Terbit online pada laman web jurnal: **https://jurnal.plb.ac.id/index.php/tematik/index**

T E M A T I K

Jurnal **Teknologi Informasi Komunikasi (e-Journal)**

Vol. 12 No. 1 (2025) 118 - 124 ISSN Media Elektronik: 2443-3640

Pengamanan Dokumen Digital Menggunakan Kombinasi Algoritma Enkripsi Blowfish dan *Encoding* Base64

Digital Document Security Using a Combination of Blowfish Encryption Algorithm and Base64 Encoding

Muhammad Ziad Ziayuddin[1], Imelda[2]
[1]Magister Ilmu Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur
[2]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1]2211600545@student.budiluhur.ac.id*, [2]imelda@budiluhur.ac.id

Abstract

In the digital era, document protection is crucial to safeguard the confidentiality and integrity of organizational information against various security threats. Numerous data breach incidents occur due to weak protection of internal documents that are not properly encrypted. This study aims to develop and evaluate a multi-format digital document security system by combining the Blowfish encryption algorithm with Base64 encoding. Blowfish serves as the main encryption algorithm to convert plaintext into binary ciphertext, while Base64 is used to convert the encrypted output into ASCII text format to facilitate storage and transmission, but not as a primary security mechanism. The system is implemented as a web-based application to enable convenient access and operation through a browser interface. The research methodology includes problem identification, data collection using Excel documents, system design, algorithm implementation, and testing. Functional validation and brute-force resistance tests were conducted. The results show that the system successfully performed encryption and decryption with 100% accuracy on five test files without data loss. The encrypted file size increased by approximately 0.3% due to the encoding process, which remains within acceptable limits. Security testing using CrypTool indicated that the ciphertext could not be deciphered without a valid key, even under systematic key search attempts. The primary contribution of this study is the integration of Blowfish encryption and Base64 encoding into an efficient web-based digital document security system, validated for brute-force resistance, which has not been widely explored in previous research.

Keywords: document encryption, Blowfish algorithm, Base64 encoding, digital data security, brute force

Abstrak

Dalam era digital, perlindungan dokumen menjadi hal krusial untuk menjaga kerahasiaan dan integritas informasi organisasi dari berbagai ancaman keamanan. Banyak insiden kebocoran data terjadi akibat lemahnya proteksi terhadap dokumen internal yang belum terenkripsi dengan baik. Penelitian ini bertujuan untuk mengembangkan dan menguji sistem pengamanan dokumen digital multi-format melalui kombinasi algoritma Blowfish dan Base64. Blowfish berperan sebagai algoritma enkripsi utama untuk mengubah *plaintext* menjadi *ciphertext* biner, sementara Base64 digunakan untuk mengonversi hasil enkripsi ke dalam format teks ASCII guna memudahkan penyimpanan dan transmisi data, namun bukan sebagai mekanisme keamanan utama. Sistem ini dibangun sebagai *web-based* system agar dapat diakses dan dioperasikan secara praktis melalui antarmuka web. Metodologi penelitian mencakup identifikasi masalah, pengumpulan data berupa dokumen Excel, perancangan sistem, implementasi algoritma, serta pengujian dan evaluasi. Pengujian dilakukan melalui validasi fungsional dan uji ketahanan terhadap serangan *brute force*. Hasil menunjukkan bahwa sistem mampu melakukan proses enkripsi dan dekripsi dengan akurasi 100% pada lima *file* uji tanpa kehilangan data. Ukuran *file* terenkripsi meningkat sebesar ±0,3% karena proses *encoding*, namun masih dalam batas toleransi. Uji keamanan menggunakan CrypTool memperlihatkan bahwa *ciphertext* tidak dapat dibuka tanpa kunci sah, bahkan ketika dilakukan pencarian kunci secara sistematis. Kontribusi utama penelitian ini adalah integrasi algoritma Blowfish dan *encoding* Base64 ke dalam sistem enkripsi dokumen digital berbasis web yang efisien, dengan validasi keamanan terhadap *brute force*, yang belum banyak diterapkan dalam studi-studi sebelumnya.

Kata kunci: enkripsi dokumen, algoritma Blowfish, *encoding* Base64, keamanan data digital, *brute force*

1. Pendahuluan

Dalam era digital, dokumen yang berisi informasi strategis perusahaan menjadi sasaran utama ancaman keamanan siber. Banyak insiden kebocoran data disebabkan oleh lemahnya proteksi terhadap dokumen internal yang belum terenkripsi secara memadai [1]. Kondisi ini tidak hanya menimbulkan kerugian finansial, tetapi juga berpotensi merusak reputasi organisasi [2]. Oleh karena itu, pengembangan metode

Diterima Redaksi: 11-05-2025 | Selesai Revisi: 30-06-2025 | Diterbitkan Online: 01-07-2025

pengamanan dokumen digital yang andal menjadi semakin penting.

