PENINGKATAN KEAMANAN APLIKASI CHAT MOBILE MELALUI ANALISIS DAN IMPLEMENTASI ADVANCED ENCRYPTION STANDARD (AES)

Mohamad Iqbal Suriansyah dan Muhammad Ihsan Rahman

Universitas Pakuan
Jl. Pakuan, RT.02/RW.06, Tegallega, Bogor Tengah, Jawa Barat 16129
mohamad.iqbal@unpak.ac.id, mihsanrahman19@gmail.com

ABSTRAK

Aplikasi pesan instan komersial sering kali menghadapi tantangan serius dalam menjaga keamanan data. Penelitian ini bertujuan meningkatkan keamanan komunikasi di lingkungan Universitas Nusa Bangsa dengan mengimplementasikan Algoritma Advanced Encryption Standard (AES) pada aplikasi chat mobile. Metode prototyping digunakan untuk mengembangkan aplikasi ini, meliputi metode prototyping, pengujian kompabilitas dan usabilitas . Hasil pengujian menunjukkan bahwa implementasi AES berhasil melindungi komunikasi pesan, meningkatkan privasi, dan menjaga integritas data pengguna. Algoritma AES menunjukkan performa yang sangat baik dengan waktu enkripsi dan dekripsi yang cepat, yaitu 0,23 detik dan 0,13 detik berturut-turut. Dengan demikian, komunikasi antar mahasiswa, dosen, dan tenaga pendidik terjamin keamanannya dan terjaga privasinya.

Kata Kunci: *Enkripsi, Algoritma AES, Keamanan Data, Aplikasi Chat Mobile.*

PENDAHULUAN

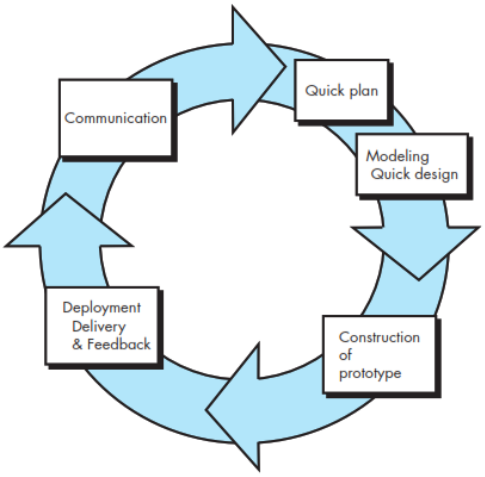
Internet Relay Chat (IRC), atau yang lebih dikenal sebagai chat, adalah sebuah platform komunikasi online yang memungkinkan terjadinya percakapan secara langsung (real-time) antara dua orang atau bahkan lebih melalui pertukaran pesan teks di jaringan internet [1]. Aplikasi pesan instan komersial sering kali rentan terhadap ancaman keamanan data karena kemungkinan adanya penyalahgunaan informasi [2]. Salah satu solusi untuk mengatasi risiko tersebut adalah dengan menerapkan Algoritma Advanced Encryption Standard (AES), yang telah diadopsi oleh National Institute of Standards and Technology (NIST) sejak tahun 2001 dan kini banyak digunakan untuk melindungi data elektronik [3]. AES beroperasi dengan memanfaatkan kunci enkripsi simetris yang memiliki panjang 128, 192, atau 256 bit. Penggunaan kunci ini memungkinkan AES mencapai tingkat keamanan yang sangat tinggi karena data yang akan dienkripsi akan dipecah menjadi blok-blok berukuran 128 bit, dan setiap blok tersebut akan dienkripsi secara terpisah. Semakin panjang kunci yang digunakan, semakin kompleks proses enkripsi dan dekripsi, sehingga semakin sulit bagi pihak yang tidak berwenang untuk membobol data tersebut [4].

