Jurnal Quancom
Vol.3, No.2, Desember 2025, pp. 26~34
e-ISSN: 2988-0998
P-ISSN: 2988-1005
DOI : https://doi.org/10.62375/jqc.v3i2.424

Hybrid Cryptosystem dengan Algoritma 3DES dan AES dalam Pengamanan File Text

Yulanda Pasaribu[1], David Kristian Silalahi[2], Vlen Jeremy Simanjuntak[3]
[1,2,3]Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del
pasaribuyulanda94@gmail.com

Article Info

Article history:

Received 14 November 2025
Revised 03 Desember 2025
Accepted 20 Desember 2025

Keyword:

AES, Cryptography, Enkripsi, Triple DES, Keamanan

ABSTRACT

This research aims to improve the security of text data by implementing a hybrid cryptosystem that combines the 3DES and AES algorithms, both of which are symmetric cryptographic algorithms that have proven to be effective. This research utilizes the Python programming language and cryptography library to implement the algorithm with CBC mode of operation and PKCS7 padding. The encryption process begins by randomly generating a symmetric key for each session, which is then encrypted using the recipient's public key. The 3DES algorithm is implemented to provide additional protection through three rounds of encryption, while AES is used for efficiency and speed in data processing. In the decryption stage, the recipient uses the private key to unlock the encrypted symmetric key, and then uses the key to decrypt the encrypted data. The results obtained show that the combination of these two algorithms increases resistance to attacks such as brute force and chosen plaintext, providing an optimal balance between efficiency and security in securing text data. With these results, the hybrid cryptosystem approach can be applied in various applications that require text data protection, offering a robust and reliable security solution in the secure exchange of information.

PENDAHULUAN

Pertukaran informasi dalam lingkungan digital menuntut perlindungan data yang efektif dan handal. Seiring dengan perkembangan teknologi, kebutuhan akan sistem keamanan yang mampu menghadapi berbagai ancaman keamanan siber menjadi mendesak. Oleh karena itu, penelitian ini bertujuan untuk mengatasi masalah kerentanan data teks terhadap serangan siber dengan salah satu pendekatan yang banyak digunakan, yaitu kriptografi simetris, yang menyediakan enkripsi data dengan satu kunci yang sama dalam proses enkripsi dan dekripsi. Pengimplementasian dua algoritma dalam Hybrid Cryptosystem berbasis 3DES dan AES diharapkan dapat meningkatkan keamanan dalam pengiriman data sensitif melalui jaringan yang tidak aman. Penggunaan kriptografi simetris untuk pengamanan data sangat relevan, terutama dalam konteks data yang perlu diproses dengan cepat dan efisien, seperti pada pengamanan file teks. 3DES dan AES, dua algoritma kriptografi simetris yang banyak digunakan, masing-masing menawarkan kekuatan dalam mengamankan data, dengan 3DES memberikan keamanan yang lebih baik daripada DES dan AES menawarkan efisiensi serta kecepatan dalam pengolahan data [1].

Latar belakang penelitian ini muncul dari kesadaran akan kerentanan data dalam pertukaran informasi digital. Oleh karena itu, paper ini secara kritis mengidentifikasi masalah keamanan data dalam konteks pertukaran informasi dan menetapkan dasar-dasar teoritis melalui eksplorasi kriptografi simetris. Penggunaan literatur terkait menjadi landasan untuk memahami perkembangan terbaru dan kecanggihan algoritma kriptografi yang digunakan. Hal ini memperkuat dasar teoritis penelitian, memastikan

pemahaman mendalam terhadap aspek keamanan informasi yang ingin diatasi. Seperti yang dikemukakan oleh [2], dalam penelitian mereka, penggunaan algoritma kriptografi yang tepat sangat penting untuk membangun sistem keamanan yang efektif, sehingga mengurangi potensi ancaman terhadap data yang ditransmisikan.

Paper ini mencoba memberikan solusi konkret dengan merinci implementasi Hybrid Cryptosystem yang menggabungkan algoritma Triple DES (3DES) dan Advanced Encryption Standard (AES) untuk melindungi file teks. Pendekatan ini diusulkan sebagai respons terhadap kebutuhan akan keseimbangan antara efisiensi dan keamanan dalam pertukaran informasi. Sebagaimana yang dikemukakan oleh [1], algoritma 3DES dan AES masing-masing menawarkan keuntungan dalam hal keamanan dan efisiensi dalam pengolahan data, yang mendasari penggunaan kedua algoritma ini dalam penelitian ini. Melalui tinjauan literatur dan pembahasan konsep dasar, paper ini memperkenalkan nilai inovatif dengan menyajikan solusi terintegrasi yang mampu memberikan tingkat keamanan yang tinggi dalam konteks pengamanan file teks.

Pada akhirnya, penelitian ini memberikan nilai tambah dengan menyajikan solusi inovatif dalam bentuk Hybrid Cryptosystem. Inovasi ini muncul sebagai tanggapan terhadap kebutuhan mendesak akan perlindungan data yang andal dalam pertukaran informasi digital. Dengan memperkenalkan keseimbangan antara efisiensi dan keamanan melalui penggabungan 3DES dan AES, yang telah terbukti efektif dalam meningkatkan keamanan data [1], paper ini memberikan kontribusi signifikan terhadap perkembangan teknologi keamanan informasi. Kesimpulannya, penelitian ini diharapkan dapat mendukung evolusi positif dalam upaya menghadapi tantangan keamanan siber di era digital [2].

