

IMPLEMENTASI ALGORITMA ONE TIME PAD PADA KEAMANAN PESAN KOMUNIKASI MODERN

Yuliana Cantikasari^{1*}, Muhamad Fachrudin², Wartono³

^{1,2,3} Universitas Tunas Pembangunan Surakarta

*Corresponding Author:  f0223001_yulianacantikasari@student.utp.ac.id

Info Artikel

Sejarah Artikel:

Diterima: 2026-07-21

Direvisi: 2026-07-27

Disetujui: 2026-07-28

Keywords:

One Time Pad, XOR Operation, Cryptography, Modern Communication, Data Encryption

Kata Kunci:

One Time Pad, Operasi XOR, Kriptografi, Komunikasi Modern, Enkripsi Data

Abstract. Data security is a major challenge in the increasingly complex modern communication era, especially with the increasing threats to the privacy and integrity of digital information. This study explores the application of the One-Time Pad (OTP) algorithm as one of the most secure encryption methods to protect digital messages from unauthorized access. OTP works by combining plaintext with a random key through an XOR operation, producing ciphertext that cannot be cracked without the correct key. This study uses experimental methods to compare OTP with other modern cryptographic algorithms, including analyzing their efficiency, security level, and resilience to various types of cyber attacks. The results show that although OTP offers perfect security in theory, its implementation faces major challenges, especially in distributing keys securely and efficiently. To overcome these challenges, the study suggests combining OTP with modern algorithms such as RSA to improve the efficiency of the encryption and decryption process, while maintaining a high level of security. These findings provide new insights that are expected to contribute to the development of a more reliable data security system in the rapidly evolving digital era.

Abstrak. Keamanan data merupakan tantangan utama di era komunikasi modern yang semakin kompleks, terutama dengan meningkatnya ancaman terhadap privasi dan integritas informasi digital. Studi ini mengeksplorasi penerapan algoritma One-Time Pad (OTP) sebagai salah satu metode enkripsi paling aman untuk melindungi pesan digital dari akses yang tidak sah. OTP bekerja dengan menggabungkan plaintext dengan kunci acak melalui operasi XOR, menghasilkan ciphertext yang tidak dapat dipecahkan tanpa kunci yang benar. Studi ini menggunakan metode eksperimental untuk membandingkan OTP dengan algoritma kriptografi modern lainnya, termasuk menganalisis efisiensi, tingkat keamanan, dan ketahanan terhadap berbagai jenis serangan siber. Hasil menunjukkan bahwa meskipun OTP menawarkan keamanan sempurna secara teori, implementasinya menghadapi tantangan besar, terutama dalam mendistribusikan kunci secara aman dan efisien. Untuk mengatasi tantangan ini, studi ini menyarankan penggabungan OTP dengan algoritma modern seperti RSA untuk meningkatkan efisiensi proses enkripsi dan dekripsi, sambil mempertahankan tingkat keamanan yang tinggi. Temuan ini memberikan wawasan baru yang diharapkan dapat berkontribusi pada pengembangan sistem keamanan data yang lebih andal di era digital yang berkembang pesat.

Alamat korespondensi:

Alamat Penulis Korespondensi

 f0223001_yulianacantikasari@student.utp.ac.id

Penerbit:

Program Studi Sistem Informasi Universitas Flores.

 ftiuniflor.ois@gmail.com

1. PENDAHULUAN

Perkembangan pesat di berbagai bidang, mulai dari komunikasi, informasi, hingga transportasi, telah mengubah cara manusia berinteraksi, bekerja, dan menjalani kehidupan sehari-hari. Inovasi teknologi tidak hanya mempermudah aktivitas manusia, tetapi juga membuka peluang baru di berbagai sektor, seperti pendidikan, bisnis, kesehatan, dan hiburan [1]. Namun, di balik berbagai manfaatnya, teknologi juga menghadirkan tantangan, terutama terkait privasi, keamanan data, dan ketergantungan yang berlebihan. Salah satunya adalah kerentanan keamanan data, yang menimbulkan tantangan dan tuntutan akan ketersediaan sistem keamanan data yang seanggih kemajuan teknologi komputer itu sendiri [2]. Ilmu pengetahuan dan teknologi memungkinkan munculnya teknik baru yang disalahgunakan oleh pihak-pihak tertentu yang mengancam keamanan sistem informasi. Ironisnya, teknik yang digunakan untuk mengancam keamanan data tersebut satu tingkat lebih maju [3].

