PENERAPAN TEKNOLOGI BLOCKCHAIN UNTUK VERIFIKASI IDENTITAS PENGGUNA PADA SISTEM AUTENTIKASI WEB BERBASIS BACKEND.

Siprianus Tabuni[1];
[1]Program Studi Teknik Informatika, Fakultas Telematika Energi, Institut Teknologi PLN
siprianus1931253@itpln.ac.id

ABSTRACT

Conventional authentication systems based on username and password still dominate, despite being vulnerable to various cyber attacks. This study aims to implement and evaluate blockchain technology as an identity verification mechanism and comprehensively compare it with a traditional authentication system strengthened with modern cryptography. The research uses Research and Development (R&D) approach with prototyping method. Two systems were developed: (1) traditional login system with MySQL database integrating ECDSA algorithm for key generation, AES-256-GCM for private key encryption, and bcrypt for password hashing; (2) passwordless blockchain login system utilizing Solidity smart contracts, Ganache local network, and MetaMask wallet. Security testing was conducted through penetration testing scenarios including SQL Injection, Replay Attack, Signature Forgery, and JWT Tampering. The results show that the traditional login system successfully blocked 100% of SQL Injection attacks (45 out of 45 attempts failed) with response time under 50 ms. The blockchain login system demonstrated fundamental security resilience with a single-use nonce mechanism successfully preventing replay attacks and ECDSA digital signature verification mathematically ensuring user identity authenticity. This research concludes that both systems have excellent security resistance according to their respective defense mechanisms.

Keywords: *Blockchain, ECDSA, Authentication Security, Replay Attack, SQL Injection, JWT.*

ABSTRAK

Sistem autentikasi konvensional berbasis username dan password masih mendominasi, meskipun rentan terhadap berbagai serangan siber. Penelitian ini bertujuan untuk menerapkan dan mengevaluasi teknologi blockchain sebagai mekanisme verifikasi identitas, serta membandingkannya secara komprehensif dengan sistem autentikasi tradisional yang diperkuat dengan kriptografi modern. Penelitian menggunakan pendekatan Research and Development (R&D) dengan metode prototyping. Dua sistem dikembangkan: (1) sistem login tradisional dengan database MySQL yang mengintegrasikan algoritma ECDSA, AES-256-GCM, dan bcrypt; (2) sistem login blockchain tanpa password memanfaatkan smart contract Solidity, jaringan lokal Ganache, dan wallet MetaMask. Pengujian keamanan dilakukan melalui skenario penetration testing meliputi SQL Injection, Replay Attack, Signature Forgery, dan JWT Tampering. Hasil penelitian menunjukkan sistem login tradisional berhasil menangkal 100% serangan SQL Injection (45 percobaan gagal) dengan waktu respons di bawah 50 ms. Sistem login blockchain menunjukkan ketahanan keamanan fundamental dengan mekanisme nonce sekali pakai berhasil mencegah serangan replay dan verifikasi ECDSA memastikan keaslian identitas pengguna secara matematis. Penelitian ini menyimpulkan bahwa kedua sistem memiliki ketahanan keamanan yang sangat baik sesuai mekanisme pertahanan masing-masing.

Kata kunci: *Blockchain, ECDSA, Authentication Security, Replay Attack, SQL Injection, JWT.*

1. PENDAHULUAN.

Perkembangan teknologi informasi telah membawa perubahan signifikan terhadap cara manusia berinteraksi dengan sistem digital, khususnya dalam proses autentikasi dan manajemen identitas. Sistem autentikasi konvensional berbasis username dan password masih menjadi metode dominan dalam aplikasi web maupun sistem backend. Namun, metode ini memiliki sejumlah kelemahan, seperti kerentanan terhadap serangan phishing, credential stuffing, brute force attack, serta penyalahgunaan data akibat sentralisasi penyimpanan identitas pada server pusat [1].