Berbagai studi telah menunjukkan efektivitas algoritma Blowfish dan Base64 dalam pengamanan data digital. Terdapat penelitian yang membuktikan bahwa Blowfish mampu mengamankan database SQL dengan kunci yang tahan terhadap serangan *brute force* [3]. Penelitian selnajutnya, mengimplementasikan Blowfish dalam sistem email berbasis *web* untuk menjaga kerahasiaan pesan [4]. Di sisi lain, Base64 telah digunakan secara luas untuk *encoding* data menjadi format ASCII agar lebih mudah ditransmisikan [5]. Penelitian berikutnya juga menerapkan Base64 dalam pengamanan file video [6]. Meskipun demikian, sebagian besar penelitian terdahulu menerapkan algoritma Blowfish atau Base64 secara terpisah, atau mengombinasikannya dalam konteks terbatas, seperti database atau pesan teks sederhana. amun, sebagian besar penelitian terdahulu masih terbatas pada penerapan algoritma secara parsial atau dalam konteks tertentu, tanpa mengevaluasi ketahanannya terhadap serangan kriptografi secara komprehensif, terutama dalam sistem berbasis web untuk file dokumen multi-format.

Penelitian ini menghadirkan kebaruan dengan menggabungkan algoritma Blowfish dan Base64 ke dalam satu sistem pengamanan dokumen digital berbasis web, dan secara eksplisit menguji ketahanannya terhadap serangan *brute force* menggunakan alat bantu CrypTool. Blowfish dipilih karena merupakan algoritma enkripsi blok yang cepat, ringan, dan fleksibel dengan panjang kunci hingga 448-*bit*, sehingga efisien untuk file dokumen [7]. Sementara itu, Base64 digunakan untuk mengonversi hasil enkripsi menjadi format teks ASCII, sehingga memudahkan penyimpanan dan transmisi sekaligus menyamarkan pola data terenkripsi [8]. Pada sistem yang diusulkan, Blowfish berfungsi sebagai enkripsi utama sedangkan Base64 sebagai lapisan *encoding* tambahan.

Untuk mengevaluasi ketahanan sistem, pengujian menggunakan metode *brute force* dipilih. *Brute force* merupakan pendekatan serangan dasar dalam kriptografi yang mencoba seluruh kemungkinan kunci hingga menemukan kecocokan [9]. Pengujian ini penting untuk menilai kekuatan kombinasi algoritma terhadap upaya pembobolan dengan sumber daya komputasi terbatas, sehingga dapat menunjukkan tingkat ketahanan enkripsi yang diterapkan [10].

Tujuan dari penelitian ini adalah membangun dan menguji sistem enkripsi-dekripsi dokumen multi-format menggunakan kombinasi algoritma Blowfish dan Base64, serta menganalisis ketahanannya terhadap serangan *brute force*. Proses enkripsi dilakukan terhadap file dengan format umum seperti .docx, .pdf, .xls, dan .ppt, diikuti dengan *encoding* hasil enkripsi ke dalam format teks ASCII. Kontribusi utama penelitian ini adalah perancangan dan implementasi sistem pengamanan dokumen berbasis web yang mengombinasikan algoritma Blowfish dan *encoding* Base64, serta validasi ketahanan sistem terhadap *brute force* secara eksperimental.

2. Metode Penelitian

Penelitian ini bertujuan untuk mengembangkan sistem pengamanan dokumen digital multi-format menggunakan kombinasi algoritma Blowfish dan Base64. Tahapan penelitian yang jelas dan terstruktur sangat penting dalam memastikan bahwa sistem yang dikembangkan dapat memenuhi tujuan dan persyaratan yang ditetapkan [11]. Tahapan penelitian yang diterapkan pada penelitian ini divisualisasikan pada Gambar 1.

Gambar 1. Tahapan Penelitian

Gambar 1 menunjukkan diagram alur prosedur penelitian yang menggambarkan langkah-langkah utama dalam pengembangan sistem pengamanan dokumen digital menggunakan algoritma Blowfish dan Base64. Penjelasan terinci terkait prosedur penelitian dijelaskan sebagai berikut.

2.1. Identifikasi Masalah

Tahap awal dilakukan dengan mengidentifikasi permasalahan dalam sistem pengamanan dokumen digital, khususnya pada lingkungan organisasi yang masih menggunakan sistem penyimpanan file tanpa

DOI: https://doi.org/10.38204/tematik.v12i1.2316

enkripsi. Ancaman seperti akses tidak sah, pencurian informasi, dan kebocoran data menjadi isu krusial yang melatarbelakangi penelitian ini. Studi literatur dan observasi menunjukkan bahwa solusi pengamanan dokumen yang hanya menggunakan satu algoritma cenderung tidak cukup dalam menghadapi serangan yang semakin kompleks. Oleh karena itu, diperlukan pendekatan yang mengombinasikan dua metode, yaitu algoritma Blowfish untuk enkripsi dan Base64 untuk *encoding*, dalam satu sistem pengamanan yang utuh.

2.2. Pengumpulan Data

Setelah permasalahan diidentifikasi, tahap selanjutnya adalah pengumpulan data berupa dokumen digital yang digunakan sebagai objek pengujian sistem. Data yang digunakan dalam penelitian ini terdiri dari lima file berformat ".xlsx" (Microsoft Excel), yang diperoleh dari lingkungan kerja PT. INTEGRA pada tahun 2022. File-file tersebut dipilih secara purposif untuk merepresentasikan dokumen operasional yang umum digunakan dalam organisasi.