Keamanan dan kerahasiaan data merupakan aspek penting dalam sistem informasi modern. Sistem dengan keamanan tinggi tentu akan memberikan kepercayaan yang tinggi kepada penggunanya. Dengan meningkatkan tingkat kepercayaan pada sistem, hal itu secara langsung meningkatkan nilai dan kegunaan sistem [4]. Pengguna akan merasa nyaman dan aman ketika berurusan dengan sistem yang dapat mengamankan data pengguna dari penyerang. Seiring dengan perkembangan ilmu pengetahuan dan teknologi yang pesat, berbagai teknik telah muncul yang dapat disalahgunakan oleh pihak yang tidak bertanggung jawab, sehingga mengancam keamanan sistem informasi [5]. Salah satu pendekatan untuk menjaga keamanan informasi adalah melalui kriptografi, ilmu yang mempelajari teknik untuk menjaga keamanan pesan (teks biasa) dari akses yang tidak sah. Kriptografi berfungsi untuk memastikan bahwa pesan atau kunci tetap aman meskipun saluran komunikasi antara pengirim dan penerima

sepenuhnya dikendalikan oleh penyadap [6].

Penelitian ini mengusulkan penggunaan algoritma One-Time Pad (OTP) untuk meningkatkan keamanan pesan. OTP ditemukan oleh Mayor Joseph Maugborne dan Gilbert Vernam pada tahun 1917 yang saat ini dikenal sebagai salah satu metode kriptografi paling aman [7]. Algoritma ini termasuk dalam kategori kriptografi kunci simetris, di mana kunci yang digunakan untuk enkripsi dan dekripsi adalah sama [8]. Proses enkripsi bertujuan untuk mengamankan pesan dengan mengubahnya menjadi format yang tidak dapat dibaca oleh siapa pun kecuali penerima yang berwenang, sedangkan dekripsi mengembalikan pesan ke format aslinya [7]. OTP biasanya diimplementasikan menggunakan penjumlahan modular (XOR) untuk menggabungkan elemen plaintext dengan elemen aliran kunci. Kunci yang digunakan untuk enkripsi juga digunakan untuk dekripsi, menerapkan kunci yang sama pada ciphertext menghasilkan kembali plaintext [9].

Studi ini bertujuan untuk mengeksplorasi penerapan OTP dalam mengamankan pesan terenkripsi, khususnya dalam konteks komunikasi modern. Beberapa faktor yang akan dianalisis meliputi kompleksitas kunci, ketahanan terhadap serangan kriptanalisis, dan perbandingan OTP dengan algoritma kriptografi modern lainnya. Diharapkan studi ini dapat memberikan kontribusi penting dalam meningkatkan keamanan pesan digital, serta memperkuat pemahaman tentang pentingnya perlindungan data di era teknologi yang terus berkembang. Studi ini juga berupaya memberikan solusi praktis dalam menjaga keamanan komunikasi digital modern.

2. TINJAUAN LITERATUR

One-Time Pad (OTP) adalah salah satu metode kriptografi yang paling aman karena menggunakan kunci acak yang panjangnya sama dengan pesan terenkripsi, dan kunci tersebut hanya digunakan sekali. OTP bekerja dengan menggabungkan teks asli (*plaintext*)

dengan kunci acak melalui operasi XOR (*Exclusive OR*). Kunci yang digunakan harus benar-benar acak dan tidak boleh digunakan kembali. Setelah operasi XOR antara plaintext dan kunci dilakukan, hasilnya adalah ciphertext yang hanya dapat didekripsi jika kunci yang sama diketahui [2, 6].

Metode One Time Pad dapat digunakan untuk komunikasi rahasia tingkat tinggi dengan volume yang tidak terlalu besar, tetapi untuk penggunaan skala besar dalam sistem teknologi informasi, One Time Pad tidak praktis. Meskipun tidak digunakan secara langsung, konsep One Time Pad ditiru dalam teknik enkripsi stream cipher [10].

Penelitian mengenai peningkatan keamanan data teks melalui skema penggabungan dan analisis algoritma kriptografi telah banyak dilakukan sebelumnya. Penelitian pertama oleh Reymond dkk. (2024) mengkaji secara mendalam penerapan dasar algoritma simetris One-Time Pad (OTP) berbasis operasi *Exclusive-OR* (XOR) dan konversi kode ASCII. Penelitian tersebut membuktikan bahwa sifat acak mutlak dan penggunaan kunci sekali pakai pada OTP mampu menghasilkan ciphertext yang sangat aman dan terhindar dari peretasan secara teoritis. Namun demikian, hasil pengujian mereka menegaskan bahwa penerapan praktis OTP berdiri sendiri mengalami kendala besar pada manajemen dan distribusi kunci, sehingga membutuhkan metode pendukung untuk menyempurnakan penerapannya [6].

