Penerapan Enkripsi dan Dekripsi Dokumen Data UMKM Menggunakan Algoritma ChaCha20-Poly1305

I Made Chandra Widjaya[a1], I Komang Ari Mogi[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]madecandra55@gmail.com
[2]arimogi@unud.ac.id

Abstract

Despite the increasing importance of data security in safeguarding sensitive information, this study addresses the potential risks associated with unauthorized access to critical data. Employing the ChaCha20-Poly1305 algorithm, the research focuses on implementing encryption and decryption processes for Small and Medium Enterprise (SME) documents, supplemented by key derivation from AES-256 for enhanced security. A nonce Initialization Vector (IV) is generated using the ChaCha20-Poly1305 algorithm, with users inputting secret keys for encryption. The system then encrypts the data using the ChaCha-Poly1305 algorithm and derives keys from AES using SHA-256 hashing. For decryption, users input the encrypted document into the program, along with the previously used key. The system design employs a simple web-based application, with the ChaCha20-Poly1305 cryptography algorithm implemented in PHP. The study successfully tests the ChaCha20-Poly1305 algorithm, and the program exhibits secure decryption processes, evidenced by consistent byte sizes of tested SME documents.

Keywords: *Cryptography, ChaCha20-Poly1305, File Security, Encryption, Decryption*

