APLIKASI *MESSENGER* KRIPTOGRAFI UNTUK MENGAMANKAN PESAN TEKS MENGGUNAKAN ALGORITMA RIJNDAEL

Idris Pradyatna[1], Nurudin Santoso[2]

[1] Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
[1] idris.pra@gmail.com, [2] nurudin@polinema.ac.id

Abstrak

Saat ini keamanan pesan teks diperlukan pada saat kita mengirimkan pesan-pesan yang penting atau bersifat rahasia. Dan salah satu media yang digunakan untuk berbagi informasi adalah *messenger*. Makalah ini mengusulkan sebuah aplikasi *messenger* dengan mengimplementasikan algoritma *block cipher Rijndael* untuk mengamankan pesan teks. Selain itu, makalah ini juga menjelaskan nilai akurasi proses enkripsi-dekripsi yang hampir mencapai 100%. Dengan demikian, aplikasi *messenger* ini diharapkan untuk dapat membantu dalam pengamanan pesan teks.

Kata Kunci: *Messenger*, Kriptografi, *Block Cipher*, *Rijndael* 128-bit, Android

1. Pendahuluan

Manusia tidak bisa lepas dari keingintahuan terhadap informasi, bisa yang bersifat umum, pribadi, penting. Salah satu cara untuk bertukar informasi adalah melalui aplikasi messenger. Pesan-pesan yang dikirimkan melalui aplikasi messenger yang bersifat pribadi seharusnya dijaga keamanan dan kerahasiannya, sehingga dibutuhkan sebuah cara untuk menjaga keamanan dan kerahasiaan informasi tersebut. Salah satu cara yang dapat dilakukan adalah dengan teknik enkripsi dan dekripsi. Teknik ini berguna untuk mengubah pesan, data, maupun informasi agar tidak dapat dimengerti oleh orang lain selain penerima yang berhak dan mengetahui teknik dekripsinya. Teknik enkripsi dan dekripsi dikenal dan dipelajari dalam kriptografi.

Kriptografi secara umum adalah ilmu dan seni untuk menjaga kerahasiaan informasi. Algoritma kriptografi mempunyai karakteristik, keunggulan serta kelemahan masing-masing. Algoritma kriptografi dibedakan berdasarkan pada penggunaan kunci saat enkripsi dan dekripsi, yaitu: kunci simetris atau kunci non simetris.

Berdasarkan proses dan alur pengolahan datanya, kriptografi kunci simetri dibagi menjadi dua, yaitu block cipher dan stream cipher. Pada block cipher, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama dan blok plainteks yang sama tersebut akan dienkripsi menjadi blok cipherteks yang sama bila digunakan kunci yang sama pula. Ini berbeda dengan stream cipher dimana bit-bit plainteks yang sama akan dienkripsi menjadi bit-bit cipherteks yang berbeda setiap kali dienkripsi. Jika berbicara tentang algoritma block cipher, Rijndael merupakan salah satu algoritma simetris block cipher yang biasa digunakan. Rijndael dikembangkan oleh dua kriptografer asal Belgia, Joan Daemen dan Vincent Rijmen. Selain itu, algoritma ini juga ditetapkan oleh NIST (National Institute of Standards and Technology) sebagai pengganti DES yang dirasa tidak aman lagi.

2. Landasan Teori

2.1 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat dan mata-mata. Namun saat ini kriptografi lebih dari sekedar privacy, tetapi juga tujuan data integrity, authentication dan non-repudiation.

2.2 Algoritma Rijndael

Rijndael termasuk dalam jenis algoritma kriptografi yang sifatnya simetri dan block cipher. Dengan demikian algoritma ini mempergunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu.

Rijndael mempunyai ukuran blok dan kunci yaitu 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi. Garis besar Algoritma Rijndael yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan round key):

a. AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga initial round.

b. Putaran sebanyak 9 kali, karena menggunakan kunci 128-bit. Proses yang dilakukan pada setiap putaran adalah:
   - SubBytes: substitusi byte dengan menggunakan tabel substitusi (S-box).
   - ShiftRows: pergeseran baris-baris array state secara wrapping.
   - MixColumns: mengacak data di masing-masing kolom array state.
   - AddRoundKey: melakukan XOR antara state sekarang round key.

c. Final round: proses untuk putaran terakhir:
   - SubBytes
   - ShiftRows
   - AddRoundKey

Tabel 1. Variasi blok Rijndael

	Nk	*Nb*	*Nr*
128 bit	4	4	10
192 bit	6	4	12
256 bit	8	4	14

Nk = Panjang Kunci
Nb = Ukuran Blok
Nr = Jumlah Putaran

Algoritma Rijndael memiliki tiga parameter, yaitu:

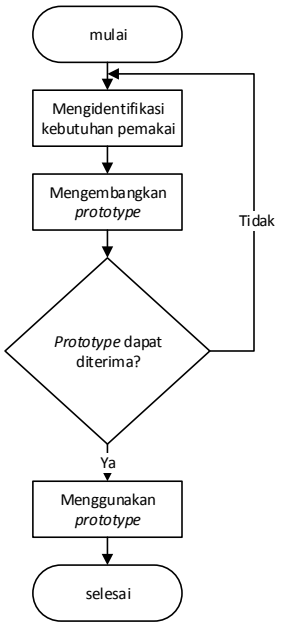
a. Plainteks: array yang berukuran 16-byte, yang berisi data masukan.
b. Cipherteks: array yang berukuran 16-byte, yang berisi hasil enkripsi.
c. Kunci: array yang berukuran 16-byte, yang berisi kunci cipher (disebut juga cipher key).

