# Penerapan Algoritma Twofish dalam Perancangan Aplikasi Chat Berbasis Android

**Florensius Laylim[*1], Muhammad Qadafi Khairuzzaman[2]**
[1,2]Jurusan Teknik Informatika; STMIK Pontianak. Jl. Merdeka No.372 Pontianak, 0561-735555
e-mail: [*1]florensiusxodc@gmail.com, [2]m.qadafi.k@gmail.com

***Abstrak***

*Chatting merupakan komunikasi dua arah antara satu orang atau beberapa orang dalam teks, suara maupun video. Penelitian ini dirancang berdasarkan kelebihan dari WhatsApp yaitu tampilannya yang baik dan sederhana sehingga mudah dimengerti, tetapi pengguna WhatsApp tidak dapat melihat pengguna yang lain. Oleh sebab itu, chatting yang dirancang ini memiliki tampilan yang menyerupai WhatsApp dan dapat melihat pengguna yang lain. Aplikasi chatting ini tidak dirancang untuk menggantikan WhatsApp. Aplikasi chatting ini menggunakan algoritma Twofish sebagai pengaman pesan chat dan disimpan dalam firebase. Algoritma Twofish dapat menghasilkan chiperteks dengan panjang karakter yang berbeda dibandingkan plainteks atau pesan aslinya. Metode perancangan perangkat lunak menggunakan Rapid Application Development. Pengujian aplikasi menggunakan User Acceptance Test. Chatting dirancang menggunakan bahasa pemrograman Java dengan aplikasi Android Studio. Penelitian dari penerapan algoritma Twofish dalam perancangan aplikasi chatting berbasis android dalam kecepatan proses enkripsi dan dekripsi pesan dapat berjalan dengan baik dan fitur chatting juga dapat berjalan dengan baik. Demi perkembangan aplikasi ini agar dapat menjadi lebih baik, diharapkan agar pembaca atau programmer yang handal dapat menambah fitur pada aplikasi ini supaya pengguna lebih nyaman menggunakan aplikasi ini.*

***Kata kunci***: Chatting, Algoritma Twofish, Rapid Application Development, Android Studio, Firebase.

***Abstract***

*Chatting is two way communication between one person or several people in text, voice and video. This research was designed based on the advantages of WhatsApp, which looks good and simple so that it's easy to understand, but WhatsApp users can't see other users. Therefore, this designed chat has a look that resembles WhatsApp and can see other users. This chat application is not designed to replace WhatsApp. This chat application uses the Twofish algorithm as a security chat message and is stored in the firebase. The Twofish algorithm can generate chipertext with a different character length than a original plaintext or message. Software design method using Rapid Application Development. Application testing using the User Acceptance Test. Chats are designed using the Java programming language with Android Studio Application. The research from the application of Twofish algorithm in the designing android-based chat application in the speed of message encryption and decryption process can run well and the chat future can also run well. For the development of this application can to be better, it is expected that reliable reader or programmer can add futures to this application so that users are more comfortable using this application.*

***Keywords***: Chatting, Twofish Algorithm, Rapid Application Development, Android Studio, Firebase.

## 1. PENDAHULUAN

Aplikasi *chatting* merupakan komunikasi dua arah antara satu orang atau beberapa orang baik teks, suara, maupun video. Pemanfaatan aplikasi *chatting* ini akan menghemat waktu, tenaga, dan biaya karena tidak perlu melakukan perjalanan jauh dan melelahkan untuk menjalin komunikasi [1]. Saat ini sudah terdapat banyak aplikasi *chatting* yang tersedia dengan berbagai fitur tambahan selain fitur pengiriman pesan untuk memenuhi kebutuhan manusia seperti aktivitas mengunggah foto dan update status. Salah satu contoh aplikasi *chatting* yang populer adalah WhatsApp.

WhatsApp memiliki banyak kelebihan diantaranya tampilannya yang sederhana serta fitur yang cukup lengkap. Hal yang unik dari WhatsApp adalah fitur menambah teman dengan hanya menambah nomor ponsel dari teman yang ada. Namun, kelebihan menambah kontak hanya berdasarkan dari nomor ponsel juga menyebabkan keterbatasan dalam melihat profil pengguna yang lain. Pengguna WhatsApp tidak dapat melihat pengguna yang lain jika tidak mengetahui nomor ponselnya atau tidak melihat pengguna yang lain jika tidak mengetahui nomor ponselnya atau tidak berada di kontaknya. Hal tersebut menyebabkan keterbatasan pengenalan terhadap pengguna lain.

Berdasarkan alasan tersebut, aplikasi *chatting* yang akan dirancang ini menambah fitur dari celah cara kerja WhatsApp seperti fitur untuk dapat melihat profil pengguna yang lain. Fitur untuk melihat profil pengguna lain bermaksud untuk meningkatkan kenyamanan pengguna dalam hal membantu pengguna agar dapat mengetahui pengguna lain dan berkomunikasi dengan orang yang tidak dikenal secara lebih mudah. Selain itu tampilan *chatting* ini dibuat dengan sederhana dan mudah dimengerti oleh pemula. Aplikasi *chatting* ini juga akan menerapkan keamanan seperti WhatsApp. Akan tetapi, aplikasi *chatting* yang dirancang ini tidak bermaksud untuk menggantikan WhatsApp.