Menurut survei Verizon Data Breach Investigations Report (DBIR, 2022), lebih dari 80% insiden pelanggaran data melibatkan pencurian atau kebocoran kredensial [2]. Hal ini membuktikan bahwa mekanisme autentikasi tradisional tidak lagi memadai untuk menjawab tantangan keamanan digital yang semakin kompleks. Di Indonesia, laporan Badan Siber dan Sandi Negara (BSSN, 2023) mencatat peningkatan 47% serangan siber dibandingkan tahun sebelumnya, dengan 65% di antaranya menargetkan kredensial pengguna [3].

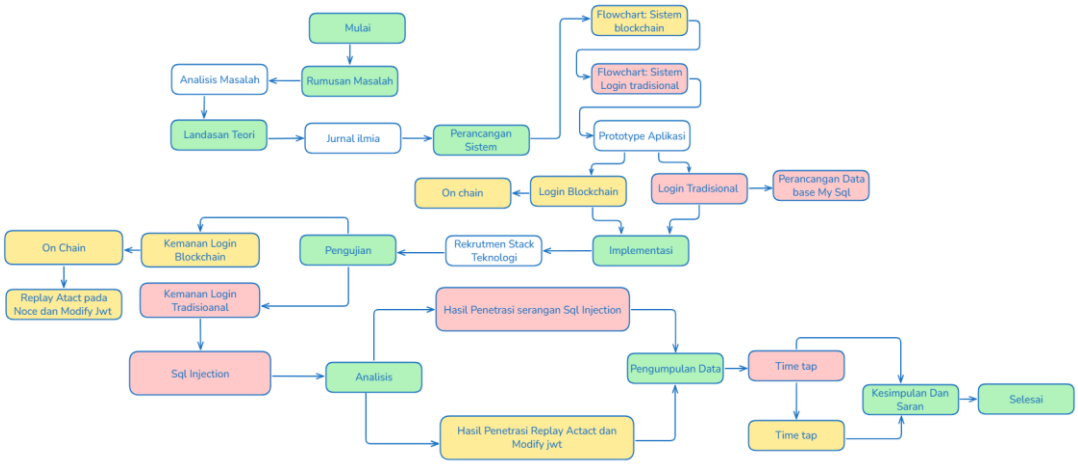
Di sisi lain, teknologi blockchain muncul sebagai solusi inovatif dalam mengatasi kelemahan sistem autentikasi tradisional. Blockchain memiliki karakteristik immutability, desentralisasi, transparansi, dan keamanan berbasis kriptografi, yang menjadikannya cocok digunakan untuk manajemen identitas digital [4]. Implementasi blockchain untuk identitas digital telah terbukti efektif dalam beberapa studi, seperti penerapan sistem identitas terdesentralisasi di Estonia (e-Residency) yang berhasil mengurangi insiden pencurian identitas hingga 99% [5].

Salah satu konsep yang berkembang dari blockchain adalah Self-Sovereign Identity (SSI), yaitu identitas digital yang sepenuhnya dikendalikan oleh individu tanpa ketergantungan pada pihak ketiga [6]. Dengan mengadopsi mekanisme Decentralized Identifier (DID) dan Verifiable Credentials (VC), sistem autentikasi berbasis blockchain memungkinkan pengguna membuktikan identitasnya tanpa harus membagikan data sensitif secara langsung [7].

Penelitian ini bertujuan untuk menerapkan dan mengevaluasi teknologi blockchain sebagai mekanisme verifikasi identitas, serta membandingkannya secara komprehensif dengan sistem autentikasi tradisional yang telah diperkuat dengan kriptografi modern. Fokus utama penelitian adalah pada analisis keamanan kedua sistem terhadap berbagai vektor serangan siber.

