 ISSN:2302-8149
e-ISSN:2540-9719

Rancang Bangun Sistem Berbagi Berkas dengan Jaminan Integritas dan Audit Trail menggunakan *Permissioned Blockchain* dan IPFS: Studi Kasus DISPUSIPDA Jawa Barat

Design and Development of a File Sharing System with Integrity Assurance and Audit Trail using Permissioned Blockchain and IPFS: A Case Study of DISPUSIPDA West Java

[1]Sansan Syahrul Hidayah*, [2]Ichsan Ibrahim
[1,2]Teknik Informatika, STMIK IM, Bandung, Indonesia
[1,2]Jl. Belitung No. 7, Merdeka, Kec. Sumur Bandung, Kota Bandung, Jawa Barat, Indonesia
*e-mail: san.syahrul.7@gmail.com, ichsanibrahim@stmik-im.ac.id

(***received*:** 11 May 2026, ***revised*:** 24 May 2026, ***accepted*:** 25 May 2026)

Abstrak

Transformasi digital pada instansi pemerintahan menuntut sistem berbagi berkas yang mampu menjamin integritas dokumen, keamanan akses, transparansi aktivitas pengguna, dan audit trail yang dapat ditelusuri. Penelitian ini merancang dan mengimplementasikan sistem berbagi berkas berbasis *permissioned* blockchain Hyperledger Besu dan InterPlanetary File System (IPFS) untuk pengelolaan dokumen di Dinas Perpustakaan dan Kearsipan Daerah (DISPUSIPDA) Jawa Barat. Metode yang digunakan adalah Design Science Research Methodology (DSRM), mencakup identifikasi masalah, perancangan, implementasi, demonstrasi, dan evaluasi sistem. Sistem mengintegrasikan IPFS sebagai penyimpanan dokumen secara *off-chain*, blockchain untuk pencatatan metadata dan audit trail *immutable*, serta *smart contract* untuk mencatat perubahan hak akses secara permanen. Dokumen dienkripsi menggunakan AES-256 sebelum disimpan, sedangkan integritas diverifikasi melalui perbandingan hash SHA-256. Hasil pengujian menunjukkan seluruh fitur utama berjalan sesuai kebutuhan, manipulasi dokumen berhasil terdeteksi pada seluruh skenario uji, dan mekanisme *revocation* efektif membatasi akses pengguna yang hak aksesnya telah dicabut. *Latency* transaksi blockchain berada dalam rentang yang dapat diterima, sedangkan waktu unggah IPFS meningkat seiring bertambahnya ukuran berkas. Penelitian ini berkontribusi pada rancangan sistem berbagi berkas yang mengintegrasikan enkripsi, RBAC, *revocation*, IPFS, dan *permissioned* blockchain dalam satu arsitektur untuk mendukung integritas dokumen dan audit institusional pada lingkungan pemerintahan.

Kata kunci: audit trail, blockchain, integritas dokumen, IPFS, kontrol akses, *permissioned* blockchain

Abstract

Digital transformation in government institutions requires file-sharing systems capable of ensuring document integrity, secure access control, user activity transparency, and traceable audit trails. This study designs and implements a file-sharing system based on Hyperledger Besu and InterPlanetary File System (IPFS) for document management at the Dinas Perpustakaan dan Kearsipan Daerah Jawa Barat (DISPUSIPDA). The research employed the Design Science Research Methodology (DSRM), which includes problem identification, system design, implementation, demonstration, and evaluation stages. The proposed system integrates IPFS for off-chain document storage, blockchain technology for immutable metadata recording and audit trails, and smart contracts to permanently record access rights modifications. Documents are encrypted using AES-256 before storage, while integrity verification is performed through SHA-256 hash comparison. The testing results demonstrate that all core system features operated according to the specified requirements. Document manipulation attempts were successfully detected across all testing scenarios, and the revocation

 ISSN:2302-8149
e-ISSN:2540-9719

mechanism effectively restricted access for users whose permissions had been revoked. Blockchain transaction latency remained within an acceptable range, while IPFS upload time increased proportionally with file size. This study contributes to the development of a file-sharing system architecture that integrates encryption, Role-Based Access Control (RBAC), revocation mechanisms, IPFS, and permissioned blockchain technology within a unified framework to support document integrity and institutional auditability in government environments.

Keywords: *access control, audit trail, blockchain, document integrity, IPFS, permissioned blockchain.*

Pendahuluan

Transformasi digital pada instansi pemerintahan mendorong peningkatan kebutuhan terhadap sistem berbagi berkas elektronik yang mampu menjamin integritas data, transparansi aktivitas pengguna, keamanan akses, serta kepatuhan terhadap audit institusional. Dalam konteks pengelolaan arsip publik, instansi seperti Dinas Perpustakaan dan Kearsipan Daerah (DISPUSIPDA) Jawa Barat membutuhkan mekanisme pengelolaan dokumen yang tidak hanya memudahkan distribusi berkas antarunit kerja, tetapi juga mampu memastikan bahwa setiap dokumen memiliki keaslian yang dapat diverifikasi dan riwayat aktivitas yang dapat ditelusuri secara akuntabel [1]. Kebutuhan ini menjadi semakin krusial karena dokumen arsip pemerintahan memiliki nilai administratif, hukum, dan kelembagaan yang harus dijaga dari perubahan tidak sah, kehilangan data, maupun penyalahgunaan akses.

Sistem berbagi berkas yang umum digunakan pada lingkungan organisasi masih banyak bergantung pada arsitektur penyimpanan terpusat. Pendekatan ini memiliki beberapa keterbatasan mendasar: audit log umumnya disimpan dalam basis data yang dapat dimodifikasi oleh administrator sehingga tingkat kepercayaan terhadap rekaman aktivitas menjadi terbatas [2]; ketergantungan pada satu server menimbulkan risiko single point of failure; serta mekanisme kontrol akses yang belum cukup granular menyebabkan proses pencabutan akses tidak selalu tercatat secara sistematis dan sulit diaudit setelah perubahan hak akses dilakukan.

Beberapa penelitian sebelumnya telah mengeksplorasi integrasi blockchain dan IPFS sebagai pendekatan hybrid dalam pengembangan sistem berbagi dokumen [3], [4], [5]. Namun, sebagian besar masih menggunakan blockchain publik yang kurang sesuai untuk lingkungan pemerintahan karena menimbulkan tantangan privasi dan biaya transaksi [2],[6]. Selain itu, implementasi kontrol akses berbasis peran belum dirancang spesifik untuk struktur organisasi pemerintahan [7], [8], mekanisme *revocation* belum diintegrasikan secara langsung dengan audit trail *immutable* [7], [9] dan sebagian besar penelitian berfokus pada domain sertifikasi digital, rekam medis, atau sistem voting sehingga belum mengakomodasi kebutuhan audit institusional pemerintahan [10], [11]. Dengan demikian, belum ada penelitian yang menghadirkan integrasi lengkap antara enkripsi, RBAC, *revocation*, IPFS, dan blockchain dalam satu sistem kohesif untuk pengelolaan arsip instansi pemerintahan [5], [7].

Berdasarkan kesenjangan tersebut, penelitian ini mengusulkan rancang bangun sistem berbagi berkas berbasis *permissioned blockchain* Hyperledger Besu dan IPFS untuk mendukung pengelolaan dokumen pada DISPUSIPDA Jawa Barat. *Permissioned blockchain* dipilih karena partisipasi node dapat dibatasi pada entitas yang berwenang, sehingga lebih sesuai dengan kebutuhan organisasi pemerintahan dibandingkan blockchain publik. Kontribusi utama penelitian meliputi: (1) perancangan arsitektur sistem berbagi berkas berbasis *permissioned blockchain* dan IPFS untuk konteks kearsipan pemerintahan; (2) implementasi RBAC pada *backend* serta pencatatan *grant access* dan *revocation* melalui *smart contract* sebagai bagian dari audit trail *immutable*; (3) pengembangan mekanisme verifikasi integritas dokumen berbasis hash SHA-256 dan blockchain; serta (4) evaluasi sistem melalui pengujian fungsional, deteksi manipulasi dokumen, analisis performa, dan efektivitas *revocation*.

Tinjauan Literatur

Blockchain merupakan teknologi *distributed ledger* yang memungkinkan pencatatan transaksi secara terdistribusi dan tidak dapat diubah (*immutable*). Teknologi ini telah digunakan secara luas untuk meningkatkan integritas data dan transparansi sistem melalui mekanisme konsensus yang

 ISSN:2302-8149
e-ISSN:2540-9719

menghilangkan ketergantungan pada otoritas tunggal [2], [3]. Selain itu, *smart contract* memungkinkan otomatisasi validasi transaksi serta pencatatan aktivitas pengguna secara transparan dan tidak dapat disangkal [12], [13]. Meskipun demikian, sebagian besar implementasi blockchain dalam penelitian sebelumnya masih menggunakan blockchain publik, yang memiliki keterbatasan dalam hal biaya transaksi dan privasi data organisasi [2], [6]. Oleh karena itu, diperlukan pendekatan yang lebih sesuai dengan kebutuhan instansi, seperti penggunaan *permissioned blockchain*.

