Pengembangan Sistem Pencatatan Sipil dengan Blockchain dan Algoritma Kriptografi AES-256

Hilmi Baskara Radanto 18221072 (Author)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
18221072@std.stei.itb.ac.id:

***Abstract*— Sistem pencatatan sipil merupakan elemen penting dalam administrasi suatu negara, memainkan peran vital dalam mendokumentasikan peristiwa kehidupan seperti kelahiran, pernikahan, perceraian, kematian, dan perubahan nama. Namun, sistem pencatatan sipil konvensional menghadapi masalah terkait keamanan dan keandalan data. Makalah ini mengusulkan pengembangan sistem pencatatan sipil berbasis *blockchain* dengan implementasi algoritma kriptografi AES-256 untuk mengatasi tantangan ini. Teknologi *blockchain* menyediakan solusi penyimpanan data yang terdesentralisasi dan tahan terhadap manipulasi, sementara algoritma AES-256 mengenkripsi data pada *blockchain* sehingga meningkatkan keamanan data. Hasil pengujian sistem yang dikembangkan menunjukkan kemampuan sistem untuk mengenkripsi dan menyimpan data dengan aman di *blockchain* serta mendekripsinya kembali untuk ditampilkan dengan akurat. Pendekatan ini efektif dalam meningkatkan keamanan dan keandalan data pencatatan sipil, menawarkan manfaat seperti peningkatan keamanan data pribadi, efisiensi operasional, dan peningkatan kepercayaan publik terhadap pencatatan sipil.**

Keywords—Sistem Pencatatan Sipil, AES, Blockchain

I. PENDAHULUAN

Pencatatan sipil merupakan elemen yang sangat penting dalam administrasi sebuah negara dengan memainkan peran krusial dalam mendokumentasikan kejadian-kejadian penting dalam kehidupan warganya. Beberapa kejadian penting yang dicatat adalah kelahiran, pernikahan, perceraian, kematian, dan penggantian nama [1]. Sistem pencatatan yang efektif tidak hanya membantu pemerintah dalam perencanaan dan pemberian layanan, tetapi juga memberikan identitas hukum yang sah bagi setiap individu, yang penting untuk mengakses berbagai hak dan layanan publik [2].

Namun, sistem pencatatan sipil konvensional rentan menghadapi berbagai masalah, terutama terkait dengan keamanan dan keandalan data. Salah satu contoh peristiwanya adalah pada berita yang ditulis oleh *The Jakarta Post* pada Juli 2023 lalu bahwa telah terjadi satu peristiwa kasus dugaan kebocoran data pencatatan sipil yang melibatkan Kementerian Dalam Negeri [3].

Dalam menghadapi tantangan tersebut, diperlukan solusi yang dapat meningkatkan keamanan dan keandalan pada data pencatatan sipil. Teknologi *blockchain* telah muncul sebagai salah satu solusi potensial yang mampu mengimplementasikan penyimpanan data sistem pencatatan sipil yang aman, terdesentralisasi, dan sulit untuk dimanipulasi.

Selain itu, algoritma kriptografi AES (*Advanced Encryption Standard*) dapat diterapkan untuk memberikan lapisan keamanan. Dengan AES, data pencatatan sipil yang disimpan dalam sistem *blockchain* dapat dienkripsi sehingga hanya pihak yang berwenang yang dapat mengakses informasi tersebut. Kombinasi antara *blockchain* dan AES menciptakan sistem yang tidak hanya tahan terhadap manipulasi, tetapi juga memastikan bahwa data pribadi tetap aman dan terlindungi.

Oleh karena itu, penulisan makalah ini bertujuan untuk mengembangkan sistem pencatatan sipil berbasis *blockchain* dengan implementasi algoritma kriptografi AES. Sistem ini diharapkan dapat berkontribusi untuk mengatasi berbagai masalah yang dihadapi oleh sistem pencatatan sipil konvensional. Beberapa manfaat utama yang dapat diperoleh meliputi peningkatan keamanan data pribadi, efisiensi operasional, dan peningkatan kepercayaan publik terhadap pencatatan sipil. Dengan demikian, sistem ini tidak hanya akan menguntungkan pemerintah dalam pengelolaan data warganya, tetapi juga memberikan perlindungan dan layanan yang lebih baik bagi masyarakat.

II. METODE

A. Studi Literatur

Tahap awal dalam pembuatan makalah ini adalah melakukan studi literatur yang dilakukan dengan melakukan pencarian informasi dari sumber-sumber digital seperti *paper*, jurnal, dokumentasi, dan materi perkuliahan yang berkaitan dengan topik *blockchain*, AES-256, dan Sistem Pencatatan Sipil. Pada bagian ini, didapatkan informasi berkaitan dengan struktur, mekanisme, dan pengetahuan terkait dengan topik sebagai dasar dalam pengembangan lebih lanjut.

