Journal of Artificial Intelligence and Software Engineering
Vol. 5, No. 3, September 2025, pp. 1141~1151
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i3.7629

Enhancement of Rivest Shamir Adleman (RSA) Key Generation Utilizing the Diffie-Hellman Algorithm for PDF File Security

Maliyah Murti[1], Rahmadani[2], Ratih Puspadini[3*]
[1] Teknik Informatika, STMIK Kaputama, Kota Binjai, Indonesia
[2] Universitas Pembangunan Panca Budi, Kota Medan, Indonesia
[3] STMIK Kaputama, Kota Binjai, Indonesia

Informasi Artikel

Diterima : 20 Agustus 2025
Revisi : 9 September 2025
Publikasi : 30 September 2025

Kata Kunci:

Kriptografi
Algoritma RSA
Diffie-Hellman
File Pdf
Keamanan Data.

ABSTRAK

Perkembangan teknologi digital menimbulkan tantangan keamanan data, terutama pada file PDF yang sering digunakan. Kriptografi dapat digunakan untuk keamanan data. Algortima RSA konvensional dengan kunci publik statis rentan terhadap serangan. Penelitian ini mengintegrasikan Diffie-Hellman untuk menghasilkan kunci bersama sebagai eksponen publik (e) pada RSA, sehingga pembangkitan kunci jadi lebih dinamis dan aman. Metode meliputi analisis sistem, implementasi gabungan Diffie-Hellman dan RSA, serta pengujian pada file PDF. Data diubah ke format numerik desimal sebelum enkripsi dan dekripsi. Enkripsi disimpan dalam (*.txt), dekripsi mengembalikan file ke PDF asli. Hasil menunjukkan penggunaan kunci bersama meningkatkan keamanan meski ukuran file terenkripsi bertambah sekitar 4-5 kali lipat dari file asli. Sistem ini diharapkan menjadi solusi keamanan PDF dan kriptografi modern yang lebih baik.

ABSTRACT

The development of digital technology poses challenges to data security, especially in frequently used PDF files. Cryptography can be used for data security. The conventional RSA algorithm with a static public key is vulnerable to attacks. This study integrates Diffie-Hellman to generate a shared key as a public exponent (e) in RSA, making key generation more dynamic and secure. The method includes system analysis, a combined implementation of Diffie-Hellman and RSA, and testing on PDF files. Data is converted to decimal numeric format before encryption and decryption. The encryption is stored in (*.txt), decryption returns the file to the original PDF. The results show that the use of a shared key improves security even though the encrypted file size increases by about 4-5 times from the original file. This system is expected to be a better solution for PDF security and modern cryptography.

***Penulis Koresponden**

Email: maliyahmurti@gmail.com

Cara sitasi IEEE:

M. Murti, Rahmadani, & R. Puspadini, "Enhancement of Rivest Shamir Adleman (RSA) Key Generation Utilizing the Diffie-Hellman Algorithm for PDF File Security," *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 3, pp. 1141-1151, September 2025, doi: 10.30811/jaise.v5i3.7629

1. PENDAHULUAN

Kemajuan teknologi informasi telah membawa kemudahan besar dalam pertukaran dan penyimpanan informasi digital, salah satunya melalui format Portable Document Format (PDF) yang populer karena fleksibilitasnya. Namun, keamanan dokumen PDF menjadi perhatian utama karena adanya risiko kelemahan enkripsi yang dapat dimanfaatkan dalam serangan siber modern. Seiring meningkatnya penggunaan dokumen digital, kejahatan siber yang mengeksploitasi celah enkripsi semakin berkembang pesat. Potensi kerugian global akibat serangan siber diperkirakan mencapai USD 10,5 triliun pada tahun 2025[1]. Ancaman ini menegaskan perlunya inovasi dalam pengamanan enkripsi dokumen PDF agar informasi sensitif terlindungi secara efektif dari serangan siber.

Melalui kriptografi, ilmu dan seni yang bertujuan untuk menjaga kerahasiaan suatu pesan[2]. Ada empat tujuan mendasar dari ilmu kriptografi yang menjadi aspek keamanan yaitu kerahasiaan, integritas data, autentikasi, dan Non-repudiasi[3]. RSA merupakan salah satu dari Public Key Cryptosystem yang sangat sering digunakan untuk memberikan kerahasiaan terhadap keaslian suatu data digital. Keamanan enkripsi dan dekripsi data model ini terletak pada kesulitan untuk memfaktorkan modulus n yang sangat besar [4]. Namun, kunci publik RSA yang bersifat tidak rahasia dan mudah diprediksi menjadi celah keamanan yang potensial. Oleh karena itu, diperlukan pendekatan untuk meningkatkan kompleksitas dan dinamika proses pembangkitan kunci publik RSA agar sistem enkripsi menjadi lebih aman dan tahan terhadap serangan.

Penelitian ini mengusulkan penggabungan algoritma Diffie-Hellman dan RSA, di mana kunci bersama yang dihasilkan oleh Diffie-Hellman digunakan sebagai nilai eksponen kunci publik (e) pada RSA. Algoritma Diffie-Hellman dikenal sulit untuk dihitung kembali nilai logaritma diskritnya, sehingga meningkatkan keamanan pertukaran kunci rahasia melalui saluran publik tanpa harus mengirim kunci secara langsung [5] Kombinasi kedua algoritma ini diharapkan menghasilkan kunci RSA yang lebih dinamis dan sulit diprediksi, sehingga menguatkan sistem enkripsi dan dekripsi dokumen PDF.