Aplikasi *chatting* ini akan menerapkan algoritma Twofish yaitu salah satu algoritma kriptografi yang dapat digunakan untuk melindungi data pada firebase sebagai tempat penyimpanan data *chatting.* Keamanan pada firebase dari aplikasi *chatting* sangat diperlukan untuk menjaga privasi dan keamanan data pengguna dari pihak yang tidak berwenang. Algoritma Twofish sebagai pengaman data pesan chat memiliki kelebihan meskipun tidak sebaik algoritma populer seperti AES secara keseluruhan yaitu keamanan dari algoritma Twofish jauh lebih kuat dibandingkan AES [2].

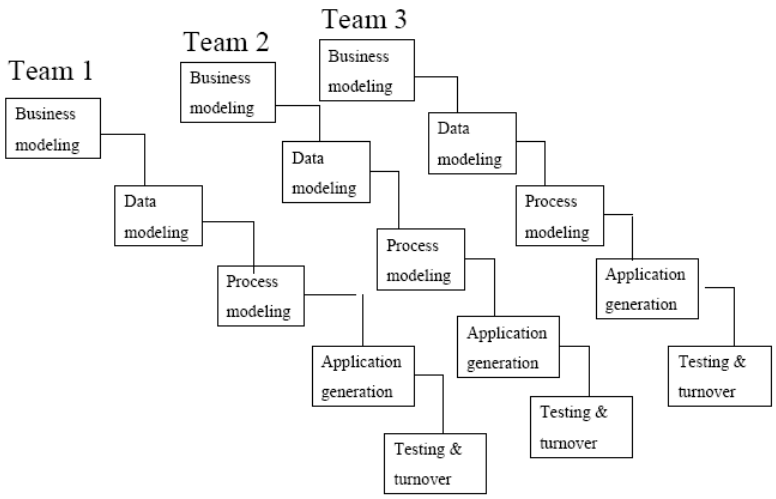
Algoritma Twofish merupakan algoritma yang diciptakan oleh Bruce Scheiner, sebelumnya beliau menciptakan Algoritma Blowfish. Algoritma Twofish merupakan salah satu kandidat AES disebabkan Twofish memenuhi semua kriteria NIST, yaitu 128-bit block, 192 bit dan 256 bit *key* atau kata kunci. Algoritma ini cukup efisien pada platform manapun termasuk android [3]. Pemilihan algoritma Twofish untuk diimplementasikan pada aplikasi *chatting* cukup baik dikarenakan proses enkripsi dan dekripsi cukup cepat sehingga tidak akan terlalu mempengaruhi proses pengiriman chat [4].

Sebelumnya telah dilakukan penelitian mengenai rancang bangun *secure chatting* menggunakan algoritma Twofish, menyatakan bahwa algoritma Twofish merupakan algoritma kuat dan sampai saat ini dinyatakan aman karena belum ada serangan kriptanalisis yang benar-benar dapat mematahkan algoritma ini [5]. Pada penelitian lain dengan penerapan Twofish pada pesan, menyatakan bahwa algoritma twofish dapat mengenkripsi secara cepat dan pemakaian memori tidak mengganggu kinerja perangkat yang dipakai [6]. Penelitian lainnya penerapan algoritma twofish pada aplikasi enkripsi dan dekripsi file menyatakan bahwa datanya menjadi tidak dapat terbaca, sehingga kriptosistem yang dirancang dapat memenuhi prinsip kriptografi [7]. Sementara pada penelitian ini akan menerapkan algoritma twofish pada aplikasi *chatting* untuk mengamankan firebase sebagai tempat penyimpanan data chat agar dapat menjaga privasi

*user* dan pesannya tidak mudah dipecahkan oleh pihak ketiga.

## 2. METODE PENELITIAN

Penelitian ini menggunakan bentuk penelitian studi literatur, yaitu mengumpulkan dan mengkaji data dengan membaca berbagai literature seperti buku, skripsi, jurnal maupun bentuk tulisan lainnya yang isinya berkaitan dengan masalah yang akan diteliti sebagai bahan referensi tertulis. Metode penelitian yang digunakan adalah metode eksperimental dengan cara membuat suatu *software* atau perangkat lunak terlebih dahulu kemudian membandingkan kebenaran dari proses yang dibuat dengan hasil yang diharapkan setelah menggunakan perangkat ini. Metode pengumpulan data yang digunakan adalah data primer dan data sekunder sedangkan teknik pengumpulan data menggunakan observasi dan studi literatur. Metode perancangan perangkat lunak menggunakan metode perancangan RAD (*Rapid Application Development*) karena proses perkembangan perangkat lunak ini menekankan pada siklus perkembangan yang singkat. Metode *Rapid Application Development* terdiri dari lima tahapan yang dimulai dari *business modeling, data modeling, process modeling, application generation, testing and ternover* [8]. Pada tahap *business modeling*, yaitu mencari dan mendefinisikan fungsi-fungsi yang akan dipakai dalam pembuatan aplikasi *chatting* dengan menggunakan algoritma Twofish berbasis android. Tahap *data modeling* harus menggunakan informasi yang didapat dalam tahap pertama untuk mementukan banyaknya modul dan form yang akan digunakan dalam perancangan aplikasi chatting. Tahap *process modeling* merupakan tahap mulai merancang form dan modul yang sudah didefinisikan sebelumnya sehingga membentuk aplikasi yang utuh. Tahap *Application Generation* merupakan penggunaan Android Studio dengan instrument penelitian berupa *flowchart*. Tahap terakhir yaitu *testing and ternover* dilakukan pengujian terhadap aplikasi untuk menguji apakah proses yang diharapkan dapat berjalan atau tidak (Gambar 1).



