

BAB IV

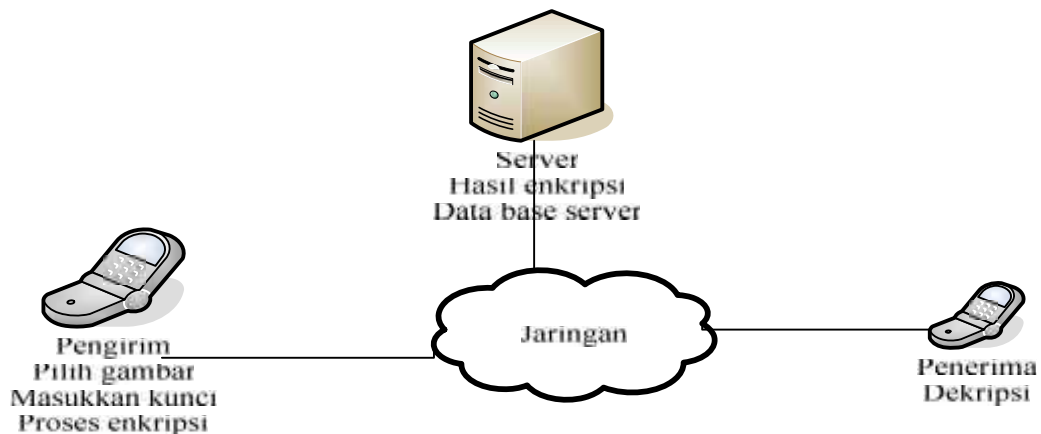
ANALISA DAN PERANCANGAN

4.1 Analisa Deskripsi Umum Sistem

Perangkat lunak yang akan dibangun merupakan perangkat lunak yang diterapkan pada telepon seluler yang bersistem operasi Android dan memiliki fungsi untuk melakukan pengamanan gambar. Perangkat lunak yang akan dibangun harus dapat melakukan pengiriman dan menerima gambar.

Perangkat lunak yang akan dibuat merupakan perangkat lunak yang digunakan untuk berkomunikasi, oleh sebab itu perangkat lunak yang dibuat akan diterapkan pada pengirim dan penerima.

Pengguna akan berinteraksi dengan perangkat lunak melalui *user interface* yang disediakan oleh perangkat lunak, pengguna memilih gambar yang ada dalam *memory*, kemudian pilih gambar tersebut dan disertai kunci yang telah disepakati oleh kedua belah pihak dan kirim ke telepon *seluler* lain dengan jaringan internet. Secara umum *arsitektur* global perangkat lunak dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Arsitektur Global Sistem

4.2. Analisa Sistem

Analisa yang akan digunakan dalam pengembangan aplikasi ini antara lain, analisa pengguna, analisa fitur dan kontent yang akan dibangun, dan analisa fungsional pada pengembangan aplikasi ini.

4.2.1. Analisa Pengguna

Pengguna yang terlibat dalam aplikasi yang akan dibangun adalah: orang umum karena aplikasi yang akan dibangun berfungsi untuk mengamankan gambar. Sehingga *user* biasa pun bisa menggunakannya.

4.2.2. Analisa Fitur dan Isi

Beberapa fitur dan isi yang akan dikembangkan dan diterapkan pada aplikasi ini adalah:

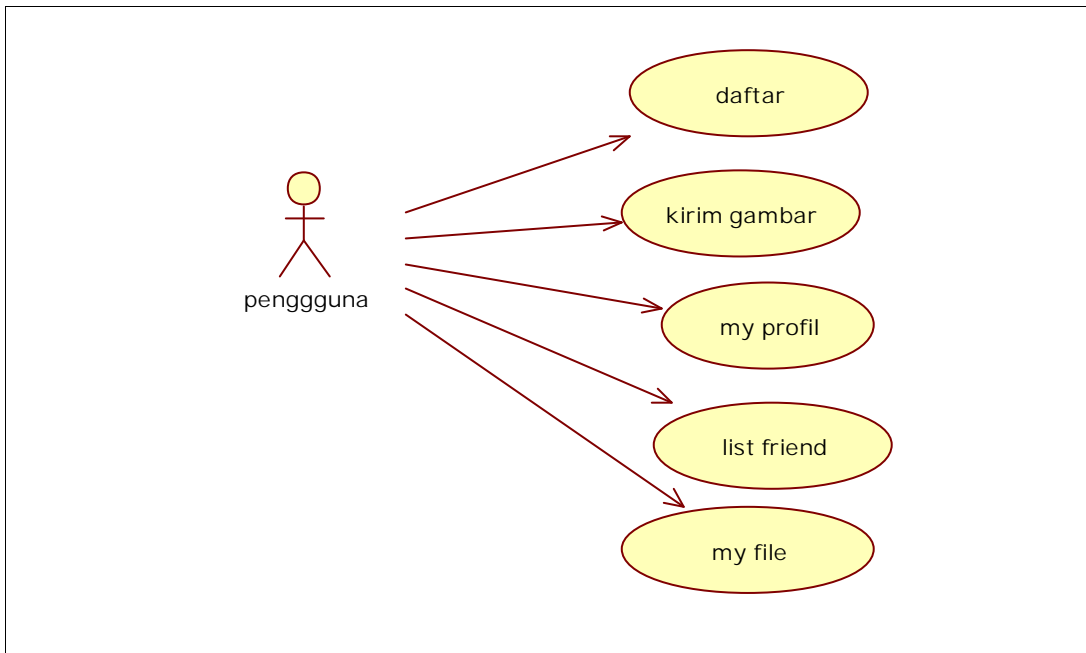
1. *Daftar* : Fitur ini berfungsi untuk user ingin menggunakan aplikasi diharuskan mendaftarkan terlebih dahulu.
2. *Login* : *Fitur* ini berfungsi untuk user masuk kedalam aplikasi.
3. *Profil*: *Fitur* ini berfungsi untuk melihat data diri pengguna.
4. *List friend*: *Fitur* ini berfungsi untuk melihat teman.
5. *File*: *Fitur* ini berfungsi untuk menerima dan *history* kiriman gambar dari user lain.
6. Kotak masuk : *Fitur* ini berfungsi untuk menerima gambar dari teman.
7. Kotak keluar : *Fitur* ini berfungsi untuk *history* pengiriman gambar kepada teman.