Penelitian terdahulu mengaplikasikan algoritma RSA untuk mengamankan data digital seperti gambar dengan hasil waktu enkripsi dan dekripsi yang optimal[6] Penelitian lain juga menunjukkan bahwa pemanfaatan algoritma pertukaran kunci seperti Diffie-Hellman dapat meningkatkan keamanan sistem enkripsi file teks maupun citra dengan performa yang dapat diterima[7].

Penelitian terdahulu, Menunjukkan proses pengamanan data gambar menggunakan algoritma RSA dari segi Keamanan dapat berjalan dengan baik dan dapat melakukan tugasnya yaitu mengamankan data gambar dengan baik. Segi keefesienan semakin besar ukuran serta resolusi gambar yang dienkripsi maka semakin lama juga serta semakin berat hasil enkripsi dari gambar tersebut. Gambar grayscale lebih cepat diproses dibandingkan gambar berwarna. Semakin besar panjang kunci RSA maka semakin tinggi tingkat keamanan karena meningkatnya kompleksitas enkripsi[8].

Penelitian ini bertujuan mengembangkan model sistem enkripsi yang mengombinasikan kunci bersama Diffie-Hellman sebagai nilai eksponen kunci public (e) RSA untuk menghasilkan kunci yang lebih dinamis dan sulit diprediksi, sehingga meningkatkan perlindungan terhadap file PDF yang dienkripsi. Metode ini meliputi proses analisis kebutuhan sistem, perancangan dan implementasi perangkat lunak kombinasi algoritma Diffie-Hellman dan RSA, serta pengujian praktis pada berbagai ukuran file PDF. Dengan pendekatan ini, diharapkan sistem enkripsi dapat memproteksi file PDF secara lebih efektif dari serangan siber dan memberikan kontribusi signifikan dalam pengembangan kriptografi modern yang lebih aman pada keamanan data digital.

2. METODE

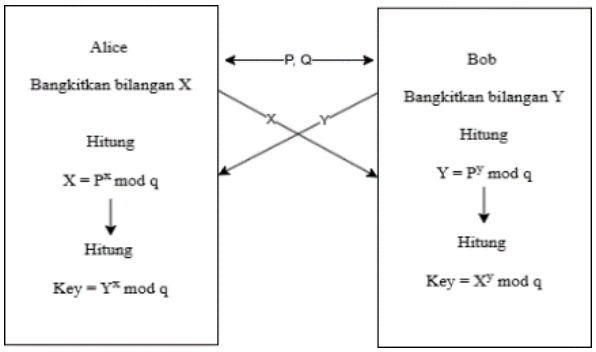
Adapun metode dalam pemecahan masalah pada penelitian ini sebagai berikut:

2.1 Kriptografi

Kriptografi merupakan cabang ilmu matematika yang berfokus pada pengubahan data untuk menyembunyikan maknanya, mencegah perubahan yang tidak sah, atau menghindari penggunaan yang tidak diizinkan. Jika kita melihat konversi, istilah *crypto* juga dapat diartikan sebagai proses mengembalikan data yang telah dienkripsi ke dalam bentuk yang dapat dimengerti. Dengan demikian, kriptografi dapat dilihat sebagai proses perlindungan data secara umum[9]. Kriptografi memiliki dua jenis yaitu kriptografi klasik yang bekerja dalam mode karakter dan kriptografi modern yang bekerja dalam mode bit[10]. Proses utama dalam kriptografi meliputi enkripsi dan dekripsi. Enkripsi mengubah pesan asli atau plaintext menjadi bentuk acak yang tidak dapat dibaca (ciphertext), sedangkan dekripsi melakukan kebalikan dari enkripsi, mengubah data terenkripsi kembali ke bentuk asli yang dapat dipahami. Dalam kriptografi modern, terdapat dua pendekatan utama yaitu algoritma simetris dan asimetris. Algoritma simetris menggunakan satu kunci yang sama untuk enkripsi dan dekripsi, sementara algoritma asimetris menggunakan pasangan kunci publik dan privat yang saling terkait[11].

2.2 Algoritma Diffie-Hellman

Algoritma Diffie-Hellman (DH), dinamai dari penemunya Whitfield Diffie dan Martin Hellman, merupakan sistem kunci asimetris pertama yang diperkenalkan pada tahun 1976. Algoritma ini berfungsi sebagai metode pertukaran kunci yang memungkinkan dua pihak untuk secara aman membentuk kunci rahasia bersama (shared secret key) yang nantinya digunakan untuk enkripsi dan dekripsi pesan. Keamanan DH bergantung pada kesulitan menyelesaikan masalah logaritma diskrit, yang dianggap sulit secara komputasi apabila nilai bilangan prima yang digunakan cukup besar, sehingga menjadikan protokol ini tetap relevan dan dipercaya dalam berbagai sistem keamanan modern[12]. Adapun Gambaran algoritma Diffie-Hellman seperti pada gambar 1. Berikut.