Implementasi algoritma enkripsi memiliki peran krusial dalam menjaga keamanan data sensitif dari berbagai ancaman siber, seperti peretasan dan pencurian data. Selain itu, enkripsi juga memastikan kepatuhan terhadap standar keamanan yang berlaku, sekaligus membangun kepercayaan pengguna bahwa privasi data mereka terlindungi dengan baik [5]. Tanpa adanya perlindungan enkripsi, data pengguna menjadi sangat rentan terhadap akses yang tidak sah. Hal ini dapat berakibat fatal, menyebabkan kerugian finansial, kerusakan reputasi, bahkan masalah hukum baik bagi pengembang aplikasi maupun pengguna itu sendiri [6].

METODE PENELITIAN

Penelitian ini mengadopsi metode prototyping, yaitu sebuah pendekatan pengembangan perangkat lunak yang berfokus pada pembuatan model awal sistem untuk diuji oleh pengguna. Umpan balik yang didapatkan kemudian digunakan untuk menyempurnakan dan menyesuaikan sistem, sehingga produk akhir lebih sesuai dengan kebutuhan pengguna. Pendekatan ini efektif dalam mengurangi kesalahan dengan

mengandalkan proses iterasi berkelanjutan berdasarkan evaluasi langsung [7].



Gambar 1. *Prototyping oleh Pressman[8]*

Gambar 1 merupakan tahapan-tahapan dari metode *prototyping* yang memiliki empat tahap. Tahap tersebut adalah:

a. Communication

Komunikasi berperan penting dalam proses pengembangan *Prototype* aplikasi chat ini. Penelitian ini melibatkan diskusi berkala dengan mahasiswa, dosen, serta tenaga pendidik lainnya untuk menggali kebutuhan dan harapan mereka terhadap aplikasi tersebut.

b. Quick Plan and Modeling Quick Design

Tahap perencanaan dan perancangan awal dilakukan dengan cepat untuk mengidentifikasi fitur-fitur utama yang akan diimplementasikan pada aplikasi. Teknik brainstorming dan kajian literatur digunakan untuk mengumpulkan ide-ide kreatif serta praktik terbaik dalam desain aplikasi chat, sekaligus memberikan gambaran visual awal mengenai cara kerja aplikasi tersebut.

c. Construction of Prototype

Setelah tujuan dan desain sistem dirumuskan dengan jelas, *Prototype* aplikasi mulai dibangun. Tahap ini mencakup pembuatan antarmuka pengguna, implementasi fitur-fitur utama seperti chat pribadi dan grup, serta integrasi sistem pendaftaran dan autentikasi. Pengujian awal dilakukan untuk memastikan setiap komponen berfungsi dengan baik sebelum *Prototype* tersebut diberikan kepada pengguna untuk diuji coba.

d. Deployment Delivery and Feedback

Prototype tersebut kemudian diuji coba oleh pengguna untuk mendapatkan evaluasi mengenai tampilan dan fungsionalitasnya. Kuesioner dengan skala penilaian 1 hingga 5 disebarkan kepada mahasiswa, dosen, dan staf untuk mengumpulkan umpan balik mereka. Selain itu, pengujian black box juga dilakukan untuk memastikan bahwa fungsionalitas aplikasi sesuai dengan spesifikasi yang telah ditentukan, tanpa perlu melihat struktur internalnya.

HASIL DAN PEMBAHASAN

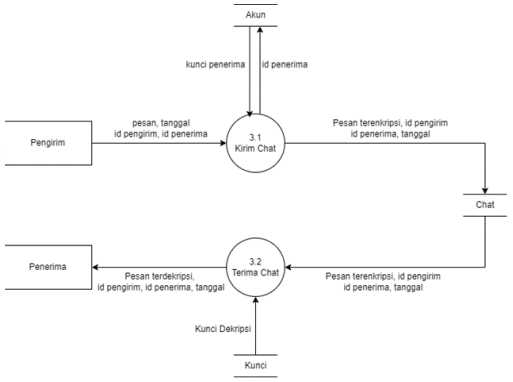
Rancangan Aplikasi Chat dan Implementasi Algoritma AES

Tahap perancangan dilakukan untuk mengidentifikasi fitur-fitur utama yang esensial dalam aplikasi chatting. Gambar 2 menggambarkan pengembangan sistem ini dengan sebuah Context Diagram. Diagram tersebut menjelaskan bahwa pengirim dapat mengirimkan pesan terenkripsi dengan memasukkan pesan berupa teks. Setelah diproses oleh sistem, penerima akan menerima pesan yang telah didekripsi menjadi pesan yang dapat dibaca.