4.3. Analisa Fungsional

Analisa fungsional sistem merupakan gambaran dari model aplikasi menggunakan UML, yang terdiri *use case diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

4.3.1. Model Use Case

Hal-hal yang dapat dilakukan oleh pengguna terhadap *sistem* dapat dilihat pada diagram *use case* pada Gambar 4.2 berikut



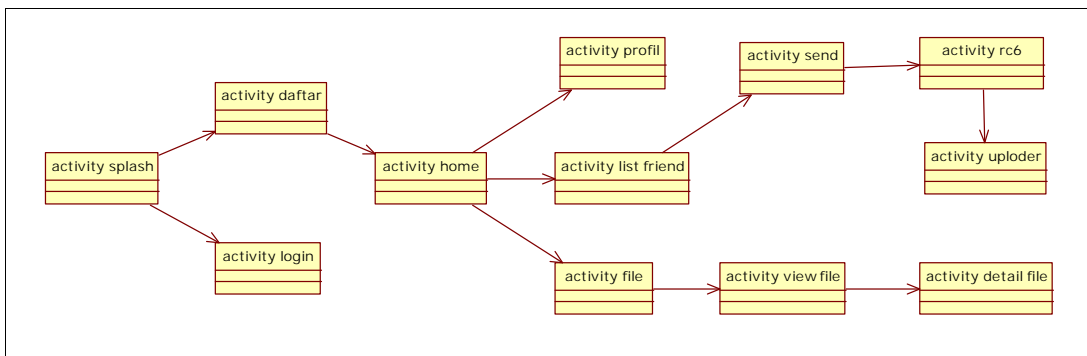
Gambar 4.2. Use Case Diagram

Tabel 4.1. Spesifikasi Use Case Diagram Sistem

Actor	Dekripsi
Pengguna	My file <ol style="list-style-type: none"> 1. Pengguna dapat melihat hasil kiriman dari <i>user</i> lain. 2. Pengguna dapat melihat kiriman pengguna ke pada <i>user</i> lain
	Kirim gambar <ol style="list-style-type: none"> 1. Pengguna dapat mengirim gambar kepada <i>user</i> yang memiliki aplikasi ini.
	Daftar <ol style="list-style-type: none"> 1. Pengguna yang belum mempunyai <i>ID</i> dalam aplikasi harus mendaftar terlebih dahulu sebelum memakai aplikasi ini.
	<i>myprofil</i> <ol style="list-style-type: none"> 1. Pengguna dapat melihat dan <i>edit profil</i>.
	<i>List friend</i> <ol style="list-style-type: none"> 1. Pengguna dapat melihat teman yang memiliki aplikasi tersebut.

4.3.2. Class Diagram

Class adalah sebuah spesifikasi yang jika dinisialisasikan akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut/properti*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metoda/fungsi*). Pada Gambar 4.3 merupakan rancangan *class diagram* aplikasi ini.



Gambar 4.3 Class Diagram Enkripsi Gambar.

Berikut adalah deskripsi dan penjelasan dari kelas-kelas yang muncul pada diagram perancangan kelas, untuk lebih jelas *class*, *atribut*, dan operasi pada lampiran A.

Tabel 4.2 Deskripsi Perancangan Class Diagram

Nama Kelas	Nama File	Keterangan
<i>Activity Splash</i>	<i>Activity splash</i>	Kelas ini merupakan tampilan awal dari aplikasi.
<i>Activity Login</i>	<i>Activity Login</i>	Kelas ini merupakan kelas untuk masuk kedalam aplikasi.
<i>Activity Daftar</i>	<i>Activity Daftar</i>	Kelas ini merupakan tampilan mengisi data sebelum menggunakan aplikasi.
<i>Activity Profil</i>	<i>Activity Profil</i>	Kelas ini merupakan tampilan data pengguna.

Tabel 4.2 Deskripsi Perancangan *Class Diagram* lanjutan

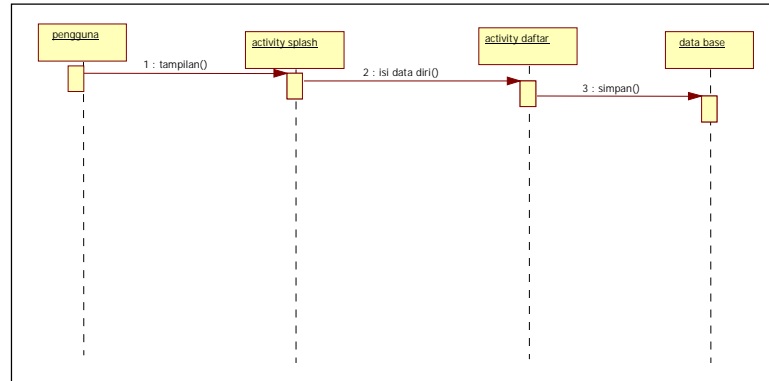
<i>Nama Kelas</i>	<i>Nama File</i>	Keterangan
<i>Activity Home</i>	<i>Activity Home</i>	Kelas ini menu utama dalam tampilan aplikasi.
<i>Activity List friend</i>	<i>Activity List friend</i>	Kelas ini menampilkan pengguna lain yang menggunakan aplikasi ini.
<i>Activity File</i>	<i>Activity File</i>	Kelas ini menampilkan gambar yang masuk dan keluar dari aplikasi.
<i>Activity Send</i>	<i>Activity Send</i>	Kelas ini merupakan mengirim ke pengguna lain.
<i>Activity RC6</i>	<i>Activity Rcc6</i>	Kelas ini merupakan kelas proses enkripsi dan dekripsi.
<i>Activity Uploder</i>	<i>Activity Uploder</i>	Kelas ini merupakan kelas proses kirim gambar pada data base.
<i>Activity Detail file</i>	<i>Activity Detail file</i>	Kelas ini merupakan kelas melihat gambar lebih rinci.
<i>Activity View file</i>	<i>Activity View file</i>	Kelas ini menampilkan gambar.