Data ini digunakan untuk menguji keakuratan proses enkripsi-dekripsi, serta mengevaluasi ketahanan sistem terhadap upaya pembobolan melalui metode *brute force attack*.

2.3. Perancangan Sistem

Pada tahap ini dirancang arsitektur sistem pengamanan dokumen berbasis *web* dengan dua komponen utama: modul enkripsi menggunakan algoritma Blowfish, dan modul *encoding* menggunakan Base64.

Sistem juga mencakup fitur *login* untuk autentikasi pengguna, serta antarmuka pengguna untuk mengunggah file, memasukkan *password* enkripsi, dan mengunduh file hasil enkripsi maupun dekripsi. Desain sistem mempertimbangkan aspek efisiensi pemrosesan, keamanan algoritma, dan kemudahan penggunaan.

2.4. Penerapan Algortima Pengaman Dokumen

Sistem pengamanan dokumen yang dikembangkan dalam penelitian ini menerapkan dua algoritma utama, yaitu Blowfish dan Base64. Keduanya memiliki peran yang saling melengkapi dalam membentuk lapisan pengamanan data digital. Blowfish digunakan sebagai algoritma utama untuk proses enkripsi dan dekripsi, sedangkan Base64 digunakan untuk menyandikan hasil enkripsi ke dalam format teks ASCII agar lebih aman dan kompatibel saat disimpan atau ditransmisikan. Penjelasan konseptual masing-masing algoritma dijelaskan sebagai berikut.

Base64 bukanlah algoritma enkripsi, melainkan metode *encoding* yang digunakan untuk mengonversi data biner menjadi format teks ASCII [12]. Teknik ini banyak digunakan dalam sistem transmisi data seperti email dan HTTP karena menghasilkan teks ASCII yang aman untuk media komunikasi berbasis teks [13]. Proses *encoding* Base64 membagi data biner menjadi kelompok 6-bit, kemudian setiap kelompok direpresentasikan dengan karakter dalam set alfabet Base64 yang terdiri dari 64 simbol (huruf A–Z, a–z, angka 0–9, serta simbol + dan /). Indeks Base64 ditampilkan pada Tabel 1.

Tabel 1. Tabel Indeks Base64

Data 6-bit	Karakter *Encoding* 64	Data 6-bit	Karakter *Encoding* 64	Data 6-bit	Karakter *Encoding* 64
0	A	21	V	42	q
1	B	22	W	43	r
2	C	23	X	44	s
3	D	24	Y	45	t
4	E	25	Z	46	u
5	F	26	a	47	v
6	G	27	b	48	w
7	H	28	c	49	x
8	I	29	d	50	y
9	J	30	e	51	z
10	K	31	f	52	0
11	L	32	g	53	1
12	M	33	h	54	2
13	N	34	i	55	3
14	O	35	j	56	4
15	P	36	k	57	5
16	Q	37	l	58	6
17	R	38	m	59	7
18	S	39	n	60	8
19	T	40	o	61	9
20	U	41	p	62	+
				63	/
				pad	=

Tabel 1 menunjukkan pemetaan antara nilai desimal 6-bit dan karakter Base64 yang sesuai. Pemetaan ini digunakan dalam proses *encoding* data biner ke dalam teks ASCII. Proses ini menjadikan data terenkripsi yang semula berbentuk biner dapat dikodekan menjadi teks yang aman dan lebih mudah untuk disimpan atau ditransmisikan melalui media digital seperti email, URL, atau database teks [14]. Dalam konteks sistem yang dikembangkan, Base64 berperan sebagai lapisan tambahan yang menyamarkan pola *ciphertext* hasil Blowfish, sekaligus meningkatkan kompatibilitas data hasil enkripsi dalam berbagai format penyimpanan.

Blowfish merupakan algoritma kriptografi simetris yang dirancang oleh Bruce Schneier pada tahun 1993 [15]. Algoritma ini menggunakan teknik enkripsi berbasis blok (block cipher) dengan panjang blok 64-bit dan kunci variabel yang dapat mencapai hingga 448-bit [16]. Algoritma ini dikenal karena efisiensinya dalam perangkat lunak dan kemampuannya memberikan keamanan tinggi tanpa memerlukan sumber daya komputasi yang besar [17]. Blowfish termasuk ke dalam struktur *Feistel cipher* dan terdiri dari 16 putaran enkripsi. Setiap putaran melibatkan pembangkitan subkunci yang kompleks melalui penggunaan *P-array* dan *S-box*, serta penerapan fungsi *F* yang menggabungkan operasi XOR, penjumlahan modulo 2^{32}, dan substitusi non-linier [18]. Inisialisasi awal *P*-

DOI: https://doi.org/10.38204/tematik.v12i1.2316

Array pada algoritma Blowfish ditunjukkan pada Tabel 2.

