Jurnal Coding Sistem Komputer Untan
Volume 05, No. 2 (2017), hal. 1-12 ISSN : 2338-493X

# PENGEMBANGAN APLIKASI *CHAT MESSENGER* DENGAN METODE *ADVANCED ENCRYPTION STANDARD* (AES) PADA *SMARTPHONE*

**[1]Indra Suryanto, [2]Drs. Cucu Suhery, M.A., [3]Yulrio Brianorman, S.Si., M.T.**
[1][2][3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura Pontianak
Jl. Prof. Dr. H. Hadari Nawawi, Pontianak, Kalimantan Barat
Telp./Fax.: (0561) 577963
e-mail:
[1]indra@untan.ac.id, [2]csuhery@siskom.untan.ac.id, [3]yulrio.brianorman@siskom.untan.ac.id

***Abstrak***

*Chat messenger merupakan salah satu sarana komunikasi yang paling banyak digunakan oleh pengguna smartphone saat ini. namun pesan yang dikirimkan belum tentu aman dari kejahatan cybercrime seperti penyadapan transmisi pesan dan pemanipulasian pesan. Untuk mengatasi permasalahan tersebut, pada penelitian ini akan dibuat sebuah aplikasi chat messenger yang menggunakan kriptografi Advanced Encryption Standard (AES) sebagai sistem keamanan pesannya. Kriptografi AES merupakan standar informasi federal yang ditetapkan oleh National Institute of Standards and Technology (NIST). Pesan yang dikirimkan akan dienkripsi terlebih dahulu menggunakan kunci menjadi chipertext sehingga pesan tersebut tidak dapat dibaca oleh orang yang tidak berkepentingan walaupun terjadi penyadapan pada saat transmisi pesan. Pada saat pesan sampai di penerima dilakukan proses dekripsi pesan menggunakan kunci yang sama saat proses enkripsi dimana chipertext yang masuk akan diubah kembali menjadi plaintext atau pesan yang dapat dibaca sesuai pesan awal yang dikirimkan. Jika pada saat transmisi, pesan yang dikirimkan diubah atau dimanipulasi oleh orang maka pesan yang masuk tidak akan dapat dibaca karena pada saat proses dekripsi, kunci dan chipertext tidak akan cocok satu sama lain. Dengan aplikasi chat messenger yang terimplementasi dengan kriptografi AES pengguna dapat berkomunikasi dengan aman tanpa takut pesan tersebut disadap atau dimanipulasi. Pertukaran pesan akan dikelola oleh NodeJS sesuai dengan id perangkat yang terhubung pada server sehingga pengguna dapat bertukar pesan secara realtime.*

**Kata Kunci**: *chat messenger, kunci, kriptografi, Advanced Encryption Standard, AES, enkripsi, dekripsi, chipertext, plaintext, NodeJS*

## 1. PENDAHULUAN

Perkembangan teknologi informasi saat ini sangat pesat terutama dalam masalah komunikasi. Hal ini dapat dilihat dari banyaknya bermunculan *chat messenger* seperti Line, Whatapps, KakaoTalk dan WeChat yang memberikan kemudahan kepada pengguna untuk melakukan pengiriman pesan dan data dengan waktu singkat. *Chat Messenger* adalah suatu sistem pengiriman pesan secara *realtime* melalui jaringan internet dari satu perangkat ke perangkat yang lain. Menurut survei dari organisasi We Are Social pada Januari 2015, Indonesia memiliki populasi penduduk sebanyak 255,5 juta orang dengan pengguna paket data internet melalui *mobile* sejumlah 398,2 juta orang atau setara 121% populasi penduduk Indonesia. Pada survei perkembangan pengguna telepon genggam sejak januari 2014 sampai januari 2015 menunjukan adanya perkembangan sebanyak 9%.[1]

Dilihat dari data diatas maka dapat disimpulkan bahwa pengguna telepon genggam sangat banyak, dan menurut survei yang dilakukan oleh Nielson menunjukan penggunaan telepon genggam lebih banyak digunakan untuk *Chat* yaitu dengan rata-rata sebanyak 72% sehari. Hal ini juga menyebabkan perlu kewaspadaan munculnya masalah kejahatan yang disebut dengan *CyberCrime* atau kejahatan melalui jaringan informasi. Beberapa contoh dari kasus *CyberCrime* yang dapat menyerang pengguna *chat messenger* antara lain seperti penyadapan transmisi pesan dan pemanipulasian pesan.[2]

Untuk menghadapi permasalahan diatas maka dirasakan perlu untuk membuat suatu sistem keamanan. Salah satu cara untuk meningkatkan keamanan pesan adalah dengan menggunakan kriptografi. Kriptografi merupakan ilmu yang mempelajari cara-cara mengamankan pesan dengan cara disandikan. Dengan penggunaan Kriptografi, data atau pesan akan diamankan dengan cara dienkripsi menjadi *ciphertext* sehingga data atau pesan tidak dapat dibaca langsung. Supaya data atau pesan dapat dibaca diperlukan suatu proses dekripsi yang berfungsi untuk mengubah data *chipertext* kembali menjadi *plaintext* yang dapat dibaca.

Penelitian-penelitian tentang *chat messenger* dan sistem pengamanan data sudah pernah dilakukan. Penelitian dalam pembangunan aplikasi *chat messenger* pada android dengan judul "Merancang dan Membangun Aplikasi Chat Messenger Untuk Android"[3]. Penelitian yang dilakukan oleh Bakhtiar Wijayanto adalah membangun sebuah aplikasi *chat messenger* yang dapat berjalan di sistem operasi Android, tetapi tidak memperhatikan aspek keamanan pesan. Penelitian yang dilakukan rawan akan serangan *CyberCrime* sehingga perlu untuk dibuat sistem keamanan.