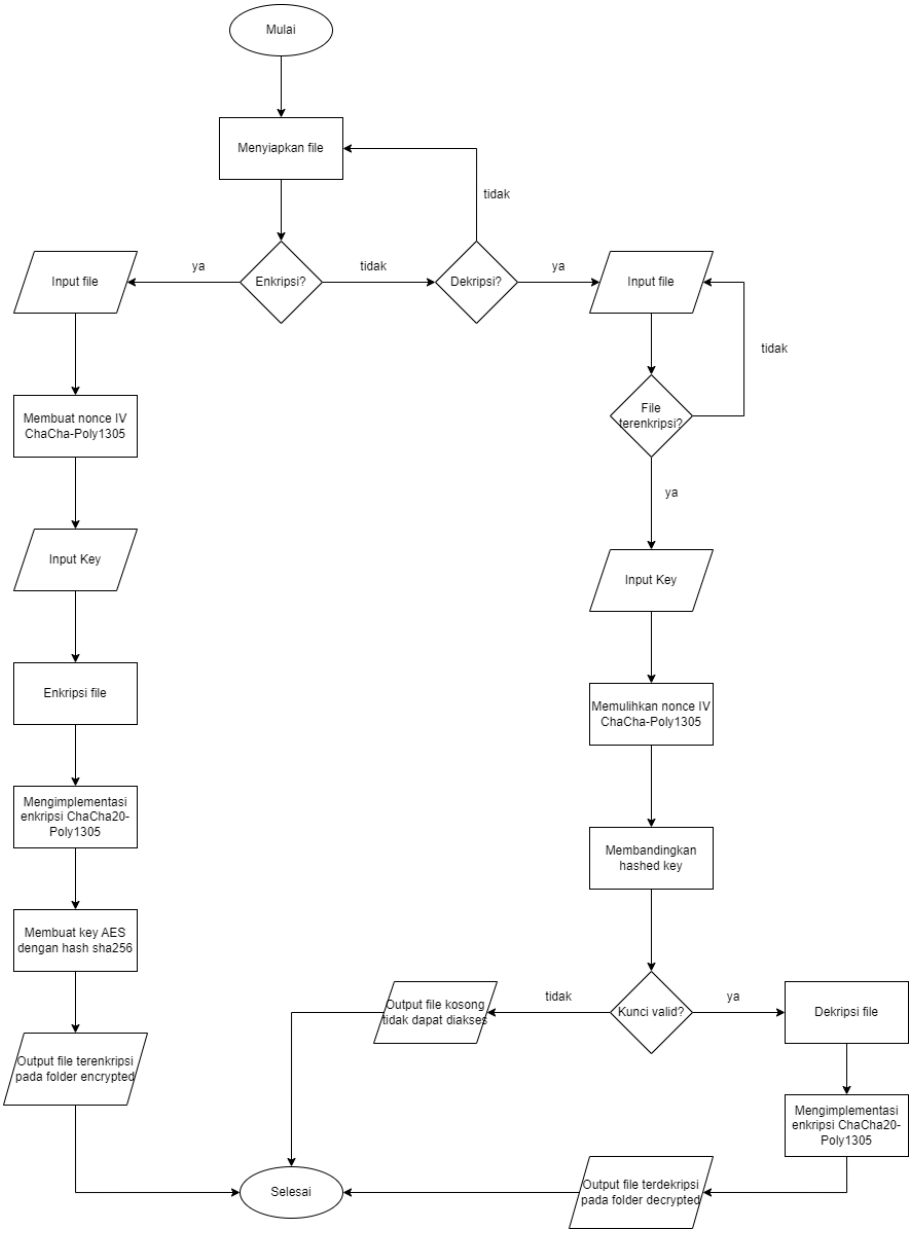
1. Pendahuluan

Dalam periode perkembangan teknologi yang semakin canggih, informasi yang tersebar dalam dunia maya menjadi semakin rentan. Meski demikian, keamanan data dianggap sebagai hal yang krusial untuk melindungi informasi sensitif, mengingat potensi dampak negatifnya jika data penting disadap oleh pihak yang tidak berwenang[1]. Salah satu cara untuk mengamankan informasi sensitif adalah melalui penggunaan teknik enkripsi dan dekripsi. Enkripsi menjadi salah satu proses mengubah teks menjadi format yang tidak dapat dibaca atau sulit dipahami. Sedangkan dekripsi menjadi proses mengembalikan teks tersebut ke dalam bentuk semula. Dengan menerapkan teknik ini, informasi yang dikirimkan melalui internet atau disimpan dalam database dapat dijaga keamanannya meskipun rentan terhadap serangan pihak yang tidak bertanggung jawab. Penerapan bentuk enkripsi dan dekripsi teks dapat dilakukan juga pada dokumen-dokumen penting. Dokumen-dokumen tersebut mungkin berisi informasi rahasia perusahaan, data keuangan sensitif, atau bahkan dokumen pribadi seperti surat-surat elektronik dan file-file identitas. Dengan menggunakan teknik enkripsi yang tepat, dokumen-dokumen tersebut dapat diamankan dari akses yang tidak sah, baik saat disimpan dalam sistem penyimpanan digital maupun saat dikirimkan melalui jaringan internet. Selain itu, penerapan dekripsi yang hanya dapat dilakukan oleh pihak yang berwenang juga dapat memberikan tingkat keamanan tambahan terhadap dokumen-dokumen penting tersebut[2]. Kriptografi menjadi sangat populer dengan berbagai macam algoritma yang ada. Beberapa diantaranya terdapat algoritma AES-256, RSA, Diffie Helman, ChaCha20, dan masih banyak lagi. Penelitian ini menggunakan ChaCha20-Poly1305 sebagai algoritma untuk menerapkan proses enkripsi dan dekripsi pada dokumen data UMKM. Tidak hanya itu, dalam penerapannya menggunakan key turunan dari AES-256. Maka dari itu, diharapkan penerapan kriptografi ini dapat menjaga dengan baik kepentingan setiap dokumen atau *file* data terkait [3].

p-ISSN: 2986-3929
e-ISSN: 3032-1948

2. Metode Penelitian

2.1 Desain Sistem



Gambar 1. Flowchart Sistem

Untuk alur cara kerja sistem, diawali dengan menyiapkan dokumen file penting Usaha Mikro Kecil Menengah (UMKM). Setelah itu, masuk ke dalam program untuk melakukan proses enkripsi dengan memasukkan dokumen. Program akan membuat *nonce Initialization Vector* (IV) dari algoritma ChaCha20-Poly1305. Selanjutnya, pengguna menginputkan kunci rahasia dan melakukan proses enkripsi. Pada tahap ini, sistem akan melakukan proses enkripsi dengan algoritma ChaCha-Poly1305 dan membuat kunci turunan dari algoritma *Advanced Encryption Standard* (AES) dengan *hash* SHA-256. Setelah proses enkripsi selesai, program otomatis mengunduh *environment* disesuaikan dengan ekstensi dokumen, lalu menyimpannya pada *folder encrypted*.

