Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2023106558
Vol. 10, No. 5, Oktober 2014, hlm. 945-954 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

IMPLEMENTASI ALGORITMA *AES 256 CBC, BASE 64,* DAN *SHA 256* DALAM PENGAMANAN DAN VALIDASI DATA UJIAN ONLINE

Ferzha Putra Utama[*1], Gusman Wijaya[2], Ruvita Faurina[3], Arie Vatresia[4]

[1,2,3,4]Universitas Bengkulu, Bengkulu
Email: [1]fputama@unib.ac.id, [2]gusman.w.jaya@gmail.com, [3]ruvita.faurina@unib.ac.id, [3]arie.vatresi@unib.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 22 September 2022, diterima untuk diterbitkan: 26 September 2023)

Abstrak

Terdapat berbagai macam cara untuk melaksanakan ujian di tingkat perguruan tinggi, selama masa pandemi Covid-19 metode ujian *online* menjadi banyak digunakan. Meskipun ujian *online* dapat dilaksanakan di mana saja dan kapan saja, sayangnya masih banyak terjadi kecurangan seperti bocornya soal ujian, tersebarnya kunci jawaban secara ilegal, dan pengubahan pada data hasil ujian. Salah satu solusi dalam menjaga integritas hasil ujian berbasis *online* adalah mengenkripsi data ujian dengan metode kriptografi. Penelitian ini mengusulkan menerapkan beberapa metode kriptografi sebagai upaya dalam mengamankan dan memastikan keaslian data ujian *online* menggunakan algoritma AES 256 CBC, Base 64, dan SHA 256. Penelitian ini menghasilkan aplikasi ujian *online* berbasis website yang dibangun menggunakan teknologi *MERN Stack*. Hasil pengujian dalam memvalidasi data ujian *online* yang telah dienkripsi menggunakan sistem dan *OpenSSL* menunjukkan nilai *hash* yang sama. Hal ini menunjukkan sistem telah mampu mengenkripsi, mendekripsi, dan memvalidasi data ujian *online* dengan efektif.

Kata kunci: *AES 256 CBC, Base* 64, SHA 256, ujian online

IMPLEMENTATION OF AES 256 CBC, BASE 64, AND SHA 256 ALGORITHM IN ONLINE EXAM DATA SECURITY AND VALIDATION

Abstract

There are various ways to organize exams at the higher education level. During the Covid-19 pandemic, the online examination method has become widely used. Although online exams can be held anywhere and anytime, unfortunately, many violations and fraud exist, such as leaking exam questions, spreading answer keys illegally, and changing exam result data. One solution for maintaining the integrity of online-based exam results is to encrypt exam data with cryptographic methods. This study proposes applying several cryptographic methods to secure and ensure the authenticity of online exam data using AES 256 CBC, Base 64, and SHA 256 algorithms. This research resulted in a website-based online exam application built using MERN Stack technology. The test results in validating online exam data that has been encrypted using the system and OpenSSL show the same hash value. This shows that the system has been able to encrypt, decrypt, and validate online exam data effectively.

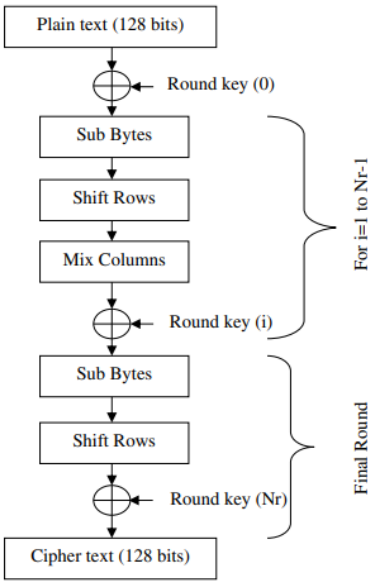
Keywords: *AES 256 CBC, Base 64, SHA 256, online exam*

1. PENDAHULUAN

Selama masa pandemi Covid-19 pelaksanaan ujian secara *online* masih menjadi pilihan bagi pendidik di perguruan tinggi. Pelaksanaan ujian secara *online* memang memiliki banyak manfaat, namun tetap saja masih terdapat beberapa isu (Junus *et al.*, 2021; Rahayu, Dwirifqi and Putra, 2022). Meski dilaksanakan dengan metode *online*, mutu pendidikan perlu dijaga agar pengukuran kemampuan siswa melalui ujian tetap terjamin. Upaya menjaga mutu ujian dapat dilakukan dengan menjaga pelaksanaan ujian dari praktik kecurangan (Dendir and Maxwell, 2020; Cádiz, Balderas and Caballero-hernández, 2021; Noorbehbahani and Mohammadi, 2022) seperti pengubahan hasil jawaban ujian yang telah di-*submit*. Upaya menjaga integritas pada ujian berbasis *online* dilakukan menggunakan metode kriptografi (Manoharan, 2021; Rayan *et al.*, 2021; Zhu and Cao, 2021). Pengamanan pada sistem ujian berbasis *online* dapat membantu lembaga pendidikan, khususnya perguruan tinggi dalam menjaga kualitas evaluasi pembelajaran.

Deformasi perilaku pada aktivitas pendidikan telah terjadi secara masif akibat pandemi Covid-19 (Cahyadi, 2020; Rachmawati *et al.*, 2021). Salah satu hal yang penting dilakukan adalah peningkatan aspek keamanan sistem, termasuk pada ujian berbasis *online*.

