IMPLEMENTASI ALGORITMA KRIPTOGRAFI DENGAN ALGORITMA CAESAR CIPHER, ADVANCED ENCRYPTION STANDARD 256, DAN RC6 UNTUK APLIKASI CHATTING BERBASIS ANDROID

Dimas Bayu Gumelar[1)], **Wahyu Pramusinto**[2)]
[1]Program studi, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : dbayu15@gmail.com[1)], wahyu.pramusinto@budiluhur.ac.id[2)]

Abstrak

Dengan semakin pesatnya alat yang manusia gunakan untuk berkomunikasi, mulai dari, telepon, telepon genggam, komputer, dan sekarang smartphone, melahirkan begitu banyak inovasi teknik dan metode yang berbeda-beda pada setiap alat yang digunakannya. Misalnya semenjak ditemukannya email pada komputer, hal itu merenovasi hampir keseluruh teknik komunikasi menggunakan alat tersebut. Hingga akhirnya teknik email yang dirasa kurang efektif untuk komunikasi Real Time, membuat para ahli berhasil menemukan solusi yang diinginkan oleh banyak orang dengan munculnya aplikasi Chatting. Seiring dengan banyaknya aplikasi chatting, kejahatan cybercrime seperti penyadapan dan manipulasi pesan juga banyak terjadi. Oleh karna itu, penelitian ini akan membuat aplikasi chatting dimana pesan yang disampaikan dapat terjaga kerahasiaannya, karena aplikasi ini menggunakan beberapa metode enkripsi, antara lain metode enkripsi Caesar Cipher, Advanced Encryption Standard 256, dan RC6. *Pesan yang berupa teks dan gambar (plaintext), melalui proses enkripsi menjadi pesan acak yang tak bisa dibaca (chipertext) menggunakan kunci yang telah ditentukan oleh sistem aplikasi. Pesan hanya dapat dibaca melalui proses dekripsi pada aplikasi chatting ini sehingga kerahasiaan pesan tetap terjaga.*

Kata kunci: *Caesar Cipher*, AES, RC6, *Firebase*, *Chatting*

1. PENDAHULUAN

1.1. Latar Belakang

Komunikasi di era teknologi informasi tidak lagi harus dilakukan dengan cara bertemu langsung atau bertatap muka. Komunikasi dapat dilakukan dengan beragam bantuan baik perangkat keras maupun perangkat lunak. Salah satu bentuk komunikasi yang sering dilakukan adalah menggunakan teks. Pengiriman informasi melalui teks dapat dilakukan dengan fasilitas *e-mail*, *chatting*, sms, dan bentuk komunikasi lain berbasis teks. Dengan berkirim pesan melalui teks, pesan dapat sampai dari sisi pengirim ke sisi penerima. Atas dasar itulah dibuatnya aplikasi *instant messaging* atau yang biasa disebut aplikasi *chatting*. Aplikasi *chatting* merupakan aplikasi yang mengijinkan penggunanya dapat mengirimkan pesan secara satu waktu atau *real time* yang membuat jarak yang jauh seolah-olah tidak berarti di dunia internet. Aplikasi *chatting* banyak dilakukan karena penggunaannya yang relatif mudah, serta dalam keadaan yang sibuk masih tetap dapat memanfaatkan aplikasi tersebut. Adanya internet juga semakin banyak diminati karena mudah digunakan, dan dapat diakses setiap orang dari berbagai kalangan. Oleh karena itu dengan memanfaatkan layanan internet, aplikasi dapat berjalan dengan cepat, mudah, tanpa perlu menunggu lama balasan dari orang yang dituju dan cara bertukar pesan sangat praktis.

Saat ini teknologi *chatting* sudah berkembang dengan cepat, dan banyak aplikasi *chatting* yang sudah terkenal seperti *Whatsapp*, *BBM*, *Line*, dan lainnya yang memiliki kelebihan masing-masing. Namun aplikasi-aplikasi informasi tersebut masih beresiko untuk dapat dimengerti, diketahui, hingga dicuri oleh pihak yang tidak berkepentingan. Hal ini dapat merugikan bagi pengguna, bila pesan yang dikirimkan berisikan informasi yang bersifat rahasia, khususnya bagi perusahaan. Dalam komunikasi data, melakukan pertukaran data yang rahasia adalah hal yang penting, tidak terlepas dari tujuan keamanan suatu perusahaan maupun terhadap privasi individu. Menyiasati cara untuk mengamankan informasi yang akan dikomunikasikannya merupakan keinginan dari mereka yang menginginkan agar datanya tidak dapat dilihat maupun diketahui oleh berbagai pihak yang tidak berkepentingan. Salah satu cara menyiasati untuk mengamankan informasi tersebut yaitu dengan menyandikan data komunikasi atau enkripsi.