METODE

A. *Desain Penelitian*

Penelitian ini mengusung pendekatan eksperimental guna menguji keefektifan penggunaan algoritma enkripsi. Desain penelitian melibatkan penggunaan dua algoritma utama, yaitu Triple DES (Data Encryption Standard) dan AES (Advanced Encryption Standard). Dalam literatur yang dikemukakan oleh [3], dijelaskan bahwa popularitas dan keandalan kedua algoritma ini dalam dunia keamanan informasi sudah terbukti, sehingga keduanya dipilih sebagai objek penelitian. Triple DES dikenal dengan tingkat keamanan yang lebih tinggi dibandingkan DES, berkat penggunaan tiga ronde enkripsi. Sementara itu, AES, yang telah diakui sebagai standar enkripsi paling canggih di berbagai aplikasi, menawarkan kecepatan dan efisiensi yang dibutuhkan dalam pengolahan data yang besar dan kompleks [4].

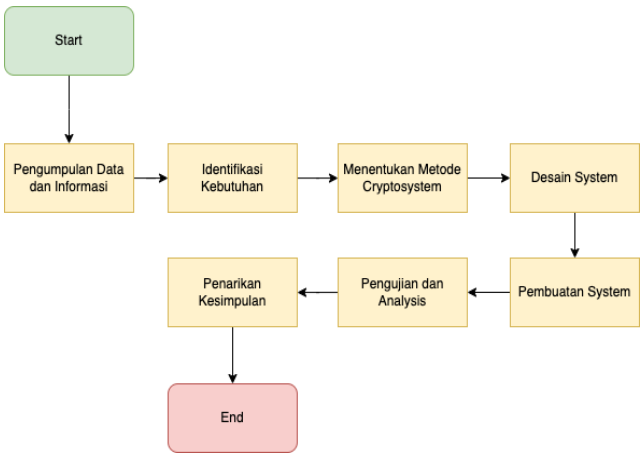
Kombinasi dari kedua algoritma ini diuji dengan tujuan untuk mengeksplorasi apakah penggabungan tersebut dapat menghasilkan tingkat keamanan yang lebih baik dalam pengamanan file teks. Seperti yang dijelaskan dalam penelitian [3], penggunaan Triple DES dan AES secara terpisah maupun digabungkan telah menunjukkan hasil yang baik dalam menanggulangi serangan kriptanalisis dan brute force. Penelitian ini bertujuan untuk menguji kekuatan kedua algoritma ini dalam konteks serangan-serangan tersebut serta untuk mengevaluasi kinerja dan efisiensi enkripsi dalam menjaga kerahasiaan data teks.

Penggunaan mode operasi seperti CBC (Cipher Block Chaining) dan teknik padding PKCS7 memainkan peran penting dalam proses pengamanan file teks yang diuji. Mode CBC menambahkan ketergantungan antar blok data, yang meningkatkan kerahasiaan dengan menciptakan ketidakpastian pada ciphertext. Di sisi lain, PKCS7 memastikan data dengan panjang yang tidak sesuai blok dapat diproses dengan benar, sehingga menghindari kebocoran informasi pada blok terakhir. Patil et al. (2016) menjelaskan bahwa penerapan mode operasi dan teknik padding yang tepat memberikan perlindungan tambahan terhadap serangan seperti brute force dan serangan menggunakan plain text yang sudah dikenal [5].

Kombinasi ini juga diharapkan mampu menghasilkan efek avalanche yang kuat, di mana perubahan kecil pada input plaintext menghasilkan perubahan besar pada ciphertext, sehingga mengurangi risiko pola-pola tertentu dalam data terenkripsi yang dapat dianalisis oleh pihak tidak berwenang.

B. *Prosedur Penelitian*

Tahapan yang dilakukan dalam rangka melakukan penelitian Hybrid Cryptosystem dengan Algoritma 3DES dan AES dalam Pengamanan File Text dapat dilihat pada Gambar1.



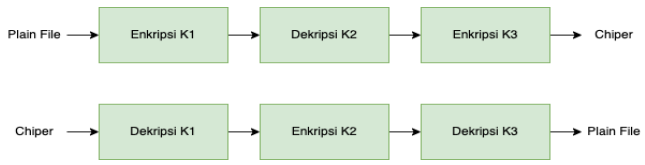
Gambar 1. Langkah-langkah penelitian

C. Algoritma TripleDES(3DES)

Pada tahun 1976, Biro Standar Nasional menetapkan DES (Data Encryption Standard), salah satu metode enkripsi yang diakui di dunia. DES merupakan salah satu algoritma kriptografi simetris dengan tipe cipher block. Enkripsi simetris adalah algoritma simetris yang menggunakan kunci yang sama selama proses enkripsi dan dekripsi. Cipher Block merupakan salah satu jenis enkripsi asimetris yang memiliki ukuran bit tetap atau spesifik, yaitu 64 bit dalam kasus DES [6].

TripleDES merupakan evolusi dari algoritma Double DES yang sudah ada sebelumnya untuk meningkatkan keamanan DES [7]. Algoritma Triple DES menggunakan tiga kunci pada proses enkripsi dan dekripsi. Variasi kunci triple DES dapat dibagi menjadi tiga kategori: kunci yang sama, dua kunci berbeda, atau tiga kunci berbeda. Wahyuni dkk. (2020) menjelaskan bahwa Triple DES terbukti efektif dalam menjaga keamanan data sensitif pada sistem perkantoran, seperti pada pengimplementasian di kantor Walikota Pematangsiantar [8]. Selain itu, penerapan algoritma Triple DES banyak dimanfaatkan dalam peningkatan keamanan data yang penting dalam berbagai sektor, termasuk pada perlindungan basis data hukum dan sistem yang menerapkan kemiliteran, seperti yang dicontohkan dalam penelitian Zalukhu, dkk. (2018) , yang menggunakan keunggulan AES dalam pengamanan data pelanggaran hukum [9].