Untuk mengatasi keterbatasan tersebut, Putra dkk. (2023) mengembangkan skema penggabungan algoritma simetris Vigenère Cipher dan One-Time Pad (OTP) berbasis operasi XOR yang dilengkapi dengan fungsi *hash Message Digest 5* (MD5). Penelitian ini

berhasil membuktikan bahwa enkripsi ganda mampu mengubah teks asli menjadi karakter alfanumerik acak yang sulit diprediksi sehingga tebal terhadap ancaman analisis frekuensi (*frequency analysis*), sementara penambahan fungsi MD5 secara efektif bekerja sebagai checksum untuk menjamin integritas pesan selama transmisi data [7].

Penelitian yang dilakukan oleh Asrofa dkk. (2025) membahas pengamanan sistem *Smart Door* berbasis IoT dan QR Code dari ancaman penyadapan data seperti *sniffing* dan *Man-in-the-Middle*. Penelitian ini menerapkan algoritma kriptografi ringan *One-Time Pad* (OTP) berbasis XOR untuk mengamankan transmisi data antara server, ESP32-CAM, dan NodeMCU ESP32. Hasil pengujian menunjukkan bahwa enkripsi OTP XOR memiliki tingkat ketahanan 100% terhadap serangan *brute force* dan tidak dapat dibaca saat disadap. Selain itu, proses komputasinya sangat cepat—hanya membutuhkan waktu sekitar 0,34 ms untuk enkripsi dan 0,17 ms untuk dekripsi—sehingga sangat efisien dan ideal digunakan pada perangkat mikrokontroler [11].

Di sisi lain, penelitian yang dilakukan oleh Zalukhu dkk. (2024) berfokus pada penyelesaian kendala manajemen kunci pada algoritma One-Time Pad (OTP) melalui penggunaan *Linear Feedback Shift Register* (LFSR) sebagai pembangkit kunci acak otomatis berbasis plaintext. Ketika dikombinasikan dengan Vigenère Cipher menggunakan Modulo 256 dan konversi ASCII, metode LFSR terbukti mampu memproduksi kunci acak dan unik secara efisien. Kompleksitas karakter yang dihasilkan dari periode shift LFSR ini secara signifikan dapat menekan tingkat kerentanan sistem

terhadap ancaman peretasan maupun kebocoran pesan teks [4].

Berdasarkan rangkaian penelitian terdahulu, dapat disimpulkan bahwa algoritma One-Time Pad (OTP) berbasis XOR terbukti sangat handal dalam menjaga keamanan data dan efisien dari segi komputasi, namun penerapannya secara mandiri terkendala oleh masalah manajemen serta distribusi kunci. Untuk mengatasi keterbatasan tersebut, kombinasi skema enkripsi ganda seperti penggunaan Vigenère Cipher dan fungsi hash MD5 terbukti efektif memperkuat integritas data sekaligus mencegah analisis frekuensi [5, 6, 11]. Selain itu, kendala distribusi kunci OTP dapat diselesaikan secara praktis melalui penggunaan pembangkit kunci acak otomatis seperti *Linear Feedback Shift Register* (LFSR), yang mampu menghasilkan kunci unik berbasis plaintext sehingga secara signifikan meningkatkan kompleksitas enkripsi dan ketahanan sistem terhadap ancaman peretasan [4].

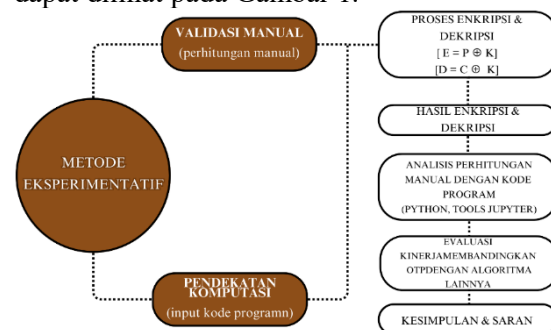
Komunikasi modern semakin bergantung pada teknologi enkripsi untuk melindungi data dan memastikan keamanan pertukaran informasi. Kombinasi mikrokontroler, komunikasi nirkabel, dan kriptografi merupakan aspek penting dalam mendukung komunikasi yang aman dan efisien di era digital ini [13]. One Time Pad (OTP) adalah metode enkripsi yang sangat aman, tetapi memiliki tantangan utama dalam distribusi kunci. Meskipun demikian, OTP tetap relevan dalam situasi di mana keamanan absolut diperlukan, terutama dalam komunikasi rahasia tingkat tinggi, seperti di bidang militer atau aplikasi cloud.

