Implementasi Teknologi Blockchain dan Algoritma AES 256 dalam Pengelolaan Rekam Medis Elektronik

Alya Apriliyanti 18220050 (*Author*)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
18220050@std.stei.itb.ac.id

***Abstract*—Perkembangan teknologi digital dalam masyarakat meningkatkan kesadaran Kementerian Kesehatan Indonesia untuk melakukan transformasi digitalisasi pelayanan kesehatan. Salah satu langkah untuk mendukung implementasi transformasi di bidang kesehatan adalah diselenggarakannya peralihan rekam medis pasien yang semula berbasis manual menjadi berbasis elektronik. Rekam medis elektronik berisi kumpulan informasi pribadi dan sensitif berkaitan dengan riwayat medis pasien sehingga tantangan yang rentan terjadi dalam pengelolaannya adalah risiko kebocoran data serta akses dan perubahan yang tidak terautentikasi. Oleh karena itu, pada makalah ini, penulis akan mengimplementasikan sistem keamanan dengan menggunakan algoritma AES 256 dan teknologi *blockchain* untuk melindungi integritas dan menjaga kerahasiaan data rekam medis elektronik sehingga informasi sensitif pasien tetap terlindungi dan hanya dapat diakses oleh pihak yang berwenang dengan kunci enkripsi yang tepat.**

Keywords—Kesehatan; Rekam Medis Elektronik; Blockchain; AES-256;

I. PENDAHULUAN

Perkembangan teknologi digital yang sangat pesat telah memberikan dampak yang besar dalam kehidupan manusia. Transformasi digital telah membawa perubahan signifikan dalam berbagai sektor karena memungkinkan terjadinya peningkatan efisiensi kegiatan, peningkatan keamanan dalam pengelolaan data, serta perubahan signifikan lainnya. Banyaknya manfaat dan peluang yang diberikan teknologi digital menyebabkan berbagai sektor berlomba-lomba untuk melakukan transformasi digital, tidak terkecuali pada sektor kesehatan.

Fasilitas pelayanan kesehatan (Fasyankes) berperan penting sebagai akses utama masyarakat untuk mendapatkan kesehatan yang baik serta memastikan bahwa setiap warga negara Indonesia mendapatkan layanan kesehatan yang sama dan terbaik. Salah satu langkah pelaksanaan transformasi digital dalam fasilitas pelayanan kesehatan yaitu pengembangan dan penggunaan Rekam Medis Elektronik (RME) sebagai pengganti rekam medis berbasis manual. Berdasarkan penerbitan Peraturan Menteri Kesehatan (PMK) Nomor 24 Tahun 2022 tentang Rekam Medis, terdapat pernyataan bahwa fasilitas pelayanan kesehatan (Fasyankes) diwajibkan menjalankan sistem pencatatan riwayat medis pasien secara elektronik. [8] Fungsi rekam medis elektronik dalam layanan kesehatan merupakan sesuatu yang vital dan bersifat rahasia. Rekam Medis Elektronik dapat memungkinkan akses yang mudah, penyimpanan yang efisien, dan pembagian informasi pasien yang lebih cepat.

Namun, peralihan data medis pasien menjadi eletronik dapat menyebabkan beberapa masalah yang rentan terjadi. Sistem keamanan dalam Rekam Medis Elektronik (RME) yang tidak dikelola dengan baik dapat meningkatkan risiko kebocoran data serta perubahan dan akses yang tidak terautentikasi pada informasi pribadi dan sensitif pasien. Dilansir dari KOMPAS.com, pada awal tahun 2022 terjadi kebocoran data 6 juta pasien di server Kementerian Kesehatan. [9] Data yang dibobol oleh *hacker* tersebut diduga merupakan hasil rekam medis pasien yang berasal dari berbagai rumah sakit besar di seluruh Indonesia.

Untuk mengatasi masalah tersebut, diperlukan solusi yang dapat melindungi integritas dan kerahasiaan data rekam medis elektronik yaitu dengan menerapkan teknologi *blockchain* dan algoritma AES 256. Teknologi *blockchain* menyediakan mekanisme terdistribusi yang memastikan keamanan dan integritas data dengan mengamankan transaksi dan mencatat setiap perubahan atau penambahan data dalam rantai (*chain*) yang tidak dapat dimanipulasi. Sementara itu, algoritma AES, sebagai salah satu algoritma kriptografi yang kuat, dapat digunakan untuk mengenkripsi data RME sehingga hanya dapat diakses oleh pihak yang berwenang dengan kunci enkripsi yang tepat.

Dengan menerapkan sistem keamanan *blockchain* dan AES 256, diharapkan keamanan dan kerahasiaan Rekam Medis Elektronik dapat ditingkatkan. Informasi pasien akan lebih terlindungi dari ancaman peretasan atau pengaksesan yang tidak sah. Hal tersebut dapat membantu meningkatkan kepercayaan pasien, memfasilitasi pertukaran data antara berbagai fasilitas kesehatan dengan cepat, serta mengefisiensikan proses pengambilan keputusan.

II. METODE

A. Studi Literatur

Tahap awal dalam pembuatan makalah ini adalah melakukan tinjauan pustaka dengan melakukan pencarian informasi dari sumber-sumber digital seperti *paper*, jurnal, dan materi perkuliahan yang berkaitan dengan topik *blockchain*, AES 256, dan Rekam Medis Elektronik (RME). Pada bagian ini, didapatkan informasi berkaitan dengan struktur, mekanisme, dan pengetahuan terkait dengan topik sebagai dasar dalam pengembangan lebih lanjut.

B. Implementasi

Setelah itu, penulis mengembangkan sebuah program menggunakan *library crypto* pada aplikasi desktop. Program ini dibuat menggunakan bahasa pemrograman Python. Program akan dibuat dengan bantuan Tkinter sebagai antarmuka dengan pengguna. Implementasi program dibatasi pada penerapan algoritma AES 256 dan *blockchain* pada Rekam Medis Elektronik (RME) dan berfokus pada sudut pandang admin atau tenaga kesehatan sebagai pengelola data tersebut.

III. DASAR TEORI

A. Rekam Medis Elektronik

Rekam Medis Elektronik (RME) adalah bentuk pengelolaan dan penyimpanan informasi kesehatan pasien dengan menggunakan teknologi digital yang menggantikan rekam medis konvensional yaitu biasanya berupa dokumen fisik seperti kartu atau berkas kertas. RME mencakup berbagai data kesehatan pasien, termasuk riwayat medis, hasil tes laboratorium, diagnosis, catatan dokter, resep obat, dan informasi lainnya yang relevan dengan perawatan dan pengobatan pasien.

Dalam RME, informasi kesehatan pasien diubah menjadi format digital yang dapat diakses dan dikelola secara elektronik. Hal ini membuat penyedia layanan kesehatan yang berwenang, seperti dokter, perawat, atau petugas medis dapat mengakses informasi pasien dengan lebih mudah dan cepat. Selain itu, RME juga memfasilitasi pertukaran informasi antara berbagai fasilitas kesehatan yang terlibat dalam perawatan pasien, seperti rumah sakit, klinik, atau apotek.

Keuntungan utama dalam penggunaan RME adalah peningkatan efisiensi dalam pengelolaan data kesehatan. Dengan data yang tersimpan secara digital, pencarian, pengambilan, dan pembagian informasi menjadi lebih cepat dan efisien. RME juga dapat mengurangi ketergantungan pada rekam medis fisik yang rentan terhadap kerusakan, kehilangan, atau kesalahan pencatatan. Selain itu, RME dapat mendukung pengambilan keputusan klinis yang lebih baik. Dengan akses mudah ke riwayat medis pasien, dokter dapat melihat data yang lengkap dan *up-to-date*, memungkinkan diagnosis yang lebih akurat dan perencanaan perawatan yang tepat. RME juga dapat membantu dalam pemantauan dan analisis data kesehatan populasi untuk tujuan penelitian medis dan perbaikan kualitas layanan kesehatan.

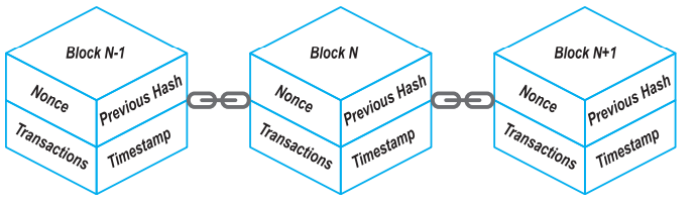
Namun, penggunaan RME juga menimbulkan beberapa tantangan. Keamanan dan privasi data menjadi perhatian utama, karena informasi kesehatan pasien termasuk data yang sensitif dan harus dilindungi dengan baik. Dengan demikian, diperlukan langkah-langkah keamanan yang kuat untuk melindungi kerahasiaan dan integritas RME.

B. Kriptografi

Kriptografi adalah suatu ilmu pengetahuan yang berhubungan dengan teknik pengubahan pesan agar pesan tersebut tetap rahasia dan tidak dapat dibaca oleh pihak yang tidak berwenang. Istilah "Kriptografi" berasal dari Bahasa Yunani, yang secara harfiah berarti menulis secara tersembunyi untuk menyampaikan pesan-pesan yang perlu dijaga kerahasiaannya. Dengan kata lain, kriptografi adalah sebuah cabang ilmu yang membahas tentang cara-cara untuk melindungi keamanan pesan. Kriptografi menyediakan empat layanan utama, yaitu kerahasiaan pesan (*confidentiality*), keaslian pesan (data *integrity*), otentikasi pengirim dan penerima pesan (*authentication*), serta anti penyangkalan (*non -repudiation*).

C. *Blockchain*

Blockchain merupakan sebuah buku besar terdistribusi (*distributed ledger*) dan terdesentralisasi (*decentralized*) yang digunakan untuk menyimpan semua transaksi yang terjadi. Teknologi *blockchain* terdiri dari serangkaian blok yang saling terhubung menjadi suatu rantai dengan menyimpan nilai *hash* dari blok sebelumnya. Tiap blok minimal memiliki informasi berupa nilai *hash* blok sebelumnya, nilai *hash* blok tersebut, dan data transaksi. Berikut ini ilustrasi struktur dari *blockchain.*



Gambar 1. Ilustrasi Struktur *Blockchain*
Sumber: *Blockchain-Based Elctronic Health Records Management: A Comprehensive Review and Future Research Direction (IEEE Access)*

Salah satu sifat yang dimiliki *blockchain* yaitu terdistribusi (*distributed*) yang berarti setiap *peer* berupa komputer, perangkat, atau pengguna pada jaringan *blockchain* akan memiliki catatan transaksi yang sama. Apabila ada satu *peer* yang melakukan perubahan, *peer* lainnya dapat melakukan pengecekan dan dapat dengan mudah menolak perubahan tersebut. *Blockchain* juga bersifat *immutable*. Setiap blok data yang telah dimasukkan ke dalam *blockchain* tidak dapat diubah atau dimanipulasi sesuai dengan protokol yang ada, sehingga perubahan pada data tersebut mengakibatkan terputusnya rangkaian *blockchain*.

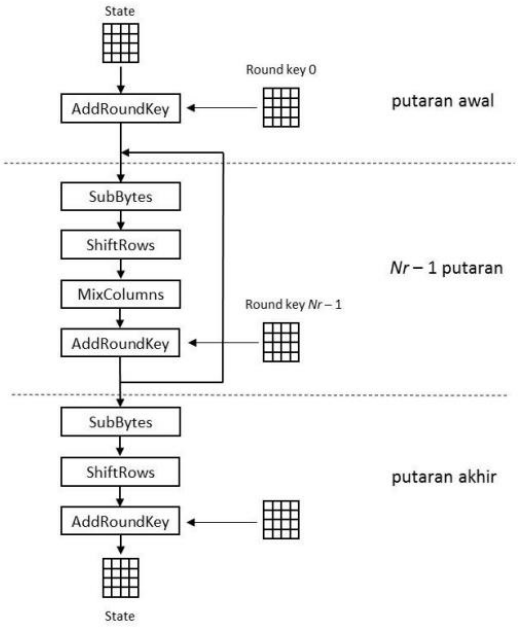
Selain itu, sifat *blockchain* lainnya adalah terdesentralisasi (*decentralized*). Penggunaan sistem terdesentralisasi ini tidak memiliki sistem terpusat atau *central authority* yang

mengelola *blockchain* sehingga validasi data pada sistem tersebut ditentukan melalui consensus dari mayoritas sistem yang ada (*consensus driven*).

D. AES 256

Algoritma AES yang dikenal sebagai algoritma Rijndael adalah sebuah algoritma kriptografi kunci simetris yang menggunakan kunci berukuran 128, 192, atau 256 bit untuk mengubah teks biasa (*plaintext*) berukuran 128 bit menjadi teks sandi (*ciphertext*) berukuran 128 bit. Algoritma ini mengandalkan fungsi substitusi dan permutasi dalam operasinya. Proses enkripsi yang dilakukan pada setiap varian AES, baik AES 128, AES 192, maupun AES 256, pada dasarnya sama. Perbedaannya terletak hanya pada jumlah putaran permutasi yang dilakukan. Pada AES 256, terdapat 14 putaran permutasi yang dilakukan selama proses enkripsi.

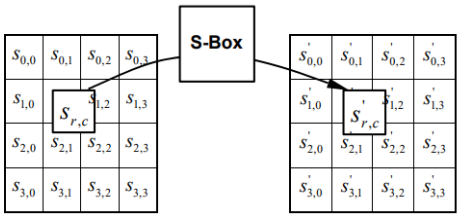
Berikut ini gambaran langkah-langkah operasi algoritma AES secara garis besar.



Gambar 2 Langkah-langkah Operasi AES

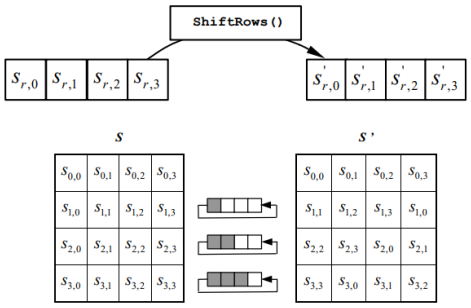
Berdasarkan gambaran tersebut, operasi algoritma AES secara rinci adalah sebagai Berikut.

1. *Initial Round* merupakan tahapan awal untuk melakukan XOR antara plainteks dengan *cipherkey*. Pada tahap berikutnya, proses ini disebut juga sebagai *AddRoundKey*.
2. Proses Permutasi yang dilakukan sebanyak Nr-1 kali. Pada setiap putaran, dilakukan proses berikut ini.
    a. *SubBytes*, melakukan substitusi *byte* dengan menggunakan table substitusi (*S-box*).



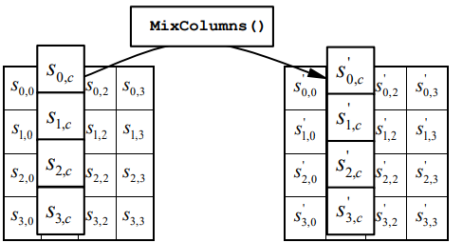
Gambar 3 *SubBytes*(Sumber: NIST, Announcing the AES)

    b. *ShiftRows*, melakukan pergeseran baris-baris *array state* secara *wrapping*.



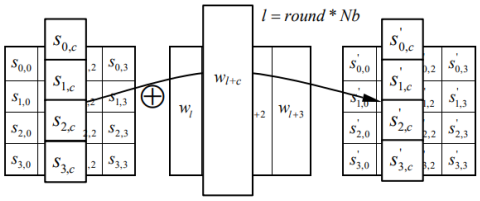
Gambar 4 *ShiftRows* (Sumber: NIST, Announcing the AES)

    c. *MixColumns*, melakukan pengacakan data di masing-masing kolom *array state.*



Gambar 5 *MixColumns* (Sumber: NIST, Announcing the AES)

    d. *AddRoundKey*, melakukan XOR antara *state* sekarang pada *round key* dengan *cipher key.*



Gambar 6 *AddRoundKey* (Sumber: NIST, Announcing the AES)

3. *Final Round*, merupakan tahapan akhir dengan melakukan proses putaran terakhir
    a. *SubBytes*, melakukan substitusi *byte* dengan menggunakan table substitusi (*S-box*).
    b. *ShiftRows*, melakukan pergeseran baris-baris *array state* secara *wrapping*.

c. *AddRoundKey*, melakukan XOR antara *state* sekarang pada *round key* dengan *cipher key*.

IV. Desain dan Rancangan

Untuk menjaga keamanan Rekam Medis Elektronik, akan dikembangkan sistem *blockchain* dengan menerapkan algoritma AES 256 yang dapat melindungi data rekam medis pasien yang bersifat rahasia. Proses enkripsi menggunakan algoritma AES 256 akan dilakukan sebelum *blockchain* dibentuk.

A. Format Data Rekam Medis Elektronik

Rekam Medis Elektronik yang akan digunakan dalam implementasi sistem ini memiliki bentuk JSON (JavaScript Object Notation). JSON dipilih karena kemudahannya dalam memanipulasi data yang awalnya berbentuk *string* menjadi bentuk objek *dictiona*ry. Terdapat dua jenis data rekam medis yang akan penulis gunakan dan hasilkan selama proses eksperimen.

Data rekam medis yang pertama adalah data rekam medis sebelum dimasukan ke dalam *blockchain*, terdiri dari tiga *key* utama yaitu *patient_id*, *name*, dan *medical data* Untuk *key medical data* memiliki 13 *key* turunan yang bersifat pribadi dan rahasia. Selain itu, terdapat data rekam medis yang sudah dimasukkan ke dalam *blockchain*, terdiri dari 6 *key* utama yaitu sama dengan *key* pada rekam medis sebelumnya ditambah dengan *key timestamp*, *previous hash*, dan *hash* dari blok data rekam medis saat ini.

Berikut ini rancangan format data yang akan disimpan ke dalam *blockchain*.


```
[
    {
        "patient_id": "12345",
        "name": "John Doe",
        "medical_data": {
            "tanggal_pemeriksaan": "2023-05-22",
            "nama_lengkap": "John Doe",
            "jenis_kelamin": "Laki-laki",
            "tanggal_lahir": "1990-01-01",
            "alamat": "Jl. Contoh No. 123",
            "nomor_telepon": "08123456789",
            "email": "johndoe@example.com",
            "keluhan": "Demam tinggi dan batuk",
            "riwayat_penyakit": ["Asma", "Hipertensi"],
            "riwayat_alergi": ["Alergi makanan", "Alergi...
        }
    },
….
```


Setelah diterapkan *blockchain* pada data rekam medis, dihasilkan format pesan sebagai berikut.


```
[
    {
        "patient_id": "12345",
        "name": "John Doe",
        "medical_data": {
            "tanggal_pemeriksaan": "2023-05-22",
            "nama_lengkap": "John Doe",
            "jenis_kelamin": "Laki-laki",
            "tanggal_lahir": "1990-01-01",
            "alamat": "Jl. Contoh No. 123",
            "nomor_telepon": "08123456789",
            "email": "johndoe@example.com",
            "keluhan": "Demam tinggi dan batuk",
            "riwayat_penyakit": ["Asma", "Hipert....
        },
        "timestamp": "2023-05-22 20:11:40",
        "previous_hash": null,
        "hash":
"3e78abef70ccd9ffa954c056cc429147d47bdc42d5314127cfafd832d83cc35a"
    },
```


B. Rancangan Algoritma AES 256 dalam *Blockchain*

Langkah-langkah dalam merancang algoritma AES 256 (*Advanced Encryption Standard* dengan kunci 256-bit) pada konteks *blockchain* adalah sebagai beikut.

1. Pemilihan Mode Operasi: Untuk mengenkripsi dan mendekripsi data, mode operasi CBC (*Cipher Block Chaining*) dapat digunakan. Mode ini memastikan bahwa setiap blok data dienkripsi dengan menggunakan hasil enkripsi blok sebelumnya dalam rantai.
2. Pembangkitan Kunci: Sebelum proses enkripsi dan dekripsi dilakukan, kunci enkripsi AES 256-bit yang kuat harus dibangkitkan. Kunci ini harus bersifat rahasia dan hanya diketahui oleh pihak yang berwenang.
3. *Padding:* Data yang akan dienkripsi dengan AES harus memiliki panjang yang merupakan kelipatan dari ukuran blok AES, yaitu 128 bit. Jika data tidak memiliki panjang yang sesuai, padding harus diterapkan untuk memastikan panjang yang tepat sebelum enkripsi dilakukan.
4. Enkripsi: Setelah data dipad, proses enkripsi AES 256 dapat dilakukan. Data dipecah menjadi blok-blok 128-bit dan setiap blok dienkripsi menggunakan kunci AES 256-bit yang telah dibangkitkan sebelumnya. Mode operasi CBC digunakan dengan menggunakan vektor inisialisasi (IV) yang unik untuk setiap blok.
5. *Hashing*: Setelah proses enkripsi selesai, tanda tangan digital (nilai *hash*) dapat ditambahkan ke data yang

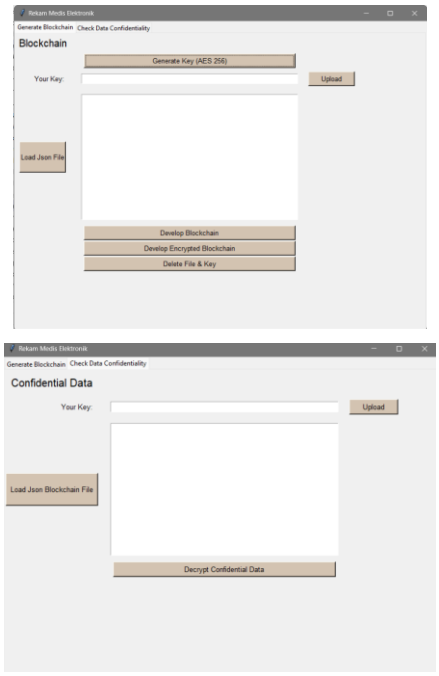
dienkripsi. Nilai *hash* ini menggunakan algoritma SHA-256, untuk memastikan integritas dan otentikasi data yang dienkripsi.

6. Penyimpanan Data: Data yang telah dienkripsi dan ditandatangani digital dapat disimpan dalam blockchain. Data tersebut akan menjadi bagian dari transaksi atau blok dalam blockchain dan akan terdistribusi di seluruh jaringan.
7. Dekripsi: Ketika data yang dienkripsi perlu diakses, proses dekripsi dilakukan menggunakan kunci enkripsi yang sesuai. Data yang dienkripsi dan ditandatangani digital akan didekripsi dengan menggunakan kunci AES 256-bit yang benar, dan tanda tangan digital akan diverifikasi untuk memastikan integritas data yang diterima.

Dengan menggunakan algoritma AES 256 dalam *blockchain*, data sensitif dapat diamankan dan terjamin keutuhannya saat berada di dalam jaringan terdistribusi. Keandalan dan keamanan AES 256 membuatnya menjadi pilihan yang kuat dalam mengamankan data dalam konteks *blockchain*.

V. Implementasi

Proses implementasi sistem kemanan pada Rekam Medis Elektronik berdasarkan desain sebelumnya akan dilakukan dengan menggunakan bahasa pemrograman Python. Bahasa Python dipilih karena musah untuk digunakan, dapat dijalankan di berbagai sistem operasi utama, dan memiliki berbagai *library* yang mendukung kebutuhan pengembangan. Selain itu, bahasa Python banyak digunakan kalangana *developer* sehingga tersedia banyak sumber daya *online* sebagai referensi untuk pengembangan sistem. Berikut ini antarmuka dari sistem penerapan *blockchain* dan algoritma AES 256.



Gambar 7 Tampilan Antarmuka Sistem

Pengembangan program dan antarmuka sistem diwujudkan dalam modul AES-SHA modul pembuatan *blockchain*, dan modul utama penerapan algoritma AES 256 pada *blockchain.*

A. Modul AES-SHA

Modul AES-SHA berisi fungsi enkripsi dan deskripsi, serta pembangkitan kunci secara otomatis.

```
# AES encryption functions
def encrypt_data(key, data):
    cipher = AES.new(key, AES.MODE_CBC)
    encrypted_data = cipher.encrypt(pad(data.encode(),
AES.block_size))
    return cipher.iv + encrypted_data

# AES decryption functions
def decrypt_data(key, ciphertext):
    try:
        iv = base64.b64decode(ciphertext[:24])  # Extract IV and
decode from base64
        if len(iv) != 16:
            raise ValueError("Invalid IV length")
    except (binascii.Error, ValueError) as e:
        try:
            iv = base64.b64decode(ciphertext[:24] + '==')  # Add
padding manually
            if len(iv) != 16:
                raise ValueError("Invalid IV length")
        except (binascii.Error, ValueError) as e:
            print("Error decoding IV:", e)
            return None
    try:
        cipher = AES.new(key, AES.MODE_CBC, iv)
        decrypted_data =
unpad(cipher.decrypt(base64.b64decode(ciphertext[24:])),
AES.block_size)
        return decrypted_data.decode()  # Convert decrypted data
to string
    except (ValueError, binascii.Error) as e:
        print("Error decrypting data:", e)
        return None

# Generate a random encryption key
def encryption_key():
    return get_random_bytes(32)  # 256-bit encryption key
```

B. Modul Pembuatan *Blockchain*

Modul pembuatan *blockchain* dikembangkan untuk membentuk struktur *blockchain* pada masukan rekam medis pasien. Di dalamnya terdapat fungsi untuk menambahkan *timestamp*, nilai *hash* data tersebut, dan nilai *hash* blok sebelumnya.

```
# Function to calculate the hash of a block
def calculate_hash(data):
    # You can implement your hash calculation logic here
    data_string = json.dumps(data, sort_keys=True).encode()
    return hashlib.sha256(data_string).hexdigest()

# Create a function to add a new block to the blockchain
def add_block_encrypted(patient_id, name, medical_data,
encryption_key):
    # Encrypt the medical data
    encrypted_medical_data = encrypt_data(encryption_key,
json.dumps(medical_data))

    # Create a new block dictionary
    block = {
        "patient_id": patient_id,
        "name": name,
        "medical_data": encrypted_medical_data.hex(),
        "timestamp": datetime.now().strftime("%Y-%m-%d
%H:%M:%S"),
        "previous_hash": None
      }
    return block

def add_block_decrypted(patient_id, name, medical_data,
encryption_key):
    # Encrypt the medical data
    decrypted_medical_data = decrypt_data(encryption_key,
json.dumps(medical_data))

    # Create a new block dictionary
    block = {
        "patient_id": patient_id,
        "name": name,
        "medical_data": decrypted_medical_data,
        "timestamp": datetime.now().strftime("%Y-%m-%d
%H:%M:%S"),
        "previous_hash": None
    }
    return block
```

```
def add_block(patient_id, name, medical_data):
    # Create a new block dictionary
    block = {
        "patient_id": patient_id,
        "name": name,
        "medical_data": medical_data,
        "timestamp": datetime.now().strftime("%Y-%m-%d
%H:%M:%S"),
        "previous_hash": None
    }
    return block

def add_hash(blockchain, block):
    # Calculate the hash of the previous block
    if len(blockchain) > 0:
        previous_block = blockchain[-1]
        block["previous_hash"] =
calculate_hash(previous_block)
    # Calculate the hash of the current block
    block["hash"] = calculate_hash(block)
    # Append the block to the blockchain
    blockchain.append(block)
    return blockchain
```

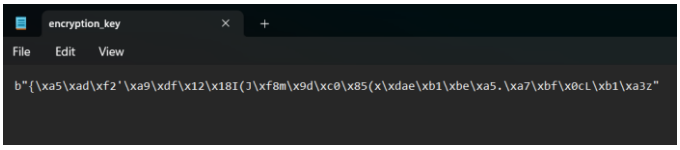
C. Modul Utama

Modul utama digunakan sebagai eksekusi untuk menerapkan algoritma AES 256 dan *blockchain* pada Rekam Medis Elektronik (RME). Pada modul ini pun antarmuka sistem dikembangkan dengan menggunakan *library* Tkinter.

VI. Pengujian dan Diskusi

Setelah implementasi dilakukan, sistem keamanan sederhana yang sudah dibuat akan diuji untuk memastikan bahwa sistem dapat digunakan sesuai fungsinya. Untuk melakukan pengujian apakah penerapan algoritma AES 256 dan *blockchain* telah berhasil dibuat, penulis membuat skenario pengujian pembuatan *blockchain* pada rekam medis elektronik yang akan menghasilkan *blockchain* rekam medis yang telah terenkripsi dan *blockchain* yang tidak dilakukan enkripsi.

Pertama-tama, penulis akan membangkitkan kunci enkripsi dengan fungsi *randomizer* dan didapatkan kunci dengan ukuran 32 *byte* (sesuai dengan AES 256). Selanjutnya, kunci akan disimpan menggunakan format *file txt*. Berikut kunci yang telah berhasil di-*generate*.



Gambar 8 Kunci Enkripsi

Selanjutnya, data rekam medis pasien yang berbentuk *file* json akan di-*upload*. Lalu, proses penerapan *blockchain* yang terenkripsi dan *blockchain* yang tidak dienkripsi dapat dimulai dengan menekan tombol masing-masing. Didapatkan hasil penerapan *blockchain* yang tidak dilakukan enkripsi adalah sebagai berikut.


```
[
    {
        "patient_id": "12345",
        "name": "John Doe",
        "medical_data": {
            "tanggal_pemeriksaan": "2023-05-22",
            "nama_lengkap": "John Doe",
            "jenis_kelamin": "Laki-laki",
            "tanggal_lahir": "1990-01-01",
            "alamat": "Jl. Contoh No. 123",
            "nomor_telepon": "081….
            ....
        },
        "timestamp": "2023-05-22 23:04:56",
        "previous_hash": null,
        "hash": "b2594f91b8e16933bb4c64d03f15bfa5dfb
19960cf540637a503cccd73a29828"
    },
    {
        "patient_id": "67890",
        "name": "Jane Smith",
        "medical_data": {
            "tanggal_pemeriksaan": "2023-05-23",
            "nama_lengkap": "Jane Smith",
            "jenis_kelamin": "Perempuan",
            "tanggal_lahir": "1985-03-15",
            "alamat": "Jl. Contoh No. 456",
            "nomor_telepon": "08123456788",
            "email": "janesmith@....
            ....
        },
        "timestamp": "2023-05-22 23:04:56",
        "previous_hash": "a46850f32787382c11bad1fff09f8fa4
87b36a9c7c2c9bf17d7f464584d067ef",
        "hash": "24129698766ca14a17dde61231347d6b19
7b2e7262842f2a21c6564cc956abde"
    },
....
```


Sementara itu, hasil *blockchain* terenkripsi adalah sebagai Berikut.


```
[
    {
        "patient_id": "12345",
        "name": "John Doe",
        "medical_data": "8ee7c715c9c960ef022cb3328580a3479c5
e48e378d9e0ac79bc381fb72d86c01d04806927c508393259fb04bb79c70
0016d2f4b3e3af42d3525308b62821866190dd50ad555e794758b8b6b78e
f9ffafeba9d210abb8560921833c3f6ea8bb555feb2ade38568b1eb7a282
ed59936ea8d1cf281e84a0e5ff06ae5223482b78a464c89a2cc15dc503f7
cfe098beb4975c15c78398bac3ba7e3ebf5a4f3bf4a8e45c14091a9e94e3
ea436ead26267b196775d6e1a0be99876f0c2d8b0873085a8dc9ad458ff7
944a54da4535c0174eaca81e778c4127543930583f0621f2dd6c8f12d5e6
6197da695d61f8bd48a866335ada3c0d2c48021918fb29aad725cece3c4
78ded09ff23a13c763be5aa6ea48e7d0b1d379c514642f5b829366c520b
479cb7f482d58185759e5e7e8b94af2c4db69fe36cac933f4eafae407a8
19eb223e49ef3e0521c48ad4159a1e42a785c8a0a89b3b6e66592533cc4
02bb9a650d3c7d6faee0bddd6162754c855a9475fcfa4db090e1f2a29d4a
c8322d94dc74fefd0783dea74eb4af3d84b30ad88ed6b7615044fc8b4929
aea118a5c87536fc11d02f1c3b6d5a69133f5b64fcac492e658aa719c4c2
71a3c84c9191b02b68b9490fe4ff5f3540cba5afc84599c77eb3fd8a44ac
7d6b37f4822e853f6e60992c185eef08268baaf06adb7cd4d9594c77925
f39187e93253800198c470c9edd4e15ee334afa02d83a47f7e55d8c2be1b
dec63bec3418a289a15648c24d4d6ab65ce0e35a8571f7646a23598a883d
407e117b33688d0b3353a44191a77b3fe631cd652f5c20e3c313c6189cbc
668f75b89b791b4a32a3def3bed9482878a526b572ce3a759a13daf10848
3a540d96051b6bffb3bab70eba3c15344ef5550130b2aa2271c248a46696c
1ff8b798e0d09927e41745c624647ea208678d83d17b7625f8046a46a06b3
dc55ce8b869704deefd5641e0c4dd5336c2eb7455b08441ee8befe7d",
        "timestamp": "2023-05-22 23:15:54",
        "previous_hash": null,
        "hash": "acf3b5725717b294032c66e447104e068645b3a293ff0
5307ee3ed760d7091f3"
    },
    {
        "patient_id": "67890",
        "name": "Jane Smith",
        "medical_data": "06b43c49f05fb97c51cfab889e38c9e8caf9a91e
ff178d57502d87ad0edbb9ab7c5da3ccf4faf5bda2507ac28da94dac7db0bf47f
8a00363ee73e209aa68f8d33cb4e8b1d0d3aec5b22eedb1bb3649a6eb0b011b
891b5a09e5ced4f7394878eb7abd2d3d54cb0d6be0237b8a0b5f411131af07ca
c20aed766fbd0b3cf25658d3ffe27bd0c0c11a4ed5207f3cbd515617c217d82
f11b7c9d7a6e36a9de9612f1fd662ed26a7dbebceecfa2012b851de11f3750d
2a07bb17660ddb84a3428a0df7683f1a190161737cc32c0b4c81b8098b8d7022
239e65aae8a3e7a63fdfb0da1e80b9f1b9fda447147a9c24dba8741bc80875b6
```

```
e3c026d8077847e45a690c5f360244308f03b9c07b6bdb93ab6ff0839a0498d6
4d308499c93e2bdf0f9921fdd4dce59bad71086486e83274747eea05a22ea159
6df888f0f642d91ff560fef522771b4ef6ea58214712b15014ef6e3c5a8a5024
7f6672770f11c14668eb4449ccb1e6efd2a2dbf5a66527690a9be042acf88936
e0d787b730106c88688006b10a890f00ec44820f34e7a0e1fc13aefd345013ca
608dc12ab92da89ad453c5ed69c8841d757044a87231c9625c4703f01d93c6bc
7e0fbba0047b827dc29c950facdec7fb6adfa0d6fec86a78355d2ca73fe4a2b4
c80987453aa9d0f875086f4405195d5d6b9eaef0416e184b8f371726d89bf731
3e420314bf97c1bd5ef8cac59f81acb4900b145ff44bfe63430b1454744b7491
c93cc4e91bddad4212bee655878d8e3865e2ba1554f4ccb01450a25629d83d5e
2d70ae674ba90ce51c073d1edbc5808f22a9aebf8c6f0c181ec441fd11f014a4f
dd4ac2253d51e44d1dc7fea5568a2add52f3dd355aa826e5fc9ba90f72f1b758
17aff343c9714a6b2198a28ebd178129fc4819ac246e69c342cdd8f8179b0ed10
fbe67b1ea5e74bc24210e024",
        "timestamp": "2023-05-22 23:15:54",
        "previous_hash":
"75a36ca99131ac499407f333c7c7ded1f1c89effe8edb14c68a982efaf838132",
        "hash":
"ffd89e7ed041bcdea8dd0c1f2bb69c76e75e9913415f9b1c8672d9bad4f5f8f1"
    },
…
```


Dari segi keamanan data, Rekam Medis Elektronik yang diterapkan oleh *blockchain* hanya bisa menjaga integritas atau keaslian data pasien sehingga membuatnya sulit untuk dimanipulasi karena terdapat nilai *hash* yang saling terhubung pada tiap bloknya. Namun, data yang disimpan dalam blok yang tidak dienkripsi dapat diakses oleh siapa pun yang memiliki akses ke *blockchain* tersebut, sehingga belum dapat menjaga kerahasiaan data sepenuhnya selain dari perlindungan *hardware* dan kebijakan yang telah diterbitkan. Sementara itu, data rekam medis yang dienkripsi menggunakan kunci rahasia AES 256 sebelum disimpan ke dalam *blockchain* dapat memberikan lapisan tambahan keamanan karena data yang disimpan dalam blok telah dienkripsi dan hanya bisa diakses oleh pihak tertentu yang memiliki kunci enkripsi yang sesuai. Dengan demikian, data rekam medis menjadi lebih sulit dilakukan pembacaan, penyalahgunaan, maupun manipulasi oleh pihak yang tidak berwenang.

VII. Kesimpulan dan Saran

Berdasarkan hasil pengujian dan diskusi, dapat disimpulkan bahwa penerapan algoritma AES-256 pada teknologi *blockchain* sangat bermanfaat untuk mengurangi kebocoran data yang bersifat pribadi dan sensitive pada data Rekam Medis Elektronik (RME). Penggunaan algoritma AES-256 dilakukan sebelum data rekam medis disimpan dalam blok pada *blockchain* sehingga dapat meningkatkan perlindungan keamanan data pasien.

Algoritma AES-256 dan teknologi *blockchain* berhasil diimplementasikan dengan hasilnya berupa *blockchain* pada data yang telah dienkripsi. Berikut ini kesimpulan dan saran yang dapat diambil.

A. Kesimpulan

Penerapan algoritma AES-256 pada teknologi *blockchain* efektif dalam meningkatkan keamanan data rekam medis elektronik. Selain itu, kombinasi antara enkripsi AES-256 dan struktur desentralisasi *blockchain* memberikan lapisan tambahan keamanan dan privasi pada data medis elektronik.

B. Saran

Dalam pengembangan sistem rekam medis elektronik dengan menggunakan *blockchain* dan algoritma AES-256, perlu lebih memperhatikan kembali aspek keamanan secara menyeluruh, termasuk kebijakan akses, manajemen kunci, dan deteksi intrusi. Lalu, diharapkan terus melakukan pembaruan dan peningkatan terhadap sistem, termasuk pemantauan terhadap perkembangan teknologi keamanan terbaru, guna menjaga keamanan data yang optimal.

Source Code

https://github.com/alyaaapril/blockchain-aes-emr

References

[1] Jogi Oliver Yohanes Tampubolon, Adhitya Bhawiyuga, dan Reza Andria Siregar, “Implementasi Blockchain berbasis BigchainDB untuk Menjamin Keamanan Data dalam Sistem Pencatatan Rekam Medis”, 3rd ed., vol. 6. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2022, hlm. 1471-1480.

[2] Sascha Kraus, Francesco Schiavone, Anna Pluzhnikova, Anna Chiara Invernizzi, “Digital transformation in healthcare: Analyzing the current state-of-research,” Elsevier Inc., 2020.

[3] Abdullah Al Mamun, Sami Azam, (Member, IEEE), and Clementine Gritti, “Blockchain-Based Electronic Health Records Management: A Comprehensive Review and Future Research Direction,” IEEE Access.

[4] S Joseph Gabriel, P. Sengottuvelan, “An Enhanced Blockchain Technology with AES Encryption Security System for Healthcare System”, IEEE Explore.

[5] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Penggunaan Kriptografi di dalam Blockchain.

[6] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Advanced Encryption Standard (AES).

[7] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Secure Hash Algorithm (SHA)

[8] Rokom. Fasyankes Wajib Terapkan Rekam Medis Elektronik. 2022 [diakses pada 19 Mei 2023]

[9] Kompas.com. Data 6 Juta Pasien di Server Kemenkes Diduga Bocor, Ini Kata Kominfo. 2022 [diakses pada 19 Mei 2023]

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 22 Mei 2023

Alya Apriliyanti (18220050)