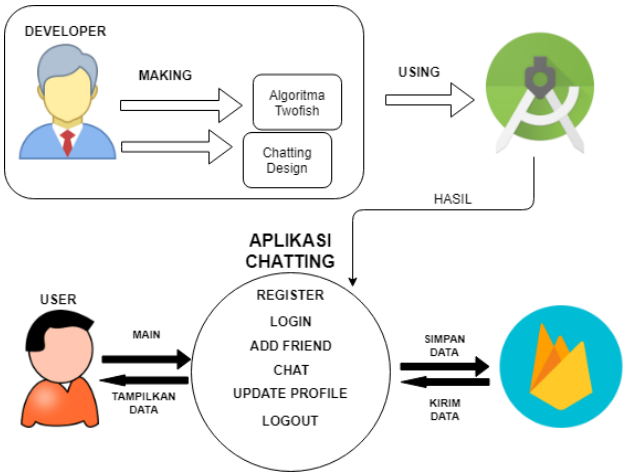

Gambar 1 Tahapan Metode *Rapid Application Development* [9].

Penelitian ini, perangkat analisis dan pemodelan sistem yang digunakan adalah UML (*Unified Modeling Language*) dan *Flowchart* untuk pemodelan proses algoritma. UML menspesifikasikan langkah-langkah penting dalam pengambilan keputusan analisis, perancangan serta implementasi dalam sistem yang sangat bernuansa perangkat lunak (*software intensive system*). UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya statis atau dinamis, seperti diagram kelas, diagram objek, *use-case diagram*, *sequence diagram*, *collaboration diagram*, *statechart diagram*, *activity diagram*, *component diagram*, dan *deployment diagram* [10].

## 3. HASIL DAN PEMBAHASAN

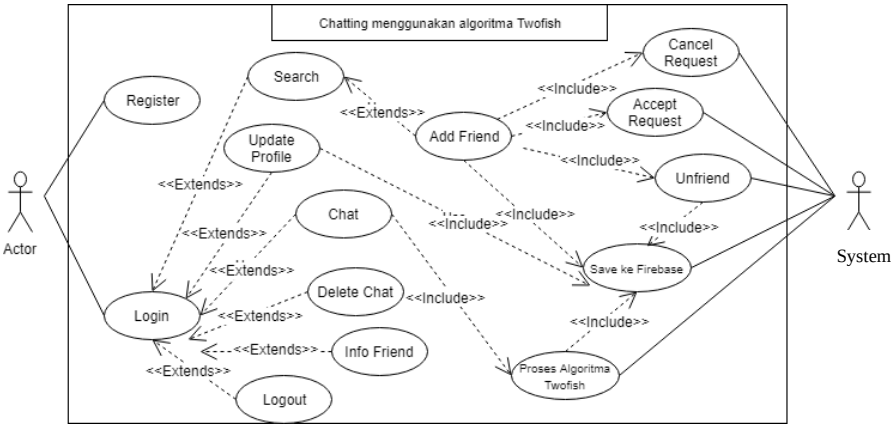
Penerapan Algoritma Twofish dalam perancangan aplikasi *chatting* berbasis android menggunakan metode *Rapid Application Development* dimana tahapannya dimulai dari tahap *process modeling* yaitu menentukan fungsi-fungsi yang akan dipakai dalam perancangan aplikasi *chatting*. Fungsi yang dipakai dalam merancang aplikasi *chatting* adalah algoritma Twofish. Algoritma ini memiliki beberapa tahap panjang dari mengenkripsi *plainteks* hingga kembali mendekripsi *plainteks*. Prosesnya akan menampilkan chiperteks dalam bentuk heksadesimal dari plainteks dan disimpan dalam firebase.

Setelah mementukan fungsi yang dipakai, tahap berikutnya adalah *data modeling* untuk menentukan berapa banyak modul dan form yang akan digunakan. Perancangan aplikasi *chatting*nya sendiri terdiri dari form awal, form *register*, form *login*, form menu utama, form *request*, form *chat*, form *friends*, form *settings*, form pencarian teman, dan form *logout*. Form awal akan menampilkan dua form pilihan yaitu form *login* dan *register*. Selain itu, gambaran komponen perangkat lunak digambarkan dalam arsitektur perangkat lunak (Gambar 2).