2.3 Android

Android adalah sebuah sistem operasi mobile yang berbasiskan pada versi modifikasi dari linux. Pertama kali sistem operasi ini dikembangkan oleh perusahaan Adroid.Inc.

3. Metodologi Penelitian

Metode penelitian yang digunakan adalah metode prototyping. Metode ini dibagi menjadi empat tahap, antara lain: analisis, rancangan, pengujian sistem, dan evaluasi.



Gambar 1. Alur Metode *Prototyping*

3.6. Analisis

Pada tahapan ini menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Tahapan ini meliputi identifikasi objek kebutuhan, resiko kesalahan aplikasi, dan merumuskan hipotesa prototype.

3.7. Rancangan

Setelah tahap analisis dilakukan, maka pemrogram mendesain secara terperinci sebuah rancangan prototype aplikasi yang menggambarkan keseluruhan aplikasi dan resiko-resiko yang mungkin berpengaruh pada aplikasi.

3.8. Pengujian Sistem

Pengujian sistem bertujuan untuk menemukan kesalahan-kesalahan yang terjadi pada sistem dan melakukan revisi sistem. Tahap ini penting untuk memastikan bahwa sistem bebas dari kesalahan.

3.9. Evaluasi

Prototype harus dicoba-coba untuk menentukan perilakunya dan mengumpulkan keluaran dari hasil eksekusi sistem sehingga didapat aplikasi yang sesuai dengan keinginan. Hasil dari implementasi akan dievaluasi untuk menilai kebenaran dan efisiensi aplikasi.

4. Analisis dan Perancangan

4.1. Analisis

Analisa kebutuhan sistem dibagi menjadi dua, yaitu kebutuhan software dan hardware.

a. Kebutuhan software, antara lain:
   - Microsoft Windows 7
   - Eclipse 4.4 (Luna)

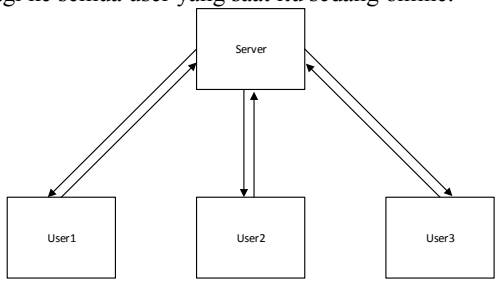
- Node JS
- Sublime Text
- Android ADT
- Android SDK
- Genymotion Android Emulator

b. Kebutuhan minimum hardware, antara lain:

- Laptop atau Komputer (PC) dengan spesifikasi minimum P4 2.4GHz, 1GB RAM
- Smartphone Android dengan OS minimum 4.0 (Ice Cream Sandwich)

4.2. Perancangan

Gambaran umum alur proses aplikasi messenger yaitu dimulai dengan pesan teks yang dikirim akan diterima oleh server (dibuat menggunakan Node JS dibantu oleh modul express dan socket.io). Kemudian pesan teks yang yang diterima server akan dikirimkan lagi ke semua user yang saat itu sedang online.



Gambar 2. Alur Proses Aplikasi

4.3. Peancangan Antar Muka

Perancangan antar muka sangat dibutuhkan. Karena perancangan antar muka dapat digunakan untuk dijadikan rapor awal bagi developer dari pengguna. Sehingga dengan rancangan tersebut aplikasi tersebut dapat dinilai apakah layak digunakan atau tidak.

Gambar 3. Mock Up Halaman Utama

5. Implementasi

5.1 Server

Aplikasi yang digunakan dalam pembuatan server adalah Node JS dibantu dengan modul express dan socket.io. Server ini akan digunakan untuk meneruskan data (pesan teks) yang dikirimkan oleh klien kepada klien yang lainnya. Sehingga peran server sangat penting untuk aplikasi messenger yang akan dibuat.

5.2 Mengirim Pesan Teks

Fitur utama dari messenger, fitur ini digunakan untuk berbagi pesan teks kepada user lainnya. Sehingga para user yang menggunakan aplikasi ini bisa bertukar informasi berupa teks.

5.3 Enkripsi Pesan Teks

Fitur ini digunakan untuk membuat pesan teks menjadi tidak bisa dipahami oleh orang selain penerima. Jadi, sebelum dikirimkan, pesan teks akan dienkripsi. Selain itu penggiriman juga disertai kunci untuk mengembalikan pesan teks ke kondisi semula.