Hampir semua orang membutuhkan pengamanan untuk bertukar informasi saat melakukan percakapan dimana percakapan yang dilakukan tidak hanya percakapan yang bersifat biasa, tapi saat ini hampir semua orang melakukan percakapan yang bersifat

rahasia dan tidak boleh diketahui oleh pihak yang tidak bertanggung jawab melalui aplikasi *chatting*. Oleh karna itu, penelitian ini akan membuat aplikasi *chatting* dimana pesan yang disampaikan dapat terjaga kerahasiaannya, karena aplikasi ini menggunakan beberapa metode enkripsi, antara lain metode enkripsi *enkripsi Caesar Cipher, Advanced Encryption Standard* 256, dan RC6. Aplikasi chatting yang dibuat berbasis android dengan bahasa pemrograman Java. Dan dari hasil penelitian ini dapat memudahkan kinerja lembaga tersebut tanpa harus kuatir pesan yang disampaikan diketahui pihak yang tidak berkepentingan, terlebih jika pesan tersebut berisi informasi yang sedang di-*share*.

1.2. Batasan Masalah

Demi tercapainya tujuan penelitian ini agar tetap terjaga dan tidak keluar pembahasan maka dibuatlah pembatasan masalah sebagai berikut :

a. Algoritma kriptografi yang digunakan adalah *Rivest Cipher* 6 (RC6), *Advanced Encryption Standart* (AES) 256 dan *Caesar Cipher*.
b. Aplikasi yang dibuat berbasis Android dan *database server* menggunakan *Firebase*.
c. Isi pesan yang dienkripsi berupa teks dengan algoritma *Rivest Cipher* 6 (RC6) dan *Advanced Encryption Standart* (AES) 256, serta *link* gambar dengan algoritma *Caesar Cipher*.

1.3. Tujuan Penulisan

Sebagai acuan penelitian, maka peneliti membuat tujuan penulisan sebagai berikut :

a. Membuat aplikasi *chatting* berbasis *Android* dengan menggunakan tiga algoritma kriptografi.
b. Menerapkan algoritma *Rivest Cipher* 6 (RC6), *Advanced Encryption Standart* (AES) 256 sebagai pengamanan untuk pesan teks.
c. Menerapkan algoritma *Caesar Cipher* sebagai pengamanan untuk pesan gambar.

2. LANDASAN TEORI

2.1. Komunikasi

Arti dari komunikasi dalam bahasa inggris yaitu "*Communication*", yang berasal dari bahasa latin yaitu *communicates*, yang sumbernya berasal dari kata *communis*. Arti dari kata *communis* ini memiliki makna berbagi atau menjadi milik bersama yaitu usaha yang dilakukan demi kepentingan atau tujuan kebersamaan atau kesamaan makna.

Maksud dari komunikasi dalam Kamus Besar Bahasa Indonesia (KBBI terbitan Balai Pustaka, 2002), komuikasi merupakan Pengiriman dan penerimaan pesan antara dua orang atau lebih sehingga pesan yang dimaksud dapat dipahami, hubungan, kontak. (Anon, 2013)

2.2. Kriptografi

a. Sejarah Kriptografi

Kriptografi atau yang disebut dengan *Cryptography* merupakan bahasa Yunani yaitu "cryptos" yang memiliki arti "secret" atau rahasia, sedangkan "graphein" yang memiliki arti "writing" atau tulisan. Kriptografi sudah ada sejak 300 tahun SM dan memiliki sejarah yang panjang dan juga mengagumkan saat digunakan oleh bangsa Mesir. Bangsa Mesir yang saat itu menggunakan teknik *Hieroglyphcs* dalam menyembunyikan pesan mereka dari orang-orang yang tidak diharapkan. Sekitar 400 SM, Bangsa Spatan menggunakan kriptografi militer yang dibuat dalam bentuk sepotong papiru atau perkamen yang dibungkus dengan sebatang kayu, yang saat ini system itu disebut dengan *Scytale*. (Pandiangan, H & Sijabat, S, 2016)

b. Definisi Kriptografi

Kriptografi merupakan bidang ilmu yang mempelajari cara untuk mengamankan pesan dan atau informasi dengan teknik penyandian, sehingga pesan tidak dapat dimengerti lagi maknanya. Definisi ini cocok pada masa dimana kriptografi dipergunakan untuk melakukan kemanan dalam berkomunikasi penting, seperti komunikasi yang dilakukan oleh kalangan militer, diplomat dan juga mata-mata. (Pratama, 2013)