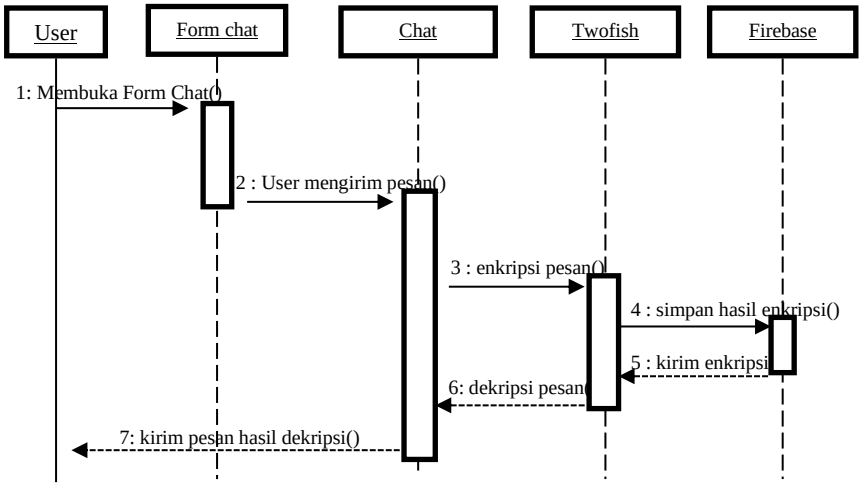

Gambar 2 Arsitektur Perancangan Aplikasi *Chatting*

Tahap selanjutnya adalah process modeling untuk merancang form dan modul yang telah ditentukan pada tahap sebelumnya. Perancangan sistem usulan aplikasi *chatting* dengan menggunakan *Unified Modeling Language* (UML) yang terdiri dari *Use Case Diagram, Sequence Diagram, Activity Diagram, Model View*, dan perancangan *interface* aplikasi *chatting*. *Use Case Diagram* berfungsi untuk menjelaskan manfaat dan fungsionalitas suatu sistem atau kelas dari bagaimana sistem berinteraksi dengan dunia luar. Diagram ini menunjukkan bahwa *user* yang telah login akan memiliki otoritas untuk mengelola aktifitas *chatting* dan informasi akun (Gambar 3).



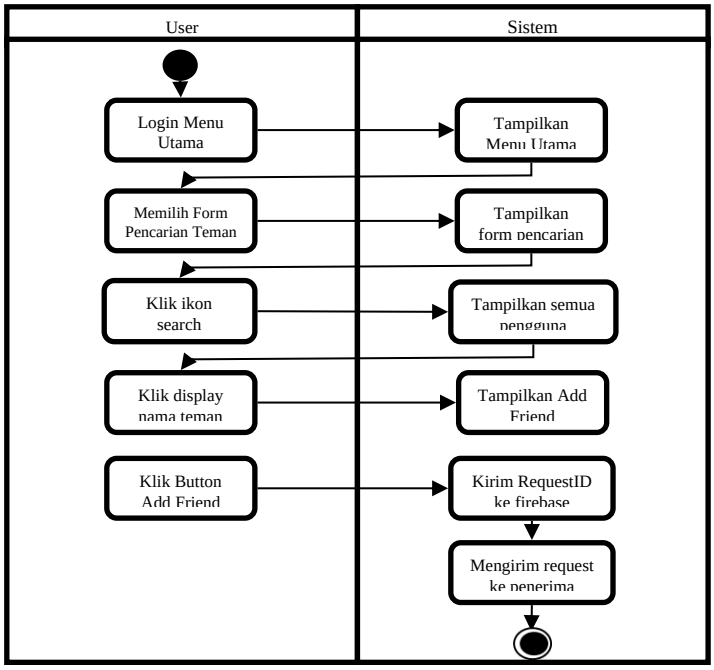
Gambar 3 *Use Case Diagram* Sistem

*Sequence Diagram* proses algoritma Twofish menjelaskan alur dari proses pengiriman chat dan dilakukan proses enkripsi hingga mencapai firebase dan proses pengembalian data pesan yang telah dienkripsi kembali menjadi pesan asli. Proses algoritma Twofish dilakukan setelah *users* mengirim pesan dalam bentuk teks maupun gambar sebelum disimpan dalam firebase. Dalam firebase akan tersimpan data *messageID* bertipe sesuai dengan tipe pesan yang dikirim yaitu teks atau gambar. Selain itu pesan akan didekripsi kembali sebelum dikirim kembali ke penerima (Gambar 4).