Gambar 1. Algoritma Diffie-Hellman

Berdasarkan Gambar 1. menunjukkan langkah-langkah dalam pertukaran kunci dengan menggunakan algoritma Diffie-Helman sebagai berikut:[13]

1. Pilih bilangan (p) yang merupakan bilangan prima besar dan bilangan (q), q biasa disebut sebagai bilangan basis atau generator, sebagai bilangan yang tidak melebihi nilai pada p.
2. Pilih bilangan acak pada pengirim (a) yang tidak dapat diketahui oleh orang lain.
3. Pilih bilangan acak pada penerima (b) yang tidak diketahui oleh orang lain.
4. Pengirim menghitung (A = ga mod p). Bilangan yang terdapat pada A merupakan kunci publik.
5. Penerima menghitung (B = gb mod p). Bilangan yang terdapat pada B merupakan kunci publik.
6. Lakukan pertukaran bilangan A dan B terhadap pengirim dan penerima.
7. Kemudian pengirim menghitung (KA = Ba mod p).
8. Kemudian penerima menghitung (KB = Ab mod p).
9. Berdasarkan hitungannya di dapatkan KA=KB yang merupakan kunci bersama rahasia.

Adapun penggunaan Algoritma Diffie-Hellman sebagai berikut:

1. p = 53 ; g = 47
2. a = 23 (pengirim) ; b = 40 (penerima)
3. A = ga mod p = 4723 mod 53 = 13
4. B = gb mod p = 4740 mod 53 = 47
5. Lakukan pertukaran bilangan A dan B terhadap pengirim dan penerima.
6. KA = Ba mod p = 4723 mod 53 = 13
7. KB = Ab mod p = 1340 mod 53 = 13
8. Berdasarkan hitungannya di dapatkan KA=KB yang merupakan kunci bersama rahasia yaitu 13.

Hasil algoritma Diffie-Hellman digunakan untuk pembangkitan kunci nilai eksponen (e) kunci publik RSA yang nantinya akan digunakan pada proses enkripsi dan dekripsi menggunakan algoritma RSA

2.3 Algortima Rivest Shmair Adleman (RSA)

RSA (Rivest Shamir Adleman) adalah algoritma kriptografi asimetris yang banyak digunakan dan dikenal karena keamanan matematika. Keunggulan RSA adalah sulitnya memecahkan kunci privatnya, yang didasarkan pada sulitnya memfaktorkan hasil perkalian dua bilangan prima besar. Pada RSA, sepasang kunci terdiri dari kunci publik (e, n) yang dapat didistribusikan secara bebas, dan kunci privat (d, n) yang harus dirahasiakan oleh pemiliknya [6]. Algoritma RSA adalah sebuah algoritma enkripsi blok yang bekerja dengan cara membagi pesan asli (plaintext) menjadi beberapa blok data terlebih dahulu sebelum dienkripsi menjadi pesan terenkripsi (ciphertext). Setiap blok plaintext diubah menjadi bilangan biner yang nilainya harus kurang dari suatu bilangan nn. Oleh karena itu, ukuran setiap blok harus lebih

kecil atau sama dengan log2(n) agar proses enkripsi dapat dilakukan dengan benar[14]. Proses enkripsi pada RSA menggunakan rumus $C=M^e \bmod n$, di mana M adalah pesan asli dalam bentuk numerik dan C adalah pesan terenkripsi. Sebaliknya, proses dekripsi menggunakan rumus $M=C^d \bmod n$, untuk mendapatkan kembali pesan asli. Penggunaan RSA dalam pengamanan file digital sudah banyak diterapkan, termasuk dalam enkripsi data teks, gambar, dan dokumen lainnya. Terdapat 3 proses pada algoritma RSA yaitu pembangkitan kunci, proses enkripsi dan proses dekripsi

2.2.1 Pembangkitan Kunci

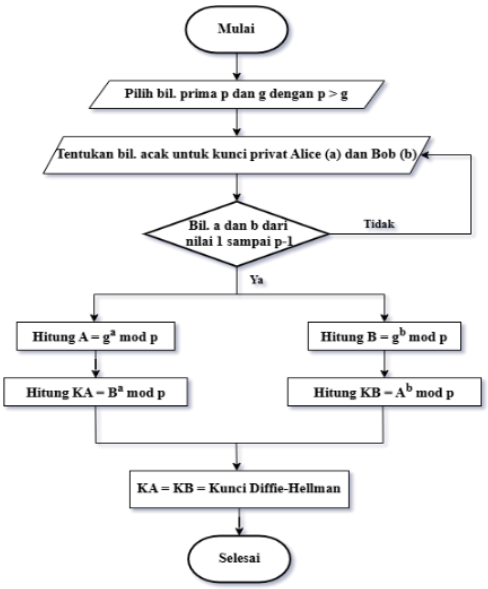
Proses pembangkitan kunci yang melibatkan pembangkitan publik dan privasi. Kunci publik adalah kunci terbuka yang dapat dilihat oleh semua orang, dan digunakan untuk berpartisipasi dalam enkripsi pesan. Pesan yang dikirimkan akan dienkripsi menggunakan kunci publik, dan setelah itu dapat didekripsi menggunakan kunci privat. Adapun langkah-langkah berikut:[6]

1. Pilih dua bilangan prima secara acak p dan q. Dua bilangan ini bersifat rahasia.
2. Hitung nilai modulus n yang merupakan perkalian p x q. nilai n bersifat tidak rahasia. n = p x q.
3. Hitung nilai fungsi $\phi(n)$. Nilai $\phi(n)$ tetap dirahasiakan karena menjadi bagian penting dalam pembentukan kunci privat. $\phi(n) = (p - 1)(q - 1)$.
4. Tentukan bilangan bulat e sehingga $1<e<\phi(n)$ dan $\gcd(e, \phi(n)) = 1$, dimana relatif prima terhadap $\phi(n)$. Nilai e bersifat tidak rahasia.
5. Hitung nilai d sebagai inverse modular dari e terhadap $\phi(n)$. Nilai d bersifat rahasia dan menjadi bagian dari kunci privat dengan syarat: $d \equiv e^{-1} \bmod \phi(n)$ atau $d * e \bmod \phi(n) = 1$.
6. Maka didapatlah pasangan kunci publik dan kunci privat yaitu: Kunci publik terdiri dari pasangan (e,n) dan kunci privat terdiri dari pasangan (d,n).

