# SISTEM VERIFIKASI SERTIFIKAT BERBASIS *BLOCKCHAIN* DENGAN IMPLEMENTASI ALGORITMA *SHA-256* DAN *SMART CONTRACT*

**SKRIPSI**

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika

**SYARIFUL MUJADDIQ**
**105841103219**

**PROGRAM STUDI INFORMATIKA**
**FAKULTAS TEKNIK**
**UNIVERSITAS MUHAMMADIYAH MAKASSAR**
**2025**

## PENGESAHAN

Skripsi atas nama Syariful Mujaddiq dengan nomor induk Mahasiswa 105 84 11032 19, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

Makassar, 6 Rabi'ul Awal 1447 H / 30 Agustus 2025 M

1. Pengawas Umum
   a. Rektor Universitas Muhammadiyah Makassar
      Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU :
   b. Dekan Fakultas Teknik Universitas Hasanuddin
      Prof. Dr. Eng. Muhammad Isran Ramli, S.T.,M.T.,ASEAN.,Eng :
2. Penguji
   a. Ketua : Prof. Dr. Ir. Hafsah Nirwana, M.T :
   b. Sekertaris : Titin Wahyuni, S.Pd., M.T :
3. Anggota : 1. Fahrim Ihhamna Rachman, S.Kom., M.T :
   2. Ir. Ida, S.Kom., M.T :
   3. Chyquita Danuputri, S.Kom., M.Kom :

Mengetahui :

| Pembimbing I | Pembimbing II |
|---|---|
| **Muhyiddin A M Hayat, S.Kom., M.T** | **Lukman, S.Kom.,M.T** |

Dekan

**Ir. Muh. Syafaat S. Kuba, S.T., M.T.**
NBM : 795 288

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK

# HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

| | | |
|---|---|---|
| Judul Skripsi | : | **SISTEM VERIFIKASI SERTIFIKAT BERBASIS BLOCKCHAIN DENGAN IMPLEMENTASI ALGORITMA SHA-256 DAN SMART CONTRACT** |
| Nama | : | Syariful Mujaddiq |
| Stambuk | : | 105 84 11032 19 |

Makassar, 20 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

| Pembimbing I | Pembimbing II |
|---|---|
| **Muhyiddin A M Hayat S.Kom., M.T** | **Lukman S.Kom., M.T** |

Mengetahui,
Ketua Prodi Informatika

**Rizki Yusliana Bakti, S.T., M.T**
NBM : 1307 284

Gedung Menara Iqra Lantai 3
Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
Web: https://teknik.unismuh.ac.id/, e-mail: teknik@unismuh.ac.id

## ABSTRAK

**SYARIFUL MUJADDIQ**, Sistem Verifikasi Sertifikat Berbasis *Blockchain* dengan Implementasi Algoritma SHA-256 dan *Smart Contract* (Di bawah bimbingan Muhyiddin A M Hayat, S.Kom., M.T. dan Lukman, S.Kom., M.T.)

Penelitian ini merancang dan mengimplementasikan sistem verifikasi sertifikat berbasis *blockchain* dengan penerapan *Secure Hash Algorithm 256* (SHA-256) untuk menjaga integritas data, serta *smart contract* sebagai mekanisme validasi otomatis pada jaringan Ethereum. Sistem ini dikembangkan untuk Fakultas Teknik Universitas Muhammadiyah Makassar guna mengatasi pemalsuan sertifikat digital dan inefisiensi metode verifikasi konvensional. Setiap sertifikat diproses untuk menghasilkan *hash* SHA-256, diunggah ke *InterPlanetary File System* (IPFS) melalui Pinata, kemudian *hash* dan *Content Identifier* (CID) dicatat *on-chain*. Dua metode verifikasi disediakan, yaitu unggah file PDF untuk verifikasi internal dan pemindaian *Quick Response* (QR) Code untuk verifikasi publik. Arsitektur sistem memanfaatkan Node.js, Express, React, dan *smart contract* berbasis Solidity dengan manajemen melalui Hardhat. Hasil *unit testing*, integrasi, dan *end-to-end testing* menunjukkan tingkat keberhasilan 100% dalam penyimpanan dan verifikasi *hash*, keberhasilan penuh unggah IPFS, serta waktu respons verifikasi rata-rata di bawah lima detik. Sistem ini memberikan verifikasi sertifikat yang aman, transparan, dan terdesentralisasi, sehingga meningkatkan kepercayaan dan efisiensi dalam autentikasi dokumen digital.



**Kata Kunci**: Verifikasi Sertifikat, Blockchain, SHA-256, *Smart Contract*, IPFS.

## ABSTRACT

**SYARIFUL MUJADDIQ**, *Blockchain-Based Certificate Verification System Implementing the SHA-256 Algorithm and Smart Contract* (*Under the guidance of Muhyiddin A M Hayat, S.Kom., M.T. and Lukman, S.Kom., M.T*)

*This study designs and implements a blockchain-based certificate verification system that applies the Secure Hash Algorithm 256 (SHA-256) to ensure data integrity and utilizes smart contracts for automated validation on the Ethereum network. Developed for the Faculty of Engineering, Universitas Muhammadiyah Makassar, the system addresses digital certificate forgery and inefficiencies in conventional verification. Each certificate is processed to generate a SHA-256 hash, uploaded to the InterPlanetary File System (IPFS) via Pinata, and recorded on-chain along with its Content Identifier (CID). Two verification methods are provided: PDF upload for internal verification and Quick Response (QR) code scanning for public validation.The architecture integrates Node.js, Express, and React for application layers, with Solidity-based smart contracts managed using Hardhat. Unit, integration, and end-to-end testing confirmed 100% success in hash storage and retrieval, full IPFS upload reliability, and average verification response times below five seconds in a local environment. This system delivers a secure, transparent, and decentralized verification framework, enhancing trust and efficiency in digital document authentication.*



***Keywords***: *Certificate Verification, Blockchain, SHA-256, Smart Contract, IPFS*

## KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Allah Subhanahu Wa Ta'ala atas limpahan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul **"Sistem Verifikasi Sertifikat Berbasis *Blockchain* dengan Implementasi Algoritma *SHA-256* dan *Smart Contract*"** tepat waktu sesuai dengan rencana yang telah ditetapkan.Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, dengan segala hormat penulis menyampaikan apresiasi dan ucapan terima kasih yang sebesar-besarnya kepada:

1. **Allah Subhanahu Wa Ta'ala** yang telah memberikan kesehatan, kekuatan, serta kesempatan kepada penulis dalam menyelesaikan skripsi ini.
2. **Kedua orang tua** tercinta, serta **saudara** , atas segala doa, dukungan moral dan materil, serta kasih sayang yang tiada henti dalam setiap proses kehidupan penulis.
3. Bapak **Ir. Muh. Syafaat S. Kuba, S.T., M.T., IPM** selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar, yang senantiasa memberikan arahan dan dukungan akademik.
4. Ibu **Rizki Yusliana Bakti, S.T., M.T** selaku Ketua Program Studi Informatika, atas arahan, dan kontribusi akademiknya dalam proses penyusunan skripsi ini.
5. Bapak **Muhyiddin A. M. Hayat, S.Kom., M.T** selaku Dosen Pembimbing I, atas bimbingan, arahan, dan kontribusi akademiknya dalam proses penyusunan skripsi ini.
6. Bapak **Lukman, S.Kom., M.T** selaku Dosen Pembimbing II, atas dedikasi, masukan, serta waktu yang diberikan dalam membimbing penulis.
7. Rekan-rekan di **COCONUT Computer Club** yang telah memberikan masukan konstruktif serta motivasi selama proses penulisan berlangsung.
8. Seluruh rekan dari **MARVEL 012** yang turut memberikan semangat dan dukungan selama proses penyusunan skripsi.
9. Seluruh rekan seangkatan di Program Studi Informatika yang turut memberikan semangat dan dukungan selama proses penyusunan skripsi.

10. Staf akademik dan administrasi yang telah memberikan bantuan teknis dan informasi yang dibutuhkan dalam penyusunan dokumen ini.

Penulis menyadari bahwa skirpsi ini masih memiliki kekurangan baik dari segi teknis maupun substansi. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan guna penyempurnaan karya ilmiah ini di masa mendatang. Penulis berharap skripsi ini dapat memberikan kontribusi positif dalam pengembangan ilmu pengetahuan, khususnya di bidang teknologi informasi.

Makassar, 28 Agustus 2025

Hormat penulis,

Syariful Mujaddiq

# DAFTAR ISI

## DAFTAR GAMBAR

## DAFTAR TABEL

## DAFTAR LAMPIRAN

# DAFTAR ISTILAH

| | |
|---|---|
| ***Blockchain*** | Sistem pencatatan digital *(distributed ledger technology*) yang menyimpan data dalam bentuk blok (*block*) yang saling terhubung berurutan (*chain*) menggunakan kriptografi.. |
| **Desentralisasi** | Suatu konsep distribusi wewenang, kontrol, dan pengambilan keputusan yang tidak terpusat pada satu entitas tunggal, melainkan tersebar ke berbagai pihak. |
| **Immutable** | Sifat suatu data atau informasi yang tidak dapat diubah atau dimodifikasi setelah dicatat, sehingga menjamin keaslian dan integritas data. |
| **Sertifikat Digital** | Representasi elektronik dari dokumen legal yang berfungsi untuk membuktikan identitas, pencapaian, ataupun kualifikasi tertentu dari individu maupun institusi. |
| **SHA-256** | Fungsi hash kriptografi yang menghasilkan output tetap sepanjang 256 bit (32 byte) dari suatu input data |
| ***Smart Contract*** | Program atau protokol digital yang berjalan di atas blockchain dan dieksekusi secara otomatis ketika kondisi yang telah ditentukan terpenuhi. |
| **Solidity** | Bahasa pemrograman tingkat tinggi yang dirancang khusus untuk menulis *smart contract* pada platform blockchain, terutama Ethereum. |
| **Ethereum** | Sebuah platform blockchain terdesentralisasi yang memungkinkan pengembangan dan eksekusi *smart contract* serta aplikasi terdesentralisasi (*dApps*). |

| | |
|---|---|
| **QR Code** | Jenis kode matriks dua dimensi yang dapat menyimpan informasi dalam bentuk teks, tautan, atau data lainnya, dan dapat dipindai dengan cepat menggunakan kamera atau pemindai khusus. |
| **InterPlanetary File System (IPFS)** | Protokol dan jaringan terdistribusi yang dirancang untuk menyimpan dan berbagi data dalam bentuk sistem file terdesentralisasi. |
| ***Content Identifier (CID)*** | Penanda unik berbasis hash yang digunakan dalam protokol IPFS untuk mengidentifikasi suatu konten atau file secara permanen. |
| **Pinata** | Layanan berbasis cloud yang menyediakan infrastruktur untuk mengunggah, menyimpan, dan mengelola konten di jaringan *InterPlanetary File System* (IPFS) |
| ***Hash*** | Hasil keluaran dari fungsi hash kriptografi yang mengubah suatu input data menjadi string karakter dengan panjang tetap |
| ***Hardhat*** | Sebuah lingkungan pengembangan untuk membangun, menguji, dan men-deploy *smart contract* pada jaringan blockchain Ethereum. |
| **Unit Testing** | Metode pengujian perangkat lunak yang dilakukan pada unit terkecil dari kode program, seperti fungsi atau modul, untuk memastikan bahwa bagian tersebut bekerja sesuai dengan yang diharapkan secara terisolasi dari komponen lainnya. |
| **End-To-End Testing** | Metode pengujian perangkat lunak yang memverifikasi alur kerja aplikasi secara menyeluruh dari awal hingga akhir, untuk memastikan semua komponen berfungsi |

| | |
|---|---|
| | bersama sesuai dengan kebutuhan dan skenario penggunaan nyata. |
| ***JSON Web Token* (JWT)** | Standar Internet yang diusulkan untuk membuat data dengan tanda tangan opsional dan atau enkripsi opsional yang muatannya berisi JSON yang menyatakan sejumlah klaim . |
| ***Proof of Concept*** | Sebuah implementasi awal atau prototipe yang dibuat untuk membuktikan bahwa suatu ide, konsep, atau teknologi dapat dijalankan dan berfungsi sesuai dengan tujuan yang diharapkan. |
| **Latensi** | Waktu tunda yang terjadi antara pengiriman suatu perintah atau data dengan respons yang diterima. |
| ***Modifier*** | Fitur dalam bahasa pemrograman Solidity yang digunakan untuk mengubah atau membatasi perilaku suatu fungsi tanpa harus menulis ulang logika yang sama. |

# BAB I
# PENDAHULUAN

## A. Latar Belakang

Dalam beberapa tahun terakhir, keabsahan sertifikat digital menjadi perhatian utama dalam berbagai sektor, termasuk sektor pendidikan tinggi. Sertifikat kelulusan, transkrip nilai, maupun ijazah merupakan dokumen penting yang menjadi bukti formal kompetensi akademik seseorang. Namun, permasalahan umum seperti pemalsuan dokumen, verifikasi manual, seperti pengecekan sertifikat yang memakan waktu, serta ketergantungan pada pihak ketiga masih menjadi tantangan yang serius dalam sistem verifikasi tradisional. Oleh karena itu, diperlukan pendekatan teknologi yang mampu menjamin keaslian dan integritas dokumen akademik secara transparan dan efisien.

Teknologi *blockchain* hadir sebagai solusi inovatif dalam menjawab tantangan ini. Sebagai buku besar terdistribusi (*distributed ledger*), blockchain memiliki sifat *immutable* dan transparan, yang menjadikannya sangat ideal untuk implementasi sistem verifikasi dokumen. Dengan menggunakan blockchain, setiap data sertifikat yang dimasukkan akan dicatat dalam blok-blok yang saling terkait dan diverifikasi oleh jaringan secara desentralisasi, sehingga mengeliminasi kemungkinan manipulasi atau kehilangan data (Rakhmansyah et al., 2021).

Salah satu komponen penting dalam pengamanan data pada blockchain adalah algoritma hash SHA-256. Algoritma ini berfungsi menghasilkan nilai hash unik dari setiap data input, yang bersifat satu arah dan sangat sulit untuk direkayasa balik. Dengan demikian, setiap perubahan pada dokumen asli akan secara otomatis menghasilkan nilai hash yang berbeda, memungkinkan deteksi dini terhadap pemalsuan sertifikat (Wijayanto, 2024). SHA-256 juga merupakan algoritma yang telah teruji dalam berbagai sistem keamanan digital karena tingkat kompleksitas dan efisiensinya dalam memproses data.

Selain itu, penggunaan *smart contract,* yaitu skrip program otomatis yang berjalan di dalam jaringan *blockchain* menambahkan dimensi efisiensi dan otomatisasi dalam proses validasi. *Smart contract* memungkinkan verifikasi dilakukan tanpa keterlibatan pihak ketiga secara langsung, dengan aturan yang telah ditetapkan sebelumnya, dan akan berjalan otomatis ketika kondisi terpenuhi. Dalam konteks pendidikan tinggi, hal ini sangat membantu untuk mempercepat proses validasi ijazah oleh instansi pengguna seperti perusahaan atau lembaga pemerintah (Firza & Yuhandri, 2024).

Implementasi sistem verifikasi sertifikat berbasis blockchain yang menggabungkan SHA-256 dan *smart contract* tidak hanya mampu menjawab permasalahan pemalsuan dokumen, namun juga mendorong transformasi digital yang terpercaya di lingkungan akademik. Bagi Fakultas Teknik Universitas Muhammadiyah Makassar, pengembangan sistem ini sejalan dengan kebutuhan untuk meningkatkan kualitas layanan administrasi akademik dan memberikan solusi teknologi yang adaptif terhadap perkembangan Revolusi Industri 4.0.

Dengan demikian, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem verifikasi sertifikat akademik berbasis *blockchain* dengan penerapan algoritma SHA-256 dan *smart contract*. Sistem ini diharapkan mampu memberikan keamanan, efisiensi, serta transparansi dalam proses verifikasi dokumen, sekaligus menjadi kontribusi nyata dalam pengembangan teknologi informasi di lingkungan pendidikan tinggi.

## B. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah yang dapat diidentifikasi dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang sistem verifikasi sertifikat yang aman dan terpercaya menggunakan teknologi *blockchain* dan algoritma SHA-256?

2. Bagaimana penerapan *smart contract* dapat mengotomatisasi proses verifikasi sertifikat akademik pada lingkungan Fakultas Teknik Universitas Muhammadiyah Makassar?
3. Seberapa efektif dan aman sistem yang dikembangkan dalam meningkatkan integritas dan efisiensi verifikasi sertifikat dibandingkan dengan metode konvensional?

## C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

1. Untuk merancang dan mengembangkan sistem verifikasi sertifikat berbasis *blockchain* dengan menggunakan algoritma SHA-256 sebagai mekanisme hash.
2. Untuk menerapkan *smart contract* pada jaringan *blockchain* sebagai sarana otomatisasi proses validasi sertifikat akademik.
3. Untuk mengevaluasi efektivitas dan keamanan sistem yang dibangun dalam konteks verifikasi sertifikat akademik di Fakultas Teknik Universitas Muhammadiyah Makassar.

## D. Manfaat Penelitian

Penelitian ini diharapkan memberikan kontribusi yang signifikan baik secara teoritis maupun praktis, antara lain:

1. Manfaat akademisi: Memberikan kontribusi ilmiah dalam pengembangan sistem keamanan data berbasis blockchain, khususnya dalam penerapan algoritma *SHA-256* dan *smart contract* untuk proses verifikasi Sertifikat atau dokumen
2. Manfaat Praktis: Menyediakan solusi sistem verifikasi sertifikat yang dapat diterapkan secara langsung oleh institusi pendidikan, perusahaan, maupun lembaga sertifikasi untuk mencegah terjadinya pemalsuan sertifikat digital serta meningkatkan efisiensi dalam proses validasi.

3. Manfaat Sosial: Meningkatkan kepercayaan masyarakat terhadap keaslian sertifikat yang beredar secara daring serta mendorong transparansi dan akuntabilitas dalam sistem penerbitan dokumen digital.

## E. Ruang Lingkup Penelitian

Agar penelitian dapat terfokus dan terarah, ruang lingkup dari penelitian ini dibatasi pada beberapa hal berikut:

1. Pengembangan sistem verifikasi sertifikat hanya dilakukan untuk skenario penggunaan terbatas (simulasi) dengan sertifikat *dummy* yang menyerupai sertifikat pendidikan atau pelatihan.
2. Platform *blockchain* yang digunakan adalah *Ethereum*, dengan penerapan *smart contract* menggunakan bahasa pemrograman *Solidity.*
3. Proses hash data sertifikat dilakukan menggunakan algoritma *SHA-256*, dan hash tersebut disimpan ke dalam *blockchain* sebagai representasi data yang akan diverifikasi.
4. Sistem tidak mencakup integrasi dengan sistem akademik atau *database* institusi secara penuh, melainkan difokuskan pada pembuktian konsep *(proof of concept)* dari mekanisme verifikasi yang aman dan desentralistik.
5. Aspek legalitas, regulasi, atau kebijakan privasi tidak dibahas secara mendalam dan berada di luar cakupan penelitian ini.
6. Administrator hanya dapat memvalidasi sertifikat yang di terbitkan melalui sistem.

## F. Sistematika Penulisan

Agar mempermudah pemahaman terhadap alur penelitian, penulisan skripsi ini disusun dengan sistematika sebagai berikut:

**BAB I PENDAHULUAN**

Menguraikan latar belakang permasalahan, rumusan masalah, tujuan dan manfaat penelitian, ruang lingkup penelitian, serta sistematika penulisan.

