5th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
7 September 2024 – Jakarta, Indonesia
Volume 3, Nomor 2, September 2024 - ISSN 2962-8628 (*online*)

IMPLEMENTASI DAN ANALISIS KOMBINASI RSA, AES DAN STEGANOGRAFI PADA ENKRIPSI DATA KELURAHAN KADEMANGAN

Iskandar Zulkarnain[1] , Hari Soetanto[2]

[1,2] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Kota Tangerang, Indonesia

Email: [1]2011501281@student.budiluhur.ac.id, [2]hari.soetanto@budiluhur.ac.id
(* : corresponding author)

Abstrak- Di era digitalisasi saat ini, perkembangan teknologi komputer dan telekomunikasi telah menjadi bagian esensial dari kehidupan sehari-hari. Namun, kemajuan ini juga membawa dampak negatif, termasuk risiko pencurian data. Pencurian data merupakan ancaman serius dalam pertukaran informasi dan penyimpanan data, terutama yang bersifat pribadi dan sensitif. Dalam mengamankan data tersebut, kriptografi menawarkan solusi melalui teknik penyandian yang mengubah informasi menjadi bentuk tersandi yang hanya dapat diakses oleh pihak berwenang. Penelitian ini bertujuan untuk merancang sistem keamanan yang mengintegrasikan Algoritma kriptografi simetris dan asimetris serta teknik steganografi untuk mengamankan data sensitif dari akses tidak sah.Penelitian ini menggunakan kombinasi tiga teknik keamanan: Algoritma simetris *Advanced Encryption Standard* (AES-256) untuk penyandian cepat data, Algoritma asimetris *Rivest Shamir Adleman* (RSA) yang menggunakan kunci publik dan privat, dan metode steganografi yang menyembunyikan informasi dalam media digital. Dari hasil pengujian terhadap sepuluh berkas, diperoleh akurasi sebesar 100%, yang menunjukkan bahwa algoritma ini dapat berfungsi dengan baik. Integrasi ketiga teknik ini diharapkan meningkatkan keamanan data dengan memanfaatkan kelebihan masing-masing metode dalam skenario yang berbeda.

Kata Kunci: Steganografi, AES-256, RSA, Pencurian Data.

IMPLEMENTATION AND ANALYSIS OF THE COMBINATION OF RSA, AES AND STEGANOGRAPHY IN THE ENCRYPTION OF URBAN VILLAGE DATA

Abstract- *In the current era of digitalisation, the development of computer and telecommunications technology has become an essential part of everyday life. However, these advancements also bring negative impacts, including the risk of data theft. Data theft is a serious threat to information exchange and data storage, especially of a personal and sensitive nature. In securing such data, cryptography offers a solution through encryption techniques that convert information into an encrypted form that can only be accessed by authorised parties. This research aims to design a security system that integrates symmetric and asymmetric cryptographic algorithms and steganographic techniques to secure sensitive data from unauthorised access.This research uses a combination of three security techniques: Advanced Encryption Standard (AES-256) symmetric algorithm for fast encoding of data, Rivest Shamir Adleman (RSA) asymmetric algorithm that uses public and private keys, and steganography method that hides information in digital media. From the test results of ten files, an accuracy of 100% was obtained, which shows that this algorithm can function properly. The integration of these three techniques is expected to improve data security by utilising the advantages of each method in different scenarios.*

***Keywords**:* Steganography, AES-256, RSA, Data theft

1. PENDAHULUAN

Kemajuan teknologi komputer dan telekomunikasi sekarang ini menjadi kebutuhan yang penting bagi setiap orang, tetapi dengan keuntungan tersebut ada juga dampak negatif yang muncul. Salah satu dampak negatif dalam perkembangan teknologi adalah pencurian data. Karena adanya pencurian data, keamanan dalam pertukaran informasi dan penyimpanan data dianggap penting. Salah satu data yang penting adalah masalah data yang bersifat pribadi, data pribadi merupakan sebuah data yang berisifat sensitif. Untuk itu diperlukannya sebuah sistem yang mampu untuk mengamankan data sensitif tersebut.[1] Kriptografi adalah cara yang dapat diimplementasikan dalam menjaga keamanan informasi atau pesan dengan cara menyamarkannya menjadi bentuk tersandi(*Chipertext*) yang tidak dapat dibaca. Bentuk tersandi tersebut hanya bisa dibaca atau diketahui oleh pihak

yang berhak untuk membacanya saja. Kriptografi sendiri memiliki beberapa jenis Algoritma yang dapat diterapkan, seperti Algoritma simetris (kunci tunggal) dengan salah satu contohnya AES (*Advanced Encryption Standard*) dan Algoritma asimetris (kunci ganda) dengan salah satu contohnya RSA (*Rivest-Shamir-Adleman*) .[2]