5.4 Dekripsi Pesan Teks

Fitur ini digunakan untuk mengembalikan pesan teks ke kondisi dimana pesan teks dapat dibaca. Fitur ini membutuhkan kunci untuk mengaktifkan prosesnya.

6. Uji Coba dan Pembahasan

6.1. Uji Fungsional

Uji coba fungsional ini dilakukan menggunakan blackbox testing, karena blackbox memfokuskan pada keperluan fungsional dari software.

Tabel 2. Pengujian Fungsional

No.	Proses	Keterangan	Cek
1.	Kirim Pesan	Pesan yang ditulis dan berhasil dikirim akan hilang dan muncul pada area *messaging*	V
2.	Terima Pesan	Pesan yang dikirimkan oleh pengguna lainnya akan masuk secara real-time pada area *messaging*	V
3.	Enkripsi Pesan	Setelah menulis pesan dan memasukkan kunci, maka pesan akan dienkripsi dan menghasilkan cipherteks yang dikirim serta dimunculkan pada area *messaging*	V
4.	Dekripsi Pesan	Sebelum pesan teks didekripsi, pengguna diharuskan untuk memasukkan kunci terlebih dahulu agar pesan teks yang sebelumnya telah dienkripsi dapat dibaca	V

6.2. Pengujian Performa

Pengujian menggunakan kunci yang sama yaitu "idris" dengan lima kalimat yang berbeda.

Tabel 3. Pengujian Performa

Kalimat Awal	Hasil Enkripsi	Hasil Dekripsi
Bagaimana kalau bertemu di Jl.Mawar no 5	BPc9fSDuBjV8DMKDjfC2rMv3bvlyWt8UQXePjuJx0Ix5H9D9QYHZ8QVtcPCq+JM8	Bagaimana kalau bertemu di Jl.Mawar no 5
Hati-hati ya	UMmramwWjsDe64JGg8P4YA==	Hati-hati ya
Kemarin kamu kemana?	CjZAhpolmHMEpxmfwhrVPSOupW0I4KZ5utQszozy+L0=	Kemarin kamu kemana?
Nomor hp ku 081234567	6xRXBPl7xgj7+jl3rTktpO+cWfjHmqxKNV9ehIT9tWA=	Nomor hp ku 081234567
hayo	nYjrraNalY41o2Xt7lpvFQ==	hayo

6.3. Perhitungan Nilai Akurasi

$$Nilai\ Akurasi = \frac{\sum proses\ dekripsi\ yang\ berhasil}{\sum proses\ uji\ coba} \times 100\%$$

Nilai akurasi = 5/5 * 100%

Nilai akurasi = 100%

7. Kesimpulan dan Saran

7.1. Kesimpulan

Dari hasil pengamatan selama perancangan, implementasi, dan uji coba serta pengujian yang telah dilakukan. Maka dapat disimpulkan bahwa. Aplikasi messenger dengan mengimplementasikan algoritma block cipher Rijndael sebagai pengaman pesan teks dapat berjalan dengan baik. Selain itu dalam implementasinya, nilai akurasi proses enkripsi-dekripsi sebesar 100%.

7.2. Saran

Aplikasi messenger sederhana ini masih dapat dikembangkan. Mulai dari pengembangan chat room agar dapat berkomunikasi secara private serta menambahkan database agar pesan-pesan yang sebelumnya telah terkirim dapat dilihat kembali. Selain itu, dalam sisi pengamanan juga masih bisa dikembangkan, seperti menggabungkan algoritma block cipher Rijndael ini dengan algoritma kriptografi lainnya agar lebih aman.

Daftar Pustaka

Aprianto, Y. dan Kurniawan, R. 2014. Rancang Bangun Aplikasi Enkripsi dan Dekripsi Citra Digital Menggunakan Algoritma Rijndael Berbasis Java SE. Jurnal. Palembang: STMIK GI MDP.

Fadli, H.G. 2009. Studi dan Implementasi Algoritma Rijndael Untuk Enkripsi Halaman Web HTML. Jurnal. Bandung: Institut Teknologi Bandung.

Munir, R. 2004. Kriptografi: Advance Encryption Standard (AES). Bahan Perkuliahan. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.

NIST. 2004. National Institute of Standards and Technology. http://www.nist.gov. Tanggal akses: 29 Januari 2015.

Sadikin, R. _. Kriptografi Untuk Keamanan Jaringan. Yogyakarta: ANDI.

Surian, D. 2006. Algoritma Kriptografi AES Rijndael. Makalah. Jakarta: Teknik Elektro, Universitas Tarumanagara.

Utama, F.B. 2014. Aplikasi SMS Kriptografi Dengan Metode RSA Pada Smartphone Android. Skripsi Naskah Publikasi. Yogyakarta: Sekolah Tinggi Manajemen Informatika Dan Komputer Amikom.