Jika pemilik dokumen ingin melakukan dekripsi, maka pemilik harus menggunakan dokumen yang telah dienkripsi dan masuk ke dalam program untuk di dekripsi. Di dalam program, diharuskan untuk memasukkan kunci yang telah dimasukkan sebelumnya untuk dokumen pada saat di enkripsi, lalu sistem akan memulihkan *nonce Initialization Vector* (IV) dari algoritma ChaCha20-Poly1305. Selanjutnya, program membandingkan kunci yang telah di *hash* sebelumnya. Jika kunci tidak *valid*, maka program hanya akan memberikan *output* berupa dokumen kosong atau tidak dapat diakses. Jika kunci *valid*, maka proses dekripsi dilanjutkan dan menyimpan dokumen terdekripsi yang sama seperti awalnya di dalam *folder decrypted*.

2.2 Pemrosesan Awal

Proses awal sistem dengan pembuatan nonce Initialization Vector (IV) menggunakan ChaCha20-Poly1305, program mengambil langkah-langkah yang cermat untuk memastikan keamanan. ChaCha20-Poly1305 digunakan sebagai konstruksi kriptografi yang menggabungkan algoritma *stream cipher* ChaCha20 dengan autentikasi pesan Poly1305 untuk melindungi data dokumen. Saat proses enkripsi dimulai, program secara otomatis menghasilkan *nonce* IV yang unik setiap kali dengan menggunakan fungsi acak yang kuat. *Nonce* IV ini membuat nilai acak yang hanya digunakan satu kali dan berperan penting dalam mencegah serangan repetisi yang mungkin terjadi. Selanjutnya, *nonce* IV ini digunakan bersama dengan kunci rahasia dan data yang akan dienkripsi, memastikan keamanan dan keunikan setiap proses enkripsi. Sementara itu, terkait dengan koneksi antara kunci AES256 dan hash SHA-256, sistem juga mengimplementasi langkah-langkah yang efisien dalam pengelolaan kunci enkripsi. Sebelum proses enkripsi dimulai, pengguna diminta untuk memasukkan kunci rahasia yang akan digunakan. Kunci ini kemudian diberikan fungsi *hash* SHA-256 untuk menghasilkan kunci turunan yang akan digunakan dalam proses enkripsi AES. *Hashing* SHA-256 mengubah kunci rahasia menjadi nilai *hash* yang panjang dan unik, memastikan keamanan kunci yang digunakan dalam proses enkripsi[4]. Hasil *hash* SHA-256 ini kemudian menjadi kunci untuk algoritma AES, yang akan digunakan dalam proses enkripsi data. Dengan demikian, program memastikan bahwa kunci enkripsi yang digunakan dalam proses AES unik dan aman.

2.3 Enkripsi ChaCha20-Poly1305

Proses enkripsi dengan ChaCha20-Poly1305 adalah sebuah algoritma kriptografi simetris yang terdiri dari dua komponen utama, yaitu ChaCha20 untuk enkripsi data dan Poly1305 untuk otentikasi pesan[5]. Pertama, ChaCha20 mengambil kunci enkripsi dan *nonce* (nomor sekali pakai) sebagai *input*, kemudian menghasilkan *keystream*. *Keystream* ini kemudian digunakan untuk melakukan enkripsi data dengan operasi XOR. Selanjutnya, Poly1305 digunakan untuk menghasilkan *tag* otentikasi pesan yang unik untuk setiap pesan yang dienkripsi. Proses ini melibatkan penggunaan polinomial pada *keystream* dan kunci enkripsi. *Tag* otentikasi ini memastikan bahwa pesan tidak diubah oleh pihak yang tidak sah selama proses pengiriman. Proses enkripsi dengan ChaCha20-Poly1305 terjamin keamanannya karena ChaCha20 adalah algoritma stream cipher yang aman, sementara Poly1305 adalah fungsi hash yang kuat untuk otentikasi pesan. Kombinasi dari kedua komponen ini memberikan keamanan yang kuat dan efisien untuk proses enkripsi data.

