BAB III
ANALISIS DAN DESAIN SISTEM

Pada bab ini akan dibahas mengenai Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64yang meliputi analisa sistem dan desain sistem.

III.1. Analisis Masalah

Adapun analisa masalah pada Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64yaitu :

1. Banyaknya pihak-pihak yang melakukan modifikasi file text dan gambar gambar yang bertujuan negatif sehingga merugikan pihak-pihak tertentu.
2. Terjadinya interupsi yang dapat mengganggu ketersediaan file gambar yaitu file gambar yang ada dapat dihapus sehingga pihak yang membutuhkan gambar tersebut tidak dapat menemukan gambar tersebut begitu juga dengan file text.
3. Seringnya terjadi ancaman intersepsi yaitu merupakan ancaman terhadap kerahasiaan file text dan gambar.

III.2. Metode *Base64*

Algoritma Base64 merupakan algoritma yang menggunakan salah satu konsep algoritma enkripsi modern 4 yaitu algoritma Block Cipher yang berupa operasi pada mode bit namun algortma Base64 ini lebih mudah dalam pengimplementasiannya dari algoritma-algoritma yang lainnya . Base64 adalah metoda yang untuk melakukan encoding (penyandian) terhadap data binary menjadi format 6-bit character. Pada algoritma ini, rangkaian bit-bit palainteks dibagi

menjadi blok-blok bit dengan panjang yang sama, biasanya 64 bit yang direpresentasikan dengan karakter ASCII. Base64 menggunakan karakter A – Z, a – z dan 0 – 9 untuk 62 nilai pertama, sedangkan 2 nilai terakhir digunakan symbol (+ dan /). Standar yang penulis gunakan adalah MIME (Multipurpose Internet Mail Extensions)/RFC 1521. RFC ini menegaskan sebuah standar untuk implementasi Base64 terhadap data binary dan melampirkan sebuah karakter padding "=" jika terdapat kekurangan pada byte. Dalam streaming base64, spesifikasi mengharuskan setiap baris menjadi paling banyak 76 basis-64 karakter.

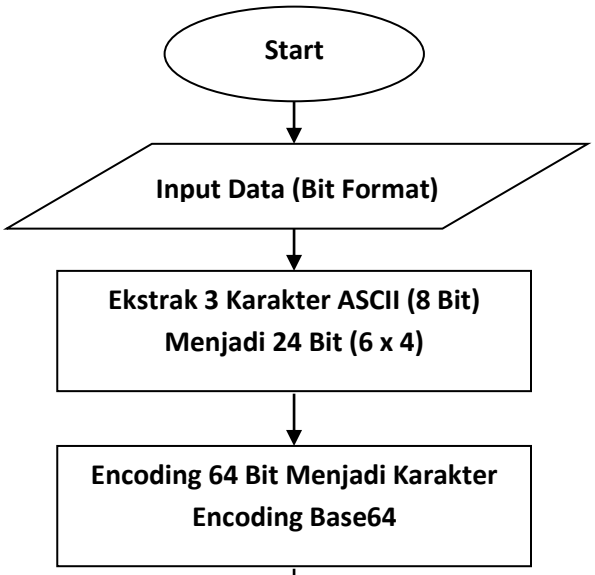
Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Gambar III.1. ***Mapping TableBase64***

III.2.1 Enkripsi

Proses enkripsi meliputi input data (bit format), ekstrak 3 karakter ascii (8 bit) menjadi 24 bit (6 x 4), encoding 64 bit menjadi karakter encoding base64. Data input biner

(yang merupakan hasil ekstrak dari karakter dengan menggunakan tabel ASCII) dimana satu karakter diwakili 8 bits kemudian kumpulan 8 bits tersebut di ekstrak menjadi kumpulan per 6 bits yang mewakili satu karakter yang disusun membentuk 4 bagian per blok (jadi tersusun beberapa blok dimana satu blok terdapat 24 bit data). Untuk kemungkinan jika terdapat data bit yang tidak mencapai 6 bits setelah proses ekstrak tadi, solusinya adalah dengan menambahkan bit 0 pada bit 6 hingga mencapai 6 bit dan bit 1 sisanya hingga mencapai 24 bit pada blok yang tersisa. Setelah kumpulan bit tersusun menjadi 24 bits dimana setiap blok nya dibagi menjadi 4 bagian, yang terdiri dari susunan-susunan 6 bits, barulah dari setiap bagian 6 bits tersebut data-data binary dapat dikonversikan menjadi karakter encoding base64 berdasarkan tabel data Encoding 64 radix diatas dimana satu bagian 6 bits mewakili satu karakter encoding base64. Gambar 1 di bawah ini menunjukkan Diagram Alir proses enkripsi.