Penelitian ini mengusulkan penggabungan dua buah metode kriptografi sebagai upaya dalam mengamankan data seperti soal ujian, kunci jawaban ujian dan hasil ujian berbasis *online* dengan algoritma *AES 256 CBC* dan *Base64*. Kombinasi dua algoritma kriptografi (*hybrid*) dapat menghasilkan super enkripsi untuk meningkatkan keamanan data dikarenakan metode kriptografinya menjadi lebih kuat (Modugula, 2020; Wu and Li, 2022). Kedua metode ini merupakan algoritma standar enkripsi yang digunakan dalam berbagai sektor keamanan (Yudha and Laluma, 2019). Penelitian yang telah dilakukan (Krishna *et al.*, 2016; Park *et al.*, 2019; Fathurrahmad, 2020) telah menerapkan algoritma kriptografi AES, SHA dan Base64 untuk objek gambar, *password*, dan keamanan web. Sayangnya penerapan kombinasi algoritma kriptografi dalam upaya menjaga integritas hasil ujian belum dilakukan. Penelitian ini penting dilakukan untuk memperluas penerapan algoritma kriptografi pada dunia akademik di perguruan tinggi. Algoritma enkripsi *AES* memiliki proses yang terdiri dari 4 jenis *byte* transformasi, yaitu *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* (Kumar and Karthigaikumar, 2017; Mohammad and Abdullah, 2022). Pada langkah pertama proses enkripsi, *input* yang berupa *plaintext* akan mempertahankan transformasi *byte AddRoundKey*. Tahap selanjutnya, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* berulang sebanyak *Nr*. Pada putaran terakhir, *state* tidak lagi mengalami transformasi *MixColumns*. Blok diagram dari *AES* dapat dilihat pada Gambar 1.

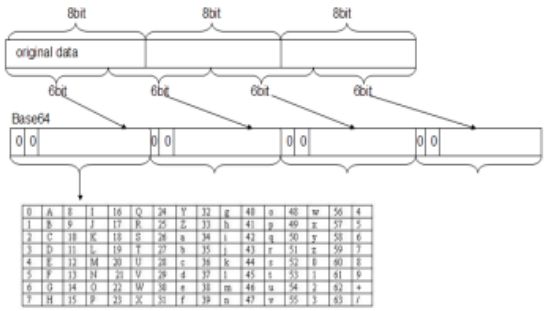


Gambar 1. Blok diagram *AES* (Venkataraman and Kumar, 2020)

Selain algoritma *AES*, terdapat juga algoritma kriptografi *Base64* yang merupakan salah satu algoritma untuk *encoding* dan *decoding* data biner yang berbentuk *ASCII*, yang dasarnya adalah bilangan 64 atau dapat dikatakan sebagai salah satu metode penyandian terhadap data biner (Azlin, Musadat and Nur, 2018). Keamanan algoritma *Base64* terdapat pada *index*-nya, di mana *plaintext* yang akan dilakukan enkripsi akan diproses dan dikonversi ke dalam tabel *index Base64*. Menurut analisis berdasarkan penelitian (Apreja, Syarif and Ibrahim, 2017), tingkat keamanan algoritma *Base64* sudah cukup tinggi karena melewati beberapa proses pemecahan *bit* huruf atau angka dari pesan asli.

2. METODE PEBELITIAN

Prinsip enkripsi *Base64* adalah 3 *byte* 8 *bit* (3 x 8 = 24 *bit*) diubah menjadi 4 *byte* 6 *bit* (4 x 6 = 24 *bit*). Setelah itu, di setiap awal *byte* dari 6 *bit* ditambahkan dua nol, dibentuk untuk setiap *byte* dari 8 *bit*. Setiap nilai *byte* diganti dengan karakter tabel *Base64*. Karakter yang terdapat pada tabel ini terdiri dari karakter, angka, dan simbol umum. Bahkan jika nilai *byte* data asli melebihi nilai batas, kembali ke kisaran tabel *ASCII* karena *split* dan *recombine*. Blok diagram dari *base64* dapat dilihat pada Gambar 2.



Gambar 2. Blok Diagram *Base 64* (Jianli, Yongdao and Xia, 2013)

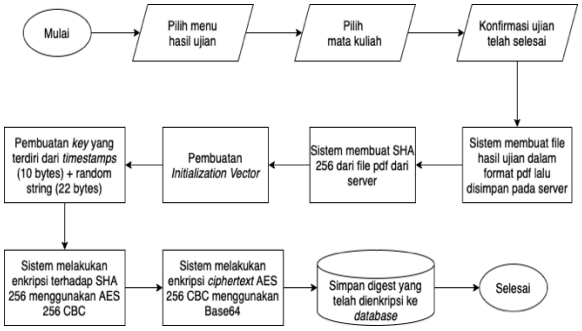
Algoritma *SHA 256* merupakan algoritma kriptografi satu arah, algoritma ini memproses blok data *input* dalam dua tahap: *preprocessing* dan *digest computing*. *Preprocessing* menjamin bahwa pesan memiliki ukuran kelipatan nilai tertentu, memungkinkan untuk membagi pesan menjadi ukuran blok yang telah ditentukan dan memberikan nilai *hash* awal. Pada tahap kedua, setiap blok pesan digunakan selama jumlah iterasi yang tetap, di mana setiap iterasi mendefinisikan fungsi, konstanta, dan operasi kata untuk menghasilkan serangkaian nilai *hash*. Setelah semua blok selesai diproses, nilai *hash* akhir digunakan sebagai intisari pesan. Secara khusus, tahap kedua dari algoritma *SHA 256* melakukan 64 iterasi pada blok pesan 512 *bit* dan nilai *hash* 256 *bit* dijelaskan sebagai 8 kata 32 *bit* (A, B, …, H). Panjang pesan *hash* nya adalah 256 *bit*.