IPFS merupakan sistem penyimpanan terdistribusi yang menggunakan mekanisme *content addressing* berbasis hash untuk mengidentifikasi berkas secara unik melalui *Content Identifier* (CID). Pendekatan ini memungkinkan verifikasi integritas data secara langsung serta meningkatkan ketersediaan data melalui jaringan *peer-to-peer* tanpa ketergantungan pada server tunggal [14], [3]. Santoso [14] mengkaji infrastruktur IPFS untuk *censorship resistance* dan membuktikan keunggulan desentralisasi penyimpanan, namun penelitiannya tidak membahas kebutuhan audit organisasi maupun tata kelola distribusi dokumen internal instansi pemerintah. Athanere dan Thakur [3] mengusulkan pendekatan hierarkis semi-terdesentralisasi berbasis blockchain dan IPFS yang terbukti mengurangi *single point of failure* serta meningkatkan auditabilitas akses data. Namun demikian, IPFS tidak menyediakan mekanisme kontrol akses maupun audit aktivitas pengguna secara bawaan, sehingga perlu dikombinasikan dengan teknologi lain untuk memenuhi kebutuhan keamanan sistem [14].

Integrasi blockchain dan IPFS telah digunakan dalam berbagai penelitian sebagai pendekatan *hybrid* dalam sistem berbagi berkas. Steichen et al. [8] menjadi salah satu penelitian awal yang mengintegrasikan *smart contract* Ethereum dengan IPFS untuk sistem berbagi berkas dengan kontrol akses terdistribusi (*ACL-IPFS*), namun masih berbasis blockchain publik dan belum mempertimbangkan privasi organisasi. Naz et al. [9] mengusulkan platform berbagi data aman berbasis blockchain dan IPFS dengan skema *secret sharing*, membuktikan efisiensi komputasi lebih baik dibandingkan enkripsi AES, namun mekanisme *revocation* belum dibahas eksplisit. Sun et al. [15] mengembangkan skema penyimpanan rekam medis elektronik berbasis blockchain dan IPFS dengan enkripsi berbasis atribut (CP-ABE) yang memungkinkan kontrol akses granular, meski belum diadaptasi untuk kebutuhan audit institusional pemerintahan. Dalam konteks kearsipan, Asmiyanto et al. [5] mengembangkan prototipe sistem arsip berbasis blockchain dan IPFS yang lebih dekat dengan kebutuhan pengelolaan dokumen institusional, namun belum mencakup mekanisme *audit trail immutable* dan *revocation* yang teraudit. Kumar et al. [4] dan Muis [7] juga menunjukkan efektivitas integrasi ini, namun keduanya masih menggunakan blockchain publik tanpa mempertimbangkan mekanisme *revocation* maupun kepatuhan audit lembaga pemerintah.

Role-Based Access Control (RBAC) merupakan model kontrol akses yang memberikan hak akses berdasarkan peran pengguna dalam organisasi, bukan berdasarkan identitas pengguna secara individual. Dalam sistem berbagi berkas institusional, RBAC memungkinkan pembatasan hak akses seperti melihat, mengunggah, mengunduh, memperbarui, dan mencabut akses dokumen sesuai peran pengguna, misalnya admin, pemilik dokumen, staf, atau pimpinan. Pada sistem berbasis blockchain, *smart contract* dapat digunakan untuk mencatat perubahan hak akses, memvalidasi aksi tertentu, serta menghasilkan event yang dapat ditelusuri sebagai bagian dari audit trail. Dengan demikian, integrasi RBAC dan *smart contract* relevan untuk meningkatkan akuntabilitas kontrol akses, terutama pada lingkungan organisasi yang membutuhkan pencatatan perubahan kewenangan secara transparan dan sulit dimanipulasi.

Untuk memperjelas posisi penelitian ini terhadap studi sebelumnya, Tabel 1 menyajikan perbandingan pendekatan berdasarkan jenis blockchain dan penyimpanan, mekanisme kontrol akses dan keamanan, dukungan *revocation*, audit trail, serta keterbatasan utama masing-masing penelitian.

Tabel 1 Komparasi penelitian terdahulu

Penelitian	Pendekatan	Kontrol Akses dan Keamanan	*Revocation*	Audit Trail	Keterbatasan Utama
Steichen et al. [8]	Blockchain publik dan IPFS	ACL berbasis *smart contract*; enkripsi terbatas	Belum kuat	Ada	Belum sesuai untuk organisasi privat karena masih menggunakan

 ISSN:2302-8149
e-ISSN:2540-9719

					blockchain publik
Naz et al. [9]	Blockchain publik dan IPFS	Mekanisme berbagi data aman dengan *secret sharing*	Belum eksplisit	Ada	Fokus pada keamanan berbagi data, tetapi belum membahas pencabutan akses secara rinci
Sun et al. [15]	Blockchain publik dan IPFS	CP-ABE untuk kontrol akses granular	Terbatas	Ada	Berfokus pada rekam medis elektronik, belum pada audit arsip pemerintahan
Asmiyanto et al. [5]	Blockchain dan IPFS untuk arsip	Kontrol akses belum dijelaskan secara kuat; enkripsi belum spesifik	Belum dibahas	Terbatas	Belum mengintegrasikan audit trail *immutable* dan *revocation* secara eksplisit
Penelitian ini	*Permissioned blockchain* Hyperledger Besu dan IPFS	RBAC pada *backend*, enkripsi AES-256, serta pencatatan grant/revoke pada *smart contract*	Ada	Ada	Masih berupa prototipe lokal dengan pengujian satu node dan manajemen kunci terpusat

Berdasarkan Tabel 1, penelitian terdahulu telah menunjukkan potensi integrasi blockchain dan IPFS dalam mendukung integritas serta penyimpanan dokumen terdistribusi. Namun, sebagian besar studi masih menggunakan blockchain publik, belum mengintegrasikan *revocation* secara eksplisit dengan audit trail *immutable*, dan belum sepenuhnya disesuaikan dengan kebutuhan tata kelola arsip pada lingkungan pemerintahan. Oleh karena itu, penelitian ini mengusulkan integrasi *permissioned blockchain*, IPFS, RBAC, enkripsi AES-256, *revocation*, dan audit trail dalam satu prototipe sistem berbagi berkas untuk mendukung integritas dokumen dan akuntabilitas akses pada instansi pemerintahan.

Metode Penelitian

1. Metodologi Penelitian

Penelitian ini menggunakan pendekatan Design Science Research Methodology (DSRM) karena berfokus pada perancangan, pengembangan, demonstrasi, dan evaluasi artefak sistem informasi. DSRM dipilih karena sesuai untuk penelitian yang menghasilkan prototipe teknologi sebagai solusi atas permasalahan nyata pada lingkungan organisasi. Pemetaan tahapan DSRM terhadap aktivitas dan luaran penelitian disajikan pada Tabel 2.

Tabel 2. Pemetaan tahapan DSRM

No	Tahap DSRM	Aktivitas Utama	Luaran
1	Identifikasi masalah	Analisis kebutuhan berbagi berkas, integritas dokumen, audit trail, kontrol akses, dan *revocation* pada DISPUSIPDA Jawa Barat.	Rumusan masalah dan model ancaman.
2	Penentuan tujuan solusi	Menetapkan solusi berbasis *permissioned blockchain*, IPFS, RBAC, enkripsi, dan audit trail	Tujuan solusi dan kebutuhan sistem.

 ISSN:2302-8149
e-ISSN:2540-9719

		immutable.	
3	Perancangan dan pengembangan artefak	Merancang arsitektur, model integritas, kontrol akses, *smart contract*, *backend*, *frontend*, serta integrasi IPFS dan Hyperledger Besu.	Prototipe sistem dan komponen aplikasi.
4	Demonstrasi sistem	Menjalankan fitur login, unggah, unduh, verifikasi, audit trail, *grant access*, dan *revocation* pada lingkungan Docker.	Prototipe berjalan dan skenario demonstrasi.
5	Evaluasi	Menguji fungsi sistem, tamper detection, latensi blockchain, waktu unggah IPFS, dan *revocation*.	Data hasil pengujian dan analisis evaluasi.
6	Komunikasi hasil	Menyusun artikel ilmiah berdasarkan rancangan, implementasi, pengujian, dan pembahasan.	Artikel ilmiah untuk publikasi.

Berdasarkan Tabel 2, penelitian ini diawali dengan identifikasi masalah pada sistem berbagi berkas yang berjalan, kemudian dilanjutkan dengan perumusan tujuan solusi, perancangan dan pengembangan artefak, demonstrasi prototipe, evaluasi sistem, serta komunikasi hasil penelitian. Alur tersebut memastikan bahwa artefak yang dikembangkan tidak hanya dirancang secara konseptual, tetapi juga diuji melalui skenario fungsional, integritas dokumen, performa, dan *revocation*.

