APLIKASI KRIPTOGRAFI DENGAN METODE RSA DAN AES UNTUK PENGAMANAN CHATTING BERBASIS ANDROID

Firman Mujahidin[1)], Subandi[2)]
[1]Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : firmanmujahidin@gmail.com[1)], Subandionline@gmail.com[2)]

ABSTRAK

PT. Juke Solusi Teknologi adalah sebuah perusahaan yang bergerak di bidang jasa Aplikasi web maupun Mobile apps development. Berfokus membuat sebuah aplikasi web maupun mobile apps untuk client. Banyaknya permintaan dari client untuk pembuatan aplikasi maka pemilik perusahaan berkeinginan membuat aplikasi Chatting yang nantinya dapat di gunakan oleh client. Selain itu pemilik perusahaan menginginkan fitur keamanan guna meningkatkan keamanan pesan dalam pertukaran data / informasi dalam chatting. Aplikasi chatting yang dibangun dapat berjalan diatas sistem operasi android pada smartphone. Teknik yang digunakan yaitu teknik kriptografi dengan menggunakan algoritma RSA dan AES untuk mengenkripsi pesan tersebut. Pesan akan di enkripsi dengan metode RSA terlebih dahulu setelah itu dienkripsi kembali dengan metode AES. Hasil enkripsi berupa cipherteks yang tidak bisa di baca. Untuk mengembalikan kebentuk semula maka pesan harus melewati proses dekripsi dengan mengunakan metode AES terlebih dahulu dan kembali dienkripsi dengan metode RSA, jika pesan berhasil di dekripsi maka pesan plainteks kembali semula sebelum dienkripsi. Hasil dari implementasi algoritma RSA dan AES pada perusahaan dapat membantu proses keamanan pesan pada aplikasi chatting untuk client.

Kata kunci : *RSA, AES, enkripsi, dekripsi*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini sangat pesat sehingga kerahasiaan data adalah aspek penting dalam suatu informasi. Demi menjaga kemanan sebuah data, maka diperlukan sebuah kunci untuk menjaga data tersebut dari pencurian oleh pihak-pihak tertentu yang tidak bertanggung jawab. Metode yang di gunakan adalah kriptografi.

PT.Juke Solusi Teknologi adalah sebuah perusahaan yang bergerak di bidang jasa Aplikasi web maupun *Mobile apps development*. Berfokus membuat sebuah aplikasi web maupun *mobile apps* untuk *client*. Banyaknya permintaan dari *client* untuk pembuatan aplikasi maka pemilik perusahaan berkeinginan membuat aplikasi *Chatting* yang nantinya dapat di gunakan oleh *client*. Selain itu pemilik perusahaan menginginkan fitur keamanan guna meningkatkan keamanan pesan dalam pertukaran data / informasi dalam *chatting*.

Untuk menjaga keamanan data dan informasi, data dan informasi tersebut perlu di enkripsi yaitu proses penyamaran dari *plaintext* (teks asli) ke *ciphertext* (hasil penyamaran) dan dekripsi yaitu proses pembalikan dari *ciphertext* ke *plaintext*. Baik proses enkripsi dan dekripsi melibatkan satu atau beberapa algoritma.

Berdasarkan pernyataan di atas, perlu ada suatu aplikasi pengaman *chat*, yang dapat meminimalisir dampak kebocoran informasi rahasia perusahaan. Maka penulis mencoba membuat aplikasi *chatting* menggunakan algoritma RSA dan AES berbasis android. Aplikasi ini menggunakan *web service* sebagai penjembatan antara *engine* dengan *database*. Aplikasi ini juga menggunakan metode *Firbase Clound messangging* untuk membuat aplikasi secara *realtime*, dimana ketika ada perubahan data maka saat itu juga di aplikasi ada perubahan atau setidaknya muncul notifikasi.

2. LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Menurut Bruce Scheiner dalam bukunya "Applied Cryptography", kriptografi adalah ilmu pengetahuan dan seni menjaga message-message agar tetap aman (secure) [1].