Kelurahan adalah salah satu unit pemerintahan di tingkat desa yang bertanggung jawab atas urusan Pemerintahan, Pembangunan, Kemasyarakatan, dan Perekonomian. Selain itu, kelurahan bertanggung jawab atas penyelenggaraan pemerintahan yang melayani dan memberdayakan masyarakat di tingkat desa. Kepala kelurahan dipilih melalui pemilihan kepala desa. Kelurahan memiliki wilayah administrasi sendiri yang terdiri dari berbagai dusun atau lingkungan.[3]

Penelitian ini menggunakan kombinasi Algoritma kriptografi asimetris dan simetris, yaitu Algoritma *Advanced Encryption Standard* (AES-256), RSA (*Rivest Shamir Adleman*) dan metode Steganografi. Algoritma AES merupakan Algoritma simetris yang cukup aman untuk mengamankan data atau informasi yang bersifat rahasia. Sementara Algoritma RSA merupakan Algoritma kriptografi asimetri, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Kunci yang digunakan untuk mengenkripsi disebut dengan kunci *public*, dan yang digunakan untuk mendekripsi disebut dengan kunci *private*. Oleh karena itu, diharapkan dengan kombinasi dua Algoritma ini dapat memiliki tingkat keamanan yang cukup tinggi untuk menjaga keamanan dan kerahasiaan data atau informasi yang penting tersebut. Steganografi adalah teknik atau sebuah metode yang bertujuan untuk menyembunyikan sebuah pesan didalam pesan seperti menyembunyikan pesan tertulis kedalam sebuah gambar atau audio .

Aplikasi ini dinamakan *Cryptsec*, yang merupakan singkatan dari *Cryptography Secure*. Sesuai dengan tujuan utamanya, aplikasi ini dirancang untuk mengamankan file menggunakan metode yang menggabungkan tiga metode kriptografi . *Cryptsec* memanfaatkan teknik enkripsi dan dekripsi untuk memastikan bahwa data pengguna tetap terlindungi dari akses tidak sah dan potensi ancaman keamanan.

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi ialah bidang yang mempelajari metode matematika untuk melindungi data dan informasi, seperti validasi, otentikasi, dan integritas. Enkripsi adalah proses konversi antara pesan teks asli (*Plaintext*) menjadi pesan terenkripsi (*Chipertext*) melalui sistem Kriptografi. [4] Kriptografi melindungi data dari askes yang tidak sah di dunia modern, memastikan bahwa data tetap rahasia, dapat dipercaya, dan asli. Dalam keamanan dunia maya, ada tiga pilar utama yaitu, Kerahasiaan (*Confidentiality*), Autentikasi (*Authentication*), dan Integritas atau keutuhan data (Data *Integrity*).

2.2 *Advanced Encryption Standard* (AES)

Algoritma *Advanced Encryption Standar(*AES*)* adalah Algoritma kriptografi yang digunakan untuk mengenkripsi dan dekripsi data. Ini adalah Algoritma blok kode simetris dan menggantikan Algoritma Enkripsi Data Standar (DES)[5], dengan panjang kunci yang berbeda-beda sepanjang 128 bit. Studi ini memilih AES-256 bit karena blok 256 bit cukup baik untuk menyediakan mekanisme pengamanan tambahan untuk data meskipun lebih banyak membutuhkan sumber daya daripada blok AES-192 dan AES-128.

Secara umum, langkah-langkah untuk enkripsi AES-256 dengan kunci 256bit adalah sebagai berikut:

a. *AddRoundKey:* Langkah pertama enkripsi AES, di mana kunci ronde ditambahkan ke blok data menggunakan operasi XOR.
b. *SubBytes:* Setiap *byte* dalam blok data digantikan dengan *byte* lain menggunakan tabel substitusi (S-Box) untuk meningkatkan keacakan.
c. *ShiftRows:* Baris dalam blok data digeser ke kiri dengan jumlah tertentu untuk meningkatkan keacakan.
d. *MixColumns:* Setiap kolom dalam blok data dikalikan dengan matriks tetap untuk menghasilkan lebih banyak difusi.
e. *AddRoundKey*: Diulangi setiap ronde kecuali ronde terakhir, kunci ronde ditambahkan lagi ke blok data menggunakan XOR.
f. *Nr-1:* Kondisi pengecekan ronde terakhir. Jika belum, ulangi proses *SubBytes, ShiftRows, MixColumns*, dan *AddRoundKey*.
g. *SubBytes:* Dilakukan kembali pada ronde terakhir, setiap *byte* digantikan dengan byte lain menggunakan

tabel substitusi (S-Box).

h. *ShiftRows:* Dilakukan kembali pada ronde terakhir, baris digeser ke kiri dengan jumlah tertentu.
i. *AddRoundKey*: Terakhir kali pada ronde terakhir, kunci ronde ditambahkan ke blok data menggunakan XOR.
j. *Ciphertext:* Hasil akhir enkripsi, yaitu teks yang telah dienkripsi.[6]

.