Gambar 2. *Data Flow Diagram (DFD) Aplikasi Chat*

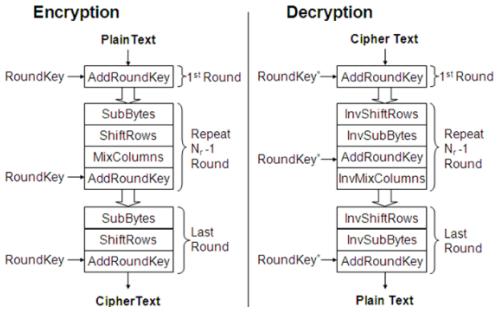
Gambar 3 dibawah menyajikan diagram yang menggambarkan alur pengiriman dan penerimaan pesan dalam aplikasi chat yang memanfaatkan enkripsi AES untuk menjamin keamanan data. Prosesnya bermula dari pengirim yang mengirimkan pesan. Pesan tersebut kemudian dienkripsi menggunakan kunci enkripsi penerima yang diambil dari akun penerima, beserta informasi tambahan seperti ID pengirim, ID penerima, dan tanggal pengiriman. Pesan terenkripsi ini lalu dikirimkan ke penerima. Penerima kemudian menggunakan kunci dekripsi yang sesuai untuk mengembalikan pesan ke bentuk aslinya. Informasi seperti ID pengirim, ID penerima, dan tanggal tetap terjaga selama proses ini. Akhirnya, pesan

yang telah berhasil didekripsi ditampilkan kepada penerima.



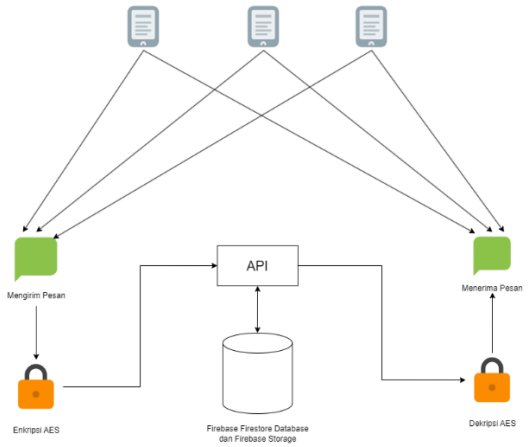
Gambar 3. *Alur pengiriman dan penerimaan pesan*

Pada proses enkripsi, data asli diubah menjadi data terenkripsi melalui serangkaian transformasi yang kompleks menggunakan kunci enkripsi yang telah ditentukan. Proses dekripsi, di sisi lain, membalikkan transformasi ini untuk mengembalikan data terenkripsi ke bentuk aslinya dengan menggunakan kunci yang sesuai. Transformasi-transformasi ini melibatkan operasi seperti substitusi byte (mengganti setiap byte dengan byte lain berdasarkan tabel substitusi), pergeseran baris (menggeser setiap baris dalam blok data secara siklis), pencampuran kolom (mengubah setiap kolom dalam blok data menggunakan operasi matematika), dan penambahan kunci bulat (menggabungkan data dengan bagian dari kunci enkripsi). Semua operasi ini diulang beberapa kali (disebut ronde) untuk mencapai tingkat keamanan yang maksimal. Ilustrasi proses enkripsi dan dekripsi dalam algoritma AES tersebut dapat terlihat pada Gambar 4 dibawah.



Gambar 4. *Proses Enkripsi dan Dekripsi AES (Kho et al., 2020).*

Rancangan arsitektur jaringan pada sistem pesan terenkripsi menggunakan Firebase dapat dilihat pada Gambar 5. Proses dimulai dengan pengiriman pesan yang dienkripsi menggunakan AES (Advanced Encryption Standard) sebelum dikirim melalui API. Pesan terenkripsi tersebut kemudian disimpan dalam Firebase Firestore Database dan Firebase Storage. Pada saat pesan diterima, API mengakses database untuk mendapatkan pesan yang terenkripsi, yang kemudian didekripsi menggunakan AES sebelum akhirnya diterima oleh penerima pesan.



