Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

ENKRIPSI DATA MENGGUNAKAN RSA & AES PADA APLIKASI *INSTANT MESSAGING* BERBASIS *MOBILE*

Yanuar Bimantoro[1], Ratih Titi Komala Sari[2]

[1,2]Informatika, Fakultas Teknologi Komunikasi & Informatika
[1,2]Universitas Nasional
[1,2]Jl. Sawo Manila, Jakarta Selatan, Jakarta 12520
E-mail: [1]yanuarbimantoro@gmail.com, [2]ratih.titi@civitas.unas.ac.id

Artikel:
Diterima: 06 Desember, 2021
Direvisi: 31 Desember, 2021
Diterbitkan: 07 Januari, 2022

***Alamat Korespondensi:**
yanuarbimantoro@gmail.com

ABSTRACT

Currently communication can be through various media, one of the most favorite is the instant messaging application because it presents complete features and can be used anywhere [1]. But digital data is vulnerable to theft, one example is when an application sends data to a server can be intercepted or the server is hacked [2]. Therefore required data security system one of which is encryption [3]. The research aims to implement encryption for data security using AES and RSA algorithms combined [4]. The method to be used is that AES will encrypt the original data and then the resulting AES key will be encrypted with RSA. Based on the results of combined testing these two algorithms can process data in the form of text and images in milisekon time and can still secure the original data.

Keywords: *Encryption, Instant Messaging, RSA, AES*

ABSTRAK

Saat ini komunikasi dapat melalui berbagai media, salah satu yang paling favorit adalah aplikasi instant messaging karena menghadirkan fitur yang lengkap serta dapat digunakan dimana saja[1]. Tetapi data digital rentan terhadap pencurian, salah satu contohnya adalah seperti pada saat aplikasi mengirimkan data ke server dapat disadap ataupun server diretas [2]. Oleh karena itu dibutuhkan sistem pengamanan data salah satunya adalah enkripsi[3]. Penelitian ini bertujuan untuk mengimplementasikan enkripsi untuk pengamanan data dengan menggunakan algoritma AES dan RSA yang digabungkan[4]. Metode yang akan digunakan adalah AES akan mengenkripsi data asli lalu kunci AES yang dihasilkan akan di enkripsi dengan RSA. Berdasarkan hasil pengujian gabungan kedua algoritma ini dapat memproses data berupa teks dan gambar dalam waktu milisekon serta tetap dapat mengamankan data asli.

Kata Kunci: *Enkripsi, Pesan Instan, RSA, AES*

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

I. PENDAHULUAN

Seiring perkembangan teknologi yang semakin pesat dari masa ke masa, memberikan dampak yang besar bagi kehidupan manusia di berbagai aspek. Salah satunya adalah aspek komunikasi, sekarang sudah tidak perlu lagi mengirimkan pesan dalam bentuk surat, ataupun teknologi sebelumnya yaitu pengiriman pesan singkat atau biasa disebut sms, serta telepon biasa yang menggunakan kartu sim yang membutuhkan biaya mahal. Saat ini dengan adanya teknologi internet kita dapat berkomunikasi dengan orang lain di belahan bumi lainnya secara langsung atau *realtime* [1].

Oleh karena itu dibutuhkanlah sebuah wadah yang dapat menampung atau menyediakan sarana komunikasi tersebut berupa aplikasi *instant messaging* atau pesan instan yang berbasis mobile yang dapat dibawa kemana-mana sehingga komunikasi tidak terputus selama terhubung di jaringan internet. Serta memiliki berbagai macam fitur seperti *video call*, pengiriman gambar dan lain sebagainya [5].

Tetapi setiap data digital yang dikirimkan melalui jaringan internet tidak dapat dipastikan keamanannya seperti pada saat pengiriman data ke server ataupun server diretas, hal itu menyebabkan masalah privasi pengguna yang tidak ingin datanya di gunakan oleh pihak ketiga yang tidak berkepentingan [2], [6].

Solusi terbaik yang dapat diberikan dalam menyelesaikan masalah ini adalah penggunaan *End-to-End Encryption* yaitu sebuah metode enkripsi saat mengirimkan pesan, dimana hanya pengguna yang bersangkutan yaitu pengirim dan penerima yang dapat mengubah kembali menjadi teks asli [3].

Berdasarkan penelitian yang telah dilakukan oleh A. Sukma Wijaya, D. T. Nugrahadi, M. I. Mazdadi, A. Farmadi, dan A. Rusadi dalam implementasi algoritma RSA pada aplikasi *instant messaging* menghasilkan waktu yang lumayan lama untuk proses enkripsi dan dekripsi untuk pesan dengan panjang maksimal 1000 karakter [1].