Menurut Harahap (2016) algoritma kriptografi adalah cara logis bagaiman melakukan penyembunyian pesan dari orang-orang yang tidak berhak untuk mengetahui pesan tersebut. Ada tiga fungsi dasar dari kriptografi, diantaranya:

1) Enkripsi

Pesan asli atau yang disebut *plaintext* diproses dan diubah menjadi kode-kode yang tidak dapat dimengerti. *Cipher* atau kode bisa diartikan dengan enkripsi. Sama seperti saat kita tidak paham akan sebuah kalimat atau kata maka kita akan melihatnya dalam kamus atau daftar istilah. Berbeda dengan enkripsi, kita menggunakan algoritma yang dapat mengkodekan data yang diinginkan untuk mengubah teks asli menjadi *cipher* atau ke bentuk kode.

2) Dekripsi

Adalah kebalikan dari enkripsi, yaitu pesan atau kalimat yang telah dienkripsi dikembalikan ke bentuk asalnya dengan menggunakan algoritma yang sudah ditentukan. Adapun algoritma yang digunakan untuk melakukan proses dekripsi merupakan algoritma yang berbeda dari proses enkripsi, hanya saja kedua algoritma ini harus saling berpasangan untuk bisa merubah hasil enkripsi ke dekripsi

3) Kunci

Maksud dari kunci adalah sebuah kata-kata yang dipakai untuk melakukan proses enkripsi dan dekripsi. Ada dua bagian dari kunci, yaitu kunci rahasia atau yang disebut *private key* dan kunci umum atau *public key*.

Keamanan algoritma kriptografi bergantung kepada bagaimana algoritma tersebut bekerja.

Keamanan dari kriptografi modern juga didapat dari cara merahasiakan kunci yang dimiliki, tanpa harus merahasiakan algoritma yang digunakan. Kunci pada algoritma kriptografi memiliki fungsi yang sama dengan *password*. Jika keamanan pada algoritma kriptografi tergantung pada kunci yang digunakan, maka algoritma kriptografi bisa dipublikasikan dan dianalisis oleh orang lain. Jika algoritma kriptografi yang telah dipublikasikan dapat dipecahkan perhitungannya dalam waktu yang singkat oleh orang lain maka algoritma kriptografi tersebut sudah tergolong algoritma yang tidak aman untuk digunakan untuk mengamankan data. Dalam kriptografi, sudah banyak sekali ditemukan berbagai macam metode untuk mengenkripsi file, mulai dari teks, dokumen, suara, video. (Harahap, 2016)

2.3. Caesar Cipher

Algoritma Caesar Cipher termasuk kriptografi klasik yang menggunakan *plainteks*, *cipherteks* dan kunci untuk melakukan proses enkripsi dan dekripsi dalam pengamanan data. Caesar Cipher adalah salah satu algoritma tertua dan sangat diketahui dalam perkembangan kriptografi. Caesar Cipher adalah salah satu jenis *Cipher* subtitusi yang membuat *Cipher* dengan cara melakukan pertukaran karakter pada teks awal (*Plaintext*) menjadi tepat satu karakter pada *Ciphertext*. Teknik pada *Caesar Cipher* juga disebut sebagai *cipher* abjad tunggal. Algoritma ini juga sangat mudah untuk diterapkan karena struktur penyandiannya yang tidak rumit. Inti dari proses algoritma *Caesar Cipher* adalah melakukan pergeseran terhadap karakter yang akan diubah dengan nilai pergeseran yang sama. Adapun langkah yang dilakukan dalam membentuk *Ciphertext* dengan algoritma *Caesar Cipher* adalah menentukan besar pergeseran karakter yang ingin dilakukan untuk membentuk *Ciphertext* ke *Plaintext*, melakukan pertukaran karakter pada *Plaintext* menjadi *Ciphertext* dengan pergeseran yang telah ditentukan sebelumnya. (Priyono, 2016)