Adapun pembangkitan kunci RSA yang di kombinasikan dengan Diffie-Hellman sebagai berikut :

1. Pilih bilangan prima acak p dan q. Misalnya:
   $p = 47$; $q = 71$
2. Hitung nilai modulus n
   $n = p * q = 47 * 71 = 3337$
3. Hitung nilai $\phi(n)$
   $\phi(n) = (p - 1)(q - 1) = (47-1)(71 - 1) = (46)(70) = 3220$
4. Menetapkan nilai bilangan bulat e dari kunci Diffie-Hellman. Ambil nilai kunci bersama rahasia Diffie-Hellman sebagai nilai e dengan $1 < e < \phi(n)$ dan $GCD(\phi(n), e) = 1$, dimana relatif prima terhadap $\phi(n)$.
   $e = 13$, $GCD(3220, 13) = 1$. → Nilai e ditentukan berdasarkan hasil kunci rahasia bersama Diffie-Hellman
5. Hitung d, $d * e \bmod \phi(n) = 1$
   $2477 * 13 \bmod 3220 = 1$
   $32201 \bmod 3220 = 1$
   Sehingga, nilai d = 2477
6. Didapat lah pasangan kunci publik dan privat yaitu:
   Pasangan kunci publik (e, n) = (13, 3337)
   Pasangan kunci privat (d, n) = (2477, 3337)

Adapun Flowchart Algoritma Diffie-Hellman pada gambar 2. berikut.



Gambar 2. Flowchart Diffie-Hellman

2.2.2 Enkripsi

Enkripsi adalah proses mengubah pesan asli (plaintext) menjadi bentuk yang tidak dapat dibaca secara langsung tanpa akses ke kunci yang tepat[15]. Proses enkripsi dilakukan dengan rumus:

$$C = M^e \bmod n$$

Keterangan:

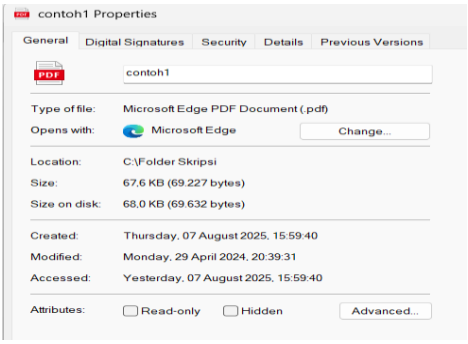
M = Pesan Awal (Plaintext)

C = Pesan Terenkripsi (Chipertext)

e = Eksponen Kunci Publik

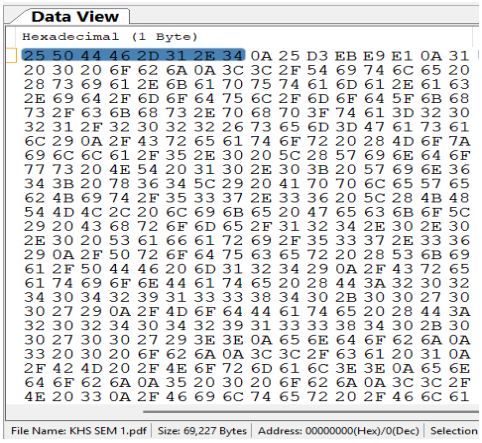
n = Nilai Modulus.

Data File pdf akan dienkripsi terlebih dahulu dikonversi ke format numerik (desimal) agar dapat diproses. Sistem kemudian melakukan enkripsi dan dekripsi secara berurutan untuk menjaga keamanan dan integritas data pada file pdf. Contoh file pdf yang digunakan adalah size 67.6 KB yang dapat dilihat pada Gambar 3.



Gambar 3. Ukuran File Pdf Asli

Pada gambar 3. Menunjukkan ukuran file pdf asli. Sebelum di proses ke langkah enkripsi dan dekripsi, file tersebut akan diambil nilai heksadesimalnya terlebih dahulu sebanyak 8 bytes sebagai sampel menggunakan software Binary Viewer seperti yang ada pada gambar 4. berikut:



Gambar 4. Hexadesimal Dari File Asli

Dari Gambar 4 dapat diambil 8 bytes *ASCII* hexadesimal dan di konversi ke *ASCII* decimal. Nilai desimal ini yang akan dijadikan sebagai sampel metode. Adapun nilai hexadecimal tersebut yaitu: [**25 50 44 46 2D 31 2E 34].** Nilai heksadesimal tersebut akan di konversi kedalam bilangan desimal terlebih dahulu agar dapat diproses ke enkripsi dan dekripsi menjadi [**37 80 68 70 45 49 46 52].**

Proses enkripsi *file* menggunakan kunci publik yang telah dihasilkan dengan rumus : $C = P^e \text{ Mod } n$.