Gambar III.2. EnkripsiMetode *Base64*

Dalam tahap enkripsi menggunakan Base64 data yang diinput akan ditransformasikan kedalam ASCII 8 bit.. Berikut adalah contoh proses enkripsi yang akan dilakukan.

Text yang akan di enkripsi adalah “Ini” tanpa tanda kutip.

Text	I	n	i
ASCII	73	110	105
Biner	01001001	01101110	01101001

Setelah diketahui kode binernya maka akan dipecah menjadi 4 blok 6 bit

Text	I		n	i
ASCII	73		110	105
Biner	01001001		01101110	01101001
Biner	010010010110111001101001			
6 bit x 4	010010	010110	111001	101001

Blok tersebut dikembalikan kedalam bentuk decimal dari 4 blok diatas

Text	I		n	i
ASCII	73		110	105
Biner	01001001		01101110	01101001
Biner	010010010110111001101001			
6 bit x 4	010010	010110	111001	101001
Index	18	22	57	41

Mapping dilakukan terhadap index 4 blok terhadap *Mapping TableBase64* yang ada di gambar III.1

Text	I		n	i
ASCII	73		110	105
Biner	01001001		01101110	01101001
Biner	010010010110111001101001			
6 bit x 4	010010	010110	111001	101001
Index	18	22	57	41
Base64	S	W	5	p

Maka hasil dari eknripsi Tes = SW5p

Enkripsi Key

Text yang akan di enkripsi adalah nama file “Ini.txt” tanpa tanda kutip.

Text	**I**	**n**	**i**	**.**	**t**	**x**	**t**
ASCII	**73**	**110**	**105**	**46**	**116**	**120**	**116**
Biner 8 bit	**01001001**	**01101110**	**01101001**	**00101110**	**01110100**	**01111000**	**01110100**

Setelah diketahui kode binernya maka akan dipecah menjadi 4 blok 6 bit, untuk blok yang kurang dari 3 akan ditambahkan 0 seperti berikut

Text	**I**	**n**	**i**	**.**	**t**	**x**	**t**	**kosong**	**kosong**
ASCII	**73**	**110**	**105**	**46**	**116**	**120**	**116**	**000**	**000**
Biner 8	**01001001**	**01101110**	**01101001**	**00101110**	**01110100**	**01111000**	**01110100**	**000000**	**000000**

bit								**00**	**00**
Biner 24 bit	**010010010110111001101001**			**001011100111010001111000**			**011101000000000000000000**		
6 bit x 4	**010010**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**000000**	**000000**

Blok tersebut dikembalikan kedalam bentuk decimal dari 4 blok diatas

Text	**I**	**n**	**I**	**.**	**t**	**x**	**t**	**kosong**	**kosong**
ASCII	**73**	**110**	**105**	**46**	**116**	**120**	**116**	**000**	**000**
Biner 8 bit	**01001001**	**01101110**	**01101001**	**00101110**	**01110100**	**01111000**	**01110100**	**00000000**	**00000000**
Biner 24 bit	**010010010110111001101001**			**001011100111010001111000**			**011101000000000000000000**		
6 bit x 4	**010101**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**kosong**	**kosong**
Index	**18**	**22**	**57**	**41**	**11**	**39**	**17**	**56**	**29**	**0**	**=**	**=**

Mapping dilakukan terhadap index 4 blok terhadap *Mapping TableBase64* yang ada di gambar III.1

Text	**I**	**n**	**i**	**.**	**t**	**x**	**t**	**Koso**	**Kos**

								ng	ong
ASCII	**73**	**110**	**105**	**46**	**116**	**120**	**116**	**000**	**000**
Biner 8 bit	**01001001**	**01101110**	**01101001**	**00101110**	**01110100**	**01111000**	**01110100**	**00000000**	**00000000**
Biner 24 bit	**010010010110111001101001**			**001011100111010001111000**			**011101000000000000000000**		