**BAB II TINJAUAN PUSTAKA**

Menyajikan teori-teori dasar yang relevan, termasuk pengertian *blockchain*, algoritma SHA-256, *smart contract*, dan lainnya serta studi literatur dari penelitian terdahulu yang mendukung dan menjadi dasar pengembangan sistem.

**BAB III METODOLOGI PENELITIAN**

Berisi metode yang digunakan dalam proses penelitian, termasuk metode pengumpulan data, desain sistem, arsitektur teknis, serta tahapan implementasi dan pengujian sistem.

**BAB IV HASIL DAN PEMBAHASAN**

Menyajikan hasil implementasi sistem verifikasi, cara kerja smart contract yang digunakan, serta analisis terhadap performa dan efektivitas sistem berdasarkan skenario uji.

**BAB V PENUTUP**

Menyimpulkan hasil penelitian serta memberikan saran-saran yang dapat dijadikan dasar pengembangan sistem di masa mendatang atau pengaplikasian di lingkungan nyata.

# BAB II

# TINJAUAN PUSTAKA

## A. Landasan Teori

### 1. *Blockchain*

a. Definisi dan karakteristik umum *Blockchain*

*Blockchain* merupakan suatu bentuk *distributed ledger technology* (DLT) yang memungkinkan pencatatan transaksi secara digital dalam struktur data berbentuk rantai blok *(blockchain)*, di mana setiap blok saling terhubung melalui fungsi hashing kriptografis. Sistem ini bersifat desentralistik dan terdistribusi, sehingga tidak memerlukan otoritas pusat dalam memverifikasi transaksi (Luhkito et al., 2021). Selain itu, *blockchain* mengedepankan prinsip transparansi, integritas data, serta imutabilitas, yaitu ketidakmampuan data untuk diubah atau dimanipulasi setelah terekam dalam blok (Nanda Sari & Gelar, 2024).

b. Kriptografi dan Struktur Blok

Setiap blok dalam *blockchain* terdiri dari beberapa komponen utama, yakni *timestamp, Merkle root, nonce,* serta *hash* dari blok sebelumnya. Integritas dan keautentikan data dijamin melalui fungsi hashing kriptografis seperti SHA-256 atau algoritma alternatif seperti *U-Quark*, yang menghasilkan nilai *hash* unik untuk setiap blok (Luhkito et al., 2021). Struktur ini membentuk rantai yang saling terhubung, di mana modifikasi terhadap satu blok akan merusak keseluruhan rantai, menjadikannya tahan terhadap pemalsuan data (Lestari et al., 2024).

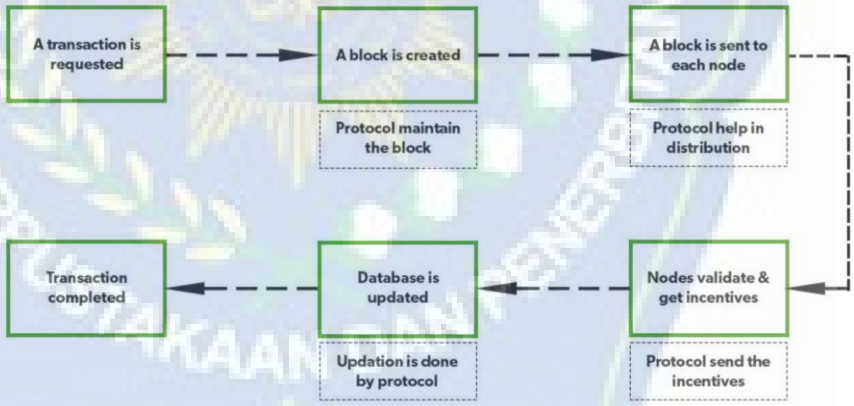
c. Arsitektur Jaringan *Peer-to-Peer* dan Desentralisasi

*Blockchain* beroperasi dalam arsitektur jaringan *peer-to-peer* (P2P), di mana setiap *node* dalam jaringan memelihara salinan lengkap dari ledger dan berpartisipasi dalam proses validasi transaksi secara kolektif. Pendekatan ini mengeliminasi ketergantungan terhadap *trusted*

*third party*, meningkatkan efisiensi, serta memperkuat ketahanan jaringan terhadap serangan terpusat (Lestari et al., 2024). Desentralisasi ini juga memungkinkan sistem bersifat *fault-tolerant* dan mendukung verifikasi yang dapat dilakukan oleh siapa pun (Nanda Sari & Gelar, 2024).

d. Karakteristik Transparansi, Imutabilitas, dan Keamanan Data

*Blockchain* menawarkan transparansi tinggi karena setiap transaksi dapat diverifikasi secara publik. Selain itu, sifat imutabilitas menjamin bahwa data yang telah dikonfirmasi tidak dapat diubah, sehingga cocok diterapkan pada sistem yang membutuhkan keabsahan dan auditabilitas tinggi (Nanda Sari & Gelar, 2024). Aspek keamanan juga diperkuat oleh penggunaan algoritma kriptografi yang kompleks dan replikasi data di seluruh node jaringan (Sari & Nasution, 2024).




*Gambar 1 Ilustrasi Blockchain Bekerja*

**2. Sertifikat Digital**

a. Definisi Sertifikat Digital

Sertifikat digital merupakan representasi elektronik dari dokumen legal yang berfungsi untuk membuktikan identitas,

pencapaian, ataupun kualifikasi tertentu dari individu maupun institusi. Dalam ranah sistem informasi, sertifikat digital mengandung atribut seperti nama penerima, entitas penerbit, tanggal validitas, serta elemen keamanan digital untuk menjamin keaslian dan integritas dokumen tersebut. Penerapan sertifikat digital dimaksudkan untuk meminimalkan risiko pemalsuan dokumen dan mempercepat proses validasi secara daring (Sintyaningrum et al., 2022).

b. Mekanisme Verifikasi Sertifikat Digital

Verifikasi terhadap sertifikat digital merupakan proses krusial yang bertujuan untuk memastikan autentisitas serta integritas dokumen. Proses ini mencakup pembuktian bahwa dokumen tidak mengalami modifikasi setelah diterbitkan, serta validasi identitas pihak yang menerbitkan dokumen tersebut. Dua pendekatan yang umum digunakan dalam sistem konvensional untuk proses verifikasi adalah penggunaan tanda tangan digital *(digital signature)* dan kode respons cepat (QR Code) sebagai identifikasi unik.

*1) Digital Signature*

Tanda tangan digital merupakan hasil penerapan algoritma kriptografi asimetris yang berfungsi untuk menjamin integritas data sekaligus autentikasi sumber dokumen. Teknik ini umumnya melibatkan proses hashing terhadap isi dokumen menggunakan algoritma tertentu, seperti SHA-256, kemudian hasil hash tersebut dienkripsi menggunakan kunci privat milik penerbit. Di sisi penerima, proses verifikasi dilakukan dengan mendekripsi tanda tangan digital menggunakan kunci publik, lalu membandingkannya dengan hasil hash dokumen saat ini (Goffar & Arifah, 2023). Dengan demikian, validitas dokumen hanya dapat dikonfirmasi apabila dokumen tidak mengalami perubahan dan pasangan kunci publik-privat berasal dari entitas yang sah.

2) Penggunaan QR Code dalam Proses Verifikasi

QR Code dimanfaatkan sebagai sarana penyimpanan informasi verifikasi secara ringkas dan mudah diakses. Informasi yang tertanam dapat berupa URL menuju sistem verifikasi daring, ID unik sertifikat, atau hash dari dokumen yang diterbitkan. Pendekatan ini dinilai efisien dalam konteks implementasi sistem verifikasi berbasis web karena memfasilitasi proses validasi yang cepat dan *user-friendly* tanpa memerlukan pemahaman teknis dari pengguna akhir (Sintyaningrum et al., 2022).

c. *Public Key Infrasturcture* (PKI)

Dalam sistem verifikasi digital tradisional, *Public Key Infrastructure* (PKI) merupakan arsitektur utama yang mendasari pengelolaan sertifikat dan autentikasi identitas entitas digital. PKI terdiri atas komponen utama seperti *Certificate Authority* (CA), *Registration Authority* (RA), dan sistem manajemen kunci. CA bertugas sebagai otoritas penerbit sertifikat yang dipercaya untuk menjamin kebenaran identitas pemegang sertifikat. Format sertifikat yang umum digunakan adalah X.509, yang telah menjadi standar global dalam pengelolaan sertifikat digital.

Studi yang dilakukan oleh FONICA (2021) menunjukkan bahwa validasi sertifikat dapat dioptimalkan melalui penerapan PKI yang dikombinasikan dengan paradigma *fog computing*, terutama pada sistem jaringan cerdas (*smart grid*). Dalam hal ini, verifikasi status sertifikat dilakukan melalui protokol seperti *Certificate Revocation List* (CRL) dan *Online Certificate Status Protocol* (OCSP) untuk memastikan sertifikat masih aktif dan belum dicabut, dengan latensi yang rendah berkat distribusi komputasi di lapisan lokal.

### 3. *Secure Hash Algorithm 256* (SHA-256)

*Secure Hash Algorithm 256* (SHA-256) merupakan salah satu varian dari keluarga SHA-2 yang dirancang oleh *National Institute of Standards and Technology* (NIST) dan dikenal sebagai algoritma fungsi hash satu arah dengan tingkat keamanan tinggi. SHA-256 beroperasi dengan menghasilkan *message digest* sepanjang 256 bit dari data masukan yang memiliki panjang variabel. Keunikan fungsi hash ini terletak pada kemampuannya dalam menjaga integritas data, karena perubahan sekecil satu bit pada input akan menghasilkan hash yang sepenuhnya berbeda, mencerminkan prinsip *avalanche effect* yang kuat (Sitorus et al., 2024).

Algoritma *SHA-256* melakukan proses pemrosesan data melalui beberapa tahapan utama: padding pesan agar kelipatan 512 bit, pembentukan *message schedule* sebanyak 64 elemen 32-bit, serta eksekusi fungsi kompresi melalui delapan variabel hash internal yang diperbaharui secara iteratif dalam 64 putaran (Suhendra & Maslan, 2024). Operasi internalnya menggunakan fungsi logika *bit-level* seperti AND, OR, XOR, *shift right, dan rotate right* untuk menghasilkan distribusi hash yang acak dan tidak dapat diprediksi. Karena sifat satu arahnya, sangat sulit untuk merekonstruksi input dari output hash, menjadikan SHA-256 cocok untuk kebutuhan autentikasi, verifikasi, dan penyimpanan data secara aman.

Dalam konteks implementatif, SHA-256 telah diterapkan dalam berbagai sistem keamanan informasi. Misalnya, dalam penelitian Nainggolan (2022), algoritma ini digunakan untuk mendeteksi file duplikat pada sistem penyimpanan dokumen berbasis hash. Penelitian tersebut menunjukkan bahwa SHA-256 mampu membedakan file yang identik dari sisi konten meskipun hanya terdapat perubahan kecil pada satu *byte*, dengan tingkat akurasi mencapai 100% terhadap deteksi perubahan konten.

Selain itu, penerapan SHA-256 dalam *digital signature* juga menunjukkan hasil yang signifikan. Pada studi yang dilakukan oleh (Sutopo

et al., 2019), *SHA-256* digunakan sebagai fungsi hash dalam proses pembangkitan tanda tangan digital berbasis *Digital Signature Algorithm* (DSA) pada media gambar digital. Dengan mengubah gambar menjadi representasi ASCII dan selanjutnya dienkripsi menggunakan SHA-256, algoritma ini berhasil membuktikan keefektifannya dalam menjamin keaslian pesan visual melalui pembangkitan hash *fingerprint* yang unik dan sulit untuk dipalsukan.

Di sisi lain, penelitian oleh Halimi dkk (2024) menyoroti implementasi SHA-256 dalam sistem informasi penerimaan mahasiswa baru (PMB). Dalam penelitian tersebut, SHA-256 digunakan untuk menghitung hash dari kode unik pendaftar, yang selanjutnya dikombinasikan dengan metode enkripsi AES-256 dan *encoding Base64*. SHA-256 berfungsi sebagai lapisan validasi integritas data yang menjamin bahwa informasi yang tersimpan dalam basis data tidak mengalami manipulasi.

Dengan demikian, SHA-256 tidak hanya unggul dari sisi teoritis sebagai fungsi hash kriptografis yang tahan terhadap *collision*, *preimage*, dan *second preimage attack*, tetapi juga telah terbukti relevan dan efektif dalam penerapan nyata pada sistem keamanan informasi digital yang memerlukan validitas, keutuhan, dan keaslian data.

Berikut ringkasan persamaan matematika SHA-256 sesuai standar NIST beserta langkah-langkahnya :

Bekerja pada word 32-bit, operasi aritmetika modulo $2^{32}$.

Fungsi bit-level :

$$\mathrm{Ch}(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$$
$$\mathrm{Maj}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$
$$\Sigma_0(x) = \mathrm{ROTR}^2(x) \oplus \mathrm{ROTR}^{13}(x) \oplus \mathrm{ROTR}^{22}(x)$$
$$\Sigma_1(x) = \mathrm{ROTR}^6(x) \oplus \mathrm{ROTR}^{11}(x) \oplus \mathrm{ROTR}^{25}(x)$$
$$\sigma_0(x) = \mathrm{ROTR}^7(x) \oplus \mathrm{ROTR}^{18}(x) \oplus \mathrm{SHR}^3(x)$$
$$\sigma_1(x) = \mathrm{ROTR}^{17}(x) \oplus \mathrm{ROTR}^{19}(x) \oplus \mathrm{SHR}^{10}(x)$$

(ROTR = rotasi kanan, SHR = geser kanan, $\bigoplus$XOR).

a. Pra-proses (padding & parsing)

Untuk pesan bita *M* sepanjang *L* bit:

1) Tambahkan bit '1'.
2) Tambahkan *k* bit '0' sehingga $L + 1 + k \equiv 448 \pmod{512}$.
3) Tambahkan representasi 64-bit big-endian dari *L*. Hasilnya dipecah menjadi blok 512-bit $M^{(1)},\ldots, M^{(N)}$, masing-masing berisi 16 word 32-bit.

b. Nilai awal & konstanta

1) Inisialisasi delapan register hash $H_0,\ldots, H_7$ ke konstanta tetap (dari pecahan akar prima).
2) Gunakan 64 konstanta word $K_0,\ldots, K_{63}$.

c. Jadwal pesan

Untuk tiap blok *i*, bentuk $W_t$ ($t$ = 0..63t=0..63t=0..63):

$$W_t = \begin{cases} M_t^{(i)}, & 0 \le t \le 15 \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16} \mod 2^{32}, & 16 \le t \le 63 \end{cases}$$

d. Kompresi (64 putaran per blok)

Setel ($a, b, c, d, e, f, g, h$) ← ( $H_0, H_1, H_2, H_3, H_4, H_5, H_6, H_7$ )

Untuk t=0..63:

$$T_1 = h + \Sigma_1(e) + \mathrm{Ch}(e, f, g) + K_t + W_t \ (\mathrm{mod} 2^{32})$$
$$T_2 = \Sigma_0(a) + \mathrm{Maj}(a, b, c) \ (\mathrm{mod} 2^{32})$$
$$h \leftarrow g;\ g \leftarrow f;\ f \leftarrow e;$$
$$e \leftarrow d + T_1 \ (\mathrm{mod} 2^{32});\ d \leftarrow c;\ c \leftarrow b;\ b \leftarrow a;$$
$$a \leftarrow T_1 + T_2 \ (\mathrm{mod} 2^{32})$$

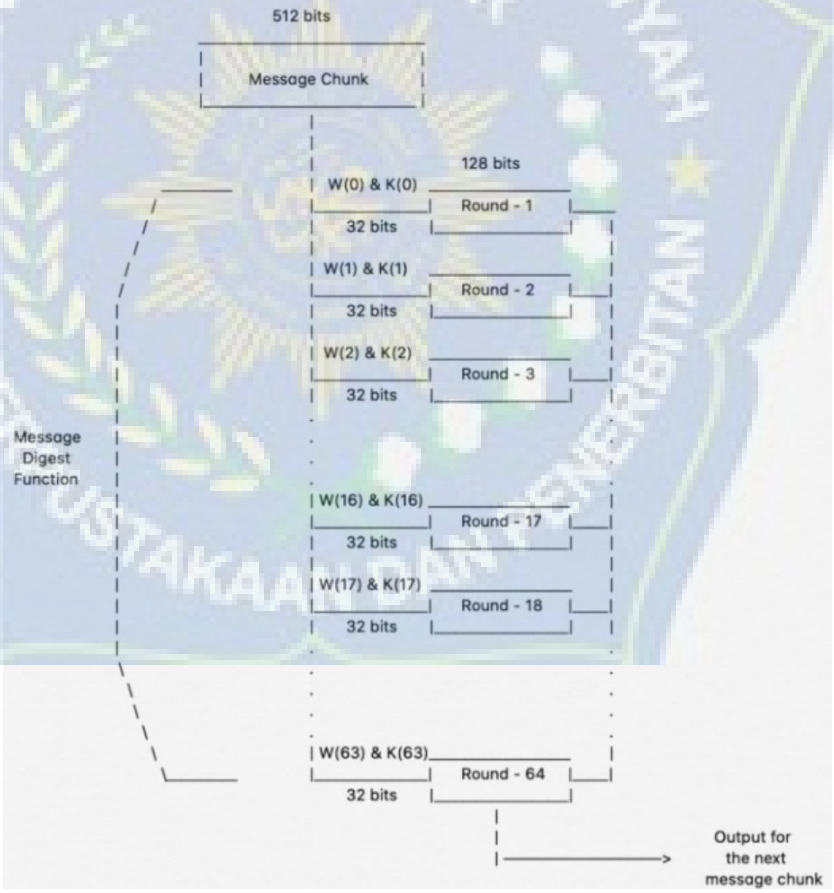
e. Setelah 64 putaran, lakukan penjumlahan penutup:

$H_{0,} \leftarrow H_0 \leftarrow a;\ldots ; H_7 \leftarrow H_7 + h\ (mod 2^{32})$

Maka;

$$Hash = H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4 \parallel H_5 \parallel H_6 \parallel H_7$$

Berikut adalah ilustrasi umum mengenai proses *hashing* dengan SHA-256:




*Gambar 2 Ilustrasi Proses Hashing SHA-256*

**4. *Smart Contract***

a. Definisi dan Karakteristik Dasar

*Smart contract* merupakan protokol digital yang dirancang untuk mengeksekusi, mengontrol, atau mendokumentasikan kejadian dan aksi sesuai dengan ketentuan kontrak yang telah diprogram secara otomatis dalam platform *blockchain*. *Smart contract* memungkinkan penghapusan kebutuhan terhadap pihak ketiga dalam transaksi karena eksekusi dilakukan secara otonom dan berbasis logika kode program yang *immutable* (Susanto et al., 2022, hlm. 249; Taherdoost, 2023, hlm. 2).