Pada penelitian (Kaffah *et al.*, 2020) menghasilkan enkripsi pesan teks dengan algoritma *AES* dan *Huffman* dengan jumlah 32.200 karakter dengan nilai akurasi 90,62% dan selisih waktu kinerja enkripsi dan dekripsi adalah 1% dengan proses dekripsi sedikit lebih lama dari proses enkripsi. Kemudian pada penelitian (Wen and Dang, 2018) menyatakan bahwa algoritma *base64* bukanlah algoritma yang mengenkripsi data yang lengkap, karena proses konversi tidak memerlukan kunci, hanya untuk menyembunyikan data dengan cara mengenkripsinya. Karakter yang sama akan dikodekan secara berbeda tergantung pada posisinya. Enkripsi multimedia dengan AES dan SHA 256 yang dilakukan (Fauziah *et al.*, 2018) untuk mengamankan konten dari data. Proses enkripsi dan dekripsi berhasil dilakukan meskipun mengubah sedikit ukuran data asli, namun isi dan makna data tetap sama.

Penelitian ini melakukan enkrispsi data pada ujian *online* di perguruan tinggi yang terdiri dari soal ujian, pilihan jawaban, dan kunci jawaban. Saat dosen menambahkan soal, pilihan jawaban, dan kunci jawaban, sistem akan membuat *Initialization Vector (IV)* dari *random string* dengan panjang 16 *bytes*, lalu sistem juga akan membuat *Key* dari *timestamp* atau waktu saat ini ditambah dengan *random string* dengan panjang 22 *bytes*. Setelah itu, sistem akan melakukan enkripsi dengan *AES 256 CBC* pada data soal, pilihan jawaban, dan kunci jawaban menggunakan *IV* dan *Key* yang telah dibuat sebelumnya, selanjutnya hasil enkripsi dari *AES 256 CBC* ini akan dilakukan enkripsi lagi dengan *Base64*. Hasil enkripsi tersebut kemudian disimpan ke dalam *database*. *Flowchart* dari proses enkripsi algoritma *AES 256 CBC* dan *Base64* dapat dilihat pada Gambar 3.

Untuk mengetahui ada atau tidaknya perubahan data pada *file* hasil ujian digunakan algoritma *SHA 256*. Proses pengecekan dari algoritma *SHA 256* yaitu ketika dosen memilih menu hasil ujian, kemudian memilih mata kuliah, lalu mengkonfirmasi bahwa ujian telah selesai, kemudian sistem akan membuat *file pdf* hasil ujian dan menyimpannya di *server*. Selanjutnya sistem akan membuat *IV* yang terdiri dari *random string* dengan panjang 16 *bytes*, lalu sistem membuat *key* yang terdiri dari *timestamps* dengan panjang 10 *bytes* ditambah dengan *random string* dengan panjang 22 *bytes.*

Kemudian sistem akan melakukan enkripsi terhadap SHA 256 dari *file* yang ada di server menggunakan *AES 256 CBC*, lalu sistem akan mengenkripsi kembali *ciphertext AES 256 CBC* menggunakan *Base 64*, dan terakhir sistem akan menyimpan *digest* yang sudah dienkripsi ke dalam *database* sistem. *Flowchart* dari proses algoritma *SHA 256* dapat dilihat pada Gambar 4.



Gambar 4. *Flowchart SHA 256*

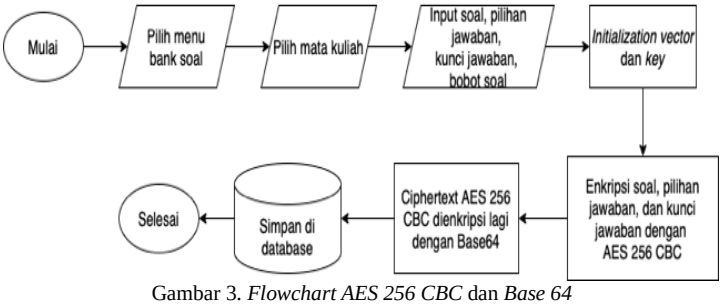
Jenis data yang akan digunakan pada penelitian ini adalah data *dummy*, data yang digunakan ditunjukkan pada Tabel 1.

Tabel 1. Data Penelitian

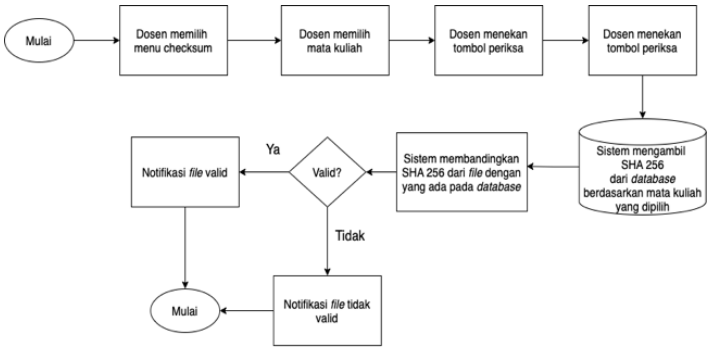
Data	Jumlah
Dosen	17
Mata Kuliah	63
Mahasiswa	94
Soal Ujian	50, yang akan diuji coba 10 soal yang diambil secara *random*
Jawaban Soal Ujian	50, menyesuaikan dengan soal ujian
Hasil Ujian	94, menyesuaikan dengan jumlah mahasiswa

Skema *dummy* yang ditunjukkan Tabel 1 merupakan suatu keadaan pada suatu jurusan di perguruan tinggi. Jumlah dosen yang terdapat pada jurusan tersebut sebanyak 17 orang, dengan 94 orang mahasiswa, dan 63 mata kuliah. Jumlah soal yang disediakan pada bank soal terdiri dari 50 soal lengkap dengan kunci jawabannya. Namun pada pelaksanaannya bisa saja tidak semua soal diujikan. Pada penelitian ini akan dilakukan pengujian *manual* terhadap algoritma *AES 256 CBC* dan *Base 64* menggunakan permisalan studi kasus dan data *dummy* yang akan penulis siapkan yaitu soal ujian. Soal ujian akan dienkripsi menggunakan *AES 256 CBC* dengan *IV* dan *Key* yang telah disiapkan, kemudian nilai *hex* dari enkripsi *AES 256 CBC* akan dienkripsi lagi menggunakan *Base64*.