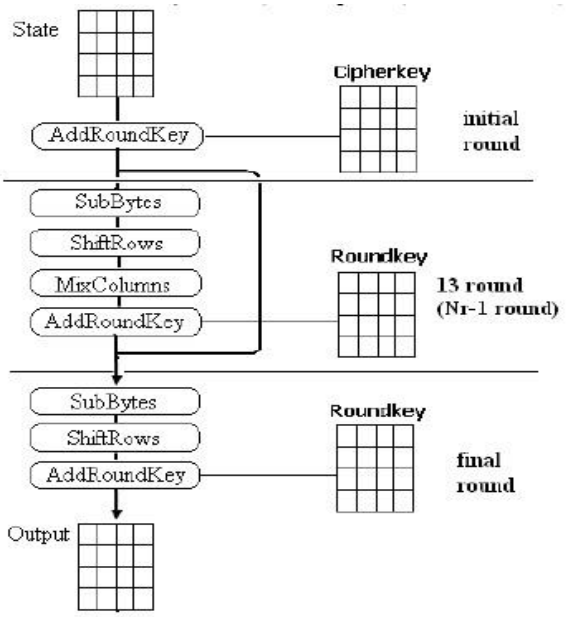
2.4. Advanced Encryption Standard (AES) 256

AES merupakan salah satu algoritma yang dipublikasiakan oleh NIST (*National Institute of Standard and Technology*) pada tahun 2001 untuk menggantikan algoritma DES yang sudah dianggap kuno dan mudah dipecahkan. AES memiliki panjang kunci sebanyak 128 bit, 192 bit, 256 bit. Perbedaan pada panjang kunci AES akan mempengaruhi jumlah putaran dan juga pembentukan kunci pada algoritm ini. Algoritma AES memiliki beberapa tahapan dalam melakukan penyandian pesan, tahapan – tahapan tersebut diulang sesuai dengan panjang kunci yang digunakan. Ada tiga macam jumlah putaran yang dimiliki algoritma AES seperti tabel dibawah ini :

Tipe	Panjang Kunci	Panjang Blok Input	Jumlah Putaran
AES-128	128 bit	128 bit	10
AES-192	192 bit	128 bit	12
AES-256	256 bit	128 bit	14

Tabel 1 : Jumlah Putaran pada Algoritma AES

ada empat jenis transformasi yang digunakan pada proses enkripsi algoritma AES yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pertama, inputan yang telah di-*copy*-kan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak Nr. *Round* yang terakhir agak berbeda dengan sebelumnya dimana pada *round* terakhir, state tidak mengalami transformasi *MixColumns*. (Bhaudhayana, G.W. & Widiarhta I.M., 2015)



Gambar 1 : *flowchart* algoritma AES 256 bit

2.5. Rivest Cipher 6 (RC6)

Algoritma RC6 merupakan salah satu algoritma simetris yang dibuat dan dikembangkan oleh Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, dan Y.L. Yin adalah salah satu algoritma yang terpilih menjadi kandidat untuk menjadi algoritma AES. Algoritma ini cukup diakui kesederhanaannya dan kesesuaiannya untuk diimplementasikan pada prosesor dengan arsitektur ARM yang banyak dipakai pada telepon selular dibandingkan dengan

finalis yang lainnya pada saat itu. (Yusfrizal, 2015) algoritma RC6 adalah kelanjutan dari algorimat yang sebelumnya diciptakan yaitu RC5 dan telah memenuhi semua kriteria yang ada pada NIST. RC6 merupakan algoritma yang menggunakan ukuran blok sampai 128 bit, dan ukuran kunci yang digunakan oleh RC6 bervariasi antara 128, 192, dan 256 bit (Defni, Rahmayun, 2014)

Menurut Yusfrizal (2015) Langkah-langkah enkripsi algoritma RC6 secara detil adalah sebagai berikut :

a. Blok plainteks dibagi menjadi 4 bagian A, B, C dan D yang masing-masing memiliki panjang w bit atau panjang blok dibagi 4. Kemudian B dan D dijumlahkan (dalam *modulo* 2w) dengan kunci internal S[0] dan S[1].
b. Selanjutnya pada setiap putaran dari 1 sampai r, lakukan XOR dan pergeseran ke kiri terhadap A dengan f(x) yang di geser ke kiri sebanyak lg w, di mana f(x) = x* (2x+1) dan x = B. Setelah itu melakukan penjumlahan (dalam *modulo* 2w) dengan kunci internal. Hal serupa dilakukan pula terhadap C dengan x = D. Kemudian melakukan swapping A B, B C, C D dan D A.
c. Setelah iterasi selesai langkah terakhir adalah melakukan penjumlahan (dalam *modulo* 2w) terhadap A dan C dengan dua kunci internal terakhir. Setelah semua selesai blok yang terbagi menjadi 4 bagian disatukan kembali.