Peneliti ini menggunakan algoritma One Time Pad yang bertujuan untuk

mengeksplorasi penerapan OTP dalam mengamankan pesan terenkripsi, terutama dalam konteks komunikasi modern. Penelitian ini diharapkan dapat meningkatkan keamanan pesan digital dan memperkuat perlindungan data di era teknologi yang terus berkembang.

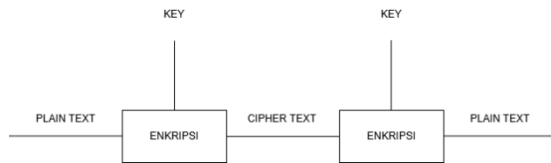
3. METODE PENELITIAN

Penelitian ini merupakan studi kasus tentang keamanan pesan dengan algoritma One Time Pad (OTP) dalam komunikasi modern. Metode yang digunakan adalah eksperimental dengan pendekatan komputasi dan validasi manual untuk menguji algoritma One Time Pad menggunakan kode program yang hasilnya diverifikasi dengan perhitungan manual untuk memastikan keakuratannya dan untuk membandingkan algoritma One Time Pad dengan algoritma lain. Alur penelitian dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Metode Penelitian

Kriptografi pada dasarnya terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Proses enkripsi adalah proses pengkodean pesan terbuka menjadi pesan rahasia (*ciphertext*) [14]. Ketika ciphertext diterima oleh penerima pesan, pesan rahasia diubah kembali menjadi pesan terbuka melalui proses dekripsi sehingga pesan dapat dibaca kembali oleh penerima pesan. Secara umum, proses enkripsi dan dekripsi dapat dilihat pada Gambar 2.



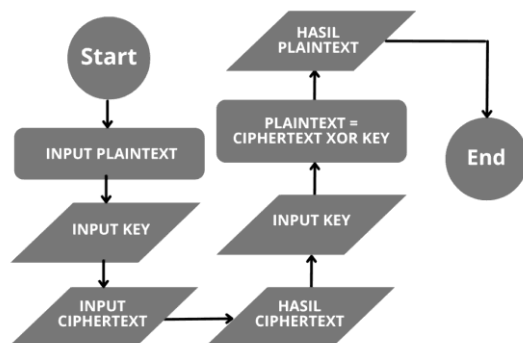
Gambar 2. Proses Enkripsi dan Dekripsi

Secara teoritis, algoritma one time pad merupakan algoritma enkripsi yang sempurna selama menggunakan kunci yang benar-benar acak dan tidak direproduksi [15]. Algoritma One Time Pad menggunakan kunci yang sama dalam proses enkripsi dan dekripsi serta memanfaatkan operasi XOR dalam proses enkripsi dan dekripsi. Operasi XOR seperti pada Tabel 1.

Tabel 1. Operasi XOR

Bit Plaintext	Bit Key	Bit Ciphertext
0	0	0
1	0	1
0	1	1
1	1	0

Operasi XOR menghasilkan nilai 0 jika argumennya sama (0 dengan 0 atau 1 dengan 1) dan menghasilkan 1 jika argumennya berbeda (0 dengan 1 atau 1 dengan 0). Detail proses enkripsi dan dekripsi OTP dapat dilihat pada Gambar 3.



Gambar 3. Diagram Alur Enkripsi dan Dekripsi OTP

Tahap enkripsi dalam algoritma One Time Pad dimulai dengan mengubah plaintext menjadi bentuk biner. Setelah itu, kunci acak dihasilkan, di mana panjang kunci acak harus sama dengan panjang plaintext yang akan dienkripsi. Kemudian proses operasi XOR dilakukan antara plaintext dan kunci yang telah dihasilkan sebelumnya, menghasilkan ciphertext. Untuk proses dekripsi, langkah pertama adalah mengubah ciphertext menjadi bentuk biner. Selanjutnya, kunci yang digunakan dalam proses enkripsi diambil. Setelah itu, operasi XOR dilakukan antara ciphertext dan kunci yang dipilih. Hasil dari operasi ini adalah plaintext asli [5, 10].

