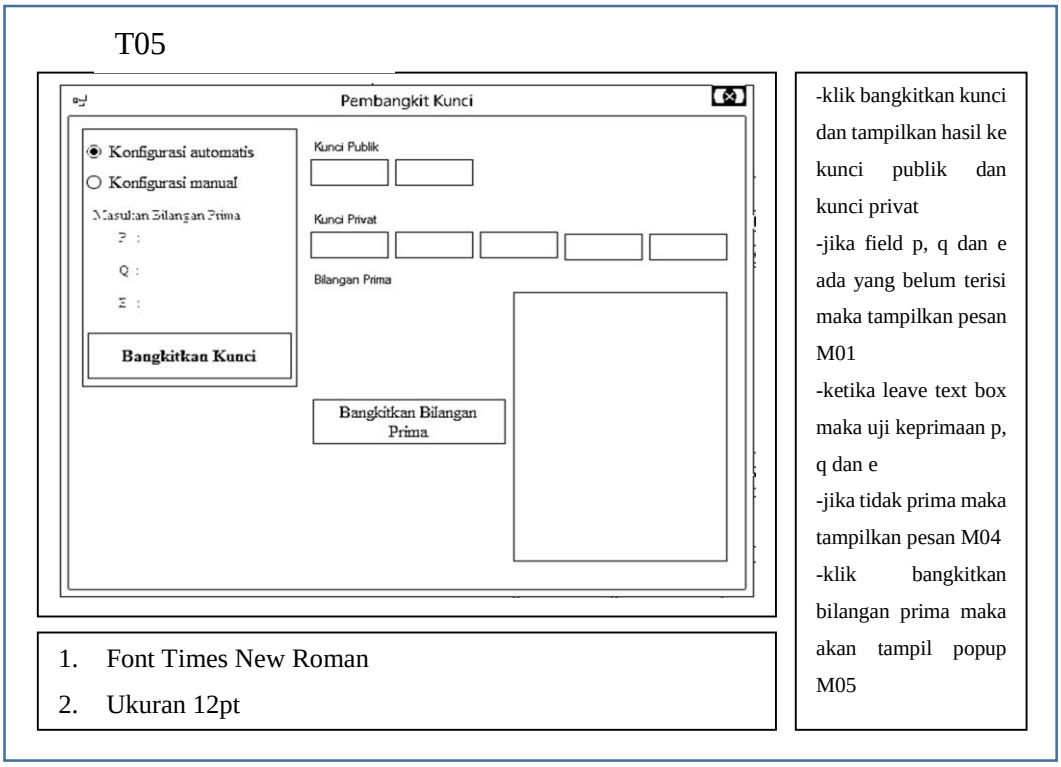
Halaman ini digunakan untuk melakukan proses dekripsi pada gambar. Berikut tampilannya :



Bagan 3.26 Perancangan antarmuka dekripsi

5. **Perancangan antarmuka pembangkit kunci**

Halaman ini digunakan untuk melakukan proses pembangkit kunci untuk kebutuhan enkripsi dan dekripsi. Berikut tampilannya :



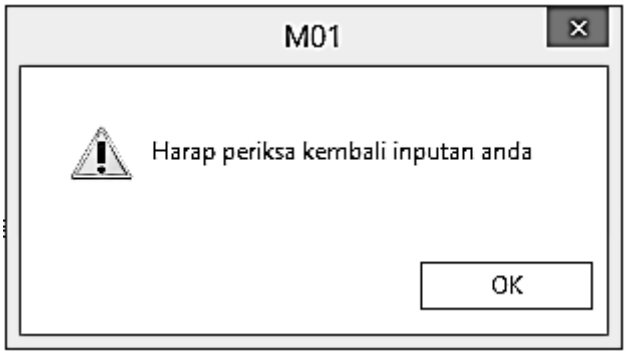
Bagan 3.27 Perancangan pembangkit kunci

3.2.3 Perancangan Pesan

Perancangan pesan merupakan perancangan pesan ketika ada terjadi suatu kesalahan, pemberian informasi ataupun peringatan terhadap proses yang terjadi dalam menjalankan aplikasi. Berikut perancangan pesan dari aplikasi kriptografi pada citra dengan menggunakan RSA – CRT di konsultan *Art and Design House The House* :