Proses dekripsi AES-256 dengan kunci 256 bit secara umumadalah sebagai berikut:

a. *InvShiftRows*: melakukan pergeseran bit ke kanan padasetiap blok baris;
b. *InSubBytes*: memetakan setiap elemen pada state dengantabel Inverse S-Box; dan
c. *InvShiftRows*: mengubah bit ke kanan pada setiap blok baris.
d. *InvMixColumns*: matriks AES kalikan semua kolom *state*.
e. *AddRoundKey*: menggabungkan *array state* dan tombol *round* dengan hubungan XOR. Proses terakhir menghasilkan karakter atau teks asli(*plaintext*).

2.3 *Rivest Shamir Adleman* (RSA)

RSA merupakan salah satu teknik kriptografi asimetris yang menggunakan kunci publik yang terkenal. Keamanan RSA tinggi karena menggunakan dua kunci yang berbeda untuk enkripsi dan dekripsi, dan kesulitan dalam memfaktorkan bilangan menjadi faktor prima untuk mendapatkan kunci dekripsi. Meskipun demikian, seringkali RSA rentan terhadap serangan *Brute Force*. Proses enkripsi dan dekripsi dalam RSA tidak bergantung pada Algoritma, melainkan pada proses matematika yang menghasilkan kunci rahasia yang dapat disebarkan secara bebas dan hanya dapat didekripsi oleh penerima pesan. Faktor-faktor prima yang digunakan dalam Algoritma ini akan difaktorkan oleh bilangan besar[7].

2.3.1 Algoritma RSA

Untuk menggunakan Algoritma RSA, langkah pertama adalah menghasilkan dua bilangan prima yang berbeda, p dan q. Modulus, disimbolkan dengan n, dihasilkan dari perkalian kedua bilangan prima tersebut. Proses matematis dalam menentukan nilai modulus melibatkan perhitungan *totient/phi n*. Selanjutnya, nilai eksponen enkripsi, disimbolkan dengan e, dipilih agar memenuhi syarat tertentudan haruslah bilangan prima terhadap *totient* n. Nilai eksponen dekripsi, disimbolkan dengan d, dihitung dengan persamaan yang melibatkan nilai variabel k, p, dan q. Nilai d, p, dan q harus dirahasiakan, sementara nilai n dan e dapat disebarkan secara terbukasebagai kunci publik. Pasangan (n, e) merupakan kunci publik, sementara pasangan (n, d) merupakan kunci rahasia.[7]

2.4 Steganografi

Steganografi merupakan seni menyembunyikan pesan dalam suatu media yang sering digunakan untuk mengirim pesan melalui jaringan internet tanpa diketahui orang lain. Penggunaan steganografi untuk menyembunyikan pesan akan mengurangi kecurigaan dan kemungkinan terdeteksinya pesan oleh pihak ketiga. Media yang digunakan sebagai media pembawa (*carrier file*) dapat berupa teks, gambar, audio, dan video.Salah satu media pembawa yang paling sering digunakan dalam steganografi adalah gambar/citra digital, karena sering dipertukarkan dalam dunia internet, seperti JPEG, PNG, GIF, dan BMP. [8]

2.4.1 Algoritma Steganografi *End Of File(EOF)*

Metode ini merupakan metode pengembangan LSB (*Least Significant Bit*). Dalam metode ini pesan disisipkan diakhir berkas. Pesan yang disisipkan dengan metode ini jumlahnya tidak terbatas. Akan tetapi efek sampingnya adalah ukuran berkas menjadi lebih besar dari ukuran semula. Ukuran berkas yang terlalu besar dari yang seharusnya, tentu akan menimbulkan kecurigaan bagi yang mengetahuinya.[9]

2.5 *Confusion Matrix*

Untuk melakukan evaluasi klasifikasi penelitian ini, metode *Confusion Matrix* digunakan. Metode ini melibatkan membandingkan matriks prediksi dengan kelas asli, yang memiliki informasi nyata dan prediksi nilai klasifikasi. Setelah sistem berhasil mengklasifikasikan tweet, ukuran diperlukan untuk mengevaluasi ketepatan dan akurasi klasifikasi. Proses pengujian klasifikasi menggunakan metode *Confusion Matrix* ini sangat penting untuk memastikan bahwa hasilnya akurat.[10]