3. ANALISA MASALAH DAN PERANCANGAN PROGRAM

3.1. Analisa Masalah

Kebutuhan masyarakat yang semakin meningkat dan perkembangan teknologi informasi yang terus berkembang khususnya dalam bidang komunikasi membuat aplikasi *instant messaging* semakin banyak beredar baik untuk kepentingan pribadi maupun perusahaan. Informasi yang terkandung didalam pesan menjadi sebuah hal yang penting bagi sebuah perusahaan, tidak terkecuali pada perusahaan maupun perorangan yang menggunakan aplikasi *chatting* sebagai perantara untuk berkomunikasi karena pesan yang dikirimkan tidak hanya berupa pesan biasa tapi juga bisa berisi informasi penting dan rahasia yang tidak boleh diketahui oleh pihak yang tidak berkepentingan. Dengan demikian aplikasi *chatting* merupakan aplikasi yang sangat penting bagi semua orang yang melakukan percakapan penting tanpa harus bertemu secara langsung.

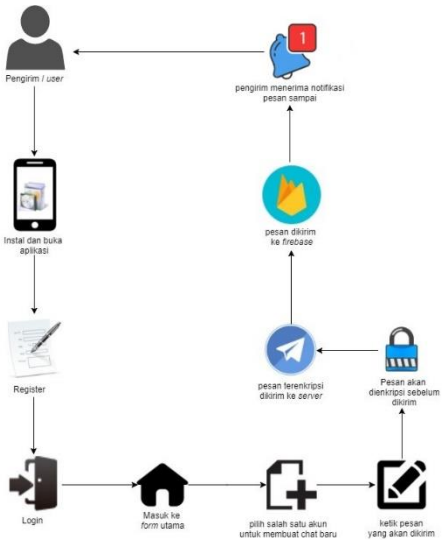
3.2. Pemecahan Masalah

Dari masalah yang telah diuraikan di atas, penulis dapat membuat sebuah aplikasi *chatting* yang sesuai dengan kebutuhan dan memiliki keamanan dalam pengiriman pesan teks dan gambar. Dalam membuat aplikasi *chatting* tersebut, digunakan sebuah metode yang disebut kriptografi. Penerapan kriptografi pada aplikasi ini diharapkan agar informasi pada pesan yang dikirimkan tidak diketahui oleh pihak yang tidak berkepentingan. Isi percakapan yang di enkripsi antara pengirim dan penerima akan disimpan pada *Firebase Console*. Algoritma kriptografi yang digunakan pada aplikasi *chatting* ini adalah algoritma *Caesar Cipher, Advanced Encryption Standard* 256, dan RC6. Aplikasi yang dibuat berbasis Android yang dapat diakses menggunakan jaringan. Kebutuhan dapat terpenuhi dengan adanya tiga metode enkripsi yang digabung menjadi satu untuk mengamankan percakapan.

3.3. Skema Proses Keseluruhan Aplikasi

Untuk menyelesaikan masalah diatas, maka diuraikanlah skema proses keseluruhan aplikasi *chatting*. Berikut adalah tahapan dan *rich picture* pada proses pengiriman pesan :

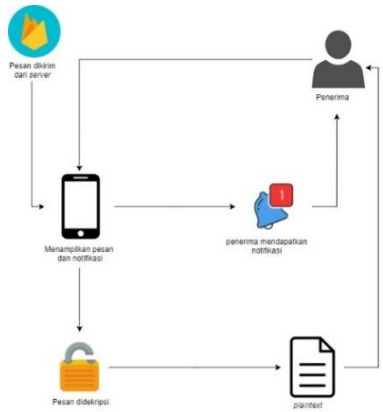
1) Langkah awal untuk menggunakan aplikasi ini adalah meng-*install* dan membuka aplikasi telah ter-*install* di perangkat Android.
2) Apabila pengirim belum memiliki akun aplikasi ini, maka harus membuat akun baru dengan melakukan *register* dan mengisi data berupa *username*, *email*, dan *password*.
3) Jika pengirim telah terdaftar, maka dapat melakukan *login* dengan *email* dan *password* dan harus terkoneksi dengan internet.
4) Setelah login pada aplikasi chatting dan berhasil masuk. Pengirim berada pada *form Main*. Untuk membuat pesan, pengirim harus meng-*invite* pengguna lain dengan memilih tombol *add* yang ada pada bagian kanan bawah *form Main* atau memilih salah satu akun penerima yang sudah tersedia (sudah di-*invite*). Setelah diplih salah satu akun, maka pengguna masuk pada halaman *room chat*.
5) Setelah itu pengirim dapat meng-*input* isi pesan yang ingin dikirim ke penerima yang sudah di pilih atau mengirimkan lampiran berupa gambar dari galeri atau kamera yang berada pada *menu bar*.
6) Sebelum pesan dikirim, aplikasi akan melakukan proses enkripsi terlebih dahulu.
7) Pesan yang di-*input* akan dirubah menjadi *chipertext* dengan menggunakan kunci yang sudah diatur oleh sistem.
8) Setelah proses enkripsi selesai, pesan dikirim ke *server Firebase* lalu diteruskan ke akun penerima pesan.
9) Tampil notifikasi pesan sampai.