Proses enkripsi dan dekripsi, algoritma One Time Pad (OTP) melakukan karakter demi karakter atau sering juga disebut metode stream cipher. Dalam stream cipher, jika terjadi kesalahan selama transmisi, kesalahan dalam teks enkripsi penerima akan terjadi tepat di tempat kesalahan terjadi. Hasil enkripsi adalah nilai yang diperoleh dari XOR antara plaintext dan bit kunci. Terdapat beberapa persyaratan utama agar algoritma One Time Pad dapat digunakan secara optimal, yaitu pemilihan kunci harus dilakukan secara acak sehingga tidak mudah ditebak, jumlah karakter kunci harus memiliki panjang yang sama dengan karakter teks biasa, dan jika kunci tidak dapat direproduksi maka algoritma dinyatakan aman [6].

Ketahanan algoritma One Time Pad bergantung pada keacakan kunci yang digunakan. Algoritma One Time Pad dapat digunakan untuk komunikasi rahasia dengan nilai yang tidak terlalu besar, tetapi dalam skala besar, algoritma One Time Pad menjadi kurang efektif karena kendala pada manajemen kunci atau distribusi kunci. Dengan metode eksperimental ini, diharapkan dapat mempermudah perhitungan dan validasi komputasi untuk menentukan keamanan pesan dalam konteks komunikasi modern menggunakan algoritma sandi One Time Pad (OTP).

3. HASIL DAN PEMBAHASAN

Algoritma One-Time Pad (OTP) digunakan untuk mengamankan pesan melalui serangkaian proses terstruktur. Proses dimulai dengan mengubah plaintext dan kunci menjadi bentuk biner, kemudian melakukan operasi XOR (Exclusive OR) bit demi bit untuk menghasilkan ciphertext. Dalam proses enkripsi, plaintext (P) di-XOR-kan dengan kunci (K) untuk menghasilkan ciphertext (C), rumus dapat dilihat pada rumus 1

$$C = P \oplus K \quad (1)$$

Proses dekripsi dilakukan dengan mengembalikan ciphertext ke bentuk aslinya menggunakan kunci yang sama melalui operasi re-XOR, rumus dapat dilihat pada rumus 2

$$P = C \oplus K \quad (2)$$

Dalam penelitian ini, para peneliti mencoba mengirimkan pesan plaintext "DATADIRI" menggunakan kunci "LOCKLOCK".

3.1. Konversi Text ke Biner

Konversikan setiap karakter dalam teks biasa dan kunci menjadi bilangan biner berdasarkan tabel ASCII. Hasil konversi dapat dilihat pada Tabel 2.

Tabel 2. Konversi Plaintext dan Kunci ke Biner

Plaintext	Biner	Key	Biner
D (68)	01000100	L (76)	01001100
A (65)	01000001	O (79)	01001111
T (84)	01010100	C (67)	01000011
A (65)	01000001	K (75)	01001011
D (68)	01000100	L (76)	01001100
I (73)	01001001	O (79)	01001111
R (82)	01010010	C (67)	01000011
I (73)	01001001	K (75)	01001011

3.2. XOR Proses Enkripsi

Operasi XOR dilakukan bit demi bit antara teks asli dan kunci. Hasil perhitungan enkripsi dapat dilihat pada Tabel 3

Tabel 3. Perhitungan Enkripsi

Plaintext	XOR ($\oplus$)	Kunci (Key)	Hasil XOR	Hasil Desimal	Ciphertext
01000100 0	$\oplus$	01001100	00001000 0	8	I
01000001 1	$\oplus$	01001111	00001111 0	14	O
01010100 0	$\oplus$	01000011	00010111 1	23	R
01000001 1	$\oplus$	01001011	00001010 0	10	K
01000100 0	$\oplus$	01001100	00001000 0	8	I
01001001 1	$\oplus$	01001111	00000111 0	6	G
01010010 0	$\oplus$	01000011	00010001 1	17	T
01001001 1	$\oplus$	01001011	00000010 0	2	C

Hasil ciphertext dari proses enkripsi adalah "IORKIGTC".

3.3. XOR Proses Dekripsi

Untuk mengembalikan teks sandi ke teks biasa, teks sandi dikonversi ke biner, diikuti dengan operasi XOR dengan kunci yang sama. Hasil konversi teks sandi ke biner menurut tabel konversi ASCII ditunjukkan pada Tabel 4

Tabel 4. Konversi Ciphertext ke Biner

Ciphertext	ASCII	Biner
I	73	00001000
O	79	00001110
R	82	00010111
K	75	00001010
I	73	00001000
G	71	00000110
T	84	00010001
C	67	00000010

Setelah mengubah teks sandi menjadi bilangan biner, peneliti kemudian melakukan perhitungan XOR yang dapat dilihat pada Tabel 5.