Tabel 2. Inisialisasi Awal *P-Array* pada Algoritma Blowfish

P-Array	Hexa	Konversi Biner (32 bit)
P0	243F6A88	00100100 00111111 01101010 10001000
P1	85A308D3	10000101 10100011 00001000 11010011
P2	13198A2E	00010011 00011001 10001010 00101110
P3	3707344	00000011 01110000 01110011 01000100
P4	A4093822	10100100 00001001 00111000 00100010
P5	299F31D0	00101001 10011111 00110001 11010000
P6	82EFA98	00001000 00101110 11111010 10011000
P7	EC4E6C89	11101100 01001110 01101100 10001001
P8	452821E6	01000101 00101000 00100001 11100110
P9	38D01377	00111000 11010000 00010011 01110111
P10	BE5466CF	10111110 01010100 01100110 11001111
P11	34E90C6C	00110100 11101001 00001100 01101100
P12	C0AC29B7	11000000 10101100 00101001 10110111
P13	C97C50DD	11001001 01111100 01010000 11011101
P14	3F84D5B5	00111111 10000100 11010101 10110101
P15	B5470917	10110101 01000111 00001001 00010111
P16	9216D5D9	10010010 00010110 11010101 11011001

Tabel 2 menunjukkan nilai heksadesimal dan biner 32-bit yang digunakan sebagai nilai awal dalam proses pembangkitan subkunci pada algoritma Blowfish. Nilai-nilai ini bersifat tetap dan digunakan sebagai bagian dari struktur internal cipher.

Keunggulan Blowfish terletak pada kecepatannya dalam pemrosesan dan efisiensinya dalam aplikasi perangkat lunak [19]. Dalam penelitian ini, Blowfish digunakan untuk mengubah *plaintext* dokumen menjadi *ciphertext* biner menggunakan kunci yang dimasukkan oleh pengguna, sehingga menjamin kerahasiaan isi dokumen dari akses tidak sah.

2.5. Pengujian dan Evaluasi

Tahap akhir dalam penelitian ini adalah pengujian dan evaluasi sistem untuk menilai fungsionalitas serta ketahanan algoritma pengamanan dokumen yang telah diimplementasikan. Pengujian ini penting dilakukan untuk memastikan bahwa sistem tidak hanya bekerja secara teoritis, tetapi juga mampu beroperasi secara andal dan aman dalam kondisi penggunaan nyata [20]. Pengujian dilakukan melalui dua pendekatan. Pertama, pengujian validasi fungsional bertujuan untuk memastikan bahwa file yang telah dienkripsi tidak dapat diakses tanpa proses dekripsi, serta dapat dikembalikan secara akurat ke bentuk semula tanpa kehilangan data. Kedua, dilakukan pengujian ketahanan terhadap serangan *brute force* menggunakan aplikasi *CrypTool*, guna mengevaluasi sejauh mana sistem dapat menahan upaya pembobolan melalui pencarian kunci secara acak tanpa informasi awal. Kedua jenis pengujian ini memberikan gambaran menyeluruh terhadap keandalan sistem baik dari sisi keakuratan maupun aspek keamanannya.

3. Hasil dan Pembahasan

Sistem pengamanan dokumen yang dikembangkan dalam penelitian ini mengombinasikan dua algoritma, yaitu Blowfish dan Base64, untuk memberikan dua lapisan pengamanan terhadap dokumen digital. Blowfish digunakan sebagai algoritma enkripsi utama yang mengubah isi file menjadi *ciphertext* biner menggunakan kunci rahasia dari pengguna. Selanjutnya, hasil *ciphertext* dilakukan *encoding* menggunakan Base64 agar data biner dapat diubah menjadi format teks ASCII yang aman untuk penyimpanan dan transmisi. Studi kasus dilakukan untuk menunjukkan bagaimana algoritma ini bekerja pada data nyata dan menghasilkan proses enkripsi-dekripsi yang valid dan akurat.

Algoritma pertama yang digunakan yaitu Base64, dimana pendekatan ini merupakan metode *encoding* yang mengubah data biner ke dalam format teks ASCII, tanpa menggunakan kunci. Proses ini digunakan setelah enkripsi untuk menyamarkan *ciphertext* dan membuatnya lebih mudah ditransmisikan atau disimpan. Sebagai ilustrasi, diberikan teks "BUDI LUHUR" sebagai input. Langkah-langkah konversi ditunjukkan pada Tabel 3.

Tabel 3. Konversi Teks “BUDI LUHUR” Menggunakan Base64

Langkah	Nilai
ASCII	66 85 68 49 32 76 85 72 85 82
Biner 8-bit	01000010 01010101 ...
Biner 6-bit	010000 010101 ...
Desimal	16 21 17 18 8 19 21 18 21 20
Base64	QVRSITVSVU

Tabel 3 menunjukkan tahapan *encoding* dari ASCII hingga representasi Base64 akhir. Proses *decoding* dilakukan secara terbalik, dimulai dari penguraian string Base64 menjadi indeks desimal, lalu ke biner 6-bit, dan digabung kembali ke bentuk 8-bit sebelum dikonversi menjadi karakter ASCII. Hasil akhir menunjukkan bahwa teks asli dapat diperoleh kembali secara utuh tanpa adanya perubahan, sehingga validitas proses *encoding-decoding* dalam sistem terbukti bekerja secara akurat.