Taherdoost (2023) menjelaskan bahwa *smart contract* tidak hanya mendigitalkan kontrak konvensional, tetapi juga mengotomatisasi proses bisnis yang sebelumnya manual, sehingga meningkatkan efisiensi, mengurangi biaya administrasi, dan mengurangi risiko operasional.

b. Prinsip Hukum dan Legalitas

Dari perspektif hukum, *smart contract* masih menghadapi berbagai tantangan, terutama dalam konteks jurisdiksi dan *enforceability*. Baso et al. (2024) menekankan bahwa kendala terbesar di Indonesia adalah tidak adanya regulasi yang secara eksplisit mengatur keabsahan *smart contract*. Padahal, unsur sahnya perjanjian sebagaimana diatur dalam Pasal 1320 KUH Perdata sering kali sulit dibuktikan melalui kontrak digital karena persoalan kesepakatan, kemampuan hukum, sebab yang halal, dan bentuk tertentu yang belum tentu dipenuhi dalam *smart contract.*

Lebih lanjut, diungkapkan bahwa validitas *smart contract* dalam sistem hukum Indonesia dapat dicapai dengan mencantumkan klausul pilihan hukum dan yurisdiksi, terutama dalam konteks lintas negara (Baso et al., 2024, hlm. 98).

c. Penerapan dan Potensi di Berbagai Industri

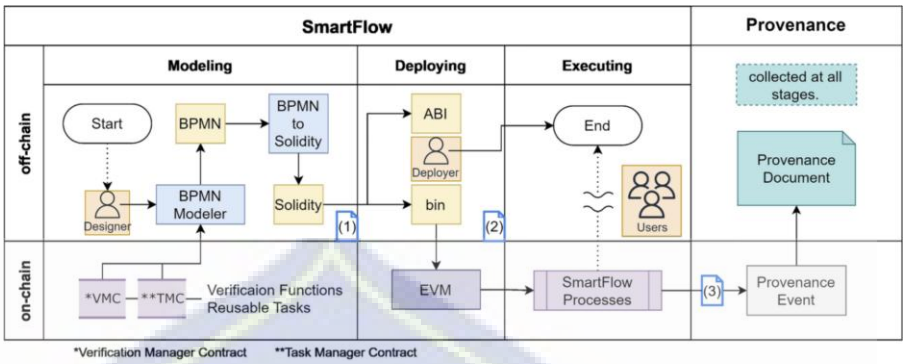
Dalam sektor jasa keuangan, *smart contract* memungkinkan otomatisasi pembayaran, eksekusi polis asuransi, dan transaksi berbasis kripto yang efisien dan aman. Susanto et al. (2022) menyebutkan bahwa teknologi ini mampu menggantikan peran perantara dengan logika digital yang dapat diandalkan dan mengurangi waktu serta biaya dalam pemrosesan transaksi.

Sementara itu, dalam industri konstruksi, smart contract mendukung otomatisasi sistem pembayaran, verifikasi progres proyek, serta integrasi dengan *Building Information Modeling* (BIM*)*. Teknologi ini dinilai mampu menyelesaikan masalah keterlambatan pembayaran dan sengketa kontrak yang sering terjadi dalam proyek konstruksi (Ye et al., 2022).

d. Resiko dan Tantangan

Dalam sektor jasa keuangan, smart contract memungkinkan otomatisasi pembayaran, eksekusi polis asuransi, dan transaksi berbasis kripto yang efisien dan aman. Susanto et al., (2022) menyebutkan bahwa teknologi ini mampu menggantikan peran perantara dengan logika digital yang dapat diandalkan dan mengurangi waktu serta biaya dalam pemrosesan transaksi.

Sementara itu, dalam industri konstruksi, *smart contract* mendukung otomatisasi sistem pembayaran, verifikasi progres proyek, serta integrasi dengan *Building Information Modeling* (BIM). Teknologi ini dinilai mampu menyelesaikan masalah keterlambatan pembayaran dan sengketa kontrak yang sering terjadi dalam proyek konstruksi (Ye et al., 2022).




*Gambar 3 Ilustrasi Alur Kerja Smart Contract*

**5. *Ethereum***

*Ethereum* merupakan sebuah platform blockchain terbuka yang memungkinkan pengembangan *decentralized application* (DApps) dan *smart contract* melalui mesin virtual yang disebut *Ethereum Virtual Machine* (EVM). Keunggulan utama *Ethereum* terletak pada kapabilitasnya dalam menyediakan infrastruktur untuk eksekusi logika bisnis otomatis dan transparan dalam jaringan *peer-to-peer*, dengan validasi berbasis konsensus.

Menurut Tanhar et al., (2023), *Ethereum* digunakan sebagai *backend* penyimpanan dan pengendali interaksi antar pengguna dalam ekosistem *metaverse* berbasis *audio player*. Teknologi *blockchain Ethereum* dimanfaatkan karena sifatnya yang terdesentralisasi dan imutabilitas data yang tinggi, sehingga meningkatkan kepercayaan dalam sistem pertukaran digital tanpa perantara (Tanhar et al., 2023).

Famuji et al., (2024) menyoroti bahwa *Ethereum* mendukung implementasi penyimpanan data sensitif seperti data genetika melalui mekanisme *smart contract* dan integrasi dengan protokol terdistribusi seperti IPFS. Dalam studi ini, *Ethereum* berperan penting dalam efisiensi biaya *(gas fee)* dan menjamin integritas data dalam konteks *bioinformatika* (Famuji et al., 2024).

Sementara itu, Rafdinal et al. (2023) menjelaskan bahwa *Ethereum* memfasilitasi pertukaran nilai digital menggunakan token kripto seperti *Ether*. Implementasi ini digunakan dalam sistem teknologi finansial *(FinTech)* modern sebagai bentuk digitalisasi transaksi dan alat tukar dalam ekosistem tanpa otoritas pusat, menjadikannya infrastruktur penting dalam ekonomi digital terdesentralisasi (Rafdinal et al., 2023).

Dengan demikian, *Ethereum* menjadi fondasi utama dalam pengembangan teknologi berbasis *blockchain* berkat kapabilitasnya dalam menjalankan kontrak pintar, mengatur penyimpanan data terenkripsi secara terdistribusi, serta kemampuannya untuk diintegrasikan dalam berbagai konteks aplikasi, termasuk *FinTech*, *bioinformatika*, dan *metaverse*.

**6. *InterPlanetary File System* (IPFS)**

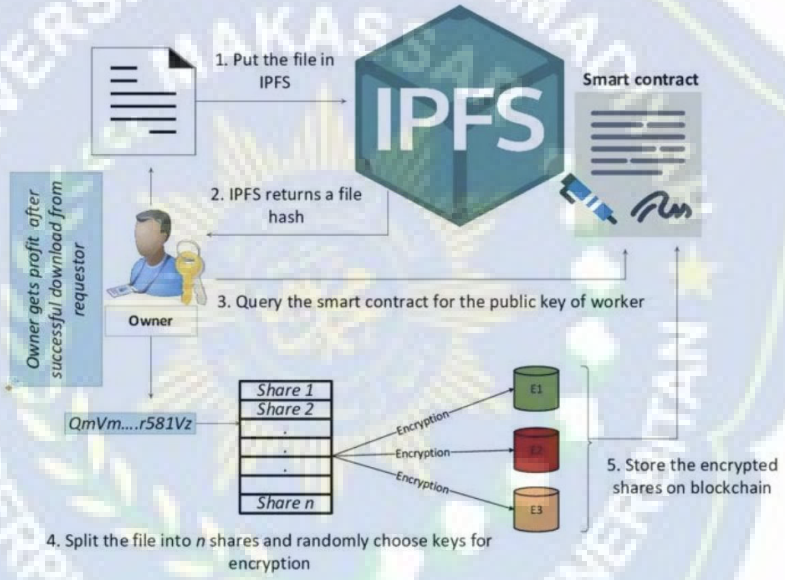
*InterPlanetary File System* (IPFS) merupakan protokol *peer-to-peer* terdistribusi yang dirancang untuk menyimpan dan berbagi konten secara permanen dan terdesentralisasi dalam sistem file global. IPFS menggantikan sistem pengalamatan berbasis lokasi (URL) dengan sistem berbasis konten *(content-addressed)*, menggunakan hash kriptografis untuk mengidentifikasi dan mengakses file.

Menurut Ahmad & Dirgahayu (2023), IPFS digunakan dalam sistem manajemen hak digital (DRM) untuk menyimpan dan mengakses file secara aman. IPFS menghasilkan hash unik untuk setiap file yang disimpan, menjamin integritas data dan meminimalkan duplikasi, sekaligus meningkatkan efisiensi dan skalabilitas sistem distribusi file (Ahmad & Dirgahayu, 2023).

Hanafi (2022) menjelaskan bagaimana IPFS diintegrasikan dalam arsitektur *digital evidence cabinet* berbasis *blockchain* untuk manajemen bukti digital. IPFS memungkinkan distribusi file bukti digital yang aman, transparan, dan tidak dapat dimodifikasi karena *hash*-nya terikat dalam *smart contract Hyperledger Fabric* (Hanafi, 2022).

Lasawedi (2023) dalam penelitiannya merancang arsitektur penyimpanan terdistribusi menggunakan IPFS yang digabungkan dengan *blockchain* untuk menciptakan sistem penyimpanan yang tidak hanya efisien dan terdesentralisasi, tetapi juga tahan terhadap sensor dan gangguan otoritas pusat (Lasawedi, 2023).

IPFS sebagai sistem file distribusi telah terbukti memperkuat prinsip *immutability*, *availability*, dan *resilience* dalam lingkungan digital berbasis *blockchain*, menjadikannya fondasi penting dalam sistem manajemen data masa depan.




*Gambar 4 Cara Kerja IPFS*

**7. *Solidity***

*Solidity* merupakan bahasa pemrograman tingkat tinggi yang dikembangkan secara khusus untuk pengembangan *smart contract* pada platform *Ethereum*. Bahasa ini bersifat *Turing-complete*, mendukung paradigma pemrograman berorientasi objek, serta dikompilasi menjadi *bytecode* untuk dijalankan di *Ethereum Virtual Machine* (EVM). Struktur sintaksnya menyerupai *JavaScript* namun dengan fitur khusus seperti

*constructor, modifier*, dan pengelolaan *state* yang bersifat *immutable* di *blockchain* (Pasqua et al., 2023).

Secara teknis, *Solidity* memfasilitasi definisi fungsi kontrak yang eksplisit, serta mendukung manajemen tipe data yang kuat dan statis. Dalam konteks pengujian, karakteristik tersebut memengaruhi cakupan pengujian dan validasi kontrak secara menyeluruh, terutama pada aspek kontrol alur program dan eksekusi fungsi yang berisiko finansial (Driessen et al., 2024). Oleh karena itu, peimmmahaman mendalam terhadap arsitektur sintaks dan semantik *Solidity* menjadi krusial dalam perancangan kontrak pintar yang aman, efisien, dan sesuai dengan praktik pengembangan terstandarisasi (Sujeetha & Preetha, 2021).

## B. Penelitian Terkait

| Peneliti | Judul | Metode | Hasil |
|---|---|---|---|
| Seni Oknora Firza, Yuhandri, Sumijan (2024) | Teknologi *Blockchain* dalam Keamanan Sertifikat Menggunakan *Smart Contracts* dan *Distributed Ledger* pada Platform Edutech | *Smart Contract*, *Distributed Ledger*, dataset dari Inatechno (48 data sertifikat peserta training). | Teknologi Blockchain meningkatkan keamanan sertifikat pada platform Edutech; sistem mengotomatisasi verifikasi dan mengurangi risiko pemalsuan. |
| Nur Khairunnisa Noorhizama, Zubaile Abdullah, Shahreen Kasim, Isredza Rahmi A Hamid, Mohd Anuar Mat Isa (2023) | *Verification of Ph.D. Certificate Using QR Code on Blockchain Ethereum* | *Solidity* (*smart contract*), PHP, HTML/CSS (web), MetaMask (integrasi *blockchain*), SHA-256 (hashing). | Implementasi dan fungsionalitas berhasil; dapat menambah, memperbarui, menghapus sertifikat, membuat/memindai kode QR, dan menerima umpan balik verifikasi |

| | | | |
|---|---|---|---|
| | | | instan; solusi kuat untuk validasi sertifikat Ph.D. |
| Mohamad Rakhmansyah, Untung Rahardja, Nuke Puji Lestari Santoso, Alfiah Khoirunisa, Adam Faturahman (2024) | *Smart Digital Signature* Berbasis *Blockchain* pada Pendidikan Tinggi Menggunakan Metode SWOT | Analisis SWOT, SHA-256, *Blockchain, Cloud storage.* | Sistem tanda tangan digital efektif dan efisien dalam mengurangi pemalsuan, pengesahan, dan meningkatkan keamanan dokumen. |
| Rizky Rachman Judhie Putra, Muhamad Nursalman, Fawwaz Kautsar (2024) | *Quality of Service Diploma Recording System Using Smart Contracts and NFT Polygon Network on Layer-2 Ethereum Blockchain* | *Smart Contracts*, IPFS, *Polygon network (Layer-2 Ethereum),* analisis QoS *(Throughput, Packet Loss, Latency).* | Jaringan Polygon menunjukkan efisiensi biaya dan performa yang lebih baik dibandingkan Ethereum (Layer-1). |

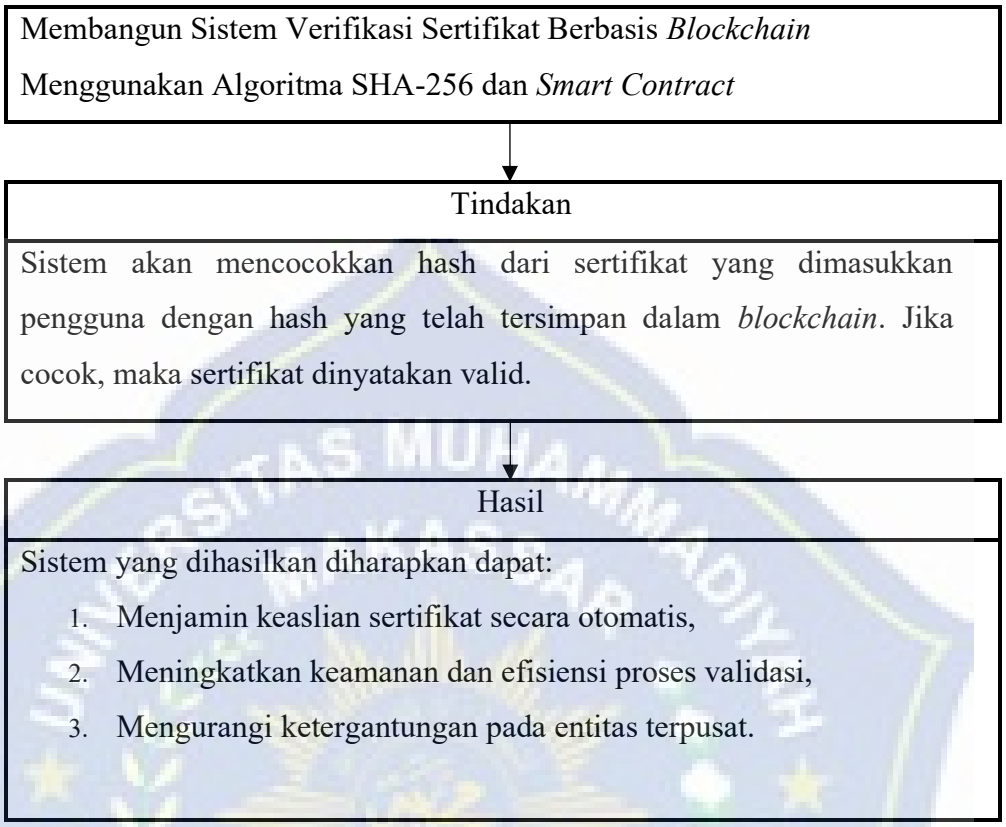
*Table 1 Penelitian Terkait*

## C. Kerangka Berpikir

| Masalah |
|---|
| Tingginya tingkat pemalsuan sertifikat digital yang berdampak pada penurunan integritas serta kepercayaan terhadap sertifikat digital yang dikeluarkan oleh institusi resmi. Sertifikat yang tidak diverifikasi dengan baik rentan terhadap duplikasi atau pemalsuan |

↓

| Solusi |
|---|




*Gambar 5 Diagram Kerangka Berpikir*

# BAB III

# METODE PENELITIAN

## A. Tempat dan Waktu Penelitian

1. Tempat Penelitian

Penelitian ini akan dilakukan di Fakultas Teknik Universitas Muhammadiyah Makassar, yang berlokasi di Gedung Iqra. Lt.3 Unismuh, Jln. Sultan Alauddin No.259, Kec. Rappocini, Gunung Sari, Kota Makassar.


*Gambar 6 Peta Universitas Muhammadiyah Makassar*

2. Penelitian ini akan dilaksanakan dalam waktu kurang lebih 3 bulan, yang akan dimulai dari minggu ke 2 bulan Juli tahun 2025 sampai september tahun 2025.

## B. Alat dan Bahan Penelitian

Adapun alat dan bahan yang akan digunakan dalam penelitian terdiri dari Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*), yaitu:

1. Perangkat Keras (*Hardware*)
   a. Laptop ASUS ROG Zephyrus G16 GU605MU
   b. HP Poco X3 Pro.
2. Perangkat Lunak (*Software*)

a. Visual Studio Code

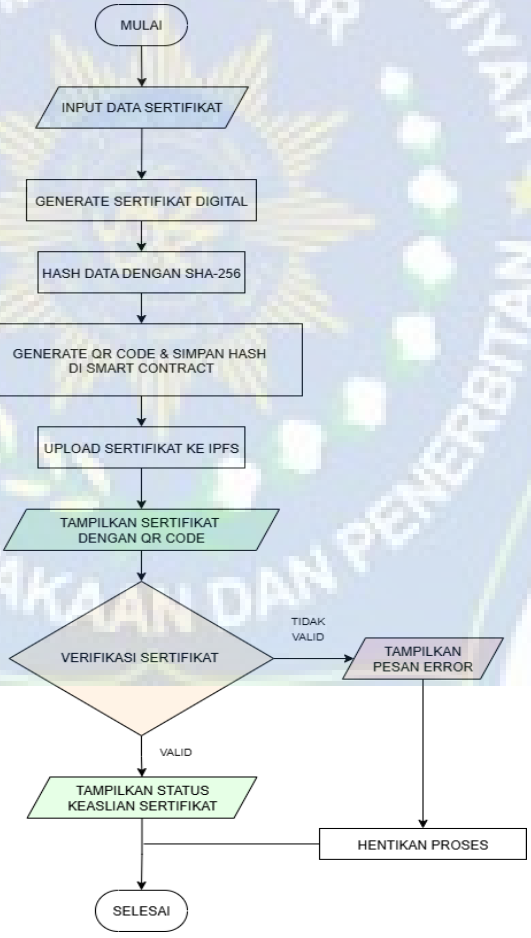
*b. Framework* hardhat

c. NodeJS dan NPM

## C. Perancangan Sistem

*1. Flowchart*

*Flowchart* sistem yang disajikan menggambarkan alur kerja mulai dari input data sertifikat hingga proses verifikasi keasliannya. Setiap tahapan dalam *flowchart* akan dibahas secara mendalam, dilengkapi dengan ilustrasi dan referensi yang relevan untuk memberikan pemahaman komprehensif mengenai implementasi teknis sistem ini.