Keamanan yang diberikan dari algoritma ini membuat Triple DES mencakup ke relevan di dalam berbagai kasus tertentu yang mana meskipun algoritma maju, seperti Advanced Encryption Standard (AES), telah banyak digunakan dalam bidang peningkatan efisiensi dan kecepatan proses enkripsi [10] [11].



Gambar 2. Alur enkripsi dekripsi pada algoritma 3des

Berdasarkan Gambar 2, dapat diketahui detail proses enkripsi 3DES yang dilakukan adalah sebagai berikut :

1) Enkripsi dilakukan terhadap naskah asli dengan menggunakan kunci K1.
2) Dekripsi dilakukan terhadap hasil pertama dengan menggunakan kunci K2.
3) Enkripsi dilakukan terhadap hasil kedua dengan menggunakan kunci K3.

Pada penelitian ini dibangun proses menggunakan algoritma 3DES yang dapat dilihat pada Gambar 3 berikut.

```
# Enkripsi dengan TripleDES
def encrypt_3des(key, data):
    cipher = DES3.new(key, DES3.MODE_CBC)
    iv = cipher.iv
    ciphertext =
cipher.encrypt(pad(data.encode('utf-8'),
DES3.block_size))
    return iv + ciphertext

# Dekripsi dengan TripleDES
def decrypt_3des(key, data):
    iv = data[:DES3.block_size]
    ciphertext = data[DES3.block_size:]
    cipher = DES3.new(key, DES3.MODE_CBC,
iv=iv)
    decrypted_data =
unpad(cipher.decrypt(ciphertext),
DES3.block_size)
    return decrypted_data.decode('utf-8')
```

Gambar 3. Code implementasi 3des

D. Algoritma AES

AES merupakan penerus algoritma enkripsi standar DES (Data Encryption Standard) yang dianggap ketinggalan zaman karena faktor keamanan. Kecepatan komputer yang sangat cepat dianggap terlalu berbahaya bagi DES, sehingga algoritma baru Rijndael disebut AES [4]. Kriteria pemilihan AES didasarkan pada tiga kriteria utama, yaitu keamanan, biaya, dan karakteristik algoritma serta implementasinya [12].

Keamanan adalah faktor terpenting dalam evaluasi algoritma enkripsi, dengan persyaratan bahwa algoritma ini harus setidaknya sama amannya dengan Triple DES. Ketahanan terhadap semua analisis sandi yang diketahui, serta kemampuan untuk menghadapi analisis sandi yang belum diketahui, menjadi aspek kunci dalam penilaian ini. Hal ini didukung oleh ketahanan AES dalam menghadapi berbagai ancaman enkripsi modern.

AES juga tercatat sangat efektif dalam implementasi enkripsi dan dekripsi file. Teguh Utomo dan Pradana (2020) menyoroti bahwa AES-128 tidak hanya mampu menjaga keamanan data dengan baik tetapi juga mudah diterapkan di berbagai platform [13]. Selain itu, Ravida dan Santoso (2020) menunjukkan bahwa AES-128 menyediakan keamanan data yang optimal untuk Internet of Things (IoT), khususnya dalam aplikasi sistem tanaman hidroponik yang memerlukan efisiensi dan kecepatan tinggi dalam pengolahan data [14]. Melenia Bayu Aryanto dkk. (2023) juga menekankan keunggulan AES-128 dalam efisiensi proses enkripsi dan dekripsi file, yang sangat ideal untuk perangkat dengan kapasitas memori kecil [15].

AES dirancang untuk tersedia secara bebas tanpa royalti, murah untuk diterapkan pada perangkat dengan kapasitas memori kecil, dan mampu beroperasi dengan efisien pada berbagai mesin, mulai dari 8-bit hingga 64-bit. Selain itu, AES juga dirancang untuk lebih cepat daripada Triple DES, menjadikannya algoritma yang unggul untuk berbagai kebutuhan keamanan data modern.

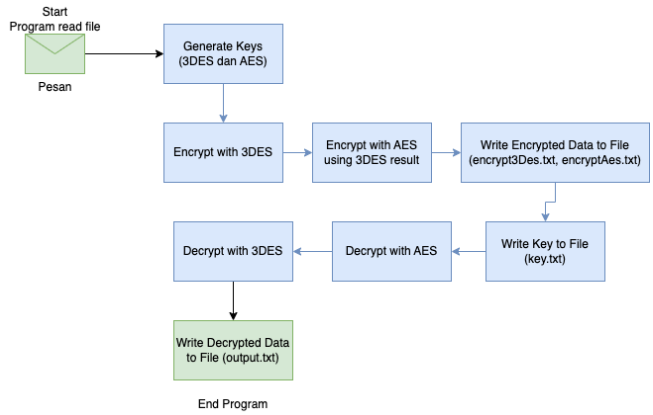
Pada penelitian ini digunakan algoritma AES yang dapat dilihat pada Gambar 4.

```
# Enkripsi dengan AES
def encrypt_aes(key, data):
    cipher = AES.new(key, AES.MODE_CBC)
    iv = cipher.iv
    ciphertext = cipher.encrypt(pad(data,
AES.block_size))
    return iv + ciphertext

# Dekripsi dengan AES
def decrypt_aes(key, data):
    iv = data[:AES.block_size]
    ciphertext = data[AES.block_size:]
    cipher = AES.new(key, AES.MODE_CBC,
iv=iv)
    decrypted_data =
unpad(cipher.decrypt(ciphertext),
AES.block_size)
    return decrypted_data
```

Gambar 4. Code implementasi aes

Gambar 5 merupakan diagram proses enkripsi yang dibangun dimulai dari Generate Key sampai mendapatkan paket pesan yang terenkripsi. Kemudian proses dekripsi yang dibangun, dimulai dari memisahkan pesan yang terenkripsi sampai mendapatkan pesan asli.