Gambar 2 : *Rich Picture* Proses Pengiriman Pesan

Setelah proses pengiriman pesan, berikut ini adalah proses penerimaan pesan :

1) Penerima mendapatkan notifikasi berupa pesan masuk dari pengirim.
2) Ketika pesan sampai pada penerima, maka sistem akan langsung mendekrip pesan yang sampai menjadi pesan awal yang dituliskan oleh pengirim.
3) Ketika pesan dibuka, maka pesan sudah dalam keadaan seperti semula.



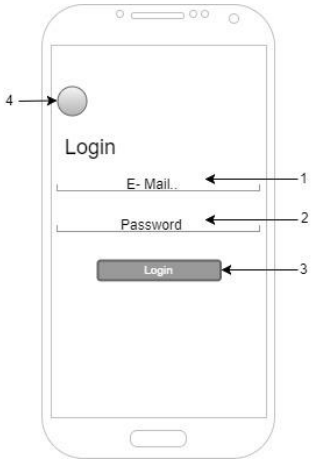
Gambar 3 : *Rich Picture* Proses Penerimaan Pesan

3.4. Rancangan Layar

Aplikasi ini akan di gambarkan rancangan layar meliputi *form login, register, main* dan *chat*. Selain itu terdapat *dropdown menu* yang berisikan *form help, about* dan *log out.* Berikut adalah rancangan layar pada masing-masing form :

a. Rancangan Layar Form Login

form login adalah *form* yang muncul ketika pengguna membuka aplikasi pada pertama kali. Pada *form login*, terdapat inputan berupa username, *e-mail*, dan *password.* Jika pengguna belum memiliki akun, maka ada pilihan *register* pada bagian bawah aplikasi untuk membuat akun baru.



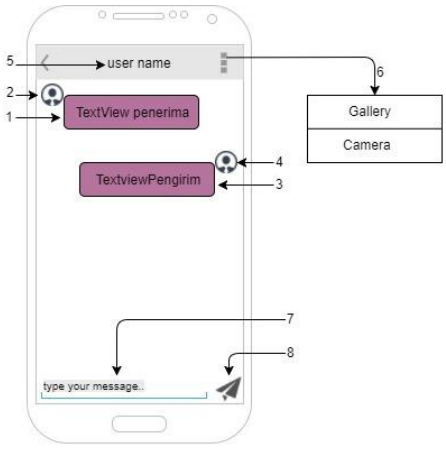
Gambar 4 : Rancangan Layar *Form Login*

Keterangan :

1) *EditText e-mail* merupakan *input* teks untuk mengisi alamat *e-mail* yang telah didaftarkan pada aplikasi ini.
2) *EditText password merupakan input teks untuk mengisi password yang sesuai dengan alamat e-mail yang telah didaftarkan.*
3) *Button login adalah tombol yang digunakan untuk login ke dalam aplikasi.*
4) *Button register adalah tombol yang digunakan untuk masuk ke dalam menu register jika belum memiliki akun pada apikasi ini.*

b. Rancangan Layar Form Chat

Seperti pada aplikasi *chatting* pada umumnya, *form chat* berisi percakapan antara sesama pengguna aplikasi dan *menu dropdown* untuk melampirkan pesan gambar baik dari galeri maupun foto. Percakapan yang dikirim dari aplikasi ini sudah dienkripsi dengan tiga metode enkripsi, yaitu *Caesar Cipher* untuk enkripsi link gambar, *Advanced Encryption Standard* 256, dan RC6 untuk enkripsi pesan teks. Dan pada saat pesan masuk, pesan yang semula terenkripsi akan didekripsikan oleh sistem agar dapat dibaca oleh penerima pesan.



Gambar 7 : Rancangan Layar *Form chat*

Keterangan :

1) *TextView* penerima merupakan tampilan percakapan yang dikirimkan penerima ke dalam ruang pecakapan. Disamping *textview* terdapat foto profil dari penerima.
2) *ImageView* penerima merupakan tampilan foto profil dari penerima.
3) *TextView* pengirim merupakan tampilan percakapan yang dikirimkan oleh pengirim ke dalam ruang percakapan. Disamping *textview* terdapat foto profil dari pengirim.
4) *ImageView* pengirim merupakan tampilan foto profil dari pengirim.
5) *TextView User name* merupakan tampilan nama lawan bicara pada saat berkomunikasi.
6) *Dropdown menu attachment* merupakan *menu* untuk mengirimkan lampiran berupa gambar dari kamera atau dari galeri.
7) *EditText input* pesan digunakan untuk menulis pesan yang ingin dikirimkan kepada pengirim.

