MEMBANGUN APLIKASI CHAT ANDROID TERENKRIPSI KLIEN SERVER

Rifqi Ahmad Pratama [1], Setia Juli Irzal Ismail S.T., M.T. [2],

Mochammad Fahru Rizal S.T., M.T. [3]

[123]Prodi D3 Teknologi Komputer, Fakultas Ilmu Terapan, Universitas Telkom

[1]rifqiahmad@student.telkomuniversity.ac.id, [2]jul@tass.telkomuniversity.ac.id, [3]mfrizal@tass.telkomuniversity.ac.id

Abstrak- Keamanan data adalah hal-hal yang perlu diperhatikan dalam dunia komunikasi dan kriptografi adalah solusinya. Algoritma RSA merupakan algoritma kriptografi asimetris yang masih memiliki ketahanan dari ancaman sampai saat ini. Sedangkan algoritma AES merupakan algoritma kriptografi simetris yang masih digunakan sebagai standar kriptografi pada saat ini. Kunci RSA dengan Panjang 1024 bit sampai 4096 bit. Sedangkan kunci AES yang digunakan 128 bit.

Dalam tugas akhir ini, akan diterapkan algoritma RSA dan algoritma AES pada aplikasi *chatting*, dimana dengan menerapkan mekanisme kriptografi akan menutup celah keamanan yang ada pada aplikasi pesan sederhana tersebut. Dengan penerapan aplikasi dapat dihasilkan mekanisme komunikasi yang aman untuk dipakai dalam komunikasi penting agar tidak disadap pihak yang tidak bertanggung jawab. Hasil penerapan dari 2 algoritma menambah tingkat keamanan dari data yang dikirimkan. Kata kunci: RSA, AES, *Cryptography, Chat Encrypted.*

Abstract*- Data security is the important thing to be noticed in the world of communication and cryptography is the solution. The RSA algorithm is algorithm of the asymmetrical cryptographic, it still has endurance from threats until now. While the AES algorithm is a symmetrical cryptographic algorithm that is still used as a cryptographic standard at the moment. The length of RSA keys used are from 1024 bits to 4096 bits. While the AES key used is 128 bits.*

In this final task, the RSA algorithm will be applied and the AES algorithm in the chat application, where by implementing a cryptographic mechanism, will close the security gaps that are present in the simple messaging application. With the implementation of aplication, it can be generated a safe communication mechanism to be used in important communication in order not to be intercepted by an irresponsible party. The result of applying from 2 algorithms increases the security level of the transmitted data. Keywords : RSA, AES, Cryptography, Chat Encrypted.

1. Pendahuluan

1.1 Latar Belakang

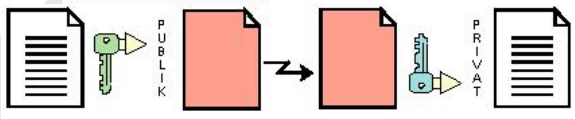
Situs jejaring sosial memang menyedot banyak minat masyarakat dari berbagai kalangan, mulai dari anak–anak hingga orang dewasa. Aplikasi

chatting adalah merupakan aplikasi yang paling banyak digunakannya. Pada saat ini sudah terdapat banyak aplikasi chatting yang tersedia dalam berbagai OS. Menurut Tjahyana, *smartphone* adalah telepon genggam yang mempunyai kemampuan dengan penggunaan dan fungsi yang menyerupai komputer. Sehingga
memberikan kemudahan dalam melakukan segala aktivitas pengguna. Seperti halnya melakukan kegiatan yang membutuhkan jaringan internet, contohnya mengirim e-mail, berinteraksi dalam jejaring sosial, sampai sekedar melakukan
chatting kepada sesama pengguna. 50% para pengguna internet lebih menggunakan *smartphone* mereka, dan 45% menggunakan laptop, lalu 4% menggunakan tablet dan sisanya menggunakan perangkat lainnya.[1]

Namun berhati-hatilah dalam mendownload dan install aplikasi chatting. Karena telah ditemukan sebuah aplikasi chatting yang berfungsi sebagai mata-mata bagi para penggunanya. Menurut salah satu pejabat AS telah ditemukan sebuah aplikasi yang dirancang untuk menyadap dan memata-matai penggunanya. Aplikasi chatting tersebut bernama ToTok, laporan ini pertama kali dilaporkan New York Times seperti dilansir CNBC Indonesia dari Engadget. Aplikasi ToTok ini digunakan oleh Uni Emirat Arab (UEA) sebagai alat pengawasan untuk mengikuti percakapan pengguna, melacak lokasi target, menentukan koneksi sosial dan melacak media. Sebagian besar pengguna aplikasi ini berada di UEA tetapi sedang berkembang di negara lain dan permintaannya sedang naik di AS.[2]