Gambar 5.Diagram proses 3des & aes

Untuk penjelasan dari diagram diatas secara mendalam dapat dilihat sebagai berikut:

1) Inisialisasi kunci. Sebelum melakukan enkripsi, kunci DES dan AES diinisialisasi. Kunci DES memiliki panjang 8 byte, sedangkan kunci AES dapat memiliki panjang 16, 24, atau 32 byte. Kedua kunci ini digunakan untuk melibatkan lapisan ganda enkripsi pada data.
2) Enkripsi dengan TripleDES. Data teks dari file input dibaca dan dienkripsi menggunakan algoritma Triple DES. Proses enkripsi melibatkan penggunaan mode Cipher Block Chaining (CBC) dengan padding PKCS7 untuk memastikan kelengkapan blok.
3) Enkripsi dengan AES Menggunakan Hasil Enkripsi Triple DES. Hasil enkripsi Triple DES digunakan sebagai input untuk algoritma enkripsi AES. Langkah ini memberikan tingkat keamanan tambahan dengan menerapkan dua lapisan enkripsi.
4) Deskripsi dengan AES. Proses dekripsi dilakukan dengan mengambil data hasil enkripsi AES dan mengembalikannya ke bentuk semula menggunakan kunci AES yang sama.
5) Deskripsi dengan TripleDES. Setelah dekripsi AES, langkah selanjutnya adalah melakukan dekripsi Triple DES untuk mendapatkan data teks asli.

E. Pengujian dan Perolehan Data

Pada tahap awal implementasi, ada dua kunci utama digunakan untuk melindungi keamanan data. Kunci yang pertama yaitu, DES (Data Encryption Standard) yang didefinisikan sebagai ‘8bytes key’ dengan panjang 8 byte, sedangkan kunci yang kedua adalah AES (Advanced Encryption Standard) yang ditentukan sebagai 'abcdefghijklmnop' dengan panjang 16 byte.

Proses enkripsi dimulai dengan mengambil data teks dari file input, yang kemudian dienkripsi menggunakan algoritma TripleDES. TripleDES ini dipilih karena kehandalannya yang telah terbukti seiring waktu. Hasil dari enkripsi TripleDES disimpan dalam file ‘encryptDes.txt’. Selanjutnya, untuk meningkatkan tingkat keamanannya, data yang telah dienkripsi dengan TripleDES itu kembali diambil dan dienkripsi sekali lagi, namun enkripsi kali ini menggunakan algoritma AES. Langkah ini bertujuan untuk memberikan lapisan keamanan tambahan dan melibatkan kunci AES yang telah ditetapkan sebelumnya. Hasil akhir dari proses ini adalah dua file terenkripsi yaitu ‘encryptDes.txt’ dan yang berisi hasil enkripsi dengan TripleDES, dan ‘encryptAes.txt’, yang berisi hasil enkripsi tambahan dengan AES.

Selanjutnya, dilakukan proses dekripsi untuk mengembalikan data ke bentuk aslinya. Data yang telah dienkripsi dengan AES dan TripleDES diambil untuk didekripsi. Proses dekripsi dimulai dengan dekripsi AES, diikuti oleh dekripsi TripleDES. Hasil akhir dari proses dekripsi ini kemudian disimpan dalam file ‘output.txt’, yang berisi data teks yang telah berhasil dikembalikan ke bentuk

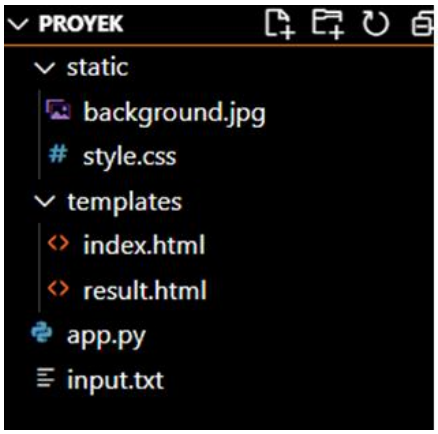
aslinya. Dengan demikian, melalui langkah-langkah enkripsi dan dekripsi ini, data teks dari file input dapat dengan aman dan efektif dilindungi dengan menggunakan dua algoritma kriptografi yang terpercaya, yaitu TripleDES dan AES.

HASIL DAN PEMBAHASAN

Pada tahap enkripsi, data teks dari file input (‘input.txt’) menjalani dua tahap enkripsi menggunakan algoritma TripleDES dan AES. Tahap pertama, data dienkripsi dengan TripleDES menggunakan kunci DES (‘8byte’). Hasil dari enkripsi TripleDES kemudian disimpan dalam file ‘encryptDes.txt’. Selanjutnya, hasil enkripsi TripleDES (‘encryptDes.txt’) lalu diambil dan dienkripsi kembali menggunakan algoritma AES dengan kunci AES (‘abcdefghijklmnop’). Hasil akhir dari proses ini disimpan dalam file ‘encryptAes.txt’. Proses dekripsi dilakukan dalam dua tahap sebaliknya. Pertama, data yang telah dienkripsi dengan AES (‘encryptAes.txt’) dekripsi menggunakan kunci AES yang sama. Hasil dekripsi AES disimpan dalam variabel decrypted_aes. Selanjutnya, hasil dekripsi AES (‘decrypted_aes’) diambil dan didekripsi kembali menggunakan algoritma TripleDES dengan kunci DES yang sama. Hasil akhir dari proses dekripsi disimpan dalam variabel decrypted_des.