Penelitian serupa dilakukan oleh A. Aminudin, G. P. Aditya, dan S. Arifianto implementasi yang digunakan adalah algoritma RSA digabung dengan pembangkit kunci ESRKGS yang dirasa masih kurang optimal karena masih membutuhkan waktu yang lama walau menggunakan pesan dengan panjang maksimal 400 karakter [3].

Penelitian dari Y. Fatma, A. Hafid, dan H. O. Dani yang mengimplementasikan AES dan LSB untuk digunakan enkripsi gambar yang diubah terlebih dahulu dalam bentuk teks dengan panjang 5000 karakter kemudian dilakukan proses enkripsi dan dekripsi menghasilkan waktu yang cepat hanya dalam kisaran waktu 100 milisekon [8]. Tetapi penggunaan AES saja dinilai kurang karena harus mengirimkan kunci bersamaan dengan pesan enkripsi.

Oleh karena itu pada penelitian ini menggabungkan dua jenis algoritma kritografi yaitu algoritma RSA dengan AES untuk proses enkripsi dan dekripsi yang lebih cepat serta meningkatkan keamanan data pada aplikasi *instant messaging* [2], [6].

II. METODOLOGI

Metode yang digunakan dalam pembuatan aplikasi *instant messaging* adalah *Software Development Life Cycle* (SDLC) memiliki tahapan berikut.

2.1 Perencanaan

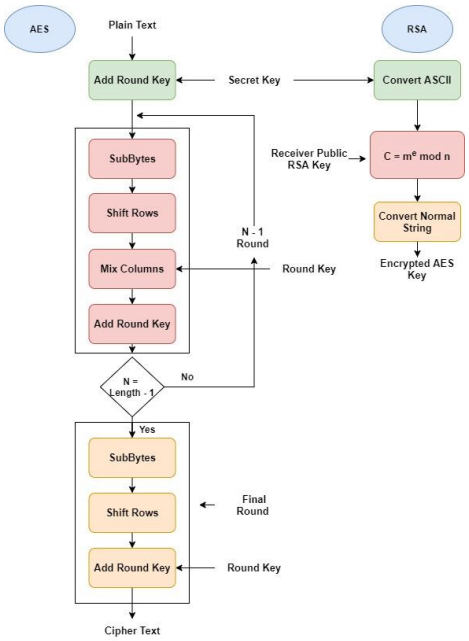
Untuk dapat mengetahui apa saja yang dibutuhkan serta fitur apa saja yang harus ada di dalam aplikasi nantinya. 2 aplikasi dijadikan referensi yaitu “WhatsApp” dan “Signal” karena memiliki fitur *end-to-end encryption* dan bertipe *instant messaging*. Dari hasil membandingkan kedua aplikasi tersebut di dapatkan fitur seperti enkripsi, *video call*, *voice call*, kirim teks, gambar, serta status *online* pengguna lainnya.

2.2 Desain

Desain yang dibuat dalam penelitian ini berupa desain enkripsi, deksripsi, alur aplikasi, struktur pesan asli, dan struktur pesan enkripsi.

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

2.2.1 Desain Enkripsi



Gambar 1. Alur proses enkripsi algoritma AES dengan RSA

Format pesan yang digunakan pada sistem seperti gambar 3 merupakan *plain text* yang akan diubah menjadi *cipher text* pada gambar 4. Proses enkripsi algoritma AES dengan RSA adalah sebagai berikut [3], [4], [9]:

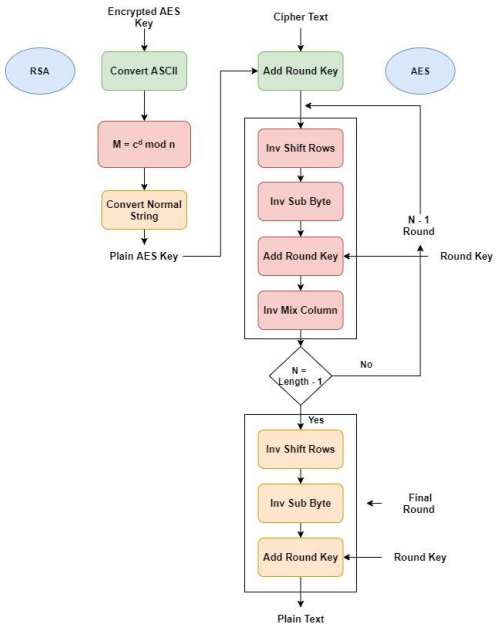
1. *Add Round Key*. Proses menggabungkan *plain text* dengan gerbang XOR.
2. Putaran (N) sebanyak N-1, dilakukan sebanyak sebanyak 10 kali untuk kunci 128 bit, 12 kali untuk kunci 192 bit, dan 14 kali untuk 256 bit [10].
   a) *SubBytes*. Mengubah setiap dua digit bilangan *hex* dari hasil proses pertama, di subtitusikan sesuai baris dan kolom tabel S-Box, misalnya *hex* A0 di subtitusikan ke baris A kolom 0 menjadi E0.
   b) *Shift Rows*. Menggeser setiap blok *byte* dalam baris ke kiri sejumlah n kolom, misalnya kolom ke-3 maka setiap *byte* dalam baris kolom tersebut akan bergeser 3 kali.
   c) *Mix Columns*. Setiap kolom dalam blok *byte* dikalikan dengan matriks pada persamaan 1.
   d) *Add Round Key*. Blok *byte* yang dihasilkan proses sebelumnya digunakan untuk menghasilkan *session key* seperti pada proses 1.
3. Proses Akhir. Proses ini hampir sama seperti pada proses ke-2 yang membedakan hanyalah proses *Mix Columns* yang dihapus yang dan juga tidak ada perulangan kembali.
4. Kunci AES yang telah dihasilkan diubah ke dalam bentuk numerik kode ASCII.
5. Dilakukan proses perhitungan seperti pada persamaan 2 menggunakan kunci publik penerima yang didapatkan saat bertukar kunci.
6. Lalu ubah numerik hasil perhitungan ke dalam bentuk normal string sesuai kode ASCII.

2.2.2 Desain Dekripsi

Urutan proses dekripsi dalam mengubah *cipher text* pada gambar 4 menjadi plain text gambar 3 sebagai berikut [3], [4], [9]:

1. Kunci AES yang sudah dienkripsi di ubah ke dalam bentuk numerik kode ASCII.
2. Bentuk numerik kode ASCII tersebut di konversikan menjadi bentuk numerik dari teks asli menggunakan persamaan 3.
3. Hasil perhitungan proses sebelumnya diubah kembali menjadi bentuk yang dapat dibaca oleh manusia. Lalu hasil kunci yang sudah di dekripsi digunakan untuk proses selanjutnya.
4. *Add Round Key*.
5. Putaran (N) sebanyak N-1:
   a) *Inverse Shift Rows*. Prosesnya mirip seperti pada saat enkripsi yang membedakannya hanyalah bergeser ke kanan.
   b) *Inverse SubBytes*. Subtitusi tiap *byte* menggunakan tabel kebalikan S-Box.
   c) Add Round Key
   d) Inverse Mix Column
6. Proses Akhir. Proses ini hampir sama seperti pada proses ke-2 yang membedakan hanyalah proses *Inverse Mix Columns* yang dihapus yang dan juga tidak ada perulangan kembali.

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469



Gambar 2. Alur proses dekripsi algoritma AES dengan RSA

2.2.3 Desain Struktur Pesan Asli

Struktur pesan asli sebelum di enkripsi ataupun sesudah di dekripsi dapat dilihat pada gambar 3, struktur ini berupa tipe data JSON (*Javascript Object Notation*) yang mudah untuk dimengerti baik manusia maupun mesin.

```
{
  "message": "string",
  "time": 0,
  "chat_id": "string",
  "uuid": "string",
  "user_id": "string",
  "message_status": 0,
  "message_type": "string",
  "sent": 0,
  "receiver_id": "string",
  "photo_url": "string",
  "call_type": "string",
  "call_state": "string",
  "call_action": "string"
}
```

Gambar 3. Struktur pesan asli

Penjelasan masing-masing dari struktur pesan asli pada gambar 3 sebagai berikut:

Tabel 1. Keterangan Struktur Pesan Asli

Deskripsi	Keterangan
Message	Pesan dari pengguna yang akan dikirimkan, jika berupa file gambar maka diubah ke dalam bentuk string dengan base64.
Time	Waktu saat ini dalam milisekon *epoch*
Chat Id	Id chat antar pengguna yang dihasilkan secara acak
Uuid	Id pesan yang akan dikirimkan, dihasilkan secara acak
User Id	Nomor telepon pengirim
Message Status	Bernilai 0 jika dikirim dan bernilai 1 jika diterima
Message Type	Tipe pesan yang dikirim, apakah teks, gambar, atau panggilan
Sent	Bernilai 0 jika masih pending dan akan bernilai 1 jika pesan berhasil terkirim
Receiver Id	Nomor telepon penerima
Photo Url	Link URL gambar pengirim, untuk kebutuhan panggilan
Call Type	Digunakan saat melakukan panggilan, bernilai video atau voice tergantung tipenya
Call State	Digunakan saat melakukan panggilan, bernilai *outgoing* atau *incoming*
Call Action	Digunakan saat melakukan panggilan untuk mendapatkan respon dari penerima, bernilai *accept* atau *reject*

2.2.4 Desain Struktur Pesan Enkripsi

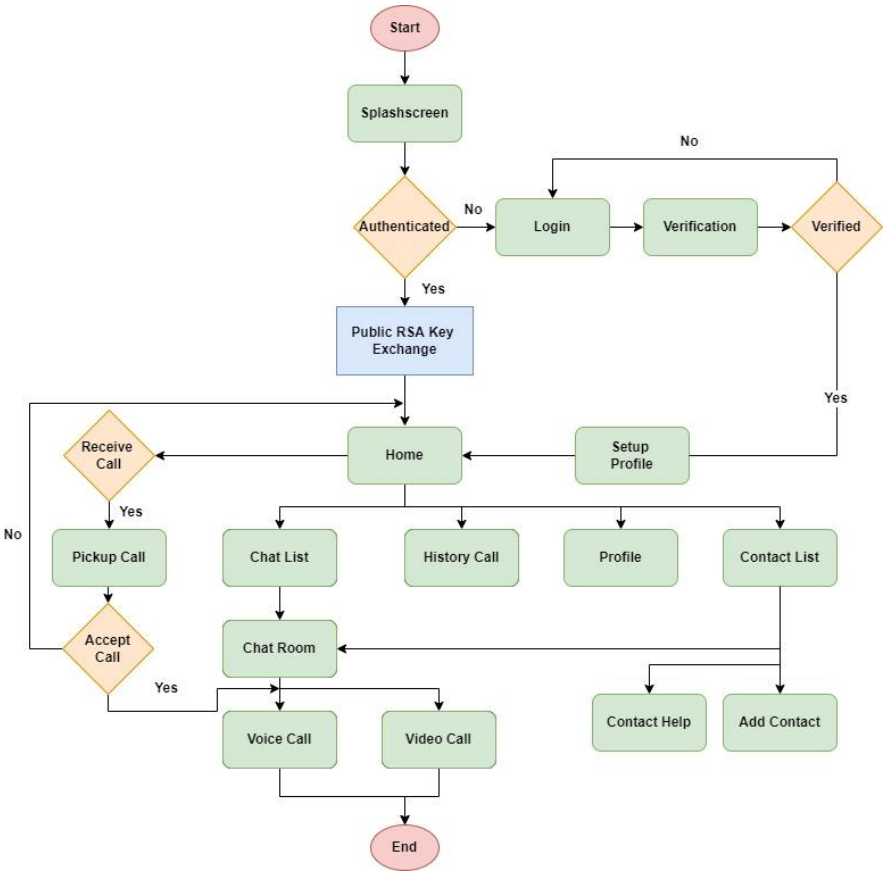
Merupakan struktur pesan asli yang diubah menjadi *cipher text*, detailnya dapat dilihat pada gambar 3. Pesan asli yang sudah di enkripsi terdapat pada key *cipher_text*.

```
{
  "cipher_text": "string",
  "key32": "string",
  "iv16": "string",
  "uuid": "string",
  "message_type": "string"
}
```

Gambar 4. Struktur pesan di enkripsi

Penjelasan masing-masing dari struktur pesan asli pada gambar 4 sebagai berikut:

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469



Gambar 5. Rancangan alur sistem

Tabel 2. Keterangan Struktur Pesan Enkripsi

Deskripsi	Keterangan
Cipher Text	Pesan asli pada gambar 3 yang sudah di enkripsi
Key32	Kunci AES yang sudah di enkripsi dengan RSA
Iv16	Sama seperti key32
Uuid	Id pesan yang akan dikirimkan, dihasilkan secara acak
Message Type	Tipe pesan yang dikirim, apakah teks, gambar, atau panggilan

2.2.5 Desain Alur Sistem

Rancangan alur sistem yang akan di implementasikan kedepannya dapat dilihat pada gambar 5, untuk penjelasan lebih detail sebagai berikut:

Tabel 3. Keterangan Alur Aplikasi

Deskripsi	Keterangan
Splashscreen	Tampilan yang pertama kali muncul pada sistem, berfungsi untuk mengecek apakah pengguna sudah sudah login atau tidak, jika belum maka akan diarahkan ke menu *login*
Login	Pengguna memasukkan nomor telepon
Verification	Pengguna memasukkan kode yang dikirimkan ke nomor pada halaman *login*
Setup Profile	Di halaman ini pengguna terlebih dahulu harus memasukkan nama dan foto profil yang nantinya akan ditampilkan ke kontak.
Home	Menu utama aplikasi yang di dalamnya terdapat tombol untuk melihat daftar pesan, riwayat telepon, profil, dan daftar kontak. Berfungsi untuk pertukaran kunci publik serta memproses saat menerima pesan
Chat Room	Menu detail pesan dengan pengguna yang lainnya, dimana pengguna dapat mengirimkan pesan teks, gambar, memulai *video call* dan *voice call*
Pickup Call	Menu untuk menampilkan bahwa pengguna menerima telepon, jika menerima maka akan diarahkan ke menu *video call* ataupun *voice call* sesuai jenisnya
Profile	Menu untuk melihat detail akun pribadi ataupun pengguna lainnya
Contact List	Menu untuk melihat daftar pengguna lainnya

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

Add Contact	Menambahkan kontak baru
Contact Help	Memberikan informasi mengenai kontak

2.3 Implementasi

Pembuatan aplikasi sesuai dengan desain akan di implementasi menggunakan *framework* Flutter dengan bahasa pemrograman Dart. Untuk server sebagai perantara pesan akan menggukan protokol MQTT, dan untuk menyimpan informasi kontak pengguna menggunakan Firebase [10].

2.4 Pengujian

Melakukan uji coba dengan sistem *User Acceptance Test* (UAT) untuk menguji seluruh fitur dalam aplikasi, menguji pesan yang diamankan apakah dapat mengirimkan pesan dari server yang sudah di retas dan analisa performa enkripsi serta dekripsi. Di tahap ini juga memperbaiki *error* atau kesalahan yang masih muncul pada tahap pengujian ini.

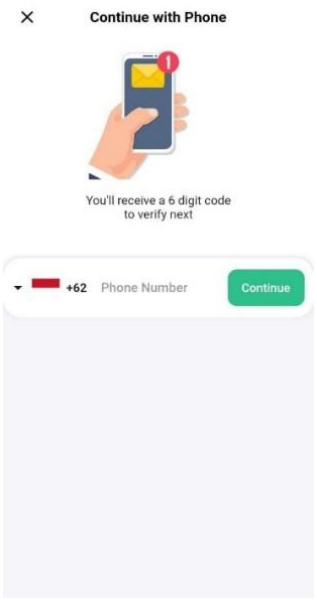
2.5 Perilisan

Aplikasi yang dibuat siap untuk dipasarkan dengan mempublikasi pada *Google Playstore*.

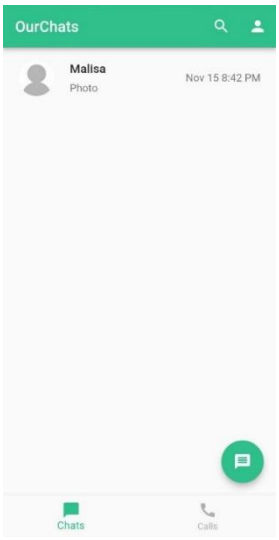
III. HASIL DAN PEMBAHASAN

3.1 Implementasi Sistem

Berikut ini tampilan dari beberapa menu yang sudah dibuat, merupakan hasil implementasi pada tahap metodologi.