$C1 = M^e \bmod n = 37^{13} \bmod 3337 = 1184$

$C2 = M^e \bmod n = 80^{13} \bmod 3337 = 1389$

$C3 = M^e \bmod n = 68^{13} \bmod 3337 = 195$

$C4 = M^e \bmod n = 70^{13} \bmod 3337 = 1561$

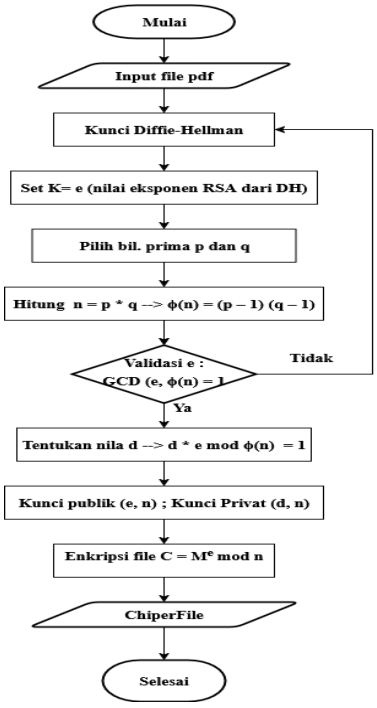
$C5 = M^e \bmod n = 45^{13} \bmod 3337 = 456$

$C6 = M^e \bmod n = 49^{13} \bmod 3337 = 1565$

$C7 = M^e \bmod n = 46^{13} \bmod 3337 = 563$

$C8 = M^e \bmod n = 52^{13} \bmod 3337 = 184$

Didapatkan chiperfile yang dihasilkan dari proses enkripsi yaitu **1184 1389 195 1561 456 1565 563 184.** Adapun flowchart proses enkripsi seperti pada gambar 5. Berikut.



Gambar 5. Flowchart Enkripsi

2.2.3 Dekripsi

Dekripsi adalah proses penting dalam algoritma RSA, yang bertugas mengembalikan pesan terenkripsi (Chipertext) ke bentuk aslinya (plaintext)[15]. Proses dekripsi dilakukan dengan rumus:

$$\mathbf{M = C^d \bmod n}$$

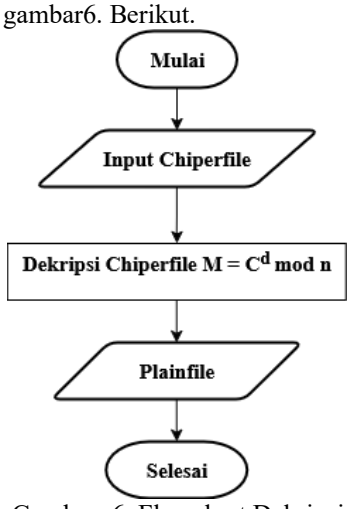
Keterangan :

d = Eksponen Kunci Privat.

Didapatkan dari hasil enkripsi yaitu chiperfile berisikan angka desimal: **1184 1389 195 1561 456 1565 563 184** yang akan di dekripsi menjadi bentuk semula yaitu plainfile.

M1 = C^d mod n = 1184^{2477} mod 3337 = 37
M2 = C^d mod n = 1389^{2477} mod 3337 = 80
M3 = C^d mod n = 195^{2477} mod 3337 = 68
M4 = C^d mod n = 1561^{2477} mod 3337 = 70
M5 = C^d mod n = 456^{2477} mod 3337 = 45
M6 = C^d mod n = 1565^{2477} mod 3337 = 49
M7 = C^d mod n = 563^{2477} mod 3337 = 46
M8 = C^d mod n = 184^{2477} mod 3337 = 52

Maka didapat hasil dekripsi pada *file* yang berisikan angka desimal adalah : **37 80 68 70 45 49 46 52.** Kemudian langkah terakhir yang harus dilakukan adalah mengambalikan semua bilangan desimal tersebut ke heksadesimal agar semua bytenya kembali ke nilai aslinya menjadi: [**25 50 44 46 2D 31 2E 34].** Adapun flowchart proses dekripsi seperti pada gambar6. Berikut.