2. Analisis Kebutuhan dan Threat Model

Tahap analisis kebutuhan dilakukan melalui observasi terhadap sistem berbagi berkas yang berjalan di DISPUSIPDA Jawa Barat. Hasil observasi menunjukkan bahwa sistem masih bergantung pada penyimpanan terpusat (Google Drive) tanpa mekanisme verifikasi integritas dan *audit trail* yang memadai. Kondisi ini berpotensi menimbulkan berbagai risiko keamanan informasi, terutama dalam pengelolaan dokumen arsip yang bersifat penting.

Untuk mengatasi risiko-risiko tersebut, sistem yang diusulkan dirancang dengan mengacu pada prinsip CIA Triad dalam keamanan informasi, yaitu:

1. *Confidentiality* (Kerahasiaan)
   Dijamin melalui penerapan enkripsi pada dokumen sebelum disimpan ke sistem, sehingga hanya pihak yang memiliki hak akses yang dapat membaca isi berkas.
2. *Integrity* (Integritas)
   Dijamin melalui penggunaan *hashing* (misalnya SHA-256) yang disimpan pada blockchain, sehingga setiap perubahan terhadap dokumen dapat terdeteksi secara langsung.
3. *Availability* (Ketersediaan)
   Dijamin melalui penyimpanan terdistribusi menggunakan IPFS, yang memungkinkan dokumen tetap dapat diakses meskipun terjadi kegagalan pada satu node atau server tertentu [9].

3. Model Sistem

Sistem berbagi berkas yang diusulkan didefinisikan sebagai suatu sistem formal sebagaimana ditunjukkan pada Persamaan (1):

$$S = (U, D, B, I, C) \quad (1)$$

Di mana U: himpunan pengguna; D: himpunan dokumen; B: blockchain sebagai distributed ledger; I: sistem penyimpanan IPFS; C: *smart contract*. Model ini merepresentasikan interaksi antara pengguna, dokumen, dan komponen sistem dalam lingkungan terdistribusi

4. Model Integritas Dokumen

Setiap dokumen dihitung nilai hash-nya menggunakan fungsi kriptografi sebagaimana ditunjukkan pada Persamaan (2):

$$H(d) = \text{SHA-256}(d) \quad (2)$$

Nilai hash ini disimpan pada blockchain sebagai representasi unik dokumen untuk menjamin integritas data [1]. Metadata yang dicatat pada blockchain didefinisikan sebagai Persamaan (3):

 ISSN:2302-8149
e-ISSN:2540-9719

$$T = (CID, H(d), u, t) \quad (3)$$

di mana CID: content identifier dari IPFS; H(d): hash dokumen; u: pengguna yang melakukan aksi; t: timestamp. Pendekatan ini memungkinkan deteksi manipulasi dokumen melalui perbandingan hash antara dokumen yang diunduh dan hash yang tersimpan pada blockchain [7].

5. Model Kontrol Akses (Access Control Model)

Kontrol akses pada sistem diimplementasikan menggunakan *role-based access control* (RBAC) pada *backend* aplikasi. Hak akses didefinisikan sebagaimana Persamaan (4):

$$ACL(u, d) = \{read, write, revoke\} \quad (4)$$

Di mana u adalah pengguna dan d adalah dokumen.

Smart contract digunakan untuk mencatat aktivitas *grant access* dan *revocation* secara *immutable*, sehingga perubahan hak akses dapat diaudit. Keputusan akses operasional tetap dilakukan oleh *backend* berdasarkan ACL dan status *revocation* pada database lokal, sedangkan blockchain berperan sebagai lapisan audit trail. Secara lebih rinci, *smart contract* yang dikembangkan memuat lima fungsi utama sebagaimana ditunjukkan pada Tabel 3.

Tabel 3 Fungsi dan event utama *smart contract*

Fungsi/Event	Peran dalam Sistem	Data yang Dicatat/Dibaca	Ancaman yang Dimitigasi
uploadDocument(_cid, _hash)	Mencatat metadata dokumen yang telah disimpan di IPFS ke blockchain.	CID IPFS, hash SHA-256, pemilik dokumen, dan waktu transaksi.	Manipulasi dokumen dan perubahan metadata tanpa jejak audit.
grantAccess(_docId, _grantedUser)	Mencatat pemberian hak akses pengguna terhadap dokumen tertentu. Validasi akses operasional tetap dilakukan oleh *backend*.	ID dokumen, alamat pengguna yang diberi akses, eksekutor, dan waktu transaksi.	Akses tidak sah dan perubahan hak akses yang tidak terdokumentasi.
revokeAccess(_docId, _revokedUser)	Mencatat pencabutan hak akses pengguna secara permanen pada blockchain.	ID dokumen, alamat pengguna yang dicabut aksesnya, eksekutor, dan waktu transaksi.	Tidak adanya mekanisme *revocation* yang dapat diaudit.
getDocument(_id)	Membaca metadata dokumen dari blockchain untuk mendukung proses verifikasi integritas.	CID, hash dokumen, dan pemilik dokumen.	Kesulitan memverifikasi keaslian dokumen.
DocumentUploaded	Event yang dipancarkan saat metadata dokumen berhasil dicatat pada blockchain.	ID dokumen, CID, hash, pemilik, dan timestamp blok.	Audit log terpusat yang dapat dimodifikasi.
AccessGranted	Event yang dipancarkan saat pemberian akses berhasil dicatat.	ID dokumen, pengguna penerima akses, eksekutor, dan timestamp blok.	Perubahan akses yang tidak terlacak.
AccessRevoked	Event yang dipancarkan saat pencabutan akses berhasil dicatat.	ID dokumen, pengguna yang dicabut aksesnya, eksekutor, dan timestamp blok.	*Revocation* yang tidak terdokumentasi dan sulit diaudit.

 ISSN:2302-8149
e-ISSN:2540-9719

Berdasarkan Tabel 3, *smart contract* digunakan untuk mencatat metadata dokumen, mendukung pembacaan metadata on-chain, dan merekam perubahan hak akses secara permanen. Fungsi *uploadDocument* mencatat CID dan hash dokumen, sedangkan *getDocument* mengambil metadata tersebut untuk mendukung proses verifikasi integritas. Fungsi *grantAccess* dan *revokeAccess* mencatat pemberian serta pencabutan hak akses, sementara event *DocumentUploaded*, *AccessGranted*, dan *AccessRevoked* menjadi dasar audit trail on-chain yang bersifat *immutable*. Dengan pendekatan ini, riwayat aktivitas penting dapat ditelusuri secara lebih akuntabel, meskipun keputusan akses operasional tetap dijalankan oleh *backend* berdasarkan ACL dan status *revocation* pada database lokal [5].

6. Security Model

Kerahasiaan dokumen dijamin melalui enkripsi simetris menggunakan algoritma AES-256 sebelum dokumen diunggah ke IPFS. Pemilihan AES-256 didasarkan pada efisiensi komputasi dan tingkat keamanan yang memadai untuk kebutuhan instansi pemerintahan dibandingkan pendekatan CP-ABE yang memiliki overhead lebih tinggi. Kunci enkripsi dikelola secara terpusat oleh *backend* dan hanya didistribusikan kepada pengguna yang memiliki hak akses valid berdasarkan ACL(u, d). Dengan demikian, meskipun berkas dapat diakses secara fisik dari node IPFS, isi dokumen tetap tidak dapat dibaca tanpa kunci yang sah. Perlu dicatat bahwa pada penelitian ini, manajemen kunci masih dikelola oleh *backend* sebagai trusted component untuk menjaga kesederhanaan implementasi prototipe. Pendekatan ini cukup untuk validasi konsep, tetapi belum sepenuhnya menghilangkan ketergantungan terhadap komponen terpusat. Oleh karena itu, pengembangan selanjutnya perlu mempertimbangkan manajemen kunci yang lebih terdistribusi, misalnya *envelope encryption*, *attribute-based encryption*, atau *threshold secret sharing*.

Pada proses unduh, pengguna terlebih dahulu melakukan autentikasi melalui aplikasi. *Backend* kemudian memeriksa hak akses pengguna berdasarkan ACL dan status *revocation* yang tersimpan pada database aplikasi serta mencocokkannya dengan catatan transaksi grant/revoke pada blockchain. Jika pengguna memiliki hak akses valid, *backend* mengambil berkas terenkripsi dari IPFS berdasarkan CID, mengambil kunci dekripsi AES-256 yang dikelola pada sisi *backend*, lalu menyediakan berkas hasil dekripsi kepada pengguna melalui sesi aplikasi yang telah terautentikasi. Pada prototipe ini, kunci tidak dikelola secara terdistribusi dan belum menggunakan mekanisme envelope encryption atau threshold *secret sharing*. Oleh karena itu, *backend* masih berperan sebagai trusted component dalam proses manajemen dan distribusi kunci.

7. Mekanisme *Revocation*

Mekanisme pencabutan akses (*revocation*) merupakan salah satu komponen utama sistem yang diusulkan. Ketika akses pengguna dicabut, kondisi hak akses dinyatakan sebagaimana Persamaan (5):

$$ACL(u, d) = \emptyset \quad (5)$$

Perubahan ini dicatat sebagai transaksi pada blockchain sebagaimana Persamaan (6):

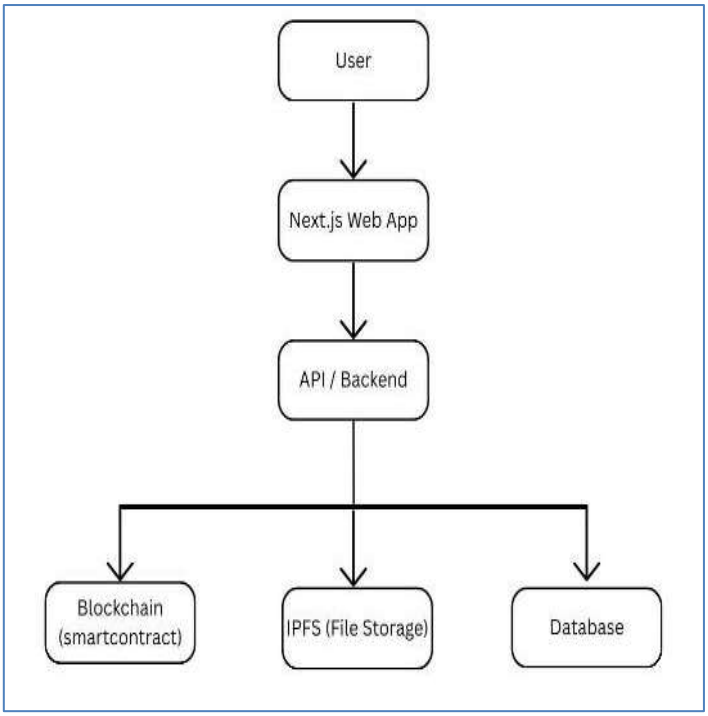
$$T\_revocation = (u, d, t) \quad (6)$$

Dengan demikian, setiap perubahan hak akses dapat ditelusuri secara transparan melalui *audit trail* yang bersifat *immutable*. Mekanisme ini mengatasi keterbatasan sistem sebelumnya yang tidak menyediakan pencabutan akses secara dinamis dan teraudit [7].

8. Arsitektur Sistem

Untuk memperjelas alur kerja serta keterkaitan antar komponen dalam sistem yang diusulkan, digunakan diagram arsitektur sebagai representasi visual. Diagram ini menggambarkan proses interaksi pengguna, pengolahan data, hingga mekanisme penyimpanan dan pencatatan informasi secara terintegrasi. Ilustrasi arsitektur sistem tersebut ditunjukkan pada Gambar 1.

 ISSN:2302-8149
e-ISSN:2540-9719



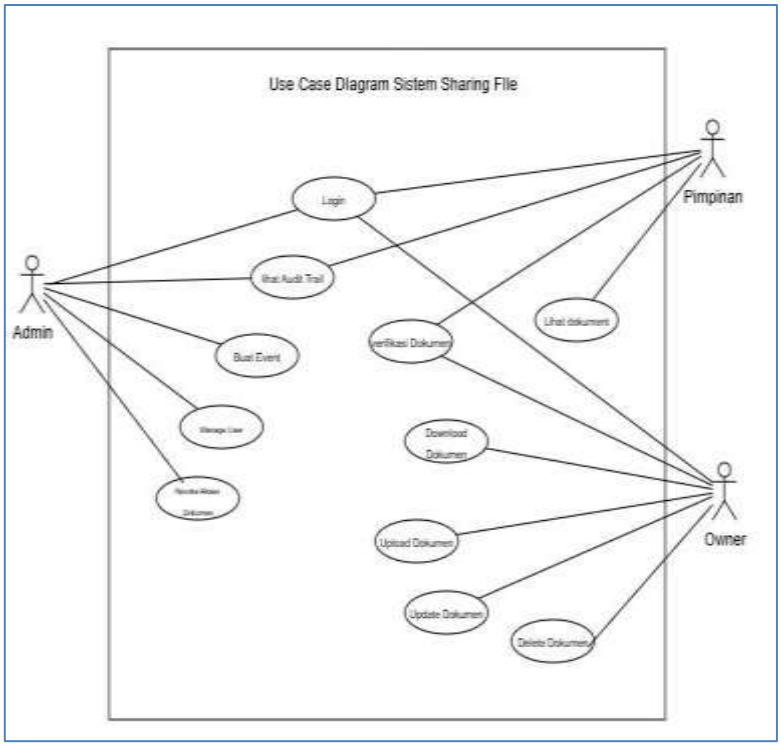
Gambar 1 Arsitektur sistem

Gambar 1 menunjukkan bahwa arsitektur sistem dirancang dengan pendekatan berlapis (layered architecture). Setiap lapisan memiliki fungsi spesifik dan saling terintegrasi satu sama lain. Pada lapisan antarmuka (*frontend layer*), sistem dibangun menggunakan *Next.js* yang berfungsi untuk menangani interaksi pengguna, proses autentikasi, serta aktivitas unggah dan unduh dokumen. Selanjutnya, *backend layer* bertanggung jawab dalam mengelola logika aplikasi serta menjembatani komunikasi antara *frontend* dengan layanan blockchain dan IPFS. Pada *blockchain layer*, sistem memanfaatkan *Hyperledger Besu* sebagai *permissioned blockchain* dengan mekanisme konsensus *IBFT 2.0* untuk mencatat metadata dokumen serta *audit trail* secara *immutable*. Sementara itu, *storage layer* menggunakan IPFS sebagai media penyimpanan terdistribusi untuk menyimpan dokumen secara efisien. Selain itu, terdapat *smart contract* layer yang berfungsi untuk mencatat metadata, aktivitas penting, serta perubahan hak akses seperti *grant access* dan *revocation* secara *immutable*. Kontrol akses berbasis peran diterapkan pada *backend* aplikasi, sedangkan blockchain berperan sebagai lapisan pencatatan audit yang dapat ditelusuri. Pendekatan hybrid yang digunakan dalam arsitektur ini mengombinasikan keunggulan blockchain dalam menjaga integritas data dengan efisiensi penyimpanan yang ditawarkan oleh IPFS.

9. Perancangan dan Pengembangan Sistem

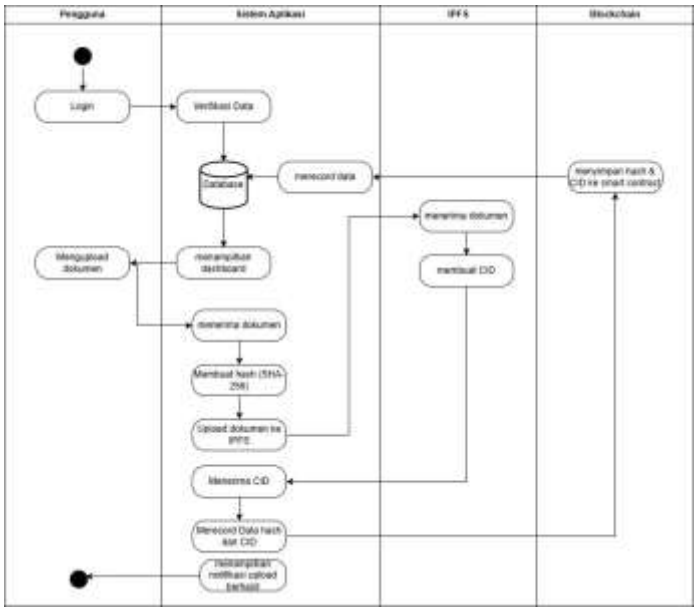
Pada tahap perancangan sistem, dilakukan pemodelan untuk menggambarkan kebutuhan fungsional serta interaksi antara pengguna dengan sistem. Pemodelan ini bertujuan untuk memberikan representasi yang terstruktur mengenai bagaimana sistem akan dibangun dan dioperasikan. Beberapa diagram yang digunakan dalam tahap ini meliputi *use case diagram*, *class diagram*, dan diagram aktivitas. Adapun *use case diagram* digunakan sebagai langkah awal untuk menggambarkan interaksi antara aktor dan sistem, seperti ditunjukkan pada Gambar 2.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 2 *Use case diagram*

Gambar 2 menunjukkan *use case diagram* dari sistem berbagi berkas yang diusulkan. Diagram ini menggambarkan interaksi antara tiga aktor utama, yaitu Admin, Owner, dan Pimpinan, dengan sistem. Admin memiliki hak akses untuk mengelola pengguna, membuat event, melihat *audit trail*, serta mengatur hak akses dokumen. Owner berperan dalam pengelolaan dokumen, seperti mengunggah, memperbarui, menghapus, dan mengunduh dokumen. Selain itu, Owner dan Pimpinan dapat melakukan verifikasi dokumen serta melihat dokumen yang tersedia. Seluruh aktor diwajibkan melakukan proses login sebelum mengakses sistem. Dengan adanya *use case diagram* ini, batasan sistem serta hak akses masing-masing aktor dapat didefinisikan dengan lebih jelas dan terstruktur.



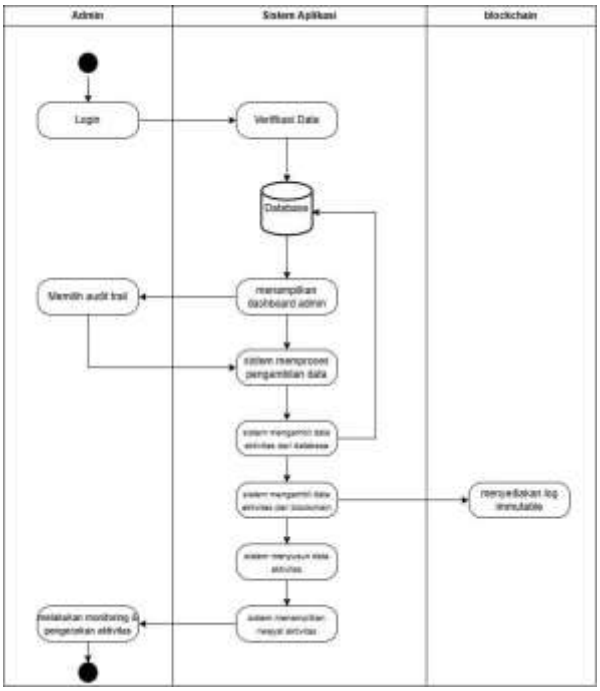
Gambar 3 Diagram aktivitas unggah dokumen

Gambar 3 menunjukkan diagram aktivitas proses unggah dokumen pada sistem berbagi berkas yang melibatkan Pengguna, Sistem Aplikasi, IPFS, dan Blockchain. Proses dimulai dari login

 ISSN:2302-8149
e-ISSN:2540-9719

pengguna yang divalidasi oleh sistem. Selanjutnya, pengguna mengunggah dokumen dan sistem secara otomatis membuat hash SHA-256 untuk menjaga integritas. Selanjutnya, dokumen disimpan ke IPFS dan menghasilkan *Content Identifier* (CID), yang bersama nilai hash dicatat sebagai metadata ke dalam database serta disimpan ke blockchain melalui *smart contract* untuk menjamin sifat *immutable*. Proses diakhiri dengan notifikasi keberhasilan kepada pengguna. Alur ini memastikan keamanan data melalui integritas dan ketersediaan dengan memanfaatkan *hashing*, blockchain, dan penyimpanan terdistribusi IPFS.

Proses pengunduhan dokumen diawali dengan autentikasi pengguna, dilanjutkan pengecekan hak akses berbasis RBAC oleh *backend*. Jika akses valid, sistem mengambil CID dari database, mengambil berkas terenkripsi dari IPFS, mendekripsinya menggunakan kunci AES-256, lalu menyediakan berkas kepada pengguna. Proses verifikasi integritas dokumen dilakukan dengan mengambil berkas dari IPFS, menghasilkan ulang nilai hash SHA-256, kemudian membandingkannya dengan hash yang tersimpan pada blockchain melalui *smart contract*; dokumen dinyatakan valid apabila kedua nilai hash sesuai, dan terindikasi telah dimanipulasi apabila terdapat ketidaksesuaian.



Gambar 4 Diagram aktivitas audit trail

Gambar 4 menunjukkan diagram aktivitas proses audit trail pada sistem berbagi berkas yang melibatkan Admin, Sistem Aplikasi, dan Blockchain. Proses dimulai ketika admin melakukan login, kemudian sistem melakukan verifikasi data untuk memastikan keabsahan akses dan menampilkan *dashboard* admin. Selanjutnya, admin memilih menu audit trail, sehingga sistem memproses permintaan dengan mengambil data aktivitas dari database yang mencakup berbagai aktivitas pengguna seperti *upload*, *download*, *update*, *delete*, dan *verifikasi*. Selain itu, sistem juga mengambil data log dari blockchain melalui *smart contract* yang bersifat *immutable*. Data dari kedua sumber tersebut kemudian disusun dan ditampilkan sebagai riwayat aktivitas kepada admin. Melalui proses ini, admin dapat melakukan monitoring dan pengecekan terhadap seluruh aktivitas yang terjadi dalam sistem. Mekanisme ini memastikan transparansi dan akuntabilitas dengan memanfaatkan pencatatan log yang tidak dapat diubah pada blockchain.

10. Lingkungan Pengujian (Experimental Setup)

Pengujian sistem dilakukan pada lingkungan lokal (localhost) menggunakan Docker Desktop pada sistem operasi Windows 11. Seluruh komponen sistem dijalankan dalam container Docker yang terisolasi untuk memastikan konsistensi lingkungan pengujian. Spesifikasi perangkat dan konfigurasi sistem pengujian disajikan pada Tabel 4 berikut.

 ISSN:2302-8149
e-ISSN:2540-9719

Tabel 4 Spesifikasi lingkungan pengujian

No	Komponen	Spesifikasi / Konfigurasi
1	Sistem Operasi Host	Windows 11, RAM 8 GB
2	Virtualisasi	Docker Desktop (container-based deployment)
3	*Permissioned Blockchain*	Hyperledger Besu v24.12.0; konsensus IBFT 2.0; 1 node (single node, localhost)
4	Penyimpanan Terdistribusi	IPFS (local node, dijalankan via Docker)
5	Database	MySQL (dijalankan via Docker container)
6	*Frontend*	Next.js (React framework)
7	*Smart Contract*	Solidity; di-deploy pada jaringan Hyperledger Besu
8	Lingkungan Jaringan	Localhost (jaringan lokal, tanpa koneksi internet eksternal)

Pengujian dilakukan pada lingkungan lokal berbasis Docker dengan satu node Hyperledger Besu dan satu node IPFS. Oleh karena itu, hasil performa digunakan untuk mengevaluasi kelayakan prototipe, bukan untuk merepresentasikan performa jaringan *permissioned blockchain* multi-node pada lingkungan produksi.

11. Skenario Pengujian dan Evaluasi

Evaluasi sistem dilakukan untuk mengukur efektivitas solusi dalam memenuhi tujuan penelitian. Pengujian dilakukan melalui beberapa skenario:

1. Pengujian Fungsional
   Memastikan seluruh fitur sistem berjalan sesuai dengan kebutuhan pengguna.
2. Pengujian Integritas Dokumen (*Tamper Detection*)
   Dokumen dimodifikasi secara sengaja untuk menguji kemampuan sistem dalam mendeteksi perubahan melalui perbandingan hash
3. Pengujian Performa
   Mengukur: *Latency* transaksi blockchain & Waktu unggah dokumen ke IPFS
4. Pengujian *Revocation*
   Mengukur waktu yang dibutuhkan untuk mencabut akses serta memastikan bahwa pengguna yang telah dicabut aksesnya tidak dapat lagi mengakses dokumen

Metode ini sesuai dengan evaluasi sistem berbasis blockchain pada penelitian sebelumnya [5], [7].

Evaluasi sistem diukur menggunakan metrik berikut:

1. *Tamper detection rate*
   Persentase manipulasi dokumen yang berhasil dideteksi, dihitung menggunakan rumus (7):

$$TDR = TP / (TP + FN) \times 100\% \quad (7)$$

   Dengan TP adalah jumlah skenario manipulasi yang berhasil terdeteksi dan FN adalah skenario manipulasi yang tidak terdeteksi. Berdasarkan pengujian terhadap empat skenario manipulasi, diperoleh:

$$TDR = 4 / (4 + 0) \times 100\% = 100\%$$

2. *Transaction Latency*
   Waktu rata-rata eksekusi transaksi blockchain (milidetik (ms)), dihitung menggunakan rumus (8):

$$L\_avg = (1/n) \times \Sigma Li, \text{ untuk } i = 1 \text{ sampai } n \quad (8)$$

   Dengan n adalah jumlah percobaan dan *Li* adalah *latency* pada percobaan ke-i. Berdasarkan pengujian dengan n=3 percobaan per variasi ukuran file, diperoleh rata-rata *latency* transaksi sebesar 4463 ms (kisaran 4237–4775 ms)
3. Waktu unggah IPFS
   Waktu unggah dokumen ke IPFS berdasarkan variasi ukuran file, dihitung menggunakan rumus yang sama dengan transaction *latency*. Berdasarkan hasil pengujian, diperoleh rata-rata

 ISSN:2302-8149
e-ISSN:2540-9719

waktu unggah sebesar 248 ms dengan kisaran 79–527 ms. Waktu unggah menunjukkan kecenderungan meningkat seiring bertambahnya ukuran file.

4. *Revocation time*
   Waktu eksekusi pencabutan hak akses (ms), dihitung menggunakan rumus (9):

$$R\text{-}rata = (1/n) \times \Sigma Ri, \text{ dengan } i = 1 \text{ sampai } n \quad (9)$$

   dengan *Ri* adalah waktu eksekusi *revocation* pada percobaan ke-i. Berdasarkan tiga transaksi *revocation* yang diuji, rata-rata waktu eksekusi *revocation* adalah 4466 ms dengan kisaran 4318–4540 ms. Pada skenario *revocation* yang diikuti grant ulang, waktu tambahan untuk transaksi grant adalah 4236 ms, sehingga total waktu skenario tersebut menjadi 8776 ms.

5. *Functional correctness*
   Kesesuaian output sistem dengan *expected result* pada black-box testing, dihitung menggunakan rumus (10):

$$FC = TC\_pass / TC\_total \times 100\% \quad (10)$$

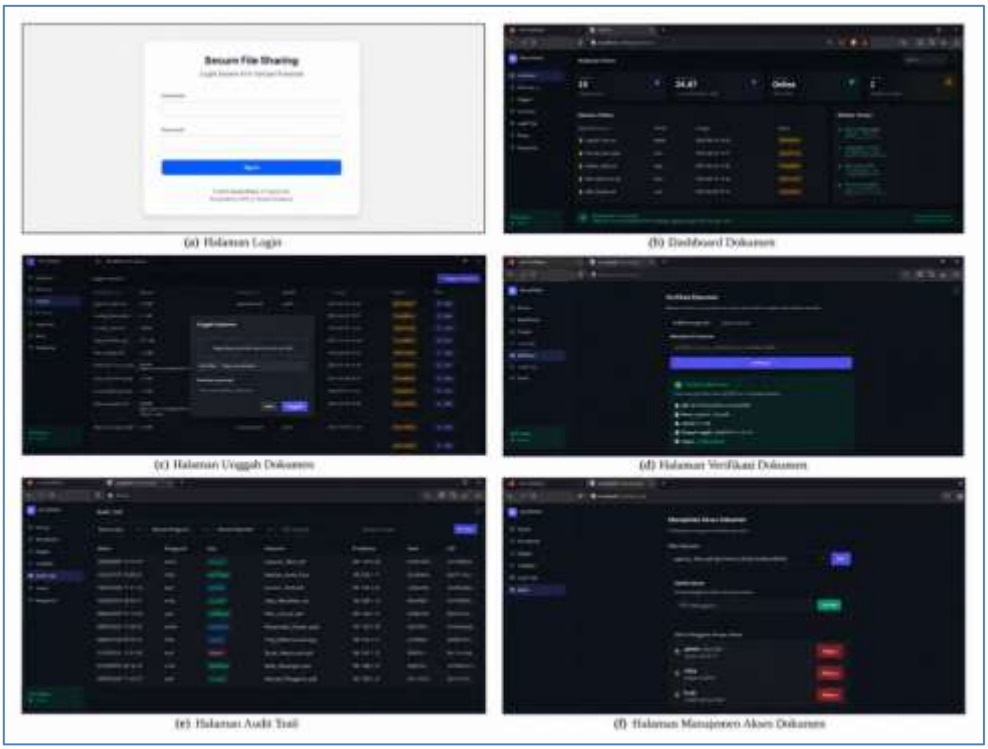
   Dengan TC_pass adalah jumlah skenario uji yang lulus dan TC_total adalah total skenario uji. Berdasarkan pengujian terhadap 10 skenario uji fungsional, dipe*role*h:

$$FC = 10 / 10 \times 100\% = 100\%$$

Hasil dan Pembahasan

1. Implementasi Prototipe

Implementasi sistem dilakukan berdasarkan hasil perancangan dengan mengintegrasikan *permissioned blockchain* Hyperledger Besu dan IPFS dalam satu arsitektur terpadu. Sistem mendukung fitur autentikasi pengguna, unggah dan unduh dokumen terenkripsi, verifikasi integritas, manajemen akses, serta *audit trail*. Antarmuka sistem dibangun menggunakan Next.js dengan pembagian akses berdasarkan *role*: Admin, Owner, dan Pimpinan. Ringkasan antarmuka utama sistem disajikan pada Gambar 5.



Gambar 5 Ringkasan antarmuka utama sistem

 ISSN:2302-8149
e-ISSN:2540-9719

Gambar 5 menunjukkan ringkasan antarmuka utama sistem berbagi berkas. Sub gambar (a)–(f) masing-masing menampilkan halaman login, *dashboard* dokumen, unggah dokumen, verifikasi dokumen, audit trail, dan manajemen akses dokumen. Keenam tampilan tersebut merepresentasikan alur utama sistem, mulai dari autentikasi pengguna, pengelolaan dan verifikasi dokumen, hingga pemantauan aktivitas serta pengendalian hak akses.

2. Pengujian Sistem

Pengujian sistem dilakukan untuk mengevaluasi kesesuaian fungsi utama, validitas dokumen, dan kemampuan sistem dalam mendeteksi manipulasi. Pengujian dikelompokkan menjadi pengujian fungsional, validasi dokumen, dan deteksi manipulasi dokumen. Rekapitulasi hasil pengujian disajikan pada Tabel 5, sedangkan rincian skenario validasi dan deteksi manipulasi untuk pengujian integritas dokumen dijelaskan lebih lanjut pada Tabel 6 dan Tabel 7.

Tabel 5 Rekapitulasi hasil pengujian sistem

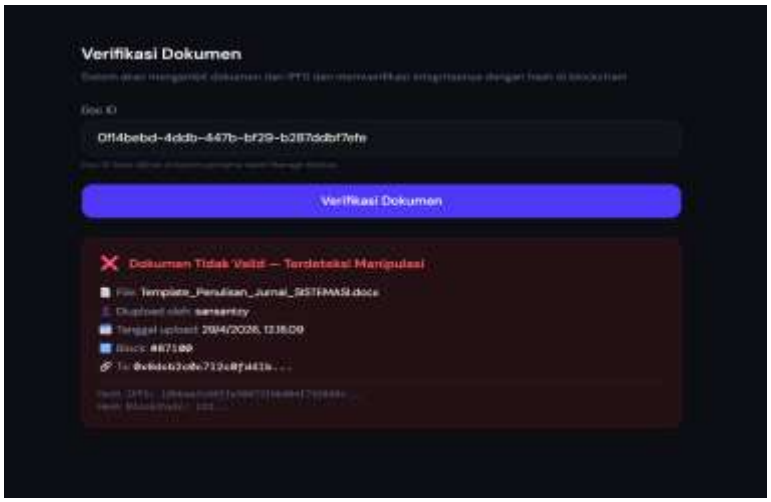
No	Pengujian	Skenario	Fokus Pengujian	Hasil	Status
1	Fungsional	10	Login, unggah, unduh, verifikasi, audit trail, grant/revoke access, dan hapus dokumen	Semua skenario sesuai *expected result*; FC = 100%.	Berhasil
2	Validasi dokumen	2	Dokumen asli dan DocID tidak terdaftar	Dokumen asli valid; DocID tidak terdaftar dikenali sebagai not found.	Berhasil
3	Tamper detection	4	Manipulasi isi dokumen dan hash database lokal	Seluruh manipulasi terdeteksi; TDR = 100%.	Berhasil
4	Total	16	Fungsi utama dan integritas dokumen	Semua skenario berhasil.	Berhasil

Berdasarkan Tabel 5, seluruh kelompok pengujian menunjukkan hasil sesuai dengan keluaran yang diharapkan. Sepuluh skenario pengujian fungsional berhasil dijalankan tanpa kegagalan, sehingga nilai functional correctness mencapai 100%. Selain itu, sistem mampu membedakan dokumen valid dan DocID yang tidak terdaftar, serta berhasil mendeteksi seluruh skenario manipulasi dokumen melalui perbandingan hash. Hasil ini menunjukkan bahwa prototipe telah memenuhi kebutuhan dasar sistem berbagi berkas, khususnya pada aspek fungsi utama dan integritas dokumen.

3. Pengujian Integritas Dokumen (*Tamper Detection*)

Pengujian integritas dokumen dibagi menjadi dua bagian: pengujian validasi dokumen (Tabel 6) dan pengujian tamper detection (Tabel 7). Pengujian validasi menunjukkan bahwa sistem mampu membedakan dokumen asli yang valid dari dokumen dengan DocID tidak terdaftar, dengan respons masing-masing berupa status *valid* dan *not found*. Bukti visual hasil deteksi manipulasi ditunjukkan pada Gambar 6.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 6 Hasil verifikasi setelah nilai hash pada database diubah secara manual

Gambar 6 menunjukkan hasil verifikasi setelah nilai hash pada database diubah secara manual untuk mensimulasikan manipulasi dokumen, di mana sistem berhasil mendeteksi ketidaksesuaian hash dan menampilkan status dokumen termanipulasi. Hasil pengujian ditunjukkan pada Tabel 6 dan Tabel 7.

Tabel 6 Pengujian validasi dokumen

No	Skenario	Status
1	Dokumen asli tanpa modifikasi	✓ Valid
2	DocID tidak terdaftar di sistem	✓ Not Found

Tabel 7 Pengujian tamper detection

No	Skenario Manipulasi	Status Perbandingan Hash	Status
1	Penambahan satu karakter pada dokumen	Hash tidak cocok	Terdeteksi
2	Penghapusan satu karakter pada dokumen	Hash tidak cocok	Terdeteksi
3	Penggantian seluruh isi dokumen	Hash tidak cocok	Terdeteksi
4	Perubahan nilai hash pada database lokal	Hash on-chain tidak berubah / tidak cocok	Terdeteksi

Hasil pengujian menunjukkan bahwa sistem berhasil mendeteksi seluruh skenario manipulasi yang diuji, mencakup penggantian isi dokumen maupun modifikasi nilai hash pada database. Hal ini dimungkinkan karena sifat fungsi SHA-256 yang menghasilkan nilai hash unik untuk setiap dokumen, sehingga setiap perubahan pada konten dokumen akan menghasilkan hash yang berbeda (*avalanche effect*) sebagai karakteristik fungsi hash kriptografi [15]. Temuan ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa pendekatan *content addressing* berbasis hash efektif dalam menjamin integritas dokumen pada sistem berbasis IPFS dan blockchain [7].

4. Pengujian Performa Blockchain dan IPFS

Pengujian performa dilakukan untuk mengukur dua metrik utama: *latency* transaksi blockchain dan waktu unggah dokumen ke IPFS. *Latency* transaksi didefinisikan sebagai total waktu sejak permintaan transaksi dikirim oleh *backend* hingga transaksi dikonfirmasi dan difinalisasi pada jaringan Hyperledger Besu. Nilai rata-rata *latency* dihitung menggunakan Persamaan (8) yang telah didefinisikan sebelumnya, dengan n = 3 percobaan per variasi ukuran file dan Li adalah *latency* pada

 ISSN:2302-8149
e-ISSN:2540-9719

percobaan ke-i. Pengujian dilakukan dengan variasi ukuran file 100 KB hingga 5 MB. Hasil pengujian ditunjukkan pada Tabel 8 dan Tabel 9.

Tabel 8 *Latency* transaksi blockchain pada proses unggah dokumen

No	Ukuran File	Rata-rata *Latency* (ms)	Min (ms)	Max (ms)
1	100KB	4343	4282	4443
2	500KB	4497	4318	4845
3	1MB	4237	4232	4245
4	5MB	4775	4448	5113

Pada Tabel 8 terlihat bahwa file berukuran 1 MB memiliki rata-rata *Latency* transaksi blockchain sebesar 4237 ms, lebih rendah dibandingkan file 100 KB dan 500 KB. Perbedaan ini tidak menunjukkan bahwa file yang lebih besar selalu menghasilkan *latency* blockchain yang lebih rendah, melainkan merupakan variasi pengukuran pada lingkungan lokal. Hal ini terjadi karena data yang dicatat ke blockchain hanya berupa metadata, CID, dan hash dokumen, bukan isi dokumen secara langsung. Dengan demikian, ukuran file tidak menjadi faktor dominan terhadap *latency* transaksi blockchain. Variasi *latency* lebih mungkin dipengaruhi oleh waktu eksekusi transaksi, penjadwalan container Docker, proses finalisasi blok, dan kondisi runtime pada saat pengujian. Selain itu, jumlah percobaan yang terbatas, yaitu tiga kali per skenario, belum cukup untuk menyimpulkan pola performa secara statistik.

Tabel 9 Waktu unggah dokumen ke IPFS

No	Ukuran File	Rata-rata Waktu Unggah (ms)	Min (ms)	Max (ms)
1	100KB	79	52	107
2	500KB	114	101	206
3	1MB	270	265	275
4	5MB	527	438	590

Hasil pengujian menunjukkan bahwa rata-rata *latency* transaksi blockchain berada pada rentang 4237–4775 ms, atau sekitar 4,24–4,78 detik. Meskipun nilai ini lebih tinggi dibandingkan operasi baca-tulis pada sistem terpusat, hal ini merupakan trade-off yang dapat diterima mengingat jaminan *immutability* yang diberikan. *Latency* yang relatif stabil di semua variasi ukuran file menunjukkan bahwa pendekatan *hybrid on-chain/off-chain* yang digunakan relatif tidak dipengaruhi ukuran dokumen, karena yang disimpan secara on-chain hanyalah metadata berupa CID dan nilai hash, bukan dokumen itu sendiri. Sementara itu, waktu unggah ke IPFS menunjukkan kecenderungan meningkat seiring bertambahnya ukuran file — dari 79 ms untuk file 100 KB hingga 527 ms untuk file 5 MB — yang merupakan perilaku wajar dari sistem penyimpanan berbasis jaringan *peer-to-peer*.

5. Pengujian Mekanisme *Revocation*

Pengujian *revocation* dilakukan dengan skenario pencabutan hak akses unduh (*download*/*read access*) pengguna melalui *smart contract*. Setelah proses pencabutan dilakukan, sistem memverifikasi bahwa pengguna yang bersangkutan tidak lagi dapat mengunduh dokumen, meskipun metadata dokumen masih dapat ditampilkan sesuai hak akses yang tersisa. Hasil pengujian ditunjukkan pada Tabel 10.

Tabel 10 Hasil pengujian *revocation*

No	Skenario	Waktu Eksekusi (ms)	Akses Unduh Setelah *Revocation*	Status Transaksi Blockchain
1	*Revocation* user *role* Owner	4540	Ditolak ✓	Tercatat *immutable* ✓
2	*Revocation* user *role* Pimpinan	4318	Ditolak ✓	Tercatat *immutable* ✓
3	*Revocation* lalu grant ulang	4540 + 4236	Diizinkan ✓	Dua transaksi tercatat ✓
4	Akses dokumen oleh user yang sudah direvoke	-	Ditolak ✓	-

 ISSN:2302-8149
e-ISSN:2540-9719

Pengujian mekanisme *revocation* dilakukan dengan mencabut hak akses unduh pengguna melalui *smart contract*, kemudian diverifikasi bahwa pengguna yang bersangkutan tidak lagi dapat mengunduh dokumen. Berdasarkan tiga transaksi *revocation* yang diuji, rata-rata waktu eksekusi adalah 4466 ms, dengan kisaran 4318–4540 ms. Meskipun nilai ini lebih tinggi dibandingkan sistem terpusat, nilai *latency* tersebut dipengaruhi oleh *overhead* transaksi blockchain dan proses finalisasi blok pada lingkungan Hyperledger Besu lokal. Karena pengujian dilakukan pada konfigurasi *single-node localhost*, hasil ini belum merepresentasikan *overhead* konsensus dan komunikasi antarvalidator pada jaringan *permissioned* multi-node. Setiap transaksi pencabutan akses tercatat secara *immutable* pada blockchain sehingga seluruh riwayat perubahan hak akses dapat ditelusuri melalui fitur *audit trail*. Pada skenario *revocation* diikuti *grant* ulang, sistem berhasil mencatat kedua transaksi secara terpisah dengan total waktu 8776 ms, yang membuktikan bahwa mekanisme ini bersifat dinamis sekaligus teraudit. Pada skenario keempat, pengguna yang telah dicabut aksesnya secara konsisten mendapatkan respons 403 Akses Ditolak dari sistem, membuktikan bahwa pengecekan status *revocation* pada database berjalan efektif sebelum sistem mengizinkan proses unduh. Hasil ini menunjukkan bahwa prototipe telah menyediakan mekanisme *revocation* yang teraudit melalui pencatatan transaksi pada blockchain. Namun, karena pengecekan akses operasional masih dilakukan melalui database lokal, mekanisme ini lebih tepat diposisikan sebagai pendekatan *hybrid on-chain/off-chain*, bukan sebagai *revocation* yang sepenuhnya divalidasi langsung oleh blockchain pada setiap permintaan akses [5]**,** [8]. Keterbatasan ini perlu diperbaiki pada pengembangan selanjutnya melalui sinkronisasi *state on-chain/off-chain* yang lebih ketat atau validasi langsung terhadap *smart contract* sebelum proses unduh diberikan.

6. Pembahasan Komparatif

Dibandingkan sistem lama berbasis Google Drive, sistem yang diusulkan menyediakan mekanisme verifikasi integritas yang lebih eksplisit melalui pencatatan hash SHA-256 pada blockchain serta audit trail *immutable* yang tidak dapat dimanipulasi administrator [1], [2], dan mekanisme *revocation* yang tercatat secara permanen pada blockchain. Dibandingkan penelitian sebelumnya, penggunaan *permissioned blockchain* menjawab keterbatasan blockchain publik dalam hal biaya transaksi dan privasi data [4], [6], sementara integrasi RBAC yang dirancang spesifik untuk struktur organisasi pemerintahan melengkapi kekurangan implementasi kontrol akses pada penelitian sebelumnya [7], [8]. *Latency* transaksi rata-rata 4237–4775 ms merupakan trade-off yang dapat diterima untuk kebutuhan operasional prototipe. Pada konfigurasi pengujian lokal, *latency* tersebut menunjukkan bahwa pencatatan metadata on-chain masih dapat dijalankan secara konsisten. Namun, performa pada jaringan multi-node memerlukan pengujian lanjutan karena overhead konsensus dan komunikasi antar validator dapat berbeda [3]. Secara keseluruhan, penelitian ini menghadirkan kontribusi yang belum ditemukan secara eksplisit pada penelitian sebelumnya [5], [7], yaitu integrasi lengkap antara enkripsi dokumen, RBAC, mekanisme *revocation*, dan *audit trail immutable* dalam satu sistem yang kohesif untuk kebutuhan pengelolaan arsip instansi pemerintahan [1], [5].

