BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Analisis masalah bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem dimana aplikasi dibangun, meliputi perangkat keras (*hardware*), perangkat lunak (*software*) dan pengguna (*user*). Analisis ini diperlukan sebagai dasar bagi tahapan perancangan sistem. Analisis sistem meliputi identifikasi permasalahan, analisis sistem, analisis kriptografi, analisis proses enkripsi, analisis proses dekripsi.

Keamanan informasi adalah suatu keharusan yang perlu diperhatikan apalagi jika informasi itu bersifat rahasia. Salah satu media yang paling sering digunakan untuk saling bertukar informasi/pesan adalah aplikasi *chatting*. Akan tetapi apabila aplikasi ini tidak memiliki sistem keamanan untuk melindungi seluruh informasi/pesan yang terjadi di dalamnya, tentu ada kemungkinan informasi/pesan tersebut disadap oleh pihak-pihak yang tidak berhak tanpa ada kesulitan sekalipun untuk membaca isi dari pesan atau informasi tersebut. Apalagi jika saluran komunikasi yang digunakan untuk *chatting* kurang aman, tentu akan mempermudah pihak yang tidak bertanggung jawab tersebut untuk memonitor seluruh isi percakapan yang terjadi di saluran komunikasi tersebut.

Untuk mengatasi hal ini, penulis akan mencoba menerapkan salah satu Algoritma kriptografi yaitu Algoritma *Blowfish* sebagai sistem pengamannya. Hal ini diharapkan mampu untuk melindungi seluruh informasi/pesan yang terjadi di

dalamnya dengan cara mengenkripsi pesan yang sebelumnya berupa *plaintext* menjadi *ciphertext* sebelum dikirim ke penerima pesan.

III.2. Penerapan Metode

Penerapan Metode/Algoritma akan membahas tentang dimana akan diterapkan metode yang digunakan di dalam penelitian penulis, dalam hal ini penulis akan menerapkan Algoritma *Blowfish* dalam pengolahan data yang nantinya akan dienkripsi maupun didekripsi dengan menggunakan Algoritma tersebut.

Pada aplikasi yang akan dirancang penulis Algoritma ini akan diterapkan pada bagian yang menangani *password,* kunci dan pesan dimana akan dilakukan proses enkripsi maupun dekripsi. Selanjutnya hasil dari enkripsi akan disimpan ke dalam database agar dapat diakses oleh *user*. Hal ini di harapkan mampu untuk melindungi seluruh informasi yang ada di dalam aplikasi yang akan dirancang oleh penulis terhadap serangan-serangan yang mungkin akan terjadi.

III.3. Analisis Kebutuhan Perancangan

III.3.1. Analisis Kebutuhan

Untuk mencapai penyelesaian dalam merancang aplikasi ini adapun kebutuhan pokok yang diperlukan adalah:

1. *Hardware*
    a) PC *(Personal Computer)*
    b) *Android Device*
2. *Software*
    a) *Eclipse IDE (Integrated Development Environment)*

b) *Android SDK (Software Development Kit)*

III.3.2. Spesifikasi dan Desain

Spesifikasi minimum *hardware* dan *software* yang dibutuhkan untuk membangun aplikasi ini adalah:

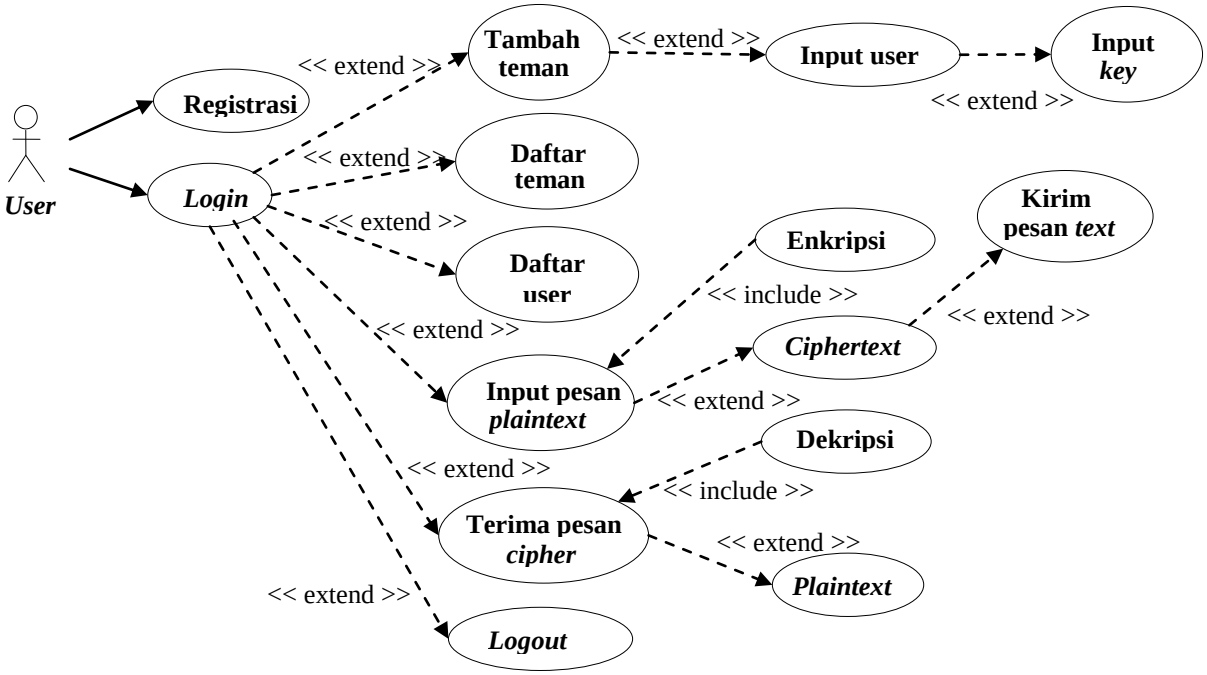
1. *Hardware*
   a) *Processor Core* 2 *Duo* 2,0 Ghz (PC), *Processor* 600 Mhz (*Android*).
   b) *Harddisk* 80GB (PC), 128MB (*Android*).
   c) *RAM* 2GB (PC), *512MB* (*Android*).
   d) *WiFi.*
2. *Software*
   a) Sistem operasi PC : *Windows* Xp Sp 3 ,*Windows* 7.
   b) Sistem operasi *Android* : *Jelly Bean* (4.3).
   c) *Eclipse IDE (Integrated Development Environment).*
   d) *Android SDK (Software Development Kit).*