2.6 Data Penelitian

Data yang digunakan dalam penelitian ini merupakan data pribadi yang diberikan oleh Kelurahan Kademangan dan digunakan untuk menguji metode kriptografi. Data tersebut mencakup informasi sensitif yang digunakan untuk mengevaluasi efektivitas dan keamanan metode enkripsi yang diterapkan. Penelitian ini bertujuan untuk mengetahui metode kriptografi yang digunakan dapat melindungi data pribadi dengan baik, mencegah akses yang tidak sah, dan menjaga kerahasiaan informasi selama proses pengujian. Analisis ini penting untuk mengembangkan sistem pengamanan data yang andal dan efisien. Pada tabel 1 merupakan nama *file*, jenis *file* dan ukuran *file* sebelum dilakukan enkripsi.

Tabel 1 Data Penelitian

Nama *file*	Jenis *file*	Ukuran *file*
Data Guru NgajiKel Kademangan	.xlsx	16 KB
Data PBI 1	.xlsx	11 KB
Data PPLS Kademangan	.xlsx	10 KB
Data Warga RT 03	.xlsx	52 KB
Data Warga BerKTP DKI	.xlsx	10 KB
Jadwal	.xlsx	10 KB
Data Mutasi	.xlsx	16 KB
Laporan Kegiatan	.xlsx	10 KB
UpperCase	.xlsx	9 KB
LowerCase	.xlsx	9 KB

2.7 Penerapan Metode

Teknik penerapan metode yang digunakan adalah melakukan kombinasi dari dua jenis Algoritma dalam kriptografi dan steganografi. Dua kombinasi itu terdari dari Algoritma *Advanced Encryption Standard (AES)* dan *Rivest-Shamir-Adleman (RSA)* dan juga menggabungkan teknik dari steganografi. Prosesnya adalah sebagai berikut.

a. Gunakan kalimat “iskandar” sebagai basis untuk menghasilkan kuncisimetris AES
b. Generasi pasangan kunci RSA yang dihasilkan oleh *system openssl*,sehingga menghasilkan
    1. Kunci public: “..”(Terlalu Panjang untuk disertakan secara manual)
    2. Kunci Private : “..”.
c. Konversi *string* “iskandar” menjadi kunci simetris 256-*bit* (32 *byte*) dengan hashing, misalkan menggunakan SHA-256.
    1. ‘symetrickey = SHA-256(“iskandar”)’
    2. ‘symetrickey‘=‘e6fb06210fafc02fd7479ddbed2d042cc3a5155e0562a11a4df7f092bfc6c82b(32 byte)
    3. Buat IV(Initialization Vector) 16 byte secara acak:
       IV = ‘d6c7e25c8eb3b5b4b3a3d7e8f7c6a5a4’ (16byte)
    4. Enkripsi data dengan AES-256-CBC :
       encryptedData = e4b7c8d6e6f8b3c4d2f9a3b4c5d7a8c6 (dalam hex)
    5. Enkripsi kunci simetri dengan RSA :
       Encrypt data = ABC123==
    6. Gabungkan kunci simetris, IV dan data yang sudah di enkrip Output = encryptedSymmetricKey + "::" + iv + "::" + encryptedData Output=

QUJDMTIzOjpkNmM3ZTI1YzhlYjNiNWI0YjNhM2Q3ZThmN2M2YTVhNDo6ZTRiN2M4ZDZl NmY4YjNjNGQyZjlhM2I0YzVkN2 E4YzY=

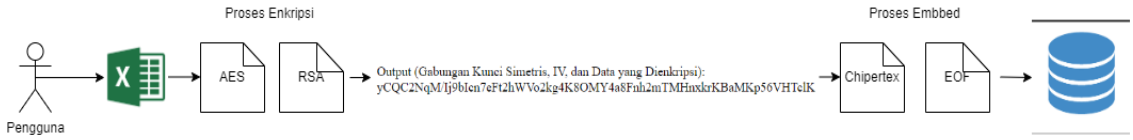
7. Sisipkan data kedalam gambar PNG
   imageContent = "89504e470d0a1a0a0000000d49484452..." (hex dari gambar PNG)
8. Gabungkan konten gambar dengan data ter-enkripsi pada akhir byte array gambar dengan separator "::ENCRYPTEDDATA::". modifiedImageContent = imageContent + "::ENCRYPTEDDATA::" + output.