Gambar 6. Tampilan masuk aplikasi



Gambar 7. Menu utama

Gambar 8. *Chat room* dengan pengguna lain

Gambar 9. *Video call* dengan pengguna lain

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

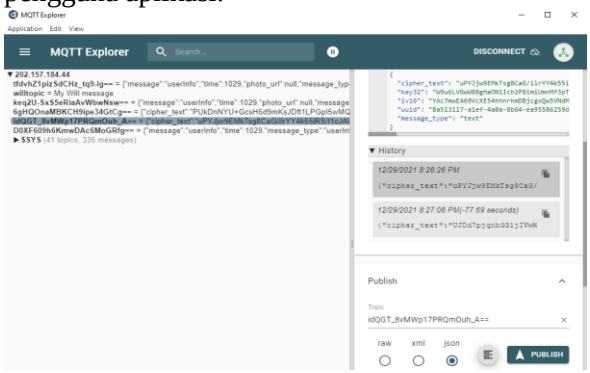
3.2 Hasil Pengujian Keamanan

Dengan sistem yang telah di implementasikan diatas, serta alur kriptografi hibrida yang telah dijelaskan sebelumnya, didapatkan hasil enkripsi pada gambar 10 yang digunakan untuk pertukaran pesan dengan pengguna yang lain.


```
{
  "cipher_text": "AS+aFiT1yY7CdbbZWVXarKr5/ZJ4pQ5Qw9DKuzFCpW
    +v2RtsoMChiwc+sFJ3JMEo8/IykNmq8JNK0QmdtW4ZC8E4uxNVd2csZOWpcDA
    /Xf0Y3YOQtAZolh
    /twopIkVzgsf8npaBDHY3TPqDapXhmlf7k3NU9Pzto2a9hNyLCYIbHbgji
    /PGByYLKXUAE+gf8v8lFelKCrtIrTGh9ukXWdTn3MpqmQWftyHujG115z0QjdO
    +JhB5kyNGs5gEVccVJMgVMOC0ahU83Gu0vQdjDMYq4
    +ri8YL5rhFPru6I1iBVbQK+pGtbOhRtLVG1/ao6o",
  "key32": "QEA94USDC1KBFV5gt2XfHJaJJwGawaxNDYTtR13suEvl8PH08o88GT
    bi1qKpCsk5ybmwjHjWYC1ViMW5Xctc7DMjgWqqBpkH4ODhh5zeX66oUs7Q
    /0oHdx0nPXuUDYoMhR5+PEJej4j+9MRuMfA/Qq5bFYo9KLN2da
    /+WItClG55HGg2Y5pmi4LNJSVO6wO2afSMnsQuC16S56mPaskg6N
    +A0HrS2ybPhExdoPAiptZzDw7rvJr7nnygNj4ZtaMJm5VKJzlKSM9e
    /FTQjKzqktjoJLIwRJ9MFuIQjJQmJcLUFndAH3vpCH2nSwBuIwHP8Jcisw56is
    Zn+pvYOh+pdA==",
  "iv16": "bIjEULgS2Ou9HJQE6AHbSJ21CDr1W/KRJETp8CWoKyOGR
    /M82OZhPdSUVmC3Xzwo73PNOiEUR2waW0F3vrcrHITNJr2vo7TmHjqIa8
    /2OJ3eXylxcb+gL3r68QvzW5E1RZjnDL1rlQYAz4NvKFaXcK0ta5VLWYVm
    +ap1NHQVdfhb6eIAsU
    /6fzGRhQcw3OxK5U2OQU93Z0mL40hcA2yVfjZpa1zuFPZtmqrgURWQcJL56YSZ
    H7azjxGAKFhmQ8
    /UyxPag71zMBtUKNf1LjK5ysHv2cAEQLryoJ8qe7VgdvMi7hLxnu1oxlrfIuZb
    +tkaqhOlnmDOknwNrD2HB45UvQ==",
  "uuid": "bbfafff0-8a0f-4534-a046-88f82c3f15df",
  "message_type": "text"
}
```


Gambar 10. Pesan yang sudah di enkripsi di server

Lalu kemudian mencoba mengirimkan pesan menggunakan aplikasi MQTT Explorer dengan format pesan yang sama untuk menguji apakah pesan tersebut dapat diterima oleh pengguna aplikasi.



Gambar 11. Mengirim pesan menggunakan MQTT Explorer

Hasilnya pesan yang dikirimkan tidak dapat di dekripsi karena pesan yang dikirimkan tidak menggunakan kunci RSA yang valid.

3.3 Hasil Pengujian Performa Enkripsi & Dekripsi

Pengujian performa ini menghitung seberapa cepat waktu yang dibutuhkan untuk proses enkripsi dan dekripsi, dengan panjang kunci yang berbeda dan juga panjang teks yang berbeda serta gambar. Untuk variasi panjang kunci serta spesifikasi perangkat pengujian dapat dilihat pada tabel 3.