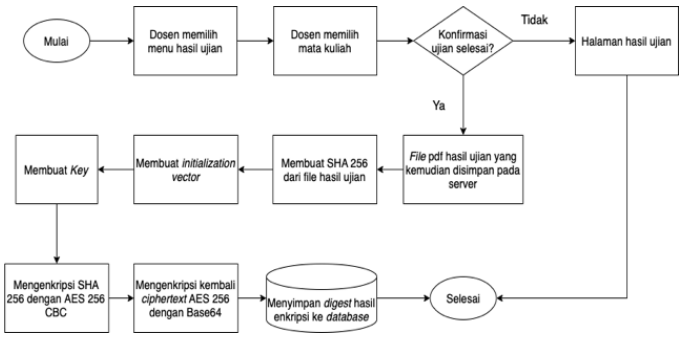
Sedangkan pengujian terhadap algoritma *SHA 256* akan dilakukan menggunakan *OpenSSL* dengan data *dummy* yang akan penulis siapkan yaitu *file* hasil ujian. *File* hasil ujian akan dilakukan *hash* menggunakan *SHA 256* untuk mendapatkan *message digest*, kemudian penulis akan melakukan manipulasi atau perubahan pada *file* hasil ujian lalu melakukan *hash* kembali menggunakan *SHA 256*, dari kedua *hash* tersebut akan dibandingkan apakah *hash* yang dihasilkan sama atau tidak, jika hasil *hash* tidak sama artinya data tidak asli lagi atau telah terjadi perubahan terhadap data.



Gambar 3. *Flowchart AES 256 CBC* dan *Base 64*



Gambar 5. Proses Pemeriksaan Keaslian Data Hasil Ujian



Gambar 6. Proses Mendapatkan *Digest SHA 256*

3. HASIL DAN PEMBAHASAN

3.1. Enkripsi Ujian

Penelitian ini menggunakan 10 soal ujian diambil secara acak dari 50 soal dari bank soal. Berdasarkan percobaan, didapatkan 94 data hasil ujian berdasarkan jumlah mahasiswa. Langkah selanjutnya yang dilakukan adalah dengan menguji keaslian data ujian melalui menu *checksum*.

Proses memeriksa keaslian hasil ujian dimulai dari ketika dosen memilih menu *checksum* dan memilih mata kuliah, kemudian dosen menekan tombol periksa, maka sistem akan mengambil *digest SHA 256* dari *file* hasil ujian yang disimpan di *server*, kemudian sistem akan mengambil *digest SHA 256* dari *database* berdasarkan mata kuliah yang dipilih, kemudian sistem akan membandingkan antara *SHA 256* dari *file* yang disimpan di *server* dengan *SHA 256* yang ada di *database*, apabila *digest SHA 256* tersebut sama, maka sistem akan menampilkan notifikasi bahwa hasil ujian terdaftar pada sistem. Namun sebaliknya, apabila *digest SHA 256* tersebut tidak sama, maka sistem akan menampilkan notifikasi bahwa hasil ujian tidak terdaftar pada sistem dan nampaknya hasil ujian tidak asli lagi.

Pada proses pemeriksaan keaslian *file* hasil ujian, yang diperiksa adalah waktu *file* dibuat dan isi atau *content* dari *file*, sistem mentoleransi *file name*, *access time*, *metadata*, *location*, dan lain sebagainya. Proses memeriksa keaslian hasil ujian dapat dilihat pada Gambar 5.

Proses mendapatkan *digest SHA 256* dimulai dari ketika dosen memilih menu hasil ujian dan mata kuliah ujian, sistem akan menampilkan notifikasi untuk mengkonfirmasi bahwa ujian telah selesai, apabila dosen tidak mengkonfirmasi, maka sistem akan menampilkan peringatan untuk segera konfirmasi bahwa ujian telah selesai, agar sistem dapat mengunci dan memeriksa keaslian hasil ujian. Namun apabila dosen melakukan konfirmasi, maka sistem akan membuat *file pdf* hasil ujian dan disimpan di *server* pada *directory public/docs/*, kemudian sistem akan membuat *SHA 256* dari *file pdf* hasil ujian di *server*, lalu sistem akan membuat *IV* yang terdiri dari *random string* (16 *bytes*),

kemudian membuat *key* yang terdiri dari *timestamps* (10 *bytes*) ditambah *random string* (22 *bytes*). Setelah itu sistem akan melakukan enkripsi terhadap *SHA 256* menggunakan *AES 256 CBC*, kemudian sistem akan melakukan enkripsi kembali terhadap *ciphertext AES 256 CBC* menggunakan *Base 64*, lalu menyimpan *digest* yang sudah dienkripsi ke dalam *database* sistem. Proses mendapatkan *digest SHA 256* dapat dilihat pada Gambar 6.