Gambar 5. *Rancangan arsitektur jaringan*

Implementasi Algoritma AES

Penerapan enkripsi AES diintegrasikan secara mulus ke dalam proses pengiriman pesan, sehingga enkripsi terjadi secara otomatis setiap kali pengguna mengirimkan pesan. Keamanan komunikasi ini dijamin dengan penggunaan kunci enkripsi yang unik dan kuat yang dapat dilihat pada kode pemograman berikut.

```
import 'package:encrypt/encrypt.dart' as
    encrypt;

class EncryptionHelper {
// Define a fixed key and IV
static final key =
    encrypt.Key.fromUtf8('12345678901
    23456789012345678901 2'); // 16
    characters for AES-128
static final iv =
    encrypt.IV.fromUtf8('123456789012
    3456'); // 16 characters for IV
```

```
static String encryptMessage(String
    message) {
final encrypted =
    encrypter.encrypt(message, iv: iv);
return encrypted.base64;
}

static String decryptMessage(String
    encryptedMessage) {
final decrypted =
    encrypter.decrypt64(encryptedMessa
    ge, iv: iv);
return decrypted;
}
}
```

Kode program ini mendemonstrasikan enkripsi dan dekripsi menggunakan algoritma AES dengan mode operasi CBC (Cipher Block Chaining) dalam bahasa Dart. Dengan memanfaatkan package `encrypt`, kelas `EncryptionHelper` mendefinisikan kunci (16 karakter untuk AES-128) dan vektor inisialisasi (IV, juga 16 karakter). Kelas ini menyediakan dua metode statis: `encryptMessage`, yang mengenkripsi pesan teks menjadi format base64, dan `decryptMessage`, yang mendekripsi pesan terenkripsi dari format base64 kembali ke teks aslinya. Penggunaan kunci dan IV yang tetap dalam proses enkripsi dan dekripsi memastikan keamanan dan integritas data, sehingga data yang terenkripsi terlindungi dengan baik dan dapat dipulihkan dengan benar saat didekripsi.

Pengujian Dengan AES dan Tanpa AES

Berikut adalah hasil pengujian performa aplikasi chat, dengan dan tanpa implementasi algoritma AES.

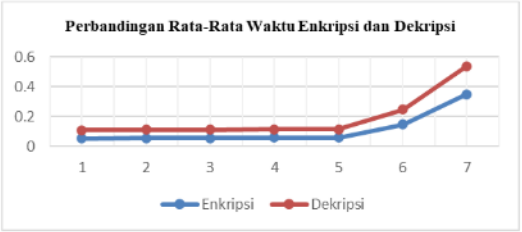
Gambar 6. *Contoh Pesan Terenkripsi*

Gambar 6 menampilkan contoh pesan yang telah dienkripsi. Pesan asli yang mungkin berisi informasi kompleks seperti paragraf panjang, simbol khusus, dan angka, diubah menjadi rangkaian karakter acak yang tidak memiliki makna tanpa kunci dekripsi yang sesuai. Di sisi lain, Gambar 7 menunjukkan bagaimana pesan tanpa enkripsi dapat dibaca dengan jelas oleh siapa saja, meningkatkan risiko penyalahgunaan informasi yang dapat merugikan pihak lain.

Gambar 7. *Contoh Pesan Tidak Terenkripsi*

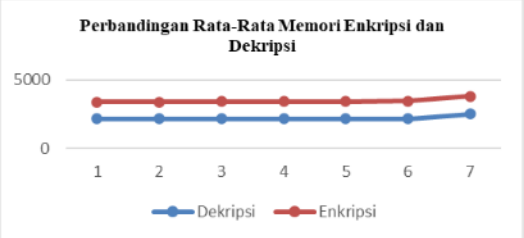
Pengujian Performa Algoritma AES

Algoritma AES menunjukkan performa yang mengesankan dengan waktu enkripsi rata-rata hanya 0,23 detik dan waktu dekripsi 0,13 detik terlihat pada Gambar 8. Kecepatan ini membuktikan bahwa AES mampu melakukan enkripsi dan dekripsi secara efisien dalam waktu yang relatif singkat.