Kriptografi merupakan pilihan tepat untuk pengamanan pesan. Algoritma kriptografi dibagi menjadi dua bagian, yaitu: algoritma simetris dan algoritma asimetris. Algoritma simetris hanya memakai satu kunci yang sama dalam proses enkripsi dan dekripsi. Algoritma asimetris memiliki 1 kunci *public* untuk enkripsi dan hanya memiliki 1 kunci *private* untuk melakukan dekripsi pesan. Jadi tingkat keamanan lebih tinggi. Oleh karena itu untuk memperkuat keamanan maka kita harus menggabungkan algoritma simetris dan asimetris.

Dengan beberapa algoritma enkripsi tersebut, pesan dapat lebih aman dari pada hanya menggunakan 1 algoritma kriptografi. Apabila hanya menggunakan algoritma AES, pesan dapat lebih mudah disadap dari pada menggunakan beberapa algoritma kriptografi. Berikut ini adalah gambaran umum dari kriptografi simetris dan kriptografi asimetris.



Gambar 1 Alur Umum Kriptografi Asimetris

Keterangan alur kriptografi asimetris
- Dokumen atau *plain text* (Kertas putih) dienkripsi dengan kunci *public*
dihasilkan *ciphertext* (Kertas merah)
- *ciphertext* dikirimkan
- *ciphertext* diterima dan didekripsi menggunakan kunci *private* maka dihasilkan
dokumen asli.[3]

Gambar 2 Alur Umum Kriptografi Simetris

Keterangan alur kriptografi simetris

- Dokumen atau *plain text* (Pesan) dienkripsi dengan kunci yang dimiliki dihasilkan *ciphertext* (Pesan Terkunci)
- *ciphertext* diterima dan didekripsi menggunakan kunci yang sama dengan kunci enkripsi pesan maka dihasilkan dokumen asli.

Proyek akhir ini membangun aplikasi chat terenkripsi klien *server* menggunakan AES untuk mengenkripsi pesan dan menggunakan RSA untuk mengenkripsi kunci AES. Aplikasi yang dibuat adalah 1 aplikasi *client* dan 1 *server* yang telah dilengkapi dengan fitur enkripsi pada pesan yang akan dikirimkan. Aplikasi ini hanya bisa di jalankan pada android.

1.2 Rumusan Masalah

Berdasarkan uraian di atas, maka dapat permasalahan yang ada, dapat dirumuskan menjadi :

1. Bagaimana cara membuat aplikasi chat android yang ter-enkripsi dengan algoritma enkripsi AES?
2. Bagaimana cara mendapatkan kunci *public* dan *private* RSA pada aplikasi chatting?
3. Bagaimana cara mengirim kunci AES ke lawan chat supaya kunci tidak dapat dibaca oleh *server* sebagai perantara pengiriman kunci?

1.3 Tujuan

Tujuan dalam pembuatan aplikasi ini adalah:

1. Dengan membangkitkan kunci AES pada saat akan melakukan enkripsi dan menggunakan kunci yang sama untuk melakukan dekripsi.
2. Menggunakan enkripsi RSA dengan kunci *public* sebagai enkripsi dan menggunakan kunci *private* sebagai dekripsi.
3. Dengan menggunakan kunci *public* yang di terima dari lawan bicara untuk enkripsi kunci AES supaya tidak dapat di baca.

1.4 Batasan Masalah

Batasan masalah aplikasi ini adalah:

1. Aplikasi dibuat dalam platform Android,
2. Pembuatan aplikasi menggunakan Android Studio,
3. Menggunakan bahasa java pada android studio sebagai fungsi pembuatan aplikasi,
4. Mengunci dan membuka pesan dengan metode AES
5. Mengunci dan membuka kunci AES menggunakan metode RSA
6. Proses Enkripsi khusus pesan berupa teks yang berada pada chat jalur pribadi, tidak untuk gambar, suara dan yang lainnya.