Penelitian tentang pengiriman pesan singkat dengan menggunakan kriptografi pernah dilakukan dengan judul "Implementasi Algoritma Kriptografi AES 128 Bit Sebagai Pengaman SMS pada Smartphone Berbasis Android"[4]. Penelitian yang dilakukan oleh Joko Tri Susilo adalah pengimplementasian Kriptografi AES pada pesan singkat. Penelitian tentang implementasi AES pada pesan singkat dirasa cukup efektif karena pesan yang dikirim sudah diamankan dengan enkripsi AES. *Advanced Encryption Standard* (AES) adalah salah satu metode kriptografi dengan blok *chipertext* simetris yang dapat mengenkripsi dan mendekripsi sebuah pesan atau data. AES merupakan Standar Pemrosesan Informasi Federal yang ditetapkan oleh *National Institute of Standards and Technology* (NIST). Metode AES ini dibuat untuk menggantikan metode *Data Encryption Standard* (DES) untuk meningkatkan aspek keamanan.

Penelitian tentang AES pernah dilakukan dengan judul "Advanced Encryption Standard (AES)"[4]. Penelitian ini membahas tentang proses AES dan keunggulan AES. Penelitian ini dapat menjelaskan keunggulan AES dibandingkan kriptografi lain. Berdasarkan referensi yang dijabarkan diatas akan dilakukan suatu penelitian untuk membuat aplikasi chat messenger yang terintegrasi dengan kriptografi AES. Penelitian ini diharapkan dapat digunakan untuk pertukaran pesan yang lebih aman.

## 2. LANDASAN TEORI

### 2.1. Kriptografi

Kriptografi adalah ilmu yang mempelajari cara mengamankan data dengan cara merubah perubahan kata biasa menjadi kata yang tidak bisa dimengerti. Bidang ilmu kriptografi sangat dekat dengan berbagai rumus dan algoritma sehingga bidang ilmu kriptografi dimasukan ke dalam bagian ilmu matematika. Ilmu yang mempelajari bagaimana mempelajari kelemahan dari sebuah kriptografi adalah kriptoanalisis[6]. Kriptografi terdiri dari tiga fungsi dasar, yaitu :

a. Enkripsi yang merupakan proses perubahan naskah asli yang disebut *plaintext* menjadi naskah acak yang tidak dapat dimengerti yang disebut *chipertext*.
b. Dekripsi yang merupakan kebalikan dari enkripsi. Naskah yang telah dienkripsi menjadi *chipertext* akan diubah kembali menjadi *plaintext*.[7]
c. Kunci atau *key* yang merupakan kunci rahasia yang digunakan untuk mengenkripsi dan dekripsi pesan. Kunci yang digunakan harus disepakati oleh pihak pengirim dan penerima yang disesuaikan dengan algoritma kriptografi yang digunakan.[6]

Kriptografi dapat dibagi menjadi tiga berdasarkan kunci yang dipakai yaitu Kriptograsi Simetri (*Symmetric Cryptography)*, Kriptografi Asimetri

(*Asymmetric Cryptography*), dan Fungsi *Hash* (*Hash Function*). Contoh dari algoritma kriptografi yang terkenal adalah DES (*Data Encryption Standar*), 3DES (*Triple Data Encryption Standard*), RC4 (*Rivest Cipher 4*), RC5, RC6, dan AES (*Advanced Encryption Standard*). (Dony, 2008)

Pada penelitian ini algoritma kriptografi yang digunakan adalah kriptografi *Advanced Encryption Standard* (AES) yang merupakan kriptograsi dengan kunci simetri. Algoritma AES ini akan digunakan untuk mengenkripsi pesan yang akan dikirim menggunakan aplikasi *chat messenger*. Pesan yang diterima akan didekripsi kembali menggunakan algoritma AES pada saat pesan telah sampai pada penerima.

### 2.2. *Advanced Encryption Standard*

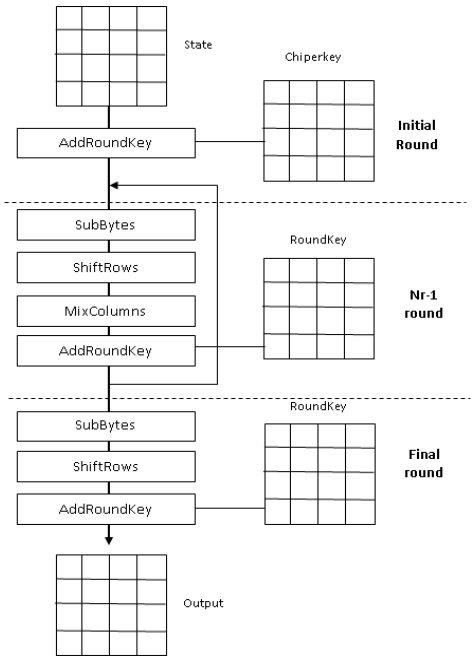
Algoritma kriptografi bernama Rijndael yang didesain oleh oleh Vincent Rijmen dan John Daemen asal Belgia keluar sebagai pemenang kontes algoritma kriptografi pengganti DES yang diadakan oleh NIST (National Institutes of Standards and Technology) milik pemerintah Amerika Serikat pada 26 November 2001. Algoritma Rijndael inilah yang kemudian dikenal dengan Advanced Encryption Standard (AES). Setelah mengalami beberapa proses standardisasi oleh NIST, Rijndael kemudian diadopsi menjadi standard algoritma kriptografi secara resmi pada 22 Mei 2002. Pada 2006, AES merupakan salah satu algoritma terpopuler yang digunakan dalam kriptografi kunci simetrik.