3.2. Pengujian Sistem

1. *AES 256 CBC* dan *Base 64*

Contoh sederhana dalam pengenkripsian menggunakan algoritma *AES 256 CBC* dan *Base 64* dengan menggunakan permisalan studi kasus. Contoh kasus yang akan dicari adalah sebagai berikut:

Dosen ingin mengenkripsi soal yang berisi kalimat "LANadalah", soal tersebut akan dilakukan enkripsi dengan *IV* yaitu "abcdefghijklmnop" dan *Key* yaitu "abcdefghijklmnopqrstuvwxyz123456", hasil enkripsi tersebut akan dilakukan enkripsi kembali menggunakan algoritma *Base 64.* Proses pengenkripsiannya adalah sebagai berikut:

Plaintext: LANadalah
Initialization Vector (IV): abcdefghijklmnop
Key: abcdefghijklmnopqrstuvwxyz123456

a) Konversi *char* ke *hex*

Tabel 2. Konversi *Char* ke *Hex*

Char	***Hex***
LANadalah	4c 41 4e 61 64 61 6c 61 68
abcdefghijklmnop	61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70
abcdefghijklmnopqrstuvwxyz123456	61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a 31 32 33 34 35 36

b) *Round Key (Key Schedule)*

Tabel 3. *Round Key* (*Key Schedule*)

Round	***Key***
1	71727374 75767778 797a3132 33343536
2	78f466a7 1d9201cf 74f86aa3 199605d3
3	a5e21812 d0946f6a a9ee5e58 9ada6b6e
4	2d8bf91f 3019f8d0 44e19273 5d7797a0
5	e91790f2 3983ff98 906da1c0 0ab7caae
6	80ff1d78 b0e6e5a8 f40777db a970e07b
7	3a4671d3 03c58e4b 93a82f8b 991fe525
8	48262296 f8c0c73e 0cc7b0e5 a5b7509e
9	3cef22d8 3f2aac93 ac828318 359d663d
10	06150500 fed5c23e f21272db 57a52245
11	67e9b1b6 58c31d25 f4419e3d c1dcf800
12	a0546678 5e81a446 ac93d69d fb36f4d8
13	68ec0ed7 302f13f2 c46e8dcf 05b275cf
14	d7c9ec13 89484855 25db9ec8 deed6a10

c) *Round 1*

Tabel 4. *Round 1*

Input	4c414e61 64616c61 68070707 07070707
SubBytes	29832fef 43ef50ef 45c5c5c5 c5c5c5c5
ShiftRows	29efc5c5 43c5c5ef 45c52fef c58350c5
MixColumns	787d03c0 f8693d00 1e4af4e0 9aedb216
AddRoundKey (RK-1)	090f70b4 8d1f4a78 6730c5d2 a9d98720

d) *Round 2*

Tabel 5. *Round 2*

Input	090f70b4 8d1f4a78 6730c5d2 a9d98720
SubBytes	0176518d 5dc0d6bc 8504a6b5 d33517b7
ShiftRows	01c0a6b7 5d04178d 853551bc d376d6b5
MixColumns	48dc5410 2ce1fbf5 a3a0cd93 44ebd6bf
AddRoundKey (RK-2)	302832b7 3173fa3a d758a730 5d7dd36c

e) *Round 3*

Tabel 6. *Round 3*

Input	302832b7 3173fa3a d758a730 5d7dd36c
SubBytes	043423a9 c78f2d80 0e6a5c04 4cff6650
ShiftRows	048f5c50 c76a66a9 0eff2380 4c342d04
MixColumns	8eb5c37f e4108117 a50e2cd5 ed572ec5
AddRoundKey (RK-3)	2b57db6d 3484ee7d 0ce0728d 778d45ab

f) *Round 4*

Tabel 7. *Round 4*

Input	2b57db6d 3484ee7d 0ce0728d 778d45ab
SubBytes	f15bb93c 185f28ff fee1405d f55d6e62
ShiftRows	f15f4062 18e16e3c fe5db9ff f55b285d
MixColumns	3aed88d3 5a4f61df 466bd018 696619cd
AddRoundKey (RK-4)	176671cc 6a56990f 028a426b 34118e6d

g) *Round 5*

Tabel 8. *Round 5*

Input	176671cc 6a56990f 028a426b 34118e6d
SubBytes	f033a34b 02b1ee76 777e2c7f 1882193c
ShiftRows	f0b12c3c 027e194b 7782a376 1833ee7f
MixColumns	23c15dee d49e93f7 a6e03254 f4286d0b
AddRoundKey (RK-5)	cad6cd1c ed1d6c6f 368d9394 fe9fa7a5

h) *Round 6*

Tabel 9. *Round 6*

Input	cad6cd1c ed1d6c6f 368d9394 fe9fa7a5
SubBytes	74f6bd9c 55a450a8 055ddc22 bbdb5c06
ShiftRows	74a4dc06 555d5c9c 05dbbda8 bbf65022
MixColumns	c55e79e8 8d970fdd 69dc5c22 1e9e8b34
AddRoundKey (RK-6)	45a16490 3d71ea75 9ddb2bf9 b7ee6b4f

i) *Round 7*

Tabel 10. *Round 7*

Input	45a16490 3d71ea75 9ddb2bf9 b7ee6b4f
SubBytes	6e324360 27a3879d 5eb9f199 a9287f84
ShiftRows	6ea3f184 27b97f60 5e28439d a9328799
MixColumns	57bfa3f3 81afc06f 1a564ca8 01c63e7c
AddRoundKey (RK-7)	6df9d220 826a4e24 89fe6323 98d9db59

j) *Round 8*

Tabel 11. *Round 8*

Input	6df9d220 826a4e24 89fe6323 98d9db59
SubBytes	3c99b5b7 13022f36 a7bbfb26 4635b9cb
ShiftRows	3c02fbcb 13bbb9b7 a735b536 46992f26
MixColumns	4ee59530 fe190342 893fb91e 3538eb30
AddRoundKey (RK-8)	06c3b7a6 06d9c47c 85f809fb 908fbbae