4.HASIL DAN PEMBAHASAN

4.1. Spesifikasi Hardware dan Software

Untuk menjalankan aplikasi ini tentu tidak terlepas dari kebuatuhan *hardware* dan *software* sebagai faktor pendukungnya. Spesifikasi perangkat yang digunakan dalam pembuatan aplikasi ini antara lain yaitu :

a. Perangkat Keras (*Hardware*)

Berikut ini merupakan spesifikasi minimum *hardware* yang harus dipenuhi untuk menjalankan aplikasi *chatting* agar berjalan dengan baik :

1) Laptop *Processor Intel Core i3.*
2) *Memory* 4GB RAM.
3) *Harddisk* 500GB.
4) *Smartphone Android* dengan sistem operasi versi 5.0 *Lollipop* atau diatasnya.

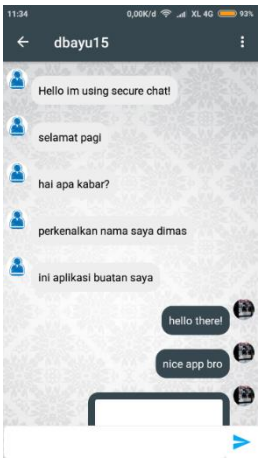
b. Perangkat Lunak (*Software*)

Dalam pembuatan aplikasi ini, perangkat lunak yang digunakan untuk implementasi aplikasi ini sebagai berikut :

1) *Operating System Windows* 10.
2) *Android Studio.*
3) *Google Chrome.*
4) *Gradle.*
5) *Microsoft Word 2013.*

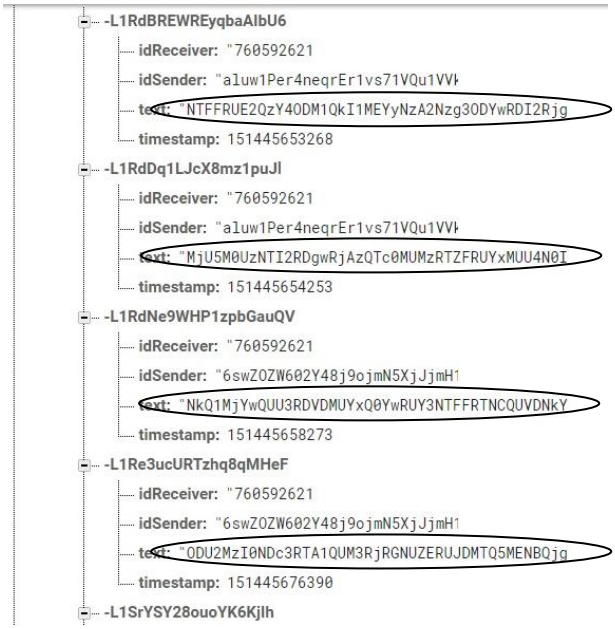
4.3. Uji Coba Program

Setelah spesifikasi perangkat keras dan perangkat lunak terpenuhi, maka aplikasi *chatting* ini dapat dilakukan uji coba. Pengujian proses enkripsi merupakan pengujian untuk mengubah *plaintext* menjadi *ciphertext*. *Key* yang digunakan untuk melakukan enkripsi ini berasal dari kombinasi antara karakter yang telah ditentukan oleh sistem ditambah dengan beberapa karakter dari *roomID* pengirim dan penerima yang ada pada *Firebase Console*.



Gambar 8 : *plainteks* yang diinputkan pada aplikasi

Setelah pengguna menginputkan pesan berupa *plainteks* pada aplikasi *chatting*, pesan akan secara otomatis terkenkripsi oleh system dan akan dikirimkan ke *server firebase* dalam bentuk pesan tereknkripsi (*Ciphertext*)



Gambar 9 : teks terenkripsi yang disimpan di *firebase*

4.2. Evaluasi Program

a. Kelebihan Program

1) Aplikasi ini tergolong pada aplikasi yang ringan karna hanya memerlukan *space memory* sebesar 5MB saat proses instalasi.
2) Aplikasi ini dapat mengirimkan pesan secara *real time*.
3) Pesan yang dikirim dapat berupa teks dan gambar.
4) Pengguna tidak perlu meng-*input key* serta melakukan penyimpanan kunci, karena sudah diatur oleh system.