III.4. Desain Sistem

Sebagai solusi dari permasalahan yang telah teridentifikasi dan untuk membuat sebuah sistem yang dapat berjalan dengan baik serta sesuai harapan yang diinginkan maka tentunya terlebih dahulu haruslah membuat tahapan perencanaan sistem berupa *use case diagram, class diagram, sequence diagram, activity diagram* dan *flowchart.*

III.4.1. *Use Case Diagram*

Use case diagram dari aplikasi *chatting* yang akan dirancang oleh penulis adalah seperti gambar III.1 sebagai berikut :



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

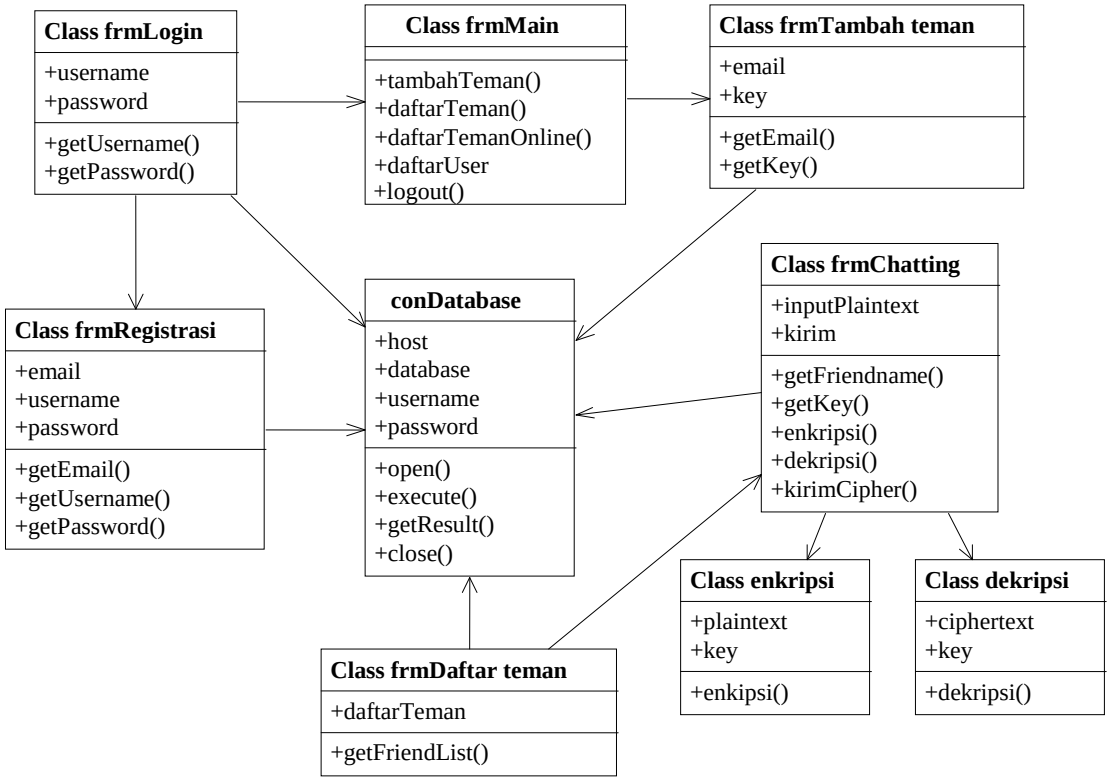
Dalam hal ini, user berfungsi sebagai *actor yang* harus melakukan proses registrasi terlebih dahulu sebelum melakukan *login.* Apabila registrasi berhasil maka *user* dapat melakukan *login*. Tampilan *login* akan muncul ketika *user* membuka aplikasi tersebut. Pada saat *login*, *user* akan diminta untuk memasukkan *username* dan *password* yang sama saat registrasi. Proses validasi akan dilakukan ketika *user* akan *login*, hal ini dimaksudkan agar tidak ada nama *user* yang bernilai "*null*" atau kosong. Jika tidak ada masalah dengan proses validasi, maka dari tampilan *login*, *user* akan diarahkan menuju tampilan utama aplikasi *chatting*.

Selanjutnya *user* dapat menambahkan teman baru, melihat daftar pertemanan, menerima pesan, mengirim pesan maupun *logout*. Kunci yang dipergunakan untuk proses enkripsi dan dekripsi dapat ditentukan pada saat menambahkan teman baru. Pada saat mengirim pesan, pesan *plaintext* akan dienkripsi terlebih dahulu dengan menggunakan Algoritma *Blowfish* lalu

selanjutnya dikirim. Ketika menerima pesan, pesan tersebut harus didekripsi terlebih dahulu dengan menggunakan Algoritma yang sama agar dapat dibaca.

