ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Perancangan Sistem Verifikasi Sertifikat Digital Berbasis Blockchain dengan Smart Contract Ethereum dan IPFS

Ryan Bagus Bimantoro[1], Henni Endah Wahanani[2*], Muhammad Muharrom Al Haromainy[3]

[1,3] Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]22081010270@student.upnjatim.ac.id
[3]muhammad.muharrom.if@upnjatim.ac.id

[2] Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
**Corresponding author email*: henniendah.if@upnjatim.ac.id

***Abstrak*— Penelitian ini bertujuan untuk merancang prototipe sistem verifikasi sertifikat digital berbasis *blockchain* dengan memanfaatkan teknologi *smart contract* pada jaringan Ethereum dan penyimpanan terdistribusi *InterPlanetary File System* guna mengatasi risiko pemalsuan sertifikat serta keterbatasan sistem verifikasi konvensional yang masih terpusat dan rentan terhadap manipulasi data. Metode yang digunakan meliputi analisis kebutuhan sistem, perancangan alur proses, arsitektur sistem, *smart contract*, serta antarmuka pengguna berbasis web. Hasil penelitian berupa rancangan sistem yang mencakup mekanisme penerbitan sertifikat oleh *issuer*, penyimpanan dokumen pada *InterPlanetary File System*, serta pencatatan metadata dan *hash* pada *blockchain* untuk menjamin integritas data. Sistem juga dilengkapi dengan fitur verifikasi berbasis perbandingan *hash* secara transparan serta rancangan antarmuka dalam bentuk *wireframe*. Dengan demikian, rancangan sistem ini diharapkan dapat menjadi dasar pengembangan sistem verifikasi sertifikat digital yang lebih aman, transparan, dan tahan terhadap pemalsuan di masa mendatang.**

Kata Kunci— **Blockchain, smart contract, Ethereum, IPFS, Verifikasi Sertifikat, Perancangan sistem.**

I. PENDAHULUAN

Sertifikat digital merupakan dokumen elektronik yang digunakan sebagai bukti resmi atas pencapaian, kompetensi, atau partisipasi seseorang dalam suatu kegiatan, seperti pendidikan dan pelatihan. Penggunaan sertifikat digital terus meningkat seiring perkembangan teknologi informasi yang mendukung distribusi dan penyimpanan dokumen secara cepat dan efisien. Namun, pengelolaan sertifikat digital masih menghadapi berbagai permasalahan, terutama terkait keaslian dan validitas dokumen. Sistem penyimpanan yang masih terpusat berpotensi menimbulkan manipulasi data, pemalsuan dokumen, serta kesulitan dalam proses verifikasi oleh pihak ketiga. Oleh karena itu, diperlukan rancangan sistem verifikasi yang lebih andal untuk menjamin integritas dan keaslian sertifikat digital [1].

Permasalahan tersebut juga terjadi dalam berbagai kasus nyata. Salah satu contohnya adalah pemalsuan sertifikat pelatihan Keselamatan dan Kesehatan Kerja (K3) yang berhasil diungkap oleh aparat kepolisian bersama Kementerian Ketenagakerjaan. Dalam kasus tersebut, pelaku memproduksi dan mendistribusikan sertifikat palsu untuk memenuhi persyaratan pekerjaan dengan memanfaatkan lemahnya mekanisme verifikasi. Hal ini menunjukkan bahwa sistem pengelolaan sertifikat yang tidak memiliki mekanisme validasi yang kuat dapat membuka peluang terjadinya pemalsuan dokumen dan menurunkan tingkat kepercayaan terhadap keaslian sertifikat [2].

Untuk mengatasi permasalahan tersebut, diperlukan rancangan sistem yang mampu mendukung keamanan, integritas, dan transparansi data sertifikat digital. Salah satu teknologi yang dapat digunakan adalah *blockchain*, yaitu sistem buku besar terdistribusi yang memungkinkan pencatatan data secara permanen dan sulit diubah. Data dalam *blockchain* diamankan melalui mekanisme kriptografi sehingga setiap perubahan dapat terdeteksi. Karakteristik ini menjadikan *blockchain* sebagai teknologi yang potensial untuk mendukung proses verifikasi sertifikat digital. Selain itu, penggunaan *blockchain* memungkinkan proses verifikasi dilakukan secara transparan tanpa bergantung sepenuhnya pada satu otoritas pusat [3].

Beberapa penelitian sebelumnya telah memanfaatkan teknologi *blockchain* dalam pengelolaan sertifikat dan dokumen akademik digital. Penggunaan aplikasi terdesentralisasi dalam manajemen sertifikat digital memungkinkan sertifikat memiliki identitas unik, tercatat secara transparan, dan dapat diverifikasi oleh pihak yang berkepentingan. Selain itu, integrasi *blockchain* dengan sistem penyimpanan terdistribusi seperti IPFS dapat meningkatkan efisiensi penyimpanan serta menekan biaya transaksi pada implementasi *smart contract* [4].

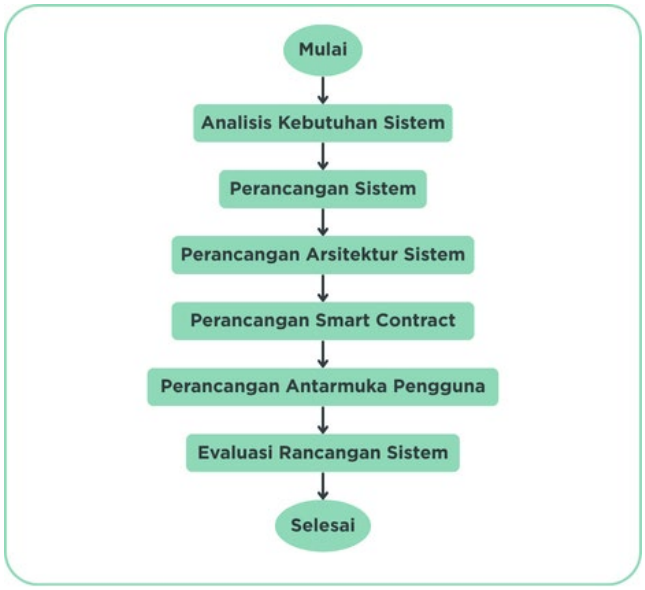
Berdasarkan permasalahan dan penelitian sebelumnya, penelitian ini bertujuan untuk merancang sistem verifikasi sertifikat digital berbasis *blockchain* dengan memanfaatkan *smart contract* Ethereum dan IPFS. Sistem yang dirancang berfokus pada proses penerbitan dan verifikasi sertifikat digital melalui pencatatan metadata dan *hash* dokumen pada *blockchain*, serta penyimpanan dokumen atau data pendukung menggunakan IPFS. Perancangan ini juga sejalan dengan penelitian terkait sistem pencatatan ijazah menggunakan *smart contract*, NFT, jaringan Polygon, dan IPFS yang menunjukkan bahwa teknologi *blockchain* dapat mendukung pencatatan dokumen akademik secara lebih aman, mudah dilacak, dan efisien dari sisi biaya transaksi. Dengan adanya rancangan ini, sistem diharapkan dapat menjadi dasar pengembangan aplikasi verifikasi sertifikat digital yang mampu meningkatkan keamanan, integritas, transparansi, dan keaslian dokumen secara lebih efektif [5].