k) *Round 9*

Tabel 12. *Round 9*

Input	06c3b7a6 06d9c47c 85f809fb 908fbbae
SubBytes	6f2ea924 6f351c10 9741010f 6073eae4
ShiftRows	6f3501e4 6f41ea24 9773a910 602e1c0f
MixColumns	64e26f56 d3ec8d52 19819d58 a117678c
AddRoundKey (RK-9)	580d4d8e ecc621c1 b5031e40 948a01b1

l) *Round 10*

Tabel 13. *Round 10*

Input	580d4d8e ecc621c1 b5031e40 948a01b1
SubBytes	6ad7e319 ceb4fd78 d57b7209 227e7cc8
ShiftRows	6ab472c8 ce7b7c19 d57ee378 22d7fd09
MixColumns	a94779f3 6fa5667c a86ffe09 d2820f5e
AddRoundKey (RK-10)	af527cf3 9170a442 5a7d8cd2 85272d1b

m) *Round 11*

Tabel 14. *Round 11*

Input	af527cf3 9170a442 5a7d8cd2 85272d1b
SubBytes	7900100d 8151492c beff64b5 97ccd8af
ShiftRows	795164af 81ffd80d becc102c 970049b5
MixColumns	cad80afb d61ac2a5 1421265d c9f9c19a
AddRoundKey (RK-11)	ad31bb4d 8ed9df80 e060b860 0825399a

n) *Round 12*

Tabel 15. *Round 12*

Input	ad31bb4d 8ed9df80 e060b860 0825399a
SubBytes	95c7eae3 19359ecd e1d06cd0 303f12b8
ShiftRows	95356cb8 19d012e3 e13feacd 30c79ed0
MixColumns	baf3ab96 a877d334 bf775d6c 7cccbbb2
AddRoundKey (RK-12)	1aa7cdee f6f67772 13e48bf1 87fa4f6a

o) *Round 13*

Tabel 16. *Round 13*

Input	1aa7cdee f6f67772 13e48bf1 87fa4f6a
SubBytes	a25cbd28 4242f540 7d693da1 172d8402
ShiftRows	a2423d02 42698428 7d2dbd40 175cf5a1
MixColumns	a6639c86 932f407b 70bbf197 9e0a42c9
AddRoundKey (RK-13)	ce8f9251 a3005389 b4d57c58 9bb83706

p) *Round 14*

Tabel 17. *Round 14*

Input	ce8f9251 a3005389 b4d57c58 9bb83706
SubBytes	8b734fd1 0a63eda7 8d03106a 146c9a6f
ShiftRows	8b63106f 0a039ad1 8d6c4fa7 1473ed6a
AddRoundKey (RK-14)	5caafc7c 834bd284 a8b7d16f ca9e877a

Sehingga *ciphertext* yang dihasilkan dari proses enkripsi dengan menggunakan *AES 256 CBC* adalah 5caafc7c 834bd284 a8b7d16f ca9e877a. *Ciphertext* tersebut dienkripsi lagi menggunakan *Base 64*, proses pengenkripsian adalah sebagai berikut:

Plaintext: 5caafc7c834bd284a8b7d16fca9e877a

Ubah *plaintext* tersebut ke dalam bentuk *biner* dan bagi menjadi *6 bit* untuk setiap blok-bloknya.

Biner:

00110101 01100011 01100001 01100001 01100110 01100011 00110111 01100011 00111000 00110011 00110100 01100010 01100100 00110010 00111000 00110100 01100001 00111000 01100010 00110111 01100100 00110001 00110110 01100110 01100011 01100001 00111001 01100101 00111000 00110111 00110111 01100001

Biner (6 bit):

001101 010110 001101 100001 011000 010110 011001 100011 001101 110110 001100 111000 001100 110011 010001 100010 011001 000011 001000 111000 001101 000110 000100 111000 011000 100011 011101 100100 001100 010011 011001 100110 011000 110110 000100 111001 011001 010011 100000 110111 001101 110110 0001

Tambahkan *2 bit 0* disetiap awal blok:

00001101 00010110 00001101 00100001 00011000 00010110 00011001 00100011 00001101 00110110 00001100 00111000 00001100 00110011 00010001

00100010 00011001 00000011 00001000 00111000 00001101 00000110 00000100 00111000 00011000 00100011 00011101 00100100 00001100 00010011 00011001 00100110 00011000 00110110 00000100 00111001 00011001 00010011 00100000 00110111 00001101 00110110 00000100

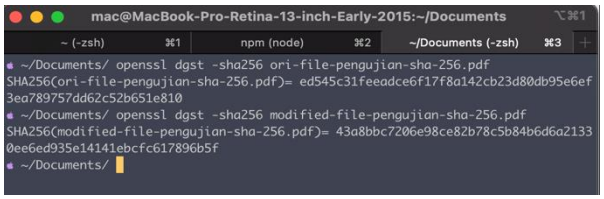
Konversikan *biner* tersebut menggunakan tabel *Base 64*, sehingga *ciphertext* menjadi sebagai berikut:
NWNhYWZjN2M4MzRiZDI4NGE4YjdkMTZmY2E5ZTg3N2E=

Ciphertext hasil perhitungan manual dengan *ciphertext* yang dihasilkan oleh sistem menunjukkan hasil yang sama, seperti pada Gambar 7.