Tabel 3. Informasi pengujian performa

Deskripsi	Keterangan
Panjang kunci	RSA (2048 & 4096 bit) & AES (128, 192 & 256 bit)
Perangkat	Huawei Nova 5T (RAM 8GB, penyimpanan 128GB, prosesor kirin 980) & Asus Max Pro M1 (RAM 3GB, penyimpanan 32GB, prosesor snapdragon 636)

Setiap hasil dari pengujian didapatkan dari nilai terendah dari 3 percobaan untuk setiap skenario panjang kunci. Semakin panjang kunci yang digunakan maka semakin panjang juga waktu yang dibutuhkan untuk proses enkripsi dan dekripsi, tetapi keuntungannya adalah keamanan yang diperoleh juga semakin tinggi.

Selain faktor panjang kunci yang memperuhi proses enkripsi dan juga dekripsi, faktor spesifikasi perangkat juga berpengaruh. Dapat dilihat pada tabel 4 dan 5, perangkat dengan spesifikasi lebih tinggi dapat memproses lebih cepat.

Jurnal Teknik Informatika Vol. 14 No. 2, Oktober 2021
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

Tabel 4. Hasil pengujian kecepatan pada Huawei Nova 5T

Panjang Kunci	Panjang Teks	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)
RSA (2048) & AES (128)	250	8	40
	95391 (Gambar)	119	170
RSA (2048) & AES (192)	250	13	50
	95391 (Gambar)	130	185
RSA (2048) & AES (256)	250	14	55
	95391 (Gambar)	160	204
RSA (4096) & AES (128)	250	10	71
	95391 (Gambar)	138	234
RSA (4096) & AES (192)	250	12	105
	95391 (Gambar)	153	249
RSA (4096) & AES (256)	250	14	131
	95391 (Gambar)	160	262

Tabel 5. Hasil pengujian kecepatan pada Asus Max Pro M1

Panjang Kunci	Panjang Teks	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)
RSA (2048) & AES (128)	250	12	55
	95391 (Gambar)	135	177
RSA (2048) & AES (192)	250	17	70
	95391 (Gambar)	160	200
RSA (2048) & AES (256)	250	20	79
	95391 (Gambar)	173	231
RSA (4096) & AES (128)	250	15	85
	95391 (Gambar)	144	249
RSA (4096) & AES (192)	250	21	113
	95391 (Gambar)	160	260
RSA (4096) & AES (256)	250	26	139
	95391 (Gambar)	172	284

3.4 Hasil Pengujian UAT

Pengujian *User Acceptance Test* (UAT) menguji seluruh fitur pada aplikasi, apakah sudah berjalan sesuai atau tidak

Tabel 6. Hasil pengujian UAT

Menu	Fitur	Status
Login	Memasukan nomor	Berhasil
	OTP terkirim ke nomor tujuan	Berhasil
Verification	Kirim ulang OTP	Berhasil
	Verifikasi OTP yang diterima	Berhasil
Setup Profile	Memilih foto profil	Berhasil
	Memasukan nama pengguna	Berhasil
Home	Terdapat daftar *chat*	Berhasil
	Terdapat daftar riwayat panggilan	Berhasil
	Melihat profil pribadi	Berhasil
	Dapat membuka menu *chat room* dari daftar *chat*	Berhasil
	Dapat menghapus *chat*	Berhasil
	Dapat menghapus riwayat panggilan	Berhasil
	Dapat melakukan panggilan video & suara dari Riwayat panggilan	Berhasil
Contact List	Terdapat daftat kontak	Berhasil
	Dapat mencari kontak	Berhasil
	Dapat mengarahkan ke menu buat *add contact*	Berhasil
	Dapat mengarahkan ke menu *contact help*	Berhasil
	Dapat mengarahkan ke menu *chat room*	Berhasil
Add Contact	Dapat menambahkan kontak baru	Berhasil
Contact Help	Dapat menampilkan informasi seputar kontak	Berhasil
Chat Room	Dapat menampilkan daftar pesan	Berhasil
		Berhasil

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

	Dapat membuka profil pengguna lain	
	Dapat mengirim pesan teks	Berhasil
	Dapat mengirim pesan gambar	Berhasil
	Dapat melakukan panggilan video	Berhasil
	Dapat melakukan panggilan suara	Berhasil
	Dapat menghapus pesan	Berhasil
Pickup Call	Dapat menolak panggilan	Berhasil
	Dapat menerima panggilan	Berhasil
Video Call	Dapat menampilkan panggilan video antar pengguna	Berhasil
	Dapat menyelesaikan panggilan	Berhasil
Voice Call	Dapat menampilkan panggilan suara antar pengguna	Berhasil
	Dapat menyelesaikan panggilan	Berhasil

3.5 Implikasi Praktis

Penelitian ini menambahkan beberapa fungsi seperti fitur *video call*, *voice call*, pengiriman gambar, status *online* pengguna. Serta menawarkan sistem keamanan yang lebih baik dengan menggabungkan dua metode kriptografi yaitu asimetris (RSA) dan simetris (AES) atau bisa disebut dengan algoritma kriptografi hibrida [7]. Untuk melengkapi kekurangan dari masing-masing metode.