2. METODE PENELITIAN.

2.1 Desain Penelitian.

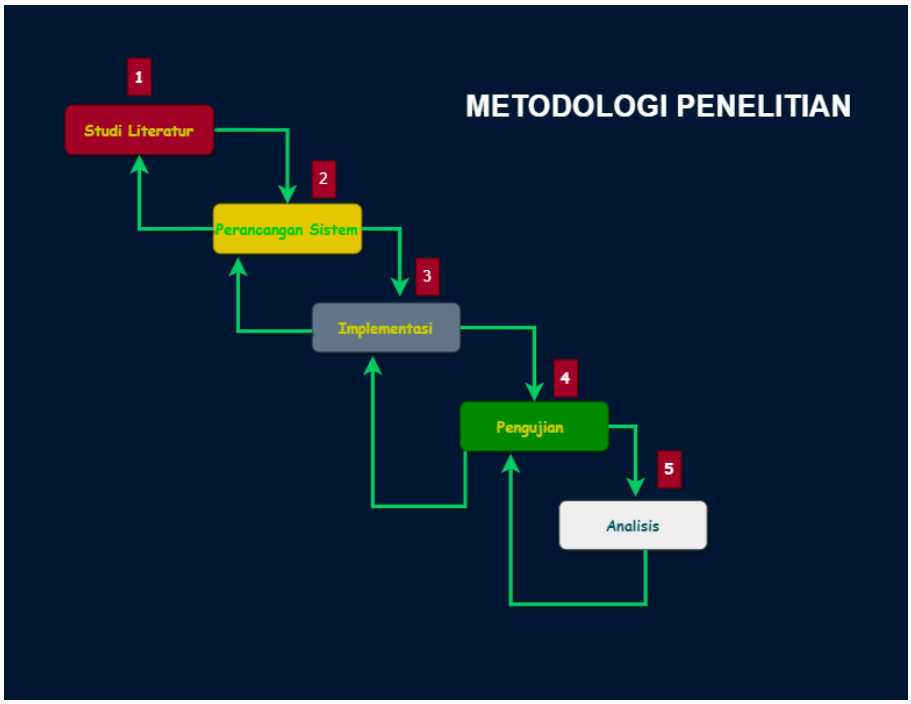


Gambar 1. Desain Penelitian.

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan metode prototyping. Paradigma penelitian yang diterapkan adalah deskriptif-eksperimental, di mana penelitian tidak hanya mendeskripsikan fenomena kelemahan sistem autentikasi tradisional, tetapi juga membangun prototipe sistem baru serta menguji secara eksperimental efektivitas dan keamanannya. Model pengembangan yang digunakan adalah model Waterfall yang terdiri dari lima tahapan: studi literatur, perancangan sistem, implementasi, pengujian, dan analisis hasil.

2.2 Metodologi Penelitian.

Metode penelitian yang digunakan adalah metode waterfal sebagai berikut.



Gambar.2 Metodologi Penelitian.

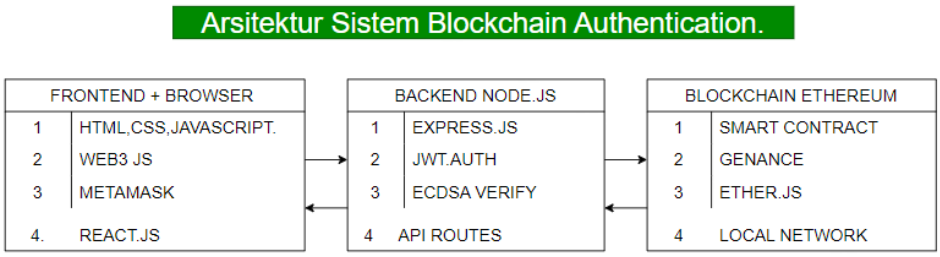
1) **Studi Literatur.**
   Mengumpulkan referensi dari jurnal, artikel ilmiah, dan laporan penelitian terkait blockchain, identitas digital, selft sovereign indentity (SSI), dan autentikasi web.
2) **Perancangan Sistem.**
   Membuat model arsitektur, diagram alur, flowchart serta merancangan backend dan smart contract serta membuat dan merancang database untuk sistem login tradisional.
3) **Implementasi.**
   Membangun sistem login autentikasi berbasis blockchain dengan menggunakan Node.js, Express.js, dan Solidity dengan tools pengujian Postman, MetaMask, dan Ganache.
   Membangun sistem login tradisional dengan mengintergasikan algoritma algoritma asimetris Ecdsa secp256k1 digunakan untuk login blockchain. sedangakan untuk login tradisonal menggunakan algortima. ECDSA (Elliptic Curve Digital Signature Algorithm) digunakan untuk membuat pasangan kunci publik dan privat, serta untuk membuat dan memverifikasi tanda tangan digital. secp256k1 kurva eliptik yang digunakan dalam algoritma ECDSA. SHA-256 (Secure Hash Algorithm 256) digunakan untuk menghasilkan hash dari password yang digunakan sebagai kunci enkripsi. AES-256-GCM (Advanced Encryption Standard 256-bit

Galois/Counter Mode) digunakan untuk mengenkripsi dan mendekripsi kunci privat ECDSA dan Keccak-256 digunakan untuk menghasilkan hash dari pesan yang akan ditandatangani pada login tradisional berbasis database.