III.4.2 *Class Diagram*

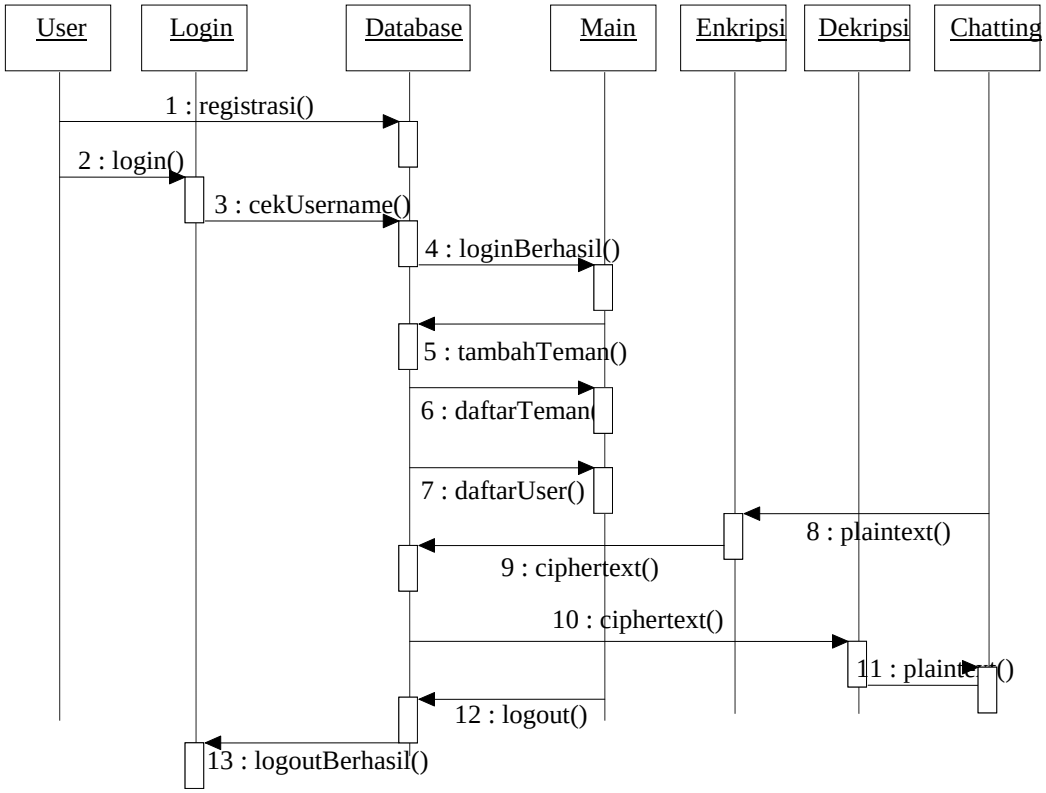
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. Adapun c*lass diagram* dari aplikasi *chatting* yang akan dirancang oleh penulis adalah sebagai berikut:



Gambar III.2. *Class Diagram* Aplikasi *Chatting*

III.4.3. *Sequence Diagram*

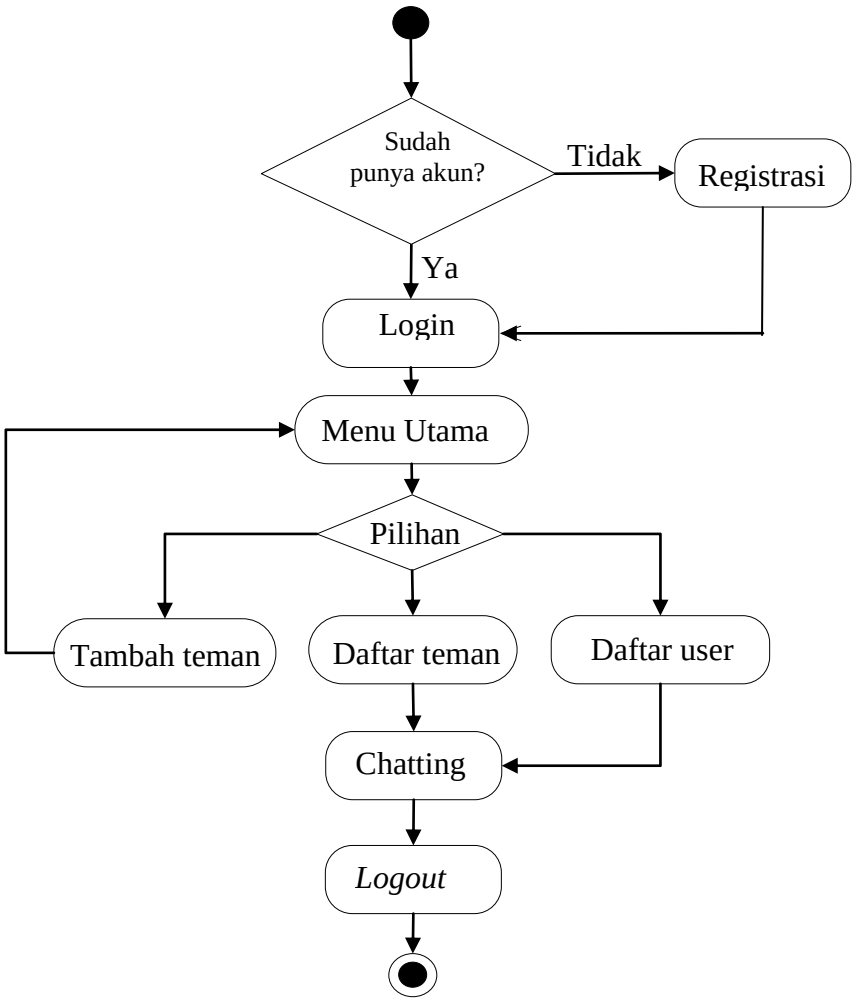
Sequence Diagram dari aplikasi *chatting* yang akan dirancang oleh penulis yang merupakan gambaran proses dari suatu sistem adalah sebagai berikut :



Gambar III.3. *Sequence Diagram* Aplikasi *Chatting*

III.4.4. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* dari aplikasi *chatting* yang akan dirancang oleh penulis adalah seperti gambar III.4 sebagai berikut :