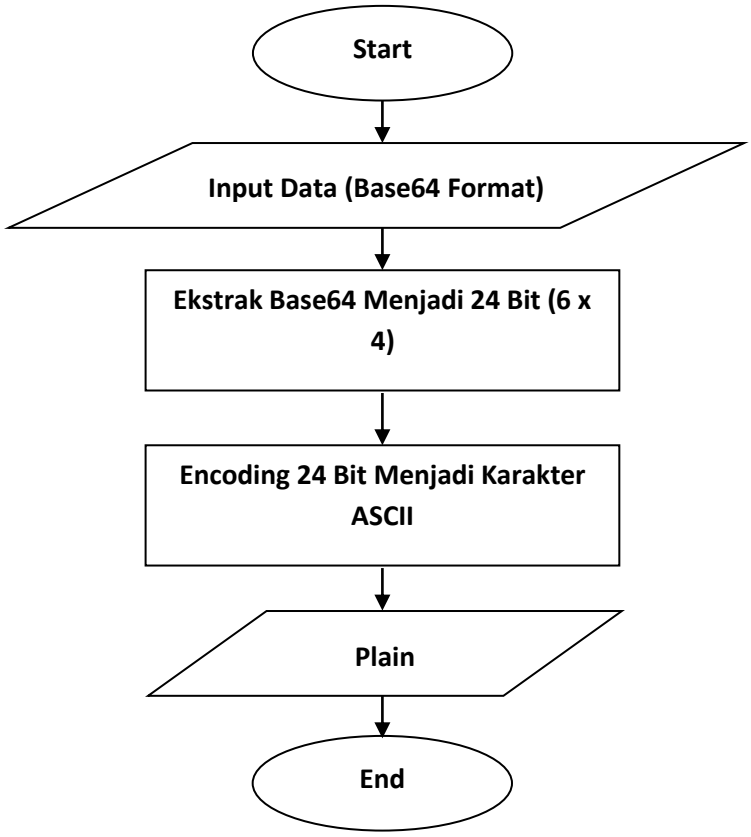
6 bit x 4	**010101**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**kosong**	**kosong**
Index	**18**	**22**	**57**	**41**	**11**	**39**	**17**	**56**	**29**	**0**	**=**	**=**
Base 64	**S**	**W**	**5**	**p**	**L**	**n**	**R**	**4**	**d**	**A**	**=**	**=**

Maka hasil dari eknripsi key dengan nama file Ini.txt = SW5pLnR4dA==

III.2.2 Dekripsi

Proses dekripsi adalah proses kebalikan dari enkripsi dimana proses yang dilakukan terlebih dahulu data karakter base64 hasil encoding pada proses sebelumnya, pertama-tama dengan menggunakan *Mapping TableBase64*, data karakter di ubah menjadi binarydimana satu karakter di wakili oleh 6 bits data yang dikelompokkkan dalam blok yang berisi 24 bits data, setelah semua kumpulan bits data tersusun, kumpulan-kumpulan bits tersebut akan diekstrak menjadi kumpulan 8 bits data, dimana satu blok berisi 24 bits data akan di ekstrak menjadi 3 karakter ASCII 8 bits. Selanjutnya dengan menggunakan tabel ASCII, kumpulan 8 bits data

tersebut di ekstrak menjadi karakter plaintext (8 bits data mewakili satu karakter Plaintext). Untuk lebih jelasnya kita bisa meilhat gambar dibawah ini.



Gambar III.3. DekripsiMetode *Base64*

Dalam tahap dekripsi menggunakan Base64 data yang diinput akan ditransformasikan kedalam ASCII 8 bit.. Berikut adalah contoh proses dekripsi yang akan dilakukan.

Text yang akan di dekripsi adalah “SW5p” tanpa tanda kutip.

Base64	S	W	5	p
Index	18	22	57	41
6 bit x 4	010010	010110	111001	101001

Setelah mendapatkan biner 6 bit maka proses selanjutnya adalah merubah kedalam 8 bit.