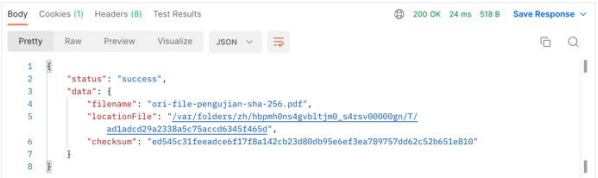
Gambar 7. *Ciphertext* yang Dihasilkan Sistem

2. *SHA 256*

Pada pengujian algoritma *SHA 256* ini menggunakan *OpenSSL* (Lim, 2019; Decker, Winters and Mercer, 2021). Penulis telah menyiapkan data *dummy* yaitu contoh *file* hasil ujian yang asli atau belum pernah dilakukan perubahan dengan nama *ori-file-pengujian-sha-256.pdf* dan *file* hasil ujian yang sudah dilakukan modifikasi atau perubahan dengan mengubah nilai hasil ujian mahasiswa semula 80 menjadi 100 diberi nama *modified-file-pengujian-sha-256.pdf*.



Gambar 8. Pengujian menggunakan *OpenSSL*



Gambar 9. *Digest ori-file-pengujian-sha-256.pdf* yang Dihasilkan Sistem

Gambar 10. *Digest modified-file-pengujian-sha-256.pdf* yang Dihasilkan Sistem

Setelah dilakukan perubahan konten *file*, *digest* yang dihasilkan sangat berbeda, dengan perbedaan yang dihasilkan tersebut, dapat dipastikan bahwa *file* hasil ujian telah mengalami perubahan atau tidak asli lagi. *Digest* yang dihasilkan oleh *OpenSSL* dan *digest* yang dihasilkan oleh sistem menunjukkan hasil yang sama, dapat dilihat pada Gambar 8 – Gambar 10.

3. Pengujian Efektifitas Algoritma

Pengujian efektifitas dari algoritma AES 256 CBC dan Base 64 dalam mengamankan data seperti soal ujian, kunci jawaban ujian, dan checksum dari file hasil ujian. Dalam pengujian kali ini akan digunakan Initialization Vector (IV) yaitu "abcdefghijklmnop" dan *Key* "abcdefghijklmnopqrstuvwxyz123456". Pengujian efektifitas algoritma tersebut dapat dilihat pada Tabel 18.

Tabel 18. Uji Efektifitas AES 256 CBC dan Base64

No	String	Panjang string (byte)	Waktu (ms)
1	Internet mulai dipergunakan secara terbuka untuk umum pada tahun	64	0,868
2	Yang dimaksud dengan istilah penggunaan jaringan tidak terpandu	63	1,34
3	Perangkat yang digunakan untuk jaringan Type PAN	48	0,862
4	OSI terdiri dari 7 lapisan layer yang urutannya dari lapisan layer paling bawah sampai pada lapisan layer paling atas adalah	124	0,753
5	Protokol yang bertanggung jawab untuk melakukan pemetaan (routing) dan enkapsukasi paket-paket data jaringan menjadi paket-paket IP adalah	140	1,175
6	Salah satu protocol yang bekerja pada lapisan Host-to-host Layer adalah	71	1,415
7	Proses pencampuran dua sinyal menjadi satu sinyal adalah	56	0,973
8	Lapisan OSI Layer yang paling bawah adalah	42	1,787
9	Satuan informasi terkecil yang dikenal dalam komunikasi data adalah	67	1,187
10	Berikut protocol yang bekerja pada lapisan Network Interface Layer, kecuali	75	0,755
	Rata-rata waktu (ms)		1,1115

Berdasarkan Tabel 18, algoritma AES 256 CBC dan Base 64 dapat melakukan enkripsi terhadap *string* dengan panjang yang bermacam dan

waktu *response* API yang diperoleh tergantung dari koneksi internet yang digunakan.

4. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan metode *AES 256 CBC, Base 64,* dan *SHA 256* untuk mengamankan data dan mengetahui keaslian *file* pada aplikasi ujian *online* berbasis *website*. Berdasarkan hasil pengujian untuk memvalidasi data ujian yang telah dienkripsi menggunakan sistem dan *OpenSSL* menunjukkan nilai *hash* yang sama. Hal ini menunjukkan sistem telah mampu mengenkripsi, mendekripsi, dan memvalidasi data ujian *online* dengan efektif terhadap 20 soal ujian dengan panjang *string* yang bermacam.

Pengamanan file pdf menggunakan metode kriptografi *AES 256 CBC, Base 64,* dan *SHA 256* dapat bekerja dengan baik. Penerapan metode ini pada hasil ujian berbasis online menghasilkan waktu enkripsi rata-rata 1,1115 ms. Hal ini tergantung dengan koneksi internet dan besar data yang dienkripsi.

Penelitian selanjutnya dapat melakukan penerpan atau kombinasi metode kriptografi lain yang lebih efektif, format file yang beragam.

DAFTAR PUSTAKA

APREJA, A., SYARIF, Z. and IBRAHIM, A., 2017. 'Analisis Tingkat Keamanan Enkripsi Data Menggunakan Algoritma Base 64 Endcode', *Computer Science and ICT*, 3(1), pp. 49–50.

AZLIN, MUSADAT, F. and NUR, J., 2018. 'Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma Base64', *Jurnal Informatika*, 7(2), pp. 1–5.

CÁDIZ, A. U. DE, BALDERAS, A. and CABALLERO-HERNÁNDEZ, J. A., 2021. 'Analysis of Learning Records to Detect Student Cheating on Online Exams : Case Study during COVID-19 Pandemic', in *TEEM: Technological Ecosystems for Enhancing Multiculturality*. Salamanca, Spain: Association for Computing Machinery, New York NY, United States, pp. 752–757. doi: 10.1145/3434780.3436662.

CAHYADI, A., 2020. 'Covid-19 Outbreak and New Normal Teaching in Higher Education : Empirical Resolve from Islamic Universities in Indonesia', *Dinamika Ilmu*, 20(2), pp. 255–266. doi: 10.21093/di.v20i2.2545.