Gambar III.4. *Activity Diagram* Aplikasi *Chatting*

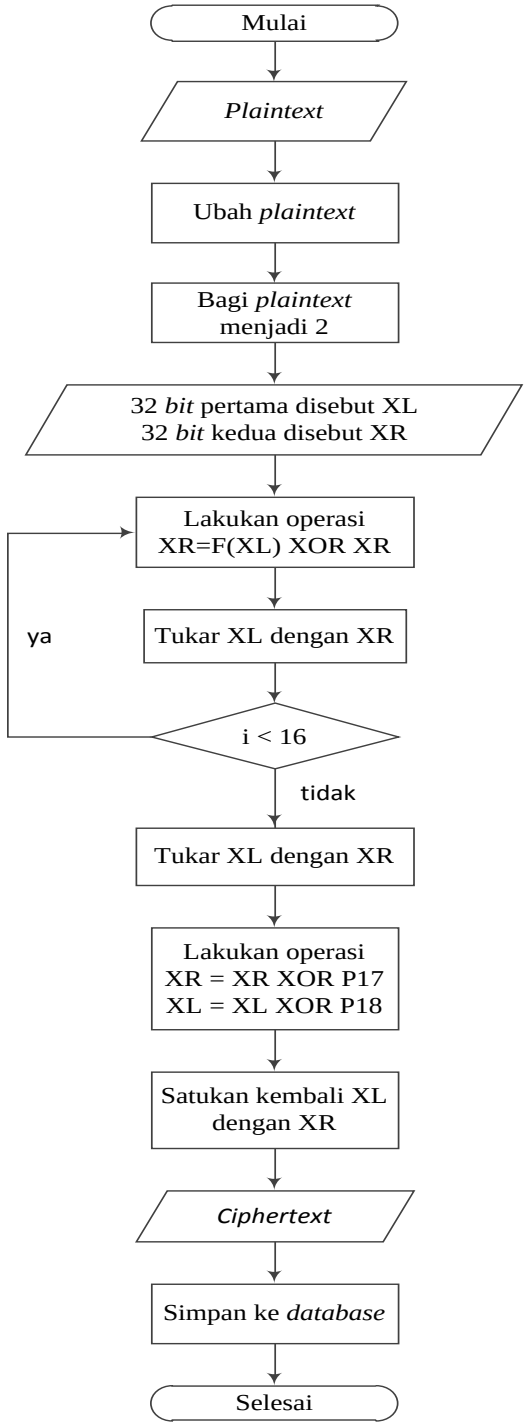
Pada gambar activity diagram di atas memiliki struktur sebagai berikut:

1. *User* dapat memilih antara *login* atau registrasi. Jika *user* sudah memiliki akun, *user* dapat langsung *login* jika tidak *user* harus registrasi terlebih dahulu.
2. Setelah *user* dapat *login*, *user* dapat melakuan aktivitas seperti tambah teman, lihat daftar user, lihat daftar teman, *chatting* dan *logout* jika telah selesai.

III.4.5. *Flowchart*

Terdapat beberapa *flowchart* dari aplikasi *chatting* yang akan dirancang oleh penulis seperti yang terdapat pada gambar III.5, III.6, III.7 sebagai berikut :

1. *Flowchart* Enkripsi *Blowfish*.



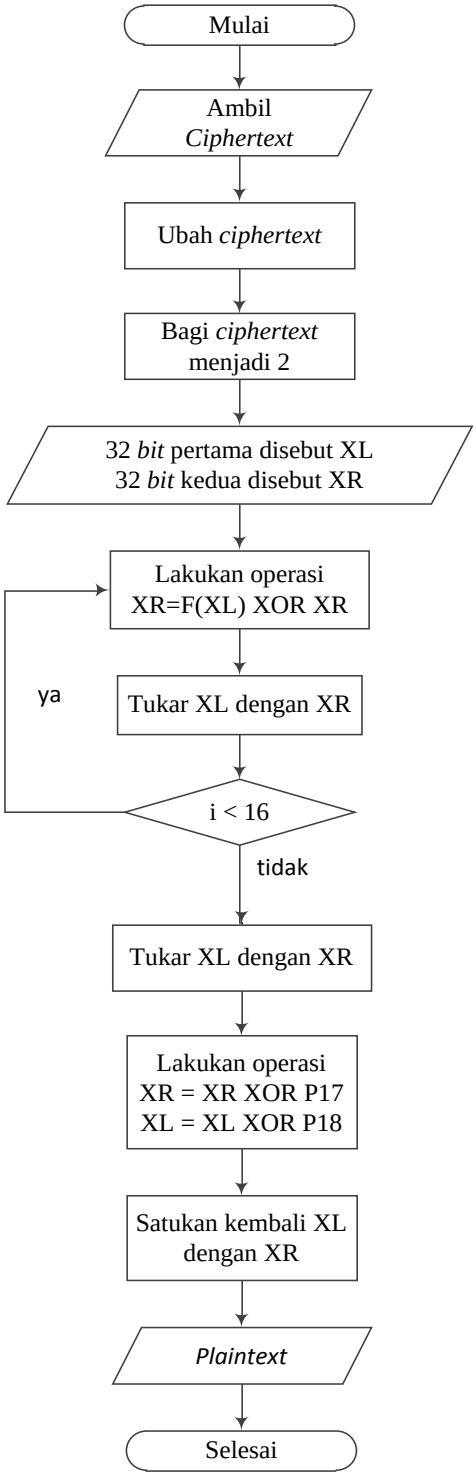
Gambar III.5. *Flowchart* Enkripsi *Blowfish*

Dari gambar III.5. *Flowchart* Enkripsi *Blowfish* dapat diperoleh keterangan sebagai berikut :

a) Memulai proses enkripsi.
b) Input *plaintext.*
c) Ubah *plaintext* ke *bit*.
d) *Plaintext* dibagi menjadi 2, XL (x *left* =32 *bit*) dan XR (x *right* =32 *bit*)
e) Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) *xor* XR
f) Hasil dari operasi di atas ditukar XL menjadi XR dan XR menjadi XL.
g) Lakukan sebanyak 16 kali.
h) Tukar kembali XL dengan XR.
i) Pada proses ke-17 lakukan operasi untuk XR = XR *xor* P17 dan XL = XL *xor* P18.
j) Satukan kembali XL dan XR sehingga menghasilkan *ciphertext* sebesar 64 *bit*.
k) Selanjutnya *ciphertext* dapat disimpan ke dalam *database*.