Base64	S	W	5	p
Index	18	22	57	41
6 bit x 4	010010	010110	111001	101001
Biner	100100101101110011101001			
Biner 8 bit	10001001	01101110	01101001	

Kemudian kita akan ngambil nilai ASCII dari biner tersebut dan merubah kedalam plain text.

Base64	S	W	5	P
Index	18	22	57	41
6 bit x 4	010010	010110	111001	101001
Biner	100100101101110011101001			
Biner 8 bit	10001001	01101110	01101001	
ASCII	73	110	105	
Text	I	N	i	

Maka hasil dekripsi dari SW5p = Ini

Dekripsi Key

Text yang akan di dekripsi adalah “SW5pLnR4dA==” tanpa tanda kutip.

Base64	**S**	**W**	**5**	**P**	**L**	**n**	**R**	**4**	**d**	**A**	**=**	**=**
Index	**18**	**22**	**57**	**41**	**11**	**39**	**17**	**56**	**29**	**0**	**=**	**=**
6 bit x 4	**010010**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**kosong**	**kosong**

Setelah mendapatkan biner 6 bit maka proses selanjutnya adalah merubah kedalam 8 bit.

Base64	**S**	**W**	**5**	**p**	**L**	**n**	**R**	**4**	**d**	**A**	**=**	**=**
Index	**18**	**22**	**57**	**41**	**11**	**39**	**17**	**56**	**29**	**0**	**=**	**=**

6 bit x 4	**010010**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**kosong**	**kosong**
Biner 24 bit	**010010010110111001101001**						**001011100111010001111000**				**011101000000000000000000**	
Biner 8 bit	**01001001**	**01101110**	**01101001**	**00101110**	**01110100**	**01111000**	**01110100**	**00000000**	**00000000**			

Kemudian kita akan ngambil nilai ASCII dari biner tersebut dan merubah kedalam plain text.

Base64	**S**	**W**	**5**	**p**	**L**	**n**	**R**	**4**	**d**	**A**	**=**	**=**
Index	**18**	**22**	**57**	**41**	**11**	**39**	**17**	**56**	**29**	**0**	**=**	**=**
6 bit x 4	**010010**	**010110**	**111001**	**101001**	**001011**	**100111**	**010001**	**111000**	**011101**	**000000**	**kosong**	**kosong**
Biner 24 bit	**010010010110111001101001**				**001011100111010001111000**				**011101000000000000000000**			
Biner 8 bit	**01010100**	**01100101**	**01110011**	**00101110**	**01110100**	**01111000**	**01110100**	**00000000**	**00000000**			

ASCII	**73**	**110**	**105**	**46**	**116**	**120**	**116**	**000**	**000**
Text	**I**	**n**	**i**	**.**	**t**	**x**	**t**	**Kosong**	**Kosong**

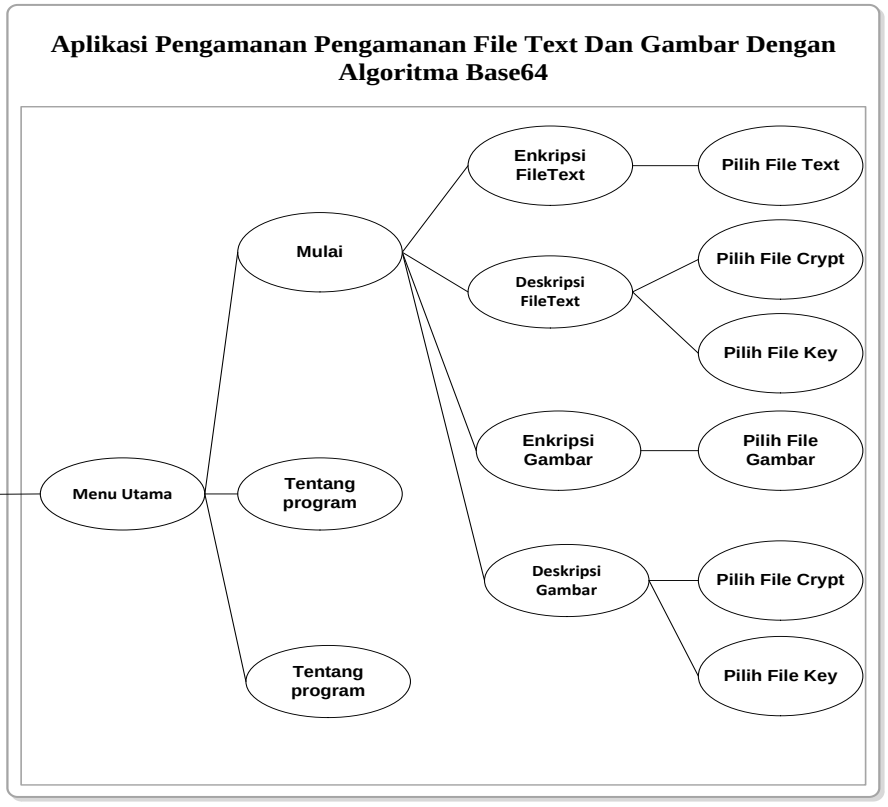
Maka hasil dekripsi dari SW5pLnR4dA = Ini.txt