Gambar 8. *Waktu Proses Enkripsi (dalam detik)*

Meskipun penggunaan memori untuk proses enkripsi dan dekripsi AES cukup besar, mencapai sekitar 897 kB untuk plainteks berukuran 512 bit, performanya dalam hal kecepatan tetap unggul dibandingkan dengan beberapa algoritma lain.



Gambar 9. *Penggunaan memori selama enkripsi dan dekripsi.*

Gambar 9 menunjukkan perbedaan yang signifikan dalam penggunaan memori antara proses enkripsi dan dekripsi. Penggunaan memori untuk dekripsi jauh lebih besar dibandingkan dengan enkripsi. Hal ini kemungkinan disebabkan oleh kompleksitas algoritma dan struktur data yang digunakan dalam proses dekripsi.

Pengujian Kompatibilitas

Pengujian kompatibilitas menunjukkan bahwa aplikasi ini berhasil berjalan dengan lancar di berbagai perangkat smartphone dan platform dengan beragam versi sistem operasi Android. Pengujian dilakukan pada beberapa model populer seperti Redmi Note 13, Samsung A34, Vivo Y21, Vivo Y28, dan Poco M6. Hasilnya, aplikasi berfungsi tanpa hambatan pada semua perangkat tersebut, termasuk dalam hal performa, responsivitas, dan tampilan antarmuka di berbagai perangkat dan versi sistem operasi Android [10].

Pengujian Usabilitas

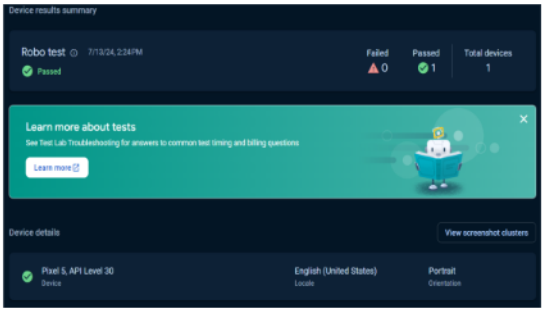
Hasil pengujian usabilitas dilingkungan internal terhadap fitur enkripsi/dekripsi pesan menunjukkan bahwa sebagian besar pengguna memberikan umpan balik positif [11]. Mayoritas halaman mendapatkan penilaian "baik" dan "sangat baik", terutama Halaman Register (65,6% baik, 31,8% sangat baik), Halaman Login (40,9% baik, 36,4% sangat baik), dan Halaman Kelola Akun (59,1% baik, 40,9% sangat baik). Hanya sebagian kecil halaman yang menerima penilaian "cukup" atau "buruk", seperti Halaman Admin Panel (13,6 cukup) dan Halaman Personal Chat (9,1% cukup).

Secara keseluruhan, evaluasi ini menunjukkan bahwa pengguna merasa fitur keamanan pesan mudah digunakan dan dipahami, mengindikasikan pengalaman pengguna yang positif dengan beberapa area yang memerlukan sedikit perbaikan. Hasil kuesioner dari 25 responden dapat dilihat pada dibawah.

Tabel 1. *Pengujian Usabilitas*

No	Halaman	Penilaian				
		1 = Sangat Buruk	2 = Buruk	3 = Cukup	4 = baik	5 = sangat baik
1	Halaman *Register*	-	-	4,5%	65,6%	31,8%
2	Halaman *Login*	-	4,5%	18,2%	40,9%	36,4%
3	Halaman Tenaga Pendidik	-	-	4,5%	59,1%	36,4%
4	Halaman Grup	-	-	4,5%	59,1%	36,4%
5	Halaman *Chat* Grup	-	-	9,1%	59,1%	31,8%
6	Halaman Admin Panel	-	-	13,6%	50%	36,4%
7	Halaman Kelola Akun	-	-	-	59,1%	40,9%
8	Halaman Kelola Grup	-	-	9,1%	59,1%	31,8%
9	Halaman *List* Kontak	-	-	-	40,9%	59,1%
10	Halaman *Profile*	4,5%	-	18,2%	36,4%	40,9%
11	Halaman *List Chat*	-	-	13,6%	54,5%	31,8%
12	Halaman *Personal Chat*	4,5%	-	9,1%	54,5%	31,8%
13	Halaman *Sign Out*	-	-	4,5%	59,1%	36,4%