1.5 Definisi Operasional

1.5.1 Android

Android merupakan sistem operasi berbasis linux yang dirancang untuk *smartphone* atau perangkat bergerak dengan layar sentuh. Android awalnya dikembangkan oleh Android *Inc*, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google *Inc*.[4]

1.5.2 Klien Server

Client dan *Server* adalah dua buah aplikasi yang berjalan dan saling berinteraksi satu sama lain, sehingga aplikasi *Client* dan *Server* bisa saja berada bersama dalam satu buah komputer secara sekaligus. Aplikasi *Server* cenderung bersifat pasif dan menunggu datangnya permintaan (*request*) dari satu atau lebih aplikasi client, kemudian memberi jawaban (*response*) dari setiap *request* tersebut secara simultan. Aplikasi *Client* cenderung bersifat aktif untuk meminta atau mengirim *request* ke aplikasi Server.

2. Tinjauan Pustaka

2.1 Kriptografi

Kriptografi adalah sebuah ilmu dan seni untuk menjaga keamanan dan kerahasiaan suatu pesan, yaitu dengan mengamankan suatu pesan sehingga pihak yang tidak bertanggung jawab tidak dapat membaca pesan tersebut.Kriptografi sangat berkembang sedemikian rupa sehingga tidak lagi hanya sebatas melakukan kegiatan enkripsi pada sebuah pesan ataupun data, tetapi juga memberikan aspek keamanan yang lainnya. Salah satu dasar untuk menjamin sebuah keamanan pada suatu data adalah dengan menerapkan algoritma kriptografi dan kunci. Algoritma kriptografi merupakan sebuah fungsi matematika yang digunakan untuk enkripsi dan dekripsi suatu pesan. Sedangkan kunci adalah sebuah parameter yang sangat digunakan untuk perubahan bentuk enkripsi dan dekripsi. Kunci bersifat rahasia (*private*) sedangkan algoritma kriptografi bersifat tidak rahasia (publik). Proses penyandian dari *plaintext* menjadi *ciphertext* disebut enkripsi. Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi.

Gambar 3 Kriptografi

Gambar di atas merupakan alur proses enkripsi dan dekripsi. Dimana *plaintext* melewati proses enkripsi untuk menghasilkan *ciphertext*. Dan kemudian, *ciphertext* melewati proses dekripsi untuk mengembalikannya menjadi *plaintext* kembali.[5]

2.1.1 Kriptografi Asimetris

Algoritma asimetris sangat sering juga disebut dengan algoritma kunci *public*, dimana kunci yang digunakan untuk melakukan enkripsi dan deskripsinya berbeda dari kunci enkripsi. Kunci untuk enkripsi hanya dibuat untuk diketahui oleh umum (kunci publik), tapi untuk proses dekripsinya hanya dapat dilakukan oleh yang bertanggung jawab yang memiliki kunci rahasia untuk melakukan kegiatan dekripsinya (privat key).

2.1.2 Kriptografi RSA

RSA adalah Algoritma kunci-publik yang sangat terkenal dan sangat sering digunakan aplikasinya. RSA Ditemukan oleh tiga peneliti dari *MIT* (*Massachussets Institute of Technology*), mereka adalah Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976. Keamanan algoritma RSA terdapat pada sangat sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.[3]

2.1.3 Kriptografi Simetris

Algoritma simetrik dapat juga disebut sebagai algoritma konvensional, yang mana kunci dekripsi sama dengan kunci enkripsinya, begitu juga sebaliknya. Pada algoritma simetrik, kunci enkripsi dan kunci dekripsinya sama. Keamanan pada algoritma tersebut terletak pada kuncinya, jika kunci disebarkan atau dibocorkan maka setiap orang dapat mengenkrip dan mendekrip pesan, jadi kunci harus benar-benar sangat rahasia dan aman.[6]

2.1.4 Kriptografi AES

Advanced Encryption Standard (AES) mulai digunakan 1997. NIST mengumumkan bahwa jika AES digunakan sebagai pengganti enkripsi Data *Encryption Standard* (DES) yang telah lama dan sangat kurang pada aman (Bodic, 2005). AES telah menjadi *block ciphertext* dimana dapat memproses blok 128 bit dari pada input yang merupakan *plaintext* dalam suatu waktu. AES juga sangat pendukung pengaturan kunci 128, 192, dan 256 bit serta sangat lebih efisien daripada DES (Hermawanm 2009).[7]

2.2 Android

2.2.1 Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android.[8] Untuk saat ini android studio sudah mengeluarkan dua jenis bahasa untuk melakukan fungsi pemograman, yaitu : Bahasa Java dan Bahasa Kotlin.