AES ini merupakan algoritma block cipher dengan menggunakan sistem permutasi dan substitusi (P-Box dan S-Box) bukan dengan jaringan Feistel sebagaiman block cipher pada umumnya. Jenis AES terbagi 3, yaitu AES-128, AES-192, AES-256. Pengelompokkan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka di belakang kata AES menggambarkan panjang kunci yang digunakan pada tiap-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round. Perbandingan jumlah round dan key dapat dilihat pada tabel 1.[5]

Tabel 1. Perbandingan Jumlah Round dan Key

| Tipe AES | Jumlah Key (Nk words) | Ukuran Blok (Nb words) | Jumlah Putaran (Nr) |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

Pada Tabel 2.1 menjelaskan perbandingan dari 3 tipe kunci AES. Word diatas diartikan sebagai 32 byte. AES-128 memiliki jumlah *key* 4 x 32 = 128 byte, ukuran blok 4x32 = 128 byte, dan melakukan 10 kali putaran proses algoritma AES. AES-192 dan AES-256 memiliki ukuran blok yang sama dengan AES-128, yang membedakannya adalah AES-192 memiliki jumlah *key* 192 byte dengan putaran sebanyak 12 kali putaran dan AES-256 memiliki jumlah *key* 256 byte dengan putaran sebanyak 14 kali putaran

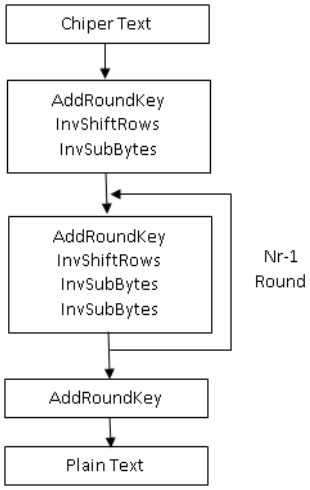



Gambar 1
Ilustrasi Proses Enkripsi AES

Pada gambar 1 mengilustrasikan proses enkripsi Pada AES. Secara garis besar cara kerja AES pada blok 128 bit adalah :

1. *AddRoundKey*: melakukan *XOR* antara *state* awal (plainteks) dengan *cipherkey*. Tahap ini disebut juga *initial round*.
2. Putaran sebanyak *Nr* – 1 kali (Total round – 1). Proses yang dilakukan pada setiap putaran adalah:
   a. *SubBytes*: substitusi *byte* dengan menggunakan tabel substitusi (*S-box*).
   b. *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
   c. *MixColumns*: mengacak data di masing-masing kolom *array state.*
   d. *AddRoundKey*: melakukan XOR antara *state* sekarang *round key*.
3. *Final round*: proses untuk putaran terakhir:
   *a. SubBytes*
   *b. ShiftRows*
   *c. AddRoundKey*.

Dekripsi AES dilakukan dengan membalikan cipher yang sudah dienkripsi. Transformasi cipher dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey seperti yang diilustrasikan pada gambar 2.[8]




Gambar 2
Ilustrasi Proses Dekripsi AES

## 2.3. Android

Android merupakan sistem operasi mobile berbasis kernel Linux yang dikembangkan oleh Android Inc dan kemudian diakuisisi oleh Google. Sistem operasi ini bersifat open source sehingga para programmer dapat membuat aplikasi ini secara mudah. Kehadiran Android diperkirakan mampu bersaing dengan sistem operasi mobile lainnya seperti Blackberry, Symbian, dan IOS. Salah satu keunggulan Android terletak pada bervariasinya merek dan bentuk ponsel yang menggunakan sistem operasi ini.

Sistem operasi android memiliki fitur yang dimiliki oleh smartphone pada umumnya seperti aplikasi yang melimpah, email, fitur online seperti browser, dan banyak lagi. Sistem operasi ini cocok digunakan bagi pengguna internet karena Android memiliki layanan internet yang tidak terbatas. Pengguna aplikasi Google seperti Gmail maupun Google Maps dapat diakses dengan cepat menggunakan sistem operasi ini.[9]

Melihat keunggulan dan fasilitas yang ditawarkan maka penulis memilih menggunakan smartphone android. Pada penelitian ini akan dibuat suatu aplikasi chat messenger baru yang terintegrasi dengan kriptograi AES pada android Gingerbread.

## 2.4. Phonegap

Phonegap adalah sebuah framework pembuatan aplikasi mobile berbasis open source yang dikeluarkan oleh sebuah perusahaan di Amerika yang bernama Nitobi. Phonegap framework memungkinkan seseorang mengembangkan aplikasi native mobile dengan menggunakan keahlian HTML, CSS, dan Javascript. Sebuah aplikasi yang dibuat menggunakan framework phonegap dapat di-deploy ke berbagai platform seperti iOS, Android, Windows Mobile, Blackberry, WebOS, Symbian dan Bada.[10]

Fitur-fitur yang bisa diakses dengan menggunakan phonegap yaitu :

1. Accelerometer: untuk menangkap gerakan device dengan arah pada sumbu x, y, atau z
2. Camera: memungkinkan mengakses default aplikasi kamera pada device.
3. Compass: menentukan arah yang ditunjukan oleh device.
4. Contact: menyediakan akses ke database kontak yang ada pada device.
5. File: API untuk membaca atau menulis atau menelusuri sistem file pada device.
6. Geolocation: menyediakan akses untuk mengetahui keberadaan device.
7. Media: menyediakan akses untuk dapat memainkan media pada device.
8. Network: menyediakan akses untuk mengakses ke jaringan atau internet.
9. Notifikasi (alert, suara, getar): memberikan notifikasi kepada penggunanya berupa alert, suara, maupun getar.
10. Storage: aplikasi dapat mengakses tempat penyimpanan seperti memori.