2.8 Gambaran Umum Metode

Gambaran umum metode ini menjelaskan bagaimana gambarandari ketiga algoritma dalam melakukan enkripsi dan dekripsi padafile dengan extention xlsx. Dengan urutan sebagai berikut :

a. Pengguna melakukan input file dengan *extention* file.
b. Kemudian dilakukan proses enkripsi dengan algoritma RSA, AES.
c. Dari hasil enkripsi tersebut menghasilkan *output Ciphertext.*
d. Proses selanjutnya adalah proses penyisipan *chipertext.* dari kedua algoritma tersebut kedalam *image.*

2.9 Rancangan Pengujian



Gambar 1 Gambaran Umum

Aplikasi ini menggunakan kombinasi tiga algoritma kriptografi, yaitu *Advanced Encryption Standard (AES), Rivest–Shamir–Adleman (RSA),* dan Steganografi. Terdapat menu *login* di mana pengguna harus memasukkan *username* dan *password* terlebih dahulu. Aplikasi ini memiliki dua halaman *login* yang berbeda untuk *user* dan admin. Halaman *dashboard* utama untuk admin berbeda dengan *user*, yang dibedakan oleh path URL.

Pengguna harus memastikan *path URL* sesuai dengan halaman login yang diinginkan sebelum memasukkan *username* dan *password*. Setelah *login*, pengguna akan diarahkan ke halaman *dashboard* utama yang memiliki dua *sidebar* dengan beberapa *subsidebar*. Tampilan *sidebar* untuk *user* dan admin terlihat sama, berisi dekripsi data, enkripsi data, dan data *user*, namun berbeda dalam pengoperasiannya. *User* hanya bisa *mendownload file* yang telah dienkripsi, sedangkan admin bisa menambahkan *file* untuk dienkripsi. Aplikasi ini juga memiliki menu *logout* untuk mengakhiri sesi aktif dan kembali ke halaman *login*. Berikut pada Tabel 2 merupakan rancangan pengujian sistem yang dilakukan pada penelitian ini

Tabel 2 Tabel Rancangan Pengujian

No.	Skenario Pengujian	Hasil yang diharapkan
1.	Form login diinput oleh pengguna	Tampil halaman *dashboard* utama
2.	*Admin* memilih menu *sidebar master data* enkripsi	Tampil halaman enkripsi dengan tombol tambah dan refresh
3.	Pilih file kemudian beri tanggaldan nama *file* klik simpan	Tampil *popup* "*file* berhasil di enkripsi"
4.	Men-*download file* yang sudah dienkripsi dan terdekripsi	Berhasil men-*download file*

5.	*User* atau admin menekan tombol *logout*	Kembali kehalaman *login*
6.	*Admin* memilih menu *data user*	Tampil halaman menu *data user*
7.	*Admin* menambah atau menghapus *user* yang terdaftarpada *data user*	*User* berhasil ditambah atau dihapus

2.10 Spesifik *Database*

Struktur *database* yang akan digunakan adalah sebagai berikut. Tabel ini menyimpan daftar yang telah diproses oleh program,lengkap dengan deskripsinya masing-masing.

a. Nama *Database* : iskand
Nama *Table* : data_dekripsi
Primary Key : id_dekripsi
Foreign Key : -
Isi : data yang sudah didekripsi

Tabel 3 Spesifikasi Tabel data dekripsi

Nama Field	Type	Ukuran	Keterangan
id_dekripsi	text	-	Id file
tanggal	date	-	Tanggal file
file	varchar	100	File nama setelah di dekripsi
Nama_file	varchar	100	Nama file awal

b. Nama *Database* : iskand
Nama *Table* : data_enkripsi
Primary Key : id_enkripsi
Foreign Key : id_*user*
Isi : data yang sudah dienrkipsi

Tabel 4 Spesifikasi Tabel data enkripsi

Nama Field	Type	Ukuran	Keterangan
id_dekripsi	text	-	Id file
tanggal	date	-	Tanggal file
file	varchar	100	File nama setelah di dekripsi
nama_file	varchar	50	Nama file awal
id_user	varchar	50	Id user
PUBLIC_KEY	text	-	Public key
PRIVATE_KEY	longtext	-	Private key

3. HASIL DAN PEMBAHASAN.

3.1 *Flowchart*

Flowchart, juga disebut sebagai bagan alur, adalah diagram yang menunjukkan langkah-langkah dan keputusan yang harus diambil untuk menjalankan sebuah proses program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. Bagan alur proses program akan lebih mudah dipahami, lebih ringkas, dan mengurangi kemungkinan salah penafsiran. Dalam dunia pemrograman, *flowchart* adalah cara yang bagus untuk menghubungkan kebutuhan non- teknis dan teknis..