5) Pada saat aplikasi sedang tidak dibuka, program dapat memberikan notifikasi pada pengguna selama koneksi tetap tersambung.
6) Setiap pesan yang dikirimkan terenkripsi oleh tiga algorima, dua algoritma untuk enkripsi teks, dan satu algoritma untuk enkripsi *link* gambar.

b. Kelemahan Program

1) Aplikasi ini tidak bisa melakukan *group chat* dan *broadcase message*.
2) Aplikasi ini tidak melakukan enkripsi pada gambar, tapi hanya meng-enkripsi *link* gambar.
3) Aplikasi ini tidak melakukan validasi terhadap *e-mail* yang digunakan untuk membuat akun.
4) Fitur pada aplikasi chatting ini masih sedikit, diantaranya tidak bisa mengirim pesan berupa suara, dokumen dan dan tidak ada setting penggunaan data.
5) Pesan gambar hanya dapat dilihat ketika gambar dalam keadaan fullscreen.
6) Aplikasi ini tidak dapat dijalankan pada system operasi android dibawah versi 5.0 Lollipop.
7) Aplikasi ini tidak bisa melakukan percakapan baik melalui voice call ataupun video call.

5. KESIMPULAN

Berdasarkan analisa dari permasalahan dan penyelesaian masalah, maka ada beberapa kesimpulan dari hasil penelitian ini:

a. Aplikasi *Chatting* sebagai aplikasi yang dibutuhkan oleh hampir sebagian besar manusia untuk berkomunikasi, membutuhkan tingkat keamanan yang tinggi karena menyangkut privasi seluruh penggunanya.
b. Dengan adanya aplikasi ini maka isi dari pesan terjaga kerahasiaannya dari pihak yang tidak berkepentingan dan yang tidak berhak untuk mengetahui apa isi dari pada pesan tersebut.
c. Enkripsi dengan menggunakan algoritma *Caesar Cipher, Advanced Encryption Standard* 256, dan RC6 menghasilkan tingkat keamanan yang cukup terjaga kerahasiaannya karena tidak hanya dienkripsi menggunakan satu enkripsi saja.

Dengan terbatasnya waktu yang diberikan untuk menyelesaikan penelitian ini, maka ada beberapa saran yang diajukan penulis guna mengembangkan dan menyempurnakan hasil penelitian ini. Saran yang dapat dikembangkan antara lain:

a. Agar ada penambahan fitur lainnya yang sesuai dengan kebutuhan pengguna dalam waktu yang akan datang.
b. Ditambahkannya kompatibilitas pada sistem operasi *android* dibawah versi 5.0 *Lollipop* agar aplikasi ini dapat berjalan di seluruh versi sistem operasi *android*.
c. Membuat autentifikasi melalui *email* untuk mengecek *email* pengguna yang dimasukkan.
d. Agar ada pengembangan lagi sehingga bisa digunakan tidak hanya pada *Smartphone Android*.

DAFTAR PUSTAKA

[1] Anon, 2013, Pengertian Komunikasi Menurut Para Ahli, Diakses 14 Desember 2015, http://www.e-jurnal.com/2013/10/pengertian-komunikasi-menurut-para-ahli.html
[2] Harahap, R.A., 2016, Implementasi Algoritma Vigenere Cipher dan Rivest Shammer Adleman (RSA) dalam Keamanan Data Teks, Jurnal INFOTEK, 1(2), hal 156-160.
[3] Pandiangan, H & Sijabat, S., 2016, Perancangan Media Pengiriman Pesan Teks dengan Penyandian Pesan Menggunakan Algoritma RC4 Berbasis Web, Jurnal Matik Penusa, 19(1), hal 63-71.
[4] Pratama, R., 2013, Penerapan Enkripsi dan Dekripsi File Menggunakan Algoritma *Advanced Encryption Standart* (AES), Journal of Research in Computer Science and Applications, 2(1), hal.13-18.
[5] Priyono, 2016, Penerapan Algoritma Caesar Cipher dan Algoritma Vigenere Cipher dalam Pengamanan Pesan Teks, Jurnal Riset Komputer (Jurikom), 3(5), hal 351-356.
[6] Bhaudhayana, G.W. & Widiarhta I.M., 2015, Implementasi Algoritma Kriptografi AES 256 dan Metode *Steganografi* LSB pada Gambar *Bitmap*, 8(2), hal 15-25.
[7] Yusfrizal, 2015, Penerapan Algoritma RC6 Untuk Perancangan Aplikasi Pengamanan SMS pada *Mobile Device* Berbasis Android, Seminar Nasional Informatika, hal 518-524.