Phonegap pada penelitian akan digunakan untuk meng¬-deploy aplikasi yang dibangun menggunakan angularjs yang merupakan bahasa pemograman web sehingga menjadi aplikasi android. Fitur phonegap yang digunakan pada penelitian antara lain network, notifikasi dan storage.

### 2.5. NodeJS

Node.js adalah platform yang dibangun pada Chrome's Javascript runtine untuk memudahkan pembuatan aplikasi dengan cepat. Node.js menggunakan event-driven, non-blocking I/O yang membuatnya ringan dan efisien. Node.js sangat cocok digunakan untuk membuat aplikasi real-time yang membutuhkan pertukaran data yang intensif.[11]

Node.js ini akan digunakan untuk membuat aplikasi pada sisi pc server dan digunakan sebagai web server. Aplikasi yang dibangun pada node.js ini berfungsi sebagai penghubung antara aplikasi client pada android dengan basis data yang digunakan yaitu MongoDB dan sebagai perantara yang menerima dan meneruskan pesan bagi pengguna baik pengirim maupun penerima.

### 2.6. MongoDB

MongoDB merupakan basis data NoSQL (not only sql) yang merupakan Document-Oriented Database dan merupakan open source project yang tersedia di github. MongoDb merupakan kumpulan-kumpulan dokumen yang berformat JSON dan kumpulan dokumen ini disebut Collection. MongoDB berbeda dengan basis data RDBMS (Relational DataBase Management System) seperti MySql, PosgreSQL, dan Oracle yang memerlukan SQL (Stucture Query Language) untuk mengakses data, MongoDB tidak mengenal istilah SQL.[10]

Basis data MongoDB ini akan digunakan pada proses pembuatan aplikasi chat messenger yang akan dibuat. MongoDb yang menggunakan Document-Oriented Database cocok digunakan untuk pembuatan chat messenger karena mendukung pengolahan basis data yang cepat dalam melakukan pertukaran pesan. MongoDB akan digunakan untuk menyimpan data-data pengguna dan juga sejarah percakapan yang dilakukan.

### 2.7. AngularJS

Angulas JS adalah framework javascript yang dikembangkan oleh Google dan banyak digunakan pada produk-produk yang dibuat oleh Google. Angular JS telah menjadi standarisasi untuk keperluan pembuatan Aplikasi Web Dinamis dari Sisi Client, karena kemudahan dan kecepatannya dalam melakukan komunikasi Server Ke Client. Angular JS digunakan untuk agar meningkatkan fungsi HTML untuk membangun Aplikasi Web. AngujarJS merupakan merupakan framework javascript yang menerapkan konsep MVC (Model View Controller).[10]

Aplikasi chat messenger yang akan dibuat memerlukan suatu sistem yang dinamis untuk melakukan perpesanan dan juga sistem lainnya maka dirasakan angularJS mampu menjadi pilihan. Kecepatan dan kemudahan melakukan komunikasi server ke client merupakan suatu poin tambahan sehingga bisa melakukan perpesanan dengan cepat. Aplikasi yang dibangun dengan AngularJS akan di-deploy ke android menggunakan phonegap.

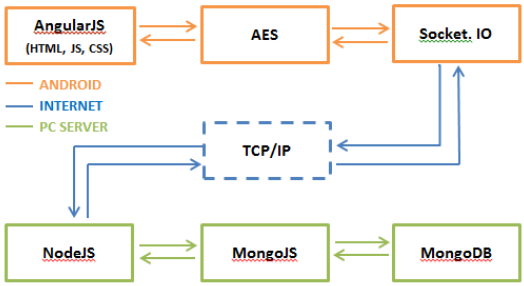
## 3. METODE PENELITIAN

Penelitian ini dimulai dari studi pustaka, yaitu mengumpulkan teori-teori serta kebutuhan yang diperlukan dalam pembuatan aplikasi. Langkah selanjutnya adalah perancangan antarmuka aplikasi yang akan dibuat yang kemudian dilanjutkan dengan pembuatan sistem aplikasi. Perancangan aplikasi sistem aplikasi sendiri akan diibagi menjadi dua tahap yaitu perancangan sistem *chat messenger* dan perancangan sistem enkripsi. Setelah perancangan sistem selesai dibuat akan dilanjutkan dengan tahap pengujian dan evaluasi sebelum kemudian di-implementasikan ke perangkat *smartphone.*

## 4. PERANCANGAN SISTEM

### 4.1. Diagram Blok Perancangan Sistem

Diagram blok perancangan sistem ditunjukan pada Gambar 3.