Selanjutnya, pada algoritma Blowfish dilakukan studi kasus terhadap teks “UPI YPTK” dengan menggunakan kunci “2905”. Proses enkripsi diawali dengan pembangkitan subkunci melalui operasi XOR terhadap

DOI: https://doi.org/10.38204/tematik.v12i1.2316

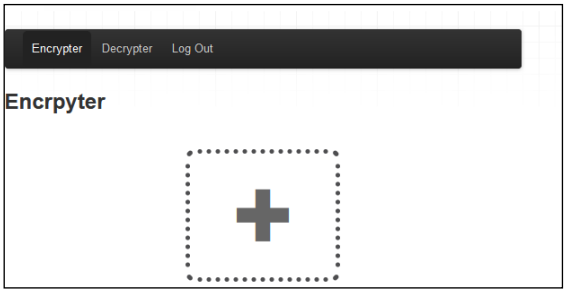
P-Array. Dua hasil awal dari proses pembangkitan subkunci ditampilkan pada Tabel 4.

Tabel 4. Contoh Pembangkitan Subkunci Blowfish

Subkunci	Operasi XOR	Hasil (Biner)
P0	P0 XOR Kunci	00010110 00000110 ...
P1	P1 XOR P0	10010011 10100101 ...

Tabel 4 menampilkan hasil XOR antara nilai awal dan kunci untuk menghasilkan subkunci yang digunakan dalam iterasi. Teks kemudian dikonversi ke biner dan dibagi menjadi dua bagian (XL dan XR), masing-masing sepanjang 32 bit. Proses enkripsi dilakukan dengan struktur Feistel, di mana fungsi F diterapkan pada XL dan hasilnya di-XOR dengan XR. Setelah iterasi dan pertukaran blok selesai, hasil akhir digabung dan dikonversi kembali menjadi karakter ASCII. *Ciphertext* yang dihasilkan adalah "Ü-/oe|9‹", sedangkan *plaintext* hasil dekripsi adalah "UPI YPTK". Hasil ini menunjukkan bahwa algoritma Blowfish yang diimplementasikan dapat menjalankan proses enkripsi dan dekripsi dengan hasil yang valid dan dapat diandalkan.

Implementasi kedua algoritma tersebut diintegrasikan dalam sebuah sistem berbasis web yang ramah pengguna. Proses interaksi dimulai dari halaman login, tempat pengguna memasukkan username dan *password* yang telah didaftarkan oleh administrator. Setelah berhasil masuk, pengguna akan diarahkan ke halaman utama yang menyediakan dua menu utama: *Encrypter* dan *Decrypter*. Pada menu *Encrypter*, pengguna dapat memilih file yang ingin dienkripsi dengan mengklik ikon [+] seperti pada Gambar 2.

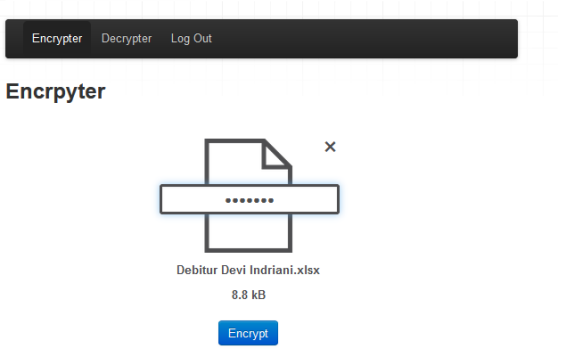


Gambar 2. Tampilan Fitur *Encrypter*

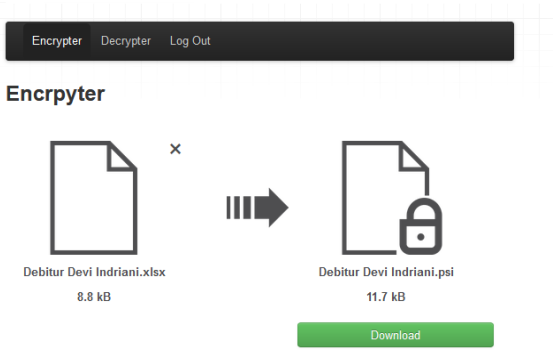
Gambar 2 menunjukkan fitur dimana pengguna dapat memasukkan file yang akan dienkripsi. Kemudian pengguna dapat memasukkan *password* sebagai kunci enkripsi seperti pada Gambar 3.

Setelah menekan tombol *Encrypt*, sistem akan memproses file dan menampilkan file terenkripsi yang dapat diunduh, seperti pada Gambar 4.

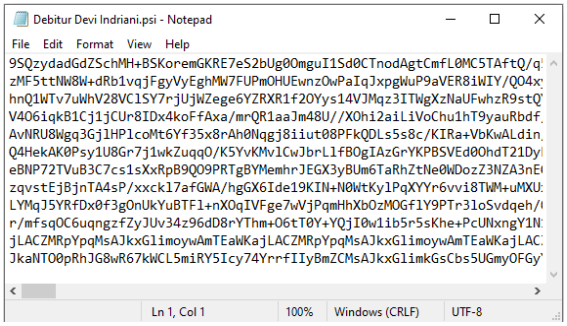
Setelah proses enkripsi dan pengunduhan selesai, file yang dihasilkan disimpan dalam format ".psi" sebagai hasil akhir enkripsi. Ilustrasi tampilan file terenkripsi dapat dilihat pada Gambar 5.