2.4 Dekripsi ChaCha20-Poly1305

Proses dekripsi dengan ChaCha20-Poly1305 mirip dengan proses enkripsinya, tetapi dilakukan secara terbalik. Pertama, *tag* otentikasi pesan diverifikasi menggunakan algoritma Poly1305 dengan menggunakan kunci enkripsi yang sama dan *nonce* yang sama yang digunakan pada saat enkripsi. Jika *tag* otentikasi tidak cocok, proses dekripsi dihentikan karena pesan mungkin telah dimanipulasi. Setelah verifikasi *tag* otentikasi berhasil, *keystream* dihasilkan menggunakan algoritma ChaCha20 dengan kunci enkripsi yang sama dan *nonce* yang sama seperti pada proses enkripsi. *Keystream* kemudian digunakan untuk melakukan operasi XOR pada pesan terenkripsi, menghasilkan pesan asli. Dalam proses dekripsi, sangat penting untuk menggunakan *nonce* yang sama seperti yang digunakan pada saat enkripsi dan memastikan bahwa kunci enkripsi tetap rahasia. Dengan menggunakan *nonce* yang sama, proses dekripsi dapat

menghasilkan *keystream* yang identik dengan yang digunakan pada saat enkripsi, sehingga memungkinkan untuk mendapatkan pesan asli dari pesan terenkripsi dengan benar[6].

3. Hasil dan Pembahasan

3.1 Rancangan Sistem

Rancangan sistem pada penelitian ini menggunakan aplikasi berbasis *website* sederhana. Bentuk algoritma kriptografi ChaCha20-Poly1305 dibuat dalam bahasa pemrograman PHP. Program juga menggunakan *webserver* Apache pada XAMPP untuk dijalankan. Perancangan sistem ini memiliki tujuan untuk menguji algoritma dapat digunakan pada proses enkripsi dan dekripsi dokumen data UMKM. Penelitian ini menggunakan contoh data Surat Keterangan Tempat Usaha (SKTU) sebagai data UMKM.

Gambar 2. Tampilan Sistem

3.2 Uji Coba Sistem

a. Proses Enkripsi Dokumen UMKM
Proses uji coba sistem ini dilakukan untuk mengetahui dapatkah dokumen terkait dienkripsi dengan menggunakan algoritma ChaCha20-Poly1305 merujuk pada tahapan-tahapan berikut:

- Menyiapkan dokumen UMKM dengan ekstensi .docx yang akan dienkripsi.
- Masukkan dokumen ke dalam program enkripsi.
- Program akan menggunakan algoritma ChaCha20-Poly1305 untuk membuat *nonce Initialization Vector* (IV).
- Memasukkan kunci rahasia untuk proses enkripsi.
- Proses enkripsi ChaCha20-Poly1305 akan dilakukan.
- Program akan membuat kunci turunan dari algoritma *Advanced Encryption Standard* (AES) dengan *hash* SHA-256.
- Setelah proses enkripsi selesai, program secara otomatis mengunduh *environment* yang disesuaikan dengan ekstensi dokumen.
- Dokumen yang telah dienkripsi disimpan dalam folder yang ditetapkan, yaitu *folder* "*encrypted*".

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Tabel 1. Proses Enkripsi

Dokumen UMKM Sebelum Enkripsi

PEMERINTAH KODTA DENPASAR
KECAMATAN DENPASAR SELATAN
DESA SANUR KAJA

Alamat : JL. By Pass Ngurah Rai No. 59 Telp. (0361) 287388 Kode Pos 80227
http://sanurkaja.denpasarkota.go.id sanurkaja@gmail.com

SURAT KETERANGAN TEMPAT USAHA

Nomor : 323/Pel.Um/IV/2021

Yang bertanda tangan dibawah ini, Perbekel Desa Sanur Kaja, Kecamatan Denpasar Selatan, Kota Denpasar, berdasarkan surat pengantar Kepala Dusun ***Belong*** Nomor ***226/KD-BLG/IV/2021*** Tanggal : ***20 April 2021*** bahwa :