4.3.3. *Sequence Diagram*

Sequence Diagram adalah representasi dari interaksi-interaksi objek yang berjalan pada sistem. Dengan menggunakan *sequence diagram* kita dapat melihat bagaimana objek-objek bekerja. *Sequence diagram* dapat menampilkan bagaimana sistem merespon setiap kejadian atau permintaan dari *user*, dapat mempertahankan integritas internal, bagaimana data dipindah ke *user interface* dan bagaimana objek-objek diciptakan dan dimanipulasi.

Setiap sistem memiliki proses dan setiap proses memiliki dua kriteria, yaitu proses sederhana dan kompleks. Dengan demikian tidak seluruh proses pada sistem akan ditampilkan pada *sequence diagram*, melainkan hanya garis besarnya saja.

A. *Sequence Diagram* Daftar

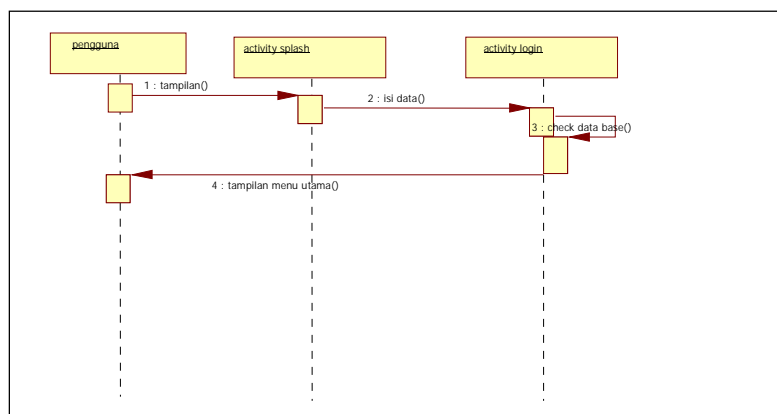


Gambar 4.4. *Sequence Diagram* Daftar

Dari Gambar 4.4. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian pengguna baru bisa mendaftar dalam aplikasi.

B. *Sequence Diagram* Login

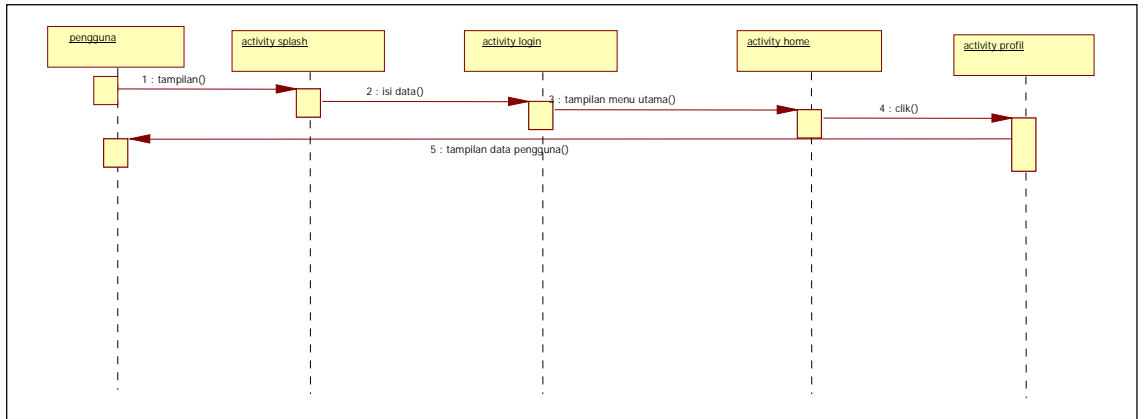
Pada Gambar 4.5. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian memilih class *activity login* terlebih dahulu.



Gambar 4.5. *Sequence Diagram* Login

C. Sequence Diagram Profil

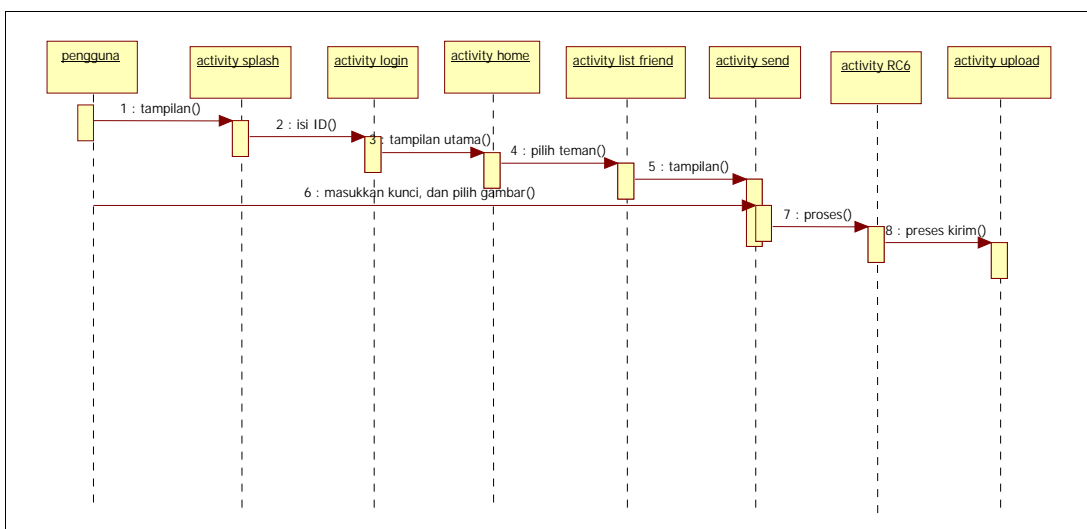
Dari Gambar 4.6. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian *activity login*, setelah berhasil *login* akan masuk pada menu utama pilih *profil*.