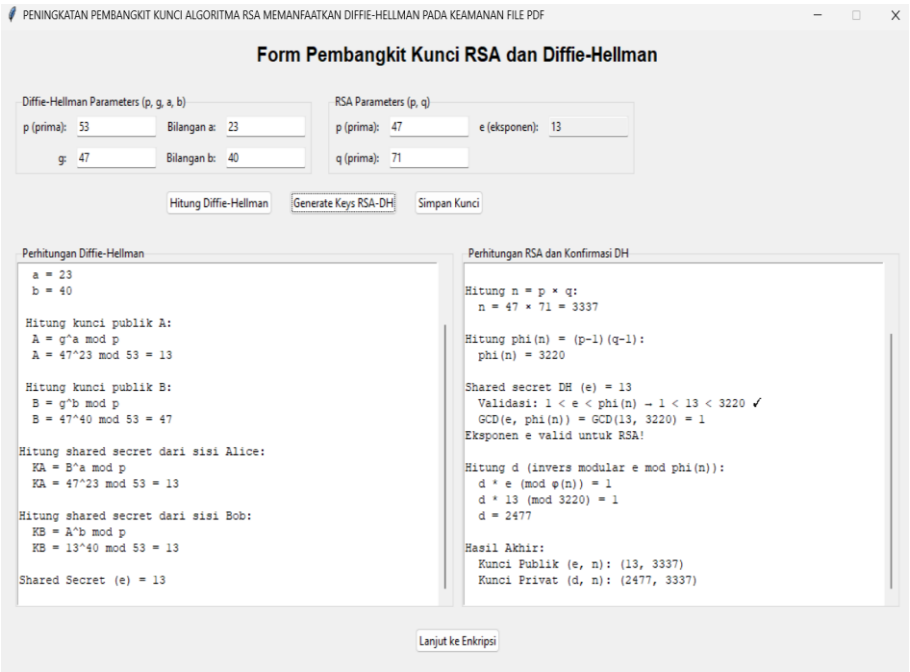
Gambar 6. Flowchart Dekripsi

3. HASIL DAN PEMBAHASAN

Implementasi dilakukan dengan menggunakan bahasa pemrograman python. Dimulai dengan mengonversi size file pdf menjadi nilai decimal, lalu di enkripsi dan dekripsi file pdf. Hasil enkripsi disimpan dalam format (*.txt) karena ciphertext RSA berupa deretan angka desimal besar, yang lebih mudah disimpan dan dikelola. Ukuran file meningkat 4-5 kali lipat setelah enkripsi, misalnya file 67,6 KB menjadi 308 KB, lalu kembali ke 67,6 KB setelah dekripsi. Hal ini terjadi karena RSA mengubah data asli menjadi angka desimal yang melebihi 1 byte (maks. 255). Proses dekripsi membaca tiap angka untuk mengembalikan data asli tanpa kehilangan, menunjukkan integritas algoritma. Penggunaan Diffie-Hellman sebagai pembangkit kunci publik nilai eksponen (e) RSA menambah keamanan lewat pertukaran kunci dinamis.

3.1 Pembangkit Kunci

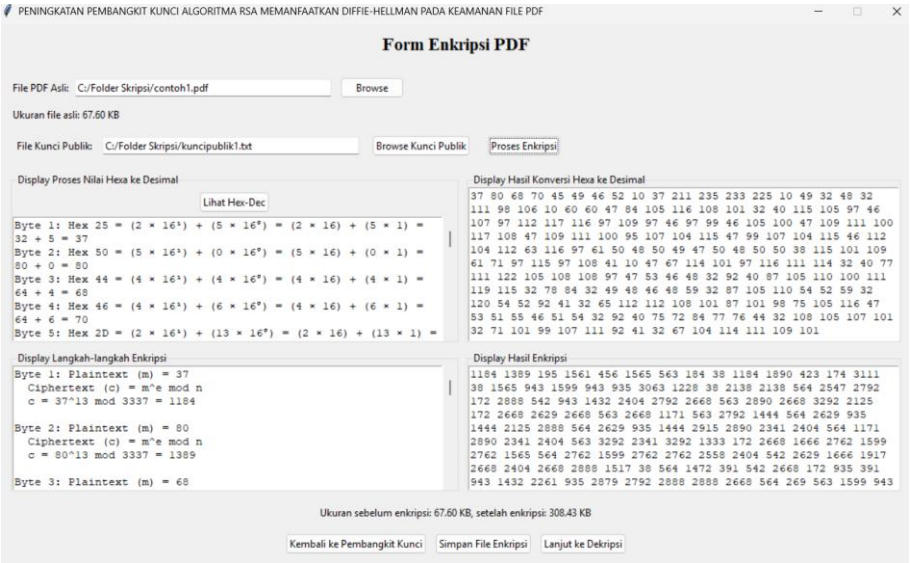
Pada halaman ini, proses pembangkitan kunci RSA dan Diffie-Hellman dilakukan dengan menginput nilai p, g, a, dan b untuk Diffie-Hellman, lalu klik tombol hitung Diffie-Hellman. Hasilnya otomatis menjadi input eksponen (e) RSA dan riwayat perhitungan tampil pada display Diffie-Hellman. Selanjutnya, input nilai p dan q RSA, klik tombol generate RSA-DH yang terintegrasi dengan hasil Diffie-Hellman. Kunci publik dan privat yang dihasilkan dapat disimpan dalam format (*.txt) dengan tombol simpan kunci. Hasil generate RSA-DH ditampilkan pada display perhitungan RSA dan konfirmasi DH. Adapun tampilan pembangkit kunci pada gambar 7. Berikut.