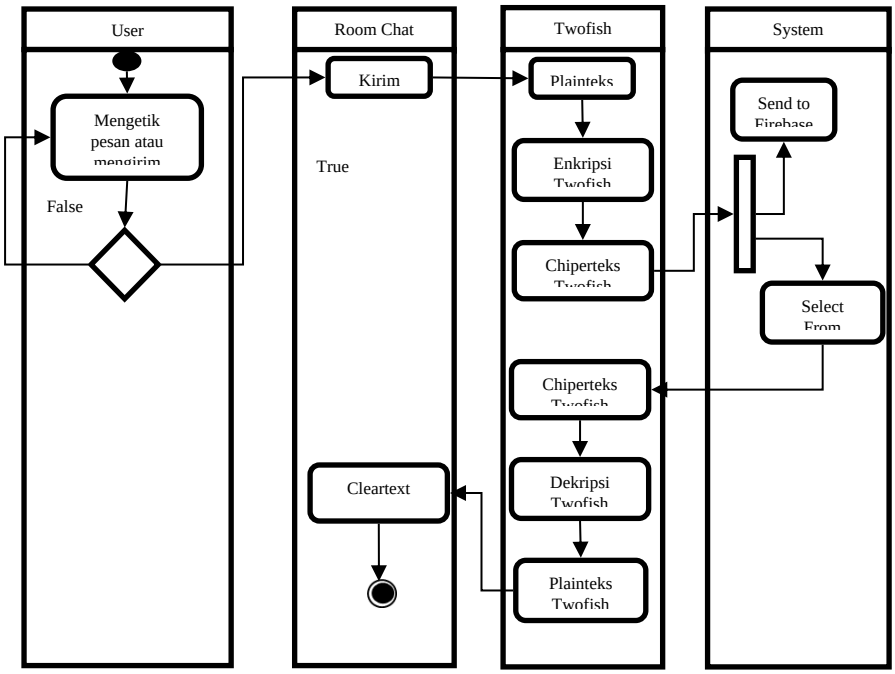

Gambar 4 *Sequence Diagram* Proses Pengiriman Pesan

*Activity diagram* mencari semua pengguna dan menambah pertemanan, pertama *user* harus klik ikon *search* yang terdapat pada form menu utama bagian atas, maka sistem akan menampilkan form pencarian teman. Pada form pencarian teman, *user* dapat memilih untuk mencari teman berdasarkan dengan mengetik nama teman atau melihat semua pengguna terlebih dahulu, untuk melihat semua pengguna, user tinggal klik ikon *search* pada form pencarian teman, kemudian sistem akan menampilkan semua nama pengguna. Untuk menambah pertemanan, *user* harus klik *display* nama *user*, maka sistem akan menampilkan pilihan *button add friend*. Saat pengguna klik *add friend*, maka sistem akan mengirim *requestID* ke firebase untuk menyimpan data dan dikirim kembali kepada penerima untuk ditampilkan pada form *request friend* penerima (Gambar 5).



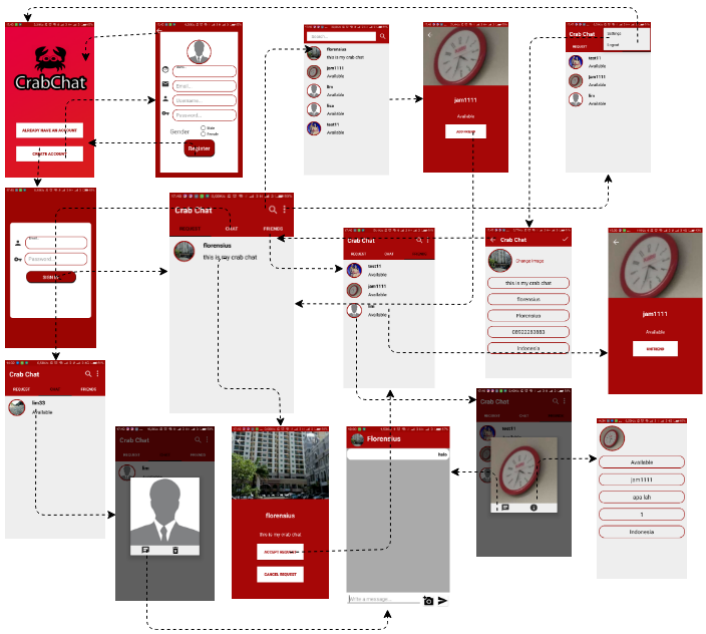

Gambar 5 *Activity Diagram* Tampilkan semua pengguna

*Activity diagram* proses *chatting*, ketika *user* mengirimkan pesan, maka pesan sebelum dikirim ke firebase akan dienkripsi terlebih dahulu dengan algoritma Twofish lalu hasil *chipertext* Twofish akan dikirim ke firebase. Firebase akan menyimpan data *messageID* dan ID dari *user sender* dan *Receiver* (Gambar 6).