*Gambar 7 Flowchart Perancangan Sistem*

a. Admin Login

Tahapan awal sistem dimulai dengan proses autentikasi administrator, di mana pengguna yang memiliki otorisasi harus masuk menggunakan kredensial yang telah ditetapkan sebelumnya, yaitu berupa *username* dan *password*. Proses ini bertujuan untuk menjamin bahwa hanya pihak yang berwenang yang dapat mengakses fungsi-fungsi administratif sistem, termasuk pembuatan dan validasi sertifikat digital dengan *upload* file.

b. Input Data Sertifikat

Pada tahap ini, administrator melakukan input terhadap data-data yang diperlukan untuk pembuatan sertifikat digital. Informasi yang dimasukkan mencakup identitas pemilik sertifikat seperti nama lengkap, NIM, dan jurusan, serta elemen lain yang relevan. Ketelitian dalam proses entri data sangat penting karena seluruh informasi ini akan direpresentasikan dalam sertifikat digital yang bersifat unik dan terikat secara kriptografis. Kesalahan atau ketidaklengkapan pada tahap ini dapat mengakibatkan ketidaksesuaian atau ketidakabsahan sertifikat yang dihasilkan.

c. *Generate* Sertifikat

Setelah data berhasil dikumpulkan, sistem secara otomatis menghasilkan dokumen sertifikat dalam format digital (PDF). Proses ini mencakup hashing dokumen menggunakan algoritma kriptografi SHA-256 untuk menjamin integritas file. Selanjutnya, file PDF diunggah ke jaringan penyimpanan terdesentralisasi *InterPlanetary File System* (IPFS) melalui layanan Pinata, yang berperan untuk melakukan *pinning* terhadap file agar tetap tersedia secara permanen. *Hash* hasil *SHA-256* dan *Content Identifier* (CID) dari IPFS tersebut kemudian dicatat ke dalam *smart contract* yang telah dideploy di jaringan *blockchain*. Dengan pendekatan ini, sistem menjamin bahwa :

1) Integritas dokumen dapat diverifikasi menggunakan hash yang disimpan secara on-chain.
2) Lokasi file dapat diakses secara publik melalui CID IPFS yang tercatat di *blockchain*.

Sebagai bagian dari proses, sistem juga akan menghasilkan *Quick Response Code* (QR Code) yang berisi hash dari sertifikat. QR Code ini akan digunakan sebagai representasi visual untuk memfasilitasi proses verifikasi berbasis publik secara efisien.

*d.* Verifikasi Sertifikat

Sistem mendukung dua metode verifikasi:

1) Verifikasi Internal (*Upload* File *PDF*): Hanya dapat dilakukan oleh administrator setelah proses autentikasi berhasil. File PDF akan diproses untuk menghitung ulang hash dan mencocokkannya dengan hash yang tersimpan dalam *smart contract*.
2) Verifikasi Publik (*Scan* QR Code): Dapat dilakukan oleh pihak manapun tanpa otentikasi. Sistem akan mengambil hash dari hasil pemindaian QR Code untuk memverifikasi keasliannya melalui data yang tersedia di *blockchain*.

Pendekatan ganda ini dirancang untuk mendukung fleksibilitas dan keamanan proses verifikasi.

e. Tampilkan Status Sertifikat

Sebagai hasil akhir dari proses verifikasi, sistem akan memberikan keluaran berupa status validitas sertifikat. Informasi ini disajikan dalam bentuk pesan yang eksplisit status validasi seperti "Sertifikat Valid" apabila hash cocok dan sertifikat terdaftar, atau "Sertifikat Tidak Valid" jika tidak ditemukan kecocokan. Fitur ini memberikan kepastian dan transparansi kepada pengguna mengenai keabsahan dokumen yang diverifikasi.

## D. Teknik Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsionalitas pada sistem verifikasi sertifikat berjalan sesuai dengan perancangan dan bebas dari kesalahan *(error).* Pengujian ini mencakup validasi setiap komponen, dari interaksi antarmuka hingga logika pada *smart contract*. Metode pengujian yang digunakan adalah *White Box Testing* dan *Black Box Testing*. *White Box Testing* digunakan untuk menguji struktur internal dan logika kode program. Pengujian ini fokus pada fungsi-fungsi spesifik di dalam *smart contract* dan *backend* untuk memastikan setiap alur logika berjalan dengan benar. Kemudian *Black Box Testing* digunakan untuk menguji fungsionalitas sistem dari sudut pandang pengguna tanpa perlu mengetahui detail implementasi internalnya. Pengujian ini fokus pada input dan output, memastikan alur kerja sistem secara keseluruhan *(end-to-end)* sesuai dengan kebutuhan.

Berikut adalah tahapan dan jenis pengujian yang akan dilakukan:

1. *Unit Testing*

   Pengujian ini dilakukan untuk memverifikasi setiap komponen atau unit terkecil dari perangkat lunak secara terpisah. Tujuannya adalah untuk memastikan setiap fungsi bekerja dengan benar sebelum diintegrasikan.

   *a.* Pengujian *Smart Contract*

   1) Tujuan

      Memastikan setiap fungsi dalam *smart contract* (ditulis dalam *Solidity*) berjalan sesuai ekspektasi.

   2) Skenario Pengujian

      a) Menguji fungsi penyimpanan data: Memastikan hash sertifikat dan IPFS CID berhasil disimpan di *blockchain*.

      b) Menguji fungsi pengambilan data: Memastikan data yang diambil dari *smart contract* sesuai dengan yang disimpan sebelumnya.

c) Menguji *modifier* dan hak akses: Memastikan hanya alamat (*address*) yang berwenang (admin) yang dapat menyimpan data baru.

b. Pengujian *Backend*

1) Tujuan

Memvalidasi logika bisnis di sisi *server*.

2) Skenario Pengujian

a) Menguji fungsi Hashing SHA-256: Memastikan file yang sama selalu menghasilkan hash yang identik, dan file yang berbeda menghasilkan hash yang berbeda.

b) Menguji fungsi *Generate* Sertifikat: Memastikan file PDF atau gambar sertifikat berhasil dibuat dengan data yang benar.

c) Menguji fungsi *Generate* QR Code: Memastikan QR Code yang dihasilkan berisi data (URL/hash) yang valid.

2. *Integration Testing*

Pengujian ini fokus pada interaksi antar modul atau komponen yang telah diuji secara unit.

a. Tujuan

Memastikan komunikasi antara *frontend*, *backend*, *smart contract*, dan IPFS berjalan lancar.

b. Skenario Pengujian

1) Integrasi *Frontend & Backend*: Menguji pengiriman data dari form *input* di antarmuka ke *server*, misalnya saat admin menginput data sertifikat.

2) Integrasi *Backend* & IPFS: Menguji proses *upload* file sertifikat dari *server* ke jaringan IPFS dan memastikan *server* menerima IPFS CID yang valid sebagai *respons*.

3) Integrasi *Backend & Smart Contract*: Menguji pemanggilan fungsi *smart contract* dari aplikasi. Misalnya, memastikan hash dan CID

yang dikirim dari *backend* berhasil tercatat dalam transaksi *blockchain*.

3. *End-To-End Testing*

Pengujian ini mensimulasikan skenario penggunaan nyata dari awal hingga akhir untuk memvalidasi alur kerja sistem secara keseluruhan. Pengujian ini menggunakan metode *Black Box*.

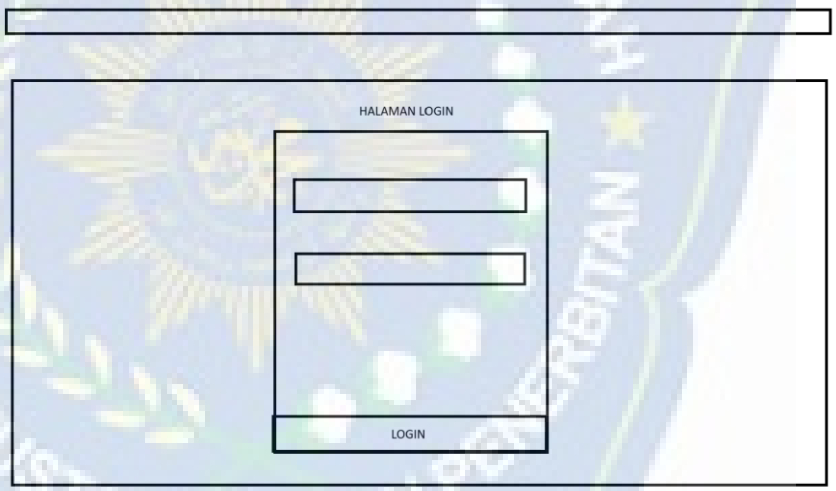
a. Tujuan

Memastikan seluruh alur proses, dari pembuatan sertifikat hingga verifikasi, berjalan sesuai dengan *flowchart*.

b. Skenario Pengujian Utama

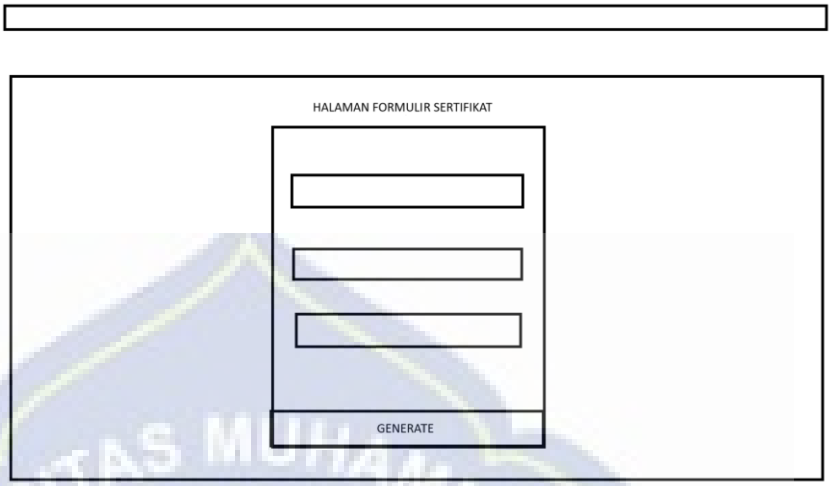
1) Alur Penerbitan Sertifikat:

a) Admin login ke sistem.




*Gambar 8 Halaman Admin Login*

b) Admin mengisi formulir data sertifikat dan menekan tombol *"Generate"*.



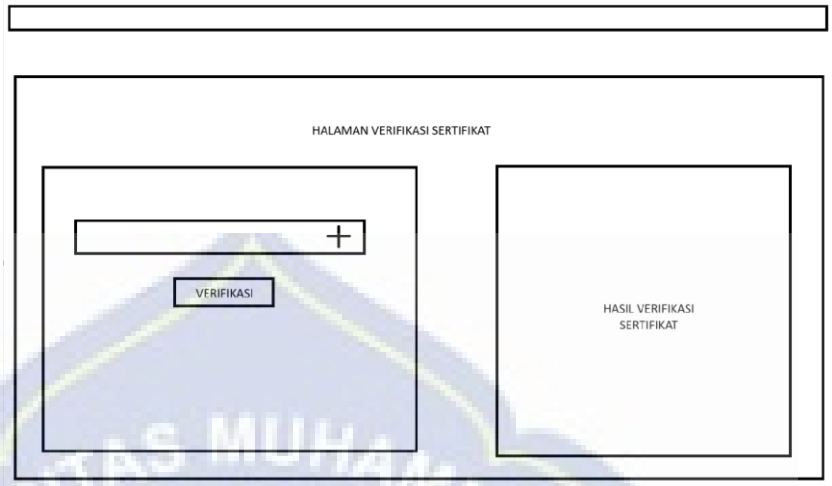

*Gambar 9 Halaman Admin Generate Sertifikat*

c) Sistem berhasil menghasilkan sertifikat digital, mengunggahnya ke IPFS via Pinata, menyimpan hash & CID di *smart contract*, dan menampilkan sertifikat final dan juga menghasilkan QR Code terpisah.


*Gambar 10 Contoh Sertifikat Hasil Generate*


*Gambar 11 Contoh QR Code Hasil Generate*

2) Alur Verifikasi Sertifikat (*upload* file):
   a) Admin login ke sistem.
   b) Sistem membuka dashboard, kemudian admin memiliih menu verifikasi.
   c) Admin upload file sertifikat PDF ke dalam sistem
   d) Sistem menampilkan status sertifikat, jika valid akan menampilkan status "SERTIFIKAT VALID" beserta data sertifikat.
   e) Jika tidak valid atau sertifikatnya palsu, sistem akan menampikan status “SERTIFIKAT TIDAK VALID”.
3) Alur Verifikasi Sertifikat (scan *QR Code*):
   a) User lain atau publik melakukan scan pada *QR Code.*
   b) Sistem akan melakukan perbandingan hash antara *hash* hasil generate sertifikat dan hash dari isi *QR Code*.
   c) Jika *hash* nya cocok atau valid, maka akan menampilkan "SERTIFIKAT VALID" beserta data sertifikat.
   d) Jika tidak cocok atau *hashnya* tidak valid, maka sistem akan menampikan status “SERTIFIKAT TIDAK VALID”.




*Gambar 12 Halaman Verifikasi Sertifikat*

# BAB IV

# HASIL DAN PEMBAHASAN

## A. Hasil Implementasi Sistem

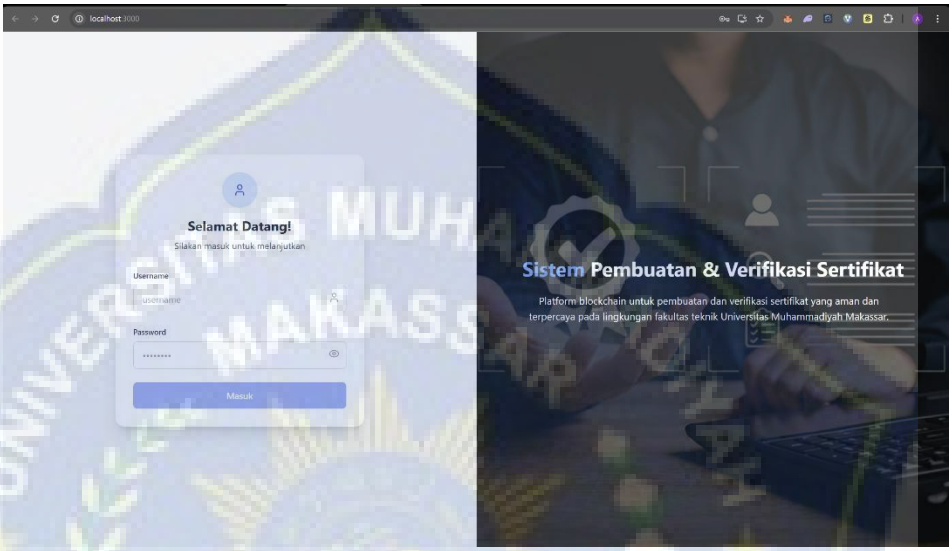
Hasil implementasi sistem menggambarkan realisasi dari rancangan yang telah disusun pada tahap perancangan sebelumnya, mencakup pengembangan, konfigurasi, dan integrasi seluruh komponen sistem verifikasi sertifikat digital berbasis *blockchain*. Pada tahap ini, sistem dibangun dengan mengintegrasikan *smart contract* untuk penyimpanan hash sertifikat dan *Content Identifier* (CID), algoritma SHA-256 untuk memastikan integritas sertifikat, serta IPFS melalui layanan Pinata sebagai media penyimpanan terdistribusi. Seluruh proses implementasi dilakukan secara bertahap, dimulai dari pengembangan sisi *backend* untuk mengelola logika bisnis dan konektivitas ke *blockchain*, pengembangan *frontend* sebagai antarmuka pengguna, hingga pengujian integrasi sistem secara menyeluruh.

Bagian ini menyajikan uraian rinci mengenai hasil penerapan setiap modul sistem, meliputi antarmuka administrator, proses *generate* sertifikat digital, mekanisme penyimpanan data terdesentralisasi, serta prosedur verifikasi publik baik melalui pemindaian QR Code maupun unggah berkas sertifikat. Dokumentasi hasil implementasi disertai dengan tangkapan layar (screenshot) setiap komponen, sehingga memudahkan proses evaluasi dan analisis kesesuaian sistem terhadap spesifikasi yang telah ditetapkan. Berikut hasil implementasi, yaitu:

1. Admin Login

Proses admin login dirancang guna memastikan hanya pengguna dengan kredensial yang sah yang diperbolehkan mengakses fitur administratif. Sistem autentikasi menggunakan mekanisme *JSON Web Token* (JWT), yang memberikan lapisan keamanan tambahan dalam proses login admin. JWT dipilih karena kemampuannya dalam mengamankan data

melalui proses enkripsi, yang memastikan token autentikasi tidak mudah diretas atau dimanipulasi. Berikut ditampilkan antarmuka halaman autentikasi admin:

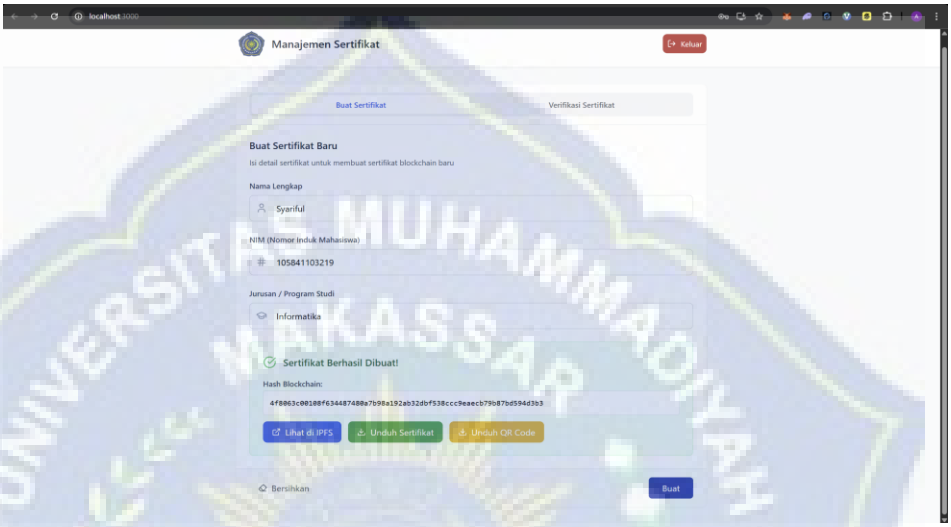



Gambar 13 Halaman Login Admin

2. *Generate* Sertifikat

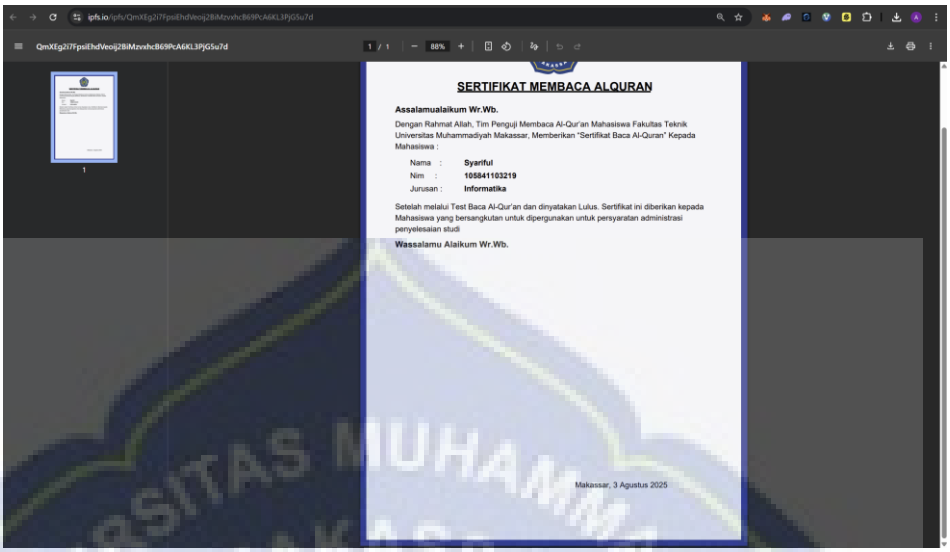
Setelah proses autentikasi berhasil, admin memasukkan data sertifikat yang mencakup nama, NIM, serta jurusan mahasiswa. Informasi ini digunakan untuk menghasilkan dokumen sertifikat dalam format PDF. Proses berikutnya adalah mengamankan dokumen dengan hashing menggunakan algoritma SHA-256 guna memastikan integritas dokumen tetap terjaga dari segala bentuk manipulasi data. SHA-256 dipilih karena menghasilkan nilai hash unik dengan panjang 256-bit yang menjamin setiap perubahan kecil dalam dokumen akan menghasilkan hash yang berbeda secara signifikan. Selanjutnya, dokumen PDF diunggah ke jaringan penyimpanan terdesentralisasi IPFS melalui layanan Pinata, yang memberikan keunggulan dalam distribusi dan ketersediaan file secara global tanpa bergantung pada satu titik kegagalan. *Content Identifier* (CID)

dari IPFS dan hasil hashing sertifikat kemudian dicatat dalam *blockchain Ethereum* melalui *smart contract* yang dikembangkan menggunakan bahasa Solidity. Berikut disajikan contoh hasil sertifikat digital, CID IPFS, serta transaksi pada *blockchain* :