3.1.1 *Flowchart* Enkripsi RSA

Flowchart ini menggambarkan alur yang terjadi pada algoritma RSA. Saat pengguna menjalankan fitur enkripsi dan meng-input kan *file* denga *extention* xlsx dan klik simpan, maka sistem akan menjalankan proses Algoritma RSA seperti *generate private key* dan *public key* yang dihasilkan oleh sistem dalam hal ini *Openssl*. Kemudian setelah Algoritma RSA mendapatkan *private key* dan *public key* proses akan dilanjutkan oleh Algoritma AES terlebih dahulu, setelah proses Algoritma AES selesai maka Proses Algoritma RSA kembali dijalankan untuk melakukan enkripsi terhadap *symetric key* AES.

3.1.2 *Flowchart* Algoritma Enkripsi AES

Flowchart pada bagian ini menggambarkan proses yang terjadipada Algoritma AES Saat pengguna menjalankan fitur enkripsi dan meng-inputkan *file* dengan*extention xlsx* dan klik simpan, maka sistem akan menjalankan *genarate key* RSA terlebih dahulu seperti yang sudah dijelaskan pada point 3.1.2, setelah proses tersebut selesai maka proses selanjutnya adalah Algoritma AES melakukan generate *symetric key*terlebih dahulu, setelah proses ini selesai sistem akan melanjutkan proses Algoritma AES untuk mengenkripsi *plaintext* dalam hal ini adalah *file* yang di inputkan oleh *user* dengan *extention* xlsx, sehingga dari proses tersebut menghasilkan *ciphertext* atau bisa disebut sebuah kalimat yang tidak terbaca.

3.1.3 *Flowchart* Algoritma Steganografi *End Of File* EOF

Setelah semua proses enkripsi sudah dilakukan dan menghasilkan *ciphertext* dari kedua algoritma enkripsi AES dan RSA, langkah yang terakhir adalah untuk memasukan *ciphertext* tersebut kedalam sebuah gambar atau yang biasa di sebut dengan teknik steganografi. Pada penelitian ini menggunakan teknik steganografi *End Of File*(EOF). Alur agloritma ini akan terlihat pada gambar *Flowchart* di bawah ini.

3.1.4 *Flowchart* Algoritma Pemisahan

Pada proses pendekripsian sebelum menjalankan proses RSA dan AES ada beberapa proses yang berjalan terlebih dahulu, pada proses ini diberi nama Algoritma pemisahan, tentunya proses ini berjalan diluar algortima RSA dan AES dan proses ini meliputi proses pengambilan konten *file* yang terenkripsi sebelumnya, pemisahan *Ciphertext* yang sudah disisipkan didalam *image*, proses *decode* dari *base* 64 dan proses pengambilan kunci *private* dari *database*.

3.1.5 *Flowchart* Algoritma Dekripsi RSA

Proses dekripsi dengan Algoritma RSA digunakan untuk mendekripsi kunci simetris dari AES, yang kemudian akan digunakan oleh Algoritma AES untuk melanjutkan proses dekripsi data. Langkah awal ini sangat penting karena kunci simetris AES yang terenkripsi oleh RSA harus terlebih dahulu diubah kembali kebentuk aslinya sebelum dapat digunakan oleh Algoritma AES. Dengan menggunakan kunci privat RSA[11], kunci simetris AES yang terenkripsi dapat didekripsi dengan aman dan diambil kembali.

3.1.6 Flowchart Algoritma Dekripsi AES

Memasuki proses dekripsi menggunakan Algoritma AES, pada proses dekripsi Algoritma AES akan terjadi beberapa proses yang terjadi beberapa tahapan penting akan dijalankan secara bertahap sebagai berikut:

Pertama, mengatur kunci dan IV (*Initialization Vector*) yang telah ditentukan sebelumnya. Kunci dan IV ini sangat penting untuk memastikan bahwa proses dekripsi dapat dilakukan dengan benar dan sesuai dengan data yang telah dienkripsi.

Selanjutnya, proses dekripsi data akan melibatkan beberapa langkah penting seperti *AddRoundKey*, yang berfungsi untuk menggabungkan kunci dengan data yang dienkripsi. Kemudian, proses *InvShiftRows* akan

dilakukan, di mana baris-baris data akan dipindahkan kembali ke posisi semula. Setelah itu, *InvSubBytes* akan menggantikan *byte* yang terenkripsi dengan *byte* yang sesuai dari tabel substitusi.