Tabel 5. Perhitungan Dekripsi

Ciphertex t	XOR	Kunci (key)	Hasil XOR	ASCII	Plaintext
00001000	⊕	01010000	01000100	75	D
00001110	⊕	01000101	01000001	79	A
00010111	⊕	01010010	01010100	68	T
00001010	⊕	01001001	01000001	69	A
00001000	⊕	01001110	01000100	65	D
00000110	⊕	01010100	01001001	77	I
00010001	⊕	01000001	01010010	65	R
00000010	⊕	01001000	01001001	78	I

Hasil perhitungan dekripsi pada tabel 6 dengan ciphertext "IORKIGTC" berhasil dienkripsi menjadi plaintext "DATADIRI". Dari tahapan perhitungan manual di atas, dihasilkan skrip kode berikut yang dapat dilihat pada Gambar 4, Gambar 5, Gambar 6, Gambar 7, Gambar 8 dan hasil eksekusinya dapat dilihat pada Gambar 9.

```
import string

def cek_kunci(teks, kunci):
    """Memvalidasi kunci yang dimasukkan pengguna"""
    # Hapus karakter non-alfabet dari teks dan kunci
    teks = ''.join(c for c in teks if c.isalpha())
    kunci = ''.join(c for c in kunci if c.isalpha())

    if len(kunci) != len(teks):
        return False, f"Panjang kunci harus sama dengan teks ({len(teks)} karakter)"

    if not kunci.isalpha():
        return False, "Kunci hanya boleh berisi huruf"

    return True, "Valid"

def teks_angka(teks):
    """Mengubah teks menjadi angka (A=0, B=1, dst)"""
    return [ord(c.upper()) - ord('A') for c in teks if c.isalpha()]

def angka_teks(angka):
    """Mengubah angka menjadi teks"""
    return ''.join(chr(n + ord('A')) for n in angka)
```

Gambar 4. Script Code 1

```
def enkripsi(plaintext, kunci):
    """Enkripsi teks menggunakan operasi XOR"""
    # Hapus karakter non-alfabet dan ubah ke huruf besar
    plaintext = ''.join(c.upper() for c in plaintext if c.isalpha())
    kunci = ''.join(c.upper() for c in kunci if c.isalpha())

    # Validasi kunci
    valid, pesan = cek_kunci(plaintext, kunci)
    if not valid:
        return None, pesan

    # Ubah plaintext dan kunci menjadi angka
    no_plaintext = teks_angka(plaintext)
    no_kunci = teks_angka(kunci)

    # Enkripsi dengan operasi XOR
    no_ciphertext = [(p ^ k) % 26 for p, k in zip(no_plaintext, no_kunci)]

    # Ubah hasil enkripsi menjadi teks
    ciphertext = angka_teks(no_ciphertext)

    return ciphertext, "Enkripsi berhasil"
```

Gambar 5. Script Code 2

```
def dekripsi(ciphertext, kunci):
    """Dekripsi teks menggunakan operasi XOR"""
    # Validasi input
    valid, pesan = cek_kunci(ciphertext, kunci)
    if not valid:
        return None, pesan

    # Ubah ke huruf besar
    ciphertext = ciphertext.upper()
    kunci = kunci.upper()

    # Ubah ciphertext dan kunci menjadi angka
    no_ciphertext = teks_angka(ciphertext)
    no_kunci = teks_angka(kunci)

    # Dekripsi dengan operasi XOR
    no_plaintext = [(c ^ k) % 26 for c, k in zip(no_ciphertext, no_kunci)]

    # Ubah hasil dekripsi menjadi teks
    plaintext = angka_teks(no_plaintext)

    return plaintext, "Dekripsi berhasil"
```

Gambar 6. Script Code 3

```
def main():
    while True:
        print("\nProgram Enkripsi dan Dekripsi One-Time Pad Operasi XOR")
        print("1. Enkripsi")
        print("2. Dekripsi")
        print("3. Keluar")

        pilihan = input("Pilih menu (1-3): ")

        if pilihan == '1':
            plaintext = input("Masukkan Pesan: ")
            kunci = input("Masukkan kunci (huruf saja, panjang sama dengan teks): ")
            ciphertext, pesan = enkripsi(plaintext, kunci)
            if ciphertext is None:
                print(f"\nError: {pesan}")
            else:
                print(f"\nHasil Enkripsi: {ciphertext}")
                print(f"\nKunci: {kunci.upper()}")

        elif pilihan == '2':
            ciphertext = input("Masukkan Pesan: ")
            kunci = input("Masukkan kunci: ")
            plaintext, pesan = dekripsi(ciphertext, kunci)
            if plaintext is None:
                print(f"\nError: {pesan}")
            else:
                print(f"\nHasil Dekripsi: {plaintext}")
```