4) **Pengujian.**
Melakukan serangan sql pada keamanan login tradisional dan melakukan penetrasi serangan replay atact nonce dan jeson wen token Jwt pada keamanan sistem login blockchain. pengujian dilakukan dengan menggunakan burpt suite dan terminal git bush untuk Application Programming Interface (API). menguji ketahanan algortima yang suda di terapkan pada kedua sistem login tersebut web yang berbasis backend.

5) **Analisis Hasil.**
Membandingkan hasil autentikasi blockchain dengan autentikasi tradisional dari sisi keamanan dan transparansi.

2.2 Arsitektur Sistem.



Gambar 3. Arsitektur Layer Sistem Blockchain Authentication.

Dua sistem dikembangkan dalam penelitian ini.

a. Sistem Login Blockchain.
Sistem login blockchain dikembangkan dengan arsitektur three-tier yang terdiri dari Frontend Layer (React.js, MetaMask), Backend Layer (Node.js, Express.js, ethers.js), dan Blockchain Layer (Solidity smart contract, Ganache local network). Sistem ini mengimplementasikan autentikasi tanpa password (passwordless) menggunakan tanda tangan digital ECDSA.

b. Sistem Login Tradisional.
Sistem login tradisional mengintegrasikan algoritma ECDSA untuk pembangkitan kunci, AES-256-GCM untuk enkripsi private key, dan bcrypt untuk hashing password. Sistem menggunakan database MySQL untuk penyimpanan data pengguna.

2.3 Spesifikasi Perangkat Lunak.

Tabel 1 menyajikan spesifikasi perangkat lunak yang digunakan dalam pengembangan kedua sistem.

Tabel 1. *Spesifikasi Perangkat Lunak.*

Kategori	Teknologi	Versi	Fungsi
Backend	Node.js/Express.js	18.17.0+ / 4.x+	Runtime server & framework API
Blockchain	Solidity/Hardhat/Ganache	0.8.x / 2.19.0 / 7.0+	Smart contract & jaringan lokal
Database	MySQL	8.0+	Penyimpanan data tradisional
Frontend	React.js/MetaMask	18.x+ / 10.28+	Antarmuka pengguna & wallet
Kriptografi	ethers.js / jsonwebtoken	6.8.0 / 9.0.2	Verifikasi signature & JWT

2.4 Skenario Pengujian.

Pengujian keamanan dilakukan melalui pendekatan penetration testing dengan lima skenario utama:

1. **Sql injection testing.**
   45 percobaan serangan dengan teknik classic SQLi, time-based, boolean-based blind, dan union-based injection pada sistem login tradisional.
2. **Replay attack testing.**
   Menangkap request login valid menggunakan Burp Suite dan mengirimkan ulang request yang sama sebanyak 10 kali pada sistem blockchain.
3. Signature Forgery Testing.
   Menguji fungsi ethers.verifyMessage() dengan manipulasi message, signature malleability, dan hash collision attack.
4. **Jwt tampering testing.**
   Modifikasi token JWT pada header Authorization termasuk perubahan algoritma dan manipulasi klaim.
5. **Cors & csrf testing.**
   Manipulasi header Origin ke domain tidak sah untuk menguji kebijakan cross-origin.