Gambar 6 *Activity Diagram* Proses *Chatting*

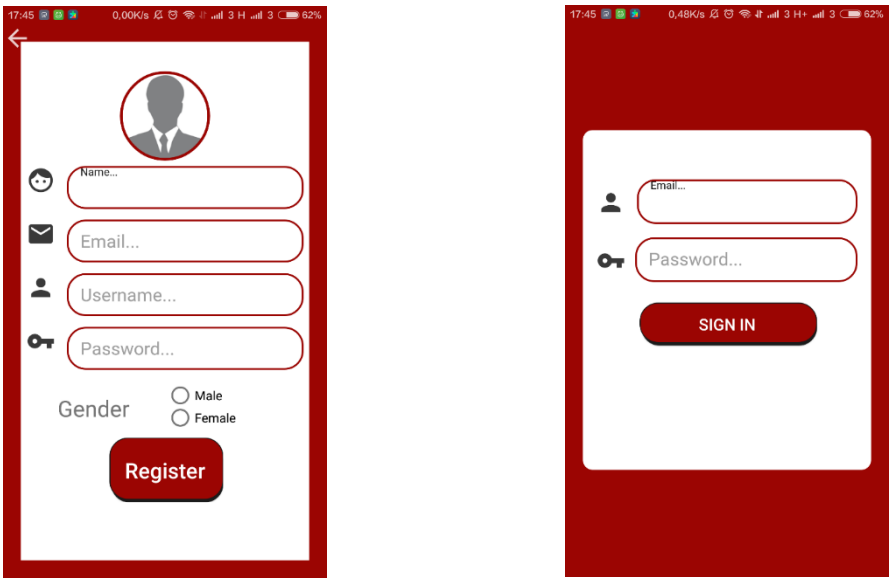
*Model View* dimaksudkan untuk memberikan gambaran alur *feedback* kepada *user* jika suatu tombol ditekan. Gambaran alur tersebut dapat menjelaskan detail proses cara kerja aplikasi. Berikut adalah *model view* dari aplikasi *chatting* (Gambar 7).




Gambar 7 *Model View* Aplikasi Chatting

Berikut ini merupakan sejumlah antarmuka dari aplikasi *chatting* yang dihasilkan. Beberapa diantaranya adalah form *register*, form *login*, form pencarian teman, *room chat*, dan *settings*.

Rancangan antarmuka register berisikan form *name*, *email*, *username*, *password*, dan *gender*. Form ini berfungsi untuk mendaftarkan akun baru untuk mendapatkan akses aplikasi *chatting*. Rancangan antarmuka halaman login berisikan form *email* dan *password* untuk diinput berdasarkan akun yang telah didaftarkan pada form *register* (Gambar 8).

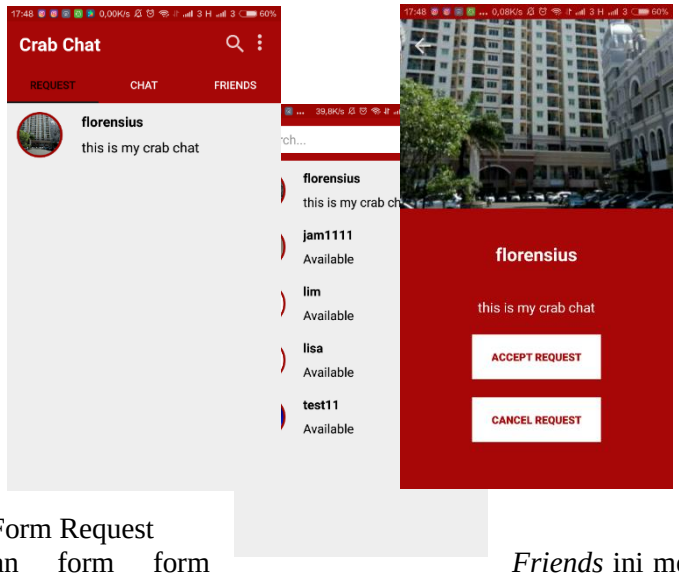



Gambar 8 Form Register dan Login

Rancangan antarmuka form pencarian teman dapat menampilkan semua pengguna agar *user* dapat memilih user lain yang ingin dijadikan teman. Pada kolom bertulisan *hint* "Search…" berfungsi untuk mencari nama *user* dengan mengetik nama *user* secara manual (Gambar 9).

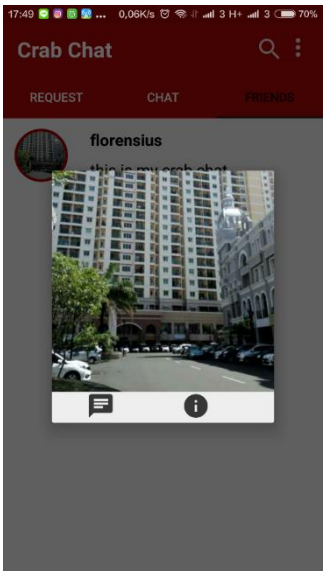
Gambar 9 Form Pencarian Teman

Rancangan form request menampilkan nama pengguna lain yang telah mengirimkan permintaan pertemanannya yang bertujuan agar dapat saling berkomunikasi. Pengguna dapat melihat dua menu disini, yaitu *accept request* atau *cancel request*. *Accept request* bertujuan untuk menerima permintaan pertemanan sehingga tampilan nama teman akan berpindah ke form *Friend*. Sedangkan *cancel request* bertujuan untuk menolak permintaan pertemanan sehingga akan menghapus tampilan teman pada form *request* (Gambar 10).