Gambar 7. Script Code 4

```
elif pilihan == '3':
    print("Terima kasih, Semoga Membantu!")
    break

else:
    print("Pilihan tidak valid!")

if __name__ == "__main__":
    main()
```

Gambar 8. Script Code 5

```
Program Enkripsi dan Dekripsi One-Time Pad Operasi XOR
1. Enkripsi
2. Dekripsi
3. Keluar
Pilih menu (1-3): 1
Masukkan Pesan: datadiri
Masukkan kunci (huruf saja, panjang sama dengan teks): locklock

Hasil Enkripsi: IORKIGTC
Kunci: LOCKLOCK

Program Enkripsi dan Dekripsi One-Time Pad Operasi XOR
1. Enkripsi
2. Dekripsi
3. Keluar
Pilih menu (1-3): 2
Masukkan Pesan: IORKIGTC
Masukkan kunci: locklock

Hasil Dekripsi: DATADIRI

Program Enkripsi dan Dekripsi One-Time Pad Operasi XOR
1. Enkripsi
2. Dekripsi
3. Keluar
Pilih menu (1-3): 3
Terima kasih, Semoga Membantu!
```

Gambar 9. Hasil Running

Perbandingan antara perhitungan manual dan skrip kode menghasilkan beberapa temuan penting. Perhitungan manual membutuhkan waktu lebih lama, terutama untuk teks biasa yang panjang, dan rentan terhadap kesalahan, terutama dalam konversi bit demi bit atau operasi XOR. Selain itu, metode ini tidak praktis untuk pesan yang panjang, tetapi membantu memahami cara kerja operasi XOR pada tingkat bit dan memungkinkan verifikasi yang lebih mudah terhadap keakuratan setiap langkah. Di sisi lain, penggunaan skrip kode lebih praktis dan lebih cepat dalam menangani teks panjang karena operasi XOR dilakukan secara otomatis, sehingga meminimalkan kesalahan manusia. Namun, skrip kode memiliki ketergantungan pada kode dan membutuhkan keamanan kunci yang ketat untuk mencegah akses yang tidak sah.

Dalam konteks pesan terenkripsi, algoritma OTP secara teoritis menawarkan keamanan sempurna dengan kondisi berikut: kunci sepenuhnya acak, panjang pesan, dan hanya digunakan sekali. Namun, tantangan utamanya terletak pada distribusi kunci yang sangat sulit. Sebaliknya, algoritma RSA memiliki keunggulan tersendiri. Keamanannya didasarkan pada kompleksitas faktorisasi bilangan besar, yang sulit dipecahkan oleh komputer klasik. RSA menggunakan kunci publik dan pribadi, memungkinkan distribusi kunci yang lebih fleksibel. Kunci publik dapat didistribusikan secara bebas, sedangkan kunci pribadi disimpan dengan aman oleh penerima.

Perbedaan mendasar antara keduanya terletak pada kepraktisan implementasinya. OTP sangat terbatas untuk komunikasi skala kecil dengan keamanan ekstrem, sedangkan RSA lebih cocok untuk sistem keamanan modern seperti enkripsi email, komunikasi internet aman (*HTTPS*), tanda tangan digital, dan otentikasi pengguna.

4. KESIMPULAN

Metode ini menjamin kerahasiaan absolut selama kunci benar-benar acak, tidak dapat diprediksi, dan hanya digunakan sekali. Keuntungan utama metode OTP adalah ketidakmungkinan mendekripsi pesan tanpa kunci yang benar, menjadikannya salah satu teknik kriptografi yang paling aman secara teoritis. Namun, tantangan utamanya adalah kompleksitas dalam distribusi kunci dan keterbatasan implementasi untuk komunikasi skala besar. Perhitungan manual berguna untuk memahami konsep dasar algoritma OTP, tetapi kurang efisien untuk pesan yang panjang. Penggunaan skrip kode menawarkan solusi praktis yang lebih cepat dan lebih andal untuk implementasi OTP. Kombinasi perhitungan manual dan skrip kode merupakan pendekatan ideal untuk pembelajaran dan implementasi skala besar.