Bahasa Java

Java Development Kit atau biasa disingkat dengan JDK adalah Perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode java ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*).Bahasa java sering digunakan untuk pembuatan aplikasi, Seperti pembuatan aplikasi android yang menggunakan bahasa java sebagai fungsinya.

2.3 Database

Database adalah tempat penyimpanan yang sangat besar dimana terdapat kumpulan berbagai jenis data yang tidak hanya berisi data operasional tetapi juga deskripsi data. Seperti yang disampaikan oleh Connolly dan Begg (2010, p.65), bahwa *database* adalah kumpulan data yang saling terhubung secara logis dan deskripsi dari data tersebut, dirancang untuk menemukan informasi yang dibutuhkan oleh sebuah organisasi. Dalam merancang *database,* salah satu hal yang perlu diperhatikan adalah efisiensi. Banyaknya data yang redudansi dapat mengurangai efisiensi pada *database* sehingga perlu dilakukan normalisasi.

Database ini digunakan tidak hanya oleh satu orang maupun satu departemen, *database* dapat digunakan oleh seluruh departemen dalam perusahaan. *Database* ini akan menjadi sumber data yang digunakan secara bersama dalm perusahaan. Hal ini kembali ditegaskan oleh Connolly dan Begg (2010, p.65), *database*

tidak lagi dimiliki oleh satu departemen tetapi sumber perusahaan yang saling berbagi. Dengan hanya *database* saja tidak cukup, diperlukan *Database Management System* (DBMS) untuk dapat menggunakan *database*.[9]

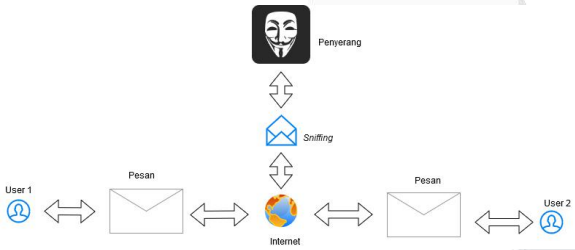
2.3.1 Firebase

Firebase adalah API yang disediakan google untuk penyimpanan dan menyamakan data masuk ke dalam aplikasi Android, iOS, atau web. *Realtimedatabase* adalah salah satu fasilitas yang menyimpan data ke *database* dan mengambil data darinya dengan sangat cepat tetapi firebase bukan hanya *realtimedatabase*, jauh lebih dari itu. Firebase memiliki banyak fitur seperti *authentication*, *database*, *storage*, *hosting*, pemberitahuan dan lain-lain.[10]

3. Analisis dan Perancangan

3.1 Analisis

3.1.1 Gambaran Sistem Saat ini

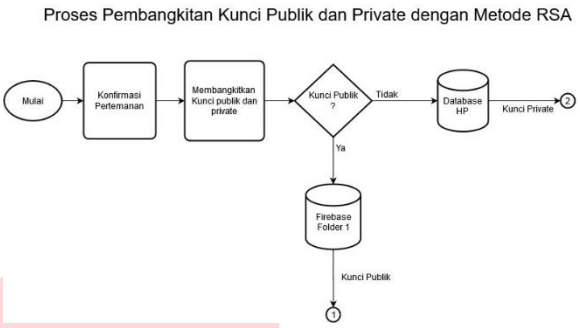


Gambar 4 Gambaran saat ini

Pada saat *user* 1 berkomunikasi dengan *user* 2, besar kemungkinan terjadi penyerangan terhadap pesan yang dikirim antar kedua user tersebut, yang dilakukan oleh orang yang tidak bertanggungjawab. Kriptografi adalah salah satu upaya untuk mengamankannya pesan tersebut.