III.3. Desain Sistem Baru

Desain Sistem Baru menggunakan bahasa pemodelan UML yang terdiri dari *UsecaseDiagram*, *ActivityDiagram*dan *SequenceDiagram*.

III.3.1. *Usecase* Diagram

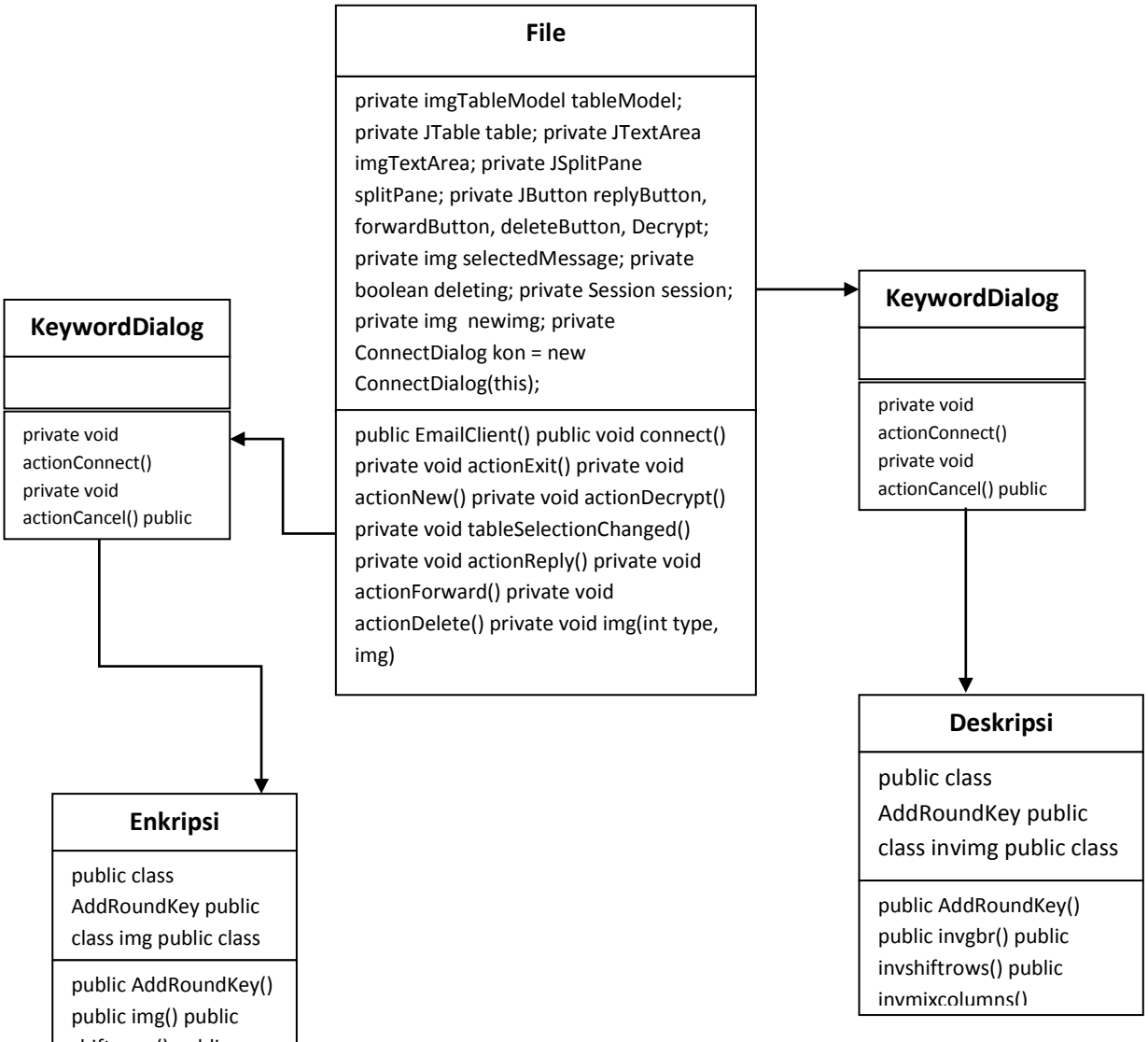
Secara garis besar, proses sistem yang akan dirancang digambarkan dengan *usecase diagram* yang terdapat pada Gambar III.1 :