Gambar 3. Fitur Memasukkan *Password* Enkripsi

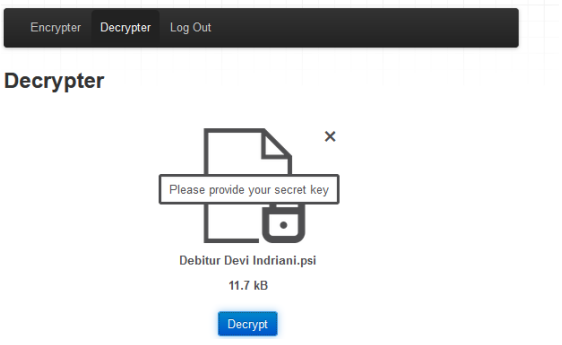


Gambar 4. Fitur Download File Hasil Enkripsi



Gambar 5. File Hasil Enkripsi

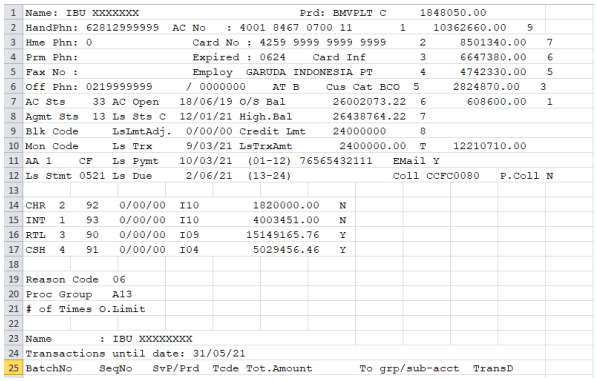
Selanjutnya, proses dekripsi dilakukan melalui tahapan yang serupa dengan proses enkripsi. Pengguna mengakses menu *Decrypter*, kemudian memilih file yang telah terenkripsi, memasukkan *password* yang sama seperti pada saat enkripsi, dan menekan tombol *Decrypt*, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6. Fitur Memasukkan *Password* Deskripsi

DOI: https://doi.org/10.38204/tematik.v12i1.2316

Gambar 6 menunjukkan fitur memasukkan *password* untuk deskripsi file. Apabila *password* benar, file akan dikembalikan ke bentuk semula dan ditampilkan hasilnya, yang dapat diunduh. Visualisasi hasil dari file yang telah berhasil dienkripsi ditampilkan pada Gambar 7.



Gambar 7. File Hasil Deskripsi

Untuk mengevaluasi efektivitas sistem, dilakukan dua jenis pengujian. Pengujian pertama adalah validasi fungsional, yang bertujuan memastikan bahwa file yang telah dienkripsi tidak dapat dibuka tanpa proses dekripsi, dan bahwa hasil dekripsi identik dengan file asli. Pengujian dilakukan pada lima file Excel. Hasil pengujian validasi enkripsi dan dekripsi dokumen ditunjukkan pada Tabel 5.

Tabel 5. Hasil Pengujian Enkripsi-Dekripsi Dokumen Excel

No	Nama File	Ukuran Awal (KB)	Ukuran Enkripsi (KB)	Waktu Enkripsi (detik)
1	Laporan 1	8.8	11.7	125
2	Laporan 2	9.1	12.1	242
3	Laporan 3	9.4	12.6	239
4	Laporan 4	9.2	12.3	248
5	Laporan 5	9.4	12.6	237

Tabel 5 menyajikan hasil pengujian yang menunjukkan bahwa file hasil enkripsi dengan ekstensi .psi mengalami peningkatan ukuran sekitar ± 0,3% akibat proses *encoding* menggunakan algoritma Base64. Meskipun demikian, seluruh file yang telah didekripsi kembali memiliki ukuran dan isi yang identik dengan file aslinya. Tingkat keberhasilan proses enkripsi-dekripsi mencapai 100%, yang dibuktikan oleh kesesuaian sempurna antara file hasil dekripsi dan file asli, baik dari sisi ukuran maupun konten dokumen. Hasil ini diperoleh dari lima file uji yang seluruhnya berhasil dikembalikan ke bentuk semula tanpa perubahan atau kerusakan data.

Pengujian kedua adalah uji ketahanan terhadap serangan *brute force*, yang bertujuan untuk mengetahui seberapa kuat sistem dalam menghadapi upaya pembobolan melalui pencarian kunci secara acak. Uji coba dilakukan menggunakan aplikasi CrypTool terhadap lima file .psi dengan ukuran berbeda. Hasil pengujian ketahanan terhadap serangan *brute force* ditampilkan pada Tabel 6.