Gambar 4.6. Sequence Diagram Profil

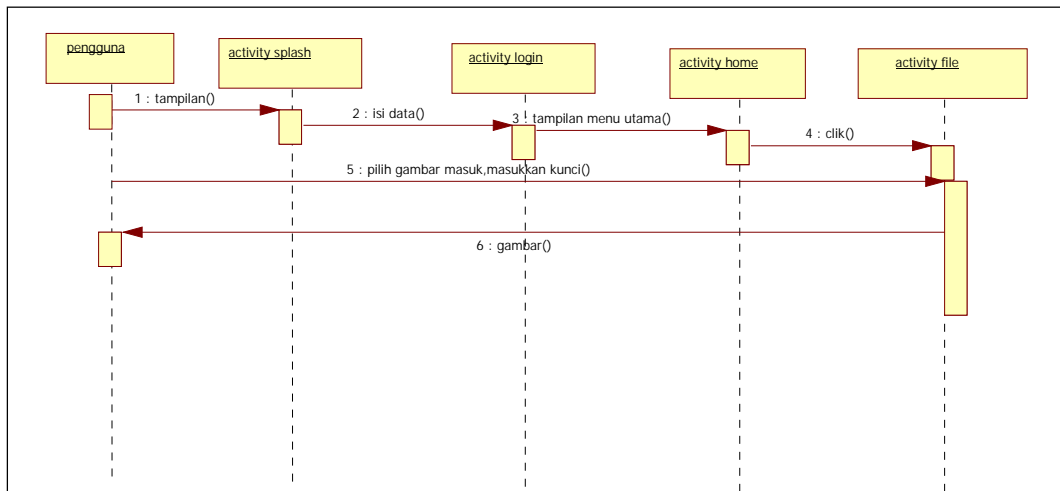
D. Sequence Diagram Kirim Gambar

Dari Gambar 4.7. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian *activity login*, setelah berhasil *login* akan masuk pada menu utama, pilih *list* teman, pengguna pilih gambar, masukkan kunci kemudian kirim, proses selanjtnya memanggil *class send,RC6* dan *uploader*. Proses tersebut berfungsi dalam pengiriman dan enkripsi.



Gambar 4.7. Sequence Diagram Kirim Gambar

E. Sequence Diagram File

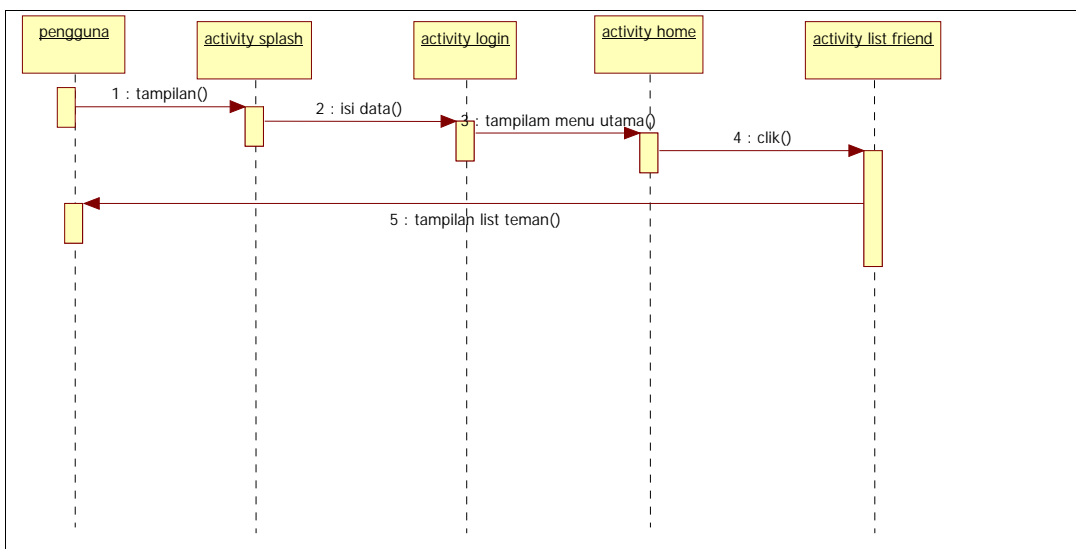


Gambar 4.8. Sequence Diagram file

Dari Gambar 4.8. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian *activity login*, setelah berhasil *login* akan masuk pada menu utama pilih *file*.

F. Sequence Diagram List Friend

Dari Gambar 4.9. Saat pengguna menggunakan aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian *activity login*, setelah berhasil *login* akan masuk pada menu utama pilih list friend.



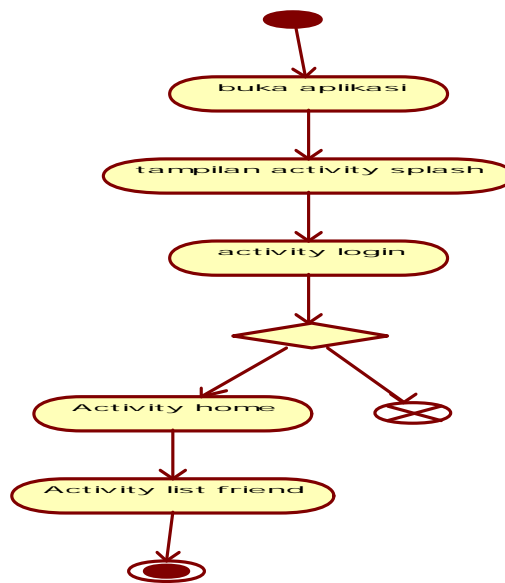
Gambar 4.9 Sequence Diagram List Friend

4.3.4. Activity Diagram

Untuk memudahkan dalam perancangan *activity diagram* maka *activity diagram* dalam aplikasi ini akan dipecah menjadi beberapa bagian.

A. *Activity Diagram* list friend

Activity diagram profil adalah proses melihat data dari pengguna. Dapat dilihat pada Gambar 4.10. saat aplikasi ini berjalan *class* pertama dipanggil adalah *activity splash* kemudian *activity login* kemudian berhasil *login* maka muncul tampilan utama dari aplikasi *activity home* kemudian *activity list friend*.