Gambar III.4.*Use Case* Diagram Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64

III.3.2.*Class* Diagram

Rancangan kelas-kelas yang akan digunakan pada sistem yang akan dirancang dapat dilihat pada gambar dibawah ini :



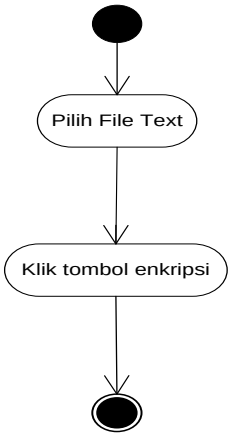
Gambar III.5.*Class Diagram* Diagram Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64

III.3.3.*Activity*Diagram

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

1. ***Activity Diagram*Enkripsi File Text**

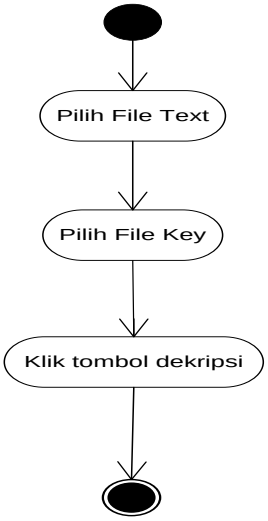
Pada activity diagram Enkripsi File Textmenjelaskan bahwa informasi atau data Enkripsi File Text. Adapun Activity Diagram Enkripsi File Text dapat dilihat pada gambar III.3.



Gambar III.6.*Activity Diagram*Enkripsi File Text

*2. Activity Diagram*Dekripsi File Text

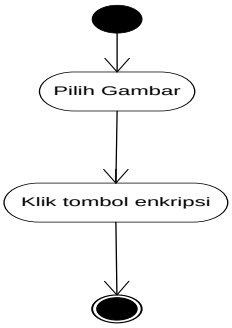
Pada activity diagram Dekripsi File Textmenjelaskan bahwa informasi atau data Dekripsi File Text. Adapun Activity Diagram Dekripsi File Text dapat dilihat pada gambar III.4.



Gambar III.7.*Activity Diagram*Deskripsi File Text

3. *Activity Diagram*Enkripsi Gambar

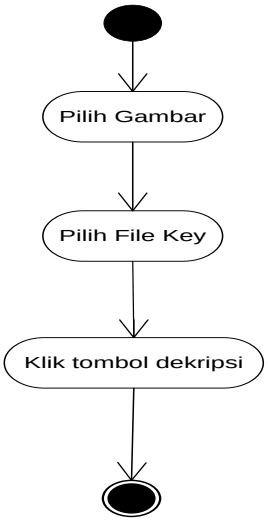
Pada activity diagram Enkripsi Gambarmenjelaskan bahwa informasi atau data Enkripsi Gambar. Adapun Activity Diagram Enkripsi Gambardapat dilihat pada gambar III.3.



Gambar III.8.*Activity Diagram*Enkripsi Gambar

***4. Activity Diagram*Deskripsi Gambar**

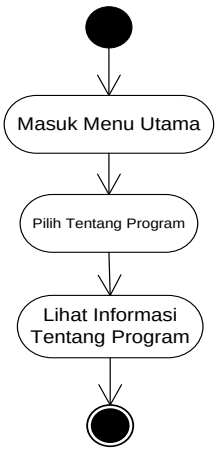
Pada activity diagram Deskripsi Gambarmenjelaskan bahwa informasi atau data Deskripsi gambar. Adapun Activity Diagram Deskripsi Gambardapat dilihat pada gambar III.4.