Tabel 6. Hasil Pengujian Ketahanan terhadap *Brute Force* Menggunakan CrypTool

No	Nama File	Ukuran File (KB)	Lama *Brute Force*	Hasil
1	Laporan Cisco	86	1 Jam 2 Menit	Tidak Terbuka
2	Cisco Catalyst 2960	26	45 Menit	Tidak Terbuka
3	Laporan.psi	1.1	10 Menit	Tidak Terbuka
4	Cisco.psi	9.4	28 Menit	Tidak Terbuka
5	Generate.psi	15	36 Menit	Tidak Terbuka

Pada Tabel 6 menunjukkan bahwa bahwa waktu yang dibutuhkan untuk proses *brute force* bervariasi tergantung pada ukuran file yang diuji. Namun, tidak ada satu pun file yang berhasil didekripsi menjadi *plaintext*. Teks hasil proses *brute force* berupa karakter-karakter acak yang tidak memiliki makna dan tidak dapat dikenali sebagai isi dokumen.

Berdasarkan hasil kedua jenis pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem pengamanan dokumen digital yang dikembangkan dengan kombinasi algoritma Blowfish dan Base64 menunjukkan performa yang efektif. Sistem ini tidak hanya mampu mengamankan dokumen dalam berbagai format, tetapi juga berhasil mengembalikan file ke bentuk aslinya secara utuh setelah proses dekripsi. Peningkatan ukuran file sebagai akibat dari proses *encoding* Base64 terpantau relatif kecil dan masih berada dalam batas yang dapat ditoleransi.

Selain itu, hasil pengujian terhadap serangan *brute force* menunjukkan bahwa sistem memiliki ketahanan yang baik. Upaya pembobolan menggunakan metode pencarian kunci secara sistematis melalui perangkat lunak khusus tidak berhasil mengungkap isi file terenkripsi, sehingga menegaskan bahwa metode enkripsi yang digunakan cukup kuat untuk menghadapi serangan tersebut.

Dengan demikian, sistem yang diusulkan layak diterapkan dalam lingkungan organisasi yang membutuhkan tingkat keamanan tinggi terhadap dokumen digital, terutama dalam menjaga kerahasiaan dan integritas data. Meski demikian, perlu dicatat bahwa pengujian dalam penelitian ini masih terbatas pada skenario *brute force*. Jenis serangan lainnya, seperti *statistical cryptanalysis*, *differential cryptanalysis*, maupun *side-channel attacks*, belum dievaluasi dan menjadi ruang terbuka untuk pengembangan dan pengujian lebih lanjut di masa mendatang.

DOI: https://doi.org/10.38204/tematik.v12i1.2316

4. Kesimpulan

Penelitian ini berhasil mengembangkan sistem pengamanan dokumen digital berbasis web dengan mengombinasikan algoritma Blowfish dan *encoding* Base64. Sistem mampu melakukan proses enkripsi dan dekripsi secara akurat, dengan tingkat keberhasilan 100% pada lima dokumen uji tanpa kehilangan informasi. Ukuran *file* hasil enkripsi meningkat secara moderat sebesar ±0,3% akibat proses *encoding*, namun tetap berada dalam batas toleransi dan tidak berdampak signifikan pada efisiensi penyimpanan. Waktu proses enkripsi yang relatif cepat dan ukuran *file* yang efisien mendukung klaim bahwa sistem ini layak untuk diimplementasikan dalam lingkungan organisasi yang membutuhkan keamanan dokumen multi-format. Pengujian ketahanan terhadap serangan *brute force* menunjukkan bahwa *ciphertext* tidak dapat dipecahkan tanpa kunci sah, memperkuat aspek kriptografis sistem. Meskipun demikian, keterbatasan penelitian ini terletak pada jumlah sampel dokumen uji yang terbatas, sehingga hasil belum dapat digeneralisasi secara menyeluruh untuk semua jenis *file* atau ukuran data. Untuk pengembangan selanjutnya, disarankan agar sistem diuji menggunakan lebih banyak sampel dan format dokumen dengan kompleksitas bervariasi. Penelitian lanjutan juga dapat mengeksplorasi integrasi algoritma tambahan seperti hashing untuk verifikasi integritas atau *digital signature* untuk otentikasi, serta membandingkan performa dengan algoritma lain seperti AES guna memperoleh hasil yang lebih komprehensif.

Daftar Rujukan

[1] N. Bahtiar, "Darurat Kebocoran Data: Kebutuhan Regulasi Pemerintah," *Dev. Policy Manag. Rev.*, vol. 2, no. 1, pp. 1–16, 2022.

[2] A. D. Saputra, F. Dione, and I. Uluputty, "Pengelolaan Keamanan Informasi dan Persandian di Dinas Komunikasi dan Informatika Provinsi Kalimantan Timur," *J. Teknol. dan Komun. Pemerintah.*, vol. 5, no. 2, pp. 159–187, 2023, doi: 10.33701/jtkp.v5i2.3735.

[3] A. Juliardi and S. M. Ladjamuddin, "Rancang Bangun Aplikasi Enkripsi dan Dekripsi Pada Database SQL Permikomas Menggunakan Algoritma Blowfish," *Incomtech*, vol. 27, no. 2, pp. 635–637, 2021.

[4] S. Megira, "Implementasi Algoritma Blowfish Pada Aplikasi Pengamanan Surat Elektronik (Mail Client) Berbasis Web," *Siskomti*, vol. 4, no. 2, pp. 16–25, 2021.

[5] D. P. Nafisah, R. Syavana, I. M. Akbar, J. Salsabilla Berliana, and N. Naissha, "Pengamanan Data Menggunakan Algoritma Base64," *J. SITEBA*, vol. 2, no. 1, pp. 19–23, 2023.