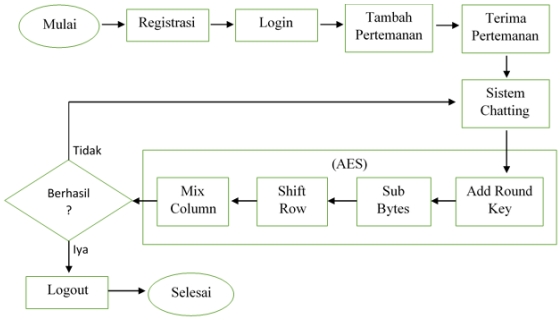

Gambar 3
Diagram Blok Perancangan *Chat Messenger*

Penjelasan dari digram blok rancangan system pada Gambar 4.1 adalah sebagai berikut:

1. AngularJS akan digunakan sebagai struktur utama dalam pembuatan aplikasi pada sisi mobile seperti pembuatan tampilan dan sistem *chat messenger*.
2. AES adalah sistem enkripsi yang akan dijalankan pada saat adanya pengiriman dan penerimaan data pesan.
3. Socket IO berfungsi sebagai komunikasi waktu nyata antara sisi *client* dan *server* yang dihubungkan melalui *TCP/IP*
4. NodeJS merupakan *server* yang akan dipakai untuk mengolah data sesuai dengan permintaan dan kebutuhan sistem. NodeJS akan dihubungkan ke MongoDB menggunakan MongoJS.
5. MongoDB merupakan database yang akan digunakan untuk menyimpan data dengan sistem *Document-Oriented*

### 4.2. Diagram Alir Perancangan Sistem

Supaya pembuatan program lebih terarah maka dibuat Diagram alir perancangan sistem seperti yang ditunjukkan Gambar 4.




Gambar 4
Diagram Alir Peracangan Sistem

Pada Gambar 4.2 dapat dilihat diagram alir perancangan sistem yang akan dibuat dengan tahapan sebagai berikut :

1. Sistem registrasi: Pada tahapan ini akan dibuat sistem registrasi agar user dapat mendaftarkan diri sebagai member.
2. Sistem login: Pada tahapan ini akan dibuat sistem login sebagai autentikasi untuk masuk ke aplikasi.

3. Tambah Pertemanan: Pada tahapan ini akan dibuat sebuah sistem yang digunakan untuk menambah daftar dengan cara mengirim permintaan pertemanan yang harus diterima oleh user yang ditambahkan.
4. Menerima Pertemenan: Pada tahapan ini akan dibuat sebuah sistem yang digunakan untuk menerima pertemanan yang sebelumnya sudah dikirimkan oleh sistem Tambah Pertemanan.
5. Sistem *Chatting*: Pada tahapan ini akan dibuat sistem pengiriman dan penerimaan data pesan yang akan terintegrasi dengan kriptografi AES.
6. Enkripsi AES: Pada tahap akan dibuat sistem kriptografi AES yang didalam terdapat pembuatan Add Round Key, Sub Bytes, Shift Row dan Mix Column. Kemudian akan diuji apakah AES berhasil terimplementasi dengan sistem *Chatting* jika gagal maka akan diulangi.
7. Logout: Pada tahap ini akan dibuat sistem untuk keluar dari aplikasi.

### 4.3. Desain Skema *Database*

Desain skema *database* dibuat agar sistem dapat melakukan penyimpanan dan penukaran data sesuai dengan kebutuhan sistem. Perancangan *database* ini dibuat dengan menggunakan database mongoDB sehingga *database* akan berupa kumpulan dokumen. Pada perancangan aplikasi *chat messenger* ini akan dibuat dua buah dokumen yaitu dokumen user dan dokumen message.

Pada dokumen akan dibuat key atau *field* jika diistilahkan menggunakan skema *database* berbasis sql. Pada dokumen user akan dibuat *key* seperti id, nama, email, password, socketed, friends (socketed, nama), pendingrequest (socketed, nama), friendrequest (socketed, nama) seperti pada Tabel 2. Pada dokumen message akan dibuat *key* seperti id, from, fromname, password, to, message, dan time seperti pada Tabel 3.

Tabel 2. *document* user

| **Key** | **Type** |
|---|---|
| _id | Objectid |
| Nama | String |
| Email | String |
| Password | String |
| Socketed | String |
| Friends | Array |
| Socketed | String |
| Nama | String |
| pendingrequest | Array |
| Socketed | String |
| Nama | String |
| friendrequest | Array |
| socketid | String |
| nama | String |

Tabel 3. *document* message

| **Key** | **Type** |
|---|---|
| _id | Objectid |
| from | String |
| fromname | String |
| password | String |
| to | String |
| message | String |
| time | String |

### 4.4. Perancangan Antarmuka (*Interface*)

Antarmuka berfungsi sebagai penghubung antara sistem dan pengguna, oleh karena itu desain adalah hal yang sangat penting untuk mempermudah pengguna menggunakan aplikasi. Desain tampilan aplikasi menggunakan AngularJS dengan *framework* yang digunakan adalah angular-mobile-ui sehingga tampilan yang dibangun sudah sesuai dengan tampilan untuk *mobile smartphone*. Tampilan aplikasi akan menyesuaikan dengan besar *smartphone* yang digunakan pengguna.

### 4.5. Perancangan Sistem *Chat Messenger*

Perancangan sistem *chat messenger* dibangun pada bagian aplikasi pada *smartphone* dan bagian s*erver*. Pada bagian *smartphone*, aplikasi yang dibangun

menggunakan AngularJS. Bagian *server* akan dibangun menggunakan NodeJS. Untuk menghubungkan kedua bagian sistem maka digunakan socketIO sehingga pertukan pesan bisa dilakukan secara realtime. Pada bagian *smartphone* akan dimasukan sistem kriptografi AES untuk sistem keamanan pesan selama proses transmisi data antar pengguna.