Gambar III.9.*Activity Diagram*Dekripsi Gambar

5. *Activity Diagram*Melihat Tentang Program

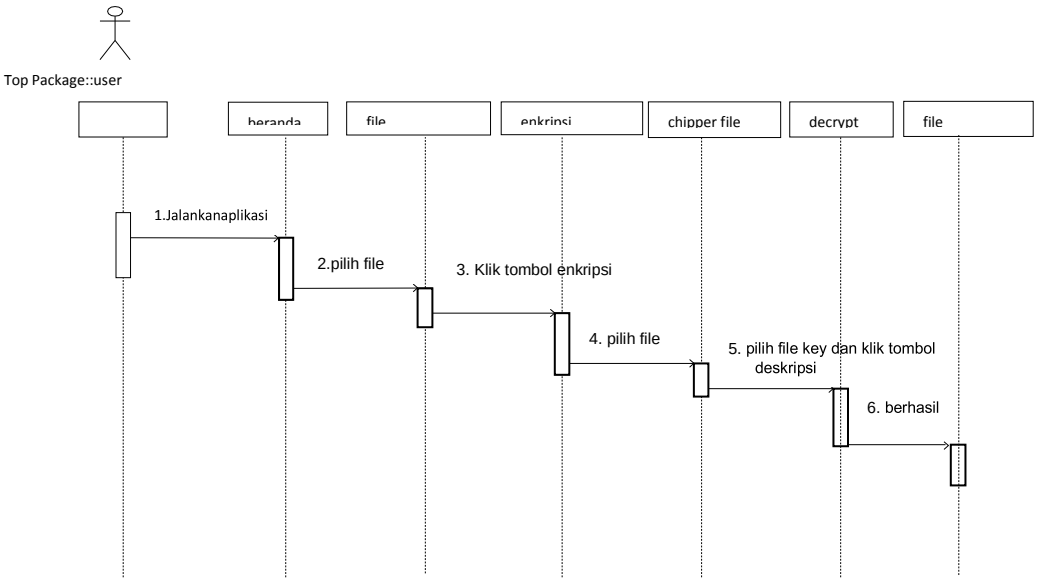
Pada activity diagram*About* menjelaskan bahwa informasi atau data diri pembuat program. Adapun Activity Diagram Tentang Program dapat dilihat pada gambar III.5.



Gambar III.10.*Activity Diagram*Melihat Tentang Program

III.3.4. *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Serangkaian kegiatan saat terjadi *event* pada aplikasi ini dapat dilihat pada gambar III.8:

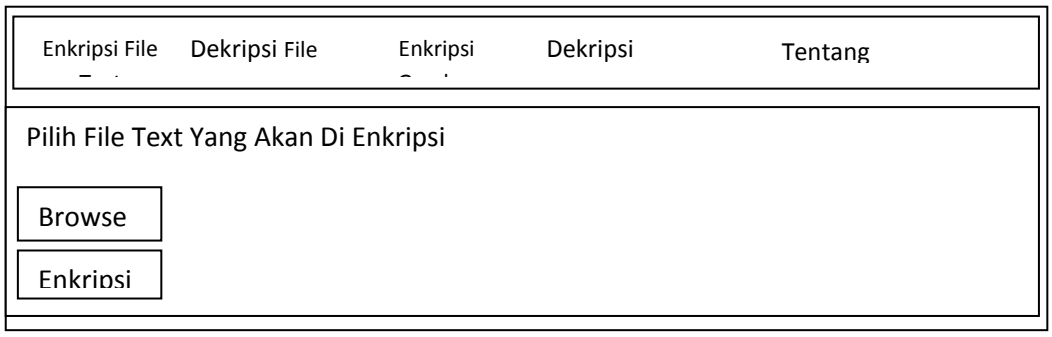


Gambar III.11.*Sequence* Diagram Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64

III.4. DesainUser Interface

1. Rancangan *Form*Beranda

*Form*ini berfungsi untuk menampilkan berandaAplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64, rancangan dapat dilihat pada gambar berikut :