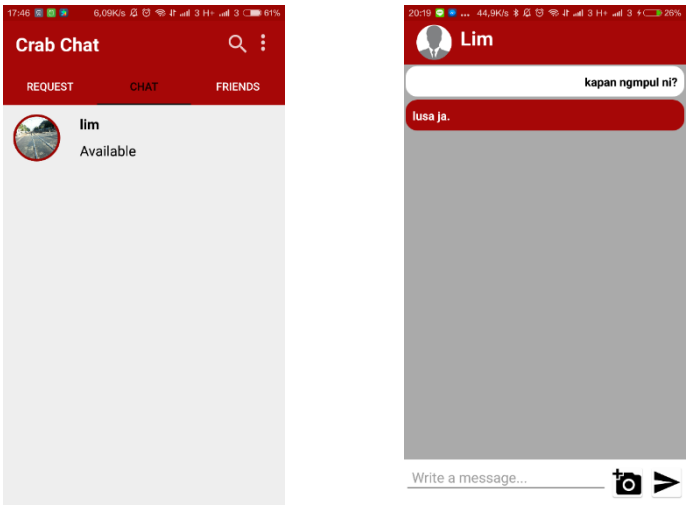

Gambar 10 Form Request

Rancangan form form *Friends* ini menampilkan *friendlist* atau daftar teman yang telah berteman dengan pengguna. Terdapat beberapa fungsi menu pada form *friends*, yaitu pengguna dapat *unfriend* teman yang diinginkan sehingga namanya akan terhapus dari *friend list*. Selain itu, pengguna dapat melihat profil teman untuk lebih mengenalnya. Profil yang dilihat oleh *user* berasal dari data *settings* yang telah diinput (Gambar 11).



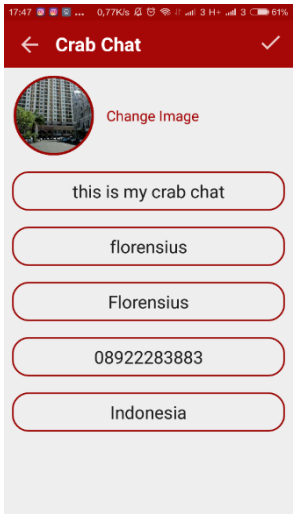

Gambar 11 Form Friends

Rancangan antarmuka form Chat menampilkan nama pengguna lain yang telah dikirim pesan oleh *user*. Di form ini terdapat dua pilihan yaitu melihat *room chat* atau *delete chat.* Dalam room *chat*, pengguna dapat mengirim pesan teks dan pesan gambar. Balon obrolan pada *room chat* bagi pengguna pengirim pesan berwarna putih, sedangkan bagi penerima adalah berwarna merah. Sedangkan *delete chat* berfungsi untuk menghapus *chat list* dari from chat. (Gambar 12).




Gambar 12 Halaman *Room Chat*

Rancangan antarmuka *settings*, halaman ini berfungsi untuk mengelola data akun *user*, dari fungsi mengganti foto profil, status, nama, *username*, nomor handphone, dan asal negara (Gambar 13).




Gambar 13 Form Settings

Tahap *application generation* untuk sistem yang sebenarnya dibangun dan pengkodean dilakkan dengan menggunakan *automatic tools* untuk mengubah data modeling dan process modeling menjadi *prototype* yang actual. Rancangan aplikasi *chatting* ini menggunakan bahasa pemrograman Java dengan aplikasi Android Studio versi 3.1.2. Instrumen penelitian yang dilakukan berupa *flowchart*.

Tahapan *testing and ternover* merupakan tahapan untuk pengujian keseluruhan sistem yang dibangun. Semua komponen perlu diuji secara menyeluruh dengan cakupan uji yang lengkap untuk mengetahui integritas serta fungsionalitas program tersebut, termasuk didalamnya pengujian validitas input tiap form. Adapun pengujian yang dilakukan menggunakan *Acceptance Test.*

Tabel pengujian *User Acceptance Test* oleh pengguna aplikasi. Responden terdiri dari 20 orang mahasiswa yang diberikan kuisioner mengenai hasil dari aplikasi *chatting* yang telah dicoba. Responden merupakan mahasiswa kelas C4 STMIK Pontianak (Tabel 1).