1. Perancangan pesan masukan yang kurang

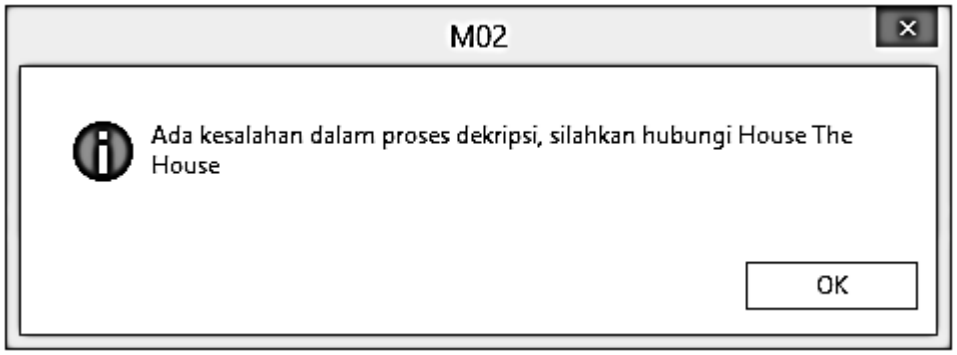
Pesan ini digunakan untuk memberikan peringatan bahwa terdapat masukan yang kurang untuk proses enkripsi, dekripsi dan pembangkit kunci.



Bagan 3.28 Perancangan pesan masukan yang kurang

2. Perancangan pesan dekripsi telah mencapai batas

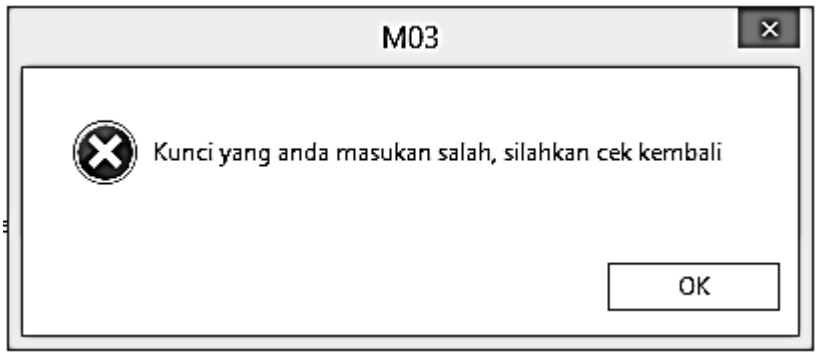
Pesan ini digunakan untuk memberikan informasi kesalahan proses dekripsi yang disebabkan proses dekripsi telah mencapai batas dekripsi yang diizinkan pihak perusahaan.



Bagan 3.29 Perancangan pesan dekripsi telah mencapai batas

3. Perancangan pesan kesalahan kunci dekripsi

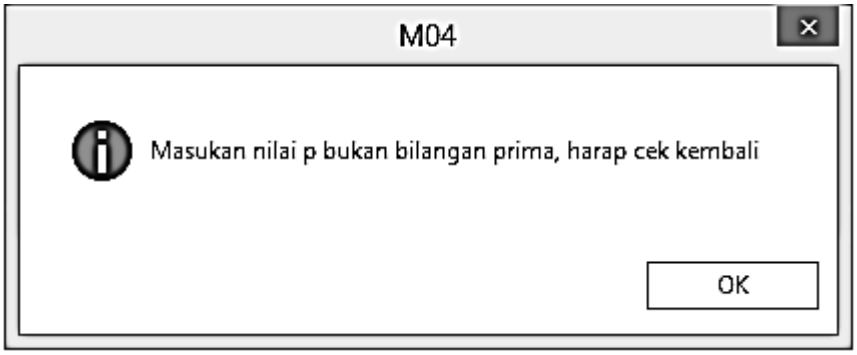
Pesan ini akan muncul ketika dalam proses dekripsi diketahui bahwa terjadi kesalahan kunci sebagai masukan.



Bagan 3.30 Perancangan pesan kesalahan kunci dekripsi

4. Perancangan pesan bukan bilangan prima

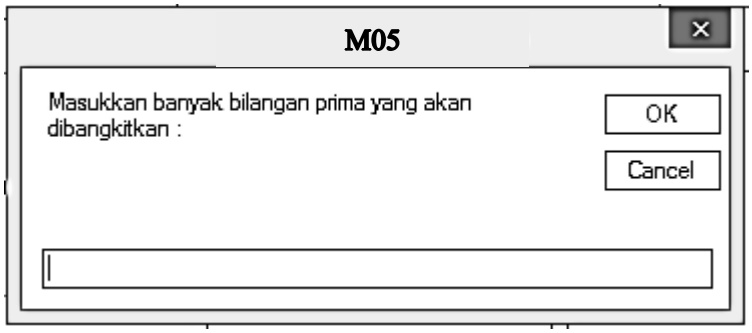
Pesan ini berfungsi memberikan informasi bahwa nilai p, q atau e bukan bilangan prima. Untuk variabel p dalam pesan diganti q atau e bergantung pada variabel mana yang masukannya bukan bilangan prima.