Gambar 4.10. *activity diagram* list friend

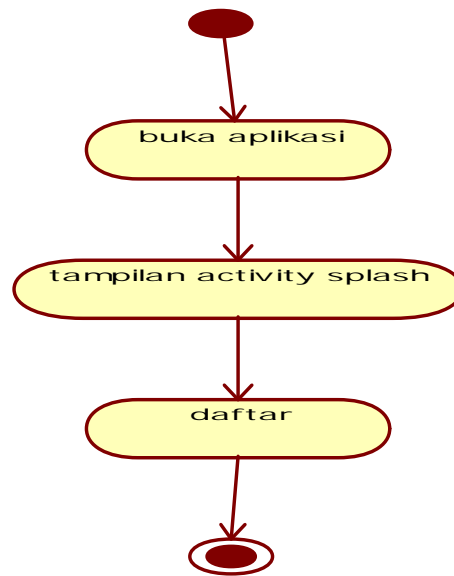
B. *Activity Diagram* Daftar

pada Gambar 4.11. saat pengguna menggunakan aplikasi ini, saat aplikasi dibuka *class* yang pertama muncul adalah *activity splash* dan pengguna harus membuat *id* terlebih dahulu sebelum menggunakan aplikasi ini lebih lanjut,

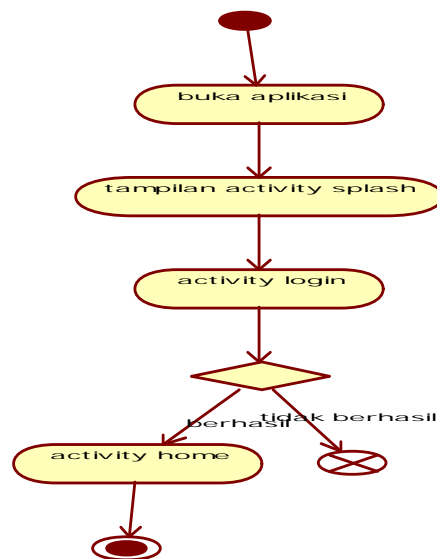
C. *Activity Diagram* Login

Activity diagram login adalah proses saat pengguna masuk kedalam aplikasi terlihat pada Gambar 4.12. Dapat dilihat saat membuka aplikasi ini *class* yang pertama dipanggil adalah *activity splash* kemudian *login*, saat *login*

diperlukan pengisian data setelah itu *id* akan di *check* kedalam *database* jika *id* itu ada maka pengguna dapat masuk kedalam aplikasi tersebut dan ketika *id* tidak ditemukan maka proses akan terhenti.



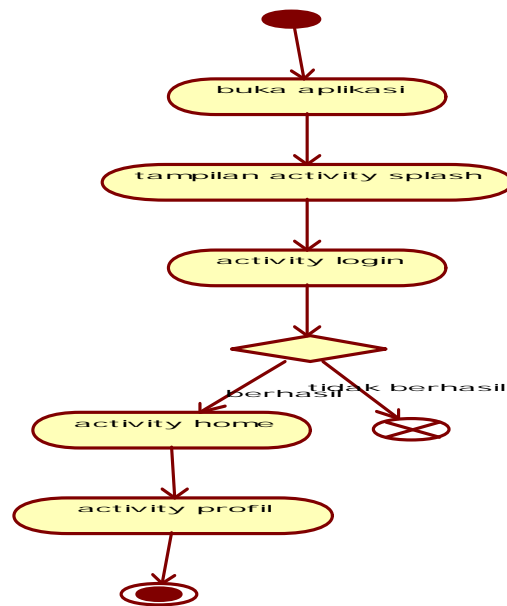
Gambar 4.11 Activity Diagram Daftar



Gambar 4.12. Activity Diagram Login.

D. Activity Diagram Profil

Activity diagram profil adalah proses melihat data dari pengguna. Dapat dilihat pada Gambar 4.13. saat aplikasi ini berjalan *class* pertama dipanggil adalah *activity splash* kemudian *activity login* kemudian berhasil *login* maka muncul tampilan utama dari aplikasi *activity home* kemudian *activity profil*,



Gambar 4.13. Activity Diagram Profil

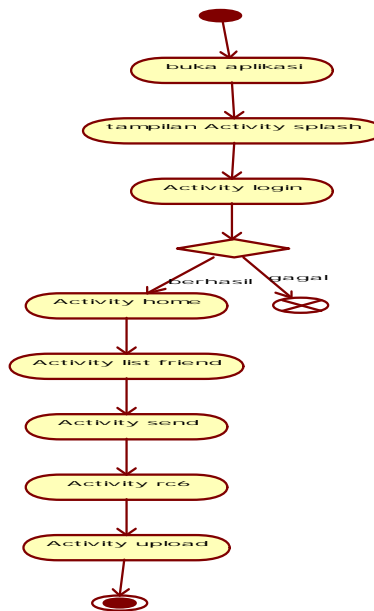
E. Activity Diagram Kirim Gambar

Activity diagram kirim gambar adalah proses kirim gambar kepada pengguna lain. Dapat dilihat pada Gambar 4.14. saat aplikasi ini berjalan *class* pertama dipanggil adalah *activity splash* kemudian *activity login* kemudian berhasil *login* maka muncul tampilan utama dari aplikasi *activity home*, pilih *list friend* dan pilih siapa teman yang kan dikirim gambar setelah itu masukkan kunci dan pilih gambar kemudian kirim. Proses kemudian gambar akan diproses oleh *activity send* kemudian *activity RC6* kemudian *activity uploder*.