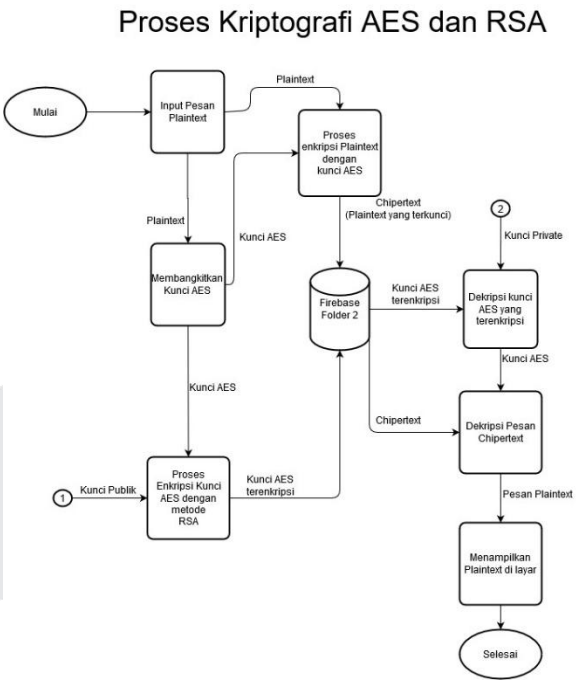
3.2 Perancangan Sistem

3.2.1 Gambaran Sistem Usulan



Gambar 5 Rancangan Sistem ke 1

Penjelasannya ada pada sub bab 3.2.2 cara kerja.



Gambar 6 Rancangan Sistem ke 2

3.2.2 Cara Kerja

Cara kerja sistem berdasarkan beberapa tahapan mulai dari RSA sampai AES diantaranya :

1. Membangkitkan Kunci Publik dan *Private* RSA.

2. Saling mengirimkan kunci *public* RSA ke lawan chatting melewati firebase dan menyimpan kunci *private* pada *storage*

3. Mengambil kunci *private* dari lawan bicara di firebase
2. Membuat pesan yang ingin dikirimkan
3. Membangkitkan kunci AES
4. Enkripsi Pesan metode AES menggunakan kunci AES yang sudah dibangkitkan
5. Enkripsi kunci AES dengan metode RSA menggunakan kunci *public* yang didapatkan dari lawan bicara
6. Mengirim pesan yang sudah di enkripsi dan kunci AES yang sudah di enkripsi ke penerima pesan melalui firebase
7. Penerima pesan mengambil pesan dan kunci aes yang sudah di enkripsi di firebase
8. Penerima mendapatkan pesan dan kunci AES yang terenkripsi
9. Proses dekripsi kunci AES dengan metode RSA menggunakan kunci *private* penerima pesan
10. Kunci AES dari lawan bicara sudah didapatkan
11. Dekripsi pesan dengan motode AES menggunakan kunci AES yang sudah didapatkan
12. Mendapatkan pesan yang dikirimkan pengirim pesan.

3.2.3 Perangkat keras dan perangkat lunak

Adapun spesifikasi perangkat keras dan lunak yang digunakan adalah:

3.2.3.1 Perangkat Keras

Spesifikasi dari perangkat keras yang digunakan antara lain:

Tabel 1 Perangkat Keras

Perangkat	Spesifikasi
Laptop	Asus Intel Core i7 4720M CPU 2,6 GHz; RAM 8 GB DDR 3; Harddisk 1 TB
HP	Oppo a37 Kecepatan CPU 1.2 GHz *Storage* 16GB RAM 2GB

3.2.3.2 Perangkat Lunak

Spesifikasi dari perangkat lunak yang digunakan adalah:

Tabel 2 Perangkat Lunak

Perangkat	Versi Perangkat lunak
Android Studio	Android Studio 3.5.2
Nox	Nox 3.5.0.3
Wireshark	Wireshark 3.0.2
HttpCanary	HttpCanary non premium

4. Implementasi dan Pengujian

Proyek Akhir ini dilaksanakan dengan 2 orang, untuk pembutan aplikasi *chatting* dibuat dengan Ray Samudra Bagas. Adapun penulis bertugas untuk membuat *Cryptografi* pada aplikasi *chatting*.

4.1 Langkah Pengerjaan

Langkah – langkah pengerjaan aplikasi enkripsi di bagi menjadi beberapa tahapan,
diantaranya:

1. Mencari Informasi
2. Mencari program algoritma kriptografi yang diperlukan
3. Memasukan program kriptografi

4.1.1 Mencari Informasi

Pada tahap ini mencari informasi tentang jenis kriptografi yang dibutuhkan untuk membuat aplikasi chatting enkripsi. Dengan memulai mempelajari fungsi dari kriptografi yang berada di aplikasi chatting yang sudah ada, aplikasi Whatsapp, Telegram dan Signal. Setelah mendapatkan informasi kriptografi yang berada di aplikasi yang sudah dipakai masyarakat pada umumnya. Sehingga mulai mempelajari fungsi kriptografi yang ingin dibangun mulai algoritma simetris dan algoritma asimetris, karena dasar dari algoritma kriptografi adalah algoritma simetris dan algoritma asimetris. Setelah mempelajari 2 jenis algoritma kriptografi, maka menentukan enkripsi yang ingin digunakan dalam aplikasi chatting yang dibangun. Pada aplikasi ini menggunakan algoritma AES yang sebagai algoritma simetris dan algoritma RSA yang sebagai algoritma asimetris. Dengan menggunakan 2 jenis algoritma kriptografi, menjadikan pesan yang ingin di amankan lebih standar enkripsi dari pada hanya menggunakan 1 jenis kriptografi.