Terakhir, proses *InvMixColumns* akan dilakukan untuk mengembalikan kolom-kolom data ke bentuk semula dengan menggunakan operasi matematika khusus.

3.2 Hasil Pengujian

Pengujian aplikasi ini bertujuan untuk mengevaluasi hasil dari sistem aplikasi pengamanan *file* yang bernama *Cryptsec*. Pengujian ini melibatkan dua proses utama: proses enkripsi dan proses dekripsi *file*.

Pada tahap pengujian proses enkripsi, Penelitian ini akan memeriksa bagaimana aplikasi *Cryptsec* mengubah data asli menjadi bentuk terenkripsi, memastikan bahwa data tersebut tidak dapat diakses atau dibaca tanpa melalui proses dekripsi yang benar. Uji ini mencakup penilaian terhadap Durasi enkripsi, serta kemampuan aplikasi dalam menangani *file* dan ukuran data yang berbeda.

Selanjutnya, pada tahap pengujian proses dekripsi, penulis melihat bagaimana aplikasi mengembalikan data terenkripsi ke bentuk aslinya.

3.2.1 Hasil Perancangan Pengujian Program

Table 5 Hasil Perancangan Pengujian Program

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian
1.	Form login diinput oleh pengguna	Tampil halaman *dashboard* utama	Valid
2.	*Admin* memilih menu *sidebar master data* enkripsi	Tampil halaman enkripsi dengan tombol tambah dan *refresh*	Valid
3.	Pilih *file* kemudian beri tanggal dan nama*file* klik simpan	Tampil *popup* "*file* berhasil dienkripsi"	Valid
4.	Men-*download file* yang sudah dienkripsidan terdekripsi	Berhasil men-*download file*	Valid
5.	*User* atau *admin* menekan tombol *logout*	Kembali kehalaman *login*	Valid
6.	*Admin* memilih menu *data user*	Tampil halaman menu *data user*	Valid
7.	*Admin* menambah atau menghapus *user*yang terdaftar pada *data user*	*User* berhasil ditambah atau dihapus	Valid

3.2.2 Hasil *Confusion Matrix*

Untuk menghitung akurasi menggunakan metode *confusion matrix* berdasarkan ukuran *file* dan durasi waktu antara enkripsi dan dekripsi, perlu mengklasifikasikan hasilnya ke dalam kategori *true positive* (TP*), true negative (TN), false positive (FP), dan false negative (FN).* Berikut adalah penjelasan kategorinya:

a) *True Positive (TP)*: Ukuran *file* dan durasi waktu antara enkripsi dan dekripsi sesuai (sama atau dalam toleransi yang dapat diterima).
b) *True Negative (TN):* Ukuran *file* dan durasi waktu antara enkripsi dan dekripsi tidak sesuai, tetapi hal ini diharapkan (misalnya karena jenis file yang berbeda).
c) *False Positive (FP):* Ukuran *file* atau durasi waktu sesuai ketika seharusnya tidak sesuai.
d) *False Negative (FN):* Ukuran *file* atau durasi waktu tidak sesuai ketika seharusnya sesuai.

Table 6 Tabel Perbandingan anatara Enkripsi dan Dekripsi

No	Nama File	Ukuran File Awal (KB)	Ukuran File setelah Enkripsi (KB)	Durasi Waktu Enkripsi (Detik)	Ukuran File setelah Dekripsi (KB)	Durasi Waktu Dekripsi (Detik)
1	PBI 1.xlsx	11	26	3,6	26	3,6
2	Data Guru Ngaji Kel Kademangan.xlsx	16	33	2,6	33	2,6
3	Data PPLS Kademangan 2024.xlsx	10	25	1,0	25	1,0
4	Data Warga RT 03	51	79	1,3	79	1,3
5	Data Warga BerKTP DKI	10	25	2,1	25	2,1
6	Jadwal	10	24	3,3	24	3,3
7	Data Mutasi	16	33	24	33	2,4
8	Laporan Kegiatan	10	25	3,9	25	3,9
9	UpperCase	9	23	7,3	23	7,3
10	LowerCase	9	23	1,9	23	1,9

Dari tabel tersebut, dapat dilihat bahwa ukuran *file* setelah enkripsi dan dekripsi serta durasi waktu untuk setiap *file* sama persis. Jadi, semua entri adalah *true positive* (TP). Berdasarkan hal ini, didapat *confusion matrix* sebagai berikut:

True Positive (TP) 10
True Negative (TN) 0
False Positive (FP) 0
False Negative (FN) 0

Dengan *Confusion Matrix* ini, didapat hitungan akurasi:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Akurasi == \frac{10+0}{10+0+0+0} = 1.0$$

akurasi proses enkripsi dan dekripsi berdasarkan data yang diberikan adalah 100%.