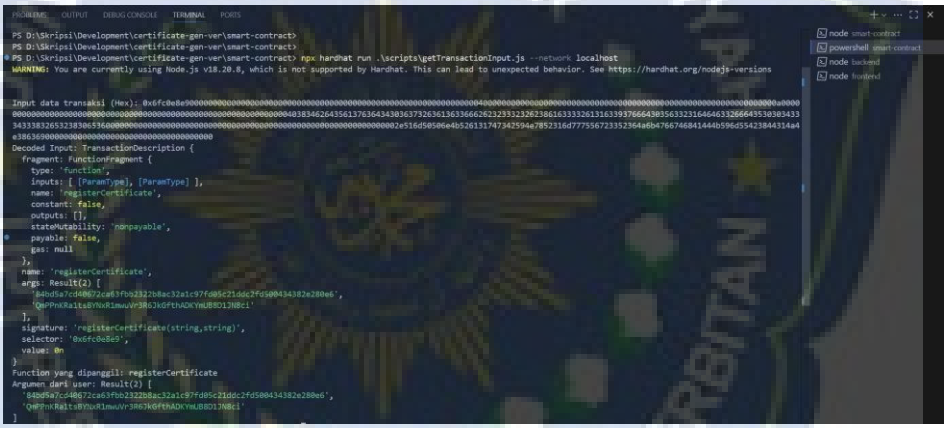



Gambar 14 Tampilan Proses Generate Sertifikat Berhasil


Gambar 15 Contoh Sertifikat Digital Hasil Generate



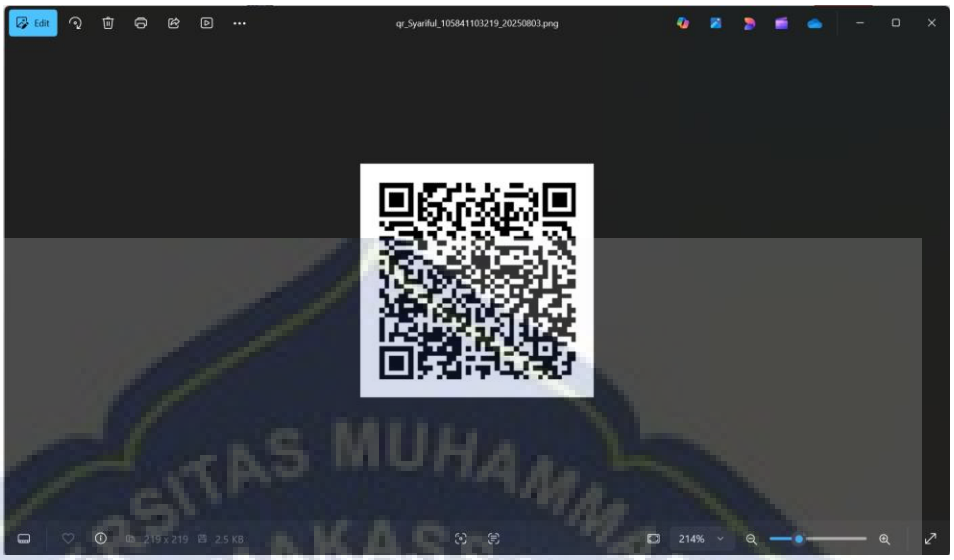

Gambar 16 Tampilan CID IPFS




Gambar 17 Tampilan Transaksi Blockchain

3. QR Code *Generation*

Untuk memudahkan proses verifikasi publik, sistem juga menghasilkan QR *Code* yang berisi informasi *hash* sertifikat. QR Code ini sebagai representasi visual yang dapat dengan mudah dipindai oleh perangkat seluler untuk verifikasi cepat. QR Code ini membantu proses validasi sertifikat yang lebih efisien dibandingkan metode manual.




Gambar 18 QR Code Hasil Generate

4. Verifikasi Sertifikat

a. Verifikasi Melalui Upload File Sertifikat (Admin)

Verifikasi melalui unggah file disediakan secara khusus bagi administrator atau pihak berwenang, dengan tujuan menguji keaslian dokumen sertifikat secara menyeluruh berdasarkan file asli yang diunggah ke sistem. Berikut tahapan yang dilakukan:

1) Unggah File Sertifikat

Administrator mengunggah file sertifikat (format PDF) ke dalam sistem melalui antarmuka *backend*. Proses ini dilakukan melalui endpoint khusus yang hanya dapat diakses oleh admin terautentikasi.

2) Esktraksi dan Hashing Dokumen

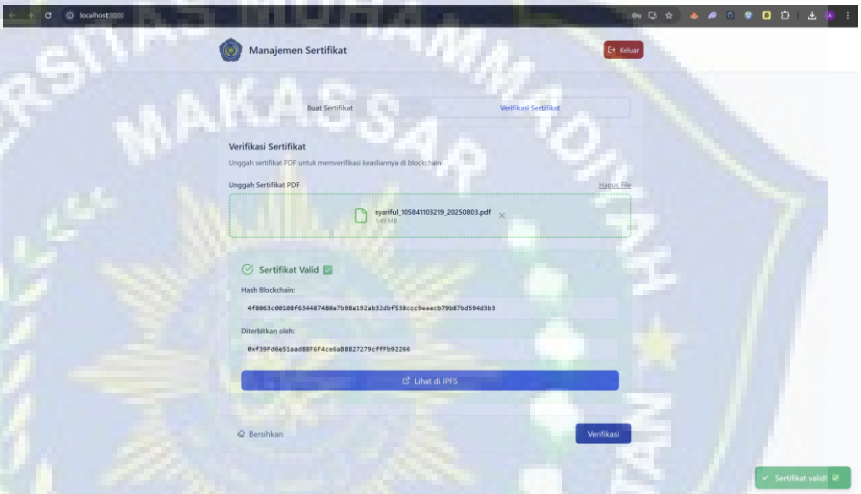
Backend akan mengekstrak isi dokumen sertifikat yang diunggah dan melakukan proses hashing menggunakan algoritma SHA-256. Hasil hash ini akan menjadi fingerprint digital dari file sertifikat yang bersangkutan

3) Validasi Hash Pada Smart Contract

Hasil hash dari dokumen kemudian dikirimkan ke smart contract untuk dibandingkan dengan hash yang telah tercatat pada blockchain Ethereum saat pendaftaran awal sertifikat.

4) Penyajian Hasil Verifikasi

Jika hash cocok dengan data di blockchain, sistem akan menampilkan metadata sertifikat terkait. Jika hash tidak ditemukan, admin diberi notifikasi bahwa file sertifikat tersebut tidak valid atau tidak terdaftar pada sistem.




Gambar 19 Hasil Verifikasi Melalui Upload File

b. Verifikasi Melalui Scan QR Code (Publik)

Proses verifikasi publik dirancang agar mudah diakses oleh masyarakat umum, institusi, maupun pihak ketiga tanpa memerlukan akses khusus ke sistem backend. Pada metode ini, QR Code yang tercetak pada sertifikat fisik maupun digital mengandung data hash sertifikat yang telah dihasilkan dan disimpan pada saat pembuatan dokumen. Langkah-langkah verifikasi sebagai berikut:

1) Pemindaian QR Code

Pengguna menggunakan perangkat seluler atau alat pemindai untuk membaca QR Code pada sertifikat. Sistem akan

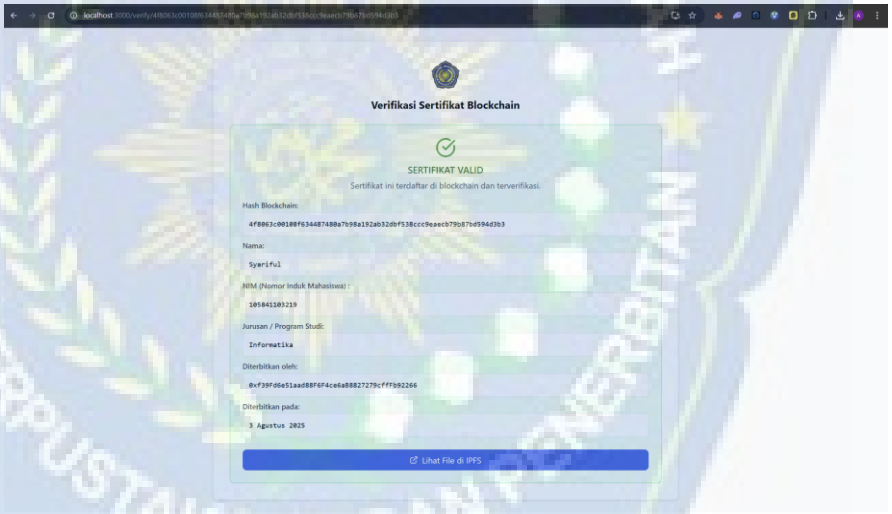
mengekstrak data hash dari QR Code dan mengarahkan pengguna ke halaman verifikasi.

2) Verifikasi Hash Pada *Blockchain*

*Backend* sistem akan menerima hash tersebut dan melakukan query ke *smart contract* di jaringan *blockchain Ethereum*. Proses ini dilakukan secara otomatis tanpa perlu input manual dari pengguna.

3) Pengambilan dan Penyajian Data Sertifikat

Jika hash terdaftar dan valid, sistem menampilkan detail metadata sertifikat (nama, NIM, jurusan, tanggal terbit, dan CID IPFS) langsung di antarmuka publik. Jika tidak ditemukan, sistem akan memberikan notifikasi bahwa sertifikat tidak terdaftar.




Gambar 20 Hasil Verifikasi Melalui *Scan* QR Code

## B. Analisis Hasil Uji Sistem

### *1. Unit Testing*

Sebelum sistem diintegrasikan secara menyeluruh, dilakukan pengujian unit (*unit testing*) untuk menilai keandalan dan validitas fungsi-fungsi utama pada *smart contract*. Pengujian ini difokuskan pada verifikasi

setiap fitur secara terpisah guna memastikan implementasi logika bisnis sesuai dengan spesifikasi rancangan dan bebas dari kesalahan logika. Ruang lingkup pengujian mencakup mekanisme pencatatan data sertifikat, proses verifikasi data yang telah disimpan, serta pengujian efektivitas pembatasan akses melalui *modifier*. Hasil dari masing-masing pengujian unit *smart contract* tersebut dapat dilihat pada tabel berikut:

| Fitur yang Diuji | Langkah Pengujian | Proses yang Diharapkan | Status |
|---|---|---|---|
| Penyimpanan Data (*Register Certificate*) | 1. *Deploy smart contract* di jaringan test (Hardhat).<br>2. Panggil fungsi untuk menyimpan hash sertifikat dan IPFS CID | Hash sertifikat dan CID IPFS berhasil tersimpan di *blockchain*, *event* terekam dengan data yang sesuai | ✓Berhasil |
| Pengambilan Data (*Verify Certificate)* | 1. Simpan data hash dan CID pada *smart contract*.<br>2. Ambil data dengan fungsi verifikasi menggunakan hash yang sama. | Data yang diambil (hash, CID, *issuer*, validitas) sesuai dengan yang telah disimpan sebelumnya. | ✓Berhasil |
| Hak Akses/*Modifier* | 1. Coba simpan data menggunakan akun admin.<br>2. Coba simpan data menggunakan akun non-admin | Hanya admin yang dapat menyimpan data baru; non-admin mendapatkan *error*. | ✓Berhasil |

Tabel 2 Hasil Pengujian *Smart Contract*

Selanjutnya pada sisi backend dilakukan secara terstruktur untuk memastikan setiap fungsi utama dapat berjalan sesuai dengan rancangan. Pengujian ini mencakup proses hashing file sertifikat menggunakan algoritma SHA-256, pembuatan file PDF sertifikat berdasarkan data yang di input, pembuatan dan validasi QR code, serta penyimpanan metadata sertifikat. Rangkuman hasil pengujian unit untuk komponen *backend* disajikan pada tabel berikut:

| Fitur yang Diuji | Langkah Pengujian | Proses yang Diharapkan | Status |
|---|---|---|---|
| Hashing SHA-256 | Lakukan hashing pada file sertifikat PDF | Hasil hash konsisten dan unik untuk tiap file, berubah jika file diubah | ✓Berhasil |
| *Generate* Sertifikat | 1. Masukkan data sertifikat (nama, NIM, jurusan).<br>2. Proses *generate* file PDF sertifikat.<br>3. Periksa hasil file dan isinya sesuai data yang di input. | File PDF sertifikat berhasil dibuat dan isinya sesuai data yang diberikan. | ✓Berhasil |
| *Generate* QR Code | 1. *Generate* QR Code dari data tertentu (misal URL/hash).<br>2. Scan QR Code dengan aplikasi pembaca.<br>3. Verifikasi hasil *scan* sesuai data asli yang di input ke *generator*. | QR Code yang dihasilkan berisi data yang valid dan dapat dibaca sesuai data sumber (URL/hash sertifikat). | ✓Berhasil |

| | | | |
|---|---|---|---|
| Penyimpanan Metadata | Simpan metadata sertifikat ke file *cert-metadata.json* | Metadata (nama, NIM, jurusan, CID, hash) tersimpan dengan benar | ✓Berhasil |

Tabel 3 Hasil Pengujian Backend

2. *Integration Testing*

Pengujian integrasi dilakukan untuk memastikan bahwa setiap komponen sistem dapat berinteraksi secara sinergis dan berfungsi dengan baik dalam satu alur kerja terpadu. Pengujian ini difokuskan pada integrasi antara *frontend* dan *backend*, *backend* dengan layanan IPFS melalui Pinata, serta *backend* dengan *smart contract* pada *blockchain*. Setiap skenario pengujian dirancang untuk memverifikasi kelancaran proses pertukaran data, validitas hasil integrasi, dan ketahanan sistem terhadap potensi *error* komunikasi. Hasil pengujian integrasi dari masing-masing komponen dapat dilihat pada tabel berikut:

| Fitur yang Diuji | Langkah Pengujian | Proses yang Diharapkan | Status |
|---|---|---|---|
| Integrasi Frontend & Backend | 1. Admin mengisi form input data sertifikat pada frontend.<br>2. *Frontend* mengirim data ke *endpoint backend.*<br>3. *Backend* menerima dan memproses data. | Data dari *frontend* berhasil diterima *backend*, data valid tersimpan, tidak terjadi *error* komunikasi. | ✓Berhasil |
| Integrasi *Backend* & IPFS | 1. *Backend* menerima file sertifikat.<br>2. *Backend* | File berhasil di-*upload* ke IPFS dan server menerima CID yang valid, | ✓Berhasil |

| | | | |
|---|---|---|---|
| | melakukan *upload* file ke jaringan IPFS melalui Pinata.<br>3. Server menerima IPFS CID sebagai *respons upload*. | CID diteruskan untuk proses berikutnya. | |
| Integrasi *Backend* & *Smart Contract* | 1. Backend memanggil fungsi *smart contract* untuk menyimpan hash & CID.<br>2. Hash dan CID dikirim ke *blockchain*.<br>3. Pastikan data tercatat di *blockchain*. | Fungsi *smart contract* berhasil dipanggil, hash & CID tercatat di *blockchain*, transaksi sukses tanpa *error*. | ✓Berhasil |

Tabel 4 Hasil Pengujian Integrasi

3. *End-To-End Testing*

Pengujian *end-to-end* dilaksanakan untuk memastikan bahwa seluruh alur proses bisnis berjalan dengan lancar mulai dari tahap pembuatan hingga verifikasi sertifikat digital. Pengujian ini bertujuan untuk mengevaluasi performa sistem secara keseluruhan dalam skenario nyata, mulai dari interaksi pengguna pada *frontend*, pemrosesan data di *backend*, integrasi dengan layanan eksternal (IPFS dan *blockchain*), hingga validasi keaslian sertifikat baik melalui unggah file maupun *scan* QR Code. Setiap tahapan diuji secara menyeluruh guna memastikan bahwa sistem dapat memenuhi kebutuhan fungsional secara komprehensif dan mampu menangani skenario penggunaan yang relevan. Rangkuman hasil pengujian *end-to-end* sistem disajikan pada tabel berikut:

| Fitur yang Diuji | Langkah Pengujian | Proses yang Diharapkan | Status |
|---|---|---|---|
| Generate Sertifikat Digital | 1. Admin login ke sistem.<br>2. Admin mengisi form data sertifikat & klik "Generate".<br>3. Sistem generate sertifikat digital (PDF), hash & upload ke IPFS via Pinata, simpan hash & CID di blockchain, dan generate QR Code. | Sistem berhasil membuat sertifikat, mengunggah ke IPFS, mencatat hash & CID di smart contract. | ✓Berhasil |
| Verifikasi Sertifikat (*Upload* File) | 1. Admin login ke sistem.<br>2. Admin membuka dashboard & menu verifikasi.<br>3. Admin *upload* file PDF sertifikat.<br>4. Sistem membandingkan hash & menampilkan status verifikasi. | Sistem dapat memvalidasi keaslian sertifikat berdasarkan hash. Jika valid: tampil status "SERTIFIKAT VALID" beserta data; jika tidak valid: "SERTIFIKAT TIDAK VALID". | ✓Berhasil |
| Verifikasi Sertifikat (*Scan* QR Code) | 1. *User* melakukan *scan* QR Code pada sertifikat.<br>2. Sistem membandingkan hash hasil *generate* & hash di QR Code. | Jika hash cocok, tampil "SERTIFIKAT VALID" beserta data. Jika tidak cocok, tampil "SERTIFIKAT TIDAK VALID". | ✓Berhasil |

| | | | |
|---|---|---|---|
| | 3. Sistem menampilkan hasil validitas & data sertifikat. | | |

Tabel 5 Hasil Pengujian End To End

## C. Evaluasi Kinerja Sistem

Evaluasi kinerja sistem dilakukan untuk mengukur efektivitas dan efisiensi implementasi sistem verifikasi sertifikat berbasis *blockchain* yang terintegrasi dengan IPFS melalui Pinata. Pengujian difokuskan pada beberapa aspek utama, meliputi kecepatan transaksi, integritas data, efektivitas kontrol akses, keberhasilan integrasi penyimpanan terdistribusi, konsistensi hasil verifikasi, serta responsivitas sistem dalam melayani permintaan verifikasi. Hasil pengujian masing-masing aspek dirangkum dalam tabel berikut:

| Aspek Evaluasi | Parameter yang di nilai | Hasil Evaluasi | Interpretasi Hasil |
|---|---|---|---|
| Kecepatan Transaksi *Blockchain* | Waktu konfirmasi transaksi (*deployment* dan *register* certificate) | Rata-rata < 8 detik (jaringan lokal Hardhat) | Proses penyimpanan data pada *smart contract* berlangsung cepat, mendukung efisiensi sistem. |
| Keberhasilan Penyimpanan Data | Konsistensi hash & CID di blockchain | 100% data hash dan CID yang didaftarkan berhasil direkam dan diverifikasi ulang dengan hasil yang sama | Menunjukkan integritas dan keandalan penyimpanan data secara desentralisasi. |
| Ketahanan Hak Akses | Pembatasan fungsi *register* (admin vs non-admin) | Non-admin gagal melakukan registrasi, hanya admin yang berhasil | Mekanisme kontrol akses berjalan efektif, menjaga otoritas dan keamanan data. |

| | | | |
|---|---|---|---|
| Keberhasilan *Upload* ke IPFS (Pinata) | Presentase file berhasil diunggah dan memperoleh CID | 100% file PDF sertifikat yang diuji berhasil diunggah dan mendapat CID | Proses integrasi *storage* terdistribusi berjalan optimal, file dapat diakses melalui IPFS. |
| Konsistensi Pengambilan Data | Kesesuaian data hasil verifikasi | Data hasil verifikasi identik dengan data saat registrasi | Fungsi verifikasi mampu menjaga konsistensi dan akurasi informasi sertifikat digital. |
| Waktu Respons Verifikasi | Latensi respon verifikasi (*get data)* | Rata-rata < 5 detik pada pengujian lokal | Sistem mampu memberikan hasil verifikasi secara *real-time*, mendukung penggunaan publik. |

Tabel 6 Hasil Evaluasi Kinerja Sistem

# BAB V
# PENUTUP

## A. Kesimpulan

Berdasarkan rangkaian proses perancangan, implementasi, serta pengujian sistem verifikasi sertifikat digital berbasis *blockchain* yang telah dilaksanakan, dapat disimpulkan bahwa sistem yang dikembangkan mampu mengakomodasi kebutuhan validasi keaslian sertifikat secara efisien, aman, dan transparan. Integrasi algoritma SHA-256 untuk proses *hashing* sertifikat memastikan integritas dan orisinalitas sertifikat, di mana setiap perubahan pada file sertifikat akan menghasilkan nilai hash yang berbeda sehingga potensi pemalsuan dapat diidentifikasi secara otomatis. Penerapan smart contract pada jaringan blockchain berperan signifikan dalam menjaga desentralisasi serta keandalan penyimpanan data hash dan *Content Identifier* (CID) hasil unggahan ke IPFS melalui Pinata. Selain itu, mekanisme kontrol akses berbasis *modifier* pada *smart contract* terbukti efektif dalam membatasi kewenangan administratif, sehingga hanya pihak yang berhak yang dapat melakukan pencatatan data baru ke *blockchain*. Seluruh hasil pengujian, baik unit, integrasi, maupun *end-to-end*, menunjukkan performa sistem yang responsif, akurat, dan bebas dari *error* kritis, sehingga sistem ini layak diadopsi sebagai solusi verifikasi sertifikat digital yang mengedepankan aspek keamanan, transparansi, dan desentralisasi.