**SB = Sangat Baik**
**B = Baik**
**C= Cukup**
**K = Kurang**
**SK = Sangat Kurang**

Tabel 1 Tabel Hasil Kuisioner Pengujian *Acceptance Test*

| No | Pertanyaan | Skala | | | | |
|---|---|---|---|---|---|---|
| | | SB | B | C | K | SK |
| 1 | Apakah tampilan chatting sudah menarik? | 2 | 13 | 4 | 1 | 0 |
| 2 | Apakah chatting ini mudah digunakan? | 4 | 10 | 6 | 0 | 0 |
| 3 | Apakah fitur pada chatting sudah lengkap? | 0 | 6 | 11 | 3 | 0 |
| 4 | Apakah nama chatting sudah sesuai dengan tema? | 3 | 14 | 3 | 0 | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **5** | Apakah logo chatting sudah sesuai dengan nama dan tema? | **9** | **9** | **2** | **0** | **0** |
| **6** | Apakah tampilan aplikasi chatting ini menyerupai WhatsApp? | **5** | **11** | **4** | **0** | **0** |
| **7** | Apakah fitur untuk mencari pengguna lain dapat berjalan dengan baik? | **4** | **15** | **1** | **0** | **0** |
| **8** | Apakah tampilan room chat sudah baik? | **3** | **7** | **10** | **0** | **0** |
| **9** | Apakah hasil pesan dan gambar yang ditampilkan sesuai dengan yang dikirim? | **4** | **12** | **2** | **2** | **0** |
| **10** | Apakah semua fitur dan fungsi pada chatting dapat berjalan dengan benar? | **1** | **10** | **6** | **1** | **0** |
| **11** | Apakah proses pengiriman pesannya cepat? | **11** | **8** | **1** | **0** | **0** |

Dari hasil tabel kuisioner responden diatas dapat disimpulkan bahwa tampilan *chatting* yang dirancang sudah cukup baik (76%). Selain itu, *chatting* yang dirancang ini cukup mudah digunakan bagi responden (78%). Dilihat dari jumlah fitur yang dirancang untuk aplikasi *chatting* masih belum lengkap jika dibandingkan aplikasi *chatting* yang sudah ada (63%). Berdasarkan nama aplikasi *chatting* yang dirancang yaitu Crab Chat, penamaannya sudah sesuai dengan tema aplikasi (76%). Selain itu, dari logo aplikasi *chat* yang didesain juga sudah sesuai dengan nama dan tema aplikasi (80%). Berdasarkan sisi tampilan *chatting* yang dirancang sudah sedikit menyerupai WhatsApp (81%). Fitur aplikasi untuk mencari pengguna lain dapat berjalan dengan baik (83%). Berdasarkan dari sisi *room chat* aplikasi yang dirancang, tampilannya sudah cukup baik (73%). Hasil pesan dan gambar yang diterima *user* lain sudah sesuai dengan yang dikirim. Dari semua fitur chat yang dirancang, tidak semua fitur dan fungsinya dapat berjalan dengan baik (65%). Selain itu, proses pengiriman pesan chat sudah cepat (90%)

## 4. KESIMPULAN

Berdasarkan hasil penerapan algoritma twofish dalam perancangan aplikasi *chatting* berbasis android, maka penulis mengambil kesimpulan bahwa algoritma Twofish cocok digunakan untuk aplikasi *chatting* dikarenakan waktu proses untuk enkripsi dan dekripsi relatif cepat dan aplikasi *chatting* berhasil dirancang dengan menggunakan Android Studio dan diberi nama Crab Chat. Selain itu, dari sisi tampilan *chatting* berhasil dibuat menyerupai WhatsApp dengan tampilannya yang sederhana dan mudah digunakan oleh pengguna.

## 5. SARAN

Dari hasil penelitian yang dilakukan, disadari masih banyak kekurangan dalam sistem. Oleh sebab itu, diharapkan bagi pembaca atau programmer yang handal agar dapat menambah jumlah fitur seperti *video call, group chat*, dan *voice note* serta dapat menambah fitur *Unsend* pada aplikasi *chatting* ini. Selain itu, diharapkan dapat meningkatkan tampilan aplikasi *chatting* sehingga lebih menarik bagi pengguna.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada orang tua penulis yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

[1] Putri, K., 2011, Analisis Perancangan dan Implementasi Aplikasi Chatting Berbasis Objek, *Skripsi*, Fakultas Sains dan Teknologi, Uin Sunan Kalijaga, Yogyakarta.

[2] Viqarunnisa, P., 2017, Studi dan Perbandingan Algoritma Rijdael dan Twofish, *Skripsi*, Program Studi Teknik Informatika, Institut Teknologi Bandung, Bandung.

[3] Indra, M., 2015, Algoritma Twofish: Kinerja dan Implementasinya Sebagai Salah Satu Kandidat Algoritma AES, *Skripsi*, Jurusan Teknik Informatika, ITB, Bandung.

[4] Kusumah, A., Secure Chatting (Instant Messaging) Menggunakan Metode Enkripsi Blowfish, Twofish, dan AES, *Skripsi*, Fakultas Teknik Informatika, Universitas Telkom, Bandung.

[5] Radhiah, A., 2014, Rancang Bangun Secure Chatting pada Platform Android, *Skripsi*, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekan Baru.

[6] Midian, O.G, 2016. Aplikasi Pesan dengan Algoritma Twofish pada Platform Android, *Journal e-Proceeding of Engineering*, No.3, Vol.3, 2355-9365.

[7] Yunarti, S., 2016, Aplikasi Enkripsi dan Dekripsi File Menggunakan Algoritma Twofish, *Skripsi*, Program Studi Sistem Informasi, STMIK Profesional, Makassar.

[8] Pressman, Roger S, 2012, *Rekayasa Perangkat Lunak*, Jilid 1, Andi Offset, Yogyakarta.

[9] Roger, S.Pressman. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu)*. Yogyakarta : Andi

[10] Herlawati Widodo, 2011. *Menggunakan UML*. Informatika, Bandung.