Gambar III.12. Desain Tampilan Beranda

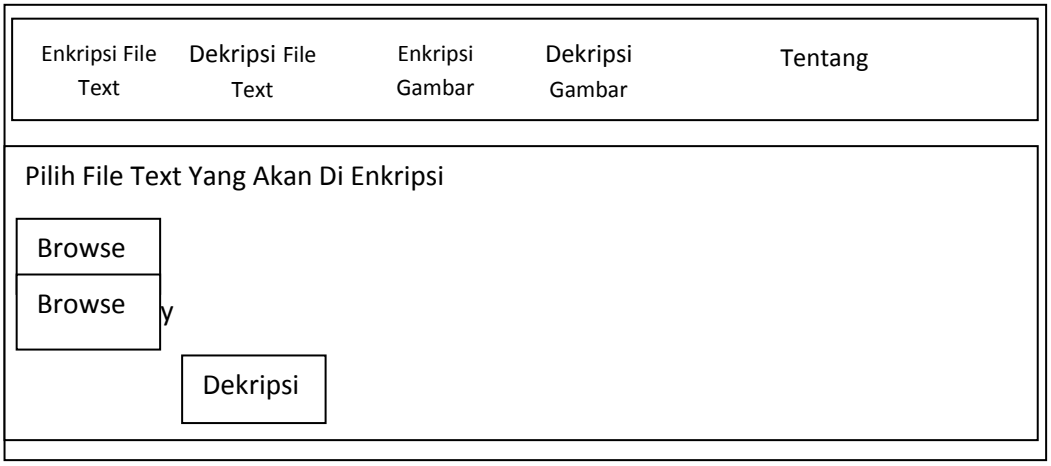
2. Rancangan *Form*Enkripsi

*Form*enkripsi berfungsi untuk menampilkan form enkripsiAplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64, rancangan dapat dilihat pada gambar berikut :

Gambar III.13. Desain Tampilan *Form*Enkripsi

3. Rancangan *Form*Deskripsi

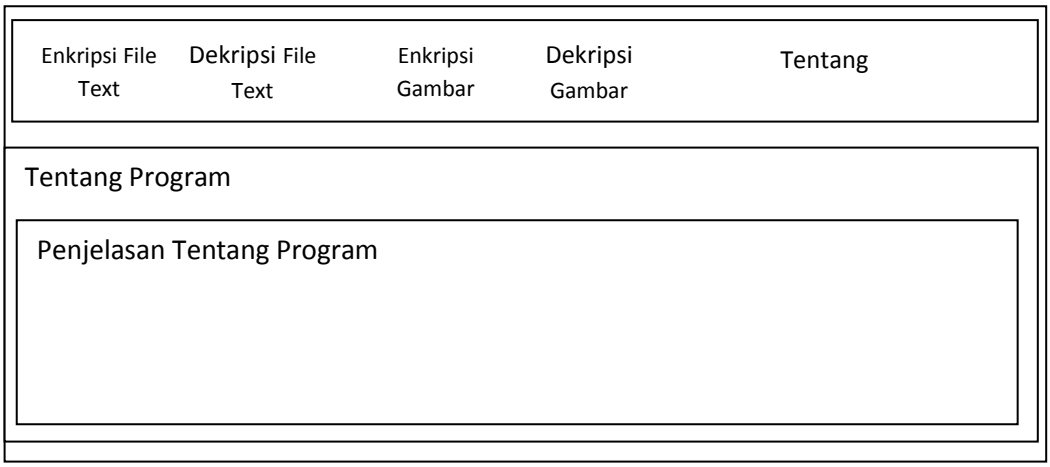
*Form*deskripsi berfungsi untuk menampilkan*form* Deskripsi Aplikasi Pengamanan File Text Dan Gambar Dengan Algoritma Base64,, rancangan dapat dilihat pada gambar berikut :



Gambar III.14. Desain Tampilan *Form*Dekripsi

4. Rancangan *Form*Tentang Program

*Form*ini menjelaskan informasi pembuat Aplikasi Pengamanan Gambar Dengan Teknik Transformasi Menggunakan Metode Arnold Cat Map (ACM), rancangan dapat dilihat pada



Gambar III.15. Desain Tampilan Tentang Program