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

II. METODOLOGI PENELITIAN

A. Metode Penelitian

Metode penelitian yang digunakan adalah metode perancangan sistem. Metode ini dipilih karena penelitian berfokus pada penyusunan rancangan sistem verifikasi sertifikat digital berbasis *blockchain*, bukan pada implementasi penuh. Tahapan penelitian meliputi analisis kebutuhan sistem, perancangan sistem, perancangan arsitektur, perancangan *smart contract*, perancangan antarmuka pengguna, dan evaluasi rancangan sistem.



Gbr. 1 Diagram Alur Metode Penelitian

Gbr. 1 menunjukkan alur penelitian yang digunakan dalam perancangan sistem. Tahapan dimulai dari analisis kebutuhan untuk mengidentifikasi aktor dan kebutuhan sistem, kemudian dilanjutkan dengan perancangan alur proses, arsitektur sistem, *smart contract*, serta antarmuka pengguna dalam bentuk *wireframe*. Tahap terakhir adalah evaluasi rancangan untuk menilai kesesuaian sistem dengan kebutuhan yang telah ditentukan.

B. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi aktor, fungsi utama, serta kebutuhan pendukung dalam perancangan sistem verifikasi sertifikat digital berbasis *blockchain*. Pada rancangan ini terdapat tiga aktor utama, yaitu *issuer*, *student*, dan *verifier*.

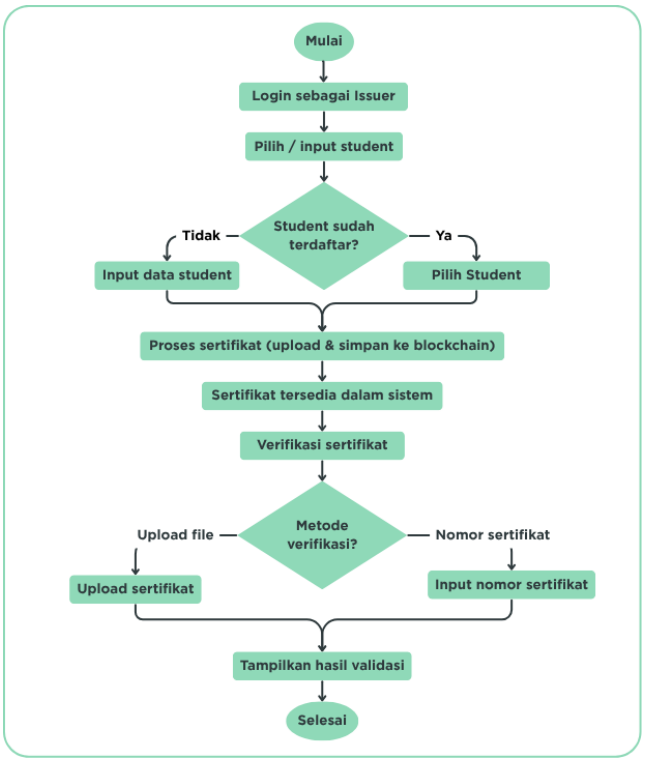
TABEL I
AKTOR SISTEM

No	Aktor	Peran
1	*Issuer*	Pihak yang menerbitkan sertifikat digital, mengelola data *student*, mengunggah dokumen, dan mencatat data sertifikat ke dalam sistem.
2	*Student*	Pihak pemilik sertifikat digital yang datanya digunakan dalam proses penerbitan sertifikat.
3	*Verifier*	Pihak yang melakukan verifikasi keaslian sertifikat melalui unggah dokumen atau input nomor sertifikat.

Berdasarkan Tabel I, *issuer* berperan sebagai pihak penerbit sertifikat, *student* sebagai pemilik sertifikat, dan *verifier* sebagai pihak yang memeriksa keaslian sertifikat. Kebutuhan fungsional sistem meliputi *login* bagi *issuer*, pengelolaan data *student*, pengisian metadata sertifikat, unggah dokumen, pembuatan nilai *hash*, penyimpanan dokumen pada IPFS, serta pencatatan metadata, CID, dan *hash* ke dalam *blockchain* melalui *smart contract*. Sistem juga dirancang untuk mendukung verifikasi melalui unggah file atau input nomor sertifikat, kemudian menampilkan status valid atau tidak valid. Kebutuhan non-fungsional sistem mencakup keamanan, integritas, transparansi, kemudahan akses, dan penyimpanan terdistribusi melalui IPFS.

C. Perancangan Sistem

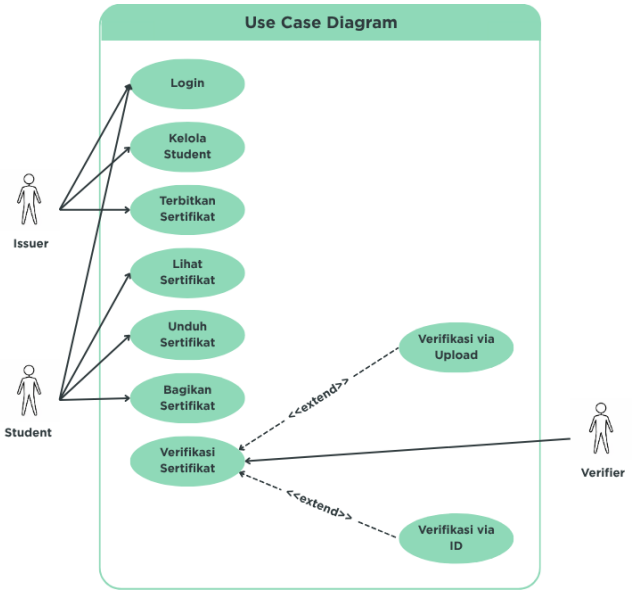
Perancangan sistem dilakukan untuk menggambarkan alur kerja utama dalam sistem verifikasi sertifikat digital berbasis *blockchain*. Pada tahap ini, sistem dirancang untuk mengakomodasi proses penerbitan sertifikat oleh *issuer*, pengelolaan data *student* sebagai pemilik sertifikat, serta proses verifikasi sertifikat oleh *verifier*. Perancangan sistem ini berfokus pada alur penerbitan dan verifikasi agar setiap proses dapat berjalan secara terstruktur, aman, dan transparan.