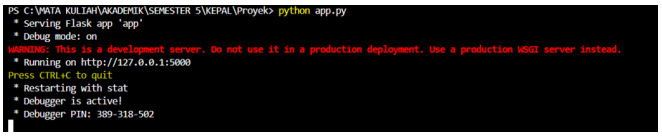
Selain itu, hasil dekripsi TripleDES (‘decrypted_des’) disimpan dalam file ‘output.txt’. Kunci AES yang digunakan dalam proses enkripsi disimpan dalam file ‘key.txt’. Untuk pengujian selanjutnya, hasil enkripsi TripleDES dan AES masing-masing disimpan dalam file ‘encryptDes.txt’ dan ‘encryptAes.txt’. Kombinasi dari dua algoritma kriptografi yang berbeda, yaitu TripleDES dan AES, memberikan lapisan keamanan ganda untuk melindungi data. Hasilnya diorganisir dan disimpan dengan rapi, memastikan keberhasilan dan keterbacaan hasil enkripsi dan dekripsi, serta penyimpanan kunci yang aman. Selanjutnya, evaluasi dan uji coba lebih lanjut dapat dilakukan untuk memastikan keandalan sistem dalam berbagai skenario penggunaan. Untuk penjelasan secara detail pada code program yang dibangun akan dijelaskan sebagai berikut.

Struktur Folder proyek code program yang belum dijalankan dapat dilihat pada Gambar 6 dibawah ini.



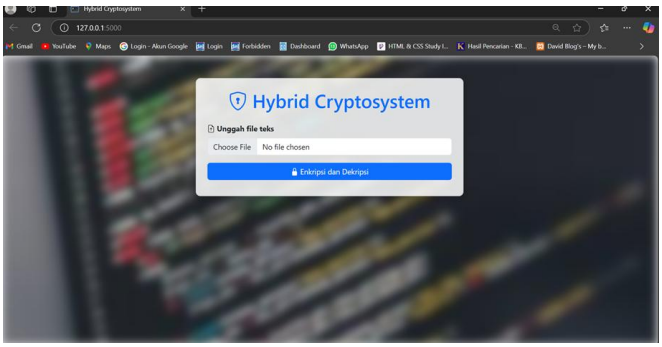
Gambar 6. Struktur proyek sebelum dijalankan

Untuk menjalankan program yang menggunakan framework Flask, syntax yang digunakan di terminal adalah flask run. Perintah ini akan menjalankan aplikasi Flask sesuai dengan file utama yang telah ditentukan, biasanya bernama app.py. Selain itu, aplikasi juga dapat dijalankan langsung melalui file utama dengan perintah python app.py. Syntax ini memastikan server Flask dapat berjalan dan aplikasi siap diakses melalui browser. Proses ini mendukung pengembangan, debugging, dan pengujian aplikasi dengan lebih efektif.

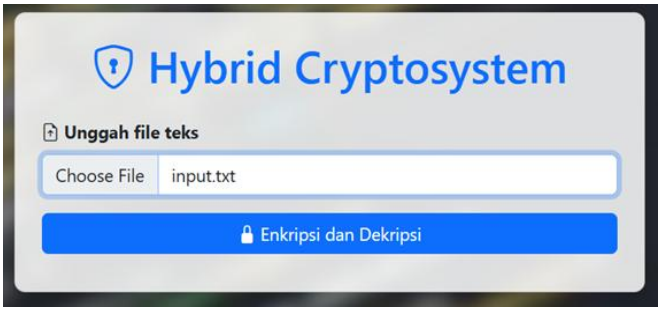


Gambar 7. Syntax pemanggilan

Berikut merupakan tampilan hasil eksekusi kode menggunakan Framework Flask, sebuah kerangka kerja berbasis Python yang dirancang untuk mempermudah pengembangan aplikasi web melalui fitur-fitur seperti routing, template rendering, serta integrasi dengan database.

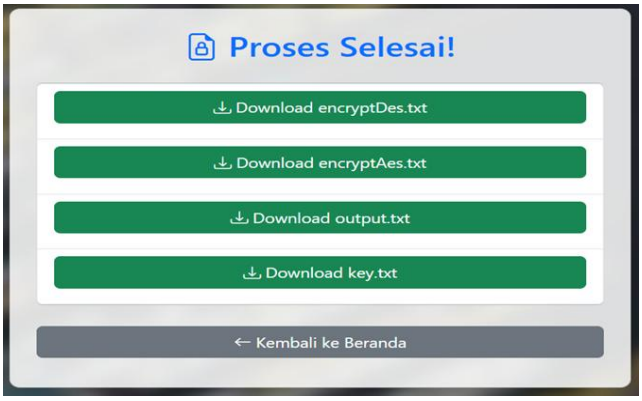


Gambar 8. Tampilan Awal program



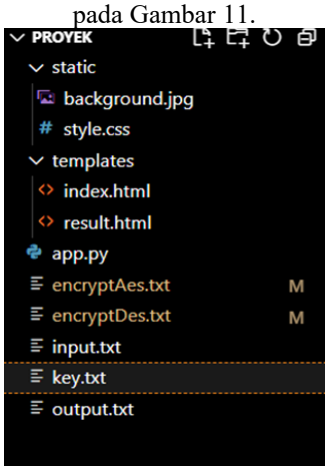
Gambar 9. Tampilan unggahan file txt

Ketika file Input.txt sudah dimasukkan maka akan dilakukan proses enkripsi dan deskripsi yang dimana nanti kan ada file baru yang terbentuk yaitu encryptDes.txt,encryptAes.txt,output.txt dan key.txt.



Gambar 10. Tampilan setelah melakukan 3des dan aes

Berikut adalah struktur folder proyek setelah code program yang telah dibangun dijalankan.yang dapat dilihat pada Gambar 11.