[6] D. I. G. Hutasuhut, N. Fadillah, E. S. Rahayu, and A. Windi, "Implementasi dan Penggunaan Algoritma Base64 Dalam Pengamanan File Video," *UNES J. Inf. Syst.*, vol. 8, no. 1, pp. 34–41, 2023.

[7] H. Hairullah, C. R. A. Pramartha, and I. A. G. S. Putra, "Aplikasi Keamanan E-Commerce Berbasis Web Menggunakan Metode Algoritma Blowfish," *JNATIA J. Nas. Teknol. Inf. dan Apl.*, vol. 1, no. 1, pp. 79–88, 2022.

[8] A. F. Cobantoro, M. B. Setyawan, and H. Oktavianto, "Rekayasa Aplikasi Eposal Menggunakan Algoritma Base64 Untuk Menyimpan Data Pengguna," *J. Komtika (Komputasi dan Inform.*, vol. 7, no. 1, pp. 31–38, 2023, doi: 10.31603/komtika.v7i1.8711.

[9] I. Gunawan, "Peningkatan Pengamanan Data File Menggunakan Algoritma Kriptografi AES Dari Serangan Brute Force," *TECHSI - J. Tek. Inform.*, vol. 13, no. 1, p. 14, 2021, doi: 10.29103/techsi.v13i1.2395.

[10] A. Rohman and A. Munawir, "Implementasi Enkripsi File Berbasis Cryptography Untuk Keamanan Data Di Windows 10 Menggunakan Algoritma (Aes)," *J. Ilm. Sist. Inf.*, vol. 3, no. 1, pp. 149–159, 2023, doi: 10.46306/sm.v3i1.90.

[11] Y. Fernando, R. Napianto, and R. I. Borman, "Implementasi Algoritma Dempster-Shafer Theory Pada Sistem Pakar Diagnosa Penyakit Psikologis Gangguan Kontrol Impuls," *Insearch Inf. Syst. Res. J.*, vol. 2, no. 2, pp. 46–54, 2022.

[12] F. F. Oktaviany and E. Ardhianto, "Pengamanan Basis Data Kasus Kekerasan pada Perempuan dan Anak Menggunakan Algoritma Vigenere Cipher dan Base64," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 1, pp. 194–201, 2024, doi: 10.35870/jtik.v8i1.1311.

[13] T. Lovian and I. Fitri, "Implementasi Algoritma Base64 Sebagai Tingkat Keamanan Data Pada Website Sistem Informasi Pencatat Barang," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 692, 2022, doi: 10.30865/mib.v6i1.3513.

[14] Y. P. Putra, F. Nuraeni, and R. Ajji Jatnika, "Implementasi Kriptografi Dalam Pengamanan Database E-Voting Menggunakan Algoritma Rsa Dan Base64 Berbasis Progresive Web Apps (Studi Kasus: Pemilihan Presiden Mahasiswa STMIK Tasikmalaya)," *J. Sist. Inf. dan Teknol. Inf.*, vol. 10, no. 1, pp. 30–40, 2021.

[15] N. Permatasari and Y. Mardiana, "Aplikasi Penyandian Pesan Teks Berbasis Web Menggunakan Algoritma Blowfish," in *Prosiding SNASIKOM*, 2023, pp. 61–68.

[16] N. M. Sitinjak, R. O. Batubara, and F. Ikorasaki, "Perancangan dan Implementasi Algoritma Blowfish Untuk Keamanan Data File Citra Digital," *J. Widya*, vol. 5, no. 1, pp. 468–481, 2024.

[17] Y. Yusmai, T. Tommy, and Rosyidah siregar, "Aplikasi Enkripsi Data Video Menggunakan Metode RSA Dan Blowfish Berbasis Web," *J. Komput. Teknol. Inf. dan Sist. Inf.*, vol. 2, no. 3, pp. 535–544, 2024, doi: 10.62712/juktisi.v2i3.143.

[18] F. Gamaliel and P. Y. D. Arliyanto, "Implementasi Algoritma Blowfish Untuk Pengamanan File PDF," *JIRE (Jurnal Inform. Rekayasa Elektron.*, vol. 7, no. 1, pp. 60–67, 2024.

[19] A. Rahayu, A. Putri Ardana, C. Pramudhita, D. Syafitri, and R. Zabitha Sirega, "Perbandingan Algoritma RSA dengan Algoritma Blowfish Pada Perancangan Aplikasi Keamanan Data," *J. Ilmu Komput. dan Sist. Inf.*, vol. 7, no. 1, pp. 203–207, 2024, doi: 10.55338/jikomsi.v7i1.2875.

[20] E. Setyawati, C. E. Widjayanti, R. R. Siraiz, and H. Wijoyo, "Pengujian keamanan komputer kriptografi pada surat elektronik berbasis website dengan enkripsi metode MD5," *J. Manajamen Inform. Jayakarta*, vol. 1, no. 1, p. 56, 2021, doi: 10.52362/jmijayakarta.v1i1.367.

DOI: https://doi.org/10.38204/tematik.v12i1.2316