Gambar 7. Form Pembangkit Kunci RSA- DH

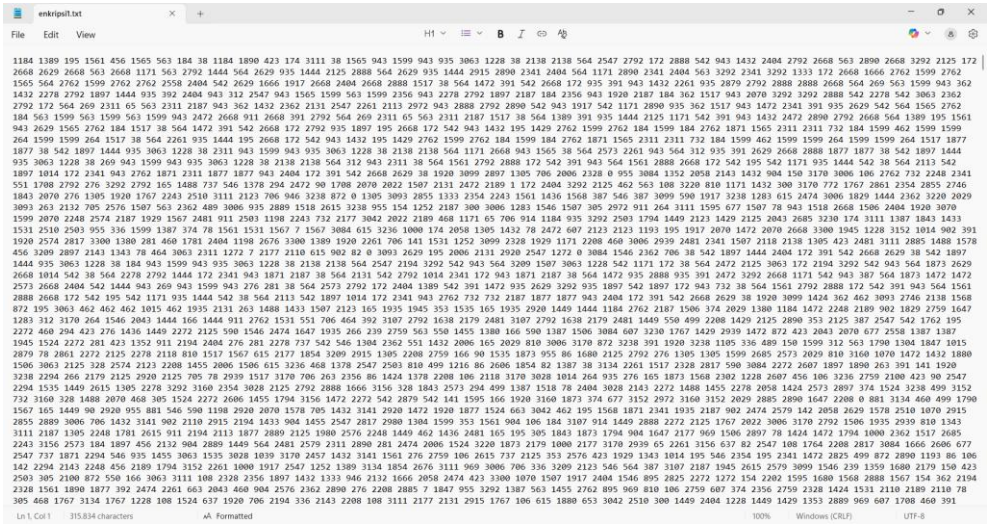
3.2 Enkripsi

Sebelum enkripsi, pilih memilih file berformat (*.pdf) (ukuran 67.6 KB). Klik tombol konversi hex-des untuk melihat nilai desimal dari data file yang diinput, Kemudian hasil konversi (maksimal 200 byte) tampil di display. Selanjutnya, input kunci publik (.txt) dan klik tombol proses enkripsi. Langkah enkripsi ditampilkan di display bersama hasilnya (hanya 200 byte pertama). Simpan file hasil enkripsi (*.txt) dengan klik tombol simpan. Hasil enkripsi berupa file (*.txt) dengan ukuran 308 KB. Adapun tampilan Proses enkripsi seperti pada gambar 8. Berikut.



Gambar 8. Form Enkripsi

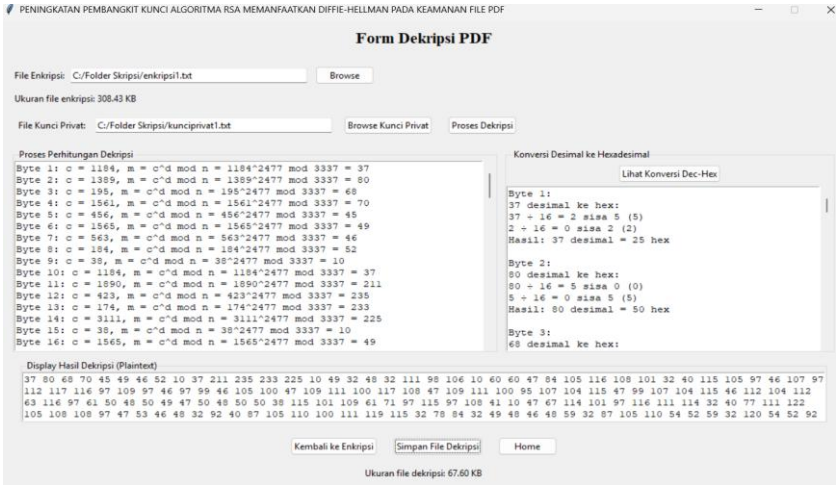
Hasil enkripsi yang disimpan dalam format (*.txt) berupa nilai-nilai yang acak. Adapun tampilan isi dari file terenkripsi seperti pada gambar 9. Berikut.



Gambar 9. Hasil Enkripsi

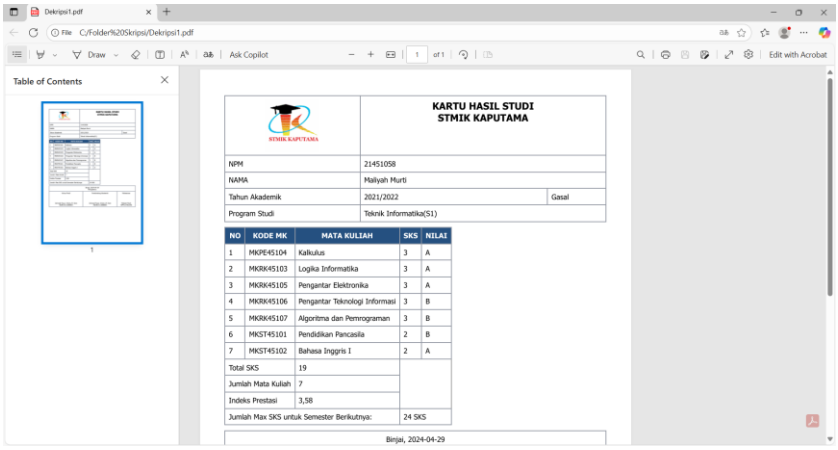
3.3 Dekripsi

Tahap selanjutnya, proses dekripsi dengan pilih file terenkirpsi kemudian pilih kunci privat dan proses dekripsi. Langkah dekripsi akan tampil di display bersama hasilnya. Simpan hasil dekripsi dengan klik tombol simpan file dekripsi. Hasil dekripsi berupa nilai decimal. Untuk melihat konversi heksadesimal, klik tombol lihat konversi dec-hex. Proses dekripsi menghasilkan file (*.pdf) berukuran 67.6 KB. Adapun tampilan dari proses dekripsi seperti pada gambar 10. Berikut.



Gambar 10. Form Enkripsi

Hasil file yang sudah didekripsi kembali seperti file asli tanpa mengurangi perubahan data. Adapun hasil dari proses dekripsi dapat dilihat pada gambar 11. Beriku.



Gambar 11. Hasil Dekripsi

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa sistem kriptografi yang mengkombinasikan RSA dan Diffie-Hellman dapat di implementasikan dengan baik. Penerapan nilai eksponen kunci publik (e) RSA yang dihasilkan dari kunci bersama rahasia Diffie-Hellman dapat menghasilkan kunci yang dinamis dan sulit diprediksi dibandingkan algoritma RSA standart, sehingga dapat meningkatkan keamanan pada proses enkripsi dan dekripsi.