Nama Pemohon/ Penanggung Jawab	:	I KOMANG TRISNA DHARMITA
Alamat Pemohon/ Penanggung Jawab	:	JL. HANGTUAH GG. MAWAR NO. 22 DPS, BR/LINK.BELONG, Kode Pos : 80227, Sanur Kaja, Denpasar Selatan, Kota Denpasar, Bali
Nama Perusahaan	:	Warung Made
Jenis Usaha	:	Berdagang Sembako

Sepanjang pengetahuan kami dan sampai surat ini dikeluarkan memang benar usaha tersebut beralamat di :

Jalan	:	Jl. Hangtuah Gg Mawar No. 22
Dusun/Lingkungan	:	Belong
Kelurahan / Desa	:	Sanur Kaja
Kecamatan	:	Denpasar Selatan
Kabupaten/Kota	:	Kota Denpasar
Provinsi	:	Bali

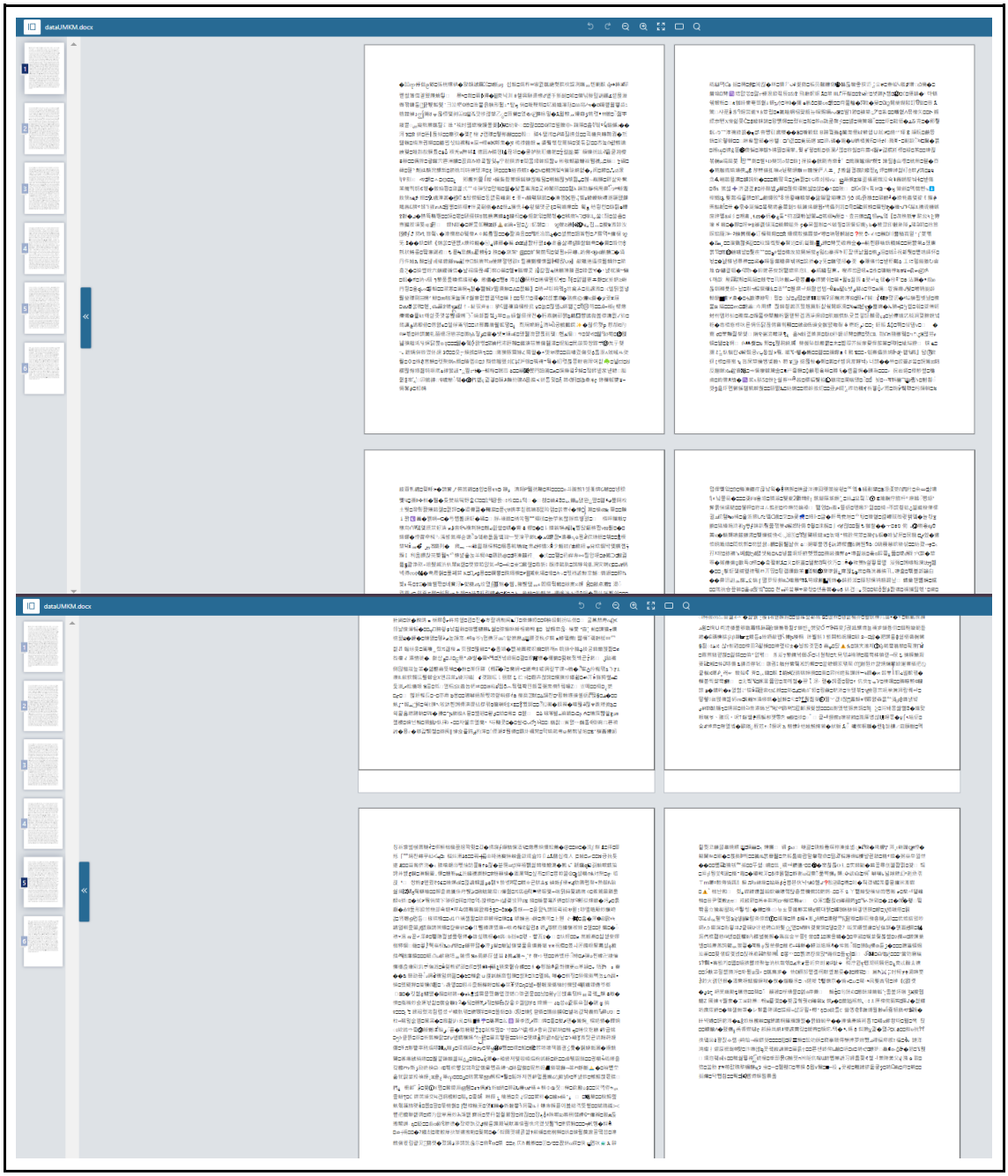
Demikian surat keterangan ini kami buat dengan sebenarnya agar dapat dipergunakan untuk ***Keperluan Administrasi BPUM Kota Denpasar***