## B. Saran

Bagian saran memuat rekomendasi pengembangan yang dirumuskan oleh penulis berdasarkan temuan selama proses perancangan, implementasi, dan pengujian sistem verifikasi sertifikat digital berbasis *blockchain*. Saran ini disusun untuk memberikan arahan bagi penelitian lanjutan maupun pengembangan sistem di masa mendatang, sehingga sistem tidak hanya berfungsi sesuai spesifikasi awal, tetapi juga mampu beradaptasi terhadap

kebutuhan pengguna dan perkembangan teknologi. Rekomendasi yang diberikan dari penulis, yaitu :

1. Penambahan Fitur Pemilihan Jenis Sertifikat

   Sistem perlu dikembangkan agar administrator dapat memilih jenis sertifikat yang akan dihasilkan, seperti sertifikat pelatihan mengaji, sertifikat kegiatan ilmu falaq, seminar, workshop, atau kegiatan akademik lainnya. Fitur ini akan meningkatkan fleksibilitas dan efisiensi proses penerbitan sertifikat sesuai dengan jenis kegiatan yang diselenggarakan.

2. Penambahan Fitur Pembuatan Sertifikat Massal dari File *Excel*

   Disarankan untuk menambahkan mekanisme pembuatan sertifikat secara massal melalui impor data mahasiswa dari file *Microsoft Excel* (.xlsx). Dengan fitur ini, administrator tidak perlu melakukan proses *generate* satu per satu, sehingga dapat menghemat waktu dan sumber daya, khususnya pada kegiatan dengan jumlah peserta yang besar.

3. Pengembangan Menggunakan Private Blockchain

   Pada pengembangan berikutnya, sistem dapat diimplementasikan pada jaringan *blockchain* privat, seperti *Hyperledger Fabric*, untuk mendukung kebutuhan institusional yang mengutamakan kontrol akses, privasi data, serta efisiensi biaya transaksi.

## DAFTAR PUSTAKA


Ahmad, J. A., & Dirgahayu, T. (2023). The Role of Blockchain to Solve Problems of Digital Right Management (DRM). Jurnal Teknik Informatika (JUTIF).

Azimi, S., Golzari, A., Ivaki, N., & Laranjeiro, N. (2025). A systematic review on smart contracts security design patterns. Empirical Software Engineering, 30, 95. https://doi.org/10.1007/s10664-025-10646-w

Baso, F., Yusuf, D. U., Djaoe, A. N. M., Iswandi, & Ramadhany, A. (2024). The overview of smart contract: Legality and enforceability. Dialogia Iuridica, 16(1), 96–111.

Driessen, S.W., Di Nucci, D., & Tamburri, D.A. (2024). SolAR: Automated test-suite genera tion for Solidity smart contracts. Journal of Computer Programming.

Famuji, T. S., Herman, H., & Sunardi, S. (2024). Smart Contract Penyimpanan Data Genetika Manusia Berbiaya Murah pada Blockchain Ethereum. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK).

Firza, S. O., & Yuhandri, Y. (2024). Teknologi Blockchain dalam Keamanan Sertifikat Menggunakan Smart Contracts dan Distributed Ledger pada Platform Edutech. *Jurnal Penerapan Sistem Informasi*, 7(2).

FONICA. (2021). Digital Certificate Verification Scheme for Smart Grid using Fog Computing. *Sustainability*, 13(5), 2549.

Goffar, E. A., & Arifah, R. P. N. (2023). Certificate Digitalization for Information Security in Verification. *ASTRAtech Proceedings*, 4(1).

Halimi, A., Tholib, A., & Yaqin, M. A. (2024). Optimasi Keamanan Data Penerimaan Mahasiswa Menggunakan AES-256, SHA-256, dan Base64. Justify: Jurnal Sistem Informasi Ibrahimy, 3(1), 38–45. https://doi.org/10.35316/justify.v3i1.5107

Hanafi, J. (2022). InterPlanetary File System pada Digital Evidence Cabinet berbasis Hyperledger Fabric untuk Manajemen Bukti Digital. Universitas Islam Indonesia.

Hardiyanto, N., Rafdinal, W., & Juniarti, C. (2023). Financial Technology In The New Era: Cryptocurrency.

Joosten, M., Santosa, D., & Rahmadani, R. (2021). Mengeksplorasi Teknologi Blockchain: Konsensus, Keamanan, dan Implementasi. Nusantara Journal of Multidisciplinary Science, 1(3), 45–52.

Lasawedi, M. F. A. (2023). Rancang Bangun Penyimpanan Terdistribusi Menggunakan Blockchain Dengan Protokol Interplanetary Filesystem (IPFS). Politeknik Negeri Ujung Pandang.

Lestari, D. A., Hartono, R., & Widodo, D. (2024). Potensi, Tantangan, dan Implementasi Blockchain untuk Pengembangan Aplikasi dalam Era Digital Modern. Kohesi: Jurnal Sains dan Teknologi, 3(1), 22 34.

Luhkito, B. R., Fadillah, A. R., & Nugraha, R. (2021). Implementasi Blockchain pada Peer-to-Peer Transaction menggunakan Algoritma U-Quark. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK), 5(12).

Nainggolan, S. (2022). Implementasi Algoritma SHA-256 Pada Aplikasi Duplicate Document Scanner. RESOLUSI: Rekayasa Teknik Informatika dan Informasi, 2(5), 201–213.

Nanda Sari, A., & Gelar, M. (2024). Blockchain: Teknologi dan Implementasinya. Jurnal Mnemonic,Politeknik Negeri Bandung.

Pasqua, M., Benini, A., Contro, F., & Crosara, M. (2023). Enhancing Ethereum smart-contracts static analysis by computing a precise Control-Flow Graph of Ethereum bytecode. Journal of Systems and Software.

Rakhmansyah, M., Rahardja, U., & Santoso, N. P. L. (2021). Smart Digital Signature Berbasis Blockchain pada Pendidikan Tinggi menggunakan Metode SWOT. *ADI Bisnis Digital*, 2(1), 12–22.

Sari, M., & Nasution, H. (2024). Implementasi Teknologi Blockchain dalam Uang Digital: Potensi dan Dampaknya terhadap Sistem Keuangan Global. Jurnal Ilmiah Ekonomi dan Manajemen, 7(2), 15–25.

Sintyaningrum, D. E., Aryanto, F. E., & Herwanto, E. (2022). Implementasi Digital Signature dan QR Code pada Sertifikat Profesi Digital untuk Mengatasi Kasus Pemalsuan Sertifikat. *Jurnal TEKNO*, 29(2).

Sitorus, N., Sinaga, J. S. G., & Samosir, S. L. (2024). Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2. Jurnal Quancom, 2(2), 9–16. https://doi.org/10.62375/jqc.v2i2.432

Suhendra, I., & Maslan, A. (2024). Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi Secure Hash Algorithm. Jurnal Comasie, 10(5), 1–6.

Sujeetha, R., & Preetha, C.A.S.D. (2021). A Literature Survey on Smart Contract Testing and Analysis for Smart Contract Based Blockchain Application Development. 2nd Int. Conference on Smart Electronics and Communication.

Susanto, B. H., Masrek, M. N., & Khairuddin, I. E. (2022). Implementation of smart contract technology in financial services institutions. Environment-Behaviour Proceedings Journal, 7 (SI10), 249–254. https://doi.org/10.21834/ebpj.v7iSI10.4129

Sutopo, S. F., Marwati, R., & Kustiawan, C. (2019). Implementasi Digital Signature Algorithm (DSA) Menggunakan Secure Hash Algorithm-256 (SHA-256) pada Media Gambar. Jurnal Eureka Matika, 7(2), 30–36.

Taherdoost, H. (2023). Smart contracts in blockchain technology: A critical review. Information, 14(2), 117. https://doi.org/10.3390/info14020117

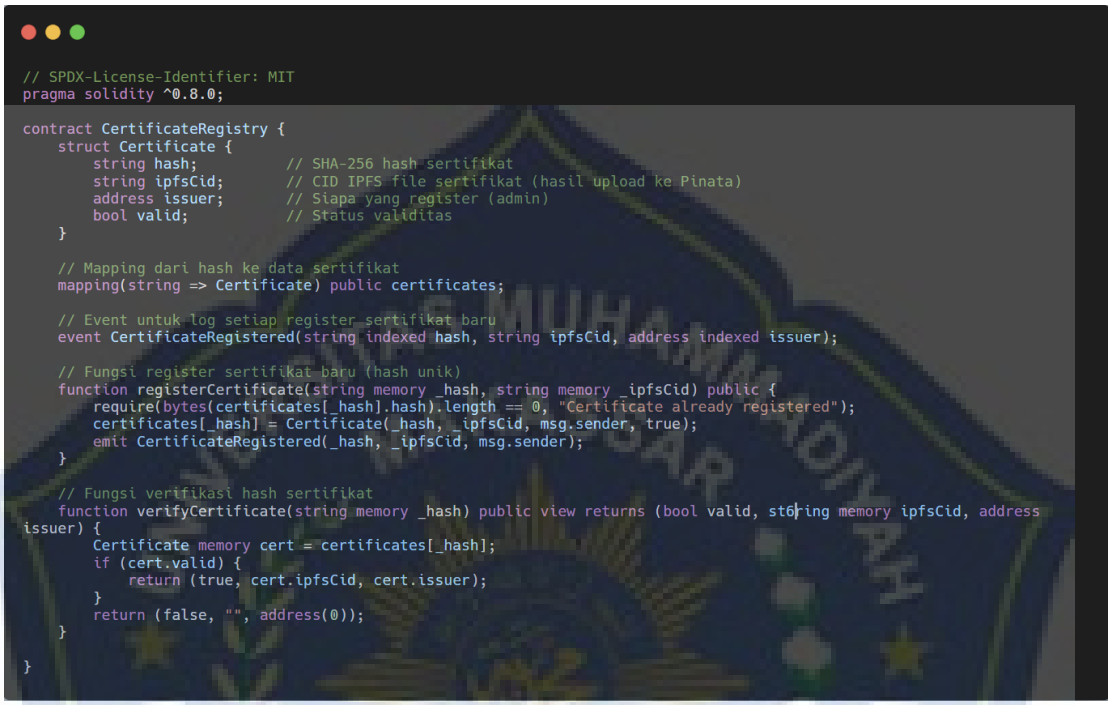
Tanhar, A. C., Hariadi, M., & Sumpeno, S. (2023). Sistem Data Sharing Berbasis Blockchain untuk Audio Player di Metaverse.

Wijayanto, H. (2024). Aplikasi Verifikasi Sertifikat Berbasis Website Menggunakan Blockchain. *Infact: International Journal of Computers*, 3(1), 44–52.

Ye, X., Zeng, N., & König, M. (2022). Systematic literature review on smart contracts in the construction industry: Potentials, benefits, and challenges. Frontiers of Engineering Management, 9(2), 196–213. https://doi.org/10.1007/s42524-022-0188-2

# LAMPIRAN

Lampiran 1. *Source Code*Smart Contract



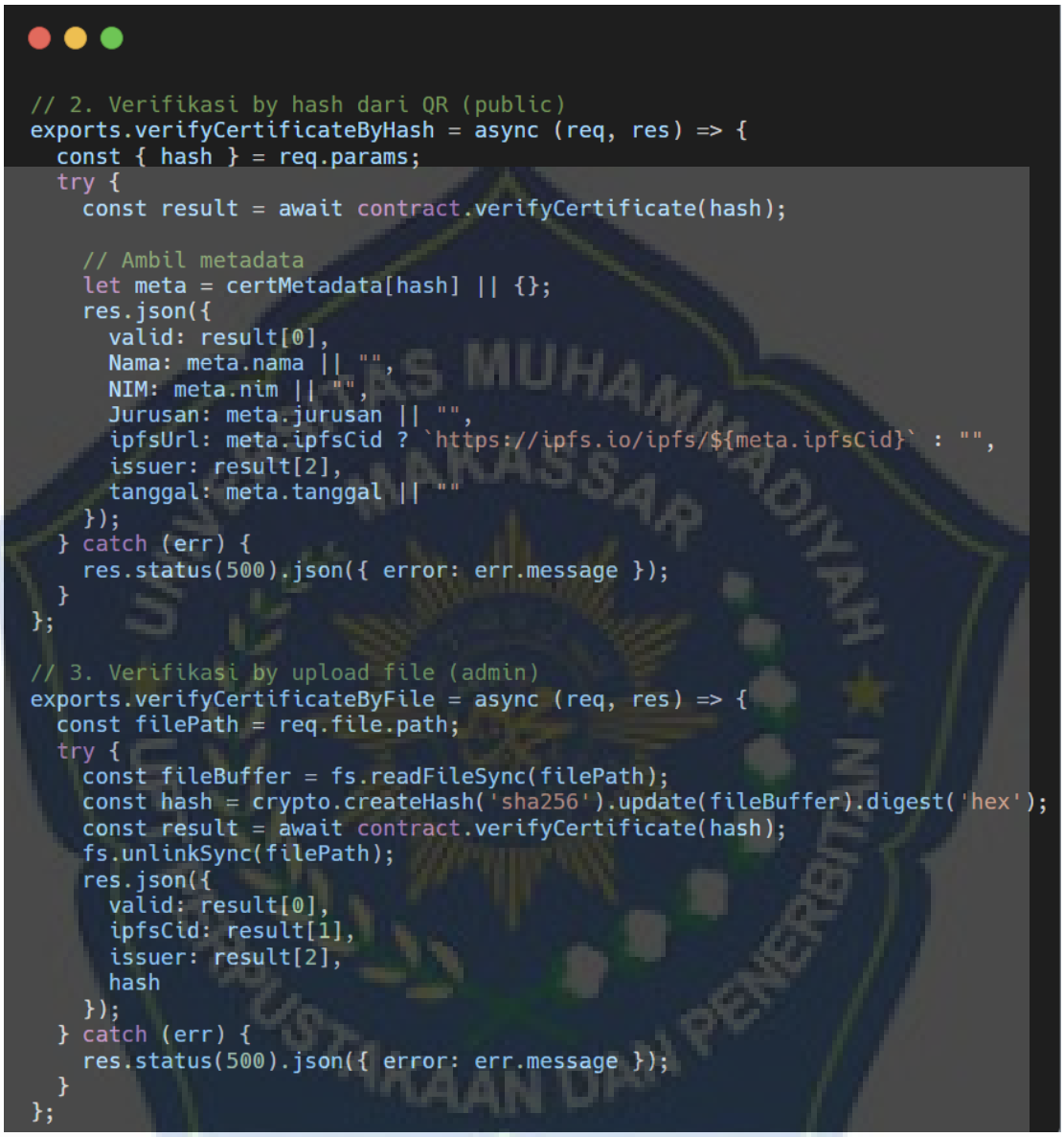
```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract CertificateRegistry {
    struct Certificate {
        string hash;          // SHA-256 hash sertifikat
        string ipfsCid;       // CID IPFS file sertifikat (hasil upload ke Pinata)
        address issuer;       // Siapa yang register (admin)
        bool valid;           // Status validitas
    }

    // Mapping dari hash ke data sertifikat
    mapping(string => Certificate) public certificates;

    // Event untuk log setiap register sertifikat baru
    event CertificateRegistered(string indexed hash, string ipfsCid, address indexed issuer);

    // Fungsi register sertifikat baru (hash unik)
    function registerCertificate(string memory _hash, string memory _ipfsCid) public {
        require(bytes(certificates[_hash].hash).length == 0, "Certificate already registered");
        certificates[_hash] = Certificate(_hash, _ipfsCid, msg.sender, true);
        emit CertificateRegistered(_hash, _ipfsCid, msg.sender);
    }

    // Fungsi verifikasi hash sertifikat
    function verifyCertificate(string memory _hash) public view returns (bool valid, string memory ipfsCid, address
issuer) {
        Certificate memory cert = certificates[_hash];
        if (cert.valid) {
            return (true, cert.ipfsCid, cert.issuer);
        }
        return (false, "", address(0));
    }

}
```