B. Implementasi

Setelah dilakukan studi literatur, penulis melakukan implementasi dengan membagi program ke beberapa bagian. Pertama, penulis mengembangkan *smart contract* untuk *blockchain* yang ditulis dengan bahasa pemrograman *Solidity* dan di-*deploy* melalui jaringan *Sepolia Testnet*. Kemudian,

algoritma kriptografi AES diimplementasikan dengan Bahasa premrograman *TypeScript* yang menggunakan *library crypto-js*. Setelah itu, penulis mengembangkan *user interface* yang memungkinkan user untuk melakukan input data dan pengecekan data menggunakan framework *Next.js* yang disediakan oleh *React*. Terakhir, program menggunakan *library React Truffle Box* untuk menghubungkan *user interface* dan *blockchain*.

III. Dasar Teori

A. Sistem Pencatatan Sipil

Sistem pencatatan sipil adalah mekanisme yang digunakan oleh pemerintah untuk mendokumentasikan kejadian-kejadian penting dalam kehidupan warganya, seperti kelahiran, pernikahan, kematian, perceraian, dan penggantian nama. Sistem ini memberikan identitas hukum yang sah bagi setiap individu, yang penting untuk mengakses berbagai hak dan layanan publik [1].

Sistem pencatatan yang efektif membantu dalam perencanaan kebijakan publik dan administrasi negara, serta mendukung pelaksanaan kebijakan publik yang lebih efektif. Sistem pencatatan sipil yang baik berkontribusi pada peningkatan hak-hak sosial dan ekonomi warga negara, serta mendukung pelaksanaan kebijakan publik yang lebih efektif [4].

Selain itu, sistem ini menyediakan data yang akurat dan terbaru yang membantu pemerintah dalam merancang dan melaksanakan kebijakan publik yang tepat sasaran. Sistem pencatatan sipil yang efektif juga memainkan peran penting dalam pengelolaan data demografis suatu negara, serta dalam identifikasi dan perlindungan kelompok rentan dengan menyediakan dokumen yang sah yang diperlukan untuk mendapatkan bantuan sosial.

B. *Blockchain*

Blockchain atau rantai blok adalah teknologi yang memungkinkan pencatatan transaksi secara terdesentralisasi, transparan, dan aman. *Blockchain* pertama kali diperkenalkan oleh Satoshi Nakamoto pada tahun 2008 sebagai bagian dari mata uang digital *Bitcoin*. Teknologi ini telah berkembang dan diterapkan dalam berbagai bidang, termasuk pencatatan sipil.

Blockchain bekerja dengan menyimpan data dalam banyak *node* (komputer) yang terdistribusi di seluruh jaringan, sehingga mengurangi risiko kehilangan data dan manipulasi. Data dalam blockchain disimpan dalam bentuk blok, di mana setiap blok berisi sejumlah transaksi dan informasi mengenai blok sebelumnya, yang kemudian dihubungkan secara berurutan membentuk sebuah rantai (*chain*). Ketika ada transaksi baru, transaksi tersebut disiarkan ke seluruh jaringan dan divalidasi oleh *node-node* di jaringan. Setelah divalidasi, transaksi dikelompokkan ke dalam blok baru yang kemudian ditambahkan ke akhir rantai *blockchain*. *Blockchain* menggunakan algoritma konsensus (seperti *Proof of Work* atau *Proof of Stake*) untuk memastikan bahwa semua *node* di jaringan setuju dengan status terbaru dari *blockchain*. Transparansi *blockchain* memungkinkan setiap transaksi yang pernah dilakukan dan disimpan dalam *blockchain* dapat dilihat oleh siapa saja yang memiliki akses ke blockchain tersebut, sehingga meningkatkan kepercayaan [5].

C. *Smart Contract*

Smart contract adalah program komputer yang berjalan di atas *blockchain* yang secara otomatis mengeksekusi perjanjian atau transaksi ketika kondisi tertentu terpenuhi. *Smart contract* ditulis dalam bahasa pemrograman yang spesifik, seperti Solidity untuk Ethereum yang di-*deploy* ke blockchain sebagai bagian dari transaksi. Keunggulan *smart contract* adalah kemampuannya untuk menghilangkan kebutuhan akan perantara, mengurangi biaya transaksi, serta meningkatkan kecepatan dan efisiensi proses. Dalam konteks sistem pencatatan sipil, *smart contract* dapat digunakan untuk mengotomatiskan proses verifikasi dan pencatatan kejadian-kejadian penting, memastikan bahwa setiap langkah dalam proses pencatatan dilakukan sesuai dengan aturan yang telah ditetapkan dan tidak dapat dimanipulasi [6].