Denpasar, 20 April 2021
Perbekel Desa Sanur Kaja

I MADE SUDANA

Dokumen UMKM Sesudah Enkripsi

p-ISSN: 2986-3929
e-ISSN: 3032-1948



b. Proses Dekripsi Dokumen UMKM

- Uji coba menggunakan dokumen yang telah dienkripsi untuk proses dekripsi.
- Dokumen dienkripsi dimasukkan ke dalam program dekripsi.
- Program meminta untuk memasukkan kunci yang digunakan saat proses enkripsi.
- Sistem memulihkan *nonce Initialization Vector* (IV) dari algoritma ChaCha20-Poly1305.
- Program membandingkan kunci yang telah di-*hash* sebelumnya dengan kunci yang dimasukkan.
- Jika kunci tidak valid, program hanya akan memberikan output berupa dokumen kosong atau tidak dapat diakses.
- Jika kunci valid, proses dekripsi dilanjutkan.
- Dokumen terdekripsi disimpan dalam folder yang ditetapkan, yaitu folder "decrypted".

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Tabel 2. Proses Dekripsi

Dokumen UMKM Setelah Dekripsi

PEMERINTAH KODTA DENPASAR
KECAMATAN DENPASAR SELATAN
DESA SANUR KAJA

Alamat : JL. By Pass Ngurah Rai No. 59 Telp. (0361) 287388 Kode Pos 80227
http://sanurkaja.denpasarkota.go.id sanurkaja@gmail.com

SURAT KETERANGAN TEMPAT USAHA

Nomor : 323/Pel.Um/IV/2021

Yang bertanda tangan dibawah ini, Perbekel Desa Sanur Kaja, Kecamatan Denpasar Selatan, Kota Denpasar, berdasarkan surat pengantar Kepala Dusun ***Belong*** Nomor ***226/KD-BLG/IV/2021*** Tanggal : ***20 April 2021*** bahwa :

Nama Pemohon/ Penanggung Jawab	:	I KOMANG TRISNA DHARMITA
Alamat Pemohon/ Penanggung Jawab	:	JL. HANGTUAH GG. MAWAR NO. 22 DPS, BR/LINK.BELONG, Kode Pos : 80227, Sanur Kaja, Denpasar Selatan, Kota Denpasar, Bali
Nama Perusahaan	:	Warung Made
Jenis Usaha	:	Berdagang Sembako

Sepanjang pengetahuan kami dan sampai surat ini dikeluarkan memang benar usaha tersebut beralamat di :

Jalan	:	Jl. Hangtuah Gg Mawar No. 22
Dusun/Lingkungan	:	Belong
Kelurahan / Desa	:	Sanur Kaja
Kecamatan	:	Denpasar Selatan
Kabupaten/Kota	:	Kota Denpasar
Provinsi	:	Bali