2. *Flowchart* Dekripsi *Blowfish*.

Pada dasarnya proses dekripsi dengan menggunakan Algoritma *Blowfish* sama halnya dengan proses enkripsi pada Algoritma *Blowfish* hanya saja P-Array digunakan secara terbalik atau di inverskan. Untuk lebih jelasnya dapat dilihat pada gambar III.6.



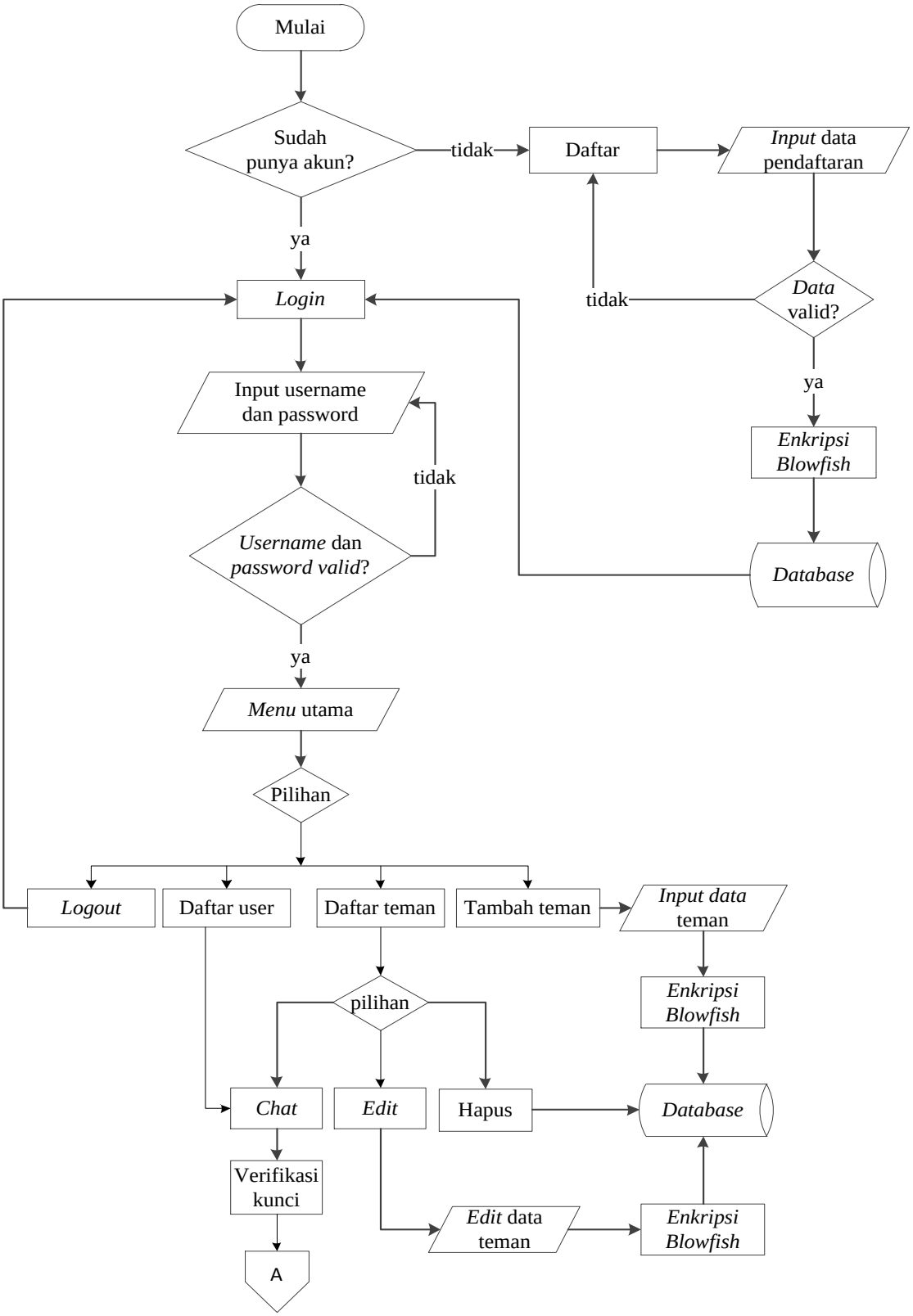
Gambar III.6. *Flowchart* Dekripsi *Blowfish*