Lampiran 2. *Source Code* Fungsi Generate Sertifikat

```
// 1. Generate Sertifikat
exports.generateCertificate = async (req, res) => {
  try {
    const { nama, nim, jurusan } = req.body;
    if (!nama || !nim || !jurusan) {
      return res.status(400).json({ message: 'nama, nim, dan jurusan wajib diisi' });
    }

    // Cek duplicate
    for (let h in certMetadata) {
      if (certMetadata[h].nim.toLowerCase().trim() === nim.toLowerCase().trim()) {
        return res.status(400).json({ message: 'Sertifikat untuk NIM ini sudah pernah dibuat.' });
      }
    }

    const logoPath = path.join(__dirname, '../assets/logo-umm.png');
    const namaSlug = slugify(nama);
    const now = new Date();
    const tanggalGenerate = [
      now.getFullYear(),
      String(now.getMonth() + 1).padStart(2, '0'),
      String(now.getDate()).padStart(2, '0')
    ].join('');
    const filename = `${namaSlug}_${nim}_${tanggalGenerate}.pdf`;
    const filepath = path.join(__dirname, '../uploads', filename);

    // Generate PDF (tanpa embed QR)
    const pdfDoc = await PDFDocument.create();
    const page = pdfDoc.addPage([595, 842]);
    const colorBiru = rgb(41 / 255, 56 / 255, 145 / 255);
    const colorAbu = rgb(245 / 255, 245 / 255, 248 / 255);
    const border = 10;
    page.drawRectangle({ x: 0, y: 0, width: 595, height: 842, color: colorBiru });
    page.drawRectangle({ x: border, y: border, width: 595 - 2 * border, height: 842 - 2 * border, color: colorAbu });

    const fontBold = await pdfDoc.embedFont(StandardFonts.HelveticaBold);
    const fontReg = await pdfDoc.embedFont(StandardFonts.Helvetica);
    const logoImg = fs.readFileSync(logoPath);
    const logoEmbed = await pdfDoc.embedPng(logoImg);
    const logoWidth = 95, logoHeight = 95;
    const logoX = (595 - logoWidth) / 2;
    const logoY = 725;
    page.drawImage(logoEmbed, { x: logoX, y: logoY, width: logoWidth, height: logoHeight });

    const title = 'SERTIFIKAT MEMBACA ALQURAN';
    const titleFontSize = 18;
    const textWidth = fontBold.widthOfTextAtSize(title, titleFontSize);
    const textX = (595 - textWidth) / 2;
    const titleY = logoY - 25;
    page.drawText(title, { x: textX, y: titleY, size: titleFontSize, font: fontBold, color: rgb(0, 0, 0) });
    page.drawRectangle({ x: textX, y: titleY - 3, width: textWidth, height: 2, color: rgb(0, 0, 0) });

    let y = titleY - 34;
    page.drawText('Assalamualaikum Wr.Wb.', { x: 54, y, size: 13, font: fontBold });

    y -= 22;
    const pembuka = "Dengan Rahmat Allah, Tim Penguji Membaca Al-Qur'an Mahasiswa Fakultas Teknik Universitas Muhammadiyah Makassar, Memberikan "Sertifikat Baca Al-Quran" Kepada Mahasiswa :";
    drawMultiline(page, pembuka, 54, y, 12, fontReg, 17, 480);

    y -= 3 * 17 + 8;
    page.drawText("Nama    :", { x: 77, y, size: 12, font: fontReg });
    page.drawText(nama, { x: 160, y, size: 12, font: fontBold });

    y -= 20;
    page.drawText("Nim     :", { x: 77, y, size: 12, font: fontReg });
    page.drawText(nim, { x: 160, y, size: 12, font: fontBold });

    y -= 20;
    page.drawText("Jurusan :", { x: 77, y, size: 12, font: fontReg });
    page.drawText(jurusan, { x: 160, y, size: 12, font: fontBold });

    y -= 28;
    const penutup = "Setelah melalui Test Baca Al-Qur'an dan dinyatakan Lulus. Sertifikat ini diberikan kepada Mahasiswa yang bersangkutan untuk dipergunakan untuk persyaratan administrasi penyelesaian studi";
    drawMultiline(page, penutup, 54, y, 12, fontReg, 17, 480);

    y -= 3 * 17 + 7;
    page.drawText('Wassalamu Alaikum Wr.Wb.', { x: 54, y, size: 13, font: fontBold });

    // Tanggal kanan bawah
    const tgl = new Date().toLocaleDateString('id-ID', { day: 'numeric', month: 'long', year: 'numeric' });
    page.drawText(`Makassar, ${tgl}`, { x: 330, y: 90, size: 12, font: fontReg });
```

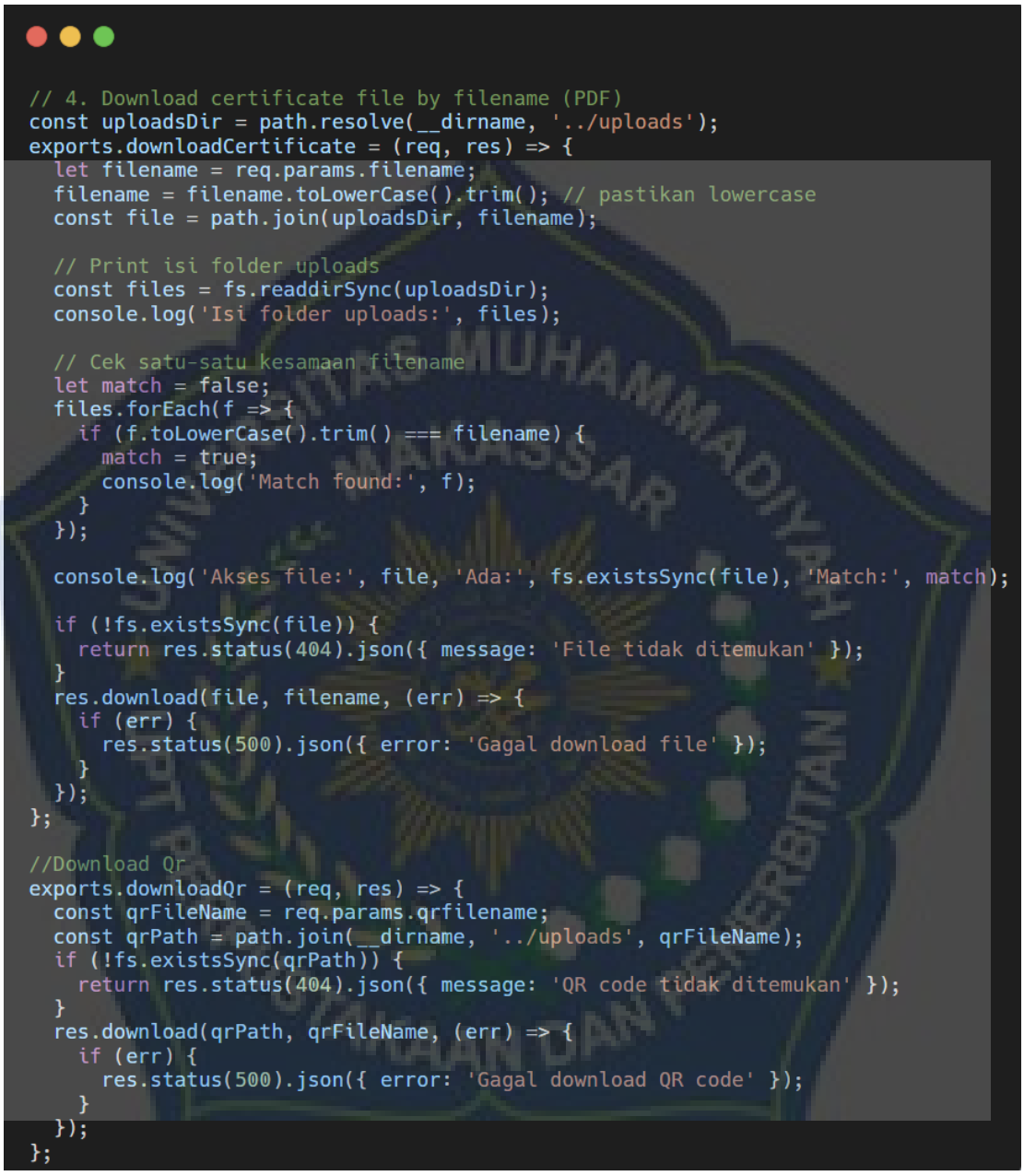
```
// Simpan PDF sementara (sebelum hash final)
    const finalPdfBytes = await pdfDoc.save();
    fs.writeFileSync(filepath, finalPdfBytes);

    // Hash file FINAL
    const fileBuffer = fs.readFileSync(filepath);
    const hash = crypto.createHash('sha256').update(fileBuffer).digest('hex');

    // --- UPDATE QR CODE LINK (ke halaman React) ---
    const hashQrData = `http://localhost:3000/verify/${hash}`;
    const qrFileName = `qr_${namaSlug}_${nim}_${tanggalGenerate}.png`;
    const qrPath = path.join(__dirname, '../uploads', qrFileName);
    await QRCode.toFile(qrPath, hashQrData, { width: 220 });

    // Hash ulang PDF FINAL (agar benar2 final)
    const finalFileBuffer = fs.readFileSync(filepath);
    const finalHash =
crypto.createHash('sha256').update(finalFileBuffer).digest('hex');
    // Upload ke Pinata
    const pinataResult = await pinata.pinFileToIPFS(fs.createReadStream(filepath));
    const ipfsCid = pinataResult.IpfsHash;

    // Simpan ke smart contract (pakai hash final)
    const adminPrivateKey = process.env.ADMIN_PRIVATE_KEY;
    const wallet = new ethers.Wallet(adminPrivateKey, provider);
    const contractWithSigner = contract.connect(wallet);
    const tx = await contractWithSigner.registerCertificate(finalHash, ipfsCid);
    await tx.wait();

    certMetadata[finalHash] = {
      nama,
      nim,
      jurusan,
      tanggal: tgl,
      hash: finalHash,
      ipfsCid,
      issuer: wallet.address,
      qrFileName
    };
    saveMetadata();

    res.json({
      message: 'Sertifikat berhasil dibuat!',
      nama, nim, jurusan, tanggal: tgl,
      hash: finalHash,
      ipfsCid,
      ipfsUrl: `https://ipfs.io/ipfs/${ipfsCid}`,
      downloadUrl: `/api/certificate/download/${filename}`,
      downloadQrUrl: `/api/certificate/download-qr/${qrFileName}` // URL download QR
    });

  } catch (err) {
    console.error(err);
    res.status(500).json({ error: err.message });
  }
};

// Helper fungsi untuk multi-line text (otomatis wrap maxWidth)
function drawMultiline(page, text, x, y, size, font, lineHeight=16, maxWidth=480) {
  const words = text.split(' ');
  let line = '';
  let lineY = y;
  for (let n = 0; n < words.length; n++) {
    const testLine = line + words[n] + ' ';
    const testWidth = font.widthOfTextAtSize(testLine, size);
    if (testWidth > maxWidth && n > 0) {
      page.drawText(line.trim(), { x, y: lineY, size, font });
      line = words[n] + ' ';
      lineY -= lineHeight;
    } else {
      line = testLine;
    }
  }
  if (line) page.drawText(line.trim(), { x, y: lineY, size, font });
}
```

Lampiran 3. *Source Code* Fungsi Verifikasi Sertifikat



```
// 2. Verifikasi by hash dari QR (public)
exports.verifyCertificateByHash = async (req, res) => {
  const { hash } = req.params;
  try {
    const result = await contract.verifyCertificate(hash);

    // Ambil metadata
    let meta = certMetadata[hash] || {};
    res.json({
      valid: result[0],
      Nama: meta.nama || "",
      NIM: meta.nim || "",
      Jurusan: meta.jurusan || "",
      ipfsUrl: meta.ipfsCid ? `https://ipfs.io/ipfs/${meta.ipfsCid}` : "",
      issuer: result[2],
      tanggal: meta.tanggal || ""
    });
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
};

// 3. Verifikasi by upload file (admin)
exports.verifyCertificateByFile = async (req, res) => {
  const filePath = req.file.path;
  try {
    const fileBuffer = fs.readFileSync(filePath);
    const hash = crypto.createHash('sha256').update(fileBuffer).digest('hex');
    const result = await contract.verifyCertificate(hash);
    fs.unlinkSync(filePath);
    res.json({
      valid: result[0],
      ipfsCid: result[1],
      issuer: result[2],
      hash
    });
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
};
```

Lampiran 4. *Source Code* Fungsi download Sertifikat dan QR Code



```
// 4. Download certificate file by filename (PDF)
const uploadsDir = path.resolve(__dirname, '../uploads');
exports.downloadCertificate = (req, res) => {
  let filename = req.params.filename;
  filename = filename.toLowerCase().trim(); // pastikan lowercase
  const file = path.join(uploadsDir, filename);

  // Print isi folder uploads
  const files = fs.readdirSync(uploadsDir);
  console.log('Isi folder uploads:', files);

  // Cek satu-satu kesamaan filename
  let match = false;
  files.forEach(f => {
    if (f.toLowerCase().trim() === filename) {
      match = true;
      console.log('Match found:', f);
    }
  });

  console.log('Akses file:', file, 'Ada:', fs.existsSync(file), 'Match:', match);

  if (!fs.existsSync(file)) {
    return res.status(404).json({ message: 'File tidak ditemukan' });
  }
  res.download(file, filename, (err) => {
    if (err) {
      res.status(500).json({ error: 'Gagal download file' });
    }
  });
};

//Download Qr
exports.downloadQr = (req, res) => {
  const qrFileName = req.params.qrfilename;
  const qrPath = path.join(__dirname, '../uploads', qrFileName);
  if (!fs.existsSync(qrPath)) {
    return res.status(404).json({ message: 'QR code tidak ditemukan' });
  }
  res.download(qrPath, qrFileName, (err) => {
    if (err) {
      res.status(500).json({ error: 'Gagal download QR code' });
    }
  });
};
```

Lampiran 5. Surat Permohonan Penelitian Ke Program Studi Informatika

**SURAT PERMOHONAN PENELITIAN**

Hal : Permohonan Surat Penelitian

Kepada Yth,

Ketua Program Studi Informatika

Di

Tempat

***Assalamu Alaikum Warahmatullahi Wabarakatuh***

Sehubungan dengan akan dilaksanakannya Penelitian yang akan dilaksanakan di **Fakultas Teknik Universitas Muhammadiyah Makassar** oleh mahasiswa Fakultas Teknik Program Studi Informatika. Adapun Mahasiswa yang bersangkutan adalah sebagai berikut :

| No | Nama | Nim |
|---|---|---|
| 1 | Syariful Mujaddiq | 105841103219 |

Maka dengan ini kami memohon dibuatkan surat pengantar atau pengajuan Penelitian pada Instansi dibawah ini.

Judul Skripsi : **Sistem Verifikasi Sertifikat Berbasis Blockchain dengan Implementasi Algoritma SHA-256 dan Smart Contract.**

Nama Instansi : **Fakultas Teknik Universitas Muhammadiyah Makassar**

Alamat : **Gedung Iqra. Lt.3 Unismuh, Jln.Sultan Alauddin No. 259, Kec. Rappocini, Gunung Sari, Kota Makassar**

Demikian surat permohonan kami ajukan, atas dukungan dan kerjasamanya kami haturkan terima kasih.

***Billahi Fii Sabiilihaq, Fastabiqul Khairat***

***Waalaikumsalam Warahamatullahi Wabarakatuh***

Makassar, 22 Muharram 1447 H
18 Juli 2025 M
Pemohon

Syariful Mujaddiq
105841103219

Lampiran 6. Permohonan Surat Pengantar Penelitian

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

بسم الله الرحمن الرحيم

Nomor : 067/05/IF/C.4-VI/VII/47/2025 Makassar, 4 Safar 1447 H
Lamp. : - 29 Juli 2025 M
Hal : **Permohonan Data Penelitian**

Kepada yang Terhormat,
**Ketua LP3M Unismuh Makassar**
Di –
Tempat

***Assalamu 'Alaikum Warahmatullahi Wabarakatuh***

Dengan Rahmat Allah SWT, Semoga aktivitas kita bernilai ibadah di Sisi–Nya. Dalam rangka penyelesaian Tugas Akhir pada Program Studi Informatika dengan judul "Sistem Verifikasi Sertifikat Berbasis Blockchain dengan Implementasi Algoritma SHA-256 dan Smart Contract ". Bersama ini kami sampaikan mahasiswa:

| Stambuk | Nama |
|---|---|
| 105 84 11032 19 | Syariful Mujaddiq |

Sehubungan dengan hal tersebut, maka kami memohon dibuatkan surat pengantar pada instansi di bawah ini:

Nama Instansi : Fakultas Teknik Unismuh Makassar
Alamat : Gedung Iqra. Lt.3 Unismuh, Jln. Sultan Alauddin No. 259, kec. Rappocini, Gunung Sari, Kota Makassar

Demikian surat kami atas perhatian dan kerja samanya kami haturkan banyak terima kasih.
***Jazakumullah Khaeran Katsiran***
***Wassalamu 'Alaikum warahmatullah Wabarakatuh***

**Ketua Program Studi Informatika**

**Muhyiddin A.M Hayat, S.Kom., M.T.**
NBM. 1304 577

Tembusan:
1. Dekan Fakultas Teknik
2. Arsip

Gedung Menara Iqra Lantai 3
Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221
Web: https://teknik.unismuh.ac.id/, e-mail: teknik@unismuh.ac.id

Lampiran 7. Surat Penelitian

**UNIVERSITAS MUHAMMADIYAH MAKASSAR**
**LEMBAGA PENELITIAN PENGEMBANGAN DAN PENGABDIAN KEPADA MASYARAKAT**
Jl. Sultan Alauddin No. 259 Telp. 866972 Fax. (0411) 865588 Makassar 90221 *e-mail:* lp3m@unismuh.ac.id

بسم الله الرحمن الرحيم

Nomor : 213/LP3M/05/C.4-VIII/VII/1447/2025
Lampiran : 1 (satu) rangkap proposal
Hal : Permohonan Izin Pelaksanaan Penelitian

*Kepada Yth:*
Dekan Fakultas Teknik
Universitas Muhammadiyah Makassar
di-
Makassar

***Assalamu Alaikum Wr. Wb***

Berdasarkan surat:Dekan Fakultas Program Pascasarjana, nomor: 067 tanggal: 30 Juli 2025, menerangkan bahwa mahasiswa dengan data sebagai berikut.

| | |
|---|---|
| Nama | : SYARIFUL MUJADDIQ |
| Nim | : 105841103219 |
| Fakultas | : Teknik |
| Prodi | : Informatika |

Bermaksud melaksanakan penelitian/pengumpulan data dalam rangka penulisan laporan tugas akhir Skripsi dengan judul; :
**"Sistem Verifikasi Sertifikat Berbasis Blockchain Dengan Implementasi Algoritma SHA-256 Dan Smart Contract"**
Yang akan dilaksanakan dari tanggal 07 Agustus 2025 s/d 07 Oktober 2025.
Sehubungan dengan maksud di atas, kiranya Mahasiswa tersebut diberikan izin untuk melakukan penelitian sesuai ketentuan yang berlaku.