3. HASIL DAN PEMBAHASAN.

3.1 Implementasi Sistem Login Blockchain.

Sistem login blockchain diimplementasikan menggunakan arsitektur RESTful API dengan endpoint utama untuk registrasi, login, dan manajemen nonce. Fungsi inti verifikasi signature diimplementasikan sebagai berikut.

```
const validateSignature = (message, signature, expectedAddress) => {
    const recoveredAddress = ethers.verifyMessage(message, signature);
    return recoveredAddress.toLowerCase() === expectedAddress.toLowerCase();
};
```

Gambar 4. Signature verification.

Fungsi ini memanfaatkan algoritma ECDSA untuk memulihkan alamat pengguna dari signature, kemudian mencocokkannya dengan alamat yang diklaim. Implementasi ini memastikan bahwa hanya pemilik private key yang sah dapat menghasilkan signature valid.

```
class CryptoUtils {
    // 1. Signature validation dengan ethers.js
    static validateSignature(message, signature, expectedAddress) {
        return ethers.verifyMessage(message, signature) === expectedAddress;
    }

    // 2. Nonce generation secara cryptographically secure
    static generateNonce() {
        return crypto.randomInt(0, 1000000);
    }

    // 3. Message creation untuk signing
    static createLoginMessage(address, username, nonce) {
        return `Login to Blockchain System\nAddress: ${address}\nUsername: ${username}\nNonc
e: ${nonce}`;
    }
}
```

Gambar 5. Mencegah Serangan Replay.

Mekanisme nonce diterapkan untuk mencegah serangan replay. Setiap permintaan login menggunakan nilai nonce unik yang hanya dapat digunakan satu kali, dengan implementasi sebagai berikut.

```
// Nonce management untuk replay protection
router.get('/nonce/:address', (req, res) => {
    const { address } = req.params;
    const user = users.get(address.toLowerCase());
    const nonce = user ? user.nonce : CryptoUtils.generateNonce();

    res.json({ success: true, nonce });
});
```

Gambar 6. Signature Verification.

Fungsi GET /nonce/:address pada gambar tersebut merupakan implementasi mekanisme nonce untuk mencegah serangan replay. Ketika pengguna akan melakukan login, sistem akan mengambil nilai nonce unik dari parameter alamat wallet yang dikirimkan. Nilai nonce ini nantinya akan disertakan dalam pesan yang harus ditandatangani oleh pengguna menggunakan MetaMask. Mekanisme ini berhasil mencegah serangan replay karena setiap kali pengguna berhasil login, sistem akan menghasilkan nonce baru, sehingga nonce lama menjadi tidak valid. Akibatnya, jika penyerang berhasil menangkap dan mencoba menggunakan ulang signature yang sama dengan nonce lama, sistem akan menolak permintaan tersebut karena nonce yang dikirim tidak cocok dengan nonce terbaru yang tersimpan di server. Dengan demikian, setiap signature hanya dapat digunakan satu kali dan serangan replay dapat digagalkan secara efektif.

3.2 Implementasi Sistem Login Tradisional.

Sistem login tradisional mengintegrasikan algoritma ECDSA dengan database MySQL. Proses registrasi melibatkan pembangkitan pasangan kunci, enkripsi private key menggunakan AES-256-GCM, dan penyimpanan data di database. Implementasi parameterized queries diterapkan untuk mencegah SQL Injection.

```
const query = 'SELECT * FROM users WHERE nim = ? OR email_kampus = ?';
connection.query(query, [login_id, login_id], callback);
```

Gambar.7. Mekanisme pertahanan serangan sql.

Sistem login tradisional mengintegrasikan algoritma ECDSA dengan database MySQL. Proses registrasi melibatkan pembangkitan pasangan kunci, enkripsi private key menggunakan AES-256-GCM, dan penyimpanan data di database. Implementasi parameterized queries diterapkan untuk mencegah SQL Injection.

3.3 Hasil Pengujian SQL Injection.

Tabel 2 menyajikan hasil pengujian SQL Injection pada sistem login tradisional dengan 45 Percobaan serangan menggunakan berbagai jenis teknik sql.