Bagaimana mengatasi tantangan

distribusi kunci dalam algoritma OTP, penelitian lebih lanjut diperlukan untuk mengembangkan metode distribusi kunci yang efisien dan aman. Selain itu, kombinasi OTP dengan algoritma modern, seperti RSA atau *Advanced Encryption Standard* (AES), dapat menjadi alternatif untuk meningkatkan efisiensi dan fleksibilitas. Dalam pembelajaran dan implementasi, pendekatan yang menggabungkan perhitungan manual dan skrip kode dapat memberikan pemahaman mendalam serta efisiensi. Diharapkan penelitian ini dapat mendorong inovasi lebih lanjut dalam keamanan data digital di era teknologi modern.

REFERENSI

- [1] I. Artikel, "PROTECTING DATA IN THE DIGITAL WORLD: THE STRATEGIC ROLE OF," vol. 6, no. 2, pp. 540–549, 2024.
- [2] C. Huda, D. I. Mulyana, A. D. Prasetyo, and A. Y. Zulkarnain, "Implementasi Algoritma One Time Menggunakan Algoritma Chipper Transposition Sebagai Pengamanan Rahasia Pesan," vol. 03, no. 01, pp. 40–48, 2022.
- [3] F. Fitriyadi and A. Ahwan, "Implementasi Algoritma One Time Pad Untuk Enkripsi dan Dekripsi pada Peresepan Data Obat di Puskesmas Purwodiningratan Surakarta," *Smart Comp Jurnalnya Orang Pint. Komput.*, vol. 12, no. 4, 2023, doi: 10.30591/smartcomp.v12i4.3958.
- [4] G. Yulusianto *et al.*, "Pengamanan pesan rahasia memanfaatkan enkripsi modifikasi vigenere," vol. 8, no. 5, pp. 11069–11073, 2024.
- [5] A. I. Zalukhu, Z. Sitorus, and N. Septiani, "Enhancing Text Messages with a Combination of Vigenère Cipher and One Time Pad Using Random Key LFSR," vol. 6, no. 1, pp. 52–57, 2024.
- [6] J. Manullang, J. Tamba, and F. P. Sitohang, "Cryptography With One-Time Pad (OTP) Algorithm Xor Based," vol. 3, no. 02, pp. 54–60, 2024.
- [7] M. Andika, F. Rustan, L. Marcellino, K. Margi, K. Margi, and M. A. Putra, "ScienceDirect ScienceDirect Securing Text File Using Combination of Vigenere and One – Time Securing Text File Using Combination of Vigenere and One – Time Pad Cipher Algorithm Pad Cipher Algorithm," *Procedia Comput. Sci.*, vol. 227, pp. 1030–1038, 2023, doi: 10.1016/j.procs.2023.10.612.
- [8] A. Bulletin, O. F. Big, and D. Managment, "A Comparative Analysis of One-Time Pad and Contemporary Cryptographic Algorithms: Security, Quantum Resistance, and Practical Usability," vol. 4, no. 4, pp. 284–292, 2024.
- [9] M. Derese and T. Birara, "Enhancing the Hill Cipher Algorithm and Employing a One Time Pad Key Generation Technique," vol. 3, no. 1, pp. 1–10, 2023.
- [10] I. Technology, I. Faisal, I. Lubis, P. Studi, T. Informatika, and U. H. Medan, "Metode One Time Pad sebagai Verifikasi Akun E-Wallet dalam Pencegahan Cybercrime," vol. 5, no. 1, pp. 59–72, 2025.
- [11] M. D. Asrofa, S. Bahri, and U. Tanjungpura, "One-Time Pad Cryptography for Secure Data Transmission in IoT Smart Door Using QR Code," vol. 2, no. 2, 2025.
- [12] H. Putri, L. Virna, T. Febrianti, and T. Sutabri, "Pengamanan Data Transmisi Aplikasi Web Menggunakan Algoritma Kriptografi RSA: Studi Kasus dan Analisis," vol. 5, no. 1, pp. 153–170, 2025.
- [13] K. Malalayang, K. Manado, and S. Utara, "IMPLEMENTASI KRIPTOGRAFI DALAM PENGAMANAN DATA GAMBAR," vol. XIII, no. 1, pp. 93–99, 2024.
- [14] O. Access, "Audio Encryption and Decryption using the Affine Cipher with the One-Time Pad (OTP)," vol. 3, no. 3, pp. 28–44, 2025.
- [15] M. Pasaribu, S. Yurinanda, and N. Rarasati,

“NPWP ENCRYPTION USES A
COMBINATION OF ONE TIME PAD
ALGORITHM AND EXCLUSIVE-OR

OPERATION,” vol. 8, pp. 196–216, 2024.