4.1.2 Mencari program kriptografi yang diperlukan

Setelah menentukan kriptografi yang ingin digunakan, maka mencari program tersebut. Jenis kriptografi yang digunakan adalah algoritma RSA dan algoritma AES.

4.1.3 Memasukan program kriptografi

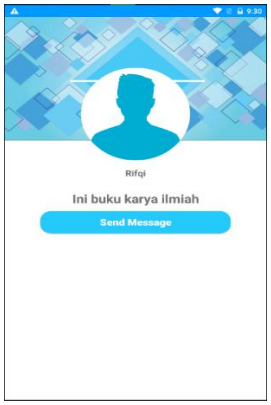
Setelah program kriptografi yang dibutuhkan sudah di dapatkan maka program di masukan ke aplikasi chatting yang sudah dibuat oleh Ray Bagas. Langkah yang dilakukan adalah:

1. Membangkitkan kunci *public* dan *private* RSA dengan menggunakan program RSA.
2. Kunci *public* akan dikirimkan melalui firebase, yang pengiriman kunci ini di bagi menjadi 2 bagian :
    1. Mengirimkan kunci *public* sebagai *sender*, dengan program bagian *SendChatRequest.*
    2. Mengirimkan kunci *public* sebagai *receiver*, dengan program sebagai *AcceptChatRequest*.
3. Ketika kunci *public* dikirim ke firebase kunci *private* disimpan di *storage.*
4. Mengenkripsi pesan yang di ketika dengan memanggil program kunci AES yang sudah di dapatkan.
5. Mengambil kunci *public* dari lawan bicara di firebase.

4.2 Tampilan Interface aplikasi

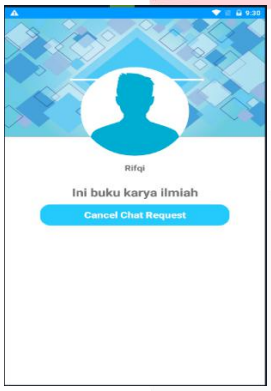
Kegiatan implementasi dan pengujian di awali dengan menampilkan interface aplikasi *chatting*, pada percobaan kali ini menggunakan 2 pengguna dalam pemakaian aplikasi yaitu :

4.2.1 User 1



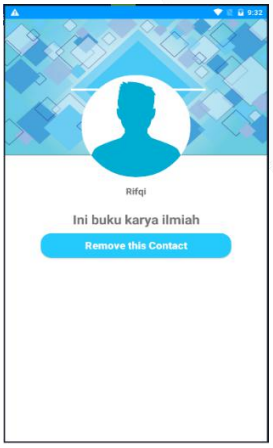
Gambar 7 User 1 Awal

Masuk ke halaman teman yang ingin di buat *request* chat, dengan mengirim permohonan untuk melakukan chatting.



Gambar 8 User 1 Menunggu Konfirmasi

Menunggu teman yang ingin di ajak bicara menyetujui chatting.

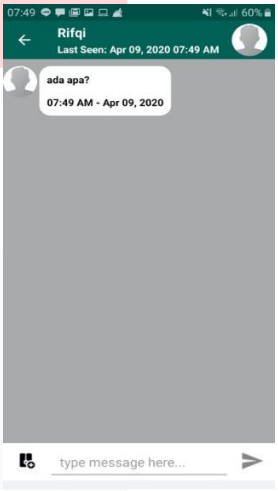


Gambar 9 User 1 Selesai

Sudah di terima untuk chatting dan langsung masuk ke halaman chatting.