4. KESIMPULAN

Dari hasil pengujian dan analisa yang dilakukan, dapat disimpulkan bahwa kombinasi algoritma RSA, AES, dan Steganografi memiliki tingkat akurasi 100% berdasarkan pengujian *file* yang telah dilakukan dengan cara enkripsi dan dekripsi. Dalam penelitian ini, algoritma RSA digunakan untuk melakukan enkripsi terhadap kunci simetris AES guna mencegah pencurian kunci selama proses distribusi kunci. Aplikasi kriptografi ini memanfaatkan Algoritma *Rivest Shamir Adleman (RSA), Advanced Encryption Standard (AES)*, dan Steganografi untuk menyediakan sistem keamanan pada aplikasi yang saya kembangkan. Aplikasi ini mengamankan *file-file* dengan ekstensi xlsx.

DAFTAR PUSTAKA

[1] Rahmad Prayogi Harahap and A. H. Hasugian, "Teknik Keamanan Data Menggunakan Metode Vigenere Cipher Dan Steganografi Dalam Penyisipan Pesan Teks Pada Citra," *J. Fasilkom*, vol. 13, no. 3, pp. 570–577, 2023, doi: 10.37859/jf.v13i3.6184.

[2] Sebastian, S., et al, "Analisis Performa Kriptografi Hybrid Algoritma Blowfish dan Algoritman RSA," *JURTEKSI: Jurnal Teknologi dan Sistem Informasi*, vol. 6, no. 1, 2019, doi: 10.33330/jurteksi.v6i1.395.

[3] Sugiman, "Pemerintah Desa," *Pemerintah. Desa, Fak. Huk. Univ. Suryadarma*, vol. 7, no. 1, pp. 82–95, 2018, [Online]. Available: https://media.neliti.com/media/publications/275406-pemerintahan-desa-bc9190f0.pdf

[4] D. I. Mulyana, A. P. Heryani, and V. Khoirunnisa, "Implementasi Metode Rivest Shamir Adleman untuk Enkripsi dan Dekripsi Text," vol. 03, no. 01, pp. 32–39, 2022.

[5] T. D. A. P. Wardhani and Y. Asriningtias, "Implementasi Algoritma AES-256 Dalam Perancangan Aplikasi Pengamanan Dokumen Digital Perusahaan Berbasis Android," *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 6, no. 2, pp. 1289–1293, 2024, doi: 10.31539/intecoms.v6i2.8027.

[6] R. H. Irawan, U. Mahdiyah, and R. D. Kurniawan, "Implementasi Algoritma AES Pada Aplikasi Pembelian Voucher Hotspot Berbasis Android," *Gener. J.*, vol. 8, no. 1, pp. 18–26, 2024, doi: 10.29407/gj.v8i1.20817.

[7] D. T. Tobing, "Implementasi Algoritma Rivest Shamir Adleman (RSA) Untuk Keamanan Data Rekam Medik Penyakit Pasien Rumah Sakit," *J. Kaji. Ilm. Teknol. Inf. dan Komput.*, vol. 2, no. 2, pp. 65–73, 2024, doi: 10.62866/jutik.v2i2.131.

[8] A. P. Ratnasari and F. A. Dwiyanto, "Metode Steganografi Citra Digital," *Sains, Apl. Komputasi dan Teknol. Inf.*, vol. 2, no. 2, p. 52, 2020, doi: 10.30872/jsakti.v2i2.3300.

[9] A. Eka Putri, A. Kartikadewi, and L. A. Abdul Rosyid, "Implementasi Kriptografi dengan Algoritma Advanced Encryption Standard (AES) 128 Bit dan Steganografi menggunakan Metode End of File (EOF) Berbasis Java Desktop pada Dinas Pendidikan Kabupaten Tangerang," *Appl. Inf. Syst. Manag.*, vol. 3, no. 2, pp. 69–78, 2021, doi: 10.15408/aism.v3i2.14722.

[10] A. Sentimen, T. Boikot, B. P. Pada, N. F. Az-haari, D. Juardi, and A. Jamaludin, "Twitter Menggunakan Metode Naïve Bayes (Studi Kasus : Starbucks)," vol. 8, no. 3, pp. 4256–4261, 2024.

[11] E. Jurnal *et al.*, "Penerapan Algoritma Rivest-Shamir-Adleman (RSA) pada Enkripsi Uniform Resource Locator (URL) Website untuk Keamanan Data," vol. 11, no. 2, pp. 205–215, 2023.