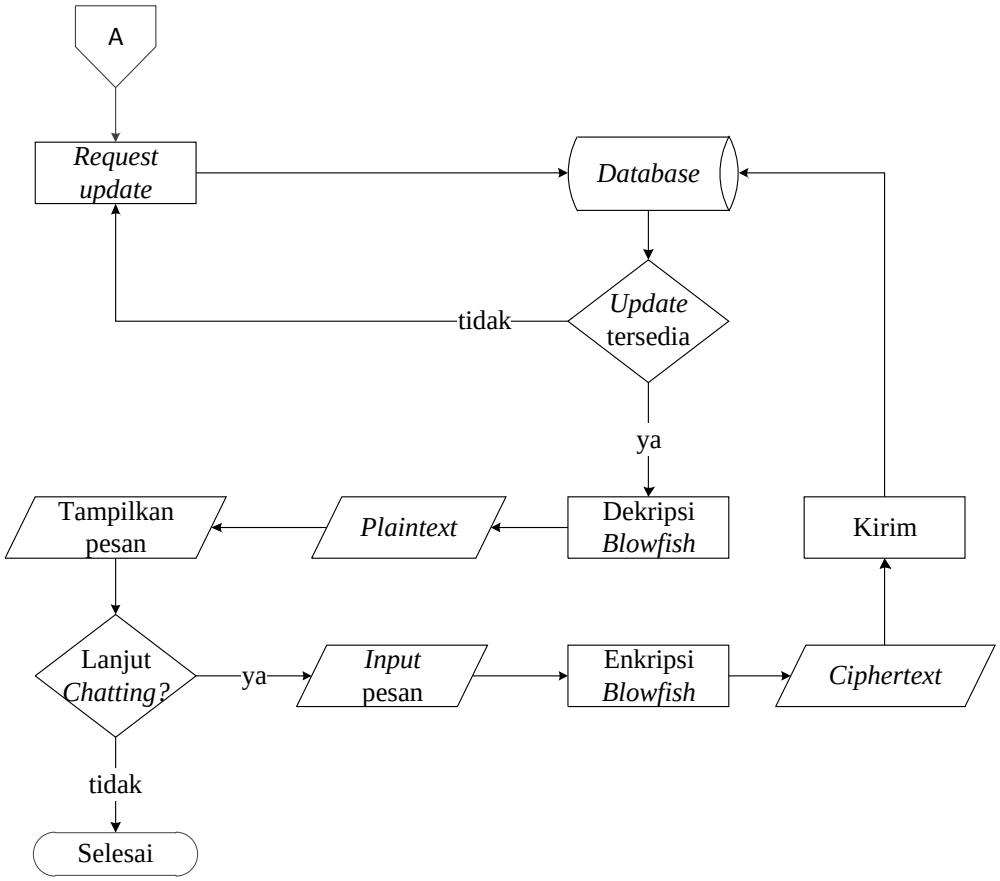
Dari gambar III.6. *Flowchart* Dekripsi *Blowfish* dapat diperoleh keterangan sebagai berikut :

a) Memulai proses dekripsi.
b) Ambil *ciphertext* dari dalam *database.*
c) Ubah *ciphertext* ke *bit*.
d) *Ciphertext* dibagi menjadi 2, XL (x *left* =32 *bit*) dan XR (x *right* =32 *bit*).
e) Selanjutnya lakukan operasi XL = XL *xor* Pi dan XR = F(XL) *xor* XR.
f) Hasil dari operasi di atas ditukar XL menjadi XR dan XR menjadi XL.
g) Lakukan sebanyak 16 kali.
h) Tukar kembali XL dengan XR.
i) Pada proses ke-17 lakukan operasi untuk XR = XR *xor* P17 dan XL = XL *xor* P18
j) Satukan kembali XL dan XR sehingga menghasilkan *plaintext* sebesar 64 *bit*.
k) Selanjutnya plaintext dapat ditampilkan.

3. *Flowchart* Penerapan Metode Pada Aplikasi

Setelah memahami proses enkripsi dan dekripsi dengan menggunakan Algoritma *Blowfish,* selanjutnya penulis akan mencoba menerapkannya di dalam aplikasi *chatting* untuk lebih jelasnya lihat gambar III.7.


Mulai
Sudah punya akun?
tidak
Daftar
Input data pendaftaran
ya
Login
tidak
Data valid?
ya
Input username dan password
tidak
Enkripsi Blowfish
Username dan password valid?
Database
ya
Menu utama
Pilihan
Logout
Daftar user
Daftar teman
Tambah teman
Input data teman
pilihan
Enkripsi Blowfish
Chat
Edit
Hapus
Database
Verifikasi kunci
Edit data teman
Enkripsi Blowfish
A



Gambar III.7. *Flowchart* Penerapan Metode Pada Aplikasi *Chatting*

Dari gambar di atas dapat dilihat bahwa, proses registrasi harus terlebih dahulu dilakukan sebelum melakukan *login.* Apabila registrasi berhasil maka *user* dapat melakukan *login*. Pada saat *login, user* akan diminta untuk memasukkan *username* dan *password* yang sama saat registrasi. Proses validasi akan dilakukan ketika *user* akan *login,* jika benar maka akan diarahkan ke menu utama dan jika tidak maka *user* harus melakukan proses *login* kembali.

Selanjutnya *user* dapat melakukan aktivitas seperti menambahkan teman baru, melihat daftar pertemanan dan *logout*. Kunci yang dipergunakan untuk proses enkripsi dan dekripsi dapat ditentukan pada saat menambahkan teman baru. Pada proses daftar teman, *user* dapat melihat daftar pertemanan serta meng-*edit*

maupun menghapusnya. Ketika *user* mengirim pesan, pesan *plaintext* akan dienkripsi terlebih dahulu dengan menggunakan Algoritma *Blowfish* lalu selanjutnya dikirim dan disimpan ke dalam *database*. Ketika menerima pesan, pesan tersebut harus didekripsi terlebih dahulu dengan menggunakan Algoritma yang sama agar dapat dibaca.

III.5. Desain *Database*

Dalam perancangan aplikasi ini, penulis menggunakan *database* sebagai tempat penyimpanan seluruh data pengguna. Adapun struktur *database* yang didesain penulis dalam perancangan aplikasi ini adalah sebagai berikut.