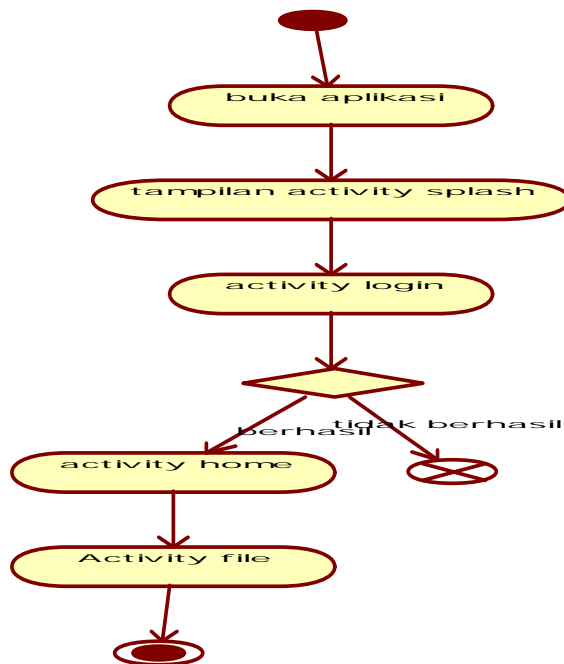
F. Activity Diagram File

Activity diagram file adalah proses melihat gambar masuk dan keluar. Proses tersebut dapat dilihat pada Gambar 4.15. saat aplikasi ini berjalan *class* pertama dipanggil adalah *activity splash* kemudian *activity login* kemudian

berhasil *login* maka muncul tampilan utama dari aplikasi *activity home*, pilih *file* kemudian akan muncul menu gambar masuk dan keluar.

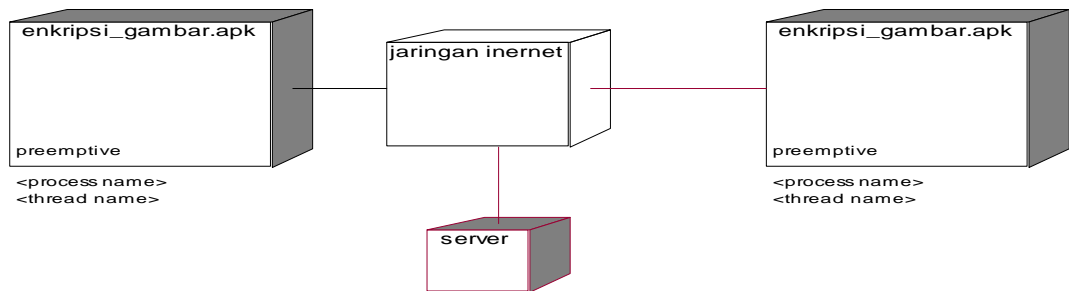


Gambar 4.14. Activity Diagram Kirim Gambar



Gambar 4.15. Activity Diagram File

4.3.5. Deploy Diagram




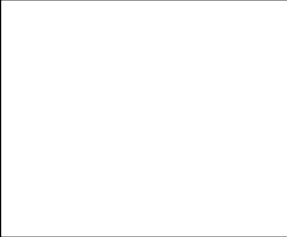

Gambar 4.16. Deploy Diagram Enkripsi Gambar

Deploy diagram menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa).

Aplikasi diterapkan pada pengirim dan penerima, gambar dikirim melalui jaringan internet sehingga *deploy diagram* yang dihasilkan dapat dilihat pada Gambar 4.16 diatas.

4.4. Analisa Perhitungan Manual Algoritma RC6

Kunci: IMAM WIBISONO123

Plantext:	Enkripsi	Dekripsi
		

Tabel 4.3 Proses Enkripsi dan dekripsi

Dimisalkan nilai empat register dari gambar diatas adalah
Register A : 01100111000101100101010101100100 (1,729,516,900)

Register B : 10010001001101010101001101100101 (1,218,095,538)

Register C : 1000111100110001000101100010110 (2,402,359,062)

Register D : 01100011011001010110110101100101 (1,667,591,525)

Untuk mempermudah perhitungan maka biner diubah dalam bentuk decimal. (Perhitungan manual pembangkit sub kunci dapat dilihat pada Lampiran B) :

1. Whitening Awal

Whitening awal, dengan menjumlahkan B dengan sub kunci S(0), dan D dengan sub kunci S(1). Penjumlahan dilakukan dalam modulo 2^{32} .

$$B = B + S(0) \text{ mod } 2^{32}$$

$$D = D + S(1) \text{ mod } 2^{32}$$

$$\begin{aligned} B &= 1,218,095,538 + 51,201,415 \text{ mod } 4,294,967,296 \\ &= 1,269,296,953 \text{ mod } 4,294,967,296 \\ &= 1,269,296,953 \end{aligned}$$

$$\begin{aligned} D &= 1,667,591,525 + 35,795,241 \text{ mod } 4,294,967,296 \\ &= 1,703,386,766 \text{ mod } 4,294,967,296 \\ &= 1,703,386,766 \end{aligned}$$

2. Iterasi

Iterasi dilakukan sebanyak 20 kali. Setiap kali iterasi mengikuti aturan sebagai berikut :

$$t \quad \text{ROTL} ((X[1]*(2*X[1]+1)), 5)$$

$$u \quad \text{ROTL} ((X[3]*(2*X[3]+1)), 5)$$

$$X[0] = (\text{ROTL}((X[0] \text{ XOR } t), u)) + S[2*i]$$

$$X[2] = (\text{ROTL}((X[2] \text{ XOR } u), t)) + S[2*i + 1]$$

$$\text{Temp} = X[0]$$

$$X[0] = X[1]$$

$$X[1] = X[2]$$

$$X[2] = X[3]$$

$$X[3] = \text{Temp}$$

Nilai t dan u didapat dari blok B dan D diproses dengan fungsi $f(x) = x(2x+1)$, kemudian dilanjutkan dengan menggeser nilai t dan u ke kiri sejauh 5 bit.

$$\begin{aligned} t &= (B * (2 * B + 1)) \text{ mod } 2^{32}. \\ &= (1,269,296,953 * (2 * 1,269,296,953 + 1)) \text{ mod } 2^{32}. \\ &= (1,269,296,953 * (2,538,593,906 + 1)) \text{ mod } 2^{32}. \\ &= (1,269,296,953 * 2,538,593,907) \text{ mod } 2^{32}. \\ &= 3,222,229,511,059,465,371 \text{ mod } 4,294,967,296 \\ &= 259,712,155 \end{aligned}$$

$$t : \text{ (dalam biner) } 1111\ 0111\ 1010\ 1110\ 0100\ 1001\ 1011$$

$$t : \text{ (digeser 5 bit) } 111\ 1010\ 1110\ 0100\ 1001\ 10111111\ 0$$

$$t : \text{ (dalam desimal) } \quad 257,725,310$$

Nilai 5 bit terakhir dari t yaitu 1111 0, atau dalam desimal sebesar 30, akan dipergunakan sebagai nilai penggeser blok C pada proses berikutnya, sejauh 16 bit.