Pengujian Firebase Test Lab



Gambar 10. *Firebase Test Lab*

Firebase Test Lab, melalui serangkaian tes fungsional, performa, dan kompatibilitas pada berbagai perangkat virtual, berhasil mengidentifikasi potensi bug dan masalah dalam aplikasi chat. Hasilnya, aplikasi tersebut dinyatakan "LULUS" setelah berhasil melewati semua pengujian tanpa kendala berarti.

PENUTUP

Implementasi algoritma Advanced Encryption Standard (AES) dalam aplikasi chat berbasis mobile) telah terbukti berhasil meningkatkan kualitas dan keamanan komunikas. Dengan menggunakan enkripsi AES, pesan yang dikirim dan diterima terjamin kerahasiaannya, melindungi data sensitif dari akses pihak ketiga yang tidak berwenang. Pesan yang dienkripsi

mencakup teks, nomor, dan simbol, sehingga informasi sensitif terlindungi dengan baik. Hal ini memastikan bahwa komunikasi antar mahasiswa, dosen, dan tenaga pendidik tetap aman dan privat. Keberhasilan ini tercermin dalam efisiensi komunikasi yang meningkat, di mana pesan dapat diterima dan dibalas secara real-time, memfasilitasi responsivitas yang lebih baik dan meningkatkan keterlibatan seluruh anggota kampus.

DAFTAR PUSTAKA

[1] Mayuuf, H. H., & Al-Ghizzy, M. J. D, *"Stylistic Features of Internet Relay Chat IRC as a Medium of Computer-Mediated Communication. International Journal of Linguistics",* Literature and Translation*, 5*(12), 161-167, 2022.

[2] Juniarmi, I, *"Analisis Keamanan Data pada Aplikasi Chatting Menggunakan Enkripsi End-to-End"*, Technologia Journal, 1(2), 30-38, 2024.

[3] Asri, S., Peryanto, T., & Sakti, E. M. S, *"Perbandingan Implementasi Algoritma Aes Dalam Pemrograman Jaringan Dan Analisis Algoritma Enkripsi Untuk Pengamanan Komunikasi Jaringan"*, Jurnal Ilmiah Teknik Informatika (TEKINFO), 25(1), 80-87, 2024.

[4] Azhari, M., Mulyana, D. I., Perwitosari, F. J., & Ali, F, *"Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)"*, Jurnal Pendidikan Sains dan Komputer, 2(01), 163-171, 2022.

[5] Aprizald, A., Hasan, M. A., & Setiawan, D, *"Aplikasi Keamanan Data Berbasis Web Menggunakan Algoritma AES 128 Untuk Enkripsi Dan Dekripsi Data"* JEKIN-Jurnal Teknik Informatika, 2(2), 85-95, 2022.

[6] Wulandari, I. W., & Hwihanus, H. *"Peran Sistem Informasi Akuntansi Dalam Pengaplikasian Enkripsi Terhadap Peningkatan Keamanan Perusahaan"*, Jurnal Kajian Dan Penalaran Ilmu Manajemen, *1*(1), 11-25, 2023.

[7] Maulana, H., Kasmawi, K., & Enda, D. *"Buku Penghubung Berbasis Android Menggunakan Metode Prototyping"*, Jurnal Teknik Informatika dan Sistem Informasi, 6(3), 2020.

[8] D.Purnomo, *"Model Prototyping Pada Pengembangan Sistem Informasi"*, JIMP-Jurnal Informatika Merdeka Pasuruan, vol. 2, pp. 54-61, 2017.