Gbr. 2 Diagram Alur Sistem Sertifikat Digital

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Berdasarkan Gbr. 2, alur sistem dimulai dari proses *login* oleh *issuer*, kemudian dilanjutkan dengan pemilihan atau penginputan data *student*. Jika data *student* belum terdaftar, maka *issuer* perlu menginput data *student* terlebih dahulu, sedangkan jika data sudah tersedia, *issuer* dapat langsung memilih data *student*. Setelah itu, sistem memproses sertifikat melalui tahapan unggah dokumen, pembuatan nilai *hash*, penyimpanan dokumen pada IPFS, serta pencatatan metadata, CID, dan *hash* ke dalam *blockchain* melalui *smart contract*. Sertifikat yang telah diproses akan tersedia dalam sistem dan dapat diverifikasi oleh *verifier* melalui dua metode, yaitu unggah file sertifikat atau input nomor sertifikat. Pada metode unggah file, sistem mencocokkan nilai *hash* file dengan data yang tercatat pada *blockchain*, sedangkan pada metode nomor sertifikat, sistem mencari data berdasarkan nomor sertifikat yang dimasukkan. Hasil dari proses tersebut kemudian ditampilkan dalam bentuk status validasi untuk menunjukkan apakah sertifikat dinyatakan valid atau tidak valid.

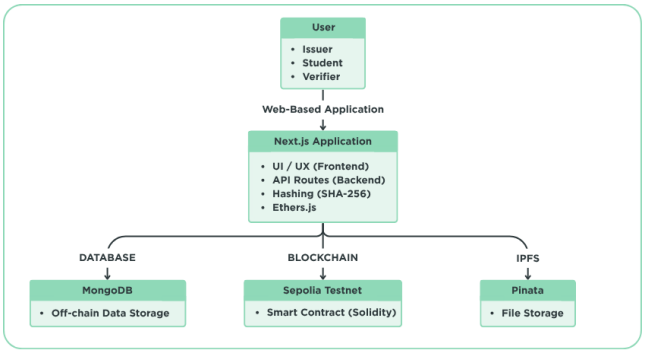


Gbr. 3 Diagram *Use Case* Sistem

Gbr. 3 menunjukkan hubungan antara aktor dan fungsi utama pada sistem verifikasi sertifikat digital. *Issuer* memiliki akses untuk melakukan *login*, mengelola data *student*, dan menerbitkan sertifikat digital. *Student* berperan sebagai pemilik sertifikat yang dapat melihat, mengunduh, membagikan, serta melakukan verifikasi sertifikat secara mandiri. Sementara itu, *verifier* berperan sebagai pihak yang melakukan verifikasi keaslian sertifikat. Proses verifikasi sertifikat dapat dilakukan melalui dua metode, yaitu verifikasi melalui unggah file dan verifikasi melalui ID atau nomor sertifikat. Dengan rancangan *use case* ini, setiap aktor memiliki fungsi dan batasan akses yang jelas sesuai dengan perannya dalam sistem yang dirancang, sehingga interaksi antaraktor dan keterkaitan fungsi utama sistem dapat dipahami secara lebih terstruktur dalam proses perancangan sistem.

D. Perancangan Arsitektur Sistem

Perancangan arsitektur sistem dilakukan untuk menggambarkan struktur dan hubungan antar komponen dalam sistem verifikasi sertifikat digital berbasis *blockchain*. Arsitektur ini menunjukkan bagaimana pengguna, aplikasi web, *database*, IPFS, dan *smart contract* saling terintegrasi dalam mendukung proses penerbitan dan verifikasi sertifikat digital. Dengan arsitektur yang terstruktur, sistem diharapkan dapat berjalan secara efisien, aman, dan sesuai dengan kebutuhan yang telah ditentukan.



Gbr. 4 Arsitektur Sistem Verifikasi Sertifikat Digital

Berdasarkan Gbr. 4, pengguna sistem terdiri dari *issuer*, *student*, dan *verifier* yang berinteraksi melalui aplikasi berbasis web. Aplikasi web dirancang menggunakan Next.js yang mencakup UI/UX sebagai *frontend*, API Routes sebagai *backend*, proses *hashing* menggunakan SHA-256, serta Ethers.js sebagai penghubung antara aplikasi dan jaringan *blockchain*. File sertifikat disimpan pada IPFS melalui Pinata, sedangkan metadata penting seperti nilai *hash*, CID, dan informasi sertifikat dicatat ke dalam *smart contract* berbasis Solidity pada jaringan Sepolia Testnet. Selain itu, MongoDB digunakan untuk menyimpan data *off-chain*, seperti data pengguna, data *student*, dan metadata pendukung. Pemisahan penyimpanan ini bertujuan agar file, data pendukung, dan data verifikasi dapat dikelola sesuai fungsinya masing-masing.

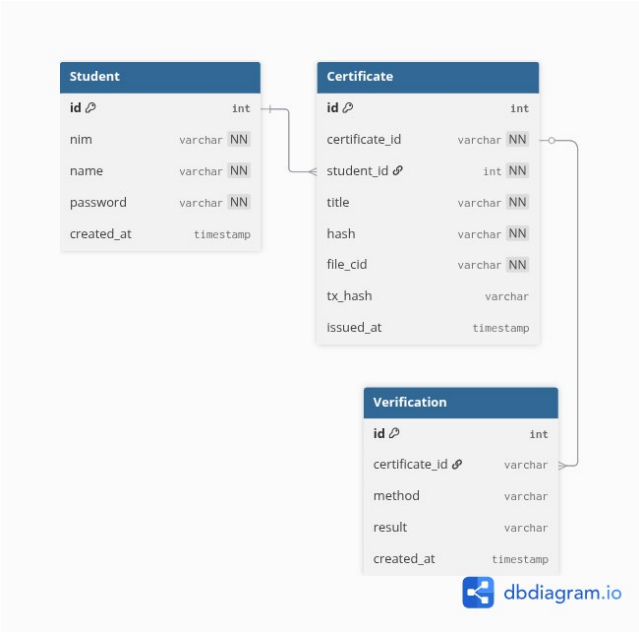
TABEL II
KOMPONEN DAN *TOOLS* SISTEM

No	Komponen	Teknologi / *Tools*	Fungsi
1	*Frontend*	Next.js	Menyediakan antarmuka pengguna.
2	*Backend*	API Routes	Mengelola logika sistem dan permintaan data.
3	*Hashing*	SHA-256	Menghasilkan nilai *hash* dari file sertifikat.
4	*Blockchain*	Sepolia Testnet	Menjadi jaringan Ethereum untuk pencatatan data sertifikat.

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

5	*Smart Contract*	Solidity	Mengatur pencatatan dan verifikasi metadata, CID, serta *hash*.
6	Penyimpanan file	IPFS/Pinata	Menyimpan file sertifikat secara terdistribusi.
7	*Database*	MongoDB	Menyimpan data *off-chain* seperti data pengguna, data *student*, dan metadata pendukung.
8	*Integration Layer*	Ethers.js	Menghubungkan aplikasi web dengan *smart contract* pada jaringan *blockchain*.

Berdasarkan Tabel II, setiap komponen memiliki fungsi yang saling mendukung dalam proses penerbitan dan verifikasi sertifikat digital. Aplikasi web berperan sebagai media interaksi pengguna, IPFS digunakan untuk menyimpan file sertifikat, *smart contract* digunakan untuk mencatat data penting pada *blockchain*, sedangkan MongoDB digunakan sebagai penyimpanan data pendukung yang tidak perlu dicatat langsung pada jaringan *blockchain*.



Gbr. 5 *Entity Relationship Diagram* Sistem

Gbr. 5 menunjukkan rancangan *Entity Relationship Diagram* (ERD) yang digunakan untuk mengelola data *off-chain* pada sistem verifikasi sertifikat digital. Rancangan basis data terdiri dari tiga entitas utama, yaitu *Student*, *Certificate*, dan *Verification*. Entitas *Student* digunakan untuk menyimpan data pemilik sertifikat, seperti NIM, nama, dan *password*. Entitas *Certificate* digunakan untuk menyimpan metadata sertifikat, seperti *certificate_id*, judul sertifikat, nilai *hash*, CID file dari IPFS, *transaction hash*, serta waktu penerbitan sertifikat. Sementara itu, entitas *Verification* digunakan untuk mencatat riwayat proses verifikasi, baik melalui metode unggah file maupun input nomor sertifikat, beserta hasil verifikasi yang diperoleh. Relasi pada ERD menunjukkan bahwa satu *student* dapat memiliki banyak sertifikat, sedangkan setiap sertifikat dapat memiliki banyak riwayat verifikasi. Dengan rancangan ini, MongoDB digunakan untuk menyimpan data pendukung secara terstruktur, sementara validasi utama keaslian sertifikat tetap mengacu pada nilai *hash* dan data yang tercatat pada *blockchain*.

E. Perancangan Smart Contract

Perancangan *smart contract* dilakukan untuk mengatur proses pencatatan dan verifikasi data sertifikat digital pada jaringan *blockchain*. *Smart contract* dirancang menggunakan bahasa Solidity dan berfungsi sebagai komponen utama yang menyimpan data penting sertifikat, seperti *hash* dokumen, ID sertifikat, CID dari IPFS, serta informasi waktu penerbitan. Pada rancangan ini, *smart contract* juga menerapkan mekanisme pembatasan akses agar proses penerbitan sertifikat hanya dapat dilakukan oleh pihak yang berwenang. Dengan demikian, *smart contract* tidak hanya berperan sebagai media pencatatan data, tetapi juga sebagai mekanisme validasi untuk menjaga integritas dan keaslian sertifikat digital.

```
1  address public owner;
2
3  constructor() {
4      owner = msg.sender;
5  }
6
7  modifier onlyOwner() {
8      require(msg.sender == owner, "Not authorized");
9      _;
10 }
11
```

Gbr. 6 Struktur Dasar dan *Access Control*

Gbr. 6 menunjukkan rancangan struktur dasar *smart contract* yang terdiri dari deklarasi variabel *owner*, fungsi *constructor*, dan *modifier onlyOwner*. Variabel *owner* digunakan untuk menyimpan alamat akun yang memiliki kewenangan utama dalam pengelolaan *smart contract*. Fungsi *constructor* dijalankan saat *smart contract* pertama kali dibuat, dengan menetapkan *msg.sender* sebagai pemilik kontrak. Sementara itu, *modifier onlyOwner* digunakan sebagai mekanisme *access control* untuk membatasi fungsi tertentu agar hanya dapat dijalankan oleh pemilik kontrak. Dengan mekanisme ini, proses penting seperti penerbitan sertifikat dapat dibatasi hanya untuk pihak yang berwenang, sehingga pengelolaan data sertifikat menjadi lebih aman dan risiko akses tidak sah terhadap fungsi utama pada *smart contract* dapat diminimalkan, terutama pada proses pencatatan data sertifikat.

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

```
struct Certificate {
    bool exists;
    string certificateId;
    uint256 timestamp;
}

mapping(bytes32 => Certificate) private certificates;
mapping(string => bytes32) private certificateIdToHash;

```

Gbr. 7 Struktur Data dan Penyimpanan Sertifikat

Gbr. 7 menunjukkan rancangan struktur data pada *smart contract* yang digunakan untuk menyimpan informasi sertifikat digital. Struktur *Certificate* terdiri dari atribut *exists*, *certificateId*, dan *timestamp*. Atribut *exists* digunakan untuk menandai apakah data sertifikat telah tercatat, *certificateId* digunakan sebagai identitas unik sertifikat, sedangkan *timestamp* digunakan untuk menyimpan waktu pencatatan data. Selain itu, digunakan *mapping certificates* untuk menghubungkan nilai *hash* sertifikat dengan data sertifikat, serta *mapping certificateIdToHash* untuk menghubungkan ID sertifikat dengan nilai *hash*. Dengan rancangan ini, proses penyimpanan dan pencarian data sertifikat pada *smart contract* dapat dilakukan secara lebih efisien dan terstruktur.

```
event CertificateIssued(
    bytes32 indexed hash,
    string certificateId,
    uint256 timestamp
);

```

Gbr. 8 *Event* Penerbitan Sertifikat

Gbr. 8 menunjukkan rancangan *event CertificateIssued* yang digunakan untuk mencatat aktivitas penerbitan sertifikat pada *smart contract*. *Event* ini memuat tiga parameter utama, yaitu *hash*, *certificateId*, dan *timestamp*. Parameter *hash* digunakan sebagai identitas kriptografis dari dokumen sertifikat, *certificateId* digunakan sebagai nomor atau ID unik sertifikat, sedangkan *timestamp* digunakan untuk mencatat waktu penerbitan sertifikat. Penggunaan *event* ini bertujuan agar setiap proses penerbitan sertifikat dapat terekam pada *blockchain* dan dapat dipantau secara transparan sebagai bukti bahwa sertifikat telah diterbitkan melalui sistem, sehingga riwayat penerbitan dapat ditelusuri kembali apabila diperlukan dalam proses verifikasi maupun pemeriksaan data sertifikat oleh pihak yang berkepentingan. Pencatatan ketiga parameter tersebut juga memberikan informasi yang terstruktur untuk menghubungkan data sertifikat dengan transaksi penerbitannya, sehingga proses penelusuran dapat dilakukan berdasarkan identitas sertifikat, nilai hash, maupun waktu penerbitannya.

```
function issueCertificate(
    bytes32 _hash, string memory _certificateId
) public onlyOwner {

    require(!certificates[_hash].exists, "Certificate already exists");
    require(certificateIdToHash[_certificateId] == bytes32(0), "ID already used");

    certificates[_hash] = Certificate({
        exists: true,
        certificateId:  _certificateId,
        timestamp: block.timestamp
    });

    certificateIdToHash[_certificateId] = _hash;

    emit CertificateIssued(_hash, _certificateId, block.timestamp);
}
```

Gbr. 9 Fungsi Penerbitan Sertifikat

Gbr. 9 menunjukkan rancangan fungsi *issueCertificate* yang digunakan untuk mencatat sertifikat baru ke dalam *smart contract*. Fungsi ini menerima dua parameter utama, yaitu *_hash* sebagai nilai *hash* dari dokumen sertifikat dan *_certificateId* sebagai identitas unik sertifikat. Pada fungsi ini terdapat proses validasi menggunakan *require* untuk memastikan bahwa nilai *hash* belum pernah tercatat dan ID sertifikat belum pernah digunakan sebelumnya. Jika data belum terdaftar, sistem akan menyimpan informasi sertifikat ke dalam struktur *Certificate* yang berisi status keberadaan data, ID sertifikat, dan waktu pencatatan berdasarkan *block.timestamp*. Setelah data berhasil disimpan, fungsi akan menjalankan *event CertificateIssued* sebagai tanda bahwa sertifikat telah diterbitkan dan tercatat pada *blockchain*.

```
function verifyCertificate(bytes32 _hash) public view returns (bool) {
    return certificates[_hash].exists;
}

function verifyById(string memory  certificateId) public view returns (bool) {
    bytes32 hash = certificateIdToHash[_certificateId];
    return certificates[hash].exists;
}

```

Gbr. 10 Fungsi Verifikasi Sertifikat

Gbr. 10 menunjukkan rancangan fungsi verifikasi sertifikat pada *smart contract* yang terdiri dari fungsi *verifyCertificate* dan *verifyById*. Fungsi *verifyCertificate* digunakan untuk memverifikasi sertifikat berdasarkan nilai *hash* dokumen yang dimasukkan, kemudian mengembalikan nilai *true* apabila data sertifikat ditemukan pada *mapping certificates*. Sementara itu, fungsi *verifyById* digunakan untuk memverifikasi sertifikat berdasarkan ID sertifikat dengan mencari nilai *hash* melalui *mapping certificateIdToHash*, lalu memeriksa keberadaan data pada *mapping certificates*. Dengan dua fungsi ini, proses verifikasi sertifikat dapat dilakukan melalui dua metode, yaitu berdasarkan *hash* dokumen maupun berdasarkan ID sertifikat.

```
function getCertificate(string memory _certificateId) public view returns (bytes32, uint256) {
    bytes32 hash = certificateIdToHash[_certificateId];
    require(certificates[hash].exists, "Certificate not found");

    return (hash, certificates[hash].timestamp);
}

```

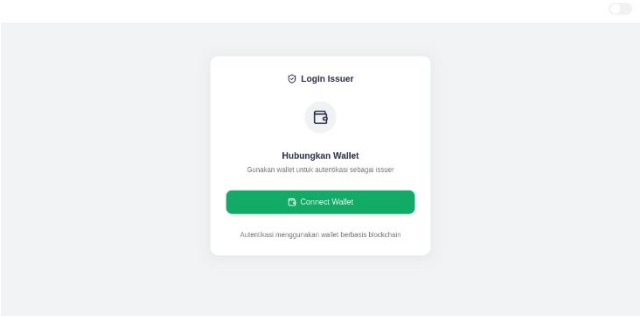
Gbr. 11 Fungsi Pengambilan Data Sertifikat

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Gbr. 11 menunjukkan rancangan fungsi *getCertificate* yang digunakan untuk mengambil data sertifikat berdasarkan ID sertifikat yang dimasukkan. Fungsi ini menerima parameter *_certificateId*, kemudian mencari nilai *hash* yang terhubung melalui *mapping certificateIdToHash*. Setelah itu, fungsi melakukan validasi menggunakan *require* untuk memastikan bahwa data sertifikat benar-benar tersedia pada *mapping certificates*. Jika data ditemukan, fungsi akan mengembalikan nilai *hash* dan *timestamp* sertifikat. Dengan fungsi ini, sistem dapat menampilkan informasi dasar sertifikat yang telah tercatat pada *smart contract* dan mendukung proses pengecekan data sertifikat secara lebih transparan.

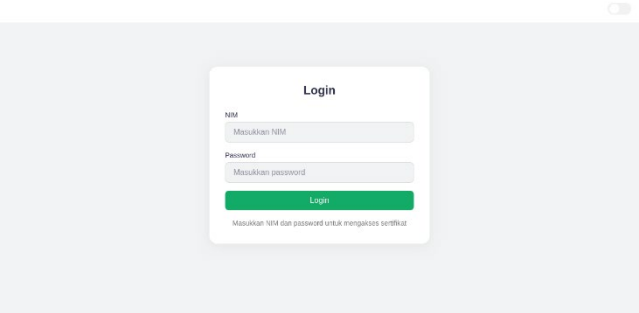
F. Perancangan Antarmuka Pengguna

Perancangan antarmuka pengguna dilakukan untuk menggambarkan tampilan dan alur interaksi pengguna dengan sistem verifikasi sertifikat digital berbasis *blockchain*. Antarmuka dirancang agar setiap aktor dapat mengakses fitur sesuai dengan perannya, mulai dari *issuer* sebagai penerbit sertifikat, *student* sebagai pemilik sertifikat, hingga *verifier* sebagai pihak yang melakukan verifikasi. Pada tahap ini, rancangan antarmuka difokuskan pada kemudahan penggunaan, kejelasan alur, serta kesesuaian dengan proses sistem yang telah dirancang sebelumnya.



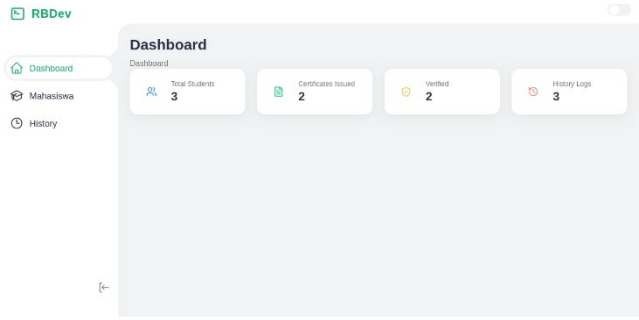
Gbr. 12 Halaman *Login Issuer*

Gbr. 12 menunjukkan rancangan halaman *login* untuk *issuer*. Pada halaman ini, *issuer* melakukan autentikasi dengan menghubungkan *wallet* melalui tombol *Connect Wallet*. Proses *login* dirancang menggunakan mekanisme *Sign-In with Ethereum* (SIWE), yaitu metode autentikasi berbasis *wallet* yang memungkinkan *issuer* masuk ke sistem menggunakan akun MetaMask tanpa menggunakan kombinasi *username* dan *password*. Melalui mekanisme ini, sistem dapat memastikan bahwa pihak yang mengakses fitur penerbitan sertifikat adalah pemilik alamat *wallet* yang sah dan memiliki kewenangan dalam sistem. Selain itu, penggunaan SIWE juga mendukung konsep autentikasi terdesentralisasi karena identitas pengguna diverifikasi melalui kepemilikan alamat *wallet*, bukan hanya melalui data akun yang tersimpan pada basis data. Dengan rancangan ini, akses *issuer* terhadap fitur penting seperti pengelolaan data *student* dan penerbitan sertifikat dapat lebih terkontrol, sehingga hanya pihak yang memiliki otorisasi yang dapat melakukan pencatatan data sertifikat ke dalam sistem.