Gambar 11. Struktur folder setelah di jalankan

Pada syntax code program dibawah ini berguna untuk membaca isi file teks dari 'input.txt' dan menyimpannya dalam variabel file_data.yang ada pada file app.py.

```python
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        file = request.files['file']
        if file:
            file.save('input.txt')
            file_data = read_file('input.txt')
```

Gambar 12. Syntax membaca isi file

Kemudian pada potongan kode dibawah ini memiliki fungsi untuk menyediakan kunci untuk algoritma TripleDES (DES) dengan panjang 8 byte dan Advanced Encryption Standard (AES) dengan panjang 16 byte.

```python
# Kunci untuk TripleDES dan AES
des_key = b'8bytekey12345678'  # Kunci 24 byte untuk TripleDES
aes_key = b'abcdefghijklmnop'  # Kunci 16 byte untuk AES
```

Gambar 13. Kunci untuk algoritma tripledes

Tahapan selanjutnya, pada code program dibawah memiliki fungsi untuk mengenkripsi data menggunakan TripleDES dengan mode Cipher Block Chaining (CBC). Proses ini melibatkan penambahan padding PKCS7 pada data sebelum dienkripsi. Kemudian, Menggunakan hasil enkripsi DES sebagai input untuk enkripsi menggunakan AES dengan mode CBC. Seperti sebelumnya, padding PKCS7 juga diterapkan sebelum enkripsi. Setelah itu, Proses dekripsi menggunakan AES untuk mendekripsi data yang sebelumnya dienkripsi dengan AES. Padding PKCS7 dihapus setelah dekripsi. Begitu juga proses dekripsi selanjutnya menggunakan TripleDES untuk mendekripsi data yang sebelumnya dienkripsi dengan DES. Padding PKCS7 juga dihapus setelah dekripsi.

```python
# Enkripsi dengan TripleDES
def encrypt_3des(key, data):
    cipher = DES3.new(key, DES3.MODE_CBC)
    iv = cipher.iv
    ciphertext = cipher.encrypt(pad(data.encode('utf-8'), DES3.block_size))
    return iv + ciphertext

# Dekripsi dengan TripleDES
def decrypt_3des(key, data):
    iv = data[:DES3.block_size]
    ciphertext = data[DES3.block_size:]
    cipher = DES3.new(key, DES3.MODE_CBC, iv=iv)
    decrypted_data = unpad(cipher.decrypt(ciphertext), DES3.block_size)
    return decrypted_data.decode('utf-8')

# Enkripsi dengan AES
def encrypt_aes(key, data):
    cipher = AES.new(key, AES.MODE_CBC)
    iv = cipher.iv
    ciphertext = cipher.encrypt(pad(data, AES.block_size))
    return iv + ciphertext

# Dekripsi dengan AES
def decrypt_aes(key, data):
    iv = data[:AES.block_size]
    ciphertext = data[AES.block_size:]
    cipher = AES.new(key, AES.MODE_CBC, iv=iv)
    decrypted_data = unpad(cipher.decrypt(ciphertext), AES.block_size)
    return decrypted_data
```

Gambar 14. Enkripsi dan deskripsi

Kemudian code program write_file('output.txt', decrypted_des) untuk menyimpan hasil dekripsi ke dalam file 'output.txt'. Code write_key_to_file('key.txt', aes_key) untuk menyimpan kunci AES ke dalam file 'key.txt'. Dan dua syntax code program terakhir memiliki fungsi untuk menyimpan data terenkripsi (hasil enkripsi AES dan DES) ke dalam file 'encryptAes.txt' dan 'encryptDes.txt' masing-masing.

```
# Enkripsi dengan TripleDES
encrypted_des = encrypt_3des(des_key, file_data)
write_file('encryptDes.txt', encrypted_des)

# Enkripsi dengan AES
encrypted_aes = encrypt_aes(aes_key, encrypted_des)
write_file('encryptAes.txt', encrypted_aes)

# Simpan kunci AES ke dalam file
write_key_to_file('key.txt', aes_key)

# Dekripsi dengan AES
decrypted_aes = decrypt_aes(aes_key, encrypted_aes)

# Dekripsi dengan TripleDES
decrypted_des = decrypt_3des(des_key, decrypted_aes)
write_file('output.txt', decrypted_des.encode('utf-8'))

return redirect(url_for('result'))
```

Gambar 15. Syntax menyimpan data terenkripsi

File: Input.txt.

input.txt
1 Kamu keren kalua bisa ngoding bro

Gambar 16. isi file input.txt

File : Key.txt

key.txt
1 61626364656667686 96a6b6c6d6e6f70

Gambar 17. isi file key.txt

File : encryptDes.txt

encryptDes.txt

Gambar 18. isi file encryptDes.txt

File : encryptAes.txt

ð_□`mU□¢LT6¦è#xF5y
-Æ□[□¢¯Z□□\[b□ã÷°6'ž=8□]□□%:r\€Szå□i
ÓXOÞ¡'□í□ýRbò-I□²ÝçŠšâûT

Gambar 19. isi file encryptAes.txt

File : Output.txt

output.txt
1 Kamu keren kalua bisa ngoding bro

Gambar 20. isi file Output.txt

Program ini adalah implementasi dari hybrid cryptosystem yang menggunakan dua algoritma enkripsi, yaitu TripleDES (3DES) dan Advanced Encryption Standard (AES), untuk mengamankan file teks. Proses dimulai dengan membaca isi file teks yang akan diamankan, kemudian dilakukan enkripsi menggunakan 3DES dalam mode Cipher Block Chaining (CBC) dengan padding PKCS7. Hasil enkripsi 3DES ini kemudian dienkripsi kembali menggunakan AES dalam mode CBC, juga dengan padding PKCS7. Proses dekripsi dilakukan secara terbalik, dengan data terenkripsi AES di-dekripsi terlebih dahulu, kemudian hasilnya di-dekripsi kembali menggunakan 3DES. Hasil akhir dari dekripsi tersebut dituliskan ke dalam file 'output.txt', sedangkan kunci AES disimpan ke dalam 'key.txt'. Data terenkripsi dari kedua algoritma (AES dan 3DES) juga disimpan dalam file ‘encryptAes.txt’ dan ‘encryptDes.txt’. Dengan pendekatan ini, program memberikan tingkat keamanan yang lebih baik dengan memanfaatkan dua lapisan enkripsi, meningkatkan keamanan file teks dari potensi akses yang tidak sah.yang dimana sistem akan dimasukkan kedalam framework flask agar memiliki tampilan yang dimana bahasa pemrograman yang digunakan python,html dan css.