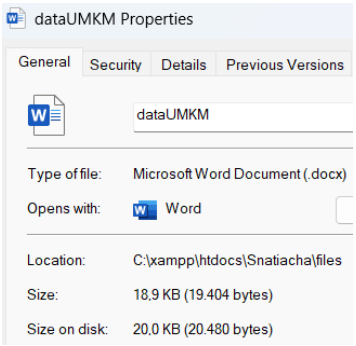
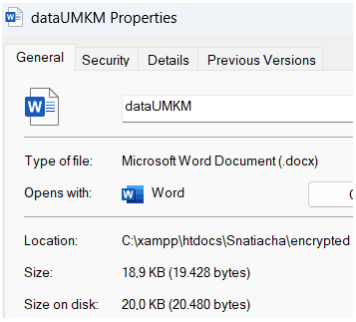
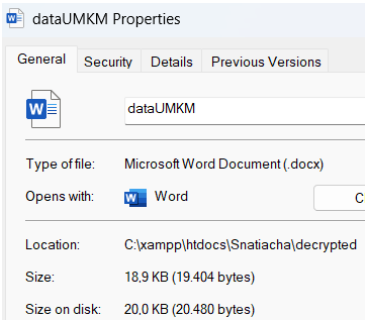
Demikian surat keterangan ini kami buat dengan sebenarnya agar dapat dipergunakan untuk ***Keperluan Administrasi BPUM Kota Denpasar***

Denpasar, 20 April 2021
Perbekel Desa Sanur Kaja

I MADE SUDANA

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Tabel 3. Perbandingan Besaran File

Default	Encrypted	Decrypted
		
Besaran: 19.404 *bytes*	Besaran: 19.428 *bytes*	Besaran: 19.404 *bytes*

4. Kesimpulan

Berdasarkan hasil dan bahasan yang dijelaskan, penelitian ini telah berhasil menguji coba algoritma ChaCha20-Poly1305 dengan kunci *hash* SHA-256 pada enkripsi dan dekripsi dokumen UMKM. Program dapat melindungi dengan baik data dokumen UMKM menggunakan *nonce Initialization Vector* (IV) yang dibangkitkan secara acak untuk dapat melakukan proses enkripsi. Program juga menunjukkan hasil proses dekripsi yang aman dan baik dilihat dari tidak berubahnya ukuran *bytes* dokumen UMKM yang diuji. Dengan penelitian ini, diharapkan untuk penelitian selanjutnya agar dapat mengembangkan algoritma sejenis pada pengamanan data.

Daftar Pustaka

[1] M. Azhari, D. I. Mulyana, F. J. Perwitosari, dan F. Ali, “Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES),” Jurnal Pendidikan Sains dan Komputer, vol. 2, no. 01, pp. 163–171, Mar. 2022, doi: 10.47709/jpsk. v2i01.1390.

[2] F. P. Utama, G. Wijaya, R. Faurina, and A. Vatresia, “Implementasi Algoritma AES 256 CBC, BASE 64, Dan SHA 256 dalam Pengamanan dan Validasi Data Ujian Online,” Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 10, no. 5, pp. 945–954, Oct. 2023, doi: 10.25126/jtiik.20231056558.

[3] Djong, Handrian Saputra, and Siswanto Siswanto. "Implementasi Kriptografi Dengan Menggunakan Metode RC4 dan AES-256 Untuk Mengamankan File Dokumen pada PT Varnion Technology Semesta." Prosiding Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI). Vol. 1. No. 1. 2022.

[4] G. D. M. Zulma, H. B. Seta, dan T. Yuniati, “Implementasi Algoritma Aes Dan Bcrypt Untuk Pengamanan File Dokumen,” Informatik : Jurnal Ilmu Komputer, vol. 18, no. 2, p. 163, Aug. 2022, doi: 10.52958/iftk.v18i2.4667.

[5] R. Serrano, C. Duran, M. Sarmiento, C.-K. Pham, and T.-T. Hoang, “*ChaCha20–Poly1305 Authenticated Encryption with Additional Data for Transport Layer Security 1.3*,” *Cryptography*, vol. 6, no. 2, p. 30, 2022, doi: 10.3390/cryptography6020030.

[6] A. M. ElRashidy, M. H. Abd AlAzeem, A. A. Abd ElHafez, and M. F. Abo Sree, “*ChaCha20-AES Combined Algorithm with 512 Bits of Security*,” *2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*. 2024, pp. 1–6. doi: 10.1109/REEPE60449.2024.10479797.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong