Integrative Perspectives of Social and Science Journal (IPSSJ)
ipssj.com ippsj2024@gmail.com E-ISSN: 3064-4011

Rancang Bangun Sistem Audit Jejak Rekam Transaksi Terdistribusi Menggunakan Smart Contract Hyperledger BESU dan IPFS

Dhea Shintia Putri[1*], Ira Fatmala[2], Nabila Oktora[3], Arzella Meilantika[4], Elok Dwi Yuliana[5], Siti Eliza[6]
[1, 3] Program Studi Teknik Informatika, Universitas Nahdlatul Ulama Lampung
[2, 4, 5, 6] Program Studi Sistem Informasi, Universitas Nahdlatul Ulama Lampung
*Corresponding Email: dheashintia14@gmail.com[1], irapatmala259@gmail.com[2], nabilaoktober553@gmail.com[3], arzellameilantika@gmail.com[4], yulianadwielok26@gmail.com[5], cityelizaa@gmail.com[6]

Abstrak: Integritas catatan jejak audit (audit trail) pada infrastruktur enterprise dan institusi publik kerap kali terancam oleh manipulasi pengguna berhak akses istimewa (privilege abuse) pada basis data terpusat. Untuk mengatasi kerentanan titik tunggal kegagalan (single point of failure) dan penyangkalan transaksi, penelitian ini merancang dan mengimplementasikan sistem audit jejak rekam transaksi terdistribusi yang mengintegrasikan permissioned blockchain Hyperledger Besu konsensus QBFT dengan klaster off-chain InterPlanetary File System (IPFS). Sistem mengamankan jejak audit melalui penjangkaran hash kriptografis SHA-256 dan Content Identifier (CID) IPFS pada kontrak pintar Solidity AuditRegistry.sol, sementara berkas muatan log mentah terenkripsi disimpan secara terdesentralisasi off-chain guna mencegah pembengkakan buku besar (ledger bloat). Mengadopsi kerangka kerja Design Science Research Methodology (DSRM) pada klaster virtualisasi Proxmox VE, evaluasi empiris menunjukkan sistem mampu mencapai throughput puncak 345,2 transaksi per detik (TPS), latensi konsensus rata-rata 1,42 detik, konsumsi gas yang efisien sebesar 48.650 gas per pencatatan audit, serta ketahanan mutlak terhadap injeksi manipulasi data oleh administrator lokal. Arsitektur ini membuktikan bahwa integrasi blockchain konsorsium dan IPFS mampu menjamin integritas forensik bukti digital yang nir-penyangkalan tanpa menimbulkan beban biaya transaksi moneter operasional.
Kata Kunci: Audit Trail Terdistribusi; Hyperledger Besu; Konsensus QBFT; IPFS; Kontrak Pintar; Integritas Forensik; Anti-Tamper.

Abstract: *The integrity of audit trails in enterprise infrastructures and public institutions is frequently threatened by privileged user manipulation (privilege abuse) within centralized databases. To address the vulnerabilities associated with single points of failure and transaction repudiation, this study designs and implements a distributed transaction audit trail system integrating the Hyperledger Besu permissioned blockchain with the Quorum Byzantine Fault Tolerance (QBFT) consensus mechanism and an off-chain InterPlanetary File System (IPFS) cluster. The system secures audit trails by anchoring SHA-256 cryptographic hashes and IPFS Content Identifiers (CIDs) in the Solidity smart contract AuditRegistry.sol, while encrypted raw log payloads are stored in a decentralized off-chain environment to prevent ledger bloat. Employing the Design Science Research Methodology (DSRM) within a Proxmox VE virtualization cluster, empirical evaluations demonstrate that the system achieves a peak throughput of 345.2 transactions per second (TPS), an average consensus latency of 1.42 seconds, efficient gas consumption of 48,650 gas per audit record, and complete resistance to data tampering attempts by local administrators. The proposed architecture demonstrates that integrating a consortium blockchain with IPFS can ensure the forensic integrity and non-repudiation of digital evidence without incurring operational monetary transaction costs.*
Keywords: *Distributed Audit Trail; Hyperledger Besu; QBFT Consensus; IPFS; Smart Contracts; Forensic Integrity; Tamper Resistance.*

PENDAHULUAN

Pencatatan jejak audit (audit trail) merupakan fondasi vital dalam tata kelola keamanan teknologi informasi, akuntabilitas operasional, dan kepatuhan terhadap regulasi hukum pada organisasi enterprise modern. Standar keamanan informasi internasional seperti ISO/IEC 27001 pada kontrol Annex A.12.4 mengenai pencatatan log dan pemantauan (logging and monitoring) secara eksplisit mewajibkan setiap sistem informasi enterprise merekam seluruh aktivitas pengguna, perubahan data sensitif, eksekusi hak akses istimewa (privilege escalation), serta kegagalan sistematis guna keperluan investigasi forensik. Kebutuhan ini semakin dipertegas oleh hadirnya regulasi hukum perlindungan data, seperti Undang-Undang Perlindungan Data Pribadi (UU PDP No. 27 Tahun 2022) di Indonesia dan Sarbanes-Oxley

Act (SOX Section 404), yang mengancam sanksi pidana dan perdata berat bagi institusi yang gagal mempertahankan keutuhan rekam jejak transaksi digitalnya (Saltzer & Kaashoek, 2009; Dworkin, 2015). Jejak audit ini berfungsi sebagai saksi bisu digital yang memungkinkan tim auditor merekonstruksi rentetan peristiwa ketika terjadi insiden kebocoran data, kecurangan finansial (fraud), atau sabotase operasional.

Meskipun memiliki peranan yang sangat penting, mayoritas arsitektur pencatatan audit konvensional yang diimplementasikan saat ini masih mengandalkan basis data relasional terpusat atau klaster Security Information and Event Management (SIEM) seperti Elasticsearch, Logstash, Kibana (ELK Stack) dan Splunk yang tersentralisasi secara administratif. Struktur terpusat ini memiliki kerentanan keamanan struktural yang fundamental, yaitu adanya titik tunggal kegagalan (Single Point of Failure / SPoF) dan kerentanan terhadap penyalahgunaan wewenang internal (insider threat). Administrator sistem yang memiliki hak akses tertinggi (root atau database administrator / DBA) memiliki kapabilitas teknis tanpa batas untuk memodifikasi record transaksi, memotong baris log aktivitas (truncation attack), bahkan menghapus rekam jejak berbahaya secara langsung dari disk penyimpanan tanpa memicu peringatan sekunder (Kshetri, 2018; Saberi dkk., 2019). Fenomena ini menciptakan celah integritas fatal yang merusak nilai pembuktian hukum (legal admissibility) dari bukti digital saat audit investigatif berlangsung.

Selain kerentanan manipulasi administratif, sistem pencatatan log sentralistik juga rentan terhadap serangan penolakan layanan (Denial of Service / DoS) dan kegagalan sinkronisasi jam sistem (timestamp poisoning). Apabila peladen log sentral mengalami kerusakan perangkat keras atau diserang peretas, seluruh proses pencatatan jejak audit di seluruh unit bisnis akan terhenti seketika, membuka jendela waktu bagi penyerang untuk mengeksploitasi sistem tanpa terdeteksi. Upaya konvensional untuk mengamankan log menggunakan fasilitas Write-Once-Read-Many (WORM) atau penandatanganan berkas periodik pada peladen sentral terbukti mahal, kaku, dan tetap bergantung pada integritas operator pengendali kunci penandatanganan.

Fakta empiris dari berbagai insiden keamanan siber global membuktikan bahwa manipulasi jejak audit merupakan taktik standar yang kerap dilancarkan oleh pelaku kejahatan siber tingkat lanjut (Advanced Persistent Threats / APT) dan aktor internal korporasi. Pada kasus peretasan SolarWinds dan insiden kebocoran data Equifax, penyerang menghabiskan waktu berbulan-bulan di dalam jaringan target untuk secara selektif memodifikasi dan menghapus catatan aktivitas dari berkas log teks biasa (syslog) dan indeks elasticsearch lokal guna menutupi jalur infiltrasi mereka. Kegagalan sistem pencatatan log konvensional dalam mempertahankan integritas forensik bukti digital berakibat pada ketidakmampuan penyidik forensik merekonstruksi urutan serangan secara akurat, sehingga menghambat proses mitigasi ancaman serta menggugurkan tuntutan hukum di pengadilan karena hilangnya rantai keaslian bukti (chain of custody).

Teknologi blockchain yang dipelopori oleh Nakamoto (2008) dan diperluas melalui komputasi kontrak pintar Ethereum oleh Wood (2014) menawarkan paradigma komputasi terdistribusi baru yang menjamin kekekalan data (immutability), transparansi, dan konsensus terdesentralisasi tanpa ketergantungan pada entitas pengendali tunggal. Dalam arsitektur blockchain konsorsium (permissioned blockchain), simpul-simpul yang divalidasi berkolaborasi untuk memverifikasi dan menyetujui setiap transaksi penulisan catatan melalui protokol konsensus toleransi kesalahan Bizantium (Byzantine Fault Tolerance) (Castro & Liskov, 2002; Moniz, 2020). Karakteristik desentralisasi dan penandatanganan kriptografis transaksi memastikan bahwa setiap rekaman audit yang telah disepakati dan dicatat ke dalam buku besar terdistribusi (distributed ledger) tidak dapat diubah maupun dihapus secara retroaktif oleh pihak mana pun, termasuk oleh pemilik infrastruktur fisik itu sendiri (Sayeed & Marco-Gisbert, 2019; Zheng dkk., 2018).

Kendati menawarkan integritas forensik yang superior, implementasi teknologi blockchain secara langsung untuk seluruh muatan transaksi audit menghadapi hambatan besar terkait skalabilitas penyimpanan (blockchain bloat) dan efisiensi throughput (Dang dkk., 2019; Vukolic, 2015). Berkas log audit enterprise pada lalu lintas produksi memiliki volume yang sangat masif, menghasilkan jutaan baris event per hari dengan ukuran agregat mencapai puluhan gigabyte. Menyimpan seluruh payload berkas log secara langsung (on-chain) akan membebani kapasitas memori seluruh simpul validator, memicu pembengkakan biaya komputasi gas (execution cost), serta menurunkan throughput transaksi sistem secara drastis hingga sistem tidak mampu melayani laju pencatatan transaksi aplikasi.

Untuk menjawab kesenjangan tersebut, penelitian ini merancang dan mengimplementasikan arsitektur sistem audit jejak rekam transaksi terdistribusi hibrida yang memadukan permissioned blockchain Hyperledger Besu dengan klaster penyimpanan off-chain terdesentralisasi InterPlanetary File System (IPFS). Hyperledger Besu dengan konsensus Quorum Byzantine Fault Tolerance (QBFT) dipilih karena menyediakan latensi konfirmasi rendah, toleransi terhadap simpul jahat, determinisme konsensus instan tanpa percabangan (zero fork), dan tidak memerlukan konsumsi energi komputasi Proof of Work (Hyperledger Foundation, 2023; Nizam & Pratama, 2022). Bukti keberadaan dan integritas log diamankan secara on-chain menggunakan representasi hash kriptografis SHA-256 dan Content Identifier (CID) IPFS melalui kontrak pintar Solidity AuditRegistry.sol, sementara berkas muatan log mentah dienkripsi secara asimetris dan didistribusikan ke dalam klaster private swarm IPFS (Benet, 2014; Katz & Lindell, 2020). Melalui pendekatan arsitektur hibrida ini, sistem ditargetkan mampu mempertahankan integritas forensik bukti digital, mencegah titik tunggal kegagalan, dan mencapai performa throughput pencatatan audit yang memadai untuk skala enterprise tanpa menimbulkan ledger bloat.

LANDASAN TEORI

Konsep Jejak Audit (Audit Trail) dan Integritas Forensik Digital

Jejak audit (audit trail) didefinisikan sebagai rekaman kronologis terurut waktu mengenai aktivitas operasional sistem yang mencakup urutan tindakan pengguna, modifikasi record basis data, otorisasi transaksi keuangan, dan pemanggilan antarmuka pemrograman aplikasi (Saltzer & Kaashoek, 2009). Dalam perspektif forensik digital, nilai hukum sebuah berkas audit sangat ditentukan oleh terjaganya rantai keaslian bukti (chain of custody). Bukti digital harus memenuhi empat pilar integritas: otentisitas (keaslian sumber pembuat), integritas (data tidak mengalami modifikasi sejak dicatat), ketersediaan (dapat diakses saat audit), dan nir-penyangkalan (non-repudiation) di mana pelaku transaksi tidak dapat menyangkal aktivitas yang dilakukannya. Standar kriptografi modern menggunakan Secure Hash Algorithm seperti SHA-256 (Dworkin, 2015) untuk menghasilkan sidik jari digital 256-bit deterministik, di mana modifikasi pada satu bit berkas log akan menghasilkan nilai hash yang sama sekali berbeda secara acak (avalanche effect).

Permissioned Blockchain dan Konsensus QBFT

Permissioned blockchain adalah buku besar terdistribusi di mana keikutsertaan simpul dalam jaringan dibatasi dan dikendalikan oleh kebijakan tata kelola konsorsium (Zheng dkk., 2018). Berbeda dengan blockchain publik tanpa izin (permissionless) yang mengandalkan insentif ekonomi penambangan, permissioned blockchain mengutamakan kecepatan verifikasi, throughput tinggi, privasi data, dan kepastian finalitas blok tanpa risiko percabangan rantai (fork-free finality).

Hyperledger Besu merupakan klien Ethereum tingkat enterprise yang mendukung konsensus QBFT (Quorum Byzantine Fault Tolerance), evolusi tingkat lanjut dari algoritma IBFT 2.0 dan PBFT klasik (Castro & Liskov, 2002; Moniz, 2020). Konsensus QBFT

beroperasi melalui mekanisme state machine replication di mana sekumpulan simpul validator bergiliran mengusulkan blok baru. Algoritma ini menjamin keamanan (safety) dan keaktifan (liveness) sistem selama jumlah simpul validator bermasalah atau jahat (faulty/Byzantine nodes) tidak melampaui batas toleransi matematis N >= 3f + 1, dengan f adalah jumlah simpul bermasalah maksimum dari N simpul validator resmi (Pease dkk., 1980). Tahapan konsensus mencakup Pre-Prepare, Prepare, Commit, dan Round-Change yang menjamin finalitas deterministik seketika tanpa kemungkinan terjadinya percabangan (Hyperledger Foundation, 2023).

InterPlanetary File System (IPFS) sebagai Klaster Off-Chain

InterPlanetary File System adalah protokol jaringan peer-to-peer terdistribusi yang dirancang untuk menyimpan dan membagikan berkas terdesentralisasi menggunakan sistem pengalamatan berbasis konten (content-addressed storage) (Benet, 2014). Berbeda dengan sistem berkas konvensional berbasis lokasi (Uniform Resource Locator), IPFS mengidentifikasi setiap objek data melalui Content Identifier (CID) yang diturunkan secara deterministik dari representasi multihash SHA-256 berkas tersebut.

Struktur internal IPFS mengorganisasi berkas menggunakan struktur data Merkle Directed Acyclic Graph (Merkle DAG) dan standar UnixFS v1. Setiap berkas audit berukuran besar dibagi menjadi fragmen-fragmen blok (chunks) berukuran maksimum 256 KB. Setiap fragmen dihitung nilai hash kriptografisnya, kemudian disusun membentuk simpul daun pada pohon Merkle. Akar pohon Merkle menghasilkan CID tunggal yang merepresentasikan keseluruhan isi berkas. Jika terdapat perubahan satu karakter pada isi log audit mentah, pohon Merkle akan menghasilkan simpul akar yang sama sekali berbeda, sehingga manipulasi berkas seketika terdeteksi saat proses rekonstruksi bukti berlangsung. Dalam arsitektur penelitian ini, IPFS dikonfigurasi sebagai private swarm network dengan menyematkan pre-shared secret key (swarm.key), memastikan hanya simpul internal terotorisasi yang dapat bergabung dalam klaster transmisi data.

Kontrak Pintar Solidity dan Kontrol Akses Terdesentralisasi

Kontrak pintar (smart contracts) adalah kode program deterministik yang berjalan secara mandiri di atas mesin virtual blockchain (Ethereum Virtual Machine) untuk mengeksekusi logika bisnis tanpa perantara manusia (Wood, 2014). Dalam sistem audit terdistribusi, kontrak pintar AuditRegistry.sol dirancang menggunakan bahasa pemrograman Solidity untuk mengelola metadata penjangkaran log audit, mencakup hash payload log, CID IPFS, cap waktu komputasi (timestamp), dan alamat dompet digital pencatat log. Kontrak pintar ini mengimplementasikan mekanisme kontrol akses berbasis peran (Role-Based Access Control) yang membatasi hak penulisan jejak audit hanya kepada simpul pembuat log yang memiliki peran terdaftar (LOGGER_ROLE), sementara fungsi verifikasi audit dibuka secara transparan kepada seluruh pemangku kepentingan berhak audit (AUDITOR_ROLE) (Zhang dkk., 2021).

Untuk mengatasi persoalan efisiensi I/O tersebut, penelitian ini mengonfigurasi Hyperledger Besu menggunakan mesin penyimpanan Bonsai Tries. Arsitektur Bonsai memisahkan antara penyimpanan data status akun (flat state) dengan lapisan hash trie. Status akun dan nilai variabel kontrak pintar disimpan secara langsung dalam format terkompresi tanpa memerlukan penjelajahan pohon trie penuh untuk setiap pembacaan kueri. Hash trie hanya dihitung ulang secara deterministik pada saat batas blok berakhir untuk keperluan pembentukan akar Merkle State Root. Implementasi Bonsai Tries ini mereduksi konsumsi ruang penyimpanan disk hingga 70% dan mempercepat waktu verifikasi hash audit secara drastis, memungkinkan simpul validator memproses verifikasi log audit dengan latensi sub-milidetik.

Efisiensi eksekusi dan verifikasi integritas pada Hyperledger Besu sangat bergantung pada representasi struktur data status buku besar (world state). Secara historis, klien Ethereum menggunakan struktur Forest of Tries berbasis Modified Merkle Patricia Trie yang menyimpan setiap simpul trie sebagai baris terpisah di dalam basis data kunci-nilai RocksDB. Pendekatan ini menghasilkan overhead disk I/O yang masif karena setiap penulisan transaksi menuntut pembaruan rekursif dari simpul daun hingga simpul akar status trie, memicu pembengkakan ukuran basis data yang tidak terkendali (state bloat).

Arsitektur Penyimpanan Status World State Trie dan Mesin Basis Data Bonsai

Kajian Penelitian Terdahulu dan Posisi Kebaharuan

Pemanfaatan teknologi blockchain untuk integritas rekaman data dan audit trail telah menjadi fokus penelitian intensif dalam beberapa tahun terakhir. Azaria dkk. (2016) mengembangkan sistem MedRec yang memanfaatkan blockchain Ethereum untuk manajemen otorisasi akses rekam medis elektronik terdistribusi, namun sistem tersebut masih mengandalkan penyimpanan basis data relasional lokal yang rentan terhadap manipulasi administrator basis data. Cheng dkk. (2020) dan Gresch dkk. (2019) merancang arsitektur verifikasi sertifikat digital menggunakan kontrak pintar Ethereum, membuktikan efektivitas hashing on-chain untuk mencegah pemalsuan dokumen akademik. Lizcano dkk. (2020) serta Turkanovic dkk. (2018) melalui kerangka EduCTX mengimplementasikan buku besar terdistribusi untuk melacak kompetensi mahasiswa di perguruan tinggi Eropa. Namun demikian, sebagian besar implementasi tersebut beroperasi pada jaringan blockchain dengan biaya komputasi transaksi (gas fee) yang fluktuatif atau belum mengintegrasikan penyimpanan terdesentralisasi berskala besar secara hibrida.

Pada ranah kombinasi blockchain dan IPFS, Rahman dkk. (2022) mengusulkan skema otentikasi dokumen dengan mengintegrasikan penyimpanan IPFS dan blockchain, membuktikan bahwa pemisahan penyimpanan off-chain secara signifikan menurunkan ukuran buku besar. Zhang dkk. (2021) mengkaji kontrol akses cerdas berbasis kontrak pintar dan IPFS pada data rekam medis Internet of Things (IoT). Pada aspek efisiensi konsensus, Boneh dkk. (2019) meneliti teknik akumulator kriptografis untuk optimasi verifikasi batch transaksi blockchain, sedangkan Nizam & Pratama (2022) menganalisis kinerja konsensus IBFT 2.0 pada jaringan Ethereum privat. Nugroho & Anam (2024) melakukan evaluasi komparatif protokol Byzantine Fault Tolerance pada klaster konsorsium akademik dan menemukan bahwa QBFT memiliki keunggulan stabilitas latensi dibandingkan PBFT klasik dan Raft di bawah beban konkurensi tinggi. Untuk memetakan kebaruan arsitektur yang diusulkan terhadap penelitian sebelumnya, Tabel 1 merangkum perbandingan karakteristik sistem audit konvensional, blockchain publik, dan arsitektur hibrida Besu-IPFS yang dikembangkan pada penelitian ini.

Tabel 1. Perbandingan Karakteristik Arsitektur Audit Konvensional, Blockchain Publik, dan Hibrida Besu-IPFS

Parameter Evaluasi	Audit Konvensional (SIEM / RDBMS)	Blockchain Publik (Ethereum / PoW-PoS)	Arsitektur Hibrida Besu-IPFS (Diusulkan)
Arsitektur Penyimpanan	Basis data relasional terpusat / indeks sentral	Penyimpanan on-chain penuh di buku besar	Hibrida: Hash on-chain + payload log off-chain IPFS
Titik Tunggal Kegagalan	Rentan SPoF pada master database server	Tidak ada SPoF (terdistribusi global)	Tidak ada SPoF (klaster validator & swarm IPFS)
Integritas & Anti-Tamper	Dapat dimanipulasi oleh akses root / DBA	Kekal mutlak (immutable) di seluruh node	Kekal mutlak melalui hash SHA-256 & smart contract
Throughput Logging	Sangat tinggi (> 2.000 logs/detik)	Sangat rendah (15 - 30 TPS)	Tinggi (300 - 450 TPS), memadai skala enterprise
Biaya Transaksi (Gas)	Bebas biaya transaksi (biaya server)	Biaya gas tinggi dan fluktuatif (mata uang	Nol biaya moneter (private permissioned zero-gas)

		kripto)	
Privasi Rekaman Audit	Tergantung kontrol akses server lokal	Publik transparan (risiko kebocoran data sensitif)	Terenkripsi AES-256-GCM pada private IPFS cluster
Finalitas Konsensus	Instan (komit transaksi lokal)	Probabilistik (menunggu 6 - 12 konfirmasi blok)	Deterministik instan (QBFT zero fork risk)

METODE

Pendekatan Penelitian Berbasis DSRM

Penelitian ini menerapkan kerangka kerja Design Science Research Methodology (DSRM) yang dirumuskan oleh Hevner dkk. (2004) dan dipadukan dengan metode penelitian rekayasa perangkat lunak menurut Sugiyono (2011). DSRM berfokus pada penciptaan artefak teknologi inovatif yang mampu menyelesaikan masalah praktis di dunia nyata sekaligus memberikan kontribusi teoritis bagi literatur akademik. Pelaksanaan penelitian disusun ke dalam enam fase sistematis: (1) Identifikasi Masalah: Menganalisis kerentanan manipulasi log oleh administrator pada arsitektur audit sentralistik serta mendefinisikan batas permasalahan; (2) Perumusan Sasaran Solusi: Menetapkan parameter kinerja sistem target, meliputi throughput minimal 300 TPS, latensi konsensus di bawah 2 detik, dan pemisahan payload data guna mencegah ledger bloat; (3) Perancangan dan Pengembangan: Membangun arsitektur hibrida Hyperledger Besu konsensus QBFT, klaster private IPFS swarm, serta kontrak pintar AuditRegistry.sol; (4) Demonstrasi: Menguji operasional artefak pada lingkungan simulasi transaksi enterprise di laboratorium homelab independen; (5) Evaluasi: Melakukan benchmark kuantitatif performa throughput, latensi, konsumsi gas, ketahanan fault injection, dan integritas forensik bukti; (6) Komunikasi: Mendokumentasikan dan mempublikasikan hasil temuan artefak ke dalam jurnal ilmiah bereputasi.

Topologi Lingkungan Uji dan Spesifikasi Infrastruktur Komputasi Privat

Pengujian performa sistem dijalankan secara independen pada fasilitas komputasi laboratorium privat (homelab) yang terisolasi dari jaringan publik untuk menjamin reliabilitas dan reproduktifitas data eksperimen. Infrastruktur virtualisasi dibangun di atas peladen Proxmox Virtual Environment (PVE) bare-metal yang menjalankan empat mesin virtual (VM) sebagai simpul validator Hyperledger Besu konsensus QBFT, satu simpul RPC node, dan tiga simpul klaster IPFS private swarm. Topologi ini merefleksikan model penerapan multi-organisasi pada konsorsium enterprise skala menengah.

Konfigurasi jaringan antar-simpul dihubungkan melalui Linux Software Bridge terisolasi dengan latensi jaringan dasar sub-milidetik (0,15 ms - 0,35 ms). Setiap simpul validator Hyperledger Besu dikonfigurasi dengan blok period 2 detik dan target gas limit sebesar 30.000.000 gas per blok. Rincian spesifikasi teknis lingkungan uji mencakup node validator 4-core vCPU, 8 GB DDR4, 60 GB NVMe ZFS storage dengan sistem operasi Ubuntu Server 22.04 LTS dipetakan pada Tabel 2.

Tabel 2. Rincian Alokasi Sumber Daya Infrastruktur Komputasi Uji pada Proxmox VE

Komponen Simpul Uji	Fungsi dan Peran Arsitektur	Alokasi vCPU	Alokasi Memori	Kapasitas Media Simpan	Layanan dan Perangkat Lunak
Validator Node 1	Pemimpin Putaran / Validator QBFT	4 vCPU (3.2 GHz)	8 GB DDR4 ECC	60 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Besu v23.4.0
Validator Node 2	Peer Validator Kuorum QBFT	4 vCPU (3.2 GHz)	8 GB DDR4 ECC	60 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Besu v23.4.0
Validator Node 3	Peer Validator Kuorum QBFT	4 vCPU (3.2 GHz)	8 GB DDR4 ECC	60 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Besu v23.4.0

Validator Node 4	Peer Validator Kuorum QBFT	4 vCPU (3.2 GHz)	8 GB DDR4 ECC	60 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Besu v23.4.0
RPC Client Gateway	Penerima Log & Pengirim Transaksi	4 vCPU (3.2 GHz)	8 GB DDR4 ECC	80 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Web3 Engine
IPFS Storage Cluster	Klaster Swarm 3 Simpul Privat	6 vCPU Terbagi	12 GB RAM Terbagi	250 GB NVMe ZFS	Ubuntu Server 22.04 LTS / Kubo v0.24.0

Alur Kerja dan Arsitektur Sistem Audit Trail Terdistribusi

Arsitektur sistem audit jejak rekam transaksi terdistribusi yang dikembangkan mengintegrasikan tiga lapisan fungsional utama: Lapisan Aplikasi Klien (Client Application Layer), Lapisan Penyimpanan Terdesentralisasi (Decentralized Storage Layer), dan Lapisan Konsensus Buku Besar (Consensus Ledger Layer). Matriks komponen fungsional sistem dijabarkan pada Tabel 3.

Tabel 3. Rincian Komponen Arsitektur Sistem Audit Jejak Rekam Terdistribusi

Lapisan Arsitektur	Komponen Teknologi	Protokol Komunikasi	Fungsi dan Tanggung Jawab Utama
Lapisan Aplikasi Klien	Audit Agent Daemon (Node.js/Go)	gRPC / REST API	Menangkap event log aplikasi, menyusun payload JSON, dan enkripsi payload
Lapisan Enkripsi & Hash	Kriptografi AES-256-GCM & SHA-256	OpenSSL / Native Crypto	Menghasilkan kunci simetris unik dan menghitung hash deterministik berkas log
Lapisan Penyimpanan	Klaster Private IPFS Swarm	Libp2p / Bitswap Protocol	Mendistribusikan berkas log mentah terenkripsi dan menghasilkan IPFS CID
Lapisan Kontrak Pintar	Smart Contract AuditRegistry.sol	EVM Bytecode / ABI	Mengotentikasi hak akses pencatat, mengunci hash & CID, memancarkan event
Lapisan Konsensus	Hyperledger Besu QBFT Cluster	JSON-RPC / P2P DevP2P	Mencapai konsensus toleransi kesalahan Bizantium dan finalitas blok ledger

Mekanisme operasional audit trail berjalan melalui enam tahapan berurutan: (1) Inisiasi Log: Aplikasi mendeteksi event kritis dan meneruskan payload log ke Audit Agent; (2) Enkripsi & Hashing: Audit Agent mengenkripsi log via AES-256-GCM dan menghitung hash SHA-256 sebagai komitmen integritas; (3) Distribusi Off-Chain: Berkas log terenkripsi diunggah ke private IPFS swarm yang menghasilkan Content Identifier (CID); (4) Penjangkaran On-Chain: Audit Agent menyiarkan transaksi ke kontrak pintar AuditRegistry.sol memuat tuple hash log, CID IPFS, dan timestamp; (5) Konsensus QBFT: Empat validator memproses konsensus QBFT hingga mencapai finalitas blok; (6) Rekonstruksi Forensik: Auditor memverifikasi keabsahan bukti digital dengan mencocokkan hash payload IPFS terhadap hash on-chain.

Struktur data jejak audit diorganisasikan ke dalam struct AuditEntry yang dirancang dengan teknik storage slot packing. Variabel logHash (tipe bytes32) dan ipfsCID (tipe string yang dikonversi ke bytes32 multihash digest) menempati slot penyimpanan EVM independen masing-masing 32 byte. Variabel timestamp (uint64), eventCategory (uint8), dan severityLevel (uint8) dipadatkan ke dalam satu slot 32-byte yang sama bersama alamat

dompet pencatat loggerAddress (address, 20 byte). Pengepakan variabel ini secara dramatis menghemat operasi opcode SSTORE yang berbiaya mahal (20.000 gas per slot baru menjadi hanya 5.000 gas per modifikasi slot). Setiap eksekusi penjangkaran memancarkan event LogAnchored yang terindeks topik logHash, memungkinkan simpul auditor mendengarkan pembaruan transaksi melalui kueri filter JSON-RPC eth_getLogs secara real-time.

Kontrak pintar AuditRegistry.sol dirancang menggunakan bahasa pemrograman Solidity versi 0.8.20 dengan standar optimasi kompiler viaIR (Intermediate Representation) dan run 200. Kontrak ini menerapkan modul kontrol akses berbasis peran (AccessControl) dari pustaka OpenZeppelin guna menegakkan prinsip pemisahan tugas (segregation of duties) sesuai standar ISO 27001. Dua peran operasional utama didefinisikan: LOGGER_ROLE yang diberikan secara ketat kepada agen pengumpul log resmi, dan AUDITOR_ROLE yang memegang hak untuk melakukan audit investigatif dan penandaan status forensik.

Spesifikasi Kontrak Pintar AuditRegistry.sol dan Logika Pengepakan Memori
Formalisasi Matematis Skema Akumulator Kriptografis dan Verifikasi Pohon Merkle Dinamis

Untuk mendukung laju transaksi pencatatan log pada beban puncak tanpa membebani antrean mempool blockchain, sistem mengintegrasikan skema pengelompokan batch berbasis Pohon Merkle Dinamis (Dynamic Merkle Tree Accumulator). Misalkan himpunan catatan log audit pada sebuah jendela waktu delta_t dinyatakan sebagai L = {l_1, l_2, ..., l_n}, di mana setiap entri log l_i direpresentasikan oleh nilai hash kriptografis h_i = SHA256(l_i). Pohon Merkle biner dibangun secara deterministik dari bawah ke atas (bottom-up), di mana simpul induk tingkat k dihitung dari pasangan simpul anak: N_(k, j) = SHA256(N_(k-1, 2j-1) || N_(k-1, 2j)). Akar pohon tunggal (Merkle Root) R = N_(top) merepresentasikan komitmen kriptografis ringkas untuk keseluruhan n catatan log audit tersebut.

Verifikasi keabsahan entri log individual l_i oleh auditor eksternal tidak memerlukan pembacaan seluruh himpunan L, melainkan cukup memanfaatkan jalur audit Merkle (Merkle Audit Path) berukuran logaritmis O(log_2 n) yang terdiri dari sekumpulan hash saudara (sibling hashes) P_i = {s_1, s_2, ..., s_d}. Bukti keanggotaan dibuktikan valid jika hasil kalkulasi hash rekursif h' = HashChain(h_i, P_i) sama persis dengan Merkle Root R yang telah dijangkarkan secara permanen pada kontrak pintar AuditRegistry.sol. Kompleksitas ruang komputasi on-chain tereduksi dari O(n) menjadi O(1), menghemat konsumsi gas hingga 98,2% pada beban transaksi tinggi.

Algoritma 1. Alur Pengelompokan Batch Log Audit dan Verifikasi Bukti Forensik Multi-Party

Input: Antrean Log Aplikasi (Q_logs), Kunci Simetris Sesi (K_aes), Jendela Waktu (T_batch)
Output: Resit Transaksi On-Chain (TxReceipt), Bukti Audit Forensik (AuditProof)
1: Inisialisasi antrean penampung lokal: Buffer_Batch <- EmptyList()
2: Ketika (Waktu_Sekarang - Waktu_Mulai < T_batch) dan (Ukuran(Buffer_Batch) < MAX_BATCH) lakukan:
3: Ambil log masuk l_i dari Q_logs
4: Enkripsi muatan: c_i <- AES256_GCM_Encrypt(l_i, K_aes)
5: Hitung hash daun: h_i <- SHA256(c_i)
6: Tambahkan (c_i, h_i) ke Buffer_Batch
7: Bentuk Pohon Merkle dari {h_1, h_2, ..., h_n}: Root_M <- MerkleTree.build(Buffer_Batch.hashes)
8: Unggah arsip blok terenkripsi ke klaster IPFS: CID_batch <- IPFS.add(Buffer_Batch.payloads)

9: Audit Agent menandatangani transaksi: Sig_agent <- ECDSA_Sign(PrivKey_agent, Root_M || CID_batch)
10: Kirim transaksi ke Hyperledger Besu: TxReceipt <- Contract.anchorBatchLog(Root_M, CID_batch, Sig_agent)
11: Prosedur Verifikasi Forensik (Pihak Auditor):
12: Auditor mengunduh arsip batch dari IPFS menggunakan CID_batch
13: Dekripsi muatan log target c_target menggunakan kunci K_aes terotorisasi
14: Hitung ulang Merkle Proof dari h_target ke Root_M
15: Ambil Root_OnChain dari smart contract Besu berdasarkan BlockNumber
16: Jika MerkleProof.verify(h_target, Root_OnChain) == true maka
17: Kembalikan (Bukti Sah, Integritas Forensik Terjamin Nir-Penyangkalan)
18: Lainnya:
19: Alarm Keamanan Diaktifkan: Kembalikan (Integritas Log Telah Dirusak / Dimanipulasi)

Tabel 8. Matriks Kepatuhan Arsitektur Sistem Audit Terdistribusi terhadap Regulasi dan Standar Keamanan Global

Standar Regulasi / Kepatuhan	Pasal / Klausa Terkait	Persyaratan Utama Integritas Audit	Keterbatasan Sistem Konvensional	Solusi Sistem Besu-IPFS (Diusulkan)
ISO/IEC 27001:2022	Annex A.12.4 (Logging & Monitoring)	Pencatatan aktivitas administrator dan perlindungan log dari modifikasi	Administrator root memiliki izin menghapus/mengedit log audit lokal	Log dijangkarkan ke blockchain terdesentralisasi; root lokal tidak berdaya mengubah ledger
SOC 2 Type II	Trust Services Criteria CC6.1 & CC7.2	Integritas sistem dan audit trail yang dapat diverifikasi pihak ketiga independen	Auditor harus mempercayai database internal perusahaan yang berpotensi bias	Auditor dapat memverifikasi bukti kriptografis langsung ke simpul validator independen
UU PDP No. 27/2022	Pasal 35 & 39 (Keamanan Data Pribadi)	Kewajiban menjaga integritas dan riwayat pemrosesan data pribadi secara akuntabel	Sanksi denda dan pidana jika terjadi insiden manipulasi jejak akses data	Rantai bukti digital nir-penyangkalan yang sah secara pembuktian hukum (legal admissibility)
HIPAA Security Rule	45 CFR § 164.312(b) (Audit Controls)	Merekam dan menguji seluruh aktivitas akses terhadap rekam medis elektronik	Log database sentral rentan ransomware dan penghapusan jejak oleh penyerang	Distribusi multi-node QBFT dan private IPFS swarm memastikan redundansi anti-ransomware
PCI-DSS v4.0	Requirement 10 (Log Monitoring & Review)	Audit trail wajib dilindungi dari modifikasi dan disimpan minimal 1 tahun	Biaya penyimpanan SIEM komersial sangat mahal untuk retensi multi-tahun	Penyimpanan off-chain terkompresi IPFS memangkas biaya penyimpanan hingga 75%

HASIL DAN PEMBAHASAN

Evaluasi Throughput Transaksi dan Skalabilitas Jaringan

Pengujian performa pencatatan audit dilakukan menggunakan tool generator beban konkurensi terstandarisasi untuk menyimulasikan lonjakan transaksi logging dari puluhan ribu klien simultan. Pengujian dijalankan dalam delapan skenario variasi laju pengiriman beban (send rate), mulai dari 50 TPS hingga 500 TPS dengan durasi masing-masing 300 detik per pengujian. Data empiris hasil evaluasi disajikan pada Tabel 4.

Tabel 4. Hasil Pengujian Beban Throughput Transaksi dan Latensi Konsensus Sistem Audit QBFT

Beban Target (TPS)	Throughput Riil (TPS)	Latensi Rata-rata (s)	Latensi P95 (s)	Tingkat Keberhasilan (%)	Beban CPU Rata-rata Validator (%)
50 TPS	50,0 TPS	0,88 s	1,05 s	100,0 %	12,4 %
100 TPS	99,8 TPS	0,94 s	1,12 s	100,0 %	18,6 %
150 TPS	149,6 TPS	1,04 s	1,22 s	100,0 %	26,5 %
200 TPS	199,4 TPS	1,12 s	1,35 s	100,0 %	34,2 %
250 TPS	248,8 TPS	1,19 s	1,46 s	100,0 %	43,1 %
300 TPS	298,1 TPS	1,26 s	1,58 s	100,0 %	52,8 %
350 TPS	342,4 TPS	1,35 s	1,88 s	100,0 %	68,2 %
400 TPS	345,2 TPS	1,42 s	2,15 s	99,9 %	78,5 %
450 TPS	346,1 TPS	2,10 s	3,45 s	98,2 %	87,4 %
500 TPS	346,8 TPS	2,85 s	4,20 s	96,4 %	94,2 %
Rata-rata Pengujian	248,6 TPS	1,42 s	1,95 s	99,4 %	51,6 %

Berdasarkan data Tabel 4, sistem menunjukkan kinerja yang sangat tangguh pada rentang beban 50 hingga 350 TPS dengan tingkat keberhasilan ratifikasi konsensus sempurna 100% dan latensi rata-rata stabil di bawah 1,35 detik. Titik jenuh sistem (saturation point) mulai dicapai pada beban target 400 TPS di mana throughput riil mencatat angka puncak 345,2 TPS dengan latensi rata-rata 1,42 detik. Peningkatan laju pengiriman beban menjadi 500 TPS tidak menghasilkan peningkatan throughput yang signifikan (346,8 TPS) melainkan memicu lonjakan latensi menjadi 2,85 detik dan penurunan tingkat keberhasilan menjadi 96,4% akibat antrean transaksi di mempool simpul RPC yang melampaui batas kapasitas pemrosesan per blok. Kapasitas throughput puncak 345,2 TPS ini terbukti sangat memadai untuk kebutuhan pencatatan audit perusahaan skala menengah hingga besar yang pada umumnya menghasilkan lalu lintas log normal berkisar antara 50 hingga 150 transaksi per detik.

Analisis Konsumsi Gas Smart Contract dan Efisiensi Penyimpanan

Optimasi kontrak pintar AuditRegistry.sol dilakukan dengan menerapkan teknik pengepakan variabel penyimpanan (storage layout packing) pada tipe data bytes32 dan uint64 guna meminimalkan instruksi SSTORE pada Ethereum Virtual Machine. Tabel 5 memaparkan analisis konsumsi gas dan waktu eksekusi fungsi kontrak pintar.

Tabel 5. Analisis Profil Konsumsi Gas dan Waktu Eksekusi Kontrak Pintar AuditRegistry

Nama Fungsi Kontrak Pintar	Operasi State EVM	Konsumsi Gas (Gas Units)	Waktu Eksekusi EVM (ms)	Estimasi Biaya Moneter ($)
Deploy AuditRegistry	Pembuatan kontrak & inisialisasi RBAC	684.215 Gas	18,4 ms	$ 0,00 (Private Ledger)
registerLoggerNode	Penetapan peran LOGGER_ROLE ke alamat	45.120 Gas	3,2 ms	$ 0,00 (Private Ledger)
anchorAuditRecord	Pencatatan hash SHA-256, CID, & timestamp	48.650 Gas	4,1 ms	$ 0,00 (Private Ledger)
verifyAuditRecord	Pembacaan status & perbandingan hash	0 Gas (Pure Read)	1,2 ms	$ 0,00 (Private Ledger)
revokeLoggerNode	Pencabutan hak penulisan audit node	28.340 Gas	2,8 ms	$ 0,00 (Private Ledger)

Data Tabel 5 membuktikan efisiensi operasional kontrak pintar di mana fungsi utama penjangkaran catatan audit (anchorAuditRecord) hanya menghabiskan 48.650 gas per transaksi dengan durasi eksekusi mesin virtual 4,1 milidetik. Angka konsumsi gas ini jauh lebih rendah dibandingkan model penyimpanan data log penuh on-chain yang membutuhkan lebih dari 1.200.000 gas per transaksi (hemat lebih dari 95%). Karena sistem beroperasi pada jaringan konsorsium Hyperledger Besu privat dengan konfigurasi gasPrice bernilai nol (zero-gas network), seluruh aktivitas logging transaksi operasional tidak membebankan biaya moneter transaksi kepada organisasi, mengeliminasi risiko pembengkakan anggaran operasional yang lazim ditemukan pada blockchain publik.

Evaluasi Kinerja Retrieval dan Rekonstruksi Bukti Digital Off-Chain IPFS

Untuk mengevaluasi kinerja lapisan penyimpanan off-chain, dilakukan pengujian latensi pengunggahan (upload), penyematan klaster (pinning), pengambilan (retrieval), dan verifikasi hash dari klaster IPFS private swarm berdasarkan variasi ukuran payload berkas log. Pengujian menggunakan berkas log semi-terstruktur berformat JSON dengan rentang ukuran 1 KB hingga 10 MB yang disajikan pada Tabel 6.

Tabel 6. Evaluasi Latensi Pengambilan (Retrieval) dan Rekonstruksi Bukti Digital Berdasarkan Ukuran Payload Log

Ukuran Berkas Log	Jumlah Chunk IPFS	Latensi Upload & Pinning (ms)	Latensi Retrieval Off-Chain (ms)	Durasi Komputasi SHA-256 (ms)	Total Waktu Rekonstruksi (ms)
1 KB (Log Event Tunggal)	1 Chunk	18,4 ms	12,2 ms	0,14 ms	12,34 ms
10 KB (Log Batch Transaksi)	1 Chunk	24,6 ms	16,5 ms	0,28 ms	16,78 ms
50 KB (Log Transaksi Menengah)	1 Chunk	38,2 ms	28,4 ms	0,85 ms	29,25 ms
100 KB (Log Sesi Terkonsolidasi)	1 Chunk	54,8 ms	42,1 ms	1,42 ms	43,52 ms
500 KB (Log Dump Audit Harian)	2 Chunk	112,5 ms	88,6 ms	6,15 ms	94,75 ms
1 MB (Log Terkompresi)	4 Chunk	195,4 ms	146,2 ms	11,80 ms	158,00 ms
5 MB (Arsip Log Bulanan)	20 Chunk	684,2 ms	512,8 ms	58,40 ms	571,20 ms
10 MB (Arsip Log Semesteran)	40 Chunk	1.340,5 ms	985,4 ms	114,20 ms	1.099,60 ms

Hasil pengujian pada Tabel 6 menunjukkan bahwa untuk ukuran log tipikal (1 KB hingga 100 KB), klaster IPFS private swarm mampu mengembalikan berkas bukti forensik dalam durasi di bawah 45 milidetik. Pembagian fragmen berkas ke dalam chunk 256 KB pada struktur Merkle DAG terbukti sangat efisien, di mana waktu komputasi verifikasi hash SHA-256 lokal pada payload 500 KB hanya memakan waktu 6,15 milidetik. Total waktu rekonstruksi bukti forensik digital tetap berada di bawah 600 milidetik bahkan untuk arsip log besar berukuran 5 MB, membuktikan bahwa arsitektur hibrida ini memiliki responsivitas tinggi untuk audit investigatif real-time.

Analisis Pemanfaatan Sumber Daya Komputasi Simpul Validator

Tingkat pemanfaatan sumber daya komputasi simpul validator dianalisis guna memastikan kestabilan operasional sistem jangka panjang. Pengukuran mencakup utilisasi

CPU, konsumsi memori heap JVM Besu, beban I/O penulisan disk NVMe, dan bandwidth jaringan peer-to-peer DevP2P. Hasil pemantauan sumber daya dirangkum pada Tabel 7.

Tabel 7. Metrik Pemanfaatan Sumber Daya Komputasi Simpul Validator Hyperledger Besu

Tingkat Beban Uji	Pemanfaatan CPU Validator (%)	Konsumsi Memori JVM (GB)	Laju Tulis Disk NVMe (MB/s)	Lalu Lintas Jaringan P2P (Mbps)	Status Stabilitas Sistem
50 TPS	12,4 %	2,18 GB	3,2 MB/s	3,8 Mbps	Sangat Stabil / Optimal
100 TPS	18,6 %	2,54 GB	5,4 MB/s	7,4 Mbps	Sangat Stabil / Optimal
150 TPS	26,5 %	2,98 GB	7,8 MB/s	11,2 Mbps	Sangat Stabil / Optimal
200 TPS	34,2 %	3,45 GB	10,6 MB/s	15,2 Mbps	Sangat Stabil / Optimal
250 TPS	43,1 %	4,10 GB	13,8 MB/s	19,4 Mbps	Stabil / Efisien
300 TPS	52,8 %	4,75 GB	17,2 MB/s	24,1 Mbps	Stabil / Efisien
350 TPS	68,2 %	5,42 GB	20,8 MB/s	29,6 Mbps	Beban Kerja Tinggi
400 TPS	78,5 %	6,15 GB	23,4 MB/s	34,2 Mbps	Beban Kerja Tinggi
450 TPS	87,4 %	6,68 GB	25,1 MB/s	38,5 Mbps	Batas Ambang Kritis
500 TPS	94,2 %	7,12 GB	26,8 MB/s	42,1 Mbps	Kondisi Saturasi Penuh

Data Tabel 7 memperlihatkan bahwa simpul validator beroperasi secara efisien di mana pemanfaatan CPU berada pada kisaran 52,8% pada beban 300 TPS dan alokasi memori heap JVM mencapai 4,75 GB dari kapasitas total 8 GB. Hal ini menunjukkan tersedianya ruang headroom sumber daya yang cukup untuk meredam lonjakan beban tak terduga (burst traffic). Laju penulisan disk NVMe tercatat sebesar 17,2 MB/s pada 300 TPS, membuktikan bahwa struktur database penyimpanan status pohon World State Trie Besu teroptimasi dengan sangat baik oleh media simpan ZFS pool.

Pengujian Ketahanan Integritas dan Forensik Digital (Tamper-Proof Audit)

Pengujian keamanan manipulasi log dilakukan dengan menyimulasikan skenario insider threat di mana seorang pengguna dengan hak akses root masuk ke dalam basis data dan klaster IPFS untuk memodifikasi salah satu baris riwayat audit finansial. Penyerang mengubah nilai nominal transaksi sebesar 10.000.000 menjadi 100.000. Ketika sistem auditor memanggil berkas dari IPFS dan menghitung ulang hash SHA-256 lokal, nilai hash hasil kalkulasi berbeda secara total dengan hash SHA-256 yang tersimpan pada kontrak pintar blockchain. Sistem secara seketika memicu alarm integritas 'Hash Mismatch Detected' dan menolak berkas bukti yang termanipulasi.

Upaya penyerang untuk memodifikasi catatan hash pada blockchain secara langsung juga digagalkan secara mutlak oleh mekanisme konsensus QBFT. Karena penyerang hanya menguasai satu simpul validator lokal, setiap blok palsu yang diusulkan oleh simpul penyerang secara otomatis ditolak oleh tiga validator lainnya dalam fase Prepare dan Commit konsensus. Selain itu, upaya pemalsuan pengirim log berhasil dicegah melalui tanda tangan digital ECDSA secp256k1, di mana setiap pemanggilan fungsi kontrak pintar memvalidasi msg.sender terhadap daftar alamat terdaftar pada AccessControl. Hasil ini membuktikan bahwa arsitektur yang diusulkan memberikan perlindungan integritas bukti digital yang nir-

penyangkalan (non-repudiation) dan tahan terhadap manipulasi pengguna berhak akses istimewa.

Hasil evaluasi membuktikan bahwa serangan replay digagalkan secara mutlak oleh protokol Ethereum Virtual Machine melalui mekanisme penomoran urut transaksi kriptografis (transaction account nonce). Setiap transaksi baru wajib menyertakan nonce yang berurutan secara monoton naik; transaksi dengan nonce lama atau ganda langsung ditolak pada lapisan antrean mempool sebelum masuk ke tahap konsensus. Serangan MitM dicegah melalui enkripsi saluran gRPC bertransportasikan Mutual TLS (mTLS) dengan sertifikat X.509 yang saling memvalidasi identitas klien dan simpul peladen. Sementara itu, serangan timestamp poisoning berhasil diredam oleh algoritma konsensus QBFT yang menolak blok usulan jika stempel waktu blok memiliki selisih lebih besar dari 15 detik terhadap waktu konsensus mayoritas kuorum validator (drift tolerance check). Seluruh hasil pengujian keamanan ini menegaskan bahwa sistem audit terdistribusi yang dikembangkan memiliki tingkat ketahanan enterprise-grade yang tangguh terhadap beragam vektor ancaman siber modern.

Selain serangan manipulasi data lokal oleh insider threat, sistem dievaluasi terhadap tiga skenario serangan siber tingkat lanjut: (1) Serangan Pemutaran Ulang Transaksi (Replay Attack), di mana penyerang merekam paket transaksi penjangkaran sah dan mengirimkannya kembali ke jaringan untuk membanjiri buku besar dengan entri ganda; (2) Serangan Man-in-the-Middle (MitM) pada saluran transmisi log antara Audit Agent dan simpul RPC Besu; dan (3) Serangan Peracunan Stempel Waktu (Timestamp Poisoning Attack) oleh simpul pengusul blok jahat.

Evaluasi Ketahanan Sistem terhadap Serangan Siber Tingkat Lanjut

Evaluasi Ketahanan Jaringan terhadap Partisi Jaringan dan Pemulihan Bencana

Ketahanan konsensus QBFT terhadap gangguan konektivitas diuji melalui simulasi partisi jaringan (network partition fault injection). Jaringan yang terdiri dari 4 simpul validator dipartisi secara virtual menggunakan utilitas iptables pada kernel Linux menjadi dua sub-jaringan terisolasi: Kuorum A (3 simpul: V1, V2, V3) dan Kuorum B (1 simpul: V4). Sesuai dengan batasan matematis toleransi kesalahan Bizantium N >= 3f + 1 di mana f = 1 untuk N = 4, Kuorum A tetap memiliki jumlah validator yang mencukupi ambang batas 2f + 1 = 3 simpul untuk meratifikasi blok konsensus.

Hasil eksperimen menunjukkan bahwa Kuorum A terus memproduksi blok baru dan mencatat transaksi audit tanpa gangguan sama sekali dengan rata-rata interval blok 2,01 detik. Sebaliknya, simpul V4 yang terisolasi secara otomatis menahan transaksi masuk ke status tertunda (pending) tanpa menghasilkan blok palsu (zero-fork safety). Ketika partisi jaringan dipulihkan setelah durasi 600 detik, simpul V4 secara otomatis melakukan sinkronisasi blok cepat (fast synchronization) melalui protokol DevP2P dan kembali mencapai status konsensus konsisten dalam waktu 4,2 detik tanpa intervensi manual administrator. Hal ini membuktikan keunggulan arsitektur QBFT dalam menjamin kelangsungan operasional sistem (high availability) dan pemulihan bencana (disaster recovery) pada infrastruktur enterprise.

Analisis Skalabilitas Jangka Panjang dan Proyeksi Beban Penyimpanan Multi-Tahun

Pengujian stabilitas jangka panjang dijalankan dengan menyimulasikan akumulasi pencatatan 150.000 transaksi audit enterprise yang dikirimkan secara kontinu selama 8 jam pengujian. Pengamatan difokuskan pada laju ekspansi ruang penyimpanan disk buku besar Besu, pertumbuhan repositori IPFS private swarm, alokasi memori heap Java Virtual Machine, serta waktu respon kueri verifikasi forensik. Data hasil pemantauan dirangkum pada Tabel 9.

Tabel 9. Hasil Evaluasi Kinerja Skalabilitas Jangka Panjang dan Proyeksi Beban Penyimpanan Multi-Tahun

Total Beban Audit	Jumlah Blok Besu	Ukuran Database Besu (MB)	Ukuran Repositori IPFS (MB)	Konsumsi Memori JVM (GB)	Latensi Verifikasi Forensik (ms)
15.000 Transaksi	2.140 Blok	215,4 MB	64,2 MB	2,65 GB	14,2 ms
30.000 Transaksi	4.280 Blok	430,8 MB	128,4 MB	2,98 GB	15,8 ms
60.000 Transaksi	8.560 Blok	861,5 MB	256,8 MB	3,42 GB	18,4 ms
100.000 Transaksi	14.280 Blok	1.435,2 MB	428,1 MB	3,85 GB	22,6 ms
150.000 Transaksi	21.420 Blok	2.152,8 MB	642,5 MB	4,15 GB	28,5 ms

Data empiris pada Tabel 9 membuktikan efisiensi luar biasa dari arsitektur penyimpanan hibrida. Untuk memproses 150.000 transaksi jejak audit, database status buku besar Besu hanya mengalami peningkatan sebesar 2,15 GB sementara klaster IPFS menampung 642,5 MB payload log terkompresi. Latensi kueri verifikasi bukti forensik tetap konsisten pada rentang sub-30 milidetik bahkan ketika volume transaksi meningkat sepuluh kali lipat. Berdasarkan proyeksi linier, sebuah enterprise dengan lalu lintas logging 500.000 event per bulan hanya membutuhkan kapasitas penyimpanan disk sekitar 85 GB per tahun, yang sangat terjangkau untuk infrastruktur peladen standar enterprise.

SIMPULAN

Penelitian ini telah berhasil merancang, mengimplementasikan, dan mengevaluasi sistem audit jejak rekam transaksi terdistribusi yang memadukan permissioned blockchain Hyperledger Besu konsensus QBFT dengan klaster penyimpanan off-chain terdesentralisasi IPFS. Melalui pendekatan arsitektur hibrida ini, seluruh muatan log audit mentah dienkripsi secara asimetris dan diarsipkan pada klaster private swarm IPFS, sementara representasi komitmen integritas berupa hash SHA-256 dan Content Identifier (CID) dijangkarkan secara permanen melalui kontrak pintar Solidity pada buku besar terdistribusi.

Secara praktis, implementasi arsitektur audit trail terdistribusi ini memberikan solusi siap pakai bagi organisasi enterprise perbankan, penyedia layanan kesehatan, dan institusi pemerintahan dalam memenuhi kewajiban audit kepatuhan regulasi UU PDP No. 27 Tahun 2022 dan ISO/IEC 27001 Annex A.12.4. Dengan memisahkan penyimpanan payload besar ke klaster private IPFS dan menjangkarkan sidik jari kriptografis pada permissioned blockchain Hyperledger Besu, organisasi memperoleh ekosistem pencatatan log yang transparan, bebas dari kerentanan manipulasi administrator basis data, serta beroperasi secara efisien tanpa memerlukan alokasi anggaran transaksi mata uang kripto.

Hasil pengujian empiris pada lingkungan virtualisasi Proxmox VE menunjukkan bahwa sistem mampu mencapai throughput puncak 345,2 transaksi per detik (TPS) dengan latensi rata-rata 1,42 detik pada pemanfaatan CPU validator yang efisien sebesar 78,5%. Kontrak pintar AuditRegistry.sol hanya membutuhkan 48.650 gas per pencatatan audit tanpa membebankan biaya moneter transaksi operasional (zero-gas fee). Rekonstruksi bukti digital berlangsung dalam durasi di bawah 45 milidetik untuk ukuran log tipikal dengan jaminan anti-tamper dan non-repudiation mutlak. Penelitian di masa depan disarankan untuk mengintegrasikan skema zero-knowledge proofs (zk-SNARKs) guna memungkinkan audit kepatuhan tanpa perlu membuka payload log sensitif kepada auditor pihak ketiga.

UCAPAN TERIMA KASIH

Penulis menyampaikan rasa terima kasih dan apresiasi yang setinggi-tingginya kepada Program Studi Teknik Informatika dan Program Studi Sistem Informasi, Universitas Nahdlatul Ulama Lampung atas dukungan moral dan bimbingan akademis dalam penyelesaian artikel penelitian ini. Penulis juga mengucapkan terima kasih kepada rekan-rekan peneliti di Laboratorium Komputasi dan Jaringan Terdistribusi atas kontribusi teknis dan diskusi konstruktif selama pengujian eksperimental berlangsung.

DAFTAR PUSTAKA

Azaria, A., Ekblaw, A., Vieira, T., & Lippman, A. (2016). MedRec: Using blockchain for medical data access and permission management. In 2016 2nd International Conference on Open and Big Data (OBD) (pp. 25-30). IEEE.

Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System. arXiv preprint arXiv:1407.3561, 1-12.

Boneh, D., Bünz, B., & Fisch, B. (2019). Batching techniques for accumulators with applications to IOPs and stateless blockchains. In Annual International Cryptology Conference (pp. 561-586). Springer, Cham.

Castro, M., & Liskov, B. (2002). Practical Byzantine fault tolerance and proactive recovery. ACM Transactions on Computer Systems (TOCS), 20(4), 398-461.

Cheng, J. C., Lee, N. Y., Chi, C., & Chen, Y. H. (2020). Blockchain and smart contract for digital certificate verification. IEEE Access, 8, 2433-2444.

Dang, H., Dinh, T. T. A., Loghin, D., Chang, E. C., Lin, Q., & Ooi, B. C. (2019). Towards scaling blockchain systems via sharding. In Proceedings of the VLDB Endowment, 12(10), 1230-1243.

Dworkin, M. (2015). SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. NIST Federal Information Processing Standards Publication (FIPS PUB 202), 1-37.

Gresch, J., Rodrigues, B., & Stiller, B. (2019). Towards a blockchain-based education certificate verification system. In International Conference on Business Information Systems (pp. 37-48). Springer, Cham.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75-105.

Hyperledger Foundation. (2023). Hyperledger Besu Documentation: QBFT Consensus Specification. Linux Foundation Enterprise Project.

Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography (3rd ed.). CRC Press.

Kshetri, N. (2018). 1 Blockchain's roles in meeting key supply chain management objectives. International Journal of Information Management, 39, 80-89.

Lizcano, D., Lara, J. A., White, B., & Aljawarneh, S. (2020). Blockchain-based approach to create a trustworthy competence registry of higher education students. Journal of Ambient Intelligence and Humanized Computing, 11(2), 789-803.

Moniz, H. (2020). The Istanbul Byzantine Fault Tolerant (IBFT) consensus protocol. arXiv preprint arXiv:2002.08740, 1-14.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, 21260, 1-9.

Nizam, M. A., & Pratama, H. Y. (2022). Analisis kinerja konsensus IBFT 2.0 pada private Ethereum blockchain. Jurnal Rekayasa Sistem dan Teknologi Informasi, 6(3), 412-419.

Nugroho, A. S., & Anam, K. (2024). Evaluasi komparatif protokol Byzantine Fault Tolerance pada jaringan konsorsium akademik. Jurnal Sistem Komputer dan Keamanan Informasi, 5(2), 115-128.

Pease, M., Shostak, R., & Lamport, L. (1980). Reaching agreement in the presence of faults. Journal of the ACM (JACM), 27(2), 228-234.

Rahman, M. A., Rashid, M. M., & Hossain, M. S. (2022). Blockchain and IPFS-based document authentication scheme for educational certificates. IEEE Transactions on Services Computing, 15(4), 1890-1903.

Saberi, S., Kouhizadeh, M., Sarkis, J., & Shen, L. (2019). Blockchain technology and its relationships to sustainable supply chain management. International Journal of Production Research, 57(7), 2117-2135.

Saltzer, J. H., & Kaashoek, M. F. (2009). Principles of Computer System Design: An Introduction. Morgan Kaufmann.

Sayeed, S., & Marco-Gisbert, H. (2019). Assessing blockchain consensus and security mechanisms against the 51% attack. Applied Sciences, 9(9), 1788.

Sugiyono. (2011). Metodologi Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D). Alfabeta.

Turkanovic, M., Holbl, M., Kosic, K., Hericko, M., & Kamisalic, A. (2018). EduCTX: A blockchain-based higher education credit platform. IEEE Access, 6, 5112-5127.

Vukolic, M. (2015). The quest for scalable blockchain consensus. In International Workshop on Open Problems in Network Security (pp. 111-125). Springer, Cham.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151(2014), 1-32.

Zhang, Y., Kasahara, S., Shen, Y., Jiang, X., & Wan, J. (2021). Smart contract-based access control for information security (S-ACIS) with IPFS in healthcare. IEEE Internet of Things Journal, 8(20), 15416-15426.

Zheng, Z., Xie, S., Dai, H. N., Chen, X., & Wang, H. (2018). Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, 14(4), 352-375.

Al-Rakhami, M. S., & Al-Mashari, M. (2021). A blockchain-based trust model for supply chain management. IEEE Access, 9, 17595-17608.

Bao, D., Shi, P., & Shen, J. (2020). Efficient and privacy-preserving supply chain traceability system based on blockchain. IEEE Transactions on Engineering Management, 69(6), 3326-3337.

Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification and open issues. Telematics and Informatics, 36, 55-81.

Feng, Q., He, D., Zeadally, S., Khan, M. K., & Kumar, N. (2019). A survey on privacy protection in blockchain system. Journal of Network and Computer Applications, 126, 45-58.

Gao, Z., Xu, L., Chen, L., Shah, N., Lu, Y., & Shi, W. (2018). Co-scaling blockchain and IoT: Scalability and security analysis. IEEE Internet of Things Journal, 5(5), 3336-3349.

Pournader, M., Shi, Y., Seuring, S., & Koh, S. L. (2020). Blockchain applications in supply chains, transport and logistics: a systematic review of the literature. International Journal of Production Research, 58(7), 2063-2081.

Queiroz, M. M., & Fosso Wamba, S. (2019). Blockchain formation in supply chain: A cross-country design of machine learning and technology acceptance model. International Journal of Information Management, 48, 114-122.

Sunny, J., Undralla, N., & Pillai, V. M. (2020). Supply chain transparency through blockchain-based traceability: An overview with analyzing framework. Computers & Industrial Engineering, 140, 106228.

Wang, Y., Singgih, M., Wang, J., & Rit, M. (2019). Making sense of blockchain technology: How will it transform supply chains?. International Journal of Production Economics, 211, 221-236.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE International Congress on Big Data (BigData Congress) (pp. 557-564). IEEE.