Bagan 3.31 Perancangan pesan bukan bilangan prima

5. Perancangan pesan popup masukan banyak bilangan prima

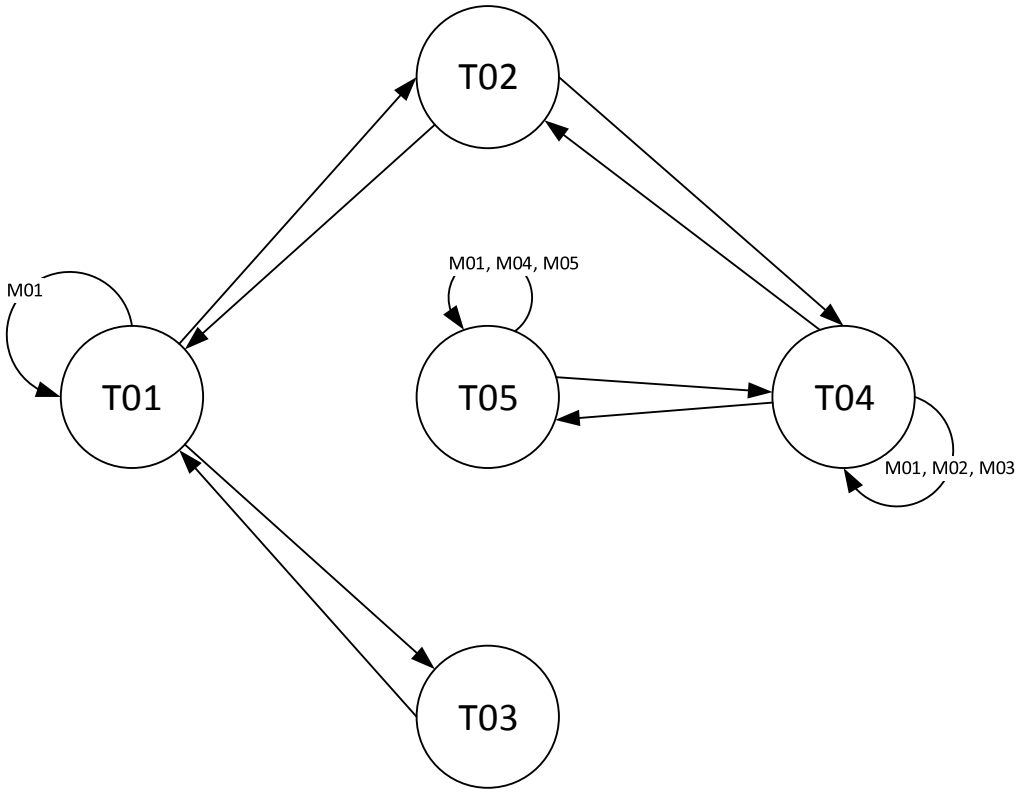
Perancangan pesan ini berfungsi untuk memasukan nilai banyaknya bilangan prima yang akan dibangkitkan.



Bagan 3.32 Perancangan pesan pop up masukan bilangan prima

3.2.4 Jaringan Semantik

Jaringan Semantik merupakan gambaran grafis tentang hubungan antar tampilan dan pesan pada antar muka. Jaringan semantik pada kriptografi dengan metode RSA – CRT dapat terlihat seperti gambar berikut :



Bagan 3.33 Jaringan Semantik Aplikasi Kriptografi dengan RSA – CRT

Keterangan jaringan semantik :

T01 = Form Enkripsi
T02 = Form Open File Dialog
T03 = Form Save File Dialog
T04 = Form Dekripsi
T05 = Form Pembangkit Kunci

M01 = Pesan Masukan yang Kurang
M02 = Pesan Dekripsi Mencapai Batas
M03 = Pesan Kesalahan Memasukan Kunci
M04 = Pesan Masukan Bukan Prima
M05 = Pesan Masukan Bilangan Prima yang Akan Dibangkitkan