Adapun hasil diskusi mengenai penelitian ini adalah sebagai berikut:

1) Keamanan Enkripsi
   Dengan menggabungkan TripleDES dan AES, membantu memberikan tingkat keamanan ganda, sehingga data mengalami enkripsi TripleDES terlebih dahulu, lalu selanjutnya enkripsi tambahan menggunakan AES.
2) Kunci
   Kunci DES dengan panjang 8 byte sesuai untuk TripleDES, sedangkan kunci AES dengan panjang 16 byte sesuai untuk AES. Hal ini menunjukkan ketepatan penggunaan kunci dalam proses enkripsi.
3) Inisialisasi Vektor (IV)
   Penggunaan Inisialisasi Vektor (IV) pada kode program diinisialisasi dengan byte nol dalam upaya meningkatkan keamanan. Namun, disarankan untuk meningkatkan tingkat keamanan dengan menggunakan IV yang dihasilkan secara acak dan unik untuk setiap pesan yang dienkripsi.
4) Penyimpanan Hasil
   Pentingnya penyimpanan hasil juga menjadi fokus, di mana enkripsi TripleDES dan AES disimpan secara terpisah dalam file ‘encryptDes.txt’ dan ‘encryptAes.txt’ masing-masing. Kunci AES juga

dijaga dengan baik, disimpan dalam file ‘key.txt’. Hasil dekripsi dari TripleDES, sebagai bagian integral dari proses, diabadikan dalam file ‘output.txt’.

5) *Penggunaan Library, Pengujian dan Evaluasi*

Dengan memanfaatkan library cryptography.hazmat, implementasi algoritma kriptografi menjadi lebih kuat, memastikan keamanan dan keandalan dalam seluruh proses enkripsi dan dekripsi. Keefektifan algoritma dapat diukur melalui pengujian lebih lanjut, yang melibatkan berbagai data input untuk memvalidasi kehandalan mereka dalam berbagai situasi dan skenario penggunaan. Uji coba lebih lanjut dapat memberikan wawasan lebih lanjut terkait keamanan dan kinerja algoritma kriptografi yang digunakan.

SIMPULAN

Berdasarkan pembahasan yang telah diuraikan pada bab-bab sebelumnya, dapat disimpulkan bahwa implementasi Hybrid Cryptosystem dengan menggabungkan algoritma TripleDES (3DES) dan Advanced Encryption Standard (AES) telah berhasil meningkatkan tingkat keamanan data dalam pertukaran informasi digital. Berikut adalah beberapa kesimpulan utama dari penelitian ini:

1) Keamanan Ganda: Penggunaan kombinasi TripleDES dan AES memberikan lapisan keamanan ganda pada data. Proses enkripsi pertama dengan TripleDES diikuti oleh enkripsi kedua dengan AES memberikan tingkat keamanan yang lebih tinggi.
2) Ukuran Kunci: Kunci yang digunakan sesuai dengan persyaratan masing-masing algoritma. Kunci DES untuk TripleDES memiliki panjang 8 byte, sedangkan kunci AES harus berukuran 16, 24, atau 32 byte, memberikan tingkat keamanan yang sesuai.
3) Inisialisasi Vektor (IV): Penggunaan IV yang diinisialisasi dengan byte nol telah diterapkan, tetapi disarankan untuk menggunakan IV yang dihasilkan secara acak dan unik untuk meningkatkan keamanan.
4) Penyimpanan Hasil: Hasil enkripsi dari TripleDES dan AES disimpan secara terpisah dalam file 'encryptDes.txt' dan 'encryptAes.txt'. Kunci AES juga disimpan dalam file 'key.txt', dan hasil dekripsi dari TripleDES disimpan dalam 'output.txt'.
5) Penggunaan Library: Pemanfaatan library cryptography.hazmat memberikan implementasi algoritma kriptografi yang kuat, memastikan keamanan dan keandalan selama proses enkripsi dan dekripsi.
6) Pengujian dan Evaluasi Lanjutan: Uji coba lebih lanjut perlu dilakukan dengan melibatkan berbagai jenis data input untuk menguji kehandalan algoritma dalam skenario penggunaan yang beragam.
7) Dengan demikian, implementasi Hybrid Cryptosystem ini diharapkan dapat memberikan solusi konkret dalam meningkatkan keamanan data dalam pertukaran informasi digital. Melalui penerapan konsep ini, diharapkan dapat memberikan kontribusi positif terhadap evolusi teknologi keamanan informasi di era digital yang penuh dengan tantangan keamanan siber.