Gambar 10 User 1 Chatting Awal

Halaman chatting dan tampilan masih kosong



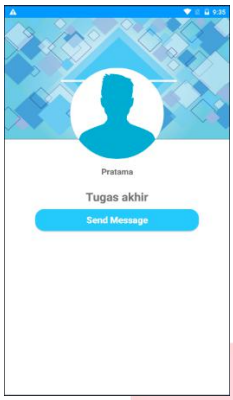
Gambar 11 User 1 Menerima Pesan

User ke 1 dapat menerima pesan dari user ke 2

Gambar 12 User 1 Mengirim Pesan

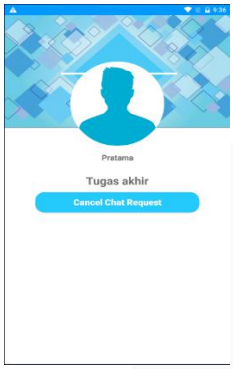
User ke 1 mengirim pesan ke user ke 2

4.2.2 User 2



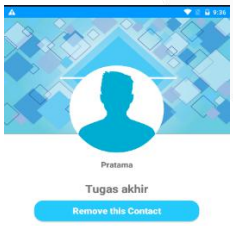
Gambar 13 User 2 Awal

Menunggu kiriman dari user 1 untuk melakukan konfirmasi chatting.



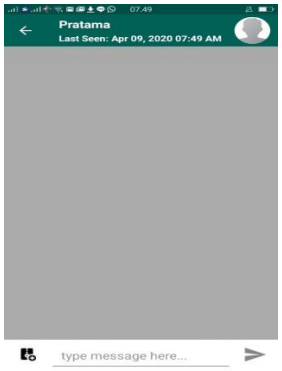
Gambar 14 User 2 Request

Melakukan konfirmasi chatting dengan tekan "*Accept Chat Request*"



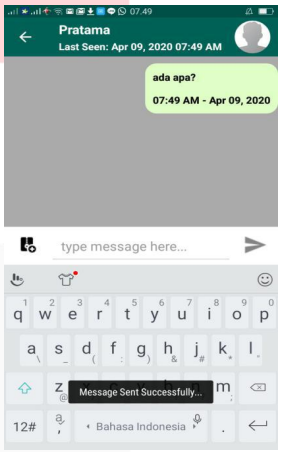
Gambar 15 User 2 Setelah Konfirmasi Chatting

Bagian setelah melakukan konfirmasi ingin melakukan chatting dan langsung masuk ke halaman chatting untuk melakukan chatting



Gambar 16 User 2 Chatting Awal User 2

Bagian awal chatting user 2 ke pada user ke 1 pada kondisi halaman masih kosong



Gambar 17 User 2 Mengirim Pesan

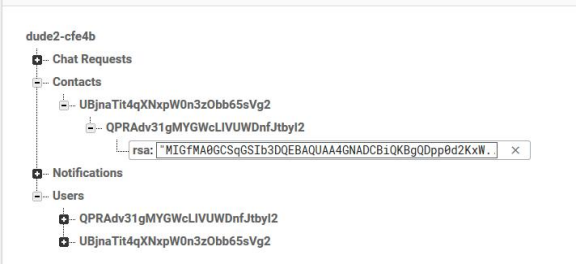
User ke 2 mengirim pesan dan menampilkan di *layout* pesan yang dikirimkan

Gambar 18 User 2 Menerima Pesan

User ke 2 menerima pesan dari user ke 1

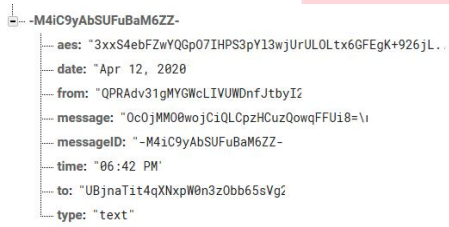
4.3 Tampilan proses kriptografi

4.3.1 Tampilan pada firebase



Gambar 19 Kunci public RSA

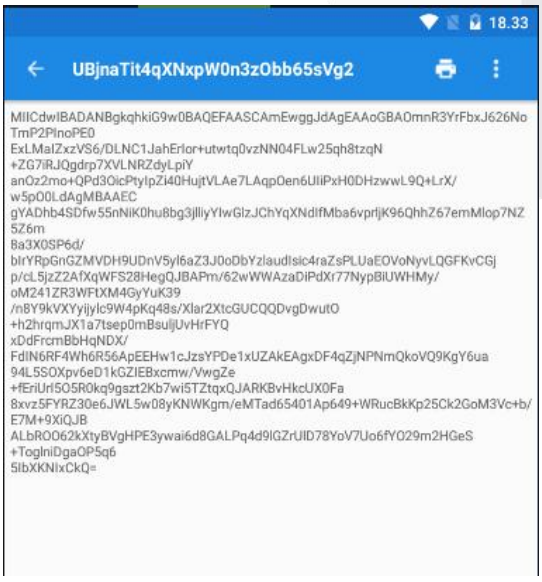
Menampilkan kunci RSA public yang berada di firebase setelah melakukan pembangkitan kunci RSA dan di kirimkan ke firebase.



Gambar 20 Kunci AES yang terkunci dan pesan yang terkunci

Menampilkan kunci AES yang terenkripsi dan pesan yang terenkripsi yang berada di firebase. Pada bagian ini adalah data yang ingin dikirimkan ke lawan bicara pada saat melakukan *chatting*.

4.3.2 Tampilan pada pada smartphone



Gambar 21 Kunci private RSA

Menampilkan kunci *private* RSA yang tersimpan di *storage* pada *smartphone.*

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan dari laporan ini, dibuat menjadi 3 poin yaitu :

1. Mengunci pesan dengan menggunakan kunci AES dan lawan bicara melakukan dekripsi pesan dengan kunci yang sama dari lawan bicara.
2. Fungsi dari kunci RSA *public* dan *private* adalah mengamankan kunci AES yang dikirimkan ke lawan bicara. Cara kerja RSA adalah mengunci kunci AES yang di inginkan dengan kunci *public* yang di terima dari lawan bicara dan dekripsi kunci AES yang terenkripsi dengan kunci *private* yang dimiliki.
3. Pada saat kunci AES yang ingin dikirimkan ke lawan bicara di enkrip dengan kunci public RSA yang di dapatkan dari lawan bicara. Pada saat itu kunci AES berubah menjadi *ciphertext*. Jika kunci AES yang terenkripsi tersebut dikirimkan ke lawan bicara melalui *server* firebase, maka *server* firebase tidak dapat melihat isi kunci AES yang asli.

5.2 Saran

Saran untuk pengembangan dengan menambahkan sertifikat pada enkripsi supaya *hacker* tidak dapat berpura-pura sebagai pihak yang ingin di ajak bicara dengan mengirimkan kunci *public* ke pihak yang diserang.

6. Daftar Pustaka

[1] E. Febryanta, “Pengaruh intensitas

penggunaan aplikasi chatting messenger terhadap proses penetrasi sosial," *e-Proceeding Manag.*, vol. 2, no. 2, pp. 1421–1427, 2015.

[2] "Hati-hati! Aplikasi Chatting Ini Disebut Sadap Penggunanya." [Online]. Available: https://www.cnbcindonesia.com/tech/20191223130107-37-125204/hati-hati-aplikasi-chatting-ini-disebut-sadap-penggunanya. [Accessed: 16-Apr-2020].

[3] A. Khairan, "Analisis dan Implementasi Kriptografi RSA pada Aplikasi Chatting Client-Server Based." Universitas Telkom, 2014.

[4] S. Gumuda, "Dynamics of the process of changes in concentration of methane in the air of ventilation currents in mines.," vol. 2, no. 2, pp. 13–21, 1978.

[5] A. K. ILAHI, "PROTOTIPE SISTEM ENKRIPSI DAN DEKRIPSI BERBASIS FPGA MENGGUNAKAN ALGORITMA STREAM CIPHER GRAIN-128." Universitas Telkom, S1 Sistem Komputer, 2018.

[6] Dafid, "Kriptografi Kunci Simetris Dengan Menggunakan Algoritma Crypton," *J. Ilm. STMIK*, vol. 2, pp. 20–27, 2006.

[7] M. Phone, "Implementasi Algoritma ECDH dan AES untuk Pengamanan Pesan SMS pada Telepon Seluler," *Bimipa*, vol. 24, no. 1, pp. 39–50, 2014.

[8] "jurnal Android." [Online]. Available: https://www.google.com/search?safe=strict&client=firefox-b-d&ei=y-rLXMD_GZSm9QP4k5CADQ&q=jurnal+pemrograman+android&oq=jurnal+pemograman+androi&gs_l=psy-ab.1.0.0i13j0i22i30l2.1050551.1051970..1054592...0.0..0.212.847.1j4j1.....0....1..gws-wiz.....0i71j. [Accessed: 03-May-2019].

[9] M. D. Setiyo, "Materi Basis Data - Pengertian Data Base.pdf." .

[10] G. R. Paraya and R. Tanone, "Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, no. 3, pp. 397–406, 2018, doi: 10.28932/jutisi.v4i3.870.