Tabel 2. *Hasil Pengujian SQL Injection.*

Teknik Serangan	Payload	Attempts	Success	Response	Waktu
Classic SQLi	' OR '1'='1	15	0	400 Bad Request	0.002s
Time-Based	' AND SLEEP(5)--	12	0	400 Bad Request	0.002s
Boolean-Based (True)	' AND '1'='1	5	0	400 Bad Request	0.003s
Boolean-Based (False)	' AND '1'='2	5	0	400 Bad Request	0.003s
Union-Based	' UNION SELECT...	8	0	400 Bad Request	0.004s

Hasil pengujian menunjukkan bahwa sistem login tradisional berhasil menangkal 100% serangan SQL Injection. Keberhasilan ini diatribusikan pada implementasi parameterized queries yang memisahkan data dari perintah SQL, serta middleware validasi input yang memeriksa format NIM dan email kampus untuk mencega serangan sql, serangan gagal di karenakan sistem yang di buat menggunakan konsep Blockchain yang terdiri dari block dan chain dan juga node dari indentitas pengguna tersebut maka yang salin terhubung anatara node jaringan penghubung, indentitas penggunak setiap nama dan nim jika di salah barisan serangan sql akan gagal.

3.4 Hasil Pengujian Replay Attack.

Pengujian replay attack pada sistem login blockchain menunjukkan efektivitas mekanisme nonce. Request pertama yang dikirimkan memperoleh respons HTTP 200 OK, menandakan autentikasi berhasil. Request identik yang dikirimkan ulang menghasilkan respons HTTP 400 Bad Request dengan pesan "Invalid nonce: Nonce already used". Pengulangan hingga sepuluh kali menghasilkan respons konsisten.

Analisis menunjukkan bahwa sistem mengimplementasikan in-memory tracking menggunakan struktur data Map() untuk mencatat seluruh nonce yang telah digunakan. Setiap nonce hanya dapat digunakan satu kali dan langsung ditandai sebagai used setelah berhasil digunakan, secara efektif menetralisir serangan replay.

3.5 Hasil Pengujian Signature Forgery.

Pengujian signature forgery dilakukan dengan menggunakan signature valid dari message asli tetapi dikirimkan bersama message yang dimodifikasi. Tabel 3 menyajikan hasil pengujian komprehensif.

Tabel 3. *Hasil Pengujian Signature Forgery.*

Skenario Pengujian	Deskripsi	Hasil
Message Manipulation	Signature valid dari message asli dengan message dimodifikasi	HTTP 401 - Signature mismatch
Signature Malleability	Transformasi nilai (r, -s mod n)	HTTP 400 - Invalid format
Hash Collision Attack	100.000 percobaan mencari collision	0% keberhasilan

Hasil pengujian membuktikan bahwa ethers.verifyMessage() melakukan verifikasi kriptografis yang ketat. Fungsi ini memulihkan alamat dari signature menggunakan algoritma ECDSA, kemudian membandingkannya dengan alamat yang diklaim. Pendekatan matematis ini memastikan bahwa hanya pemilik kunci privat yang sah dapat menghasilkan signature valid untuk message tertentu.

3.6 Hasil Pengujian JWT Tampering.

Pengujian JWT tampering dilakukan dengan empat skenario: modifikasi algoritma JWT menjadi none, brute force terhadap JWT secret, modifikasi claim payload, dan modifikasi token expired. Hasil pengujian menunjukkan bahwa seluruh percobaan gagal dengan respons HTTP 401 Unauthorized.

Implementasi library jsonwebtoken dengan algoritma HS256 dan kunci rahasia yang kuat menjadi fondasi pertahanan yang kokoh. Setiap token diverifikasi baik dari sisi signature maupun claim sebelum diproses lebih lanjut.

3.7 Analisis Perbandingan Keamanan.

Tabel 4 menyajikan perbandingan komprehensif antara sistem login tradisional dan sistem login blockchain.