Metode kriptografi simetris memiliki keuntungan untuk proses yang cepat dan data yang besar tetapi harus menggunakan kunci yang sama untuk kedua prosesnya, sehingga rentan saat data di *sniffing*, sedangkan untuk metode kriptografi asimetris tidak dapat memproses data yang besar. Hal tersebutlah yang membedakan penelitian ini daripada penelitian sebelumnya yang hanya dapat mengirimkan pesan berupa teks dan hanya menggunakan satu macam metode kriptografi.

IV. PENUTUP

Dari pengujian yang telah dilakukan gabungan kedua metode kriptografi menghasilkan performa yang bagus, dapat memproses data dalam jumlah besar dalam waktu milisekon. Performa proses enkripsi dan dekripsi ini mengikuti spesifikasi perangkat yang digunakan, semakin bagus spesifikasi maka performa yang dihasilkan juga semakin meningkat.

Dari segi keamanan kombinasi ini sudah sangat cukup, contohnya saat kunci publik pengguna atau pesan yang dikirimkan berhasil diretas, pihak ketiga masih tidak dapat mengirimkan atau membaca pesan.

Desain yang diajukan oleh penelitian ini masih belum sempurna dapat ditingkatkan dengan mengganti algoritma yang serupa tetapi memiliki performa dan keamanan yang lebih baik.

DAFTAR PUSTAKA

[1] A. Sukma Wijaya, D. T. Nugrahadi, M. I. Mazdadi, A. Farmadi, and A. Rusadi, "Implementation of RSA Encryption Algorithm on Instant Messaging Application," Journal of Data Science and Software Engineering, vol. 1, no. 1, pp. 11-21, 2020.

[2] A. Pujol, D. Magoni, L. Murphy, C. Thorpe, "Spying on Instant Messaging Servers: Potential Privacy Leaks through Metadata," Transactions on Data Privacy, IIIA-CSIC, vol. 12, no. 2, pp.175-206, 2019.

[3] A. Aminudin, G. P. Aditya, and S. Arifianto, "RSA algorithm using key generator ESRKGS to encrypt chat messages with TCP/IP protocol," Jurnal Teknologi dan Sistem Komputer, vol. 8, no. 2, pp. 113–120, Apr. 2020.

[4] R. Siringoringo, "Analisis dan Implementasi Algoritma Rijndael (AES) dan Kriptografi RSA pada Pengamanan File," Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer, vol. 2, no. 1, pp. 31–42, Apr. 2020.

[5] N. Fandier Saragih, R. Jhonson Simamora, R. Siringoringo, and E. Novita Purba, "Konsep Pengamanan Video Conference Dengan Enkripsi AES-GCM Pada Aplikasi Zoom," vol. 4, no. 2, pp. 109, 2020.

[6] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues," IEEE Communications Surveys and Tutorials, vol. 22, no. 2. Institute of Electrical and

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v14i2.23469

Electronics Engineers Inc., pp. 1191–1221, Apr. 2020.

[7] C. Biswas, U. D. Gupta, and M. Md. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography," IEEE, Feb. 2019.

[8] Y. Fatma, A. Hafid, and H. O. Dani, "Peningkatan Keamanan Pengiriman Pesan Teks: Kombinasi Advanced Encryption Standard (AES) 128 dan Least Significant Bit (LSB)," JUSIFO (Jurnal Sistem Informasi), vol. 6, no. 2, pp. 111–120, Dec. 2020.

[9] E. Arboleda, J. Kris, P. Alegro, E. R. Arboleda, M. R. Pereña, and R. M. Dellosa, "Hybrid Schnorr, RSA, and AES Cryptosystem A new method of location estimation for fingerprinting localization technique of indoor positioning system View project Classification View project Hybrid Schnorr, RSA, and AES Cryptosystem," Article in International Journal of Scientific & Technology Research, vol. 8, no. 10, pp. 1770-1776, Oct. 2019.

[10] I. I. Lutfa, "Pengembangan Aplikasi Pesan Instan Terenkripsi Menggunakan Algoritma Kriptografi AES (Advanced Encryption Standard)," Jurnal Teknik Elektro Smart, vol.1, no.1, pp. 1-6, Aug. 2021.