UCAPAN TERIMA KASIH

Peneliti ingin mengucapkan rasa terima kasih yang mendalam kepada semua yang telah berperan penting dalam menyelesaikan karya ini. Pertama-tama, penghargaan dan terima kasih disampaikan kepada Bapak Rudy Chandra, S.Kom.,M.Kom yang telah memberikan bimbingan, arahan, dan inspirasi yang sangat berharga selama proses penulisan. Selanjutnya, penghargaan juga diberikan kepada setiap rekan peneliti yang telah berkolaborasi dengan baik dalam penelitian ini, memberikan pengalaman yang berarti, dan berkontribusi pada pengetahuan yang memperkaya pemahaman peneliti. Keberhasilan proyek ini tidak mungkin tercapai tanpa kontribusi dan kerja keras dari semua pihak terlibat. Setiap ide dan sudut pandang yang diberikan oleh setiap peneliti telah menjadi landasan penting dalam pembentukan karya ini. Terakhir, terima kasih kepada semua yang turut serta dalam perjalanan penelitian ini, baik secara langsung maupun tidak langsung. Semoga hasil penelitian ini dapat memberikan kontribusi yang signifikan dalam pengembangan ilmu pengetahuan.

DAFTAR PUSTAKA

[1] P. C. Mandal, “Evaluation of performance of the Symmetric Key Algorithms : DES , 3DES , AES and Blowfish,” *J. Glob. Res. Comput. Sci.*, vol. 3, no. 8, pp. 67–70, 2012.

[2] A. Pudoli and D. Kusumaningsih, “PENGGUNAAN HYBRID CRYPTOSYSTEM UNTUK ENKRIPSI DAN DEKRIPSI PESAN MESSANGER MENGGUNAKAN ALGORITMA RIVEST SHAMIR ADLEMAN (RSA) DAN ADVANCED ENCRYPTION STANDARD (AES) DENGAN FIREBASE PADA ANDROID,” vol. 9, no. 3, pp. 125–131, 2017.

[3] M. Suwarni, J. Wahyudi, and K. Khairil, “Comparison of the DES Cryptographic Algorithm and the AES Algorithm in Securing Document Files,” *J. Media Comput. Sci.*, vol. 2, no. 1, pp. 41–48, 2023, doi: 10.37676/jmcs.v2i1.3348.

[4] D. Surian, “Algoritma Kriptografi Aes Rijndael,” *Tesla*, vol. 8, no. 2, pp. 97–101, 2006.

[5] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, “A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish,” *Procedia Comput. Sci.*, vol. 78, no. December 2015, pp. 617–624, 2016, doi: 10.1016/j.procs.2016.02.108.

[6] E. J. Kusuma, O. R. Indriani, C. A. Sari, E. H. Rachmawanto, and D. R. I. M. Setiadi, “An imperceptible LSB image hiding on edge region using des encryption,” *Proc. - 2017 Int. Conf. Innov. Creat. Inf. Technol. Comput. Intell. IoT, ICITech 2017*, vol. 2018-Janua, pp. 1–6, 2017, doi: 10.1109/INNOCIT.2017.8319132.

[7] G. Ardiansyah, D. R. I. M. Setiadi, C. A. Sari, and E. H. Rachmawanto, “Hybrid method using 3-DES, DWT and LSB for secure image steganography algorithm,” *Proc. - 2017 2nd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. ICITISEE 2017*, vol. 2018-Janua,

pp. 249–254, 2017, doi: 10.1109/ICITISEE.2017.8285505.

[8] D. R. Wahyuni Juli, Gunawan Indra, "Jurnal Mantik Jurnal Mantik," *Mobile-Based Natl. Univ. Online Libr. Appl. Des.*, vol. 3, no. 2, pp. 10–19, 2019, [Online]. Available: http://iocscience.org/ejournal/index.php/mantik/article/view/882/595

[9] K. Zalukhu, Y. Syahra, and T. Syahputra, "Implementasi Sistem Keamanan Database Data Pelanggaran Hukum Disiplin Prajurit Menggunakan Algoritma Advanced Encryption Standard 128 Bit Pada Pengadilan Militer I-02 Medan," *J-SISKO TECH (Jurnal Teknol. Sist. Inf. dan Sist. Komput. TGD)*, vol. 3, no. 2, p. 138, 2020, doi: 10.53513/jsk.v3i2.2419.

[10] Fathurrahmad and Ester, "Development and Implementation of The Rijndael Algorithm and Base-64 Advanced Encryption Standard (AES) for Website Data Security," *Int. J. Comput. Appl.*, vol. 9, no. 11, pp. 9–12, 2020.

[11] S. R. Siburian, R. Alek, S. Sinaga, and F. Yudistira, "Kriptosistem Hybrid Menggunakan Kombinasi AES Dan RSA Untuk Enkripsi Teks Pesan," *J. JOCOTIS - J. Sci. Inform. Robot.*, vol. 1, no. 1, pp. 22–31, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jct/

[12] Munawar, "PERANCANGAN ALGORITMA SISTEM KEAMANAN DATA Munawar Jurnal Komputer dan Informatika (KOMPUTA)," *J. Komput. dan Inform.*, vol. 1, pp. 11–16, 2012.

[13] A. Teguh Utomo and R. Pradana, "Implementasi Algoritma Advanced Encryption Standard (AES-128) Untuk Enkripsi dan Dekripsi File," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 21–23, 2022.

[14] R. Ravida and H. A. Santoso, "Advanced Encryption Standard (AES) 128 Bit untuk Keamanan Data Internet of Things (IoT) Tanaman Hidroponik," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1157–1164, 2020, doi: 10.29207/resti.v4i6.2478.

[15] Melenia Bayu Aryanto, Muhlis Tahir, Silvia Irma Devita, Zuda Nuril Mustofa, Qurrotun Ainiyah, and Shelviatus Sundoro, "Implementasi Enkrip Dan Dekrip File Menggunakan Metode Advance Encryption Standard (AES-128)," *J. Ilm. Sist. Inf. dan Ilmu Komput.*, vol. 3, no. 1, pp. 89–104, 2023, doi: 10.55606/juisik.v3i1.434.