2.2 Algoritma RSA

RSA adalah sebuah algoritma berdasarkan skema public-key cryptography. Diberi nama RSA sebagai inisial para penemunya: Ron Rivest, Adi Shamir, dan Leonard Adleman. RSA dibuat di MIT pada tahun 1977 dan dipatenkan oleh MIT pada tahun 1983. Setelah bulan September tahun 2000, paten tersebut berakhir, sehingga saat ini semua orang dapat menggunakannya dengan bebas. Lebih jauh, RSA adalah algoritma yang mudah untuk diimplementasikan dan dimengerti. Algoritma RSA adalah sebuah aplikasi dari sekian banyak teori seperti extended euclid algorithm, euler's function sampai fermat theorem. Artikel ini menjelaskan algoritma RSA dari dasar. Saya akan menggunakan nilai-nilai yang kecil untuk menjelaskan bagaimana cara kerja algoritma RSA.[2]
Dalam perumusan Algoritma RSA di bagi menjadi 3 proses yaitu :

2.2.1 Proses Pembangkitkan Pasangan Kunci

Proses pembangkitan kunci akan dilakukan beberapa tahap seperti [3] :

a. Pilih dua buah bilangan prima sembarang, p dan q.
b. Hitung n = p × q (adalah p ¹ q, karena jika p = q maka n = p2 dan p dapat dihasilkan dengan mengambil akar pangkat dua dari n).
c. Hitung $\phi(n) = (p - 1)(q - 1)$
d. Pilih kunci publik, e, yang relatif prima terhadap $\phi(n)$.
e. Bangkitkan kunci privat dengan menggunakan rumus yaitu $e \times d \equiv 1 \pmod{\phi(n)}$.

Perhatikan bahwa $e \times d \equiv 1 \pmod{\phi(n)}$ ekivalen dengan $e \times d = 1 + k\phi(n)$, sehingga d dapat dihitung dengan $d = 1 + k\,\phi\,(n) : e$
Maka K merupakan bilang bulat yang dapat meberikan bilangan bulat d.
Setalah dilakukan penghitungan dari algoritma diatas maka akan dihasilkan sebagai berikut:
- Pasangan (e, n) merupakan Kunci Publik
- Pasangan (d, n) merupakan Kunci privat

Note: yang tidak bersifat rahasia yaitu n, namun ia dapat diperlukan pada saat perhitungan baik enkripsi atau dekripsi

2.2.2 Proses Enkripsi

Pada proses enkripsi akan dilakukan beberapa tahap seperti :

a. Langkah pertama yaitu kunci publik penerima pesan, e, dan modulus n dipisahkan terlebih dahulu.
b. Kemudian merubah plaintext m menjadi blok-blok m1, m2, …, sedemikian sehingga setiap.
c. blok merepresentasikan nilai di dalam selang [0, n – 1].
d. Setiap blok mi dienkripsi menjadi blok ci dengan rumus ci = mi e mod n

2.2.3 Proses Dekripsi

Rumus mi=ci d mod n digunakan pada saat proses dekripsi yang dilakukan pada saat setiap blok *cipherteks* didekripsi kembali menjadi blok mi.

2.3 Algoritma AES

AES merupakan sistem penyandian blok yang bersifat non-Feistel karena AES menggunakan komponen yang selalu memiliki invers dengan panjang blok 128 bit. Kunci AES menggunakan proses yang berulang yang disebut dengan ronde. Proses di dalam AES merupakan transformasi terhadap state. Sebuah teks asli dalam blok (128 bit) terlebih dahulu diorganisir sebagai state. Enkripsi AES adalah transformasi terhadap state secara berulang dalam beberapa ronde.
Proses enkripsi sendiri melewati beberapa proses seperti AddRoundKey, SubBytes, ShiftRows, Mix Column[4].

Dalam perumusan Algoritma AES di bagi menjadi 2 proses yaitu :

2.3.1 Proses Enkripsi

Proses enkripsi akan di lakukan beberapa tahap seperti [5] :

a. SubBytes, sebagai transformasi subtutusi.
b. ShiftRows, sebagai transformasi permutasi.
c. MixColumns, sebagai transformasi pengacakan.
d. AddRoundKey, sebagai transformasi penambahan kunci.

2.3.2 Proses Dekripsi

Proses dekripsi akan di lakukan beberapa tahap seperti

a. Transformasi byte yang berkebalikan dengan transformasi ShiftRows. Pada transformasi InvShiftRows, dilakukan pergeseran bit ke kanan sedangkan pada ShiftRows dilakukan pergeseran bit ke kiri hal ini disebut dengan InvShiftRows.
b. Transformasi byte yang berkebalikan dengan transformasi SubBytes. Pada InvSubBytes, tiap elemen pada state dipetakan dengan menggunakan tabel Inverse S-Box. Pengertian ini merupakan dari InvSubBytes.
c. InvMixColumns adalah kolom yang di dalam state yang dpat dikalikan dengan matrik perkalian dalam algoritma AES.

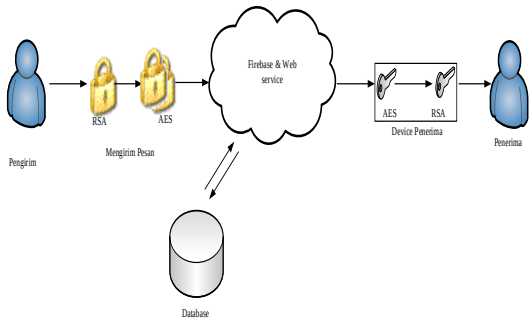
3. RANCANGAN APLIKASI DAN SISTEM

Berikut ini adalah penjelasan proses aplikasi *chatting* sebagai berikut :

3.1 Alur Proses Aplikasi

Dalam tahapan ini akan di jelaskan skema proses *chatting*nya:

a. Pada pengiriman pesan terjadi proses enkripsi dengan menggunakan algoritma RSA kemudian di enkrip menggunakan algoritma AES.
b. Pesan yang melalui *firbase* sudah terbentuk menjadi *ciphertext* dan disimpan di *database*.
c. Pesan yang tersimpan di *database* akan di kirim ke penerima melalui *firbase* ketika penerima dalam posisi *online*. Pesan akan di dekrip menggunakan algoritma AES kemudian di dekrip kembali menggunakan algoritma RSA pada *device* penerima. Maka saat penerima menerima pesan, pesan sudah bisa terbaca.
d. Saat penerima sedang *offline* pesan yang tersimpan di *database* akan ditampung sementara di *webservice*, kemudian pesan akan dikirim ke penerima ketika penerima tersebut sudah *online*. Pesan akan di dekrip menggunakan algoritma AES kemudian di dekrip kembali menggunakan algoritma RSA pada *device* penerima. Maka saat penerima menerima pesan, pesan sudah bisa terbaca.



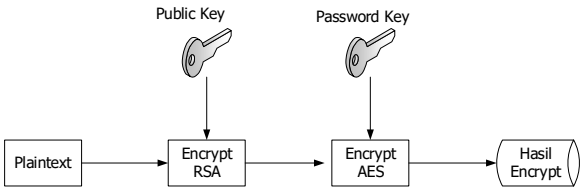
Gambar 1 : Skema proses chatting

3.2 Skema Proses Enkripsi

Proses mengubah *plaintext* menjadi pesan yang sulit dimengerti merupakan suatu proses enkripsi. langkah-langkah pada proses enkripsi yang dapat dilihat sebagai berikut :

a. Pengirim pesan melakukan *input* pesan yang akan dikirimkan.
b. *Public key* digunakan untuk enkripsi Pada saat pengiriman pesan terjadi.
c. Proses enkripsi pertama kali menggunakan algoritma RSA, hasil dari enkripsi tersebut kemudian di enkripsi kembali menggunakan algoritma AES.
d. Menghasilkan *output* berupa *ciphertext* dan disimpan di *database*.

Berikut skema proses enkripsi :



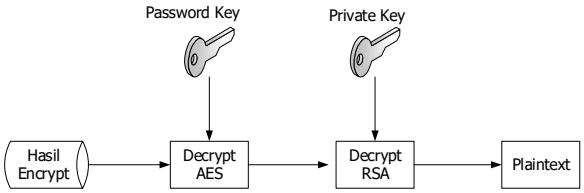
Gambar 2. Skema proses enkripsi

1.3 Skema Proses Dekripsi

Dekripsi merupakan proses pengembalian suatu informasi atau data yang telah dilakukan enkripsi. Pada aplikasi ini proses dekripsi pengguna harus memasukkan kunci *private* pada saat melakukan proses enkripsi. Pada proses ini akan menghasilkan *output* berupa teks yang dikembalikan seperti semula, sehingga isi teks dapat dimengerti kembali. Adapun langkah-langkah pada proses dekripsi dapat dilihat sebagai berikut :

i. Ketika penerima dalam posisi *on-connect* pesan yang diterima telah dilakukan proses dekripsi terlebih dahulu menggunakan kunci private.
ii. dekripsi pertama kali menggunakan algoritma AES kemudian didekripsi kembali menggunakan algoritma RSA.
iii. Menghasilkan *output* berupa pesan

Berikut merupakan skema proses dekripsi :



Gambar 3 : Skema proses dekripsi

3.4 Rancangan Basis Data

Basis data dibuat untuk menyimpan informasi atau data yang terjadi dalam sistem. Berikut merupakan rancangan basis data yang diperlukan:

3.4.1 Spesifikasi Basis Data

Berikut merupakan spesifikasi basis data pada sistem yang sedang dibangun.

i. Nama Tabel : *Chat*
Primary Key : *Uid*

Tabel 1 : Skema rancangan basis data chat

No.	Nama Field	Jenis	Lebar	Keterangan
1.	senderUid	Varchar	28	Uid pengirim pesan

2.	sender	Varchar	100	Email pengirim pesan
3.	userNameSender	Varchar	50	Name pengirim pesan
4.	receiverUid	varchar	28	Uid penerima pesan
5.	receiver	varchar	100	Email penerima pesan
6.	messageFrom	Longtext	42949672	Pengirim pesan
7.	messageTo	Longtext	42949672	Penerima Pesan
8.	timestamp	Integer	13	Waktu pengiriman pesan

ii. Nama Tabel : *User*
Primary key : *Uid*

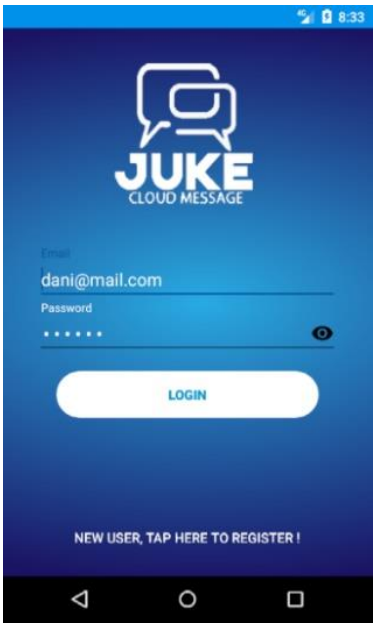
Tabel 2 : Skema rancangan basis data user

No.	Nama Field	Jenis	Lebar	Keterangan
1.	uid	Varchar	28	Uid pengguna
2.	email	Varchar	100	Email pengguna
3.	name	varchar	50	Nama pengguna
4.	Password	Longtext	4294967295	Password penguna
5.	firebaseToken	varchar	152	Token firbase
6.	pushToken	Varchar	152	Token *push notification*
7.	publickey_rsa	Longtext	4294967295	Rsa public key pengguna
8.	privatekey_rsa	Longtext	4294967295	Rsa private key pengguna

4. HASIL DAN PEMBAHASAN

4.1 Tampilan *Login*

Berikut adalah tampilan awal aplikasi chatting pada saat di jalankan, Bentuk tampilannya bisa di lihat pada gambar di bawah ini.



Gambar 4 : Tampilan Login

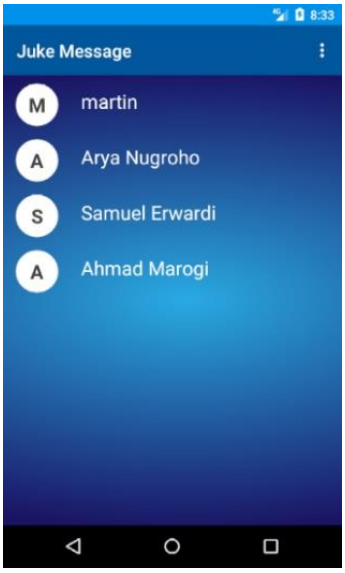
4.2 Tampilan Halaman Register

Pada halaman register pengguna diminta memasukkan beberapa informasi data seperti Nama lengkap, e-mail, password, confirm password. Setelah pengguna memasukkan informasi data yang diperlukan, pengguna dapat menekan tombol register untuk melakukan pendaftaran user tersebut. Bentuk tampilannya bisa di lihat pada gambar 5 di bawah ini.

Gambar 5 : Tampilan Register

4.3 Tampilan *List Contact*

Setelah pengguna melakukan *login*, pengguna akan masuk ke layar *home* dan dapat melakukan *chat* dengan teman yang sudah terdaftar di kontak. Dalam tampilan layar *home* berisi daftar kontak,. Bentuk dari tampilan layar *home*.



Gambar 6 : Tampilan home

4.4 Tampilan Halaman *Chatting*

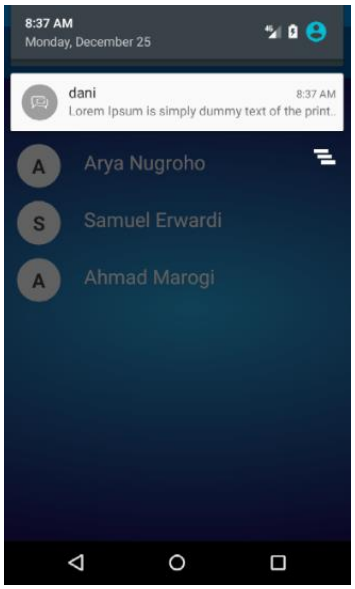
Pengguna akan melakukan *chatting* dengan teman, dengan cara memilih teman yang akan di ajak untuk *chatting* di *list contact* kemudian akan muncul ke tampilan layar *chatting*. Bentuk tampilan layar *chatting*.

Gambar 7 : Tampilan chatting

4.5 Tampilan Notification *Chatting*

Tampilan notification chatting pada pengguna akan ditampilkan bila pengguna sedang tidak membuka aplikasi chatting, dan pengguna mendapatkan pesan dari pengguna lainnya yang berada di kontak pengguna dapat di lihat pada gambar 8 di bawah ini.

.



Gambar 8 : Tampilan notification chatting

4.6 Evaluasi Aplikasi

Dari aplikasi ini dapat ditemukan beberapa kekurangan dan kelebihan saat dilakukan pengujian aplikasi, sebagai berikut :

i. Kelebihan Aplikasi

1) Dalam pengiriman pesan antar aplikasi mengunakan kriptografi sebagai pengamanan data untuk saling bertukar pesan, sehingga data yang dikirim antar penguna lebih aman dalam menjaga kerahasiaan data.
2) Dengan menerapkan algoritma RSA dan AES pada proses pengiriman data berupa pesan teks, berjalan dengan baik sehingga isi pesan terjaga keaslian datanya.
3) Aplikasi ini berjalan pada mobile android mengunakan firbase cloud messages, sehingga pertukaran data terjadi secara realtime.

ii. Kekurangan Aplikasi

1) Jumlah karakter pesan atau teks yang dikirimkan antara pengguna dalam satukali proses pengiriman terbatas.
2) Pengiriman pesan hanya berupa teks atau string.

3) Kekurangan fiture dibandingkan dengan beberapa aplikasi chatting yang berkembang saat ini.

5. KESIMPULAN

Berdasarkan pengkajian aplikasi yang telah dilakukan terhadap permasalahan yang ada maka didapatkan suatu kesimpulan dan saran untung pengembangan aplikasi ketahap berikutnya untuk menjadikan suatu aplikasi yang lebih sempurna dan dapat digunakan dengan baik. Beberapa hal yang dapat disimpulkan diantaranya sebagai berikut.

5.1 Kesimpulan

Dengan adanya implementasi kriptografi RSA dan AES pada aplikasi chatting, maka penulis dapat mengambil kesimpulan bahwa :

a. Algoritma kriptografi RSA dapat dikombinasikan dengan algoritma AES pada aplikasi chatting berbasis android.
b. Untuk aplikasi chatting yang berkembang saat ini, dan pada aplikasi yang dikembangkan pada tugas akhir ini sudah menggunakan kriptografi sebagai pengamaman data.
c. Dengan diterapkan algoritma RSA dan AES pada proses pengiriman data, maka data yang tersimpan di database tidak dapat terbaca oleh pihak luar yang tidak berwenang untuk mengetahui isi data tersebut..

5.2 Saran

Berikut beberapa saran yang diajukan untuk penggunaan ataupun pengembangan aplikasi ini adalah :

i. Pesan yang dikirimkan hanya berupa teks, sehingga kedepannya dapat di kembangkan lagi untuk pengiriman berupa file, foto, atau suara.
ii. Aplikasi dikembangkan menjadi berbasis web.
iii. Seperti kebanyakan aplikasi chatting yang berkembang saat ini, adanya fitur pesan yang terkirim sudah dibaca oleh penerima atau belum.

DAFTAR PUSTAKA

[1] Ariyus, Dony 2008, *Pengantar Ilmu Kriptografi, Teori Analisis, dan Implementasi*, Yogyakarta, Andi
[2] Arief, A. dan Saputra, R. (2016). Aplikasi Instant Messaging dengan Algoritma Kriptografi RSA-CRT, *Scientific Journal of Informatics*, 3(1), hal. 46–54.
[3] Ginting, A., Isnanto, R. R. dan Windasari, I. P. (2015) Aplikasi kriptografi email dengan Algoritma Kriptografi RSA. *Jurnal Teknologi dan Sistem Komputer*, 3(2), hal. 253–258.
[4] Justicia, Tiofan, L., Tolle, H. dan Amalia, F. (2017) Rancang Bangun Aplikasi Messaging Berbasis Voice Interaction Bagi Penderita Tunanetra Pada Sistem Operasi Android, 1(June), hal. 620–627.
[5] Wijaya, A. (2015) Sistem Enkripsi Menggunakan Algoritma Aes-128 Pada Prototype Community Messenger Berbasis Android Encryption System Using Aes-128 Algorithm on Prototype Community Messenger Android-Based, 2(2), hal. 3306–3311.