Tabel III.1. Struktur *Database Chatting*

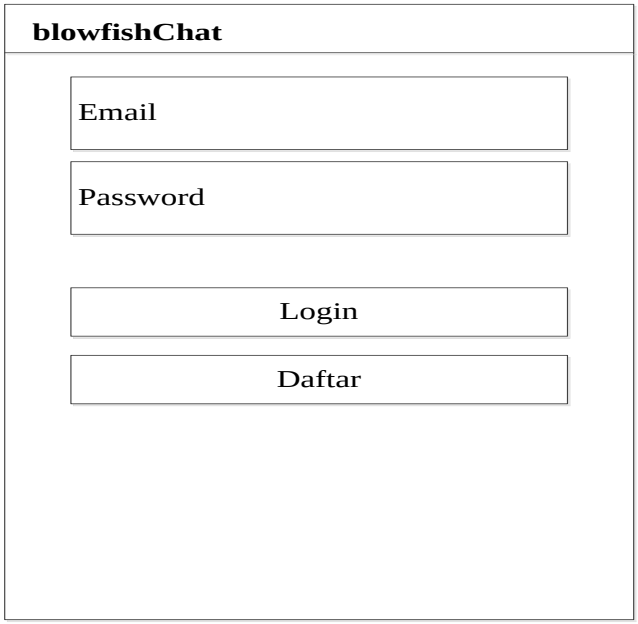
Database Chatting			
Nama Tabel	**No**	**Tipe**	**Atribut**
User_login	1 2 3 4 5	*Varchar* *Varchar* *Varchar* *Varchar* *Int*	- username (20) - *email (50) - *password* (50) - kunci(50) - status (1)
Teman	1 2 3 4	*Varchar* *Varchar* *Varchar* *Varchar*	- e_username (50) - e_teman (50) - kunci (50) - status (1)
Pesan	1 2 3 4 5 6	*Int* *Varchar* *Varchar* *Text* *Varchar* *Timestamp*	- *id (10) - dari (50) - kepada (50) - pesan - status (2) - *time*

III.6. Desain *User Interface*

Setelah perancangan diagram telah dibuat maka selanjutnya adalah perancangan *user interface* sebagai berikut :

1. *User Interface Login*

Pada gambar III.8 adalah tampilan *user interface login* dimana terdapat dua buah *textbox*. *Textbox* yang pertama digunakan untuk inputan *email* dan *textbox* yang kedua digunakan untuk inputan *password*. Selanjutnya terdapat dua buah *button* yaitu *button login* dan *button* daftar. *Button login* digunakan untuk masuk ke halaman utama *user* apabila proses *login* berhasil, dan *button* daftar digunakan untuk menampilkan *form* daftar.



Gambar III.8. Desain *User Interface Login*

2. *User Interface Daftar*

Pada gambar III.9 adalah tampilan *user interface* daftar, dimana terdapat empat buah *textbox* yaitu *textbox email*, *textbox password*, *textbox username* dan *textbox* kunci. Keempat *textbox* ini digunakan sebagai inputan data *user*. Selain itu

juga terdapat sebuah *button* daftar yang berfungsi untuk memasukkan data pengguna ke dalam *database*.

Gambar III.9. Desain *User Interface* Daftar

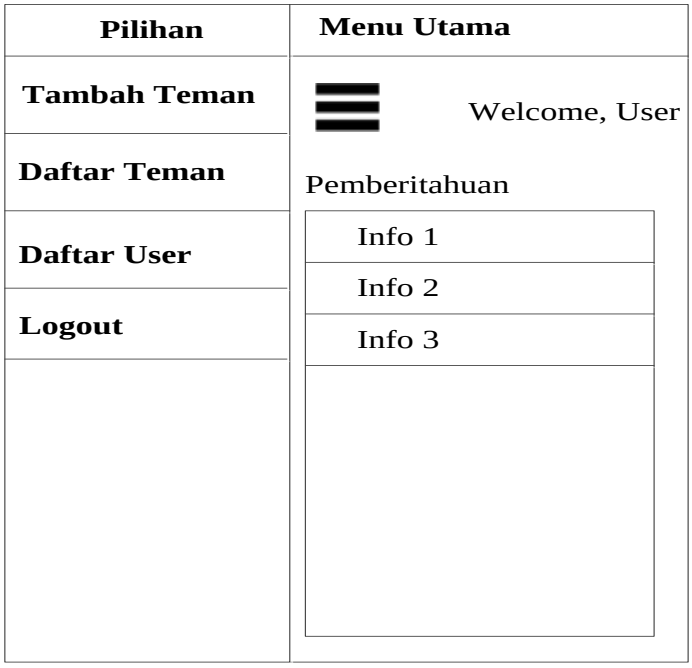
3. *User Interface* Menu Utama

Pada gambar III.10 adalah tampilan *user interface* menu utama. Tampilan ini akan muncul jika *user* telah berhasil *login*. Pada tampilan ini terdapat dua buah *image button*. Yang pertama, digunakan untuk menampilkan *user interface* pilihan dan yang kedua digunakan untuk menampilkan *user interface* teman *online*. Selain itu terdapat pula *listview* yang digunakan untuk menampilkan informasi pemberitahuan.

Gambar III.10. Desain *User Interface Menu* Utama

4. *User Interface* Pilihan

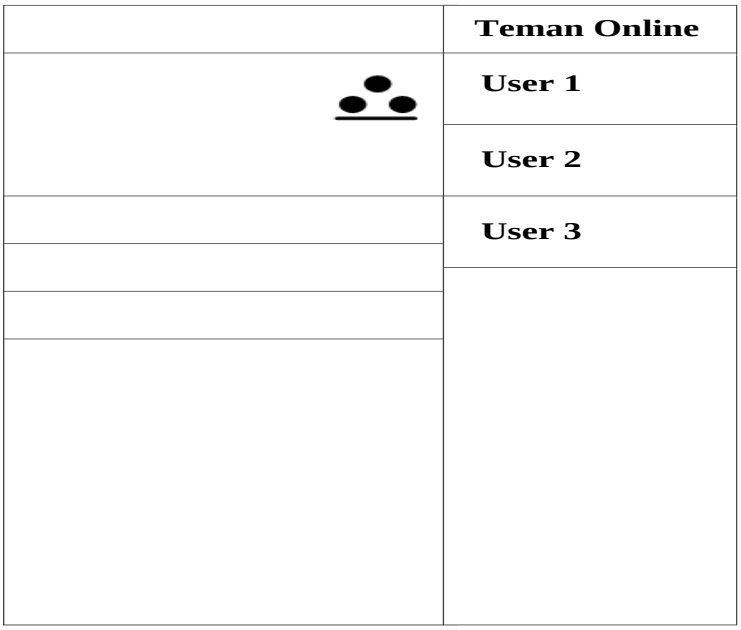
Pada gambar III.11 adalah tampilan *user interface* pilihan. Pada tampilan ini, terdapat sebuah *listview* yang digunakan untuk menampilkan *item* pilihan. Adapun isinya yaitu tambah teman, daftar teman, daftar user dan *logout*. *Item* tambah teman digunakan untuk menampilkan *user interface* tambah teman, *item* daftar teman digunakan untuk menampilkan *user interface* daftar teman, *item* daftar *user* digunakan untuk menampilkan daftar *user* dan *item logout* digunakan untuk keluar dari aplikasi dan mengakhiri sesi percakapan setelah itu *item* ini akan menampilkan *user interface login*.