DECKER, B., WINTERS, B. and MERCER, E., 2021. 'Towards verifying SHA256 in OpenSSL with the software analysis workbench', in *NASA Formal Methods: 13th International Symposium, NFM 2021, Virtual Event, May 24–28, 2021, Proceedings*. Springer, pp. 72–78.

DENDIR, S. and MAXWELL, R. S., 2020. 'Computers in Human Behavior Reports Cheating in online courses : Evidence from online proctoring', *Computers in Human Behavior Reports*, 2(October), p. 100033. doi: 10.1016/j.chbr.2020.100033.

FATHURRAHMAD, F., 2020. 'Development And Implementation Of The Rijndael Algorithm And Base-64 Advanced Encryption Standard (AES) For Website Data Security', *International Journal of Scientific & Technology Research*, 9(11), pp. 7–11.

FAUZIAH, N. A. *et al.*, 2018. 'Design and implementation of AES and SHA-256 cryptography for securing multimedia file over android chat application', *2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018*, pp. 146–151. doi: 10.1109/ISRITI.2018.8864485.

JIANLI, C., YONGDAO, S. and XIA, L., 2013. 'The Research of Mobile phone Entrance Guard System Model based on the Encryption Two-dimensional Code', *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 11(9), pp. 5284–5292. doi: 10.11591/telkomnika.v11i9.3281.

JUNUS, K. *et al.*, 2021. 'Lecturer Readiness for Online Classes during the Pandemic : A Survey Research', *Education Sciences*, 11, p. 139. doi: 10.3390/educsci11030139.

KAFFAH, F. M. *et al.*, 2020. 'E-Mail message encryption using advanced encryption standard (AES) and huffman compression engineering', *Proceedings - 2020 6th International Conference on Wireless and Telematics, ICWT 2020*. doi: 10.1109/ICWT50448.2020.9243651.

KRISHNA, V. *et al.*, 2016. 'A Novel Image Encryption Algorithm using AES and Visual Cryptography', in *2nd International Conference on Next Generation Computing Technologies (NGCT-2016)*.

KUMAR, T. M. and KARTHIGAIKUMAR, P., 2017. 'FPGA implementation of an optimized key expansion module of AES algorithm for secure transmission of personal ECG signals', *Design Automation for Embedded Systems*. doi: 10.1007/s10617-017-9189-5.

LIM, J. P., 2019 'Automatic Equivalence Checking for Assembly Implementations of Cryptography Libraries', *2019 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pp. 37–49.

MANOHARAN, S., 2021. 'On Individualized Online Assessments in STEM Subjects', in *2021 IEEE International Conference on Engineering, Technology & Education (TALE)*, pp. 255–260. doi: 10.1109/TALE52509.2021.9678631.

MODUGULA, R. S. R., 2020. 'A Hybrid approach for Augmenting password security using Argon2i hashing and AES Scheme.' Dublin, National College of Ireland.

MOHAMMAD, H. M. and ABDULLAH, A. A., 2022. 'Enhancement process of AES : a lightweight cryptography algorithm-AES for constrained devices', *TELKOMNIKA Telecommunication Computing Electronics and Control*, 20(3), pp. 551–560. doi: 10.12928/TELKOMNIKA.v20i3.23297.

NOORBEHBAHANI, F. and MOHAMMADI, A., 2022. *A systematic review of research on cheating in online exams from 2010 to 2021*, *Education and Information Technologies*. Springer US. doi: 10.1007/s10639-022-10927-7.

PARK, M. *et al.*, 2019. 'Decrypting password-based encrypted backup data for Huawei smartphones', *Digital Investigation*, 28, pp. 119–125. doi: 10.1016/j.diin.2019.01.008.

RACHMAWATI, R. *et al.*, 2021. 'Work from Home and the Use of ICT during the COVID-19 Pandemic in Indonesia and Its Impact on Cities in the Future', *Sustainability*, (13), pp. 1–17. doi: 10.3390/su13126760.

RAHAYU, W., Dwirifqi, M. and PUTRA, K., 2022. 'Development and validation of Online Classroom Learning Environment Inventory (OCLEI): The case of Indonesia during the COVID - 19 pandemic', *Learning Environments Research*, 25(1), pp. 97–113. doi: 10.1007/s10984-021-09352-3.

RAYAN, F. *et al.*, 2021. 'Results in Physics Analysis and challenges of robust E-exams performance under COVID-19', *Results in Physics*, 23, p. 103987. doi: 10.1016/j.rinp.2021.103987.

VENKATARAMAN, N. L. and KUMAR, R., 2020. 'An efficient NoC router design by using an enhanced AES with retiming and clock gating techniques', *Transactions on Emerging Telecommunications Technologies*, 31(12), pp. 1–14. doi: 10.1002/ett.3839.

WEN, S. and DANG, W., 2018. 'Research on Base64 Encoding Algorithm and PHP Implementation', *International Conference on Geoinformatics*, 2018-June(41661087). doi: 10.1109/GEOINFORMATICS.2018.8557068.

WU, K. and LI, C., 2022. 'Application of Symmetric Encryption Algorithm Sensor in the Research of College Student Security Management System', *Journal of Sensors*, 2022, pp. 1–7. doi: 10.1155/2022/3323547.

YUDHA, G. S. and LALUMA, R. H., 2019. 'Sistem Keamanan Jaringan Dalam Ujian Online Sma/Smk Menggunakan Metode Algoritma Advanced Encryption Standard (Aes)', *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, 4(2), p. 71. doi: 10.32897/infotronik.2019.4.2.261.

ZHU, X. and CAO, C., 2021. 'Secure Online Examination with Biometric Authentication and Blockchain-Based Framework', *Mathematical Problems in Engineering*, 2021(2). doi: 10.1155/2021/5058780.

Halaman ini sengaja dikosongkan