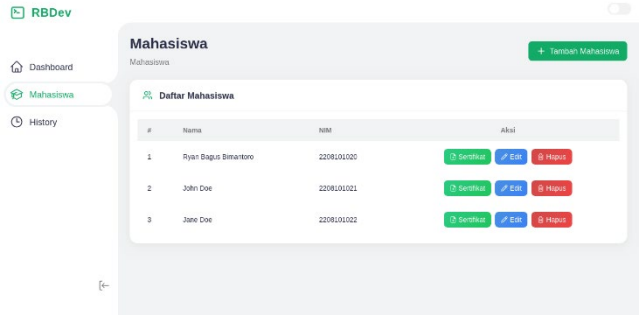
Gbr. 13 Halaman *Login Student*

Gbr. 13 menunjukkan rancangan halaman *login* untuk *student* sebagai pemilik sertifikat digital. Pada halaman ini, *student* melakukan autentikasi menggunakan NIM dan *password* yang sebelumnya telah dibuat atau ditambahkan oleh *issuer* atau admin saat proses pengelolaan data mahasiswa. Setelah berhasil *login*, *student* dapat mengakses sertifikat yang telah diterbitkan, melihat detail sertifikat, mengunduh, atau membagikannya untuk keperluan verifikasi.



Gbr. 14 Halaman *Dashboard Issuer*

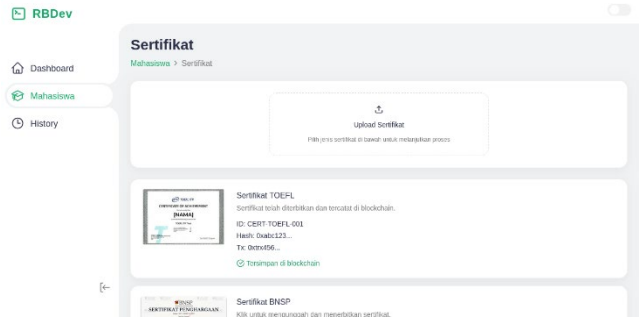
Gbr. 14 menunjukkan rancangan halaman *dashboard* untuk *issuer* setelah berhasil *login*. Halaman ini menampilkan ringkasan informasi seperti jumlah *student*, sertifikat yang diterbitkan, sertifikat yang telah diverifikasi, dan riwayat aktivitas sistem. Pada sisi kiri terdapat menu navigasi untuk mengakses *dashboard*, data mahasiswa, dan *history*, sehingga *issuer* dapat memantau serta mengelola proses penerbitan sertifikat secara lebih mudah dan terarah, termasuk melihat gambaran umum aktivitas sistem.



Gbr. 15 Halaman Kelola Data Mahasiswa

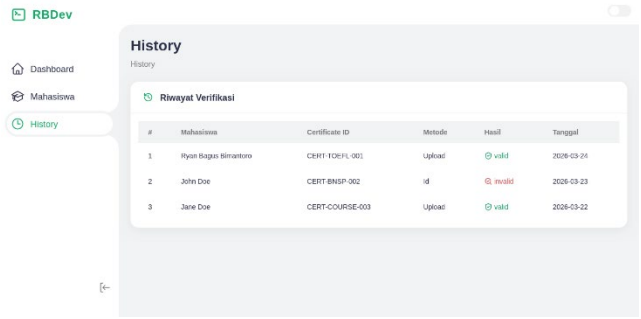
ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Gbr. 15 menunjukkan rancangan halaman kelola data mahasiswa yang digunakan oleh *issuer* untuk melihat dan mengelola data *student*. Halaman ini menampilkan daftar mahasiswa dalam bentuk tabel yang berisi nomor, nama, NIM, dan aksi. Pada bagian aksi, *issuer* dapat mengelola sertifikat mahasiswa, mengubah data, atau menghapus data mahasiswa. Selain itu, terdapat tombol tambah mahasiswa yang digunakan untuk menambahkan data *student* baru ke dalam sistem.



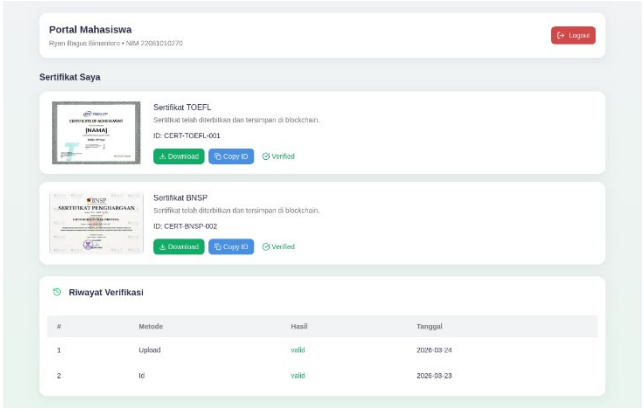
Gbr. 16 Halaman Sertifikat Mahasiswa

Gbr. 16 menunjukkan rancangan halaman sertifikat mahasiswa yang digunakan oleh *issuer* untuk mengelola dan menerbitkan sertifikat digital. Pada halaman ini terdapat fitur *upload sertifikat* yang digunakan untuk mengunggah dokumen sebelum diproses ke dalam sistem. Selain itu, ditampilkan daftar sertifikat yang telah diterbitkan beserta informasi seperti ID sertifikat, nilai *hash*, dan *transaction hash* sebagai bukti pencatatan pada *blockchain*. Halaman ini memudahkan *issuer* dalam memantau status sertifikat serta memastikan bahwa data telah tersimpan dengan baik di dalam sistem.



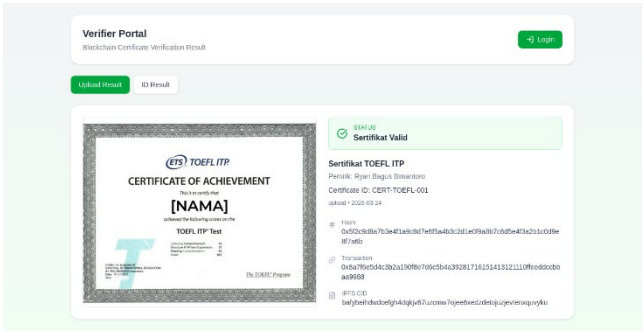
Gbr. 17 Halaman History Verifikasi

Gbr. 17 menunjukkan rancangan halaman *history* yang menampilkan riwayat verifikasi sertifikat dalam sistem. Halaman ini berisi informasi seperti nama mahasiswa, ID sertifikat, metode verifikasi yang digunakan, hasil verifikasi, serta tanggal proses verifikasi. Dengan adanya halaman ini, *issuer* dapat memantau aktivitas verifikasi yang telah dilakukan serta melihat hasil validasi sertifikat secara ringkas dan terstruktur, sehingga memudahkan dalam melakukan penelusuran riwayat verifikasi yang telah terjadi pada sistem serta membantu dalam proses monitoring aktivitas verifikasi secara keseluruhan.



Gbr. 18 Halaman Portal Mahasiswa

Gbr. 18 menunjukkan rancangan halaman portal mahasiswa yang digunakan oleh *student* untuk mengakses sertifikat yang dimiliki. Halaman ini menampilkan informasi identitas mahasiswa, daftar sertifikat yang telah diterbitkan, serta fitur untuk mengunduh sertifikat dan menyalin ID sertifikat. Selain itu, terdapat informasi status verifikasi sertifikat yang menunjukkan bahwa data telah tercatat pada *blockchain*. Pada bagian bawah juga ditampilkan riwayat verifikasi yang pernah dilakukan oleh *student*. Dengan halaman ini, *student* dapat mengelola dan memanfaatkan sertifikat digital secara lebih mudah dan terstruktur.



Gbr. 19 Halaman Verifier Portal

Gbr. 19 menunjukkan rancangan halaman *verifier portal* yang digunakan untuk melakukan verifikasi sertifikat digital. Pada halaman ini, pengguna dapat melakukan verifikasi melalui dua metode, yaitu unggah file sertifikat (*upload result*) atau menggunakan ID sertifikat (*ID result*). Sistem kemudian menampilkan hasil verifikasi berupa status valid atau tidak valid, beserta informasi detail seperti nama pemilik, ID sertifikat, nilai *hash*, *transaction hash*, dan CID dari IPFS. Dengan halaman ini, proses verifikasi sertifikat dapat dilakukan secara transparan dan mudah oleh pihak yang berkepentingan.

G. Evaluasi Rancangan Sistem

Evaluasi rancangan sistem dilakukan untuk menilai kesesuaian antara rancangan yang telah dibuat dengan kebutuhan sistem verifikasi sertifikat digital berbasis *blockchain*. Evaluasi ini mencakup aspek fungsional, keamanan, serta keterpaduan antar

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

komponen sistem yang melibatkan aplikasi web, *smart contract*, IPFS, dan basis data. Setiap bagian sistem dianalisis untuk memastikan bahwa proses penerbitan dan verifikasi sertifikat dapat berjalan sesuai dengan alur yang telah dirancang.

Dari sisi fungsional, sistem telah mampu mendukung proses utama seperti pengelolaan data *student*, penerbitan sertifikat oleh *issuer*, serta verifikasi sertifikat oleh *verifier* melalui dua metode, yaitu unggah file dan input ID sertifikat. Dari sisi keamanan, penggunaan *hash* dan pencatatan data pada *blockchain* memastikan bahwa data sertifikat tidak dapat diubah serta dapat diverifikasi keasliannya. Selain itu, integrasi dengan IPFS memungkinkan penyimpanan dokumen secara terdistribusi sehingga meningkatkan efisiensi penyimpanan.

Secara keseluruhan, rancangan sistem yang diusulkan telah memenuhi kebutuhan dalam meningkatkan keamanan, integritas, dan transparansi sertifikat digital. Namun, pengujian lebih lanjut pada tahap implementasi tetap diperlukan untuk memastikan performa sistem, khususnya terkait kecepatan transaksi, biaya *gas fee*, serta skalabilitas sistem dalam penggunaan secara nyata.

III. HASIL DAN PEMBAHASAN

A. Hasil Perancangan Sistem

Hasil dari penelitian ini berupa rancangan sistem verifikasi sertifikat digital berbasis *blockchain* yang telah disusun secara lengkap, meliputi analisis kebutuhan, perancangan sistem, arsitektur sistem, *smart contract*, serta antarmuka pengguna. Rancangan sistem mengintegrasikan teknologi *blockchain* Ethereum, IPFS, dan aplikasi berbasis web untuk mendukung proses penerbitan dan verifikasi sertifikat digital. Setiap komponen telah dirancang untuk saling terhubung, sehingga sistem mampu mendukung proses utama seperti pencatatan data sertifikat, penyimpanan dokumen secara terdistribusi, serta verifikasi keaslian sertifikat secara transparan.

B. Pembahasan Sistem

Sistem yang dirancang memanfaatkan *blockchain* sebagai media pencatatan data untuk menjamin integritas dan keaslian sertifikat digital. Penggunaan *smart contract* memungkinkan proses pencatatan dan verifikasi dilakukan secara otomatis tanpa memerlukan pihak ketiga, sehingga meningkatkan efisiensi dan kepercayaan terhadap sistem. Sementara itu, IPFS digunakan untuk menyimpan file sertifikat secara terdistribusi guna mengatasi keterbatasan penyimpanan pada *blockchain*. Integrasi ketiga komponen ini mendukung proses verifikasi sertifikat melalui dua metode, yaitu berdasarkan *hash* dokumen dan ID sertifikat, sehingga mampu memberikan fleksibilitas dalam proses validasi serta mengurangi risiko pemalsuan dokumen.

C. Analisis Keunggulan Sistem

Rancangan sistem memiliki beberapa keunggulan, di antaranya adalah tingkat keamanan yang tinggi melalui penggunaan *hash* dan pencatatan data pada *blockchain* yang bersifat *immutable*. Selain itu, sistem menyediakan transparansi dalam proses verifikasi karena data dapat diakses dan diverifikasi oleh pihak yang berkepentingan. Penggunaan IPFS juga memberikan efisiensi dalam penyimpanan data serta mengurangi ketergantungan terhadap sistem terpusat. Dengan kombinasi teknologi tersebut, sistem mampu meningkatkan kepercayaan terhadap keaslian sertifikat digital serta meminimalkan potensi manipulasi data.

D. Analisis Keterbatasan Sistem

Meskipun memiliki berbagai keunggulan, rancangan sistem ini masih memiliki beberapa keterbatasan. Sistem yang dirancang belum diimplementasikan secara penuh sehingga belum dilakukan pengujian performa secara langsung, seperti kecepatan transaksi dan penggunaan *gas fee*. Selain itu, penggunaan *blockchain* masih bergantung pada kondisi jaringan, sehingga dapat mempengaruhi waktu proses pencatatan data. Biaya transaksi pada jaringan Ethereum juga menjadi pertimbangan dalam implementasi nyata. Oleh karena itu, diperlukan pengembangan lebih lanjut untuk mengoptimalkan performa sistem serta menyesuaikan dengan kebutuhan penggunaan di lingkungan sebenarnya.

IV. KESIMPULAN

Berdasarkan hasil perancangan sistem verifikasi sertifikat digital berbasis *blockchain* yang telah dilakukan, dapat disimpulkan bahwa rancangan sistem mampu memenuhi kebutuhan dalam meningkatkan keamanan, integritas, dan keaslian sertifikat digital. Sistem dirancang dengan mengintegrasikan *blockchain* Ethereum, *smart contract*, dan IPFS, sehingga proses pencatatan dan verifikasi sertifikat dapat dilakukan secara transparan dan tidak bergantung pada sistem terpusat. Selain itu, penggunaan *hash* dokumen sebagai identitas digital memungkinkan deteksi terhadap perubahan data, sehingga dapat meminimalkan risiko pemalsuan sertifikat. Rancangan sistem juga telah mendukung proses utama, yaitu penerbitan sertifikat oleh *issuer* serta verifikasi sertifikat oleh *verifier* melalui dua metode, yaitu unggah file dan input ID sertifikat. Dengan adanya rancangan ini, sistem memiliki potensi untuk dikembangkan menjadi aplikasi yang mampu meningkatkan kepercayaan terhadap keaslian sertifikat digital. Namun demikian, penelitian ini masih terbatas pada tahap perancangan, sehingga diperlukan implementasi dan pengujian lebih lanjut untuk mengevaluasi performa sistem, efisiensi biaya, serta skalabilitas dalam penggunaan nyata.

REFERENSI

[1] B. A. I. Swari and I. M. Suartana, “Pengembangan Decentralized Application (DApp) untuk Manajemen Sertifikat Digital menggunakan Non-Fungible Tokens (NFT) berbasis Standar ERC-721 di Jaringan Ethereum,” *J. Informatics Comput. Sci.*, vol. 6, no. 03, pp. 661–668, 2024, doi: 10.26740/jinacs.v6n03.p661-668.

[2] Liputan6.com, “3 Pelaku Ditangkap karena Palsukan Sertifikat K3.” [Online]. Available: https://www.liputan6.com/news/read/4119030/3-pelaku-ditangkap-karena-palsukan-sertifikat-k3

[3] Zahwa Juwita, Febri Yollanda, Mita Azira, and Zul Azmi, “Penerapan Blockchain Dalam Meningkatkan Keamanan Akuntansi Dan

Transparansi Sistem Informasi Akuntansi," *J. Akunt. Keuang. Dan Perpajak.* | *E-ISSN 3063-8208*, vol. 1, no. 3, pp. 339–345, 2025, doi: 10.62379/jakp.v1i3.249.
[4] H. S. Kartiko, T. Rismawan, and I. Ruslianto, "Implementasi IPFS untuk Mengurangi Gas Fee Smart Contract Ethereum pada Aplikasi Penggalangan Dana," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 2, p. 195, 2023, doi: 10.26418/jp.v9i2.61876.
[5] R. R. Judhie Putra, M. Nursalman, and F. Kautsar, "Quality of Service Diploma Recording System Using Smart Contracts and Nft Polygon Network on Layer-2 Ethereum Blockchain," *J. Tek. Inform.*, vol. 5, no. 6, pp. 1505–1515, 2024, doi: 10.52436/1.jutif.2024.5.6.1647.
[6] Pradana, T. Djatna, I. Hermadi, and I. Yuliasih, "Blockchain-based Traceability System for Indonesian Coffee Digital Business Ecosystem," *Int. J. Eng. Trans. B Appl.*, vol. 36, no. 5, pp. 879–893, 2023, doi: 10.5829/ije.2023.36.05b.05.
[7] Haifaa Ahmed Hasan, Hassan F. Al-Layla, and Farah N. Ibraheem, "A Review of Hash Function Types and their Applications," *Wasit J. Comput. Math. Sci.*, vol. 1, no. 3, pp. 75–88, 2022, doi: 10.31185/wjcm.52.
[8] G. Tripathi, M. A. Ahad, and G. Casalino, "A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges," *Decis. Anal. J.*, vol. 9, no. March, p. 100344, 2023, doi: 10.1016/j.dajour.2023.100344.
[9] S. Avasthi, K. Jain, A. Prakash, and T. Sanwal, "A Blockchain Architecture for Secure Decentralized System and Computing," *2024 3rd Int. Conf. Power Electron. IoT Appl. Renew. Energy its Control. PARC 2024*, pp. 415–420, 2024, doi: 10.1109/PARC59193.2024.10486648.
[10] S. Abdullah, "Implementasi Blockchain untuk Keamanan Data Akademik dalam Sistem Informasi Perguruan Tinggi," *Go Infotech J. Ilm. STMIK AUB*, vol. 31, no. 1, pp. 149–160, 2025, doi: 10.36309/goi.v31i1.371.
[11] L. Septiani, D. Ramdani, A. T. Maula, N. B. Nugroho, and M. F. Ar Ridho, "Pengembangan Sistem Informasi Sertifikasi Menggunakan Metode Agile," *Digit. Transform. Technol.*, vol. 4, no. 2, pp. 1060–1066, 2025, doi: 10.47709/digitech.v4i2.5144.
[12] A. M. Pangidoan, P. W. Buana, and F. Purnama, "Prototype of Implementation of Smart Contract for Blockchain-Based Document Storage," *J. Appl. Informatics Comput.*, vol. 9, no. 4, pp. 1242–1253, 2025, doi: 10.30871/jaic.v9i4.9493.
[13] J. Komunikasi, Y. Universitas, P. Ganesha, P. Studi, and I. Hukum, "ANALISIS KEABSAHAN (SMART CONTRACT) TRANSAKSI ASET DIGITAL DI PLATFORM ETHERUM DALAM TEKNOLOGI BLOCKCHAIN," vol. 7, no. 1, 2024.
[14] T. R. D. Suseno, I. Afrianto, and S. Atin, "Strengthening data integrity in academic document recording with blockchain and InterPlanetary file system," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 2, pp. 1759–1769, 2024, doi: 10.11591/ijece.v14i2.pp1759-1769.
[15] A. Alfina and S. Syafrinal, "Model Sistem Verifikasi Dokumen Ijazah Digital Berbasis Teknologi Blockchain," *SMARTICS J.*, vol. 8, no. 2, pp. 59–65, 2022, doi: 10.21067/smartics.v8i2.7718.