Gambar III.11. Desain *User Interface* Pilihan

5. *User Interface* Teman *Online*

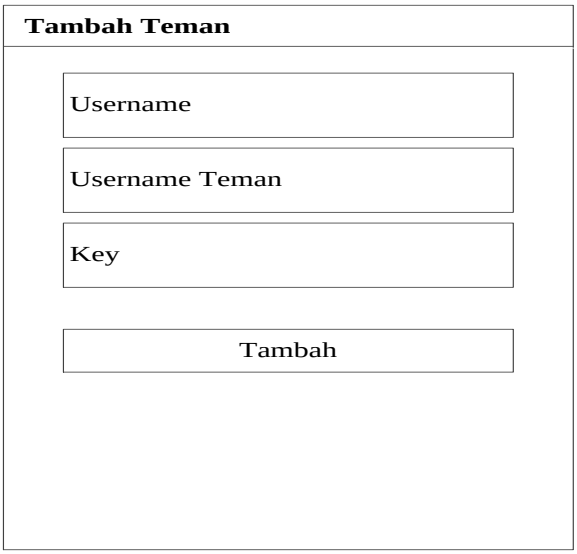
Pada gambar di bawah ini adalah tampilan *user interface* teman *online*. Digunakan untuk menampilkan daftar teman yang sedang *online*.



Gambar III.12. Desain *User Interface* Teman *Online*

6. *User Interface* Tambah Teman

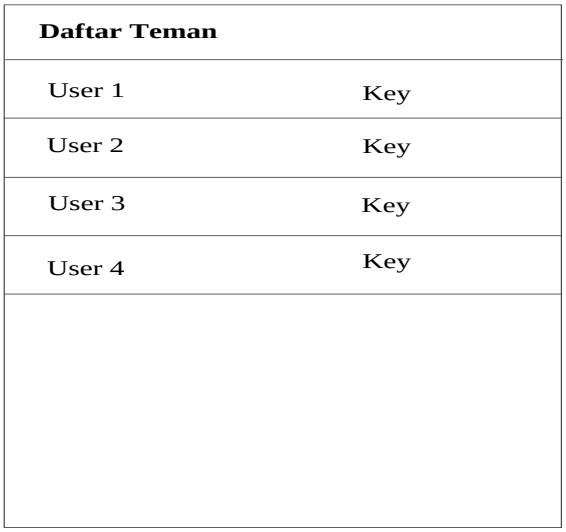
Pada gambar di bawah ini adalah tampilan *user interface* tambah teman. Tampilan ini digunakan untuk menginputkan data teman.



Gambar III.13. Desain *User Interface* Tambah Teman

7. *User Interface* Daftar Teman

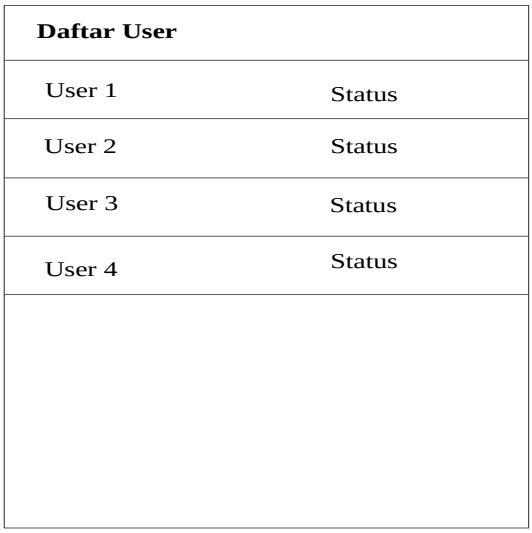
Pada gambar III.14 adalah tampilan *user interface* daftar teman. Digunakan untuk menampilkan daftar teman yang pernah ditambahkan.



Gambar III.14. Desain *User Interface* Daftar Teman

8. *User Interface* Daftar *User*.

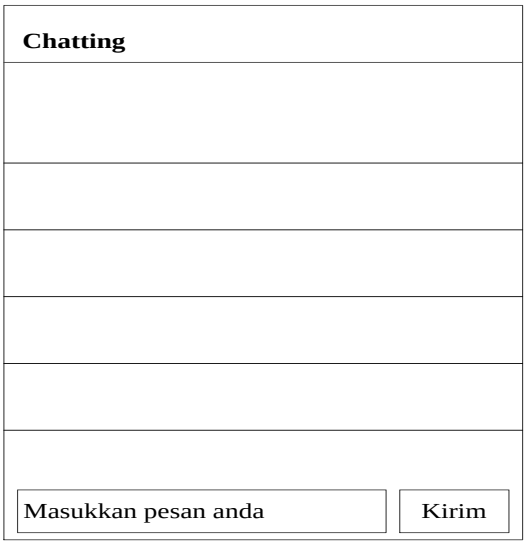
Pada gambar III.15 adalah tampilan user interface daftar user. Digunakan untuk menampilkan seluruh user yang telah berhasil registrasi.



Gambar III.15. Desain *User Interface* Daftar *User*

9. *User Interface Chatting*

Pada gambar di bawah ini adalah tampilan *user interface chatting*. Tampilan ini digunakan untuk menampilkan isi percakapan.



Gambar III.16. Desain *User Interface Chatting*