Demikian, atas perhatian dan kerjasamanya diucapkan jazakumullahu khaeran katziraa.
*Billahi Fii Sabilil Haq, Fastabiqul Khaerat.*
*Wassalamu Alaikum Wr. Wb.*

Makassar 5 Safar 1447 / 31 Juli 2025

Ketua LP3M Unismuh Makassar,

Dr. Muh. Arief Muhsin, M.Pd.
NBM. 112 7761

Lampiran 8. Surat Keterangan Bebas Plagiat

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat kantor: Jl.Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بسم الله الرحمن الرحيم

## SURAT KETERANGAN BEBAS PLAGIAT

**UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,**
**Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:**

Nama : Syariful Mujaddiq
Nim : 105841103219
Program Studi : Teknik Informatika

Dengan nilai:

| No | Bab | Nilai | Ambang Batas |
|---|---|---|---|
| 1 | Bab 1 | 10% | 10 % |
| 2 | Bab 2 | 10% | 25 % |
| 3 | Bab 3 | 9% | 15 % |
| 4 | Bab 4 | 1% | 10 % |
| 5 | Bab 5 | 0% | 5 % |

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 28 Agustus 2025
Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,

Nursinah, S.Hum., M.I.P
NBM. 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Lampiran 9. Hasil Turnitin BAB I

Bab I Syariful Mujaddiq
105841103219

*by* Tahap Tutup

**Submission date:** 27-Aug-2025 12:21PM (UTC+0700)
**Submission ID:** 2736010330
**File name:** BAB_I_113.docx (19.32K)
**Word count:** 909
**Character count:** 6506

Bab I Syariful Mujaddiq 105841103219

ORIGINALITY REPORT

| 10% | 9% | 2% | 8% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

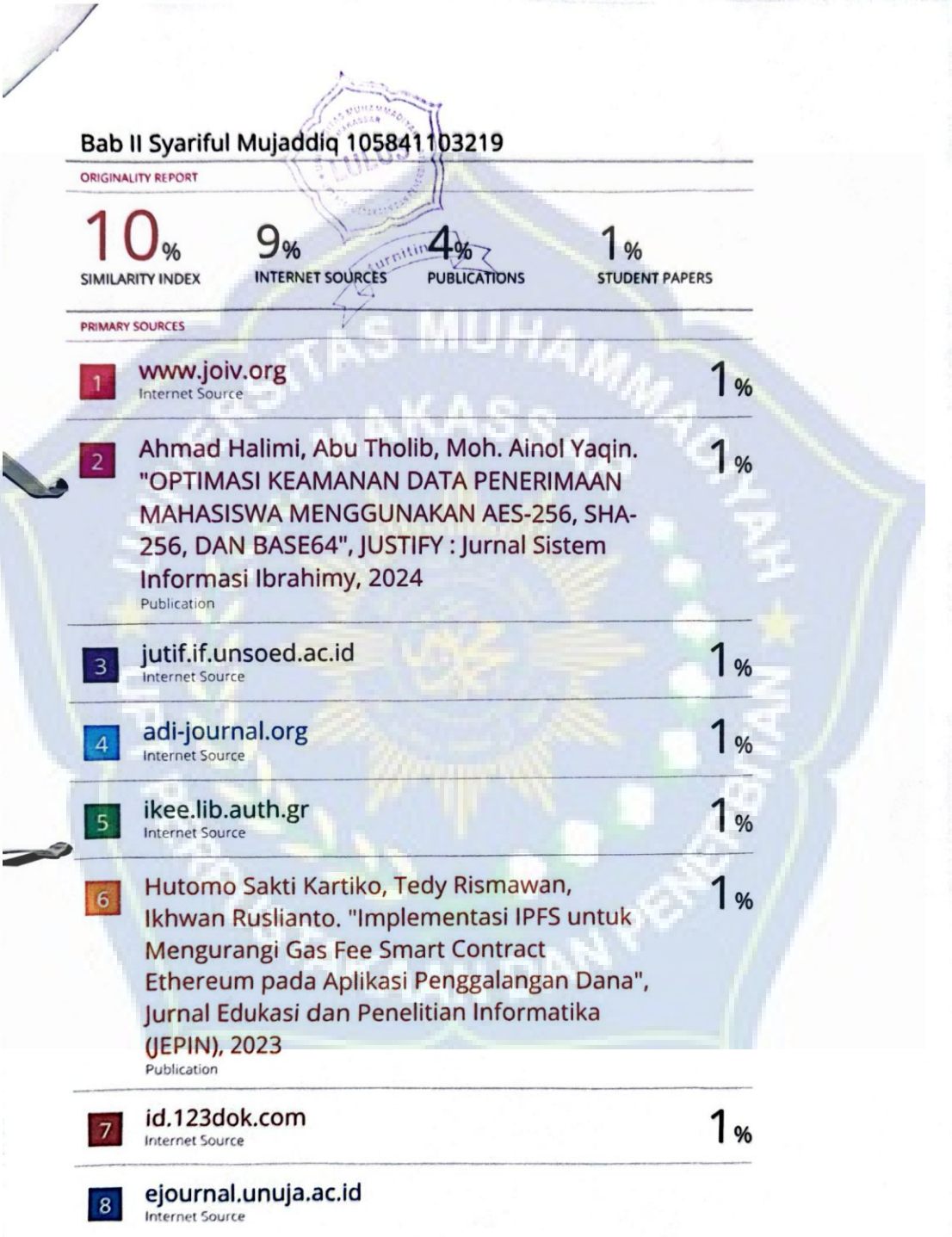
| | | |
|---|---|---|
| 1 | Submitted to Fakultas Teknik<br>Student Paper | 3% |
| 2 | digilibadmin.unismuh.ac.id<br>Internet Source | 2% |
| 3 | etheses.iainkediri.ac.id<br>Internet Source | 2% |
| 4 | www.coursehero.com<br>Internet Source | 2% |
| 5 | jurnal.pancabudi.ac.id<br>Internet Source | 2% |

Exclude quotes Off

Exclude bibliography Off

Exclude matches < 2%

Lampiran 10. Hasil Turnitin BAB II



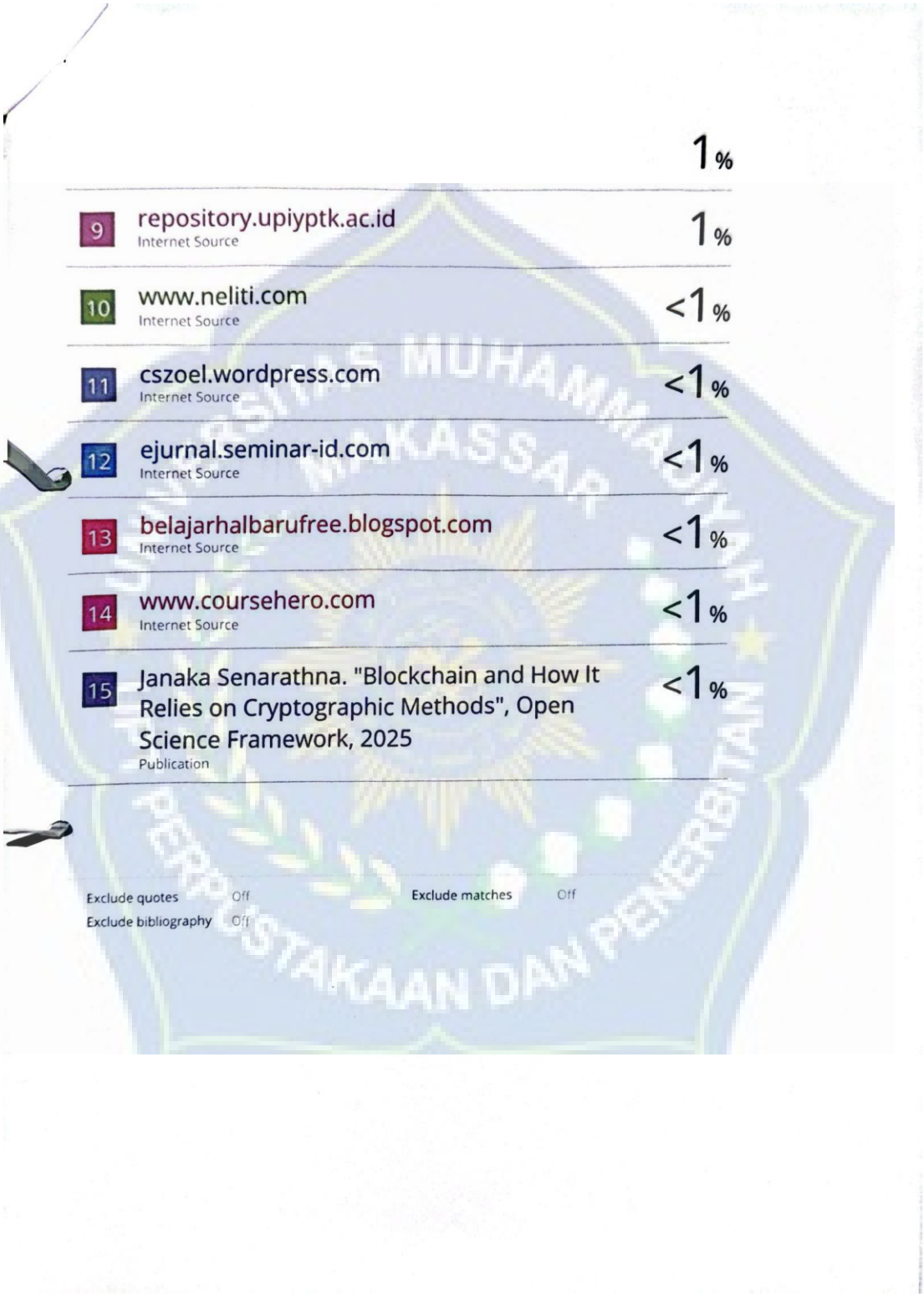
Bab II Syariful Mujaddiq 105841103219

ORIGINALITY REPORT

| 10% | 9% | 4% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

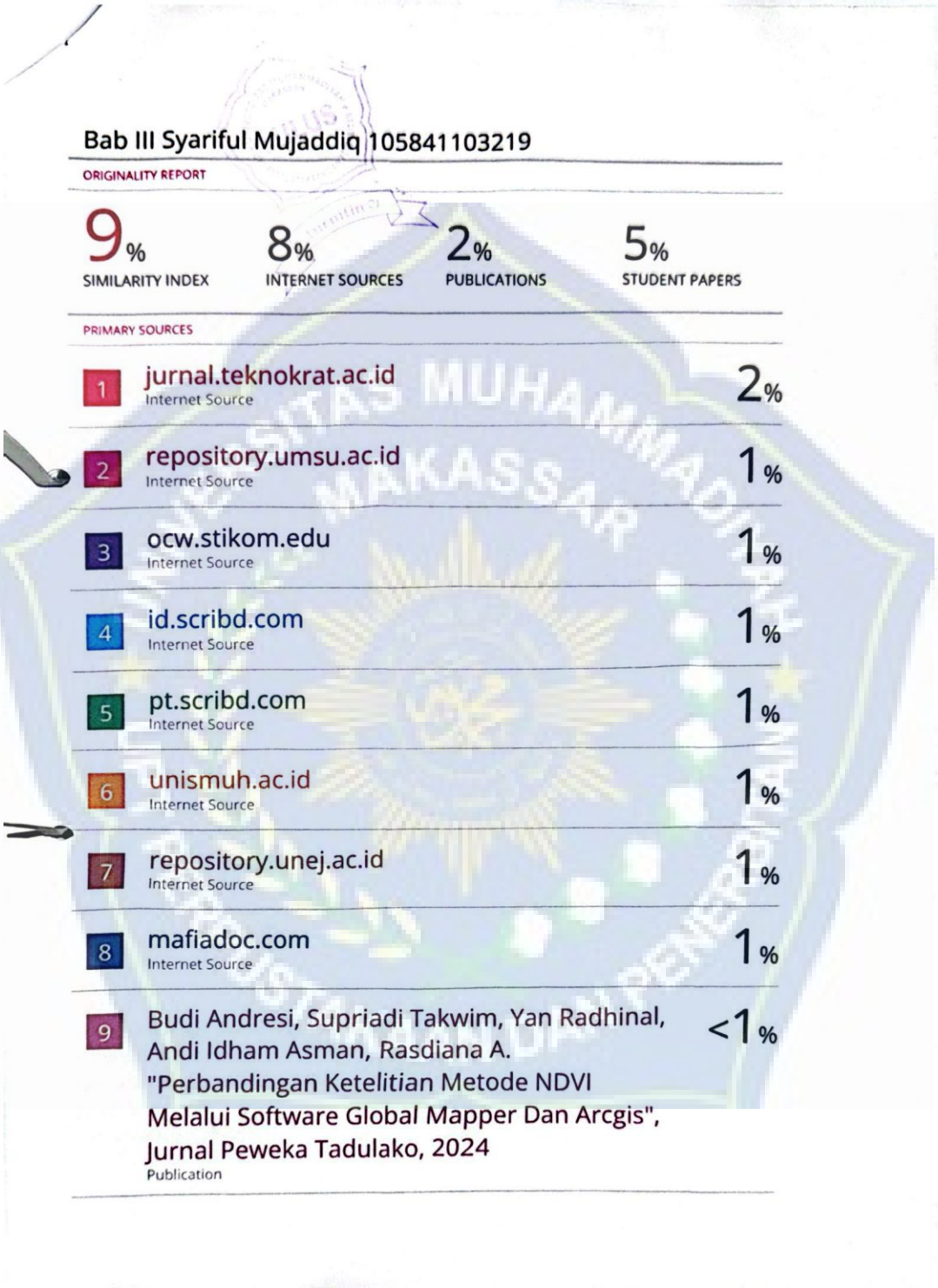
| | | |
|---|---|---|
| 1 | www.joiv.org<br>Internet Source | 1% |
| 2 | Ahmad Halimi, Abu Tholib, Moh. Ainol Yaqin. "OPTIMASI KEAMANAN DATA PENERIMAAN MAHASISWA MENGGUNAKAN AES-256, SHA-256, DAN BASE64", JUSTIFY : Jurnal Sistem Informasi Ibrahimy, 2024<br>Publication | 1% |
| 3 | jutif.if.unsoed.ac.id<br>Internet Source | 1% |
| 4 | adi-journal.org<br>Internet Source | 1% |
| 5 | ikee.lib.auth.gr<br>Internet Source | 1% |
| 6 | Hutomo Sakti Kartiko, Tedy Rismawan, Ikhwan Ruslianto. "Implementasi IPFS untuk Mengurangi Gas Fee Smart Contract Ethereum pada Aplikasi Penggalangan Dana", Jurnal Edukasi dan Penelitian Informatika (JEPIN), 2023<br>Publication | 1% |
| 7 | id.123dok.com<br>Internet Source | 1% |
| 8 | ejournal.unuja.ac.id<br>Internet Source | |



| | | |
|---|---|---|
| | | 1% |
| 9 | repository.upiyptk.ac.id<br>Internet Source | 1% |
| 10 | www.neliti.com<br>Internet Source | <1% |
| 11 | cszoel.wordpress.com<br>Internet Source | <1% |
| 12 | ejurnal.seminar-id.com<br>Internet Source | <1% |
| 13 | belajarhalbarufree.blogspot.com<br>Internet Source | <1% |
| 14 | www.coursehero.com<br>Internet Source | <1% |
| 15 | Janaka Senarathna. "Blockchain and How It Relies on Cryptographic Methods", Open Science Framework, 2025<br>Publication | <1% |

Exclude quotes Off
Exclude bibliography Off
Exclude matches Off

Lampiran 11. Hasil Turnitin BAB III



Bab III Syariful Mujaddiq 105841103219

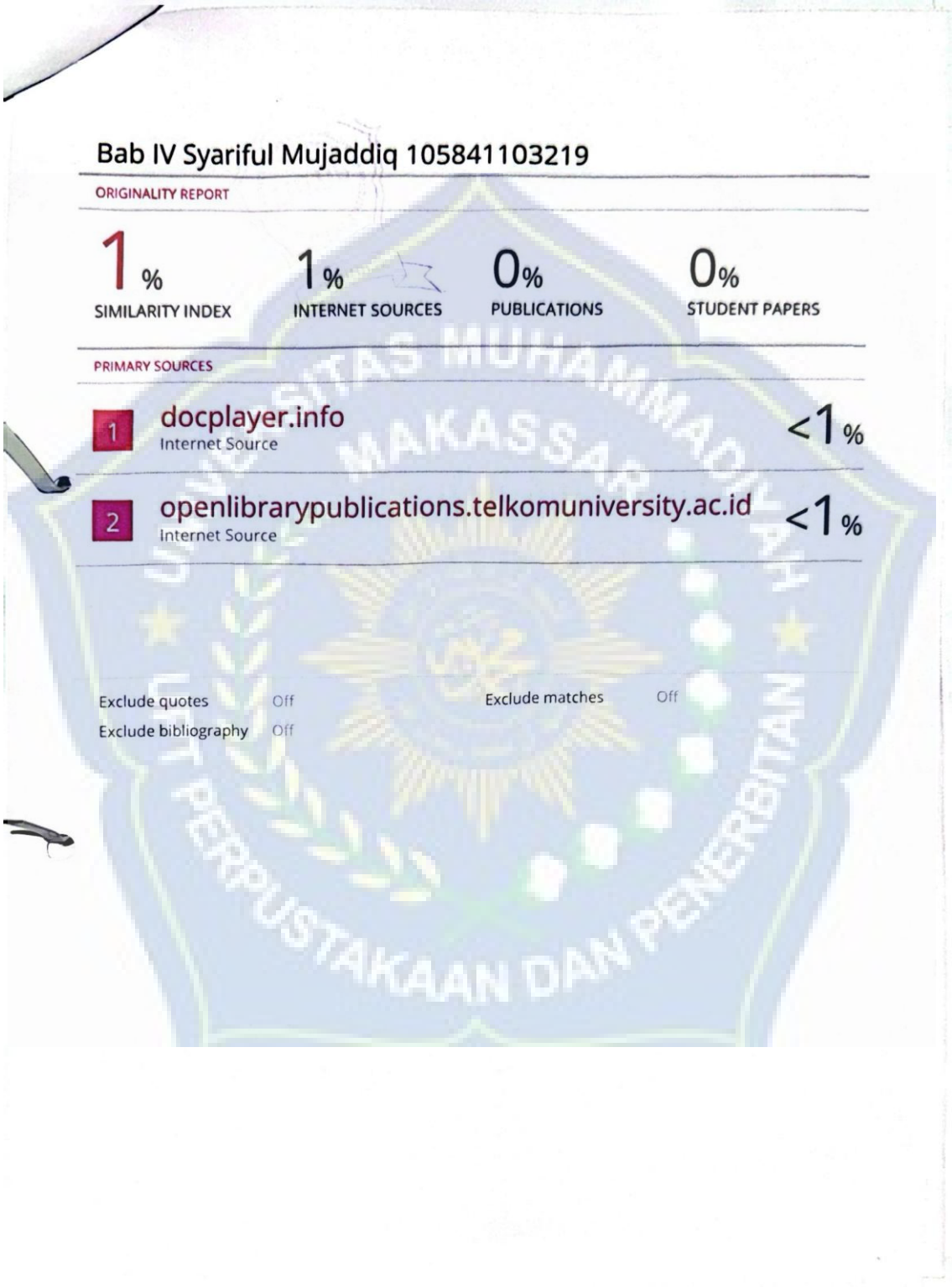
ORIGINALITY REPORT

| 9% | 8% | 2% | 5% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | jurnal.teknokrat.ac.id<br>Internet Source | 2% |
| 2 | repository.umsu.ac.id<br>Internet Source | 1% |
| 3 | ocw.stikom.edu<br>Internet Source | 1% |
| 4 | id.scribd.com<br>Internet Source | 1% |
| 5 | pt.scribd.com<br>Internet Source | 1% |
| 6 | unismuh.ac.id<br>Internet Source | 1% |
| 7 | repository.unej.ac.id<br>Internet Source | 1% |
| 8 | mafiadoc.com<br>Internet Source | 1% |
| 9 | Budi Andresi, Supriadi Takwim, Yan Radhinal, Andi Idham Asman, Rasdiana A. "Perbandingan Ketelitian Metode NDVI Melalui Software Global Mapper Dan Arcgis", Jurnal Peweka Tadulako, 2024<br>Publication | <1% |

Lampiran 12. Hasil Turnitin BAB IV



Bab IV Syariful Mujaddiq 105841103219

ORIGINALITY REPORT

| 1% | 1% | 0% | 0% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | docplayer.info<br>Internet Source | <1% |
| 2 | openlibrarypublications.telkomuniversity.ac.id<br>Internet Source | <1% |

Exclude quotes Off
Exclude bibliography Off
Exclude matches Off

Lampiran 13. Hasil Turnitin BAB V

Bab V Syariful Mujaddiq 105841103219

ORIGINALITY REPORT

| 0% | 0% | 0% | 0% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | Off | | |