Dengan demikian, kontribusi penelitian ini terletak pada integrasi arsitektur hybrid yang menggabungkan IPFS, *permissioned blockchain*, enkripsi, RBAC, *revocation*, dan audit trail dalam satu prototipe. Namun, karena manajemen kunci dan pengecekan akses operasional masih bergantung pada *backend* dan database lokal, sistem ini belum sepenuhnya menghilangkan ketergantungan terhadap komponen terpusat.

7. Keterbatasan Sistem

Penelitian ini memiliki beberapa keterbatasan yang perlu diakui. Pertama, pengujian masih terbatas pada skala jaringan satu node (dev mode) sehingga belum mencerminkan kondisi jaringan multi-node yang sesungguhnya. Kedua, manajemen kunci enkripsi masih terpusat pada *backend* sebagai *trusted component*, sehingga belum sepenuhnya menghilangkan ketergantungan terhadap komponen terpusat. Ketiga, sistem belum mencakup integrasi lintas instansi. Keempat, jumlah sampel pengujian performa (3 kali per skenario) masih terbatas untuk dapat menyimpulkan pola linier secara statistik. Kelima, mekanisme *revocation* pada prototipe ini masih menggunakan database lokal sebagai sumber pengecekan akses operasional, sedangkan blockchain berperan sebagai pencatat transaksi grant dan *revocation* yang bersifat *immutable*. Oleh karena itu, pengembangan selanjutnya

 ISSN:2302-8149
e-ISSN:2540-9719

perlu mengarah pada validasi akses yang lebih langsung terhadap *smart contract* atau mekanisme sinkronisasi state on-chain dan *off-chain* yang lebih kuat.

Kesimpulan

Penelitian ini berhasil merancang dan mengimplementasikan prototipe sistem berbagi berkas berbasis *permissioned blockchain* Hyperledger Besu dan IPFS untuk mendukung pengelolaan dokumen di DISPUSIPDA Jawa Barat dengan mengintegrasikan enkripsi AES-256, kontrol akses berbasis peran, mekanisme *revocation*, pencatatan metadata, dan audit trail *immutable* dalam satu arsitektur *hybrid on-chain/off-chain*. Hasil pengujian menunjukkan bahwa seluruh fitur fungsional berjalan sesuai kebutuhan, manipulasi dokumen berhasil dideteksi pada seluruh skenario uji melalui perbandingan hash SHA-256, *latency* transaksi blockchain berada pada kisaran 4237–4775 ms atau sekitar 4,24–4,78 detik, waktu unggah IPFS meningkat sesuai ukuran file, dan mekanisme *revocation* pada arsitektur hybrid mampu membatasi akses unduh pengguna yang hak aksesnya telah dicabut melalui pengecekan *backend* dan pencatatan perubahan hak akses pada blockchain. Temuan ini menjawab gap penelitian sebelumnya yang belum banyak mengintegrasikan IPFS, *permissioned blockchain*, RBAC, *revocation*, dan audit trail dalam konteks pengelolaan arsip institusional pemerintahan. Secara praktis, sistem ini dapat menjadi alternatif rancangan awal untuk meningkatkan integritas dokumen, akuntabilitas aktivitas pengguna, dan dukungan proses audit, meskipun pengujian masih terbatas pada lingkungan lokal satu node serta masih bergantung pada *backend* untuk manajemen kunci dan pengecekan akses operasional.

Referensi

[1] M. U. Noor, "Implementasi *Blockchain* di Dunia Kearsipan: Peluang, Tantangan, Solusi atau Masalah Baru?," *Khizanah al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, Vol. 8, No. 1, pp. 86–96, 2020, DOI: 10.24252/kah.v8i1a9.

[2] Baihaqsani, A. Kusyanti, and P. H. Trisnawan, "Implementasi Teknologi *Blockchain* dengan Sistem *Smart Contract* pada Klaim Asuransi," *Jurnal Teknologi Informasi dan Ilmu Komputer*, Vol. 11, No. 5, pp. 1105–1112, 2024, DOI: 10.25126/jtiik.2024118016.

[3] S. Athanere and R. Thakur, "*Blockchain based Hierarchical Semi-Decentralized Approach using IPFS for Secure and Efficient Data Sharing*," *J. King Saud Univ. - Comput. Inf. SCI.*, Vol. 34, No. 4, pp. 1523–1534, 2022, DOI: 10.1016/j.jksuci.2022.01.019.

[4] M. S. Kumar, S. Bhake, A. Ande, and Zaneta, "*Block Share: A Block Chain-based File Storage and Sharing System using IPFS*," in *Proc. International Conference on Computer Science and Communication Engineering (ICCSCE 2025)*, Atlantis Press, 2025, pp. 1360–1371, DOI: 10.2991/978-94-6463-858-5_113.

[5] T. Asmiyanto, N. Putrawan, Y. Widiarta, and H. Inamullah, "Armanesia *Blockchain System*: *Blockchain* and *IFPS-based Archive System Prototype*," *Khizanah al-Hikmah: Jurnal Ilmu Perpustakaan, Informasi, dan Kearsipan*, Vol. 10, No. 2, pp. 194–208, 2022, DOI: 10.24252/kah/v10i2a9.

[6] N. Permatasari and A. Novelin, "Penerapan Teknologi *Blockchain* dalam Pelayanan Publik: Meningkatkan Keamanan, Transparansi, dan Kepercayaan Masyarakat melalui *Website* Onlinepajak," *Jurnal Ilmiah Wahana Pendidikan*, Vol. 10, No. 14, pp. 764–773, 2024, DOI: 10.5281/zenodo.13739365.

[7] M. D. Muis, M. R. Fauzan, P. Sukarno, and A. A. Wardana, *"Access Control and File Distribution Management for Electronic Diploma and Transcript using Ethereum Smart Contract and InterPlanetary File System*," *Jurnal Sistem Informasi*, Vol. 17, No. 2, pp. 48–61, 2021, DOI: 10.21609/jsi.v17i2.1093.

[8] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, "*Blockchain-based, Decentralized Access Control for IPFS*," in *Proc. IEEE Int. Conf. Internet of Things (iThings)*, 2018, pp. 1499–1506, DOI: 10.1109/Cybermatics_2018.2018.00253.

[9] M. Naz, F. A. Al-zahrani, R. Khalid, N. Javaid, A. M. Qamar, M. K. Afzal, and M. Shafiq, "*A Secure Data Sharing Platform using Blockchain and Interplanetary File System*," *Sustainability*, Vol. 11, No. 24, pp. 1–24, 2019, DOI: 10.3390/su11247054.

[10] R. P. Santoso and R. R. J. Putra, "Blok IPFS *Blockchain* untuk Orisinalitas Ijazah Pendidikan

Tinggi," JUKTISI: Jurnal Komputer Teknologi Informasi Sistem Komputer, Vol. 4, No. 3, pp. 1993–1999, 2026, DOI: 10.62712/juktisi.v4i3.780.

[11] R. A. Suparlan, "Implementasi *Smart Contract Blockchain Ethereum* pada Aplikasi *E-voting*," Informatics and Digital Expert (INDEX), Vol. 7, No. 1, pp. 66–70, 2025, DOI: 10.36423/index.v7i1.2177.

[12] A. S. Putra and Y. Prayudi, "Implementasi Multi *Smart Contract* pada Bukti Digital dan *Chain of Custody* dalam meningkatkan Keamanan dan Integritas Bukti Digital," JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia), Vol. 6, No. 2, pp. 98–108, 2021, DOI: 10.32528/justindo.v6i2.3945.

[13] S. A. A. Jakaria and T. Dirgahayu, "Sistem Kependudukan Terdesentralisasi menggunakan *Blockchain*, *Smart Contract*, dan *IPFS*," *Technologia: Jurnal Ilmiah*, Vol. 16, No. 4, pp. 767–774, 2025, DOI: 10.31602/tji.v16i4.20526.

[14] N. A. Santoso, P. Juanta, S. Maulana, K. Toktar, and A. Khanza, "*Decentralized File Sharing Infrastructure with IPFS for Censorship Resistance in Blockchain Ecosystems*," Blockchain Frontier Technology, Vol. 5, No. 1, pp. 80–89, 2025, DOI: 10.34306/bfront.v5i1.835.

[15] J. Sun, X. Yao, S. Wang, and Y. Wu, "*Blockchain-based Secure Storage and Access Scheme for Electronic Medical Records in IPFS*," *IEEE Access*, Vol. 8, pp. 59389–59401, 2020, DOI: 10.1109/ACCESS.2020.2982964.