$$\begin{aligned} u &= (D * (2 * D + 1)) \text{ mod } 2^{32}. \\ &= (1,667,591,525 * (2 * 1,667,591,525 + 1)) \text{ mod } 2^{32}. \end{aligned}$$

$$\begin{aligned}
&= (1,667,591,525 * (3,335,183,050 + 1)) \bmod 2^{32}. \\
&= (1,667,591,525 * 3,335,183,051) \bmod 2^{32}. \\
&= 5,561,722,990,171,242,775 \bmod 4,294,967,296 \\
&= 2,689,057,047
\end{aligned}$$

u : (dalam biner) 1010 0000 0100 0111 1100 0001 0001 0111

u : (digeser 5 bit) 1000 1111 1000 0010 0010 1111 0100

u : (dalam desimal) 150,479,604

Nilai 5 bit terakhir dari u yaitu 1 0100, atau dalam desimal sebesar 20, akan dipergunakan sebagai penggeser blok A pada proses berikutnya, sejauh 18 bit. Maka didapatkan nilai-nilai sebagai berikut :

$$t = 257,725,310$$

$$u = 150,479,604$$

– penggeser t = 30

– penggeser u = 20

Langkah selanjutnya adalah memproses blok A dan C dengan nilai-nilai yang telah dihasilkan.

$$A = (\text{ROTL}((A \text{ XOR } t), u)) + S[2*i]$$

A : 1,729,516,900, dalam biner 110 0111 0001 0110 0101 0101 0110 0100

t : 257,725,310, dalam biner 1111 0101 1100 1001 0011 0111 1110 ⊕

A : (hasil xor) 110 1000 0100 1010 1100 0110 0001 1010

A : (digeser 20 bit) 110 0001 1010 1101 0000 1001 0101 1000

A : (dalam desimal) 1,638,730,072

Nilai A dijumlahkan dengan sub kunci S(2), dalam modulo 2^{32} :

$$A = 1,638,730,072 + 3,557,147 \text{ modulo } 2^{32}.$$

$$= 1,642,287,219 \text{ mod } 4,294,967,296$$

$$= 1,642,287,219$$

$$C = (\text{ROTL}((C \text{ XOR } u), t)) + S[2*i + 1]$$

C : 1,642,287,219, dalam biner 110 0001 1110 0011 0101 0000 0111 0011

u : 150,479,604, dalam biner 1000 1111 1000 0010 0010 1111 0100 ⊕

C : (hasil xor) 110 1001 0001 1011 0111 0010 1000 0111

C : (digeser 30 bit) 1110 1001 0001 1011 0111 0010 1000 011

C : (dalam desimal) 1,955,445,059

Nilai C dijumlahkan dengan sub kunci S(3), dalam modulo 2^{32} . :

$$C = 1,955,445,059 + 17,716,688 \text{ modulo } 2^{32}.$$

$$= 1,973,161,747 \text{ mod } 4,294,967,296$$

$$= 1,973,161,747$$

Maka didapat nilai masing-masing blok adalah :

A : 1,642,287,219

B : 1,269,296,953

C : 1,973,161,747

D : 1,703,386,766

Langkah berikutnya adalah mempertukarkan nilai blok dengan aturan (A, B, C, D) ← (B, C, D, A), sehingga pada iterasi pertama, didapat nilai pada masing-masing blok sebagai berikut :

A : 1,703,386,766

B : 1,642,287,219

C : 1,269,296,953

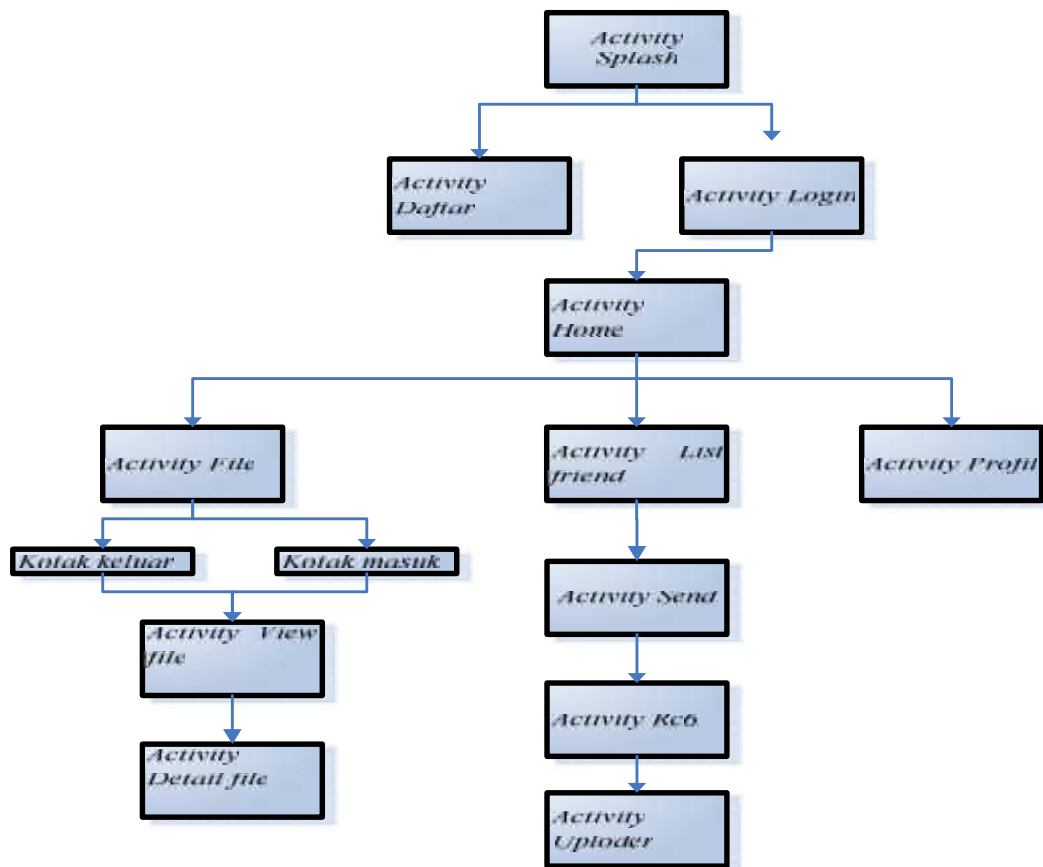
D : 1,973,161,747

Nilai masing-masing blok akan dilanjutkan pada iterasi berikutnya, dengan perhitungannya dapat dilihat pada Lampiran C

Nilai masing-masing blok akan dilanjutkan pada iterasi berikutnya sebanyak 20 kali.