Hasil uji coba sistem menunjukan bahwa file berformat PDF dapat dienkripsi dengan baik sehingga, tidak bisa dibaca. Kemudian dilakukan proses dekripsi, file berhasil di kembalikan seperti bentuk semula dan dapat dibuka serta dibaca tanpa mengalami kerusakan data. Hal ini membuktikan bahwa sistem yang dirancang mampu menjaga keamanan data file tanpa mengubah isi aslinya.

Pengujian juga memperlihatkan bahwa ukuran file hasil enkripsi lebih besar dibandingkan file asli, tedapat peningkatan 4 kali lebih besar. Hal ini disebabkan oleh karakteriktik RSA yang mengubah data asli menjadi chipertext berbentuk deretan angka decimal lebih dari 255 (maksimal 1 byte), sehingga menambah ukuran byte. Hasil file yang terenkripsi disimpan dalam format (*.txt) dan dikembalikan seperti asli pada format (*.pdf) melalui proses dekripsi.

Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi dalam pembangkitan kunci algoritma RSA dengan memanfatkan algoritma Diffie-Hellman sebagai sumber kunci untuk nilai eksponen (e) pada RSA dalam menjaga keamanan data.

REFERENSI

[1] S. Shahrear, S. S. Tinny, and K. Mohammad, “Analysis of Identification of Cybercrimes Using Cyber Security Analysis of Identification of Cybercrimes Using Cyber Security Analytics Powered By Artificial,” no. August, 2024.

[2] P. Eva, F. Achmad, and S. Milli Alfhi, “Digital Image Security Implementation With Uses Super Encryption Algorithm Myszkowski And The Algorithm Paillier Cryptosystem,” *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 3, no. 1, pp. 70–82, 2023, doi: 10.59934/jaiea.v3i1.262.

[3] M. M. I. Baharsyah, “Implementasi Algoritma RSA dalam Enkripsi dan Dekripsi File Teks,” 2023.

[4] U. Indriani, O. Alfina, and N. Syahputri, “Penerapan Algoritma RSA Dalam Keamanan File Ms Word,” *Journal of Machine Learning and Data Analytics (MALDA)*, vol. 01, no. 02, pp. 95–100, 2021, [Online]. Available: http://repository.potensi-utama.ac.id/jspui/handle/123456789/5074

[5] R. Prastya, A. M. H. Pardede, and A. Fauzi, “Teknik Pembangkit Kunci Algoritma RSA Menggunakan Algoritma Diffie Hellman pada Keamanan Citra,” *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, vol. 04, no. 01, pp. 16–22, 2022, doi: 10.54367/kakifikom.v4i1.1872.

[6] A. Sahoo, P. Mohanty, and P. C. Sethi, “Image Encryption Using RSA Algorithm,” *Lecture Notes in Networks and Systems*, vol. 431, no. May, pp. 641–652, 2022, doi: 10.1007/978-981-19-0901-6_56.

[7] L. Nisa, T. Indriyani, and M. Ruswiansari, “Aplikasi Enkripsi Citra dan Teks Menggunakan Algoritma Diffie-Hellman dan ElGamal,” *Jurnal Teknologi dan Manajemen*, vol. 1, no. 1, pp. 8–17, 2020.

[8] C. Repi, J. Titaley, and E. Ketaren, “Implementasi Kriptografi Dalam Pengamanan Data Gambar Menggunakan Algoritma Rsa,” *Jurnal TIMES*, vol. 13, no. 1, pp. 93–99, 2024, doi: 10.51351/jtm.13.1.2024750.

[9] Mariza, A. Fauzi, and Y. Maulita, “Rivest Shamir Adleman (RSA) Super Encryption Algorithm with Vigenere Cipher Algorithm Modification for Image Security,” vol. 4, no. 1, 2024.

[10] S. P. Ananda and S. Lukman, “Analisa Metode Kriptografi Modern Advance Encryption Standard (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital,” *Jurnal Ilmiah Komputasi*, vol. 21, no. 3, pp. 333–344, 2022, doi: 10.32409/jikstik.21.3.2973.

[11] N. A. Nanda, S. M. S. Silalahi, D. Fatricia Nasution, M. Sari, and I. Gunawan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," *Jurnal Media Informatika*, vol. 4, no. 2, pp. 90–93, 2023, doi: 10.55338/jumin.v4i2.428.

[12] S. F. Yousif, "Secure voice cryptography based on Diffie-Hellman algorithm," *IOP Conf Ser Mater Sci Eng*, vol. 1076, no. 1, p. 012057, 2021, doi: 10.1088/1757-899x/1076/1/012057.

[13] R. G. Sinambela, A. Fauzi, and H. Khair, "Enhancing AES Key Generation Using Diffie-Hellman Method for Image Security," vol. 3, no. 3, pp. 2–7, 2024.

[14] D. Pratama, "Implementasi algoritma rsa untuk pengamanan data berbentuk teks," no. February 2016, 2021, doi: 10.33369/pseudocode.3.1.44-49.

[15] M. A. Ritonga and D. Nofriansyah, "Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Aplikasi Verifikator Keaslian Ijazah Berbasis Quick Response (QR) Code Dengan Algoritma RSA Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD," vol. 7, pp. 246–256, 2024.