JNATIA Volume 2, Nomor 3, Mei 2024 p-ISSN: 2986-3929
Jurnal Nasional Teknologi Informasi dan Aplikasinya e-ISSN: 3032-1948

Perlindungan Transkrip Akademik Mahasiswa dengan Kombinasi Algoritma Rijndael dan SHA-3

Amsal Hamonangan Butarbutar[a1], Ida Bagus Gede Dwidasmara[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]butarbutar.2208561134@student.unud.ac.id
[2]dwidasmara@unud.ac.id

Abstract

This paper addresses the issue of securing academic transcripts for students by utilizing a combination of Rijndael encryption algorithm and Secure Hashing Algorithm-3 (SHA-3). The primary purpose is to enhance the protection and integrity of academic records against unauthorized access and tampering. SHA-3 is selected for its advanced cryptographic capabilities, surpassing its predecessors in security and efficiency. When combined with the Rijndael algorithm, which serves as the foundation for the Advanced Encryption Standard (AES), the system achieves robust encryption and data integrity verification. Through comprehensive design, implementation, and testing phases, the study demonstrates that the proposed method is effective in safeguarding academic transcripts. The results indicate that the system not only ensures data security but also operates efficiently. This research provides a valuable reference for further development and implementation of secure academic data management systems.

Keywords: *Rijndael, SHA-3, Academic Transcript, Cryptography*

1. Pendahuluan

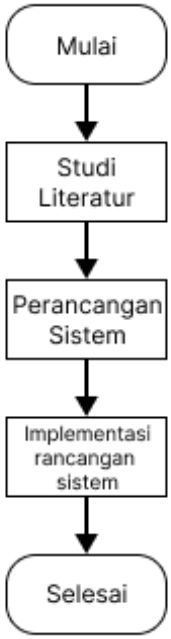
Di era digital ini, keamanan data menjadi salah satu aspek yang sangat penting dalam berbagai bidang, termasuk pendidikan tinggi. Dalam konteks akademik, transkrip mahasiswa adalah dokumen yang menyimpan rekam jejak akademis seseorang, yang mencakup detail kursus yang diambil, nilai yang diperoleh, dan prestasi lainnya. Kehandalan dan keamanan transkrip akademik menjadi krusial dalam memastikan integritas dan kepercayaan terhadap proses pendidikan. Namun, tantangan muncul ketika mempertimbangkan perlindungan data ini dalam lingkungan digital yang rentan terhadap berbagai serangan siber yang dapat dengan mudah memodifikasi data menyebabkan hilangnya integritas data pada transkrip akademik tersebut. Dalam rangka memperkuat dan mencegah hilangnya integritas dan keamanan transkrip akademik mahasiswa, penelitian ini memperkenalkan sebuah pendekatan yang menggabungkan dua teknologi kriptografi yang kuat. Kriptografi merupakan sebuah ilmu menjaga kerahasiaan pesan dengan cara menyandikannya dalam bentuk yang tidak dapat dipahami lagi[1]. Adapun teknik kriptografi yaitu dengan Algoritma Rijndael dan Secure Hash Algorithm-3 (SHA-3). Algoritma Rijndael lebih efisien dibandingkan dengan algoritma Camellia berdasarkan teori kompleksitas waktu sebagai parameter efisiensi. Algoritma Camellia lebih efektif dibandingkan algoritma Rijndael berdasarkan nilai Avalanche Effect sebagai parameter efektivitas. Algoritma Rijndael memiliki kualitas lebih tinggi dibandingkan algoritma Camellia berdasarkan perhitungan deviasi maksimum, koefisien korelasi, deviasi ketidakteraturan, dan PSNR sebagai parameter kualitas. Penelitian ini dilakukan pada file berbentuk citra[2].Sedangkan SHA-3 merupakan teknik hashing yang memiliki kinerja dan ketahanan yang lebih baik daripada algoritma SHA-1 yang menjadi alas an menggunakan algoritma SHA-3 karena merupakan algoritma SHA yang paling baru dan kinerja yang lebih baik daripada varian lainnya[3]. Dengan menerapkan pendekatan tersebut diharapkan dapat memberi perlindungan yang lebih kuat terhadap data sensitif dari ancaman yang beragam termasuk serangan terhadap keutuhan data dan upaya modifikasi data informasi pada transkrip akademik tersebut.

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

2. Metode Penelitian

2.1 Alur Penelitian

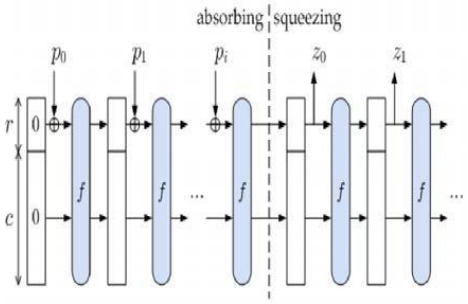
Pada penelitian ini, metode pengumpulan data yang dilakukan adalah secara kualitatif dengan melakukan studi literatur yang bertujuan untuk mengumpulkan semua metode yang digunakan dan dibutuhkan dalam penelitian ini sebagai sumber informasi yang didapat dari berbagai jurnal dan riset ilmiah yang sudah pernah dilakukan agar dapat mendukung gagasan yang akan dibuat pada penelitian ini. Dan selanjutnya akan mengimplementasikan rancangan tersebut langsung ke dalam sebuah sistem untuk mengujinya.



Gambar 1. Alur Penelitian

2.2 Algoritma SHA-3 (KECCAK)

SHA-3 merupakan salah satu bentuk dari sebuah fungsi hash yang merupakan komplementar dari fungsi SHA-1 dan SHA-2. SHA-3 memiliki sistem kerja dengan konstruksi *sponge* yang panjang dari *message digest* dapat disesuaikan[4]. Pada proses hash suatu *message* diawali dengan proses *padding message* yang secara otomatis menyesuaikan dengan panjang outputnya sesuai yang diinginkan. Selanjutnya membagi *message* yang sudah di-*padding* menjadi ukuran sesusai r-bit[5]. Perancangan SHA-3 menggunakan konstruksi spons yaitu data diserap ke dalam spons. Pada fase ini blok pesan XOR menjadi bagian dari *state* yang kemudian diubah secara keseluruhan menggunakan fungsi permutasi f, kemudian hasilnya diperas. Pada fase ini, blok keluaran dibaca dari *subset* keadaan yang sama. dengan fungsi transformasi f.



Gambar 2. Konstruksi Sponge

Berikut langkah-langkah contoh pembuatan pesan menggunakan hash SHA-3, secara umum adalah sebagai berikut:

a. Menambah Bit Buffer
b. Menambah Nilai Panjang Pesan Asli
c. Menginisialisasi MD Buffer
d. Memproses Pesan dalam Blok dengan ukuran 512 bit

Tabel 1. SHA-3 *Padding*

Type	Output Length	Rate (r)	Capacity (c)
SHA3-224	224	1152	448
SHA3-256	256	1088	512
SHA3-384	384	832	768
SHA3-512	512	576	1024

2.3 Algoritma Rijndael

Algoritma Rijndael menggunakan block berukuran 128-bit input dan memiliki 3 ukuran *key*, yaitu 128-bit, 192-bit, dan 256-bit. Input dan output dari algoritma Rijndael terdiri dari urutan data yang berukuran 128-bit[6]. Urutan data setiap satu kelompok 128-bit input disebut sebagai blok data atau *plaintext* yang kemudian akan dienkripsi menjadi *ciphertext*. Pengelompokkan jenis Rijndael ini didasarkan pada panjang kunci yang digunakan. Angka-angka yang terdapat di belakang kata Rijndael menunjukkan panjang kunci yang digunakan dalam satuan bit. Berdasarkan ukuran bloknya, Rijndael bekerja pada matriks dengan ukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Selain itu, yang menjadi perbedaan tipe Rijndael ini adalah banyaknya round yang dipakai. Rijndael-128 menggunakan 10 round, Rijndael-192 menggunakan 12 round, sedangkan Rijndael-256 menggunakan 14 round.

Adapun tahapan yang dilakukan dalam algoritma Rijndael adalah sebagai berikut ini:

Tahap 1 : AddRoundKey, di mana dilakukan operasi XOR antara state dengan *cipher key*. Tahap pertama ini juga dikenal sebagai *initial round.*

Tahap 2 : Putaran sebanyak (Nr – 1) kali. Setiap putaran mencakup:

- SubBytes: Mengganti isi matriks dengan matriks substitusi (S-Box).
- *Shift Rows*: Menggeser setiap baris dalam state, baris pertama tidak digeser, baris kedua digeser satu posisi, dan seterusnya.
- *Mix Column*: Mengalikan setiap elemen block cipher dengan *Mix-Column Matrix* dan memasukkan hasilnya ke dalam *block cipher* yang baru.

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

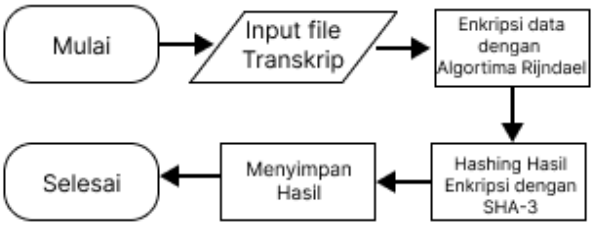
- *AddRoundKey*: Sama dengan tahap pertama, tetapi dilakukan pada setiap putaran dengan meng-XORkan *state* saat ini dengan *round key.*

Tahap 3 : Final round, yang mencakup *SubBytes, Shift Rows*, dan *AddRoundKey* untuk putaran terakhir.

3. Hasil dan Pembahasan

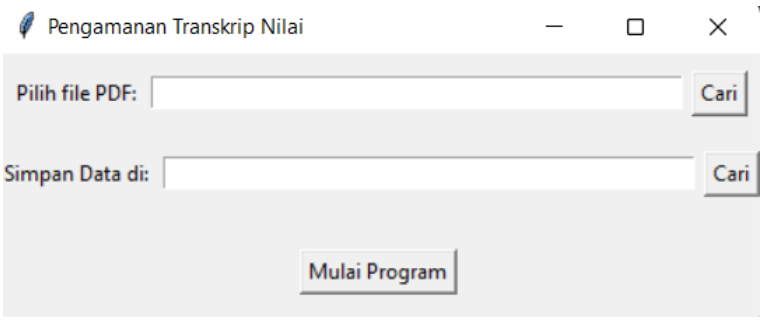
3.1. Rancangan Desain Sistem dan Pengujian

Dalam penelitian ini, rancangan desain sistem yang diharapkan adalah saat user mengunduh transkrip akademi langsung dari website pembelajaran perkuliahan, maka file transkrip akademik tersebut akan otomatis ter-hashing dan terenkripsi, namun karena adanya keterbatasan dalam hal tersebut yang disebabkan oleh beberapa alasan lainnya juga maka Adapun rancangan dan alur sistem yang diharapkan adalah sebagai berikut ini.



Gambar 3. *Flowchart* Sistem

Dimana pada tahap pertama yaitu user akan diminta untuk menginputkan file yaitu file transkrip nilai yang selanjutnya sistem akan membaca isi data dari file transkrip nilai tersebut lalu akan mengenkripsi seluruh data yang ada dalam file transkrip tersebut dengan menggunakan algoritma Rijndael.



Gambar 4. GUI Sederhana Masukan Program

Kemudian akan dilanjut dengan menciptakan sebuah nilai hash dengan SHA-3 dari hasil enkripsi tersebut agar integrasi file transkrip tersebut dapat terjamin dan mengindentifikasi jika adanya perubahan nilai atau data dari file transkrip yang sudah di enkripsi tersebut, selanjutnya data yang dihasilkan dari tahapan-tahapan tadi akan disimpan dalam sebuah database, tetapi dalam pengujian ini akan menyimpannya dalam sebuah file berbentuk *.txt* , penyimpanan nilai meliputi data yang sudah di enkripsi dengan algoritma Rijndael, kunci enkripsi dan dekripsi, dan nilai hash dari nilai data yang sudah di enkripsii tadi. Tujuan dari penyimpanan nilai-nilai tadi agar dapat dengan mudah memverifikasi kembali transkrip nilai tersebut untuk mencegah terjadinya perubahan isi file transkrip nilai.

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

3.2. Implementasi Algoritma

a. Implementasi Algoritma Rijndael

Algoritma Rijndael diimplementasikan untuk mengenkripsi isi dari file transkrip agar tidak terbaca dan dimengerti oleh orang lain terutama yang tidak mengenal teori kriptografi yang menciptakan isi data dari file transkrip tersebut akan sangat sulit untuk diubah karena perlu untuk mendekripsi kembali ke bentuk yang dapat dibaca.

Tabel 2. Penggalan Program Enkripsi Menggunakan Algoritma Rijndael

```
def encrypt_data(data, key):
    cipher = AES.new(key, AES.MODE_EAX)
    nonce = cipher.nonce
 ciphertext,tag= cipher.encrypt_and_digest(data)
    return nonce, ciphertext, tag
```

Fungsi *encrypt_data()* akan meminta variable yang akan menjadi data dan *key* yang mana data merupakan isi dari file yang sudah diinputkan dan *key* merupakan kunci yang sudah diciptakan dengan angka acak yang sudah diatur sedemikian rupa agar menghasilkan kunci yang rumit dan sulit ditebak karena kunci merupakan salah satu hal yang penting untuk diperhatikan. Pengenkripsian menggunakan fungsi dari Tabel 2 dibantu dengan salah satu *library python* yaitu *Crypto.Cipher* untuk meringkas sintaks dan logika yang panjang dalam pengenkripsian menggunakan algoritma Rijndael.

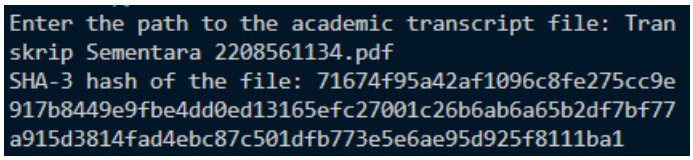
b. Implementasi Secure Hash Algorithm-3 (SHA-3)

Algoritma SHA-3 diimplementasikan menggunakan Bahasa pemrograman Python yang berutujuan untuk menciptakan seuntaian nilai hash unik dari setiap file yang akan diinputkan, setelah itu nilai hash dari tiap file akan disimpan untuk dibandingkan Kembali pada saat user ingin memeriksa integritas dari transkrip akademik mereka.

Tabel 3. Penggalan Program Menghitung SHA-3

```
def hash_sha3(pdf_file):
  with open(pdf_file, 'rb') as f:
    pdf_data = f.read()
  hasher = hashlib.sha3_512()
  hasher.update(pdf_data)
  return hasher.hexdigest()
```

Dari penggalan sintaks diatas, fungsi *hash_sha3()* akan meminta dan menerima inputan file pdf dari user setelah itu akan membuka file tersebut dan membaca isi filenya lalu dengan sintaks yang sudah dipersingkat menggunakan *library* Python akan menciptakan sebuah nilai hash yang disimpan dalam sebuah variable bernama *hasher. Dibawah* ini merupakan bukti pengujian penerapan SHA-3 dalam sistem ini



Gambar 5. Hasil Hashing File Transkrip Akademik

Pada sistem yang sudah dirancang, implementasi dari *Secure Hash Algorithm-3* (SHA-3), digunakan sebagai acuan untuk memastikan keutuhan data transkrip karena perubahan sekecil apapun akan mengubah seluruh nilai hash yang sudah disimpan. Ketika transkrip diakses atau dimodifikasi, nilai hash akan dihitung ulang dan dibandingkan dengan nilai hash yang tersimpan, jika nilai hash tidak cocok maka berarti transkrip telah diubah dan dirusak. Meskipun seseorang mendapatkan Salinan transkrip, mereka tidak dapat mengubahnya tanpa terdeteksi.

Tabel 4. Pengujian Hashing Berbagai File

Nama File	Nilai Hash
Transkrip Sementara 2208561134.pdf	71674f95a42af1096c8fe275cc9e917b8449e9fbe4dd0ed13165efc27001c26b6ab6a65b2df7bf77a915d3814fad4ebc87c501dfb773e5e6ae95d925f8111ba1
Sertifikat Anton.pdf	1cf05dc493b50d47c94e6c10c087e70d24e2f6027ed7f335232fbdedb36d31d36409793c3269052360cd3f4a5e78f45dea755b47dda6c28e5ec0dc37c3c77602
Transkrip Nilai 2208561022.pdf	67e8406c5ea2fdac3e96584397a1ea43ddd0e58c40bf5aa801e399d44276c9ed176e82d68920cbf92ed321a165605480efa1a665513bf32fc6a69cfe0967c1b7

4. Kesimpulan

Kombinasi antara algoritma Rijndael dan SHA-3 dalam penelitian ini terbukti efektif dalam meningkatkan keamanan transkrip akademik mahasiswa. Algoritma Rijndael, yang merupakan dasar dari Advanced Encryption Standard (AES), memberikan enkripsi yang kuat dan cepat, sementara SHA-3 memastikan integritas data dengan fungsi hash yang aman dan tahan terhadap serangan kriptografi modern. Namun demikian, ada beberapa aspek yang dapat ditingkatkan untuk penelitian selanjutnya. Pertama, implementasi sistem ini dapat diuji dalam skala yang lebih besar dan dalam lingkungan yang lebih beragam untuk menguji ketahanannya terhadap berbagai jenis serangan. Kedua, penggunaan teknik tambahan seperti mekanisme autentikasi multifaktor dapat diteliti lebih lanjut untuk meningkatkan keamanan sistem secara keseluruhan. Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi algoritma Rijndael dan SHA-3 adalah solusi yang efektif dan efisien untuk perlindungan transkrip akademik mahasiswa. Implementasi metode ini diharapkan dapat menjadi referensi bagi pengembangan sistem keamanan data akademik lainnya di masa depan

Daftar Pustaka

[1] M. Azhari, D. I. Mulyana, F. J. Perwitosari, and 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, 2022.

[2] B. S. W. Poetro and R. Wardoyo, “Perbandingan Efisiensi, Efektifitas dan Kualitas Algoritma Rijndael dengan Algoritma Camellia pada Citra Digital,” *BIMIPA*, vol. 24, no. 3, pp. 281–291, 2014.

[3] M. P. Sari, “Analisis Algoritma SHA-3 Keamanan pada Data Pribadi,” *JURNAL Tecnoscienza*, vol. 5, no. 2, pp. 231–242, 2021.

[4] R. Munir, “SHA-3 (Keccak),” *Diakses pada*, vol. 20, pp. 2022–2023, 2023.

[5] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, “Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 9, pp. 803–812, 2017.

[6] N. W. Hidayatulloh, M. Tahir, H. Amalia, N. A. Basyar, A. F. Prianggara, and M. Yasin, “Mengenal Advance Encryption Standard (AES) sebagai Algoritma Kriptografi dalam Mengamankan Data,” *Digital Transformation Technology*, vol. 3, no. 1, pp. 1–10, 2023.