### 4.6. Perancangan Sistem Kriptografi AES

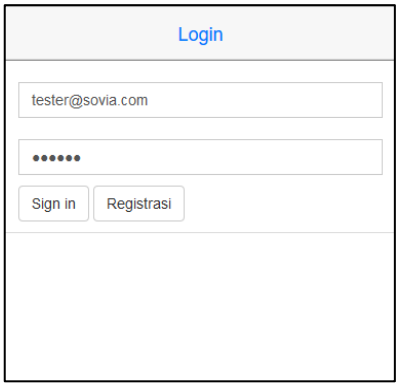
Sistem kriptografi AES akan dibuat pada aplikasi *chat messenger*. Sistem AES akan dibuat menggunakan *javascript*. Proses enkripsi dan dekripsi akan berjalan pada aplikasi *smartphone* (*end to end encryption*) sehingga pesan yang dikirimkan bisa aman selama proses transmisi data dan pesan yang masuk ke *server* juga tidak akan dapat dibaca karena berupa *chipertext*. Pesan hanya dapat dibaca setelah didekripsi pada saat sampai pada pengguna yang dituju.

## 5. HASIL DAN PEMBAHASAN

### 5.1. Perancanganan Antarmuka

*A. Halaman Login Aplikasi*

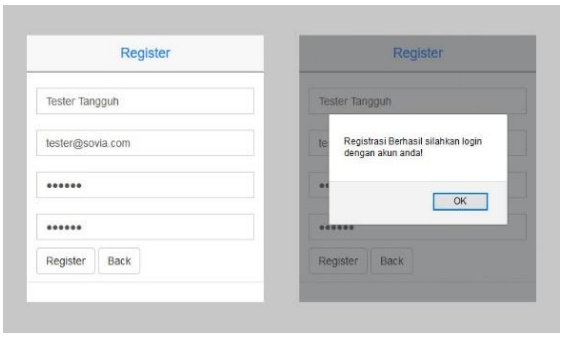
Ketika aplikasi dibuka, maka pengguna akan masuk ke halaman login dimana aka nada pilihan untuk login atau memilih registrasi jika belum terdaftar sebagai user. Tampilan login aplikasi bisa dilihat di Gambar 3.




Gambar 3
Halaman Login

*B. Halaman Registrasi Aplikasi*

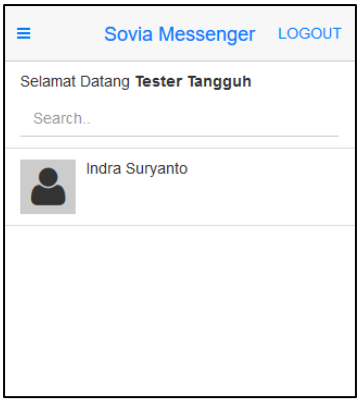
Halaman registrasi akan muncul pada saat dipilih dan memunculkan form untuk diiisi supaya pengguna bisa menggunakan fitur-fitur aplikasi. Tampilan halaman registrasi bisa dilihat pada Gambar 4.




Gambar 4
Halaman Registrasi

*C. Halaman Utama Setelah Login*

Halaman utama setelah melakukan login merupakan halaman dimana kita bisa melihat daftar teman yang telah kita undang menjadi teman chat seperti yang bisa dilihat pada Gambar 5.

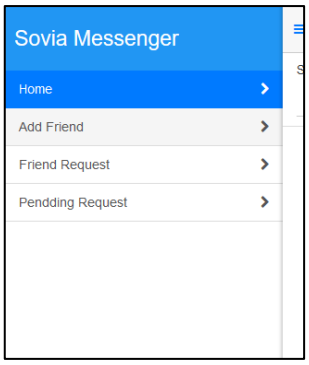



Gambar 5
Halaman Utama

*D. Halaman Menu Aplikasi*

Halaman menu aplikasi akan muncul disebelah kiri layar jika kita menekan tombol menu yang berada di header aplikasi. Halaman menu ini akan memunculkan pilihan seperti Home, Add Friend, Friend Request, dan Pending Request yang mengarahkan kita ke halaman

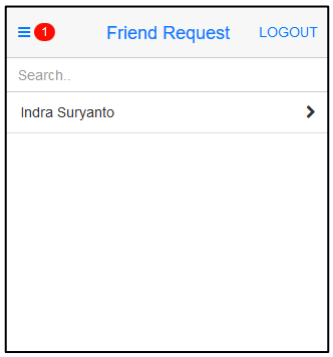
yang kita inginkan. Halaman menu ini bisa dilihat di Gambar 6.




Gambar 6.
Halaman Menu

*E. Halaman Permintaan Pertemanan*

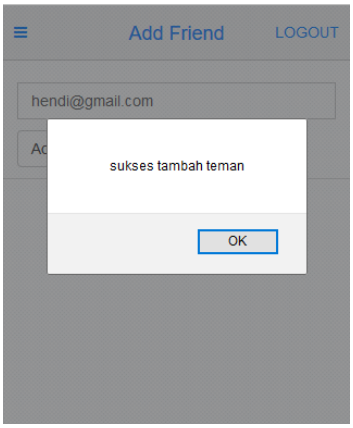
Halaman permintaan pertemanan akan memunculkan notifikasi jika adanya permintaan pertemanan yang masuk. Kita bisa memilih apakah kita mau menerima pertemanan atau menolak pertemanan. Setelah menerima pertemanan maka secara otomatis nama teman yang kita terima akan masuk ke halaman utama. Halaman ini bisa dilihat pada Gambar 7.




Gambar 7
Halaman Permintaan Pertemanan

*F. Halaman Tambah Teman*

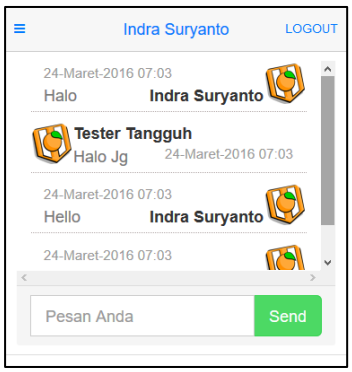
Halaman tambah teman ini bisa digunakan untuk mencari teman yang telah terdaftar pada sistem untuk ditambahkan di daftar teman. Cara menambahkan teman adalah memasukan email teman yang mau ditambah atau klik tombol add maka sistem akan secara otomatis meminta pertemanan.Halaman tambah teman ini bisa dilihat di Gambar 8.




Gambar 8
Halaman Tambah Teman

*G. Halaman Chat*

Halaman chat adalah halaman dimana kita akan melakukan chat dengan teman kita. Pesan yang kita kirimkan akan di enkripsi dengan menggunakan metode AES sehingga selama pengiriman, pesan yang kita kirimkan akan aman karena telah terenkripsi. Pada saat pesan masuk maka pesan yang semula terenkripsi itu akan didekripsi sebelum ditampilkan ke pengguna untuk dibaca. Tampilan halaman ini dapat dilihat pada Gambar 9.

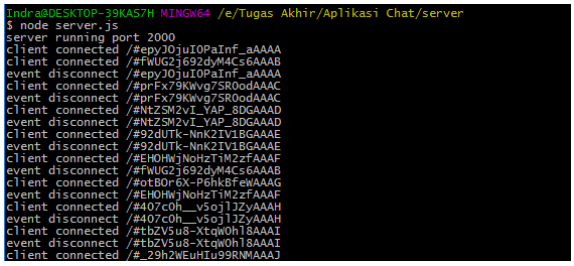



Gambar 9
Halaman *Chat*

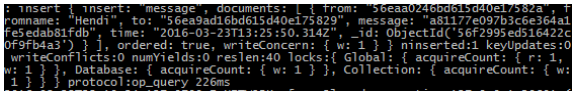
## 5.2. Pengujian Aplikasi

Hasil pembuatan aplikasi menggunakan server NodeJS berjalan dengan baik karena setiap akses akan terlihat di server dan pertukaran data juga berjalan dengan baik. Proses yang berjalan di server dapat dilihat pada Gambar 10. Pertukaran data di MongoDB juga berjalan dengan baik, setiap

data yang masuk akan tersimpan dengan baik berupa format json. Gambaran berjalannya penyimpanan data di MongoDB dapat dilihat pada Gambar 11.




Gambar 10
Sistem NodeJS




Gambar 11
Sistem Penyimpanan Data *Chat* pada MongoDB

Pada pengujian aplikasi *Chat* memerlukan NodeJs dan MongoDB untuk aktif, jika tidak maka aplikasi tidak akan berjalan karena tidak bisa terhubung ke sistem. Proses enkripsi dan dekripsi juga berjalan dengan baik tanpa mengganggu kemampuan pengiriman dan penerimaan data dari aplikasi. Proses enkripsi dan dekripsi pesan panjang memang lebih memerlukan waktu lama karena pesan akan dipecah terlebih dahulu menjadi beberapa bagian yang terdiri dari 32 huruf atau angka dan dienkripsi baru kemudian digabungkan lagi. Proses pemecahaman kata ini tidak melambatkan waktu pertukaran data karena tidak memerlukan alat dengan spesifikasi yang tinggi. Data yang masuk ke database berupa pesan enkripsi seperti yang bisa dilihat pada Gambar 11.

### 5.3. Hasil Pengujian

Berdasarkan hasil dari pengujian dengan mengirimkan pesan yang merupakan kombinasi dari berbagai jumlah kata, hasil yang keluar sesuai dengan yang diharapkan. Pesan yang dikirim dan dibaca berupa *plaintext* yang sama tetapi pesan yang masuk ke *database* berupa *chipertext*. Panjang *chipertext* yang dihasilkan akan sesuai dengan panjang pesan yang dikirimkan. Hasil pengujian bias diliat pada tabel 4

Tabel 4. Tabel Pengujian

| No | Pesan Terkirim | Pesan Di Database (Chipertext) | Pesan Terbaca |
|---|---|---|---|
| 1 | Tes | f30fe63e08beea8a87e05218f2128b83 | Tes |
| 2 | minta nomor hp | c2f9954567fd36277ca8e0558016ce97 | minta nomor hp |
| 3 | Apa kabar kawan? | 11012993de8d34205a9e04e01c71a9de | Apa kabar kawan? |
| 4 | 081234567890 | a28234c9a5a2d4b5cd5f283fd14058a7 | 081234567890 |
| 5 | maaf menggan ggu waktunya untuk pengujian tugas akhir :D | 6981748e4fd0ba845500ada7341d93f68fd7923f0351c39ca0357e52eeeb76682f82351bfb5f339d46a2de38ea10d3c13f95dcd76129ed8e21f770e6c1b4f24e | maaf menggang gu waktunya untuk pengujian tugas akhir :D |
| 6 | Apaan si AES itu? | 70cdcc73790de26b40bdb9102a6141cbebfccb09ff6b0ae22252338097f3df9a | Apaan si AES itu? |
| 7 | apa lagi ya? | 1bb58bad0df17c105fb0294c536d5251 | apa lagi ya? |
| 8 | bingung mo nulis apa | 0c2b90cd788de68b8e35d7684cb67c1ec5e7226baadb0acbb61da9857222c2f6 | bingung mo nulis apa |
| 9 | sudah berapa | 3589ab16667c07de0b52bcbc13d43723 | sudah berapa |
| 10 | apakah aplikasinya dapat berjalan dengan lancar? | 6d3a25bc402d7f4133a088da4d5321a6a1e391758d62299e76837c0317f470f26c754b6f038b47cd4f6988b749e63886 | apakah aplikasinya dapat berjalan dengan lancar? |

## 6. KESIMPULAN DAN SARAN

### 6.1. Kesimpulan

Berdasarkan hasil percobaan pengembangan aplikasi *chat messenger* dengan Metode *Advanced Encryption Standard* (AES) pada *smartphone* dapat diambil kesimpulan sebagai berikut :

1. Dari hasil pembuatan sistem enkripsi dan dekripsi pesan diperlukan sebuah beberapa modul pendukung seperti base64 yang digunakan untuk menstandarisasi pesan yang masuk sehingga bisa diubah menjadi nilai hexadesimal termasuk nilai spasi. Sehingga pesan yang dienkripsi pada saat dikirimkan dan didekripsi pada saat diterima bisa sama.
2. Perangkat telepon genggam dapat berkomunikasi secara realtime menggunakan nodeJS. Perangkat akan terhubung satu sama lain dengan menggunakan ID sementara yang disebut SocketID. SocketID akan menjadi identitas pada server pada saat login pada aplikasi sehingga pesan yang dikirimkan bisa tertuju pada penerima yang dituju dengan alamat SocketID yang ada pada server
3. Pada saat pengimplementasian sistem AES diperlukan beberapa penyesuaian sehingga pesan yang dikirimkan bisa memiliki nilai *byte* yang diperlukan untuk melakukan proses enkripsi. Pesan yang memiliki nilai *byte* yang kurang dari 128 *byte* perlu ditambahkan dengan spasi untuk mengisi nilai yang kurang. Pesan yang memiliki nilai lebih dari 128 *byte* perlu dibuatkan sistem untuk membagi pesan menjadi beberapa bagian dengan nilai 128 *byte* untuk proses enkripsi dan dekripsi kemudian digabungkan kembali saat selesai dienkripsi atau didekripsi. Proses enkripsi dan dekripsi pesan akan dilakukan pada aplikasi sehingga selama proses transmisi pesan, pesan yang yang dikirimkan merupakan *chipertext* yang walaupun disadap tidak bisa dibaca oleh orang yang menyadap tersebut.

### 6.2. Saran

Adapun saran terhadap aplikasi yang telah dibangun sebagai berikut :

1. Pengembangan fitur baru untuk melengkapi aplikasi *chat messenger* seperti fitur kirim file, foto, penghapusan pesan, dan *voice call*.
2. Penerapan kriptografi AES ini juga bisa dikembangkan ke media pertukuran data lain seperti untuk enkripsi pengiriman file atau foto.
3. Membuat fitur untuk pengembalian password jika sewaktu-waktu user lupa password.
4. Pembuatan fitur untuk mengajak teman untuk menggunakan aplikasi *chat messenger* dengan pesan singkat, email, dan media sosial lainnya.
5. Membuat autentifikasi melalui email untuk mengecek email pengguna yang dimasukan.

## DAFTAR PUSTAKA

[1] Kemp, Simon. 2015. “Digital, Social, Mobile in Apac in 2015” http://wearesocial.com/uk/ blog/2015/03/digital-social-mobile-apac-2015, diakses tanggal 7 Oktober 2015

[2] Nielson. 2014. “Blackberry messenger aplikasi chat paling banyak dipilih di indonesia” http://www.nielsen.com/id/en/press-room/2014/blackberry-messenger-aplikasi-chat-paling-banyak-dipilih-di-indonesia.html, diakses tanggal 7 Oktober 2015

[3] Wijayanto, Bakhtiar. 2012. Merancang dan Membangun Aplikasi Chat Messenger Untuk Android. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.

[4] Widodo, Joko Tri Susilo. 2014. Implementasi Algortima Kriptografi AES 128 Bit Sebagai Pengaman SMS pada Smartphone Berbasis Android. Yogyakarta: Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.

[5] Nuryantin, Dewi M, 2014. Advanced Encryption Standard (AES). Semarang: Universitas Aki Semarang
[6] Sto. 2007. Wireless Kung Fu : Networking & Hacking. Jakarta: Jasakom.
[7] Ariyus, Dony. 2008. Pengantar Ilmu Kriptografi. Yogyakarta: Penerbut Andy
[8] Yuniati, Voni, et al. 2009. "Enkripsi dan dekripsi dengan algoritma AES 256 untuk semua jenis file". Jurnal Informatika, Volume 5 No 1, April 2009.
[9] Jubilee Enterprise. 2010. Step By Step : Ponsel Android. Jakarta: Elex Media Komputindo.
[10] Jusliman, Agung. 2014. Sistem Aplikasi Travel dengan Angularjs & Codeigniter. Yogyakarta: Lokomedia
[11] Nodejs. 2015. https://nodejs.org