4.5. Perancangan Menu

Secara garis besar desain perancangan aplikasi *mobile* dapat dilihat pada gambar 4.17. berikut.



Gambar 4.17. tampilan rancangan menu aplikasi

4.6. Perancangan *Database*

Aplikasi pengamanan gambar merupakan aplikasi mobile berbasis *client server*, dan memerlukan sebuah database menjalankan fungsinya. Perancangan database pada aplikasi dirancang dengan satu database utama yaitu database server, pada Gambar 4.18 berikut adalah model logic database server pada aplikasi.



Gambar 4.18. Database Aplikasi

Tabel 4.4. Keterangan *Atribut usser*

No	Nama Field	Tipe Data	Null	Keterangan
1	Id	Integer(5)	Not Null	ID <i>file</i>
2	username	Varchar(20)	Not Null	Nama pengguna
3	password	Varchar(20)	Null	Password
4	fullname	Varchar(100)	Null	Nama lengkap pengguna
5	status	Varchar(50)	Not Null	Status

Tabel 4.5. Keterangan *Atribut file*

No	Nama Field	Tipe Data	Null	Keterangan
1	Id	Integer(11)	Not Null	ID <i>file</i>
2	user	Varchar(50)	Null	Nama pengguna
3	dari	Varchar(50)	Not Null	Nama pengirim
4	file	text	Not Null	Gambar

Tabel 4.5. Keterangan *Atribut file* lanjutan

No	Nama Field	Tipe Data	Null	Keterangan
5	waktu	Varchar(50)	Not Null	waktu
6	kunci	text	Not Null	kunci

4.7 Perancangan *Interface*

Berikut adalah contoh tampilan *interface* aplikasi enkripsi yang berjalan pada sistem operasi Android.

A. Halaman Daftar atau Login

Halama Daftar atau Login pada aplikasi ini tampil apabila tombol aplikasi Enkripsi. tampilan utama berisi menu: *user name*, *password*, daftar, dan masuk tampilan halaman utama pada aplikasi dapat dilihat seperti Gambar 4.19 berikut :



The image shows a screenshot of a mobile application interface for a login or registration screen. It features a white background with a thin black border. At the top, there is a large empty rectangular area. Below this, there are two input fields: the first is labeled 'User name' and the second is labeled 'Password'. Below the input fields, there are two buttons: 'Daftar' (with a red underline) and 'Login'. At the bottom, there is another large empty rectangular area.

Gambar 4.19. Tampilan daftar atau login

B. Daftar

Pada menu aplikasi terdapat tombol daftar apabila tombol *daftar* tersebut diklik maka tampilan menu berisi: nama depan, nama belakang, user name, password, ulang password, dan simpan. Seperti terlihat pada Gambar 4.20 dibawah ini.



The image shows a registration form titled "Daftar". It contains six input fields stacked vertically: "Nama depan", "Nama belakang", "User name", "Password", and "Retype Password". There is a large empty space at the bottom of the form, likely for a "Simpan" (Save) button.

Gambar 4.20. Tampilan daftar

C. Menu

Pada menu aplikasi terdapat tombol masuk apabila tombol masuk tersebut diklik maka akan tampil menu masuk: daftar teman, tambah teman, gambar masuk, data diri, keluar dan nama pengguna. Seperti pada Gambar 4.21 dibawah ini



The image shows a user menu titled "ID pengguna". It contains three main menu items: "My profil", "List friend", and "My file". At the bottom, there are two buttons: "Logout" and "Exit".

Gambar 4.21. Tampilan Menu

D. Lihat Teman

Pada menu aplikasi terdapat tombol lihat teman. Menu ini adalah daftar teman yang menggunakan aplikasi tersebut. Seperti terlihat pada Gambar 4.22. dibawah ini.



Gambar.4.22. Tampilan Lihat Teman

F. File

Pada menu aplikasi terdapat tombol gambar masuk apabila tombol gambar masuk tersebut diklik maka tampilan menu gambar masuk berisi kota masuk dan kotak keluar. Seperti terlihat pada Gambar 4.23 dibawah ini



Gambar 4.23. Tampilan file

G. Data Diri

Pada menu aplikasi terdapat tombol data diri apabila tombol *Data diri* tersebut diklik maka tampilan menu data diri berisi: nama lengkap, *user name*, password, dan ID registrasi. Seperti terlihat pada Gambar 4.24 dibawah ini.



The image shows a mobile application interface for a user profile. At the top, there is a header labeled "MY profil". Below the header, there are four input fields stacked vertically, each containing a label: "Nama Lengkap", "User name", "Password", and "ID registrasi". The fields are empty, suggesting they are for data entry or display.

Gambar 4.24. Tampilan Data Diri.

H. Kirim Gambar

Pada menu aplikasi terdapat tombol pilih teman apabila tombol pilih tersebut diklik dan memilih salah satu teman maka tampilan menu data diri berisi: kirim ke, gambar, kunci, pilih gambar, dan kirim. Seperti terlihat pada Gambar 4.25. dibawah.



The image shows a mobile application interface for sending a picture. At the top, there is a header labeled "To:". Below the header, there is a large input field labeled "gambar". At the bottom of the screen, there are three buttons: "pilih gambar", "masukkan kunci", and "kirim".

Gambar 4.25. Kirim Gambar.