Tabel 4. *Analisis Komparatif Sistem Autentikasi.*

Aspek	Sistem Tradisional	Sistem Blockchain
Penyimpanan Kredensial	Database MySQL terpusat dengan private key terenkripsi AES-256-GCM	Smart contract terdesentralisasi, private key di wallet pengguna
Mekanisme Autentikasi	Password + dekripsi private key + sign challenge	Passwordless, murni tanda tangan digital ECDSA
Ketahanan SQL Injection	100% (parameterized queries)	Tidak relevan (tanpa database)
Ketahanan Replay Attack	Tidak diimplementasikan	100% (mekanisme nonce)
Ketahanan Signature Forgery	Tinggi (ECDSA)	Tinggi (ECDSA secp256k1)
Identitas Pengguna	NIM/email kampus (terpusat)	Alamat wallet Ethereum (self-sovereign)
Kompleksitas Infrastruktur	Sederhana	Kompleks (membutuhkan blockchain)
Pengalaman Pengguna	Familiar	Membutuhkan pemahaman wallet

3.8 Analisis Kinerja.

Pengukuran kinerja pada kedua sistem menunjukkan hasil yang optimal. Tabel 5 menyajikan karakteristik kinerja sistem.

Tabel 5. *Karakteristik Kinerja Sistem.*

Komponen	Waktu Respons	Keterangan
Signature Verification	50-100 ms	Tergantung CPU
JWT Operations	5-10 ms	sign/verify
Map() Operations	0.1-1 ms	get/set
API Response Time	100-500 ms	end to end
Memory Usage	1KB per user	in Map

Pengujian throughput dengan 1000 request menunjukkan response time konsisten di bawah 100 ms untuk semua endpoint, termasuk endpoint autentikasi yang melibatkan proses kriptografi kompleks. Sistem mampu mempertahankan kinerja optimal sambil menjalankan mekanisme keamanan yang robust.

3.9 PEMBAHASAN.

Hasil penelitian menunjukkan bahwa kedua sistem memiliki keunggulan dan kelemahan masing-masing. Sistem login tradisional dengan implementasi parameterized queries dan validasi input berhasil menangkal 100% serangan SQL Injection dengan waktu respons di bawah 50 milidetik. Keberhasilan ini membuktikan bahwa pendekatan defense-in-depth dengan validasi input berlapis dan prepared statements sangat efektif untuk melindungi database dari serangan injeksi.

Sistem login blockchain menunjukkan ketahanan keamanan yang lebih fundamental. Mekanisme nonce sekali pakai berhasil mencegah serangan replay dengan tingkat keberhasilan 100%. Verifikasi tanda tangan digital ECDSA memastikan keaslian identitas pengguna secara matematis, dengan seluruh percobaan pemalsuan tanda tangan dan serangan JWT tampering mengalami kegagalan total.

Perbedaan paling mendasar terletak pada arsitektur penyimpanan data kredensial. Sistem tradisional mengadopsi model penyimpanan terpusat, menciptakan satu pusat data yang menjadi target utama serangan siber. Sistem blockchain mengimplementasikan penyimpanan terdesentralisasi melalui smart contract, tanpa menyimpan rahasia kriptografis pengguna sama sekali.

Dari perspektif pengalaman pengguna, sistem tradisional menawarkan pengalaman familiar dengan form username-password sederhana. Sistem blockchain menghadirkan pengalaman berbeda yang memerlukan pemahaman tentang wallet digital, meskipun memberikan jaminan keamanan lebih tinggi.

4. KESIMPULAN DAN SARAN.

Penelitian ini berhasil menerapkan dan mengevaluasi dua sistem autentikasi: sistem login tradisional dengan database MySQL yang mengintegrasikan algoritma ECDSA, AES-256-GCM, dan bcrypt; serta sistem login blockchain tanpa password memanfaatkan smart contract Solidity, jaringan lokal Ganache, dan wallet MetaMask. Analisis keamanan dilakukan melalui serangkaian pengujian penetrasi komprehensif.

Hasil penelitian membuktikan bahwa sistem login tradisional dengan implementasi parameterized queries dan validasi input berhasil menangkal 100% serangan SQL Injection (45 dari 45 percobaan gagal) dengan waktu respons konsisten di bawah 50 milidetik. Sistem login blockchain menunjukkan ketahanan keamanan fundamental dengan mekanisme nonce sekali pakai yang berhasil mencegah serangan replay dan verifikasi tanda tangan digital ECDSA yang memastikan keaslian identitas pengguna secara matematis.