D. Kriptografi AES-256

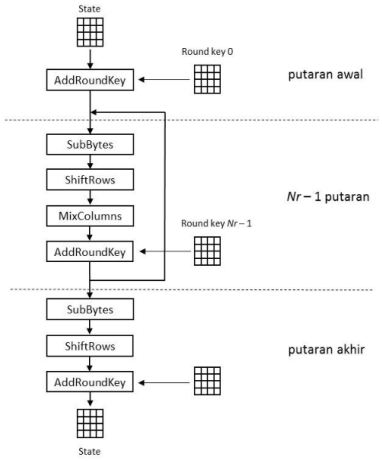
Algoritma kriptografi AES (*Advanced Encryption Standard*) adalah standar enkripsi yang diadopsi oleh pemerintah Amerika Serikat pada akhir 2001. AES menjadi standar enkripsi yang digunakan secara luas karena keamanannya yang tinggi dan efisiensinya dalam proses enkripsi dan dekripsi. AES-256 adalah algoritma enkripsi simetris, yang berarti kunci yang sama digunakan untuk enkripsi dan dekripsi data. AES-256 merupakan salah satu varian dari AES yang menggunakan kunci enkripsi sepanjang 256 bit, memberikan tingkat keamanan yang lebih tinggi dibandingkan varian lain seperti AES-128 dan AES-192. [7]

Secara garis besar, terdapat beberapa langkah terkait bagaimana algoritma kriptografi AES-256 bekerja. Langkah pertama adalah *AddRoundKey*, di mana setiap byte dari blok data (*plaintext*) di-XOR dengan byte kunci enkripsi (*round key*) untuk menghasilkan *state awal*. Pada putaran utama, terdapat tiga langkah utama: *SubBytes*, dimana setiap byte dalam state digantikan dengan *byte* lain dari tabel substitusi (S-Box) untuk memberikan sifat non-linear dan meningkatkan keamanan data; *ShiftRows*, di mana baris-baris dalam *state* digeser ke kiri dengan jumlah tertentu untuk setiap baris; dan *MixColumns*, dimana kolom-kolom dalam *state* dicampur menggunakan operasi matematika di *Galois Field* untuk memberikan difusi data yang lebih baik. Setelah itu, *AddRoundKey* dilakukan kembali untuk menggabungkan *state* yang telah diubah dengan round key menggunakan operasi XOR. Pada putaran akhir, langkah-langkah *SubBytes*, *ShiftRows*, dan *AddRoundKey* dilakukan kembali tanpa *MixColumns*, menghasilkan *ciphertext*. Kunci putaran dibangkitkan dengan dengan fungsi *KeyExpansion*() yang memastikan setiap putaran menggunakan kunci yang berbeda, menambah kompleksitas dan keamanan algoritma.

Proses dekripsi AES-256 adalah kebalikan dari proses enkripsi, yang melibatkan langkah-langkah *InvSubBytes*, di mana byte dalam state digantikan kembali dengan byte asli dari invers S-Box; *InvShiftRows*, di mana baris-baris dalam *state* digeser ke kanan dengan jumlah tertentu; *InvMixColumns*, dimana kolom-kolom dalam *state* dicampur kembali dengan

operasi invers di *Galois Field*; dan *AddRoundKey*, di mana state di-XOR kembali dengan *round key* pada setiap putaran. Dekripsi melibatkan urutan yang sama tetapi dalam arah terbalik dari enkripsi, memastikan data yang dienkripsi dapat dikembalikan ke bentuk aslinya.

Berikut adalah gambaran garis besar dari proses enkripsi dengan algoritma kriptografi AES-256.

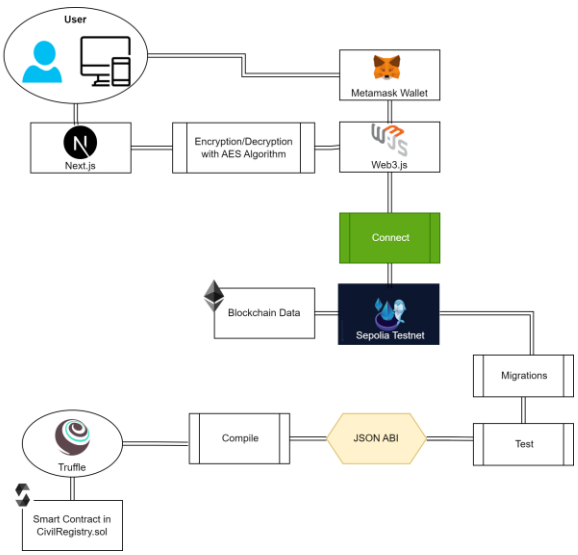


Gambar 1 Langkah-langkah algoritma AES (Sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2023-2024/08-AES-2024)

IV. Desain dan Rancangan

A. Rancangan Sistem

Diagram berikut menggambarkan alur kerja rancangan sistem pencatatan sipil dari pengembangan hingga implementasi dan interaksi pengguna.



Gambar 2 Rancangan Sistem

Terdapat beberapa komponen utama dan alur kerja yang saling terhubung. *User* mengakses aplikasi web yang dibangun menggunakan framework *Next.js*. Data yang dimasukkan melalui antarmuka web kemudian dienkripsi menggunakan algoritma AES-256 untuk memastikan keamanan data sebelum dikirim ke blockchain. Pengguna juga menggunakan *Metamask Wallet* untuk menghubungkan akun Ethereum mereka dan mengautentikasi transaksi. Web3.js, sebuah *library JavaScript*, digunakan untuk berinteraksi dengan blockchain Ethereum melalui Metamask. Proses koneksi antara aplikasi dan blockchain diwakili oleh komponen *Connect* yang menghubungkan website ke jaringan blockchain Sepolia Testnet yang menyimpan data pencatatan sipil.

Di sisi pengembangan *blockchain*, *smart contract* ditulis dalam file CivilRegistry.sol di-*compile* dan di-*deploy* menggunakan library bernama *Truffle*. Proses pengembangan mencakup *compiling* smart contract menjadi JSON ABI (*Application Binary Interface*) yang nanti akan digunakan oleh Web3.js untuk berinteraksi dengan *smart contract* yang sudah di-*deploy*. Setelah itu, smart contract yang sudah di-compile akan dimigrasi ke jaringan *blockchain* Sepolia Testnet untuk bisa berinteraksi ke website yang terhubung dengan testnet.

B. Format Data

Sistem pencatatan sipil yang penulis akan implementasikan dibatasi dengan hanya dua akta, yaitu akta kelahiran dan akta perkawinan. Penyimpanan datanya akan menggunakan *smart contract*. Saat ini, seluruh atribut dari data ditulis dengan tipe data *string*. Berikut adalah format data untuk penyimpanan akta kelahiran.

```
{
     “nik”: “1234567890”,
     "birthRegistrationNumber": "BRN123456",
     "fullName": "John Doe",
     "birthPlace": "City Hospital",
     "birthDate": "2024-06-12",
     "gender": "Male",
     "fatherName": "James Doe",
     "motherName": "Jane Doe"
}
```

Berikut adalah contoh format data untuk penyimpanan akta perkawinan.

```
{
     nik: "3273172906010005",
     marriageRegistrationNumber: "MRN123456",
     fullName: "Alice Smith",
     birthPlace: "City Hall",
     birthDate: "1990-05-20",
     spouseName: "Bob Johnson"
};
```

Setelah itu, atribut dari setiap data yang sudah tertulis tersebut akan dienkripsi menggunakan AES dengan kunci yang sudah ditetapkan pada sistem terkecuali NIK sebagai *identifier* dari data. Berikut merupakan contoh hasil format data akta kelahiran yang sudah dienkripsi dengan AES.


```
{ "nik": "1234567890",
"birthRegistrationNumber":
"U2FsdGVkX19bL8uZ5MkIjYwh6a0yE/y1xVvQOeZB5KnvjIGyOXvF
8BqZ+Y7Lt6jw",
"fullName":
"U2FsdGVkX1+O9ur8eD2cGb1YYkSzkvFIM5Op5+mBYkI=",
"birthPlace":
"U2FsdGVkX1/n1dE8y5IDbkLeTxxtQU95X0Ouk+xtbLQ=",
"birthDate":
"U2FsdGVkX18BHo0MJg2NCnB0b/4aT3O5XoZ6gZY7q+Q=",
"gender":
"U2FsdGVkX18Pb/wInTrUiwOGF5z12m1O9v2l4EX5yT0=",
"fatherName":
"U2FsdGVkX19I0kOaI5Y8VmRV+ocTZJjM1LOmpFv0yUk=",
"motherName":
"U2FsdGVkX1/BwQIn+mR9Ju9Rscl0ZtMVRwW+E7t4jXs=" }
```

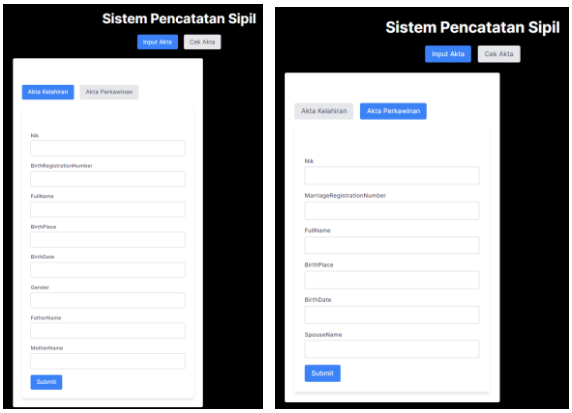

Berikut merupakan contoh hasil format data akta perkawinan yang sudah dienkripsi dengan AES.


```
{
     "nik": "1234567890",
     "marriageRegistrationNumber":
"U2FsdGVkX1+XHQQk0pVLqIl5YkSzkvFIM5Op5+mBYkI=",
     "fullName":
"U2FsdGVkX1+O9ur8eD2cGb1YYkSzkvFIM5Op5+mBYkI=",
     "birthPlace":
"U2FsdGVkX1/n1dE8y5IDbkLeTxxtQU95X0Ouk+xtbLQ=",
     "birthDate":
"U2FsdGVkX18BHo0MJg2NCnB0b/4aT3O5XoZ6gZY7q+Q=",
     "spouseName":
"U2FsdGVkX1+MB2cGFyT1ixb5H5z12m1O9v2l4EX5yT0="
}
```


V. IMPLEMENTASI

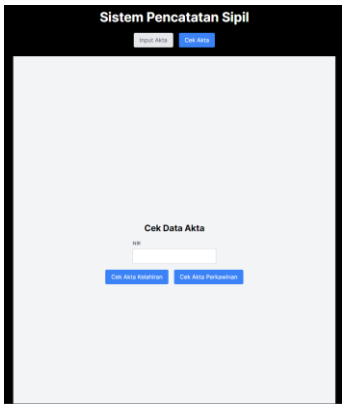
A. *Interface*

Implementasi *user interface* dari sistem pencatatan sipil menggunakan *framework Next.js* yang ditulis dengan bahasa *TypeScript*. Terdapat 2 halaman utama yang meliputi halaman Input Akta untuk memasukkan data akta kelahiran dan akta perkawinan serta Cek Akta untuk mengecek data akta kelahiran dan akta perkawinan yang sudah di-*input* ke *blockchain*. Berikut merupakan halaman input akta.



Gambar 3 *Interface input* data akta

Berikut merupakan halaman Cek Akta yang akan mengecek akta yang sudah tersimpan pada blockchain sesuai dengan NIK yang di-input.



Gambar 4 *Interface* Cek Akta

Jika data tersedia, maka *website* akan meminta data dari *blockchain* dan menampilkannya pada halaman tersebut.

B. *Smart Contract*

Smart contract CivilRegistry akan berjalan di atas *blockchain Sepolia Testnet* dan dirancang untuk menyimpan dan mengelola data pencatatan sipil yang berupa akta kelahiran dan pernikahan. Kontrak ini menggunakan dua struktur data, *BirthCertificate* dan *MarriageCertificate*, untuk mencatat informasi penting seperti nomor registrasi, nama lengkap, tempat dan tanggal lahir, serta detail orang tua dan pasangan.

Data pencatatan disimpan dalam dua *mapping*, *birthCertificates* dan *marriageCertificates*, yang masing-masing mengasosiasikan Nomor Induk Kependudukan (NIK) dengan data kelahiran dan pernikahan. Fungsi *addBirthCertificate* dan *addMarriageCertificate* memungkinkan pengguna untuk menambahkan catatan baru ke dalam sistem, sementara fungsi *getBirthCertificate* dan *getMarriageCertificate* memungkinkan pengguna untuk mengambil data berdasarkan NIK yang diberikan. Dengan menggunakan smart contract ini, pencatatan sipil dapat dikelola secara transparan, aman, dan tidak dapat dimanipulasi, karena semua transaksi dicatat di blockchain.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;
contract CivilRegistry {
    struct BirthCertificate {
        string birthRegistrationNumber;
        string fullName;
        string birthPlace;
        string birthDate;
        string gender;
        string fatherName;
        string motherName;
    }
    struct MarriageCertificate {
        string marriageRegistrationNumber;
        string fullName;
        string birthPlace;
        string birthDate;
        string spouseName;
    }
    mapping(string => BirthCertificate) public
birthCertificates;
    mapping(string => MarriageCertificate) public
marriageCertificates;
    function addBirthCertificate(
        string memory _nik,
        string memory _birthRegistrationNumber,
        string memory _fullName,
        string memory _birthPlace,
        string memory _birthDate,
        string memory _gender,
        string memory _fatherName,
        string memory _motherName
    ) public {
        birthCertificates[_nik] = BirthCertificate(
            _birthRegistrationNumber,
            _fullName,
            _birthPlace,
            _birthDate,
            _gender,
            _fatherName,
            _motherName
        );
    }
    function addMarriageCertificate(
        string memory _nik,
        string memory _marriageRegistrationNumber,
        string memory _fullName,
        string memory _birthPlace,
        string memory _birthDate,
        string memory _spouseName
    ) public {
        marriageCertificates[_nik] =
MarriageCertificate(
            _marriageRegistrationNumber,
            _fullName,
            _birthPlace,
            _birthDate,
            _spouseName
        );
    }
    function getBirthCertificate(string memory _nik)
        public
        view
        returns (BirthCertificate memory)
    {
        return birthCertificates[_nik];
    }
    function getMarriageCertificate(string memory
_nik)
        public
        view
        returns (MarriageCertificate memory)
    {
        return marriageCertificates[_nik];
    }
}
```

C. Modul AES

Kode berikut menggunakan library CryptoJS untuk mengimplementasikan fungsi enkripsi dan dekripsi menggunakan algoritma kriptografi *Advanced Encryption Standard* (AES). Kunci rahasia (SECRET_KEY) saat ini disimpan secara statis dalam kode, yang berarti kunci ini ditetapkan dan tidak berubah.

Fungsi encrypt menerima teks biasa sebagai parameter dan mengembalikan teks terenkripsi dalam bentuk string. Proses

enkripsi dilakukan oleh metode *CryptoJS.AES.encrypt*, yang menggabungkan teks dengan kunci rahasia untuk menghasilkan ciphertext.

Sebaliknya, fungsi decrypt menerima ciphertext sebagai parameter dan mengembalikan teks yang telah didekripsi ke bentuk aslinya. Proses dekripsi dilakukan oleh metode *CryptoJS.AES.decrypt*, yang menggunakan kunci rahasia untuk menguraikan ciphertext kembali menjadi teks biasa yang dapat dibaca, dan mengkonversinya ke format UTF-8 menggunakan CryptoJS.enc.Utf8.

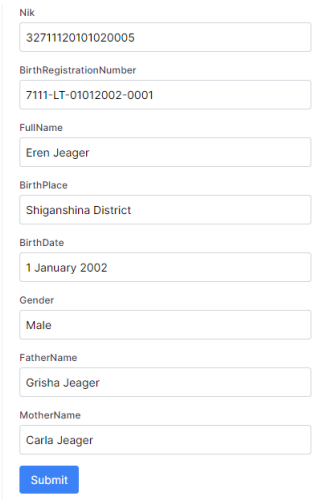
```
import CryptoJS from 'crypto-js';
const SECRET_KEY = 'kunci-rahasia';

export const encrypt = (text: string): string => {
  return                    CryptoJS.AES.encrypt(text,
SECRET_KEY).toString();
};
export const decrypt = (ciphertext: string): string
=> {
  const   bytes   =   CryptoJS.AES.decrypt(ciphertext,
SECRET_KEY);
  return bytes.toString(CryptoJS.enc.Utf8);
};
```

VI. Pengujian dan Diskusi

A. *Input Data*

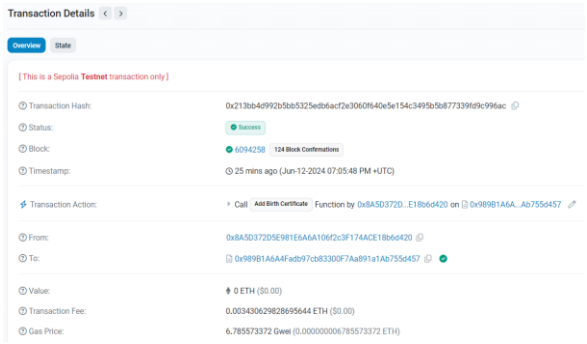
Setelah implementasi berhasil dilakukan, sistem akan diuji untuk membuktikan apakah sistem berhasil menjalankan fungsi yang diinginkan atau tidak. Skenario pengujian dilakukan dengan meng-input data akta kelahiran dan data akta perkawinan melalui halaman Input Akta pada *website*. Kemudian, cek data akan dilakukan pada halaman Cek Akta dengan memasukkan NIK sesuai dengan data yang sudah diinput. Jika data pada Cek Akta sesuai dengan saat Input Akta, maka pengujian sistem sukses.

Berikut merupakan data akta kelahiran yang dicoba untuk di-*input* pada sistem. Jika semua input *field* diisi semua akan diklik *submit*.



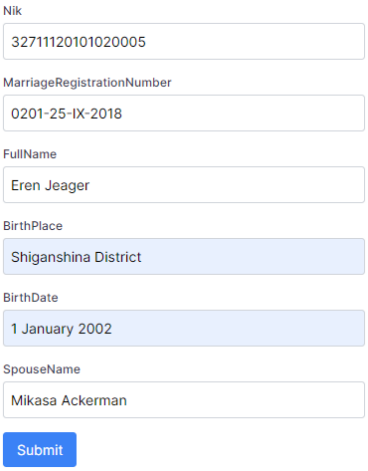
Gambar 5 Input data akta kelahiran

Setelah klik submit dan transaksi berhasil dilakukan, akan dilakukan pengecekan pada etherscan sebagai kakas untuk melihat deployment blockchain untuk memastikan bahwa transaksi memang benar-benar berhasil.



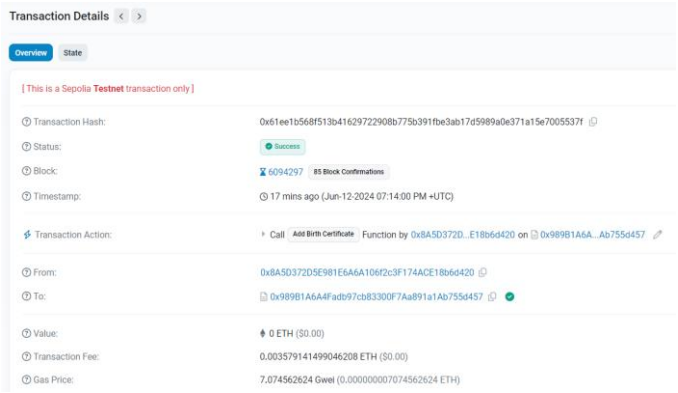
Gambar 6 Transaksi input data akta kelahiran

Berdasarkan gambar tersebut, transaksi input data akta kelahiran berhasil dilakukan dan tercatat dengan status *success*. Setelah itu, dicoba untuk melakukan input data akta perkawinan.



Gambar 7 Input data akta perkawinan

Setelah transaksi submit akta perkawinan berhasil dilakukan, dilakukan pengecekan kembali pada etherscan.



Gambar 8 Transaksi input data akta perkawinan

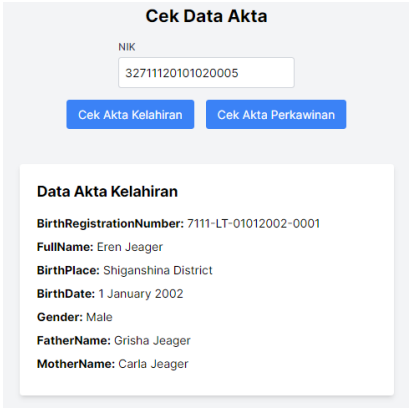
Transaksi input data akta perkawinan berhasil dan tercatat dalam status sukses.

B. *View Data*

Selanjutnya, dilakukan pengecekan pada deployment *blockchain* apakah enkripsi AES berhasil dilakukan. Pengecekan dilakukan dengan melakukan *query* pada etherscan sesuai dengan NIK yang sudah di-*input* pada subbab sebelumnya.

Gambar 9 Data akta kelahiran pada *blockchain*

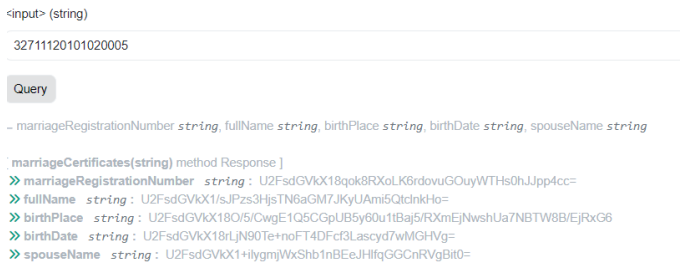
Berdasarkan gambar tersebut, query berhasil dilakukan dan tampak bahwa seluruh atribut dari data akta kelahiran tersimpan dengan *string ciphertext* yang berupa kumpulan karakter secara acak tanpa dapat dibaca. Setelah itu, dilakukan pengecekan pada Cek Data Akta Kelahiran apakah dekripsi berhasil dilakukan dan website dapat melakukan *fetching* data dari *blockchain*.



Gambar 10 Pengujian cek data akta kelahiran

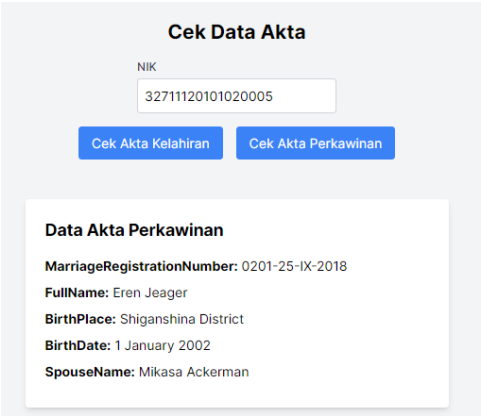
Berdasarkan gambar tersebut, seluruh atribut yang berhasil di-*fetch* sesuai dengan data yang sebelumnya di-*input*. Hal ini menandakan bahwa fungsi penyimpanan data akta kelahiran berhasil.

Setelah mengecek data akta kelahiran, perlu juga dilakukan pengecekan terhadap data akta perkawinan. Berikut merupakan hasil *query* pada etherscan berdasarkan NIK yang sudah di-input sebelumnya.



Gambar 11 Data akta perkawinan pada *blockchain*

Berdasarkan hasil tangkapan layar tersebut, enkripsi AES berhasil dilakukan dengan bukti bahwa data akta perkawinan tersimpan dengan karakter yang acak dan tidak dapat dibaca. Kemudian, dilakukan pengecekan pada Cek Akta Perkawinan pada *website*.



Gambar 12 Pengujian cek data akta perkawinan

Berdasarkan tangkapan layar tersebut, fetching data dan dekripsi ciphertext dari *blockchain* berhasil dengan data akta perkawinan yang sesuai dengan data yang di-input sebelumnya.

Dengan demikian, pengembangan sistem pencatatan sipil berbasis blockchain dan algoritma enkripsi AES berhasil dilakukan. Hal ini terbukti dengan hasil pengujian yang sukses pada fungsionalitas input dan cek data akta kelahiran dan perkawinan.

VII. KESIMPUILAN DAN SARAN

A. Kesimpulan

Berdasarkan implementasi dan pengujian yang dilakukan, disimpulkan bahwa sistem pencatatan sipil berbasis blockchain dengan implementasi algoritma kriptografi AES berhasil dilakukan untuk meningkatkan keamanan dan keandalan data pencatatan sipil, khususnya akta kelahiran dan akta perkawinan. Pengujian menunjukkan bahwa sistem dapat mengenkripsi data dengan aman, menyimpannya di *blockchain Sepolia Testnet*, dan mendekripsinya kembali untuk ditampilkan secara akurat. Kombinasi teknologi *blockchain* dan AES terbukti efektif dalam menciptakan sistem yang transparan, tahan manipulasi, dan melindungi data pribadi pengguna.

B. Saran

Untuk meningkatkan sistem ini lebih lanjut, disarankan untuk menambahkan fitur pencatatan kejadian sipil lainnya, memperbaiki mekanisme penyimpanan kunci enkripsi dengan menggunakan layanan manajemen kunci yang lebih aman, dan mempertimbangkan implementasi di jaringan mainnet Ethereum untuk penggunaan yang sebenarnya. Selain itu, peningkatan skalabilitas sistem untuk menangani volume data yang lebih besar dan melakukan audit keamanan menyeluruh terhadap smart contract dan keseluruhan sistem sangat dianjurkan sebelum implementasi luas untuk memastikan sistem ini dapat berfungsi dengan efisien dan aman dalam jangka panjang.

VIDEO LINK AT YOUTUBE

Berikut merupakan video link youtube untuk demo system pencatatan sipil pada blockchain dengan algoritmak kriptografi AES.

http://bit.ly/CivilRegistry_Makalah_Kriptografi

SOURCE CODE

Berikut merupakan video link github untuk demo system pencatatan sipil pada blockchain dengan algoritma kriptografi AES.

https://github.com/hilmibaskara/civilregistry-blockchain

DEPLOYMENT

Berikut merupakan link deployment untuk website sistem pencatatan sipil.

https://civilregistry-blockchain.vercel.app/

Berikut merupakan link deployment untuk smart contract dari blockchain sistem pencatatan sipil.

https://sepolia.etherscan.io/address/0x989B1A6A4Fadb97cb83300F7Aa891a1Ab755d457

REFERENCES

[1] Informatika, D.K. dan (no date) Artikel: Website Resmi Dinas kependudukan Dan Pencatatan Sipil Pemerintah Kabupaten Badung, Website Portal Resmi Dinas Komunikasi dan Informatika Pemerintah Kabupaten Badung. Available at: https://disdukcapil.badungkab.go.id/artikel/17825-pengertian-catatan-sipil (Accessed: 12 June 2024).

[2] Intan (2023) Kenali Manfaat Dan Pentingnya Tertib Administrasi kependudukan, Disdukcapil Kota Surabaya - Website Resmi Dinas Kependudukan dan Pencatatan Sipil Kota Surabaya. Available at: https://disdukcapil.surabaya.go.id/2023/06/12/kenali-manfaat-dan-pentingnya-tertib-administrasi-kependudukan/ (Accessed: 12 June 2024).

[3] The Jakarta Post (no date) Home Ministry plays down alleged leak of civil registry data - Fri, July 21, 2023, The Jakarta Post. Available at: https://www.thejakartapost.com/paper/2023/07/21/home-ministry-plays-down-alleged-leak-of-civil-registry-data.html (Accessed: 12 June 2024).

[4] Hidayat, E.S. (no date) Analisis implementasi Kebijakan Administrasi Kependudukan Pada Dinas kependudukan Dan Pencatatan Sipil Kabupaten Garut, Dinamika : Jurnal Ilmiah Ilmu Administrasi Negara. Available at: https://jurnal.unigal.ac.id/dinamika/article/view/1741/1404 (Accessed: 12 June 2024).

[5] Narayanan, A. (2016) Bitcoin and cryptocurrency technologies: A comprehensive introduction. Princeton, NJ: Princeton University Press.

[6] Jiantono, A.C. (2023) Mengenal Smart contract Dalam Blockchain, School of Information Systems. Available at: https://sis.binus.ac.id/2023/05/02/mengenal-smart-contract-dalam-blockchain/ (Accessed: 12 June 2024).

[7] (No date) Advanced encryption standard (AES). Available at: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf (Accessed: 12 June 2024).

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, 12 Juni 2024

Hilmi Baskara Radanto (18221072)