Penelitian ini menyimpulkan bahwa kedua sistem memiliki ketahanan keamanan yang sangat baik sesuai dengan mekanisme pertahanan masing-masing. Sistem login tradisional yang diperkuat kriptografi modern terbukti menjadi solusi peningkatan keamanan efektif untuk implementasi jangka pendek dengan performa tinggi dan kemudahan adopsi. Sementara itu, sistem login blockchain menawarkan keunggulan desentralisasi dan pencegahan serangan replay, meskipun menghadapi tantangan pada kompleksitas infrastruktur dan pengalaman pengguna.

Untuk pengembangan lebih lanjut, disarankan migrasi dari in-memory storage ke database persisten pada sistem blockchain, implementasi smart contract yang lebih komprehensif dengan mekanisme role-based access control, pengujian pada jaringan testnet publik Ethereum, serta penambahan mekanisme pemulihan akun seperti social recovery untuk mengatasi risiko kehilangan akses ke private key.

UCAPAN TERIMAKASIH.

Penulis mengucapkan terima kasih kepada Bapak M. Yoga Distra Sudirman, ST., MTI selaku dosen pembimbing skripsi yang telah memberikan arahan, bimbingan, dan motivasi selama penelitian 6 bulan.

Terima kasih saya sampaikan kepada Luthfi Arsyad, alumni Institut Teknologi PLN, yang telah membimbing secara informal dalam pemahaman arsitektur backend, keamanan web, dan teknik troubleshooting,jalur komunikasi backend, frontend dan teknik serangan sql serta mitigasi sql.

DAFTAR PUSTAKA.

[1] [1] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinel, "A survey on essential components of a self-sovereign identity," Computer Science Review, vol. 30, pp. 80-86, 2018, doi: 10.1016/j.cosrev.2018.10.002.

[2] [2] Verizon, "2022 Data Breach Investigations Report," Verizon Enterprise, 2022.

[3] [3] BSSN, "Laporan Insiden Keamanan Siber Tahun 2023," Badan Siber dan Sandi Negara, Jakarta, 2023.

[4] [4] P. Zhang, D. C. Schmidt, J. White, and G. Lenz, "Blockchain technology use cases in healthcare," IEEE Engineering in Medicine and Biology Magazine, vol. 38, no. 2, pp. 12-21, 2019, doi: 10.1109/MEMB.2018.2890453.

[5] [5] European Commission, "Estonia's Digital Identity Ecosystem: A Case Study," Publications Office of the European Union, Brussels, 2022.

[6] [6] C. Allen, "The path to self-sovereign identity," Life with Alacrity, 2016.

[7] [7] M. Kuperberg, P. Robinson, and H. Wang, "Self-sovereign identity implementation: A systematic review," ACM Computing Surveys, vol. 55, no. 3, pp. 1-35, 2023, doi: 10.1145/3494521.

[8] [8] S. Lee, J. Kim, and S. Park, "Decentralized authentication using blockchain for web applications," in Proceedings of the 2022 International Conference on Blockchain Technology, 2022, pp. 112-120, doi: 10.1145/3517998.3518012.

[9] [9] F. D. Garcia, J. Smith, and Y. Tanaka, "Privacy-preserving blockchain authentication for IoT devices," Journal of Information Security and Applications, vol. 72, pp. 103-118, 2023, doi: 10.1016/j.jisa.2023.103118.

[10] [10] L. Wang, Z. Chen, and Y. Liu, "Blockchain-based digital identity management: A survey," IEEE Transactions on Engineering Management, vol. 70, no. 4, pp. 1425-1442, 2023, doi: 10.1109/TEM.2022.3154872.

[11] [11] Microsoft Research, "Decentralized Identity: Ownership and Control," Microsoft Corporation,

2023.

[12] [12] IBM Research, "IBM Blockchain Platform: Technical Overview," IBM Corporation, 2023.

[13] [13] W. Stallings, Cryptography and Network Security: Principles and Practice, 8th ed. London: Pearson, 2021.

[14] [14] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in Proceedings of the 2012 IEEE Symposium on Security and Privacy, 2012, pp. 553-567, doi: 10.1109/SP.2012.44.

[15] [15] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf