EKSPLORASI BLOCKCHAIN, *SMART CONTRACT*, DAN IPFS UNTUK PENGEMBANGAN SISTEM KEPENDUDUKAN TERDESENTRALISASI

Disusun Oleh:

N a m a : Satria Avecinna Azzra Jakaria

NIM : 20523182

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

EKSPLORASI BLOCKCHAIN, *SMART CONTRACT*, DAN IPFS UNTUK PENGEMBANGAN SISTEM KEPENDUDUKAN TERDESENTRALISASI

TUGAS AKHIR

Disusun Oleh:

N a m a : Satria Avecinna Azzra Jakaria

NIM : 20523182

Yogyakarta, 12 Februari 2026

Pembimbing,

(Dr. Raden Teduh Dirgahayu, S.T, M.Sc)

HALAMAN PENGESAHAN DOSEN PENGUJI

EKSPLORASI BLOCKCHAIN, *SMART CONTRACT*, DAN IPFS UNTUK PENGEMBANGAN SISTEM KEPENDUDUKAN TERDESENTRALISASI

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 10 Februari 2026

Tim Penguji

Dr. Raden Teduh Dirgahayu, S.T., M.Sc. _____

Anggota 1

Dr. Nur Wijayaning Rahayu, S.Kom., M.Cs. _____

Anggota 2

Taufiq Hidayat, S.T., M.C.S. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Satria Avecinna Azzra Jakaria

NIM : 20523182

Tugas akhir dengan judul:

EKSPLORASI BLOCKCHAIN, *SMART CONTRACT*, DAN IPFS UNTUK PENGEMBANGAN SISTEM KEPENDUDUKAN TERDESENTRALISASI

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 12 Februari 2026

(Satria Avecinna Azzra Jakaria)

HALAMAN PERSEMBAHAN

Segala puji dan syukur saya panjatkan ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan kekuatan-Nya sehingga tugas akhir ini dapat terselesaikan dengan baik. Tanpa izin dan pertolongan-Nya, setiap usaha dan doa tidak akan bermakna apa-apa.

Dengan penuh kerendahan hati, karya sederhana ini saya persembahkan kepada kedua orang tua tercinta. Terima kasih atas kasih sayang, doa, dan pengorbanan yang tiada henti mengiringi setiap langkah saya. Semoga Allah SWT senantiasa melimpahkan kesehatan, keberkahan, dan kebahagiaan kepada Ayah dan Ibu.

Saya persembahkan pula karya ini kepada dosen pembimbing saya, yaitu Dr. Raden Teduh Dirgahayu, S.T., M.Sc. yang telah membimbing, mengarahkan, dan membekali dengan ilmu yang berharga. Bimbingan, arahan, serta ilmu yang diberikan merupakan bekal berharga bagi perjalanan hidup dan masa depan saya.

Tidak lupa, karya ini saya persembahkan kepada teman-teman saya yang selalu menghadirkan tawa, kebahagiaan, dan semangat di tengah padatnya ruang kehidupan. Kehadiran kalian menjadi penghibur sekaligus penguat yang membuat perjalanan ini terasa lebih ringan dan penuh warna.

Selain itu, karya ini juga saya persembahkan kepada ekosistem dan industri blockchain di Indonesia. Semoga penelitian ini dapat menjadi langkah kecil yang turut berkontribusi dalam mendorong perkembangan teknologi blockchain di tanah air, sehingga Indonesia mampu bersaing dan berkembang lebih maju dalam menghadapi tantangan era digital.

Terakhir, persembahan ini juga saya tujukan kepada diri saya sendiri yang terus berusaha, bertahan, berproses, dan berkembang di tengah keterbatasan dan lika-liku kehidupan yang setiap harinya semakin terasa berat. Semoga karya ini menjadi pengingat bahwa dengan doa, ikhtiar, dan keyakinan kepada Allah SWT, setiap langkah akan menemukan jalannya.

HALAMAN MOTO

"If A Man Empties His Purse into His Head No Man Can Take It Away"

Benjamin Franklin

"Whenever I get to a low point where I think 'why do I even bother?'

I just try to remind myself 'this is where most people stop, and this is why they don't win.'"

Alex Hormozi

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur saya panjatkan ke hadirat Allah SWT yang melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir. Tugas akhir ini disusun sebagai salah satu syarat utnuk memperoleh gelar Sarjana Komputer di Program Sarjana Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Dalam proses penyusunan tugas akhir ini, saya menyadari bahwa tanpa bantuan, dukungan, dan doa dari berbagai pihak, niscaya tugas akhir ini tidak akan terselesaikan dengan baik. Oleh karena itu, pada kesempatan ini saya menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang selalu memberikan kekuatan, kesehatan, dan keberkahan dalam setiap langkah.
2. Kedua orang tua tercinta atas doa, kasih sayang, dan dukungan yang tiada henti.
3. Bapak Teduh Dirgahayu, S.T., M.Sc. selaku dosen pembimbing yang selalu bersedia memberikan arahan, bimbingan, dan ilmu dalam proses penyusunan skripsi ini.
4. Pihak Kalurahan yang telah memberikan data beserta informasi berharga dalam mendukung penelitian ini.
5. Risto yang selalu membantu saya selama pengerjaan tugas akhir ini.
6. Teman-teman Sundawa atas kehadirannya dengan membawa canda dan tawa yang selalu menghiasi kehidupan saya.

Saya menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga karya ini dapat memberikan manfaat, baik sebagai tambahan pengetahuan maupun kontribusi kecil bagi pengembangan teknologi blockchain dalam mendukung layanan publik di Indonesia.

Yogyakarta, 12 Februari 2026

(Satria Avecinna Azzra Jakaria)

SARI

Penelitian ini mengeksplorasi potensi pemanfaatan teknologi blockchain beserta ekosistem pendukungnya, yaitu *smart contract* dan *InterPlanetary File System* (IPFS), dalam konteks pengelolaan data kependudukan dengan studi kasus layanan administrasi kependudukan di tingkat kalurahan. Eksplorasi ini dilatarbelakangi oleh karakteristik sistem kependudukan konvensional yang masih bersifat terpusat, bergantung pada proses verifikasi manual, serta memiliki keterbatasan dalam aspek transparansi, keterlacakan, dan pengelolaan data digital yang aman.

Metodologi penelitian menggunakan pendekatan rekayasa perangkat lunak dengan model pengembangan *Waterfall* yang meliputi tahap analisis kebutuhan, perancangan arsitektur, implementasi purwarupa, dan pengujian. Sistem dikembangkan dalam bentuk aplikasi terdesentralisasi (*decentralized application/dApp*) untuk mengkaji bagaimana *smart contract* dapat digunakan sebagai mekanisme pencatatan transaksi, pengendalian akses berbasis peran, serta pengelolaan alur verifikasi berlapis. Penyimpanan dokumen kependudukan dilakukan secara *off-chain* menggunakan IPFS, dengan *Content Identifier* (CID) dicatat pada blockchain guna mengevaluasi integritas dan keterhubungan data antara lapisan *on-chain* dan *off-chain*.

Hasil pengujian menunjukkan bahwa mekanisme pengajuan permohonan, verifikasi bertingkat oleh kalurahan dan Dukcapil, serta penyimpanan dan pengambilan dokumen terenkripsi melalui IPFS dapat dijalankan secara konsisten sesuai dengan rancangan. Selain itu, pencatatan *event on-chain* memungkinkan terbentuknya jejak audit digital yang transparan dan tidak dapat dimodifikasi. Temuan ini memberikan gambaran empiris mengenai karakteristik teknis, performa, serta implikasi penerapan arsitektur terdesentralisasi dalam konteks layanan publik. Penelitian ini diharapkan dapat menjadi referensi awal dalam kajian pengembangan sistem administrasi kependudukan berbasis teknologi terdesentralisasi di Indonesia.

Kata kunci: Aplikasi Terdesentralisasi, Blockchain, Data Kependudukan, IPFS, Pencatatan Sipil, *Smart Contract*, Sistem Terdesentralisasi.

GLOSARIUM

Enkripsi	Proses mengubah data asli menjadi bentuk yang tidak dapat dibaca oleh pihak yang tidak memiliki kunci dekripsi, untuk menjaga kerahasiaan informasi.
JSON	JavaScript *Object Notaion*, format pertukaran data berbasis teks yang digunakan untuk menyimpan dan mentransfer informasi dalam struktur yang ringan dan mudah dibaca oleh manusia maupun mesin.
Node	Komputer atau entitas dalam jaringan blockchain yang menyimpan salinan buku besar (*ledger*) dan berpartisipasi dalam proses validasi transaksi.
Wallet	Dompet digital yang digunakan untuk menyimpan identitas dan kunci privat pengguna dalam melakukan transaksi di jaringan blockchain.
Kalurahan	Unit pemerintahan setingkat kelurahan yang digunakan secara resmi di Daerah Istimewa Yogyakarta. Dalam penelitian ini, istilah kalurahan digunakan karena lokasi observasi berada di salah satu kelurahan di Yogyakarta, dan secara fungsional memiliki peran yang setara dengan kelurahan dalam konteks administratif nasional.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi blockchain telah membawa perubahan signifikan dalam cara sistem digital dirancang dan dikelola, khususnya dalam aspek kepercayaan, keamanan data, dan transparansi proses. Sejak diperkenalkan oleh Nakamoto (2008), blockchain berkembang dari sekadar teknologi pendukung mata uang kripto menjadi fondasi bagi berbagai sistem terdesentralisasi. Implementasi teknologi ini telah menyentuh berbagai lini bisnis modern dengan keunggulan spesifik di tiap industrinya, mulai dari sektor *Internet of Things* (IoT), keuangan, manajemen rantai pasok, hingga sistem reputasi (Gad et al., 2022). Perkembangan tersebut menunjukkan bahwa blockchain tidak hanya relevan sebagai inovasi finansial, tetapi juga sebagai teknologi dasar yang berpotensi mengubah paradigma perancangan sistem informasi, khususnya dalam konteks sistem yang menuntut tingkat keandalan, transparansi, dan kepercayaan yang tinggi.

Potensi blockchain sebagai fondasi sistem terdesentralisasi tidak terlepas dari karakteristik teknis yang melekat pada teknologi ini. Blockchain bekerja sebagai buku besar digital (*distributed ledger*) dengan sistem pencatatan terdistribusi berbasis jaringan *peer-to-peer* (P2P) yang merekam transaksi secara kronologis, tidak dapat diubah, dan diverifikasi oleh seluruh *node* dalam jaringan, sehingga menjamin akuntabilitas, integritas, transparansi, serta memiliki ketahanan terhadap manipulasi data (Nakamoto, 2008; Chander et al., 2023; Zheng et al., 2017). Selain itu, blockchain tidak memiliki ketergantungan pada satu otoritas pusat, yang secara signifikan mengurangi risiko kegagalan terpusat (*single point of failure*) serta meningkatkan keandalan sistem (Elisa et al., 2018). Kombinasi karakteristik tersebut menjadikan blockchain relevan untuk dieksplorasi sebagai alternatif arsitektur sistem informasi, khususnya pada domain yang menuntut tingkat keamanan, transparansi, dan akuntabilitas yang tinggi.

Seiring dengan perkembangannya, blockchain tidak berdiri sendiri, melainkan didukung oleh ekosistem teknologi lain seperti *smart contract* dan *InterPlanetary File System* (IPFS). *Smart contract* memungkinkan eksekusi logika bisnis secara otomatis tanpa ketergantungan pada pihak perantara, sehingga proses layanan dapat berjalan lebih efisien dan transparan (Kushwaha et al., 2022). Sementara itu, IPFS menyediakan infrastruktur penyimpanan dokumen yang efisien, terenkripsi, serta terdistribusi secara desentralisasi untuk menghindari

beban data berlebih pada jaringan blockchain (Trautwein et al., 2022). Integrasi blockchain, *smart contract*, dan IPFS sering digunakan dalam pendekatan *hybrid* antara *on-chain* dan *off-chain*.

Meskipun teknologi blockchain telah banyak diadopsi pada sektor bisnis dan industri, penerapan teknologi ini dalam konteks layanan publik dan sistem pemerintahan masih menghadapi berbagai tantangan sekaligus peluang eksplorasi. Sistem layanan publik umumnya dibangun menggunakan arsitektur tersentralisasi untuk memudahkan pengendalian, standarisasi, dan pengawasan oleh otoritas tertentu. Namun, pendekatan tersebut juga membawa keterbatasan, seperti rendahnya transparansi proses, ketergantungan pada verifikasi manual, serta tingginya risiko kegagalan sistem ketika terjadi gangguan pada pusat pengelolaan data. Kondisi ini menunjukkan adanya kebutuhan untuk mengeksplorasi pendekatan arsitektur alternatif yang mampu meningkatkan keandalan, akuntabilitas, dan kepercayaan publik terhadap sistem digital pemerintahan.

Salah satu domain layanan publik yang memiliki karakteristik sesuai untuk mengeksplorasi penerapan teknologi blockchain adalah sistem administrasi kependudukan. Data kependudukan sendiri merupakan fondasi utama dalam penyelenggaraan pelayanan publik, pemberian identitas hukum warga negara, serta perencanaan pembangunan nasional. Banyaknya hal penting yang bergantung terhadap data kependudukan menuntut sistem pengelolaan yang mampu menjamin keakuratan, integritas, keterlacakan, serta perlindungan terhadap penyalahgunaan data. Karakteristik tersebut menjadikan administrasi kependudukan sebagai konteks yang relevan untuk menguji sejauh mana teknologi blockchain dan ekosistem pendukungnya dapat diterapkan dalam sistem layanan publik yang nyata.

Di Indonesia, pengelolaan administrasi kependudukan diatur melalui Undang-Undang Nomor 23 Tahun 2006 tentang Administrasi Kependudukan yang kemudian diperbarui dengan Undang-Undang Nomor 24 Tahun 2013. Regulasi tersebut menjadi dasar bagi pemerintah dalam mengembangkan Sistem Informasi Administrasi Kependudukan (SIAK) sebagai sistem nasional yang dikelola secara terintegrasi oleh pemerintah daerah untuk menjamin keseragaman dan keakuratan data kependudukan (Dukcapil Ende, 2024). Dalam implementasinya, SIAK berfungsi sebagai sarana utama pelaksanaan administrasi kependudukan nasional dengan koordinasi di bawah Kementerian Dalam Negeri (Kemendagri).

Pada struktur penyelenggaraannya, pemerintah pusat melalui Kemendagri berperan sebagai pengelola sistem administrasi kependudukan nasional, sementara pemerintah

kabupaten/kota melalui Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil), serta petugas registrasi di tingkat desa atau kalurahan (setingkat kelurahan, sesuai nomenklatur administratif di Daerah Istimewa Yogyakarta) bertugas melaksanakan pendaftaran penduduk dan pencatatan sipil. Kehadiran sistem ini merupakan bagian dari upaya digitalisasi administrasi kependudukan yang sebelumnya dilakukan melalui Sistem Informasi Manajemen Kependudukan (SIMDUK), serta digunakan sebagai basis penyusunan kebijakan publik berbasis data (Dwiyanto, 2010; Munarja, 2014).

Pada tingkat operasional, kalurahan memiliki peran strategis sebagai pintu pertama pelayanan administrasi kependudukan. Petugas registrasi kalurahan bertanggung jawab menerima permohonan dari masyarakat, melakukan verifikasi awal atas kelengkapan dan keabsahan dokumen, serta meneruskan berkas yang telah dinyatakan valid kepada Disdukcapil untuk proses penetapan dan penerbitan dokumen resmi. Dalam praktiknya, banyak layanan pencatatan peristiwa kependudukan mensyaratkan rekomendasi administratif dari kalurahan sebagai dasar persetujuan, sehingga efektivitas pengelolaan data dan proses verifikasi di tingkat kalurahan sangat menentukan kualitas dan ketepatan layanan administrasi kependudukan.

Meskipun sistem administrasi kependudukan telah didukung oleh infrastruktur digital, implementasinya di tingkat operasional masih menghadapi sejumlah tantangan. Dalam praktik pelayanan, proses administrasi kependudukan kerap melibatkan tahapan manual, keterbatasan akses masyarakat terhadap informasi proses layanan secara transparan (Wijaya et al., 2024), serta mekanisme pengiriman berkas fisik antarunit layanan (Jai et al., 2016). Kondisi tersebut berpotensi berdampak pada meningkatnya waktu penyelesaian layanan serta keterbatasan keterlacakan proses administrasi. Selain itu, penggunaan arsitektur sistem yang bersifat tersentralisasi dalam pengelolaan data juga membawa risiko kegagalan terpusat (*single point of failure*) dan meningkatkan dampak ketika terjadi gangguan teknis atau pelanggaran keamanan data, khususnya pada pengelolaan data yang bersifat sensitif (Elisa et al., 2018).

Sejumlah penelitian sebelumnya telah mengusulkan pemanfaatan teknologi blockchain dalam sistem kependudukan, mulai dari rancangan arsitektur *smart city* yang memanfaatkan pencatatan digital terintegrasi (Nugroho et al., 2024), pengembangan prototipe otomasi pencatatan peristiwa kependudukan (Saian et al., 2024), hingga model konseptual untuk mendukung pelaksanaan sensus nasional berbasis blockchain (Rasheed & Louca, 2024). Meskipun demikian, penelitian-penelitian tersebut umumnya masih menempatkan blockchain sebagai solusi konseptual atau prototipe fungsional, sehingga peluang untuk mengeksplorasi karakteristik teknis teknologi blockchain seperti mekanisme *smart contract*, pencatatan

terdistribusi, serta integrasi penyimpanan *off-chain* dalam konteks layanan administrasi kependudukan yang nyata masih terbuka. Secara khusus, eksplorasi penerapan arsitektur terdesentralisasi yang disesuaikan dengan alur pelayanan di tingkat kalurahan serta kebutuhan perlindungan dan pengelolaan dokumen kependudukan menjadi aspek yang menarik untuk dikaji lebih lanjut.

Berdasarkan kondisi tersebut, penelitian ini berfokus pada eksplorasi pemanfaatan teknologi blockchain beserta ekosistem pendukungnya, yaitu *smart contract* dan IPFS, dalam konteks pengelolaan administrasi kependudukan. Sistem kependudukan diposisikan sebagai studi kasus untuk mengevaluasi bagaimana arsitektur terdesentralisasi dapat diterapkan pada layanan publik yang menuntut tingkat keamanan, transparansi, dan akuntabilitas yang tinggi. Penelitian ini diposisikan sebagai pengembangan purwarupa teknologi yang berfokus pada pengujian karakteristik, performa, serta implikasi teknis penerapan blockchain dalam pengelolaan data kependudukan, guna mengeksplorasi potensi arsitektur terdesentralisasi sebagai pendekatan alternatif dalam layanan administrasi publik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka didapat rumusan masalah dari penelitian ini, diantaranya:

a. Bagaimana penerapan arsitektur terdesentralisasi berbasis blockchain dalam pengelolaan data kependudukan di tingkat kalurahan ditinjau dari struktur data, alur transaksi, dan mekanisme pencatatan?
b. Bagaimana implementasi *smart contract* dalam mendukung transparansi, integritas data, serta pengendalian akses pada proses pengajuan dan verifikasi permohonan pencatatan sipil?
c. Bagaimana integrasi *InterPlanetary File System* (IPFS) digunakan sebagai media penyimpanan dokumen kependudukan secara terdistribusi, serta bagaimana karakteristik efisiensi dan ketahanan data yang dihasilkan?
d. Bagaimana hasil uji performa teknologi blockchain dan IPFS jika digunakan dalam konteks layanan administrasi kependudukan?

1.3 Batasan Masalah

Penelitian ini memiliki beberapa batasan agar pembahasan lebih terarah dan sesuai dengan tujuan penelitian, diantaranya:

a. Penelitian difokuskan pada eksplorasi penerapan teknologi blockchain, *smart contract*, dan *InterPlanetary File System* (IPFS) dalam konteks pengelolaan data kependudukan, tanpa membahas aspek kebijakan, regulasi, atau implementasi sistem administrasi kependudukan secara nasional.
b. Sistem hanya akan diuji secara lokal menggunakan jaringan Hardhat.
c. Evaluasi sistem difokuskan pada karakteristik teknis dan performa teknologi yang digunakan, seperti alur transaksi, pemrosesan *smart contract*, dan integrasi penyimpanan *off-chain* melalui IPFS, serta tidak mencakup uji performa skala besar atau pengujian pada beban pengguna nyata.
d. Implementasi algoritma kriptografi dilakukan secara sederhana dan terbatas pada kebutuhan dasar sistem, karena fokus penelitian berada pada eksplorasi arsitektur sistem terdesentralisasi, bukan pada optimasi atau analisis mendalam terhadap keamanan kriptografi.
e. Sistem yang dikembangkan tidak membahas integrasi penuh dengan SIAK nasional atau sistem Dukcapil yang sesungguhnya, melainkan sebatas purwarupa dan implementasi fungsional sebagai sarana eksplorasi teknologi.
f. Ruang lingkup sistem dibatasi pada proses pengelolaan data kependudukan dan pencatatan sipil yang meliputi pengajuan, verifikasi, dan penerbitan dokumen kependudukan seperti kelahiran, kematian, perkawinan, perceraian, serta perpindahan penduduk.
g. Pada proses perkawinan dan perceraian, alur sistem disederhanakan menggunakan mekanisme umum pencatatan sipil tanpa melibatkan lembaga keagamaan seperti KUA atau sejenisnya. Penyesuaian ini dilakukan agar penelitian berfokus pada aspek teknis sistem terdesentralisasi, bukan pada variasi prosedur antaragama.
h. Wilayah implementasi difokuskan pada satu wilayah, yaitu tingkat kalurahan, sebagai representasi dari unit pelayanan publik terdekat dengan masyarakat.
i. Data yang digunakan bersifat *dummy*/sintetis, bukan data kependudukan asli, untuk menjaga keamanan dan privasi.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk mengeksplorasi potensi penerapan teknologi blockchain beserta ekosistem pendukungnya, yaitu *smart contract* dan *InterPlanetary File System* (IPFS), dalam pengelolaan data kependudukan dengan menggunakan konteks layanan administrasi kependudukan di tingkat kalurahan sebagai studi kasus. Penelitian ini difokuskan pada

pengembangan purwarupa teknologi untuk mengkaji karakteristik, performa, serta implikasi teknis penerapan arsitektur sistem terdesentralisasi dalam layanan publik.

Secara lebih khusus, tujuan dari penelitian ini adalah sebagai berikut:

a. Merancang arsitektur sistem terdesentralisasi berbasis blockchain yang mampu mengelola proses pengajuan, verifikasi, dan penerbitan dokumen kependudukan di tingkat kalurahan secara efisien dan transparan.
b. Mengimplementasikan *smart contract* sebagai logika utama dalam mendukung transparansi proses, integritas data, serta pengendalian akses pada proses pengajuan dan verifikasi permohonan pencatatan sipil.
c. Mengeksplorasi integrasi *InterPlanetary File System* (IPFS) sebagai mekanisme penyimpanan dokumen kependudukan secara *off-chain*, serta mengkaji karakteristik efisiensi dan ketahanan data yang dihasilkan.
d. Menguji performa penerapan teknologi blockchain dan IPFS dalam konteks sistem kependudukan yang dikembangkan.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

a. Secara teoritis, penelitian ini dapat menambah referensi dan pengembangan ilmu di bidang teknologi blockchain, khususnya dalam penerapan *decentralized application (dApp)* untuk sistem informasi kependudukan dan pencatatan sipil.
b. Secara praktis, penelitian ini diharapkan dapat memberikan gambaran dan referensi teknis mengenai penerapan teknologi blockchain, *smart contract*, dan IPFS sebagai pendekatan alternatif dalam pengelolaan data kependudukan, khususnya sebagai purwarupa dan bahan evaluasi teknologi bagi instansi pemerintah atau peneliti yang ingin mengeksplorasi potensi arsitektur terdesentralisasi dalam layanan administrasi publik.

1.6 Sistematika Penelitian

Sistematika penelitian disusun untuk memberikan gambaran umum mengenai isi laporan tugas akhir ini. Adapun susunan sistematikanya adalah sebagai berikut:

a. BAB 1 Pendahuluan

Bab ini menjelaskan gambaran umum penelitian, meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan. Bab ini menjadi dasar untuk memahami konteks dan arah penelitian yang dilakukan.

b. BAB 2 Kajian Pustaka

Bab ini berisi kajian teori dan literatur yang mendukung penelitian. Di dalamnya dibahas konsep-konsep utama seperti data kependudukan, teknologi blockchain, *smart contract*, *InterPlanetary File System* (IPFS), serta aplikasi terdesentralisasi (dApp). Selain itu, bab ini juga mencantumkan hasil kajian pustaka terhadap penelitian-penelitian terdahulu yang relevan untuk memperkuat posisi penelitian ini.

c. BAB 3 Metodologi Penelitian

Bab ini menjelaskan metode dan tahapan penelitian yang digunakan dalam pengembangan sistem. Metodologi yang diterapkan menggunakan model *Waterfall*, yang terdiri dari tahap analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Bab ini juga mencakup penjelasan aktor, proses bisnis, rancangan arsitektur sistem, rancangan *smart contract*, dan mekanisme penyimpanan dokumen melalui IPFS.

d. BAB 4 Hasil dan Pembahasan

Bab ini memaparkan hasil implementasi sistem terdesentralisasi yang telah dikembangkan. Pembahasan mencakup hasil perancangan *smart contract*, integrasi IPFS, antarmuka pengguna, serta hasil pengujian fungsionalitas sistem. Selain itu, disajikan pula analisis terhadap hasil pengujian seperti keterpenuhan fungsi, keamanan data, serta efisiensi sistem dalam mendukung layanan administrasi kependudukan di tingkat kalurahan.

e. BAB 5 Kesimpulan dan Saran

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian sebagai jawaban atas rumusan masalah yang telah dirumuskan pada Bab 1. Selain itu, disajikan pula saran untuk pengembangan dan penelitian lebih lanjut agar sistem dapat diimplementasikan secara lebih luas pada konteks layanan publik yang sebenarnya.

BAB II
KAJIAN PUSTAKA

2.1 Landasan Teori

Landasan teori pada penelitian ini mencakup beberapa aspek penting yang menjadi fondasi eksplorasi sistem terdesentralisasi berbasis blockchain untuk manajemen data kependudukan dan pencatatan sipil. Aspek-aspek tersebut meliputi teori mengenai blockchain sebagai teknologi utama dari infrastruktur terdesentralisasi, *smart contract* sebagai logika otomatisasi transaksi, IPFS sebagai penyimpanan berkas terdesentralisasi, konsep aplikasi terdesentralisasi (dApp) sebagai bentuk implementasi sistem secara utuh, serta data kependudukan sebagai studi kasus pengelolaan data.

2.1.1 Blockchain

Blockchain merupakan teknologi *distributed ledger* atau buku besar terdistribusi yang memungkinkan penyimpanan dan pertukaran data secara aman, transparan, serta tidak bergantung pada otoritas pusat atau pihak ketiga. Dalam sistem ini setiap transaksi direkam ke dalam blok yang saling terhubung melalui algoritma kriptografi, membentuk rantai data (*chain of blocks*) yang sulit diubah secara sepihak setelah diverifikasi oleh jaringan (Swan, 2015).

Teknologi blockchain pertama kali diperkenalkan melalui mata uang kripto bernama Bitcoin oleh Satoshi Nakamoto (2008). Sejak itu, blockchain telah berkembang pesat melampaui ranah keuangan, mencakup berbagai bidang seperti *smart contract*, manajemen rantai pasok, identitas digital, serta layanan publik (Zheng et al., 2017).

Dalam konteks pengelolaan data kependudukan, blockchain menawarkan solusi terhadap tantangan transparansi, integritas, dan keamanan data melalui sistem pencatatan terdistribusi berbasis jaringan *peer-to-peer (P2P)*. Blockchain tidak bergantung pada server pusat sehingga dapat mengurangi risiko kegagalan terpusat (*single point of failure*) dan meningkatkan keandalan data (Elisa et al., 2018). Teknologi ini juga memungkinkan partisipasi masyarakat dalam pengelolaan data pribadinya, mendukung prinsip *data ownership* dan tata kelola informasi yang lebih bertanggung jawab (Zieglmeier et al., 2023).

Karakteristik Blockchain

Beberapa karakteristik fundamental yang membedakan blockchain dari sistem terpusat konvensional meliputi:

a. Desentralisasi: Blockchain tidak dikendalikan oleh satu entitas pusat, melainkan oleh jaringan *node* terdistribusi yang bekerja secara kolektif dalam memvalidasi transaksi (Zhang et al., 2017).
b. Keamanan: Data diamankan menggunakan teknik kriptografi sehingga sulit dimanipulasi atau diretas tanpa terdeteksi.
c. Transparansi: Semua transaksi tercata secara publik dalam jaringan, memungkinkan verifikasi oleh seluruh peserta sistem (Yaga et al., 2019).
d. Immutabilitas: Data yang telah diverifikasi dan dicatat dalam blokchain tidak dapat dihapus atau diubah tanpa persetujuan mayoritas *node* dalam jaringan (Pilkington, 2016).

Komponen Utama Blockchain

Blockchain tersusun atas beberapa komponen teknis yang bekerja secara terpadu untuk menjamin keandalan data:

a. Blok: Unit penyimpan yang memuat kumpulan transaksi, *timestamp*, dan *hash* dari blok sebelumnya, membentuk struktur rantai yang kronologis.
b. *Hash*: Nilai kriptografi unik yang dihasilkan dari isi blok dan berfungsi sebagai identitas digital untuk menjamin integritas data.
c. *Node*: Komputer atau entitas dalam jaringan yang menyimpan salinan penuh dari buku besar serta berpartisipasi dalam proses verifikasi transaksi.
d. Protokol Konsensus: Mekanisme yang digunakan untuk mencapai kesepakatan di antara seluruh *node* terkait validitas blok baru, seperti *Proof of Work (PoW)* dan *Proof of Stake (PoS)* (Nguyen, 2016).

Proses kerja blockchain dimulai ketika pengguna mengajukan transaksi. Transaksi tersebut diverifikasi oleh sejumlah *node* menggunakan protokol konsensus. Setelah dianggap valid, transaksi akan dimasukkan ke dalam blok baru yang kemudian dirantai dengan blok sebelumnya menggunakan *hash* kriptografi. Blok baru ini kemudian didistribusikan ke seluruh jaringan, memperbarui salinan buku besar terdistribusi secara serentak (Narayanan et al., 2016).

Kelebihan dan Kekurangan Blockchain

Kelebihan:

a. Keamanan tinggi: Setiap transaksi terlindungi oleh mekanisme kriptografi dan konsensus yang mencegah perubahan data secara ilegal (Zheng et al., 2017).

b. Transparansi dan akuntabilitas: Seluruh catatan transaksi dapat diverifikasi secara terbuka.
c. Desentraliasasi: Menghilangkan ketergantungan pada pihak ketiga sehingga mengurangi risiko monopoli dan manipulasi.

Kekurangan:

a. Skalabilitas: Karena setiap *node* menyimpan salinan lengkap dari blockchain, pertumbuhan data yang besar dapat memperlambat performa jaringan (Zheng et al., 2017).
b. Konsumsi energi: Algoritma seperti PoW membutuhkan daya komputasi dan energi yang tinggi (Narayanan et al., 2016).
c. Regulasi: Pengaturan hukum yang belum seragam secara global sering menjadi kendala penerapan blockchain di sektor publik (Pilkington, 2016).

Jenis-Jenis Blockchain

Berdasarkan struktur, akses, dan mekanisme konsensusnya, blockchain dibedakan menjadi empat jenis utama, yaitu *public*, *private*, *hybrid*, dan *consortium* blockchain (Damre et al., 2022; Siddharth, 2025).

Public blockchain merupakan jaringan terbuka yang dapat diakses siapa pun tanpa izin, di mana setiap *node* memiliki salinan buku besar dan berpartisipasi dalam proses validasi. Jaringan ini bersifat sepenuhnya desentralisasi dan menggunakan mekanisme konsensus seperti *Proof of Work* atau *Proof of Stake* untuk memverifikasi transaksi. Kelebihannya terletak pada transparansi, keamanan, dan kepercayaan tinggi karena semua data dapat diverifikasi publik, namun memiliki kelemahan berupa kecepatan transaksi yang lambat, konsumsi energi besar, serta masalah skalabilitas (Damre et al., 2022; Siddharth, 2025).

Private blockchain adalah jaringan tertutup yang hanya diakses oleh pihak tertentu melalui izin dari otoritas pusat. Jenis ini cocok digunakan oleh organisasi yang membutuhkan privasi dan kendali penuh terhadap data, seperti perusahaan atau lembaga pemerintah. Meskipun lebih cepat, efisien, dan mudah diskalakan dibandingkan blockchain publik, struktur sentralisasi membuatnya kurang aman dan mengurangi tingkat kepercayaan antar-*node* (Damre et al., 2022)

Hybrid blockchain menggabungkan kelebihan *public* dan *private* blockchain dengan memisahkan data menjadi dua lapisan: bagian publik untuk transparansi dan bagian privat untuk keamanan. Sistem ini memungkinkan validasi publik terhadap transaksi tanpa membuka seluruh data sensitif. Jenis ini banyak digunakan dalam sektor keuangan dan pemerintahan

karena fleksibilitas dan skalabilitasnya, meski penerapannya lebih kompleks dan memerlukan tata kelola yang cermat (Damre et al., 2022; Siddharth, 2025).

Consortium blockchain (atau *federated* blockchain) merupakan jaringan yang dikelola bersama oleh beberapa organisasi dengan hak validasi yang setara. Jenis ini memberikan keseimbangan antara privasi dan efisiensi, menjadikannya ideal untuk kolaborasi antar-lembaga seperti sistem rantai pasok, perbankan, atau layanan publik. Namun, koordinasi lintas-entitas dan keterbatasan transparansi sering menjadi tantangan utama (Damre et al., 2022).

Perbedaan dari keempat jenis blockchain tersebut dirangkum dalam Tabel 2.1 berikut.

Tabel 2.1 Ringkasan Perbedaan Empat Jenis Blockchain

Jenis	Akses	Pengelolaan	Kelebihan	Kekurangan
Public	Terbuka untuk umum	Desentralisasi penuh	Transparansi dan kepercayaan tinggi	Lembat, boros energi, dan kurang *scalable*
Private	Tertutup (izin khusus)	Sentralisasi oleh otoritas pusat	Cepat, eifisien, dan privat	Sentralisasi dan rawan manipulasi
Hybrid	Kombinasi publik-privat	Otoritas pusat dengan validasi publik	Aman, fleksibel, dan *scalable*	Implementasi kompleks dan perlu tata kelola ketat
Consortium	Tertutup antarorganisasi	Kolaboratif multi-entitas	Efisien, seimbang, dan terkontrol	Transparansi terbatas dan koordinasi rumit

2.1.2 *Smart Contract*

Smart contract atau kontrak pintar merupakan perjanjian digital yang diimplementasikan dalam bentuk potongan kode yang dapat menegakkan ketentuan secara otomatis ketika kondisi yang telah ditentukan sebelumnya terpenuhi. Menurut Kushwaha et al. (2022), "*a smart contract is a contractual agreement embedded in a self-enforceable piece of code,"* di mana para pihak yang terlibat berinteraksi berdasarkan parameter yang telah disepakati, dan ketika kondisi tertentu terpenuhi, operasi yang telah didefinisikan akan dieksekusi secara otomatis. Kontrak ini disimpan dan dijalankan di atas jaringan blockchain, yang menjamin keamanan, transparansi, serta pelaksanaan transaksi tanpa memerlukan pihak ketiga atau otoritas pusat.

Konsep *smart contract* pertama kali diperkenalkan oleh Nick Szabo (1997), yang mendefinisikannya sebagai "serangkaian perjanjian dalam bentuk digital, termasuk protokol di

mana beberapa pihak melaksanakan janji tersebut." Szabo menekankan bahwa kontrak digital dapat mengotomatiskan perjanjian hukum tanpa campur tangan pihak eksternal, sehingga meningkatkan efisiensi dan mengurangi potensi pelanggaran kesepakatan.

Kemudian, Buterin (2014) mengembangkan konsep ini dalam platform Ethereum, yang memperluas fungsi *smart contract* menjadi lebih fleksibel dan dinamis. Melalui Ethereum, *smart contract* dapat diprogram untuk menangani berbagai logika bisnis yang kompleks, memungkinkan terbentuknya aplikasi terdesentralisasi (*decentralized applications* atau dApps) yang dapat berjalan secara otomatis berdasarkan aturan yang telah diprogram.

Secara teknis, *smart contract* memiliki dua elemen utama, yaitu variabel status (*state variable*) dan nilai status (*state value*), yang berfungsi untuk menyimpan dan memperbarui kondisi sistem selama kontrak berjalan. Menurut Wu et al. (2022), *smart contract* menggunakan logika berbasis pernyataan *if–then* dan *what-if* untuk menentukan skenario pemicu (*triggering scenarios*) dan aturan respons dari setiap ketentuan kontrak. Proses eksekusi dimulai ketika pengguna menandatangani dan mengirimkan transaksi melalui jaringan blockchain menggunakan tanda tangan digital. Transaksi tersebut kemudian disebarkan ke seluruh *node* untuk diverifikasi dan disimpan dalam blok baru.

Selain mekanisme kerjanya, *smart contract* juga memiliki sejumlah karakteristik utama yang membedakannya dari sistem digital konvensional. Karakteristik-karakteristik ini menjamin bahwa *smart contract* dapat berfungsi secara otonom, aman, serta transparan dalam ekosistem blockchain (Rawat et al., 2019; Prata et al., 2021). Berikut adalah beberapa karakteristik dari *smart contract*:

a. Tidak dapat diubah (*immutable*). Setelah kontrak di-*deploy* ke dalam jaringan blockchain, kode dan ketentuannya tidak dapat dimodifikasi. Konsekuensinya, apabila ditemukan kerentanan atau kesalahan setelah penerapan, perbaikan hanya dapat dilakukan dengan membuat versi baru dari kontrak tersebut (Rawat et al., 2019).
b. Kekal (*permanence*) karena data dan kode disimpan secara permanen di blockchain. Hal ini menjamin keaslian, keterlacakan, dan transparansi setiap transaksi yang dilakukan melalui kontrak (Rawat et al., 2019).
c. Bekerja secara otomatis (*automation*). Artinya, setiap aksi yang telah diprogram akan dieksekusi secara otomatis ketika kondisi tertentu terpenuhi, tanpa memerlukan campur tangan pihak ketiga, sehingga meningkatkan efisiensi dan mengurangi potensi kesalahan manusia (Rawat et al., 2019).

d. Eksekusi tanpa kepercayaan (*trustless execution*), di mana semua proses berlangsung dalam jaringan terdesentralisasi tanpa otoritas pusat. Dengan demikian, pihak-pihak yang tidak saling mengenal tetap dapat berinteraksi dan bertransaksi dengan aman berdasarkan aturan kode (Rawat et al., 2019).
e. Transparansi (*transparency*). Semua transaksi dan aktivitas yang terkait dengan kontrak tercatat secara publik di blockchain, memungkinkan proses audit yang terbuka bagi seluruh node dalam jaringan (Prata et al., 2021).
f. Berbiaya rendah (*cost-efficiency*). Karena berjalan secara otomatis dan tidak membutuhkan perantara, sistem ini dapat memangkas biaya administratif dan operasional, menjadikannya solusi efisien untuk transaksi digital yang berulang dan berbasis aturan (Prata et al., 2021).

Smart contract beroperasi di lapisan aplikasi (*application layer*) dari arsitektur blockchain (Kushwaha et al., 2022), sedangkan penyimpanan data dan validasi dilakukan di lapisan dasar (*base layer*). Dengan memanfaatkan mekanisme konsensus blockchain, setiap eksekusi kontrak dapat diverifikasi oleh seluruh *node* jaringan, memastikan integritas, keamanan, dan imutabilitas hasil transaksi.

Dalam konteks penelitian ini, *smart contract* dapat berfungsi sebagai mekanisme otomatisasi dalam pengajuan, verifikasi, dan penerbitan dokumen kependudukan. Misalnya, status permohonan akta kelahiran atau pindah domisili dapat diperbarui otomatis setelah diverifikasi oleh petugas berwenang.

2.1.3 *InterPlanetary File System* (IPFS)

IPFS adalah sebuah sistem penyimpanan dan distribusi data berbasis *peer-to-peer* yang menggunakan pendekatan *content-addressable* untuk mengidentifikasi dan mengakses data (Trautwein et al., 2022). Berbeda dengan HTTP (*Hypertext Transfer Protocol*) yang mengandalkan lokasi server, IPFS menggunakan *Content Identifiers* (CIDs) yang merupakan *hash* dari konten itu sendiri.

Menurut Trautwein et al. (2022) IPFS bekerja dengan cara menyimpan dan mengambil data berdasarkan isi kontennya, bukan lokasi penyimpanannya. Saat sebuah file diunggah, IPFS membaginya menjadi potongan-potongan kecil dan memberi masing-masing potongan sidik jari digital unik yang disebut CID. Data ini kemudian disebarkan ke banyak komputer (*peer*) dalam jaringan IPFS. Ketika pengguna ingin mengakses file, IPFS akan mencari tahu

siapa yang memiliki potongan data tersebut melalui sistem pencarian terdistribusi, lalu mengunduhnya langsung dari peer yang menyimpannya.

IPFS memungkinan data bisa disimpan dan diakses dari banyak tempat, membuatnya menjadi lebih tahan gangguan dan tidak bergantung pada satu server pusat. Sistem ini juga memastikan keaslian data karena setiap perubahan akan menghasilkan CID yang berbeda, sehingga pengguna selalu mendapatkan versi yang benar dari konten yang dicari.

Berdasarkan penjelasan oleh Trautwein et al. (2022) dapat disimpulkan beberapa karakteristik dari IPFS sebagai berikut:

a. *Conten-Based Addressing*: IPFS menggunakan sistem pengalamatan berbasis konten, bukan lokasi. Setiap file atau potongan data diberi CID yang merupakan hash dari isi data tersebut. Ini memungkinkan data diakses dari mana saja tanpa bergantung pada server tertentu, serta memastikan keaslian data karena perubahan sekecil apa pun akan menghasilkan CID yang berbeda.
b. Desentralisasi Penuh: IPFS adalah jaringan *peer-to-peer* yang sepenuhnya terdesentralisasi. Tidak ada satu entitas pun yang mengontrol seluruh sistem. Data disimpan dan disebarkan oleh banyak *node* di seluruh dunia, dan siapa pun bisa bergabung sebagai *node* tanpa izin khusus.
c. *Immutability* dan Verifikasi Mandiri: Karena CID didasarkan pada *hash* konten, data dalam IPFS bersifat *immutable* dan *self-certifying*. Artinya, pengguna bisa memverifikasi keaslian data hanya dengan mencocokkan CID-nya, tanpa perlu sertifikat atau otoritas pusat.
d. *Open Participation*: IPFS bersifat *open-source* dan komunitas-*driven*. Siapa pun dapat menjalankan *node* IPFS, berkontribusi pada pengembangan, dan ikut serta dalam proses desain melalui voting publik. Ini mendorong inovasi dan inklusivitas dalam pengembangan teknologi.
e. Toleransi terhadap *Churn* dan Kegagalan: IPFS dirancang untuk tahan terhadap *churn* (pergantian *node*) dan kegagalan jaringan. Dengan mereplikasi data ke 20 *node* terdekat, IPFS tetap dapat beroperasi meskipun beberapa *node offline*.

Dalam konteks penelitian ini, IPFS berperan sebagai lapisan penyimpanan *offchain* yang digunakan untuk menyimpan dokumen kependudukan secara terenkripsi dan terdistribusi. Pendekatan ini bertujuan untuk mengatasi keterbatasan blockchain dalam menyimpan data berukuran besar sekaligus menjaga integritas, ketersediaan, dan keaslian dokumen. Dengan integrasi antara *smart contract* dan IPFS, sistem dapat mencatat *hash* dari setiap dokumen di

blockchain sebagai bukti keaslian, sementara data aslinya tersimpan aman di jaringan IPFS. Mekanisme ini memungkinkan proses verifikasi dokumen dilakukan secara transparan dan efisien tanpa mengorbankan privasi pengguna, sehingga mendukung terciptanya layanan administrasi kependudukan yang aman, terdesentralisasi, dan tahan terhadap manipulasi data.

2.1.4 Aplikasi Terdesentralisasi

Setelah perkembangan *smart contract*, muncul konsep aplikasi terdesentralisasi atau dApp, yaitu bentuk aplikasi yang berjalan di atas infrastruktur blockchain dan memanfaatkan *smart contract* sebagai logika utama sistem. Dengan demikian, dApp menjadi lapisan aplikasi yang memungkinkan interaksi pengguna dengan blockchain secara langsung tanpa melalui otoritas pusat.

Menurut Financial Crimes Enforcement Network (FinCEN, 2019), aplikasi terdesentralisasi adalah aplikasi yang beroperasi pada jaringan *peer-to-peer* (P2P) tanpa adanya kontrol dari satu entitas atau administrator tunggal yang dapat diidentifikasi. Kontrol terhadap sistem sepenuhnya tersebar di berbagai *node* yang terhubung dalam jaringan blockchain, menjadikannya bebas dari otoritas pusat sekaligus lebih tahan terhadap manipulasi atau sensor.

Mengacu pada dokumentasi Ethereum (ethereum.org, 2025), aplikasi terdesentralisasi memiliki beberapa karakteristik utama yang membedakannya dari aplikasi konvensional, antara lain:

a. Terdesentralisasi, data dan proses eksekusi dApp tersebar di seluruh *node* dalam jaringan blockchain. Tidak ada server pusat atau otoritas tunggal yang mengendalikan sistem, sehingga meningkatkan transparansi, keamanan, serta ketahanan terhadap kegagalan terpusat (*single point of failure*).
b. Deterministik, setiap dApp akan selalu menghasilkan keluaran yang sama untuk masukan yang sama, terlepas dari di mana atau oleh siapa aplikasi dijalankan. Sifat ini memastikan konsistensi perilaku aplikasi di seluruh *node* jaringan.
c. *Turing Complete*, dApp memiliki kemampuan komputasi penuh (*Turing completeness*), yang memungkinkan *smart contract* di dalamnya memproses logika kompleks, seperti perhitungan kondisi, manajemen data, dan pengambilan keputusan otomatis.
d. Lingkungan Terisolasi, dApp dijalankan dalam lingkungan komputasi terpisah seperti *Ethereum Virtual Machine* (EVM), yang memastikan bahwa kesalahan atau bug pada satu *smart contract* tidak akan memengaruhi fungsionalitas jaringan secara keseluruhan.

Dengan demikian, sistem informasi yang menerapkan konsep terdesentralisasi juga termasuk dalam kategori aplikasi terdesentralisasi. Dalam konteks teknologi blockchain, istilah sistem terdesentralisasi merujuk pada arsitektur aplikasi yang tidak bergantung pada satu server pusat, melainkan dijalankan oleh *node-node* dalam jaringan secara kolektif. Oleh karena itu, penerapan konsep sistem terdesentralisasi pada penelitian ini secara langsung merepresentasikan implementasi dApp sebagai bentuk nyata dari aplikasi berbasis blockchain.

Dalam konteks penilitian ini, konsep dApp dapat digunakan untuk menciptakan sistem pelayanan publik yang lebih transparan, efisien, dan akuntabel. Melalui integrasi antara *smart contract* dan dApp, proses pengajuan, validasi, serta penerbitan dokumen kependudukan dapat dilakukan secara otomatis dan terekam di blockchain.

2.1.5 Data Kependudukan

Data kependudukan merupakan landasan esensial dalam pelaksanaan segala bentuk pelayanan publik, penetapan identitas hukum bagi warga negara, serta perumusan rancangan pembangunan di skala nasional. Dasar hukum yang mengatur tata kelola kependudukan di Indonesia diatur dalam Undang-Undang Nomor 23 Tahun 2006 tentang Administrasi Kependudukan, yang kemudian mengalami revisi melalui Undang-Undang Nomor 24 Tahun 2013 tentang Perubahan atas Undang-Undang Nomor 23 Tahun 2006.

Menurut Pasal 1 Ayat 9 Undang-Undang Nomor 23 Tahun 2006, data kependudukan adalah data perseorangan dan/atau data agregat yang terstruktur sebagai hasil dari kegiatan pendaftaran penduduk dan pencatatan sipil. Ketentuan ini menegaskan bahwa data kependudukan mencakup dua jenis informasi utama:

a. Data perseorangan, yaitu data individu yang mencakup identitas dasar seperti nama, jenis kelamin, tempat dan tanggal lahir, status perkawinan, kewarganegaraan, pekerjaan, agama, serta alamat tempat tinggal; dan
b. Data agregat, yakni kumpulan data penduduk dalam bentuk statistik kuantitatif maupun kualitatif yang menggambarkan jumlah, komposisi, distribusi, dan karakteristik penduduk.

Perubahan melalui Undang-Undang Nomor 24 Tahun 2013 memperluas cakupan data perseorangan dengan menambahkan elemen biometrik dan informasi digital, seperti sidik jari, iris mata, dan tanda tangan elektronik, untuk mendukung pelaksanaan Kartu Tanda Penduduk Elektronik (KTP-el). Pembaruan ini juga menegaskan pemanfaatan teknologi informasi dalam penyelenggaraan administrasi kependudukan agar lebih terintegrasi, akurat, dan aman.

Pasal 58 Undang-Undang Nomor 24 Tahun 2013 menegaskan bahwa data kependudukan yang digunakan untuk seluruh keperluan merupakan data yang bersumber dari Kementrian Dalam Negeri, sehingga menjamin keseragaman dan konsistensi data nasional untuk pelayanan publik, perencanaan pembangunan, alokasi anggaran, pembangunan demokrasi, serta penegakan hukum.

Sebagai bentuk implementasinya, pemerintah mengambangkan Sistem Informasi Administrasi Kependudukan (SIAK) sebagaimana dijelaskan dalam Pasal 1 Ayat 21 Undang-Undang Nomor 23 Tahun 2006 yang merupakan sistem berbasis teknologi informasi dan komunikasi untuk memfalitasi pengelolaan data kependudukan secara nasional. SIAK memungkinkan integrasi data dari pemerintah pusat hingga tingkat daerah (provinsi, kabupaten/kota, serta desa atau kalurahan), sehingga setiap peristiwa kependudukan dan pencatatan sipil dapat terdata secara *real-time* dan seragam di seluruh Indonesia.

Dalam praktiknya, pengelolaan data kependudukan berperan strategis dalam berbagai bidang, antara lain:

a. Menjadi dasar dalam perumusan kebijakan publik dan perencaan pembangunan nasional maupun daerah;
b. Mendukung efisiensi penyelenggaraan pelayanan administratif, seperti penerbitan dokumen kependudukan dan identitas hukum; serta
c. Menunjang koodinasi antarinstansi pemerintahan dalam pengelolaan sumber daya dan program sosial berbasis data.

Dengan demikian, sistem pengelolaan data kependudukan di Indonesia tidak hanya berfungsi sebagai sarana administratif, tetapi juga sebagai instrumen strategis untuk membangun tata kelola pemerintahan yang transparan, efisien, dan berorientasi pada keamanan informasi melalui penerapan sistem digital nasional seperti SIAK dan KTP-el.

2.2 Kajian Pustaka

Kajian pustaka ini bertujuan untuk menelaah berbagai penelitian terdahulu yang berkaitan dengan pemanfaatan teknologi blockchain dalam pengelolaan data kependudukan dan pencatatan sipil. Tinjauan ini difokuskan pada studi-studi yang membahas penerapan blockchain dalam sistem administrasi publik, identitas digital, serta integrasi *smart contract* dalam layanan pemerintahan. Melalui kajian ini, dapat diidentifikasi pendekatan, metodologi, serta hasil yang telah dicapai oleh penelitian sebelumnya, sehingga menjadi dasar dalam merumuskan arah dan kontribusi penelitian ini.

Berdasarkan fokus tersebut, sejumlah penelitian terdahulu telah dilakukan untuk mengeksplorasi penerapan teknologi blockchain, *smart contract*, serta aplikasi terdesentralisasi dalam konteks administrasi publik dan sistem kependudukan atau topik-topik yang masih relevan.

Kajian terhadap penelitian-penelitian tersebut dilakukan dengan menelaah literatur nasional dan internasional yang relevan, baik dari aspek tujuan, metodologi, maupun hasil yang dicapai. Ringkasan dari penelitian-penelitian tersebut disajikan pada Tabel 2.2 berikut ini.

Tabel 2.2 Kajian Pustaka

Referensi	Tujuan	Metode	Hasil
(Nugroho et al., 2023)	Mengembangkan arsitektur sistem manajemen data kependudukan berbasis blockchain untuk mendukung konsep smart city.	Meliputi analisis sistem yang sudah ada, studi literatur, dan rancangan arsitektur baru.	Blockchain meningkatkan efisiensi dalam pengelolaan data populasi melalui integrasi smart contract dan identitas digital.
(Rasheed & Louca, 2024)	Mengatasi permasalahan *missing persons* dalam sensus nasional, meningkatkan transparansi, akuntabilitas, serta keamanan data dengan mengevaluasi penerapan teknologi blockchain (Hyperledger Fabric) dalam sistem sensus nasional berbasis studi kasus di Pakistan.	Kualitatif dan eksperimental melalui studi literatur, wawancara terstruktur dengan staf sensus nasional, serta pengembangan dan pengujian prototipe sistem sensus berbasis blockchain menggunakan Hyperledger Fabric, IPFS, dan ReactJS.	Sistem sensus berbasis blockchain berhasil meningkatkan cakupan pendataan, keamanan, dan partisipasi masyarakat dengan mencatat populasi yang sebelumnya terlewat oleh metode tradisional.
(Saian et al., 2024)	Merancang dan mengimplementasikan prototipe aplikasi terdesentralisasi	Studi literatur, perancangan sistem dan infrastruktur,	Prototipe dApps berhasil diimplementasikan dengan tidak

	(dApp) untuk sistem manajemen data kependudukan yang aman menggunakan teknologi blockchain dan enkripsi AES-256.	pengembangan prototipe dApp dengan *smart contract* di blockchain *permissioned*, serta pengujian keamanan menggunakan OWASP ZAP dan analisis biaya *gas fee*.	ditemukan peringatan keamanan tingkat tinggi, dan data kependudukan dapat disimpan secara terenkripsi serta hanya dapat diakses oleh pemilik data melalui kunci akun.
(Wardhana, 2024)	Mengeksplorasi potensi dan manfaat implementasi teknologi blockchain dalam meningkatkan keamanan *database* penduduk di Kementerian Dalam Negeri (Kemendagri).	Deskriptif kualitatif melalui studi literatur terhadap sumber-sumber seperti jurnal ilmiah, publikasi resmi Kemendagri, dan platform blockchain.	Teknologi blockchain dapat meningkatkan desentralisasi, transparansi, dan ketahanan terhadap serangan siber dalam pengelolaan database penduduk Kemendagri.
(Addiani, 2023)	Mengatasi berbagai risiko dalam sistem penyimpanan data berbasis *cloud* pada organisasi pemerintah melalui penerapan teknologi blockchain yang lebih aman dan terdesentralisasi.	Studi literatur untuk mengidentifikasi faktor risiko pada *cloud storage* dan menganalisis potensi solusi menggunakan karakteristik teknologi blockchain.	Teknologi blockchain dapat meningkatkan keamanan, integritas, dan efisiensi sistem penyimpanan data pemerintah dengan mengurangi risiko kehilangan data, akses tidak sah, dan manipulasi informasi.
(Faber et al., 2019)	Merancang sistem manajemen data pribadi dan identitas berbasis blockchain yang berfokus pada pengguna dan sesuai	*Design science* untuk mengembangkan desain konseptual sistem BPDIMS	Rancangan arsitektur sistem BPDIMS yang memungkinkan pengguna memiliki

	dengan regulasi GDPR.	sebagai artefak awal yang akan divalidasi dan diiterasi lebih lanjut.	transparansi dan kontrol penuh atas data pribadi mereka melalui teknologi blockchain dan *smart contract*.
(Argento et al., 2020)	Merancang dan mengimplementasikan platform berbasis blockchain yang mendukung akuntabilitas layanan dengan integrasi identitas digital publik yang sesuai dengan regulasi eIDAS.	Pengembangan sistem multi-layer yang mencakup modifikasi blockchain Ethereum, penerapan *smart contract*, dan simulasi berbasis agen untuk menilai performa dan validitas proses lintas organisasi.	Platform yang dikembangkan berhasil diuji dalam skenario nyata produksi biomassa, menunjukkan efektivitas dalam mencatat dan menjamin akuntabilitas antar organisasi melalui identitas digital yang terverifikasi dan jejak transaksi yang tidak dapat diubah.

Berdasarkan hasil kajian terhadap tujuh penelitian terdahulu yang relevan, dapat disimpulkan bahwa penerapan teknologi blockchain dalam pengelolaan data kependudukan dan layanan publik telah menunjukkan potensi besar dalam meningkatkan transparansi, keamanan, serta efisiensi proses administrasi. Penelitian-penelitian seperti Nugroho et al. (2023), Rasheed dan Louca (2024), serta Saian et al. (2024) menegaskan bahwa sistem berbasis *smart contract* dan aplikasi terdesentraliasi (dApp) mampu mengotomatiskan proses pendaftaran dan verifikasi data warga dengan tingkat keamanan yang tinggi. Sementara itu, studi oleh Wardhana (2024) dan Addiani (2023) menyoroti keunggulan blockchain dalam menjaga integritas dan keamanan *database* kependudukan pada lembaga pemerintahan. Penelitian dari Faber et al. (2019) dan Argento et al. (2020) memperkuat aspek identitas digital serta akuntabilitas layanan publik yang menjadi fondasi penting dalam sistem administrasi kependudukan modern.

Meskipun berbagai penelitian tersebut telah memberikan kontribusi yang signifikan, sebagian besar kajian masih berfokus pada tahap konseptual, pengujian awal, atau

implementasi pada skala nasional, serta belum banyak mengulas penerapan dan karakteristik teknis arsitektur terdesentralisasi pada layanan publik di tingkat kalurahan. Selain itu, aspek integrasi penyimpanan dokumen kependudukan secara terdistribusi sebagai bagian dari arsitektur sistem juga masih relatif terbatas dibahas secara teknis. Oleh karena itu, penelitian ini diarahkan untuk mengeksplorasi penerapan teknologi blockchain, *smart contract*, dan *InterPlanetary File System* (IPFS) dalam konteks pengelolaan data kependudukan di tingkat kalurahan, guna mengkaji potensi, karakteristik teknis, serta implikasi penerapannya sebagai pendekatan alternatif dalam layanan administrasi kependudukan.

BAB III
METODOLOGI PENELITIAN

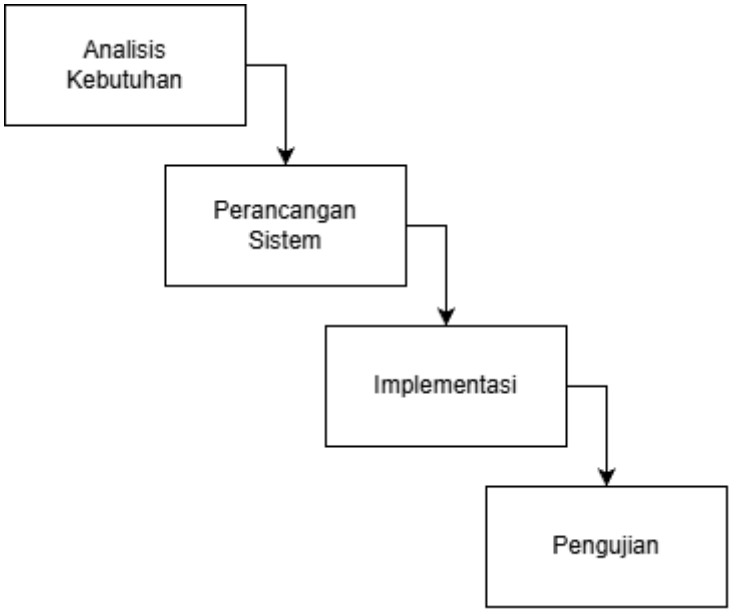
Metodologi yang digunakan dalam penelitian ini adalah pendekatan rekayasa perangkat lunak dengan tahapan yang terstruktur, meliputi analisis kebutuhan, perancangan arsitektur sistem, implementasi, hingga pengujian. Setiap tahap dirancang untuk mengeksplorasi karakteristik, mekanisme kerja, serta implikasi teknis dari penerapan arsitektur terdesentralisasi pada pengelolaan data kependudukan di tingkat kalurahan. Melalui metodologi ini, penelitian diarahkan untuk menghasilkan temuan eksploratif mengenai bagaimana teknologi blockchain beserta ekosistem pendukungnya dapat diterapkan, dievaluasi, dan dianalisis performanya dalam suatu studi kasus layanan publik, sehingga pengembangan purwarupa sistem berfungsi sebagai media pengujian dan evaluasi teknologi.

Dalam pengembangan perangkat lunaknya, penelitian ini menggunakan model pengembangan *Waterfall*. Model ini dipilih karena bersifat sistematis dan terstruktur, dengan tahapan yang jelas dari analisis hingga pengujian. Menurut Murdiani dan Sobirin (2022), model Waterfall sesuai diterapkan pada sistem yang kebutuhan dan spesifikasinya telah didefinisikan sejak awal. Kondisi tersebut selaras dengan penelitian ini, di mana kebutuhan sistem diperoleh melalui hasil studi lapangan serta analisis regulasi administrasi kependudukan di Indonesia. Selain itu, model *Waterfall* juga dinilai sesuai untuk proyek penelitian akademik yang menuntut dokumentasi teratur dan lengkap pada setiap tahap pengembangan. Hal ini sejalan dengan penelitian Thesing et al. (2021) bahwa model *Waterfall* efektif digunakan pada proyek yang membutuhkan dokumentasi sistematis dan pengujian bertahap.

Adapun tahapan dalam model *Waterfall* yang diterapkan pada penelitian ini meliputi:

a. Analisis kebutuhan, untuk mengidentifikasi aktor, proses bisnis, dan kebutuhan sistem;
b. Perancangan sistem, mencakup perancangan proses bisnis, arsitektur sistem, struktur data, *smart contract*, dan mekanisme keamanan serta enkripsi;
c. Implementasi, berupa pengembangan *smart contract*, integrasi dengan IPFS, serta pengembangan *frontend*;
d. Pengujian, yang difokuskan pada pengujian teknis *smart contract*, pengujian fungsional terbatas, serta evaluasi performa dan karakteristik sistem terdesentralisasi yang dihasilkan.

Apabila divisualisasikan maka alur tahapan model *Waterfall* yang digunakan pada penelitian ini akan tampak seperti Gambar 3.1, di mana setiap tahap dilaksanakan secara berurutan, sehingga hasil dari satu tahap menjadi dasar bagi tahap berikutnya.



Gambar 3.1 Diagram Alur Penelitian

3.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk memahami karakteristik proses bisnis yang berjalan pada pelayanan kependudukan di tingkat kalurahan sebagai konteks studi kasus, serta mengidentifikasi aspek-aspek teknis yang relevan untuk mengeksplorasi penerapan arsitektur terdesentralisasi berbasis blockchain. Tahap ini bertujuan untuk memetakan alur layanan yang ada guna mengkaji kesesuaian dan potensi integrasi teknologi blockchain, *smart contract*, dan IPFS dalam mendukung proses tersebut.

Proses analisis dilakukan melalui observasi langsung dan wawancara dengan perangkat kalurahan sebagai penyelenggara pelayanan pencatatan sipil di tingkat terdekat dengan masyarakat, serta studi terhadap regulasi administrasi kependudukan seperti Undang-Undang Nomor 23 Tahun 2006 dan Undang-Undang Nomor 24 Tahun 2013. Selain itu, dilakukan studi literatur terhadap penelitian sebelumnya yang membahas integrasi blockchain dalam tata kelola data publik untuk memperoleh acuan teknis dan konseptual. Penelusuran informasi digital dan konsultasi berbasis kecerdasan buatan juga dilakukan untuk memperdalam pemahaman mengenai prosedur administrasi kependudukan, memverifikasi kesesuaian alur dengan regulasi yang berlaku, serta menelaah potensi pendekatan alternatif dengan menggunakan teknologi blockchain dan IPFS dalam konteks pelayanan publik di Indonesia.

Berdasarkan hasil analisis tersebut didapatkan bahwa pencatatan peristiwa kependudukan seperti kelahiran, kematian, perkawinan, perceraian, dan pindah di tingkat

kalurahan masih dilakukan secara semi-manual dan terpusat melalui aplikasi mitra Dukcapil. Permohonan diajukan warga secara luring dengan datang langsung ke kantor kalurahan, mengisi formulir dan melengkapi dokumen yang diperlukan, kemudian diverifikasi oleh pertugas kalurahan, lalu diteruskan ke Dukcapil untuk diproses dan diterbitkan dokumen hasil layanannya. Alur tersebut apabila dilihat dari sudut pandang teknis masih didominasi oleh verifikasi manual dan ketergantungan pada sistem yang bersifat terpusat. Kondisi ini menjadi konteks yang relevan untuk mengeksplorasi bagaimana teknologi blockchain, *smart contract*, dan penyimpanan terdistribusi dapat diterapkan sebagai pendekatan alternatif, khususnya dalam mendukung pencatatan transaksi permohonan, pelacakan status permohonan, serta pengelolaan dokumen kependudukan secara lebih terstruktur dan dapat diaudit.

Berdasarkan kondisi tersebut, dapat diidentifikasi sejumlah aspek teknis yang menjadi perhatian dalam pengelolaan administrasi kependudukan di tingkat kalurahan dan berpotensi untuk dikaji lebih lanjut melalui pendekatan teknologi terdesentralisasi, antara lain:

a. Proses pelayanan memakan banyak waktu karena pengajuan permohonan harus dilakukan secara langsung di kantor kalurahan dan verifikasi juga masih dilakukan manual.
b. Tidak terdapat mekanisme pelacakan dan transparansi status permohonan antar pihak.
c. Ketergantungan pada arsitektur sistem yang bersifat terpusat, yang secara konseptual memiliki risiko kegagalan terpusat (*single point of failure*).
d. Potensi manipulasi data masih tinggi karena proses manual dan kurangnya sistem audit.
e. Dokumen masih dikirim secara luring tanpa enkripsi, sehingga rawan kebocoran data.

Untuk mengatasi permasalahan tersebut, sistem yang dikembangkan perlu memenuhi beberapa kebutuhan seperti yang dipetakan pada Tabel 3.1.

Tabel 3.1 Kebutuhan Sistem yang Diperlukan

No	Permasalahan Utama	Kebutuhan Eksplorasi Teknologi
1	Pengajuan harus dilakukan secara langsung di kalurahan dan verifikasi juga masih dilakukan secara manual.	Diperlukan eksplorasi mekanisme pengajuan dan verifikasi permohonan berbasis blockchain untuk mengkaji bagaimana proses dapat direpresentasikan sebagai transaksi digital yang tercatat secara terdistribusi.
2	Tidak adanya pelacakan status permohonan	Diperlukan eksplorasi pencatatan status permohonan pada blockchain untuk menguji

		keterlacakan dan transparansi status transaksi secara *real-time*.
3	Ketergantungan pada arsitektur sistem yang bersifat terpusat.	Diperlukan eksplorasi arsitektur terdesentralisasi berbasis blockchain untuk mengkaji dampaknya terhadap integritas data dan risiko kegagalan terpusat (*single point of failure*).
4	Potensi manipulasi data masih tinggi.	Diperlukan eksplorasi penggunaan *smart contract* sebagai mekanisme pencatatan yang bersifat *immutable* serta penerapan *multi-role verification* untuk menguji ketahanan data terhadap manipulasi.
5	Dokumen dikirim secara luring dan tidak terenkripsi	Diperlukan eksplorasi integrasi IPFS sebagai media penyimpanan *off-chain* untuk mengkaji efisiensi, keamanan, dan ketahanan distribusi dokumen kependudukan.

Berdasarkan hasil analisis kebutuhan, diperoleh tiga aktor utama yang terlibat dalam proses pelayanan kependudukan, yaitu Warga yang berperan sebagai Pemohon, Petugas Kalurahan, dan Petugas Dukcapil. Ketiga aktor ini saling berinteraksi dalam proses pengajuan, verifikasi, dan penerbitan dokumen kependudukan sebagaimana dijelaskan pada Tabel 3.2 berikut.

Tabel 3.2 Aktor Utama

No	Aktor	Peran Utama
1	Pemohon	Mengajukan permohonan pencatatan peristiwa kependudukan serta melengkapi dokumen yang diperlukan.
2	Petugas Kalurahan	Melakukan verifikasi awal atas kelengkapan dan keabsahan permohonan dan dokumen.
3	Petugas Dukcapil	Melakukan validasi akhir dan menerbitkan dokumen hasil layanan.

Berdasarkan pemetaan permasalahan, kebutuhan teknis, serta aktor yang terlibat dalam proses administrasi kependudukan, dapat disimpulkan bahwa konteks layanan di tingkat kalurahan menyediakan ruang yang relevan untuk mengeksplorasi penerapan arsitektur terdesentralisasi. Secara khusus, diperlukan pendekatan teknis yang mampu memfasilitasi interaksi antar pihak melalui mekanisme pencatatan transaksi yang transparan, dapat ditelusuri,

dan memiliki jaminan integritas data. Hasil analisis kebutuhan ini selanjutnya digunakan sebagai landasan perancangan purwarupa dan arsitektur sistem berbasis blockchain yang dibahas pada subbab berikutnya.

3.2 Perancangan Sistem

Berdasarkan hasil analisis kebutuhan, tahap perancangan difokuskan pada penyusunan arsitektur dan mekanisme teknis untuk mengeksplorasi penerapan teknologi blockchain, *smart contract*, dan IPFS dalam konteks administrasi kependudukan di tingkat kalurahan. Perancangan ini bertujuan untuk mengkaji bagaimana proses administrasi kependudukan dapat direpresentasikan sebagai rangkaian transaksi terdesentralisasi, bagaimana transparansi dan keterlacakan proses dapat diwujudkan melalui pencatatan pada blockchain, serta bagaimana keamanan dan pengelolaan dokumen dapat didukung melalui penyimpanan *off-chain* berbasis IPFS.

Tahap perancangan mencakup beberapa aspek utama, yaitu perancangan alur proses bisnis sebagai representasi transaksi digital, perancangan arsitektur sistem dan komponennya dalam lingkungan terdesentralisasi, perancangan struktur data *on-chain* dan *off-chain*, perancangan *smart contract*, serta perancangan mekanisme keamanan dan enkripsi dokumen. Hasil perancangan ini digunakan sebagai dasar implementasi purwarupa teknologi yang selanjutnya dianalisis untuk mengevaluasi karakteristik dan performa teknis dari pendekatan terdesentralisasi yang diusulkan.

3.2.1 Proses Bisnis

Sejalan dengan analisis kebutuhan, perancangan proses bisnis difokuskan untuk mengeksplorasi bagaimana alur pelayanan administrasi kependudukan di tingkat kalurahan dapat direpresentasikan dalam bentuk transaksi terdesentralisasi menggunakan teknologi blockchain dan IPFS. Perancangan proses bisnis ini bertujuan untuk mengkaji kemungkinan pemetaan tahapan pelayanan yang semula bersifat semi-manual dan terpusat ke dalam mekanisme pencatatan digital yang transparan, dapat ditelusuri, dan memiliki jaminan integritas data.

Dalam rancangan proses bisnis yang dieksplorasi, interaksi antaraktor utama, yaitu warga, petugas kalurahan, dan petugas direpresentasikan sebagai rangkaian aktivitas yang tercatat melalui *smart contract* pada jaringan blockchain. Setiap pengajuan dan perubahan status permohonan layanan kependudukan, seperti kelahiran, kematian, perkawinan,

perceraian, dan perpindahan penduduk, dicatat sebagai transaksi yang bersifat *immutable*. Sementara itu, dokumen pendukung dan dokumen hasil layanan disimpan secara terenkripsi pada IPFS sebagai penyimpanan *off-chain*, sehingga memungkinkan pengelolaan data yang efisien tanpa membebani jaringan blockchain. Dengan pendekatan ini, proses bisnis yang dirancang menjadi sarana untuk mengevaluasi transparansi, keterlacakan, serta keamanan data dalam arsitektur layanan administrasi kependudukan yang terdesentralisasi.

***Business Process Model and Notation* (BPMN)**

Berdasarkan hasil observasi lapangan dan wawancara dengan petugas kalurahan, proses pelayanan administrasi kependudukan di tingkat kalurahan saat ini masih dilaksanakan melalui mekanisme semi-manual dengan dukungan aplikasi mitra Dukcapil yang bersifat terpusat. Kondisi ini menjadi dasar untuk memetakan alur pelayanan yang berjalan ke dalam model proses bisnis sebagai bahan eksplorasi penerapan teknologi blockchain. Berdasarkan temuan tersebut, dapat diidentifikasi dua pola utama alur proses pelayanan, yaitu alur permohonan umum dan alur permohonan pindah.

Alur permohonan umum mencakup layanan pencatatan peristiwa seperti kelahiran, kematian, perkawinan, dan perceraian. Keempat jenis layanan tersebut memiliki mekanisme yang relatif serupa. Proses dimulai ketika warga atau pemohon datang langsung ke kantor kalurahan untuk mengisi formulir sesuai jenis permohonan yang diajukan dan melampirkan dokumen persyaratan. Selanjutnya, petugas kalurahan memverifikasi kelengkapan dan keabsahan dokumen, lalu meneruskan berkas permohonan ke Dukcapil melalui aplikasi Mitra Dukcapil. Petugas Dukcapil kemudian melakukan validasi akhir dan menerbitkan dokumen hasil layanan (seperti akta atau surat keterangan), yang kemudian dikirim kembali ke kalurahan untuk diserahkan kepada pemohon.

Sementara itu, alur permohonan pindah memiliki karakteristik yang lebih kompleks karena melibatkan lebih dari satu wilayah administratif. Proses diawali oleh warga yang datang ke kalurahan asal untuk mengisi formulir permohonan pindah dan melengkapi dokumen pendukung. Setelah diverifikasi oleh petugas kalurahan asal, warga akan menerima Surat Keterangan Pindah (SKP) yang harus diserahkan ke kalurahan tujuan. Di kalurahan tujuan, petugas melakukan verifikasi ulang atas SKP tersebut sebelum melanjutkan proses ke Dukcapil di wilayah tujuan. Petugas Dukcapil kemudian melakukan validasi dan menerbitkan Kartu Keluarga (KK) baru yang dapat diambil oleh pemohon di kalurahan tujuan.

Berdasarkan perbedaan karakteristik layanan tersebut, penelitian ini memetakan proses administrasi kependudukan ke dalam dua kategori alur, yaitu alur permohonan umum dan alur permohonan pindah. Pemisahan alur ini dilakukan untuk mengeksplorasi bagaimana perbedaan kompleksitas layanan dapat direpresentasikan dalam logika *smart contract* dan mekanisme verifikasi terdesentralisasi. Dengan pemetaan tersebut, rancangan BPMN digunakan sebagai media untuk menggambarkan variasi proses bisnis dan alur validasi pada masing-masing kategori permohonan secara lebih terstruktur dan teknis.

a. Permohonan Umum

Berdasarkan permasalahan yang ditemukan pada proses saat ini, rancangan alur proses layanan kependudukan umum disusun sebagai media eksplorasi penerapan arsitektur terdesentralisasi berbasis blockchain dan IPFS. Alur ini mencakup empat jenis layanan, yaitu kelahiran, kematian, perkawinan, dan perceraian, yang secara umum memiliki mekanisme pengajuan dan verifikasi yang sama. Rancangan alur proses ini digunakan untuk mengeksplorasi bagaimana mekanisme pencatatan terdistribusi, verifikasi berbasis s*mart contract*, serta integrasi penyimpanan *off-chain* dapat merepresentasikan proses administrasi kependudukan, khususnya dalam konteks pengurangan ketergantungan pada tahapan manual, konsistensi verifikasi dokumen, dan keterlacakan status permohonan.

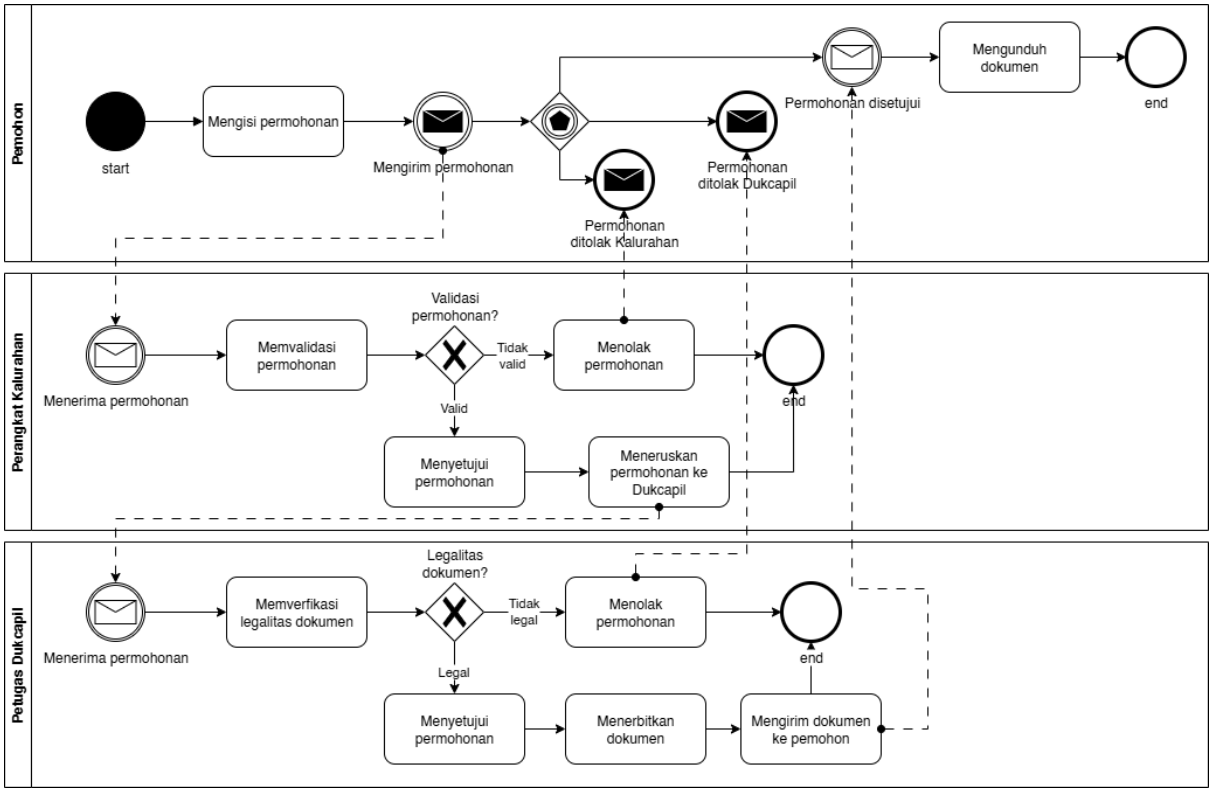
Gambar 3.2 menampilkan diagram BPMN rancangan alur permohonan umum pada model sistem terdesentralisasi. Dalam rancangan ini, interaksi utama terjadi antara tiga aktor yaitu warga, petugas kalurahan, dan petugas Dukcapil. Alur dimulai dari warga yang mengajukan permohonan melalui arsitektur terdesentralisasi dengan mengisi formulir digital dan mengunggah dokumen pendukung sesuai jenis layanan. Setelah data diunggah, *smart contract* secara otomatis mencatat transaksi permohonan ke blockchain, menghasilkan ID unik untuk setiap permohonan.

Selanjutnya, permohonan akan masuk ke petugas kalurahan melalui antarmuka yang telah dirancang sebagai media pengelolaan bagi petugas kalurahan. Petugas melakukan pemeriksaan kelengkapan dokumen dan kebenaran data. Apabila dokumen dinyatakan tidak lengkap, status permohonan akan diperbarui menjadi “Ditolak oleh Kalurahan”, dan pemohon dapat memperbaikinya. Namun jika dinyatakan lengkap, permohonan diteruskan ke petugas Dukcapil dengan status “Diverifikasi Kalurahan”. Blockchain akan merekam proses tersebut, sehingga setiap perubahan status dapat dilacak dengan jelas.

Setelah diterima oleh Dukcapil, petugas melakukan validasi akhir terhadap data permohonan. Bila dinyatakan sah, maka petugas akan menerbitkan dokumen hasil layanan

(misalnya Akta Kelahiran atau Surat Keterangan Kematian) yang kemudian dienkripsi menggunakan algoritma AES-256-CBC (*Advanced Encryption Standard*, 256-bit *key*, *Cipher Block Chaining*) dan diunggah ke jaringan IPFS. Hasil unggahan berupa CID disimpan di blockchain sebagai bukti keaslian dan referensi dokumen digital.

Warga kemudian dapat mengunduh dokumen hasil layanan melalui aplikasi dengan memverifikasi identitas *wallet*-nya. Dokumen dapat diakses kapan saja karena tersimpan secara permanen di jaringan IPFS, sementara keaslian dan histori transaksi dapat diverifikasi melalui catatan blockchain. Dengan mekanisme ini, penerapan *smart contract* dan pencatatan berbasis blockchain memungkinkan proses pengajuan dan penerbitan dokumen direpresentasikan secara lebih transparan, konsisten, dan memiliki ketahanan terhadap manipulasi.



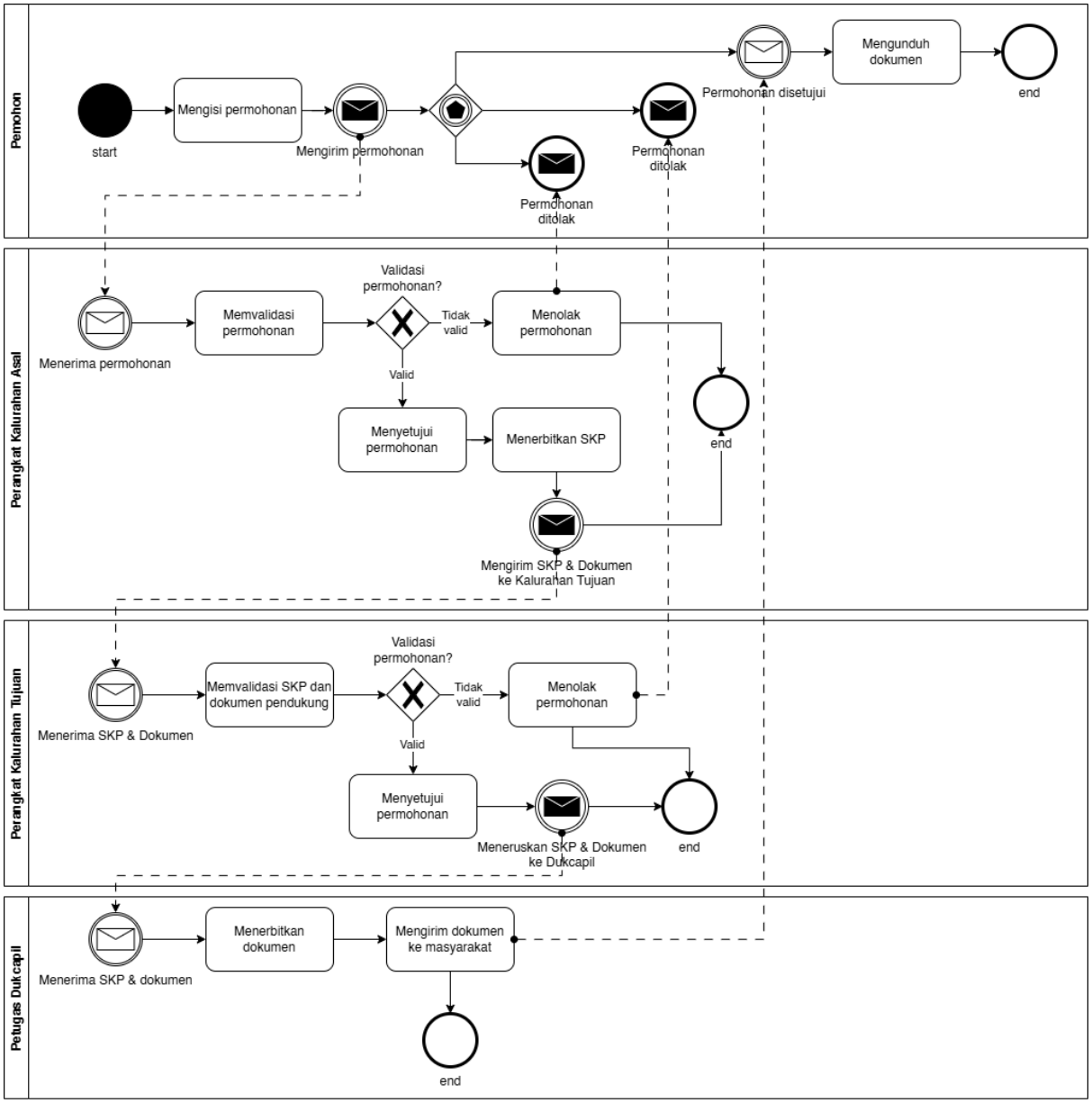
Gambar 3.2 BPMN Permohonan Umum

Rancangan BPMN pada Gambar 3.2 menunjukkan bahwa penerapan *smart contract* dan blockchain menciptakan proses yang sederhana dan memperkenalkan mekanisme pencatatan transaksi otomatis. Setiap tahap mulai dari pengajuan, verifikasi, hingga penerbitan dokumen menghasilkan jejak digital yang bersifat *immutable*, sehingga menciptakan proses yang

akuntabel dan transparan antar pihak. Selain itu, penyimpanan dokumen di IPFS memberikan akses dokumen yang lebih aman dan tahan manipulasi.

b. Permohonan Pindah

Tidak seperti layanan kependudukan umum, proses permohonan pindah memiliki karakteristik yang lebih kompleks karena melibatkan lebih dari satu wilayah administrasi, yakni kalurahan asal dan kalurahan tujuan. Berdasarkan hasil observasi pada sistem saat ini, proses ini umumnya dilakukan secara berlapis dan bergantung pada proses manual, seperti pengisian formulir, penerbitan Surat Keterangan Pindah (SKP), serta koordinasi antar wilayah yang dilakukan dengan konvensional. Kondisi tersebut menjadi salah satu aspek yang relevan untuk dikaji, khususnya terkait potensi keterlambatan proses, duplikasi data, serta konsistensi pencatatan akibat mekanisme sinkronisasi data antar kalurahan.



Gambar 3.3 BPMN Permohonan Pindah

Sebagai bentuk eksplorasi, penelitian ini memanfaatkan *smart contract* sebagai pengelola status permohonan secara otomatis dan IPFS sebagai media penyimpanan dokumen digital terenkripsi. Rancangan BPMN alur permohonan pindah ditunjukkan pada Gambar 3.3. Alur ini dirancang agar mampu mendukung tiga jenis skenario pindah, yaitu:

1. Pindah satu keluarga, di mana seluruh anggota keluarga berpindah ke wilayah baru;
2. Pindah individu, di mana salah satu anggota keluarga berpindah secara mandiri; dan
3. Pindah bergabung keluarga, di mana seseorang berpindah ke alamat keluarga lain yang sudah terdaftar.

Setiap jenis pindah mengikuti pola tahapan yang sama namun dengan variasi data dan verifikasi di bagian akhir proses. Proses dimulai dari warga sebagai pemohon yang mengajukan permohonan pindah melalui aplikasi dengan memilih jenis pindah yang sesuai dan melengkapi data serta dokumen pendukung. *Smart contract* kemudian mencatat pengajuan ke blockchain dan menghasilkan ID permohonan unik.

Petugas kalurahan asal menerima permohonan baru melalui aplikasi dan melakukan pemeriksaan terhadap kelengkapan dokumen serta keabsahan alasan pindah. Apabila dokumen tidak lengkap, permohonan dapat ditolak; apabila valid, permohonan diteruskan ke kalurahan tujuan dengan status "Diverifikasi Kalurahan Asal". Proses pengiriman ini dicatat sebagai transaksi baru di blockchain untuk menjamin keaslian data antar wilayah.

Setelah diterima oleh kalurahan tujuan, petugas memverifikasi kembali data dan dokumen pemohon, terutama kesesuaian alamat tujuan dan relasi keluarga (bila berlaku). Hasil verifikasi ini menentukan apakah permohonan dapat diteruskan ke Dukcapil wilayah tujuan. Bila disetujui, status diperbarui menjadi "Diverifikasi Kalurahan Tujuan" dan diteruskan ke petugas Dukcapil.

Petugas Dukcapil kemudian melakukan validasi akhir terhadap data kependudukan pemohon dan menerbitkan dokumen hasil layanan seperti Kartu Keluarga (KK) baru. Dokumen digital tersebut selanjutnya dienkripsi, diunggah ke IPFS, dan CID-nya disimpan ke blockchain melalui *smart contract*.

Warga dapat mengunduh dokumen hasil layanan melalui aplikasi dengan memverifikasi *wallet* yang digunakan saat pengajuan. Karena seluruh transaksi dicatat di blockchain, setiap langkah verifikasi dan penerbitan dokumen dapat diaudit secara transparan oleh pihak berwenang tanpa risiko manipulasi data.

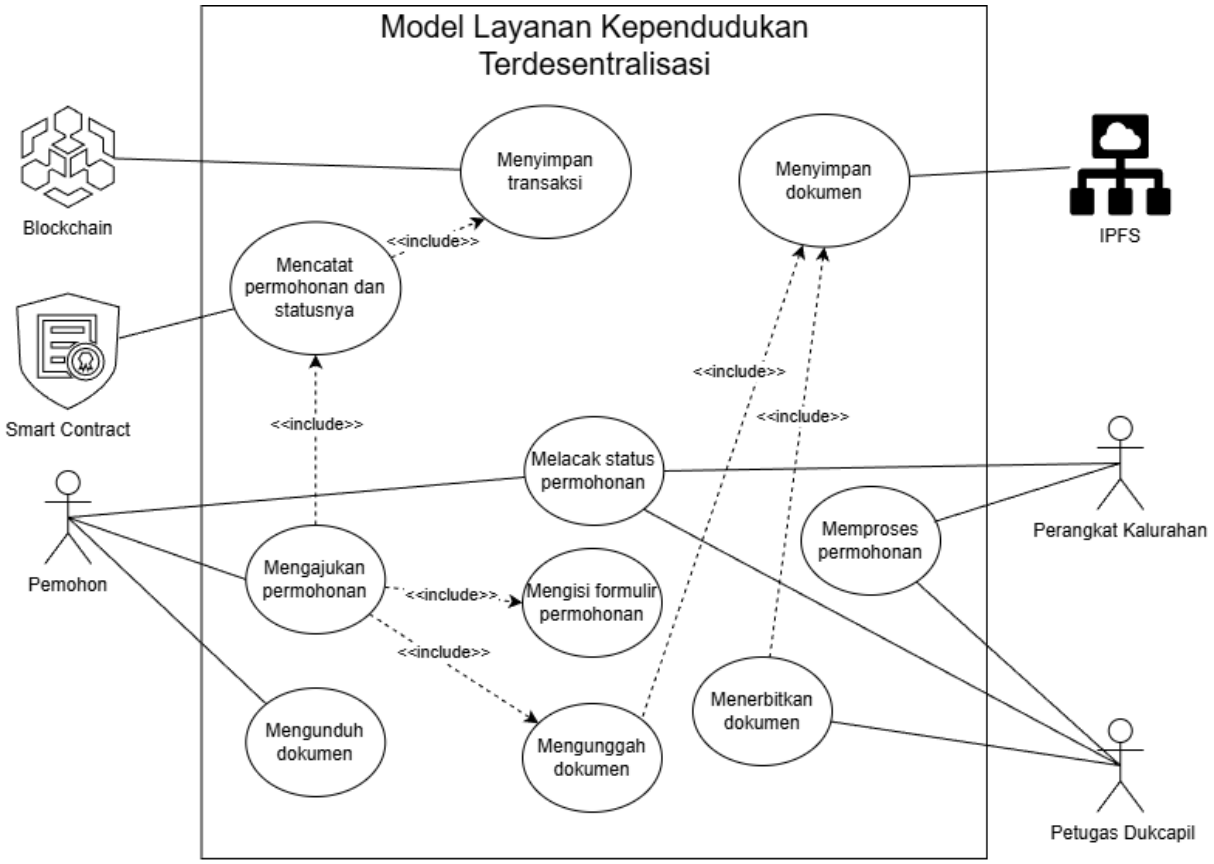
Rancangan BPMN alur permohonan pindah pada Gambar 3.3 menunjukkan penerapan mekanisme *multi-layer verification* antara kalurahan asal, kalurahan tujuan, dan Dukcapil. Pendekatan ini menjadikan setiap proses verifikasi tercatat secara permanen di blockchain. Integrasi dengan IPFS memastikan bahwa dokumen seperti SKP dan KK baru tersimpan secara terenkripsi, tahan manipulasi, dan mudah diverifikasi keasliannya melalui CID yang unik.

Dengan rancangan ini, proses perpindahan penduduk direpresentasikan dalam mekanisme pencatatan terdesentralisasi yang memungkinkan keterlacakan status permohonan secara transparan, serta mengkaji potensi peningkatan konsistensi data antar wilayah.

***Use Case Diagram* (UCD)**

Setelah dilakukan pemodelan proses bisnis melalui diagram BPMN, tahap selanjutnya adalah memetakan interaksi konseptual antara aktor dan komponen teknologi yang dieksplorasi dalam penelitian ini melalui *use case diagram* sebagaimana ditunjukkan pada Gambar 3.4. *Use case diagram* digunakan untuk menggambarkan bagaimana aktor manusia berinteraksi dengan mekanisme yang disediakan oleh teknologi blockchain, *smart contract*, dan IPFS dalam konteks layanan administrasi kependudukan.

Perancangan *use case diagram* pada penelitian ini tidak hanya berfokus pada hubungan pengguna dengan arsitektur terdesentralisasi, tetapi juga merepresentasikan peran teknologi sebagai entitas fungsional dalam proses pencatatan, verifikasi, dan penyimpanan data. Setiap *use case* menggambarkan aktivitas yang melibatkan aktor manusia dan teknologi terdesentralisasi, sehingga diagram ini berfungsi sebagai alat untuk mengeksplorasi pembagian peran antara pengguna dan teknologi sebelum tahap implementasi dilakukan.



Gambar 3.4 *Use Case Diagram*

a. Identifikasi Aktor

Aktor yang terlibat dirangkum pada Tabel 3.3 berikut:

Tabel 3.3 Identifikasi Aktor

No	Aktor	Deskripsi Peran
1	Blockchain	Infrastruktur pencatatan terdesentralisasi yang digunakan untuk mencatat dan menyimpan transaksi permohonan serta perubahan statusnya secara permanen dan tidak dapat dimodifikasi. Blockchain berperan sebagai media pencatatan yang menjamin transparansi, integritas data, dan keterlacakan proses layanan kependudukan.
2	*Smart Contract*	Mekanisme logika terotomatisasi yang berjalan di atas blockchain untuk mengelola alur permohonan, pencatatan status, serta pembatasan hak akses berdasarkan peran aktor. *Smart contract* digunakan sebagai sarana eksplorasi penerapan aturan bisnis layanan kependudukan secara terdesentralisasi dan transparan.
3	IPFS	Sistem penyimpanan berkas terdistribusi yang digunakan sebagai media penyimpanan dokumen kependudukan secara *off-chain*. IPFS berperan dalam menjaga efisiensi penyimpanan, integritas berkas, serta ketahanan data.
4	Pemohon	Individu yang mengajukan permohonan layanan kependudukan seperti kelahiran, kematian, perkawinan, perceraian, dan pindah. Pemohon juga dapat memantau status permohonan dan mengunduh dokumen hasil layanan.
5	Petugas Kalurahan	Verifikator tingkat pertama yang memeriksa kelengkapan dan keabsahan dokumen permohonan. Petugas dapat menyetujui atau menolak permohonan sebelum diteruskan ke Dukcapil.
6	Petugas Dukcapil	Pihak yang melakukan validasi akhir dan menerbitkan dokumen hasil layanan dalam bentuk digital terenkripsi.

b. Identifikasi *Use Case*

Use case yang terdapat dalam sistem diuraikan pada Tabel 3.4 berikut:

Tabel 3.4 Identifikasi *Use Case*

No	*Use Case*	Deskripsi	Aktor Terkait
1	Menyimpan transaksi	Setiap permohonan yang diajukan oleh Pemohon serta setiap perubahan status yang terjadi selama proses verifikasi akan dicatat sebagai transaksi oleh *smart contract* dan disimpan secara permanen pada blockchain. Pencatatan ini memungkinkan transaksi layanan kependudukan terdokumentasi secara transparan dan tidak dapat dimodifikasi.	Blockchain
2	Mencatat permohonan dan statusnya	Setiap permohonan beserta statusnya akan dicatat oleh s*mart contract* secara otomatis di blockchain sejak diajukan hingga proses verifikasi selesai. Setiap perubahan status akan terekam sebagai bagian dari alur permohonan.	*Smart contract*
3	Menyimpan dokumen	Dokumen persyaratan yang diunggah oleh Pemohon maupun dokumen hasil layanan yang diterbitkan oleh Dukcapil akan disimpan oleh IPFS dalam bentuk terenkripsi. Penyimpanan dilakukan secara terdistribusi untuk menjaga integritas dan ketahanan data.	IPFS
4	Mengajukan permohonan	Mengajukan permohonan layanan kependudukan, melengkapi formulir digital, dan mengunggah dokumen pendukung.	Pemohon
5	Mengunduh dokumen	Setelah permohonan disetujui dan dokumen hasil layanan diterbitkan oleh Dukcapil, maka dokumen tersebut dapat diakses oleh Pemohon.	Pemohon
6	Melacak status permohonan	Melihat status terkini dari sebuah permohonan secara *real-time*.	Pemohon, Petugas Kalurahan, Petugas Dukcapil
7	Memproses permohonan	Proses pengelolaan permohonan oleh kalurahan meliputi validasi permohonan dan kelengkapan dokumen untuk diteruskan ke Dukcapil, sementara Dukcapil setelah menerima permohonan dari Kalurahan maka akan	Petugas Kalurahan dan Dukcapil

		memproses permohonan dengan melakukan verifikasi permohonan dan dokumen untuk kemudian diterbitkan dokumen sesuai layanan.	
8	Menerbitkan dokumen	Setelah verifikasi disetujui maka dokumen baru sesuai layanan akan diterbitkan.	Dukcapil

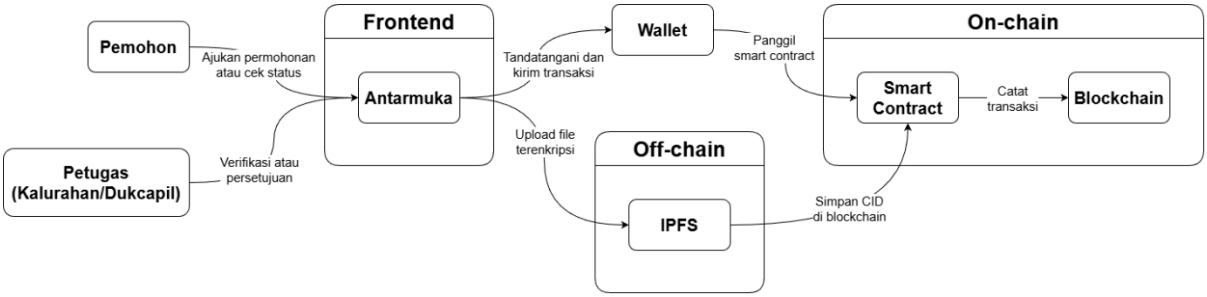
Berdasarkan *use case diagram* pada Gambar 3.4, Komponen teknologi berupa blockchain, *smart contract,* dan IPFS diposisikan sebagai aktor pendukung yang menjalankan fungsi pencatatan transaksi, pengelolaan status permohonan, serta penyimpanan dokumen secara terdistribus. Sementara itu, Pemohon berperan sebagai pengguna utama yang mengajukan permohonan layanan kependudukan, mengunggah dokumen persyaratan, serta memantau status permohonan. Petugas Kalurahan dan Petugas Dukcapil berperan sebagai aktor institusional yang melakukan verifikasi berjenjang hingga penerbitan dokumen.

Hasil perancangan proses bisnis dan *use case diagram* pada tahap ini digunakan sebagai landasan konseptual untuk mengeksplorasi keterkaitan antar komponen teknologi yang terlibat. Berdasarkan pemodelan tersebut, disusun arsitektur sistem yang merepresentasikan integrasi antara aplikasi, *smart contract*, blockchain, dan IPFS sebagai satu kesatuan mekanisme layanan terdesentralisasi. Pembahasan mengenai arsitektur sistem dan peran masing-masing komponen teknologi dijelaskan pada anak subbab berikutnya.

3.2.2 Arsitektur Sistem

Tahap perancangan arsitektur sistem dilakukan untuk menggambarkan struktur komponen utama dan hubungan antar bagian dalam sistem terdesentralisasi pencatatan kependudukan yang digunakan sebagai media eksplorasi. Arsitektur ini dirancang agar mampu mengintegrasikan tiga komponen kunci, yaitu *smart contract* sebagai pengelola logika bisnis dan pencatatan transaksi, IPFS sebagai penyimpanan dokumen terenkripsi secara terdistribusi, serta aplikasi terdesentralisasi sebagai antarmuka pengguna.

Rancangan arsitektur sistem disusun berdasarkan kebutuhan yang telah diidentifikasi pada tahap analisis, di mana sistem harus mampu mengelola proses permohonan kependudukan (lahir, mati, kawin, cerai, pindah) secara efisien, transparan, dan aman. Hubungan antar komponen ditunjukkan pada Gambar 3.5 berikut.



Gambar 3.5 Diagram Arsitektur Sistem

Arsitektur sistem terdiri atas empat komponen utama yang saling terhubung melalui jaringan blockchain dan protokol IPFS. Setiap komponen memiliki peran dan tanggung jawab tersendiri, sebagaimana dijelaskan pada Tabel 3.5.

Tabel 3.5 Deskripsi Umum Arsitektur

No	Komponen	Deskripsi Fungsi
1	*Frontend* dApp	Antarmuka utama yang digunakan oleh pemohon, petugas kalurahan, dan petugas Dukcapil. Aplikasi ini dikembangkan berbasis web dan terhubung ke *smart contract* melalui *library* seperti Web3.js atau Ethers.js. Semua interaksi pengguna (pengajuan, verifikasi, pelacakan status, dan unduh dokumen) dilakukan melalui dApp.
2	*Onchain*	Lapisan yang terdiri atas *smart contract* dan blockchain. *Smart contract* berfungsi sebagai pengatur logika bisnis dan pencatat status permohonan secara otomatis. Setiap transaksi yang terjadi (pengajuan, verifikasi, penerbitan dokumen) dicatat secara permanen di blockchain, menjamin integritas, transparansi, dan ketahanan terhadap manipulasi data.
3	*Offchain*	Lapisan penyimpanan eksternal yang digunakan untuk menyimpan dokumen hasil layanan kependudukan dalam bentuk digital. Dokumen akan terlebih dahulu dienkripsi di sisi pengguna sebelum diunggah ke IPFS. IPFS kemudian menghasilkan CID unik yang disimpan di blockchain melalui *smart contract* sebagai referensi ke dokumen asli.
4	*Wallet*	Komponen autentikasi dan otorisasi transaksi yang digunakan oleh setiap pengguna (pemohon maupun petugas). *Wallet* seperti MetaMask berfungsi untuk menandatangani transaksi secara digital serta memanggil fungsi *smart contract* di jaringan blockchain. Melalui *wallet*, identitas pengguna dapat diverifikasi tanpa harus menyimpan kredensial secara tersentralisasi.

Berdasarkan Tabel 3.5, seluruh komponen saling berinteraksi secara terintegrasi. Lapisan *frontend* bertugas sebagai penghubung antara pengguna dengan sistem, sedangkan *smart contract* di lapisan *onchain* menangani seluruh proses bisnis dan pencatatan transaksi. Lapisan *offchain* berperan sebagai media penyimpanan dokumen terenkripsi, sementara *wallet* digunakan sebagai mekanisme otentikasi dan penandatanganan transaksi secara terdesentralisasi. Dengan demikian, setiap proses yang terjadi di sistem dapat diverifikasi dan ditelusuri tanpa bergantung pada server pusat.

Secara umum, alur interaksi dalam sistem dapat dijelaskan sebagai berikut:

1. Pengguna (Warga, Kalurahan, Dukcapil) berinteraksi melalui dApp untuk mengajukan atau memproses permohonan.
2. dApp mengirimkan data permohonan ke *smart contract* untuk dicatat di blockchain.
3. Jika dokumen diunggah, sistem melakukan enkripsi di sisi klien kemudian mengunggahnya ke IPFS.
4. Hasil unggahan IPFS berupa CID dikembalikan ke aplikasi dan disimpan dalam blockchain agar dapat diverifikasi keasliannya.
5. Pengguna dapat melakukan pelacakan status permohonan dengan membaca data dari blockchain menggunakan fungsi *getter* dalam *smart contract*.
6. Seluruh proses transaksi dilakukan melalui *wallet* yang menangani penandatanganan digital dan pembayaran *gas fee* (jika diperlukan).

Untuk memudahkan pemahaman, sistem ini dapat dipandang sebagai tiga lapisan utama:

1. Lapisan Presentasi (*Frontend Layer*)

   Menyediakan antarmuka pengguna berbasis web (dApp). Lapisan ini menjadi titik interaksi antara pengguna dan sistem, serta berfungsi mengelola input, menampilkan status permohonan, dan memfasilitasi unduhan dokumen hasil layanan.

2. Lapisan Logika Bisnis (*Smart Contract Layer*)

   Menangani seluruh logika proses bisnis sesuai aturan pelayanan kependudukan, termasuk pendaftaran, pengajuan, verifikasi, validasi, dan penerbitan dokumen. Setiap tindakan direkam sebagai transaksi dalam blockchain.

3. Lapisan Data (*Data Layer*)

   Menyimpan data transaksi secara *onchain* dan dokumen secara *offchain*. Data penting seperti status permohonan dan CID IPFS disimpan di blockchain, sedangkan berkas dokumen terenkripsi disimpan di IPFS.

Integrasi antara blockchain dan IPFS dilakukan untuk menggabungkan keunggulan keduanya: blockchain sebagai pencatat transaksi yang transparan dan tidak dapat diubah, serta IPFS sebagai penyimpanan berkas terenkripsi yang terdistribusi. Proses integrasi ini berjalan dua arah yaitu blockchain menyimpan metadata (seperti CID dan status permohonan), sedangkan IPFS menyimpan dokumen aktual. Dengan demikian, sistem dapat menjaga efisiensi ruang penyimpanan di blockchain tanpa mengorbankan integritas dan keterlacakan dokumen digital.

Arsitektur sistem yang telah dirancang ini menjadi dasar dalam penyusunan struktur data yang digunakan pada tahap selanjutnya. Struktur data dirancang untuk mendukung hubungan antara entitas pada sistem, baik yang disimpan di blockchain maupun di IPFS. Pembahasan mengenai rancangan struktur data dijelaskan pada anak subbab berikutnya.

3.2.3 Struktur Data

Struktur data pada sistem ini dirancang untuk merepresentasikan entitas-entitas utama yang terlibat dalam proses pencatatan kependudukan secara terdesentralisasi. Tujuannya adalah agar data dapat dikelola secara efisien di jaringan blockchain (*onchain*) maupun di penyimpanan terdistribusi (*offchain*). Setiap data yang dicatat di blockchain bersifat permanen (*immutable*), sedangkan data yang berukuran besar seperti dokumen digital disimpan secara terenkripsi di IPFS untuk menghemat ruang dan menjaga privasi.

Struktur data sistem dibagi menjadi dua kategori utama:

a. Data *Onchain*: Data yang disimpan langsung di blockchain melalui *smart contract*, berfungsi untuk mencatat transaksi, status permohonan, dan referensi dokumen.
b. Data *Offchain*: Data yang disimpan di luar blockchain (IPFS) seperti berkas dokumen, data keluarga (KK), dan data permohonan.

Keterkaitan antara keduanya dijembatani melalui CID IPFS yang disimpan di blockchain sebagai referensi unik terhadap data *offchain*.

Data *onchain* direpresentasikan dalam beberapa *struct* dan *enum* pada *smart contract* untuk mendukung proses bisnis sistem. Struktur utama yang digunakan dipetakan ke dalam Tabel 3.6. Struktur tersebut memastikan bahwa setiap transaksi dan perubahan status permohonan dapat dilacak secara historis melalui *event log* di blockchain.

Tabel 3.6 Struktur Data *Onchain*

Nama *Struct*/*Enum*	Deskripsi
struct Permohonan	Merepresentasikan satu pengajuan permohonan layanan kependudukan. Merupakan *struct* utama dengan atribut untuk menyimpan metadata permohonan.
enum JenisPermohonan	Enumerasi jenis permohonan layanan kependudukan, yaitu kelahiran, kematian, perkawinan, perceraian, dan pindah.
enum Status	Enumerasi status permohonan layanan kependudukan, seperti diajukan, disetujui kalurahan, disetujui dukcapil, dan status permohonan lainnya.
enum JenisPindah	Enumerasi jenis permohonan pindah untuk mendukung ketiga jenis perpindahan penduduk yaitu pindah satu keluarga, pindah membuat kk baru, dan pindah dengan bergabung ke kk tujuan.

Berbeda dengan data *onchain*, data *offchain* pada sistem ini mencakup berkas-berkas yang berukuran besar dan bersifat privat, seperti dokumen persyaratan permohonan, data hasil pengisian formulir pengajuan, dokumen hasil layanan yang diterbitkan oleh Dukcapil, serta data setiap entitas keluarga. Dokumen persyaratan dan dokumen hasil layanan akan dienkripsi terlebih dahulu di sisi pengguna sebelum diunggah ke IPFS. Sementara itu, data hasil pengisian formulir dan data keluarga disimpan dalam bentuk berkas JSON terformat agar mudah diintegrasikan dengan sistem *smart contract*.

Penyimpanan data permohonan dan data keluarga dilakukan secara *offchain* karena jika disimpan di *onchain*, ukuran data yang besar akan menyebabkan biaya transaksi (*gas fee*) meningkat secara signifikan. Selain itu, data kependudukan bersifat pribadi dan rahasia, sehingga penyimpanan langsung di blockchain tidak disarankan mengingat sifatnya yang transparan dan dapat diakses publik. Penggunaan IPFS menjadi solusi untuk permasalahan tersebut karena tetap mempertahankan prinsip desentralisasi dan keterlacakan, namun tanpa mengorbankan privasi dan efisiensi penyimpanan.

Setiap entitas data kependudukan disimpan dalam bentuk berkas JSON terenkripsi yang diunggah ke jaringan IPFS. Struktur data ini dirancang untuk dapat mewakili hubungan antaranggota keluarga serta memuat metadata yang diperlukan dalam verifikasi administrasi. Struktur JSON tersebut memiliki empat bagian utama seperti yang dijelaskan pada Tabel 3.7 berikut.

Tabel 3.7 Struktur JSON Data Keluarga

Bagian	Deskripsi
Identitas Kartu Keluarga	Menyimpan nomor KK untuk setiap kartu keluarga.
Alamat	Menjelaskan alamat lengkap tempat tinggal keluarga.
Anggota Keluarga	Berisi daftar anggota keluarga dalam bentuk *array*, masing-masing memuat atribut data pribadi dan relasi keluarga.
Metadata	Menyimpan informasi administratif seperti tanggal unggah dan *log* perubahan KK.

Hubungan antara data *onchain* dan *offchain* bersifat referensial. Blockchain hanya menyimpan *pointer* berupa CID IPFS dan metadata penting seperti jenis dokumen dan status permohonan, sedangkan isi dokumen dan data keluarga tetap berada di IPFS. Dengan mekanisme ini, sistem dapat menjaga keseimbangan antara efisiensi penyimpanan dan keamanan data, sekaligus memastikan integritas data melalui verifikasi CID yang bersifat unik dan tidak dapat diubah.

Struktur data yang telah dijelaskan pada Anak Subbab 3.2.3 menjadi dasar bagi perancangan *smart contract* yang mengimplementasikan logika proses bisnis sistem. *Smart contract* dirancang untuk memanfaatkan struktur dan enumerasi tersebut dalam pengelolaan permohonan, verifikasi, serta dokumen secara otomatis. Rancangan dan fungsi utama dari *smart contract* dijelaskan pada anak subbab berikutnya.

3.2.4 *Smart Contract*

Smart contract diposisikan sebagai komponen kunci dalam eksplorasi arsitektur terdesentralisasi pada penelitian ini, karena berfungsi sebagai representasi penerapan logika bisnis dan aturan pengelolaan data kependudukan yang dieksekusi secara otomatis dan transparan di atas jaringan blockchain. Melalui *smart contract*, berbagai proses seperti pencatatan permohonan, perubahan status verifikasi berjenjang, serta pencatatan referensi dokumen digital berupa *Content Identifier* (CID) dari IPFS dapat direkam secara terdistribusi dan tidak dapat diubah. Penerapan smart contract dalam konteks penelitian ini bertujuan untuk mengeksplorasi bagaimana mekanisme otomasi, keterlacakan, dan integritas data yang ditawarkan oleh blockchain dapat mendukung proses administrasi kependudukan tanpa menghilangkan peran verifikasi manual oleh lembaga resmi. Dengan demikian, smart contract berperan sebagai lapisan kepercayaan digital yang melengkapi mekanisme tata kelola

kependudukan konvensional, sekaligus menjadi objek evaluasi terhadap karakteristik teknis arsitektur terdesentralisasi yang diusulkan.

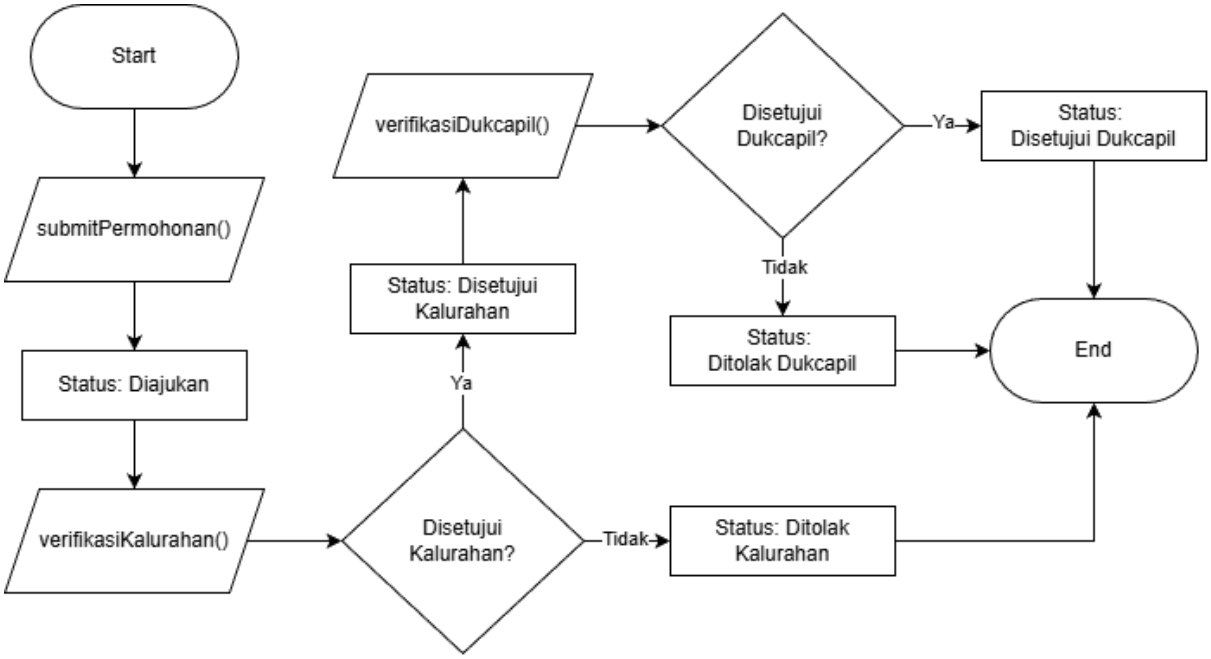
Dengan memanfaatkan *smart contract*, setiap aktivitas pada sistem seperti pengajuan permohonan, verifikasi petugas, maupun penerbitan dokumen dapat dicatat secara permanen (*immutable*), transparan, dan dapat diaudit oleh pihak berwenang tanpa memerlukan perantara.

Rancangan *smart contract* disusun secara modular agar mudah dikembangkan, diuji, dan dipelihara. Kontrak utama bernama PencatatanSipil.sol berfungsi sebagai *entry point* yang mengoordinasikan pemanggilan terhadap beberapa kontrak lain, penjelasan untuk setiap kontrak dipetakan ke dalam Tabel 3.8.

Tabel 3.8 Rancangan *Smart Contract*

Nama Kontrak	**Fungsi Utama**
PencatatanSipil.sol	Mengatur alur pengajuan dan verifikasi permohonan, serta menyimpan CID hasil dokumen.
PencatatanTypes.sol	Menyimpan definisi tipe data, *struct*, dan *enum* yang digunakan secara global.
PermohonanUtils.sol dan PencatatanViewUtils.sol	Merupakan *library* utilitas berisikan fungsi *helper* untuk presentasi (konversi *enum* ke *string*) dan *helper* pemeliharaan indeks.
PermohonanManager.sol	Mengelola pembuatan, pembaruan, dan pelacakan data permohonan.
KontrolAkses.sol	Mengatur hak akses aktor (Warga, Kalurahan, Dukcapil) menggunakan *modifier*.

Struktur modular ini memastikan bahwa setiap kontrak memiliki tanggung jawab spesifik dan dapat diuji secara terpisah menggunakan *framework* Hardhat. Pendekatan ini juga memudahkan proses perawatan kode, mengingat kompleksitas alur kependudukan yang melibatkan banyak aktor.



Gambar 3.6 Diagram Alur Logika *Smart Contract*

Diagram pada Gambar 3.6 menggambarkan urutan pemanggilan fungsi utama dalam aplikasi, di mana setiap langkah menghasilkan transaksi baru di blockchain. Proses dimulai dari warga yang mengajukan permohonan melalui dApp yang akan memanggil fungsi submitPermohonan(), dilanjutkan oleh verifikasi petugas kalurahan dengan memanggil fungsi verifikasiKalurahan(), jika disetujui maka dilanjutkan verifikasi oleh petugas Dukcapil dengan memanggil fungsi verifikasiDukcapil(). *Smart contract* mengelola setiap tahap proses ini dengan memanggil fungsi-fungsi tertentu yang dilindungi oleh mekanisme kontrol akses berbasis *modifier*.

Fungsi-fungsi utama yang diimplementasikan pada kontrak PencatatanSipil.sol beserta perannya dijelaskan pada Tabel 3.9.

Tabel 3.9 Fungsi Utama *Smart Contract*

Fungsi	Deskripsi	Aktor
submitPermohonan()	Mengajukan permohonan layanan kependudukan (lahir, mati, kawin, cerai, atau pindah). Fungsi ini membuat ID baru dan mencatat CID IPFS dari data permohonan. Status awal permohonan adalah Diajukan.	Warga

verifikasiKalurahan()	Melakukan verifikasi awal terhadap kelengkapan dokumen dan data permohonan. Jika disetujui, status berubah menjadi DisetujuiKalurahan.	Kalurahan
verifikasiDukcapil()	Melakukan verifikasi akhir terhadap permohonan. Jika disetujui, status diperbarui menjadi DisetujuiDukcapil, dan dokumen hasil dapat diterbitkan.	Dukcapil
verifikasiKalurahanAsal() dan verifikasiKalurahanTujuan()	Digunakan khusus untuk permohonan pindah. Kalurahan asal memvalidasi permohonan, sementara kalurahan tujuan memverifikasi kesesuaian dengan tujuan.	Kalurahan
konfirmasiKKTujuan()	Fungsi tambahan untuk alur pindah bergabung keluarga. Kepala Keluarga tujuan memberikan persetujuan atau penolakan terhadap permohonan pindah yang diajukan.	Warga (Kepala Keluarga)
batalkanPermohonan()	Digunakan oleh warga untuk membatalkan permohonan sebelum diproses lebih lanjut.	Warga

Setiap fungsi memanfaatkan *modifier* kontrol akses (onlyWarga, onlyKalurahan, onlyDukcapil) agar tidak terjadi penyalahgunaan hak. Selain itu, semua fungsi penting menghasilkan *event log* yang dapat digunakan untuk audit dan pelacakan proses.

Semua kontrak dirancang untuk menerapkan *role-based access control* (RBAC) untuk memastikan hanya aktor berwenang yang dapat menjalankan fungsi tertentu, yaitu warga hanya dapat mengajukan dan membatalkan permohonan miliknya sendiri, kalurahan hanya dapat memverifikasi permohonan dari wilayahnya, dan dukcapil memiliki hak tunggal untuk melakukan validasi akhir dan menerbitkan dokumen. Selain itu, seluruh data identitas pribadi (seperti NIK atau alamat) hanya disimpan di file JSON yang terenkripsi di IPFS sehingga tidak akan tampil secara langsung di blockchain, sementara dokumen hasil layanan diakses melalui CID IPFS yang hanya dapat diunduh oleh pihak yang terotorisasi melalui antarmuka dApp.

Dengan kombinasi *event logging* yang tercatat secara permanen di dalam blockchain dan *access modifier*, sistem dapat memastikan integritas proses dan menyediakan mekanisme audit yang transparan namun tetap menjaga kerahasiaan data pribadi. Beberapa *event* utama antara lain:

a. PermohonanDiajukan: Mencatat ketika warga mengirimkan permohonan baru.

b. VerifikasiKalurahan dan VerifikasiDukcapil: Mencatat hasil verifikasi petugas.
c. DokumenResmiDiunggah: Menandai penerbitan dokumen resmi.
d. KonfirmasiKKTujuan: Khusus untuk proses pindah bergabung KK.

Dengan mekanisme ini, riwayat seluruh proses permohonan dapat diverifikasi publik tanpa membuka isi dokumen sebenarnya, sehingga sistem tetap memenuhi prinsip transparansi tanpa kehilangan privasi (*privacy-preserving transparency*).

Rancangan *smart contract* yang telah dijelaskan pada Anak Subbab 3.2.4 menggambarkan bagaimana mekanisme otomasi, transparansi, dan keterlacakan proses administrasi kependudukan dapat diwujudkan melalui pencatatan transaksi di blockchain. Rancangan ini digunakan sebagai sarana untuk mengeksplorasi karakteristik teknis blockchain dalam mendukung proses pencatatan dan verifikasi data secara terdistribusi. Namun, mengingat proses tersebut melibatkan pengelolaan dokumen digital yang mengandung data sensitif, diperlukan mekanisme tambahan untuk mengkaji aspek keamanan dan kerahasiaan data. Oleh karena itu, pembahasan mengenai rancangan mekanisme keamanan dan enkripsi data disajikan pada anak subbab berikutnya.

3.2.5 Mekanisme Keamanan dan Enkripsi

Aspek keamanan menjadi salah satu fokus utama dalam eksplorasi penerapan teknologi blockchain pada konteks pencatatan kependudukan, mengingat karakteristik data yang dikelola bersifat sensitif dan memerlukan perlindungan yang memadai. Pada penelitian ini, mekanisme keamanan dikaji melalui penerapan kombinasi karakteristik bawaan blockchain dan teknik enkripsi kriptografi sebagai pendekatan untuk mengevaluasi bagaimana integritas, autentikasi, serta kerahasiaan data dapat dikelola dalam arsitektur terdesentralisasi.

Pada lapisan *onchain*, keamanan dijamin oleh karakteristik bawaan dari blockchain, yaitu *immutability* (data tidak dapat diubah setelah tercatat) dan *transparency* (catatan transaksi dapat diverifikasi publik). Setiap permohonan, hasil verifikasi, dan penerbitan dokumen dicatat melalui *event log* yang disimpan secara permanen di jaringan blockchain. Selain itu, kontrol akses diterapkan dengan pendekatan *Role-Based Access Control* (RBAC) di dalam *smart contract*. Setiap fungsi dilindungi dengan *modifier* seperti onlyWarga, onlyKalurahan, dan onlyDukcapil, sehingga hanya akun dengan peran tertentu yang dapat memanggil fungsi tersebut.

Semua data kependudukan, termasuk NIK dan informasi keluarga, disimpan secara *offchain* dalam berkas JSON yang terenkripsi sebelum diunggah ke IPFS. Enkripsi dilakukan

di sisi klien (*frontend*) menggunakan algoritma AES-256-CBC agar mengikuti standar FIPS 197 dan rekomendasi NIST SP 800-38A, hal tersebut dilakukan agar tidak ada data mentah yang pernah meninggalkan perangkat pengguna tanpa perlindungan. Berkas JSON tersebut juga diberi identitas unik menggunakan UUID untuk mencegah keterlacakan. Penggunaan enkripsi AES-256-CBC dan identitas unik tersebut juga diterapkan pada setiap dokumen hasil layanan yang diunggah ke IPFS guna menjaga kerahasiaan dokumen.

Kunci enkripsi (*secret key*) disimpan dan digunakan kembali untuk proses dekripsi saat dokumen diunduh melalui aplikasi dengan proses yang berjalan otomatis. Setelah proses enkripsi selesai, hasilnya diunggah ke jaringan IPFS dan sistem hanya menyimpan *Content Identifier* (CID) di blockchain sebagai referensi lokasi berkas terenkripsi tersebut. Dengan cara ini, integritas data tetap terjaga karena setiap perubahan sekecil apa pun akan menghasilkan CID yang berbeda serta kerahasiaan tetap terlindungi meskipun IPFS bersifat publik.

Setiap aktivitas penting dalam sistem menghasilkan *event log* di blockchain, misalnya PermohonanDiajukan, VerifikasiKalurahan, VerifikasiDukcapil, dan DokumenResmiDiunggah. *Event-event* ini berfungsi sebagai jejak audit permanen yang dapat digunakan untuk melacak siapa yang melakukan tindakan, kapan dilakukan, dan hasil dari tindakan tersebut. Dengan pendekatan ini, sistem memenuhi prinsip *accountability* (dapat dipertanggungjawabkan) dan *non-repudiation* (tidak dapat disangkal).

Tabel 3.10 berikut merangkum terkait mekanisme keamanan yang digunakan dalam pengembangan aplikasi terdesentralisasi.

Tabel 3.10 Ringkasan Mekanisme Keamanan

Lapisan	**Mekanisme**	**Pendekatan**	**Tujuan**
Onchain	*Role-Based Access Control*	Solidity *Smart Contract*	Pembatasan hak akses berdasarkan peran aktor.
Onchain	*Immutable event log*	Blockchain	Menjamin integritas dan keterlacakan proses
Offchain	Enkripsi dokumen dan data JSON, serta identitas unik	AES-256-CBC dan UUID	Melindungi privasi dan isi data keluarga.

Offchain	Penyimpanan terdesentralisasi	IPFS	Mencegah ketergantungan pada server terpusat

Melalui kombinasi mekanisme kontrol akses pada *smart contract*, pencatatan transaksi yang bersifat transparan di blockchain, serta penerapan enkripsi dokumen menggunakan AES-256-CBC sebelum penyimpanan pada IPFS, penelitian ini mengeksplorasi pendekatan pengamanan data kependudukan dalam arsitektur terdesentralisasi. Pendekatan tersebut digunakan untuk mengkaji bagaimana integritas, keterlacakan, dan perlindungan privasi data dapat dikelola secara teknis dalam sistem administrasi kependudukan berbasis blockchain, sekaligus menjadi dasar evaluasi keamanan pada tahap pengujian dan pembahasan selanjutnya.

3.3 Implementasi

Tahap implementasi dalam penelitian ini digunakan sebagai sarana untuk merealisasikan dan mengevaluasi rancangan arsitektur terdesentralisasi yang telah disusun pada tahap sebelumnya. Implementasi dilakukan dengan menerjemahkan konsep dan desain teknis ke dalam bentuk artefak perangkat lunak berupa purwarupa aplikasi terdesentralisasi. Pada tahap ini, pengembangan difokuskan pada penerapan mekanisme *smart contract*, integrasi penyimpanan dokumen *off-chain* menggunakan IPFS, serta penerapan enkripsi AES-256-CBC sebagai bagian dari eksplorasi karakteristik teknis teknologi yang digunakan. Purwarupa yang dihasilkan diposisikan sebagai media eksperimen untuk mengamati perilaku, alur eksekusi, dan implikasi teknis penerapan blockchain dan IPFS dalam konteks administrasi kependudukan.

Implementasi sistem dilakukan pada lingkungan pengembangan yang mendukung teknologi blockchain dan IPFS. Pengembangan *smart contract* menggunakan bahasa Solidity dengan bantuan *framework* Hardhat untuk proses kompilasi, serta simulasi jaringan lokal. *Library* Chai.js digunakan untuk menuliskan dan menjalankan *unit testing* pada fungsi-fungsi penting di dalam kontrak. Penyimpanan dokumen dan data dilakukan melalui IPFS yang diakses menggunakan Pinata API, sedangkan antarmuka pengguna dibangun menggunakan React.js dan diintegrasikan dengan blockchain melalui Ethers.js. Seluruh proses enkripsi dilakukan menggunakan algoritma AES-256-CBC di sisi *frontend* sebelum berkas diunggah ke IPFS. Manajemen identitas digital dan penandatanganan transaksi blockchain menggunakan MetaMask sebagai *wallet* utama. Ringkasan lingkungan pengembangan ditunjukkan pada Tabel 3.11.

Tabel 3.11 Ringkasan Lingkungan Pengembangan

Komponen	Teknologi	Keterangan
Smart Contract	Solidity	Bahasa utama untuk menulis kontrak di Ethereum-*compatible* blockchain
Framework dan jaringan pengujian	Hardhat, Chai.js	Untuk kompilasi, pengujian unit, dan simulasi jaringan lokal
Penyimpanan	IPFS (Pinata API)	Menyimpan berkas JSON dan dokumen terenkripsi
Frontend	React.js, Ethers.js, Node.js	Antarmuka dan koneksi dApp ke *smart contract*
Algoritma enkripsi	AES-256-CBC	Mengamankan berkas JSON sebelum diunggah ke IPFS
Wallet	MetaMask	Mengelola identitas digital dan menandatangani transaksi blockchain

Struktur implementasi sistem terbagi menjadi tiga lapisan utama sesuai rancangan arsitektur, yaitu *on-chain*, *off-chain*, dan *frontend*. Pada lapisan *on-chain*, sistem terdiri dari kumpulan *smart contract* modular yang meliputi PencatatanSipil.sol, PermohonanManager.sol, KontrolAkses.sol, PencatatanTypes.sol, PermohonanUtils.sol, dan PencatatanViewUtils.sol. Lapisan ini berfungsi untuk mencatat permohonan, memperbarui status, mengatur kontrol akses, mencatat CID IPFS, serta menghasilkan *event log* sebagai bukti transaksi. Setiap kontrak diuji menggunakan *framework* Hardhat dan Chai.js dengan metode *unit testing* dan *integration testing* guna memastikan seluruh logika berjalan sesuai rancangan.

Lapisan *off-chain* bertugas menyimpan berkas JSON permohonan dan semua jenis dokumen yang telah dienkripsi menggunakan AES-256-CBC. Proses unggah dilakukan melalui Pinata API untuk memperoleh CID yang unik bagi setiap berkas. CID tersebut kemudian dicatat di blockchain melalui *smart contract* guna menjamin keterlacakan serta keaslian data. Berbeda dengan data *on-chain* yang bersifat publik, data *off-chain* ini tidak dapat diakses tanpa kunci enkripsi yang sesuai sehingga privasi pengguna tetap terjaga.

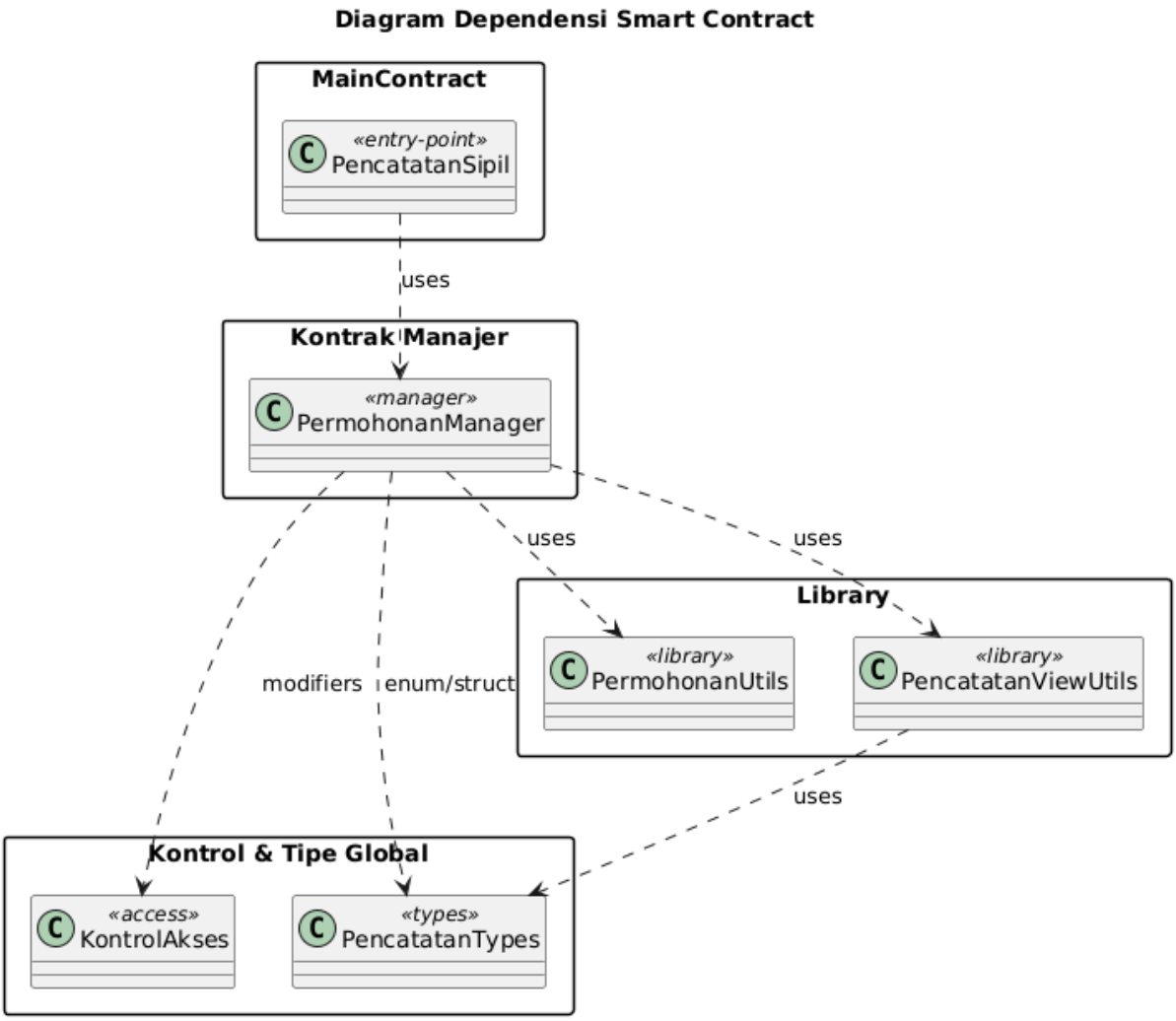
Lapisan *frontend* dikembangkan menggunakan React.js dan terhubung dengan *smart contract* menggunakan *library* Ethers.js. Antarmuka ini menangani autentikasi pengguna melalui *wallet* MetaMask, pengisian formulir permohonan, proses unggah dokumen, pelacakan status permohonan, serta proses enkripsi dan dekripsi dokumen. Enkripsi dilakukan

sepenuhnya di sisi klien menggunakan Crypto API, sehingga data sensitif tidak pernah dikirim dalam bentuk terbuka ke server mana pun.

Integrasi antar komponen diimplementasikan dengan skema komunikasi sebagai berikut:

a. *Frontend* mengirimkan data permohonan dan dokumen terenkripsi ke IPFS melalui API Pinata.
b. Setelah mendapatkan CID, *frontend* memanggil fungsi pada *smart contract* untuk mencatat transaksi permohonan dan CID tersebut.
c. *Smart contract* menyimpan CID ke blockchain dan memicu *event log* sebagai bukti transaksi.
d. Data dan dokumen dapat diakses kembali melalui CID yang tercatat, kemudian didekripsi di sisi pengguna menggunakan kunci enkripsi yang sama.

Mekanisme ini memastikan bahwa data sensitif tidak pernah tersimpan dalam bentuk terbuka di blockchain, tetapi tetap dapat diverifikasi keasliannya melalui CID yang tercatat secara publik.



Gambar 3.7 Diagram Dependensi *Smart Contract*

Implementasi *smart contract* dilakukan seperti yang tertera pada diagram dependensi di Gambar 3.7. Kontrak PencatatanSipil bertindak sebagai *entry point* yang mengorkestrasi PermohonanManager, memanfaatkan KontrolAkses (RBAC) serta tipe global dari PencatatanTypes, dan menggunakan dua *library* utilitas yaitu PermohonanUtils untuk pemeliharaan struktur data dan PencatatanViewUtils untuk konversi *enum–string*. Rancangan modular ini memudahkan pengujian per modul dan pemeliharaan kode.

Untuk proses enkripsi dan penyimpanan dokumen di IPFS dilakukan melalui tahapan sebagai berikut. Pengguna memilih dokumen yang akan diunggah, kemudian sistem mengenkripsi berkas tersebut menggunakan algoritma AES-256-CBC untuk menghasilkan *ciphertext* dan *initialization vector* (IV). Hasil enkripsi dikonversi ke format Base64 sebelum diunggah ke IPFS melalui Pinata API. Setelah Pinata mengembalikan CID, data tersebut disimpan di blockchain melalui *smart contract*. Pada saat dokumen diunduh kembali, sistem

mengambil berkas dari IPFS berdasarkan CID dan melakukan dekripsi menggunakan *secret key* yang sama. Mekanisme ini memastikan setiap berkas hanya dapat diakses oleh pihak yang berwenang dan menjamin integritas dokumen karena setiap perubahan sekecil apa pun akan menghasilkan CID yang berbeda.

Selama tahap implementasi, pengujian dilakukan secara iteratif terhadap seluruh fungsi dan logika bisnis di dalam *smart contract*. Proses pengujian dilakukan menggunakan *framework* Chai.js untuk memastikan kesesuaian antara perilaku sistem dan alur proses bisnis yang telah dirancang pada tahap sebelumnya. Pengujian ini mencakup validasi fungsi-fungsi utama seperti pengajuan permohonan, verifikasi oleh kalurahan dan Dukcapil, pembaruan status, serta pengelolaan *event log*. Pendekatan pengujian berulang ini memastikan setiap komponen bekerja secara konsisten dan mempermudah proses integrasi dengan *frontend* pada tahap akhir pengembangan.

Melalui tahap implementasi ini, rancangan arsitektur terdesentralisasi direalisasikan dalam bentuk purwarupa teknis yang digunakan sebagai media eksplorasi teknologi. Integrasi *smart contract*, mekanisme enkripsi AES-256-CBC, dan penyimpanan terdistribusi IPFS memungkinkan dilakukan pengamatan terhadap karakteristik transparansi, keamanan, dan efisiensi pengelolaan data kependudukan dalam pendekatan terdesentralisasi. Tahap selanjutnya difokuskan pada pengujian dan evaluasi teknis untuk menilai kinerja serta implikasi penerapan teknologi tersebut sesuai dengan tujuan penelitian.

3.4 Pengujian

Tahap pengujian dilakukan untuk memverifikasi kebenaran dan keandalan implementasi teknologi blockchain dan IPFS yang dieksplorasi dalam penelitian ini. Fokus utama pengujian diarahkan pada verifikasi fungsionalitas *smart contract* sebagai komponen inti sistem terdesentralisasi, serta pada evaluasi efisiensi dan konsistensi mekanisme penyimpanan dokumen terenkripsi menggunakan IPFS. Selain itu, pengujian juga mencakup validasi interaksi antar komponen *on-chain* dan *off-chain* guna memastikan bahwa integrasi teknologi yang diterapkan mampu mendukung alur administrasi kependudukan secara utuh, aman, dan dapat diaudit.

Pengujian dilakukan tidak hanya pada fungsi individual, tetapi juga terhadap integrasi antar kontrak secara keseluruhan. Meskipun seluruh skenario pengujian ditulis menggunakan *framework* Chai.js, beberapa pengujian yang dilakukan telah mencakup pengujian integrasi (*integration testing*) secara implisit, yaitu dengan menjalankan rangkaian fungsi yang

melibatkan beberapa kontrak sekaligus untuk memastikan interaksi antar modul bekerja dengan benar.

Sebagai contoh, pengujian terhadap alur pengajuan hingga verifikasi permohonan dilakukan dalam satu skenario berurutan: warga mengajukan permohonan melalui fungsi submitPermohonan(), petugas kalurahan melakukan verifikasi melalui verifikasiKalurahan(), dan petugas Dukcapil menyelesaikan proses melalui verifikasiDukcapil(). Hasil setiap tahap diverifikasi melalui pembacaan *state* permohonan dan *event log* yang dihasilkan di blockchain. Dengan demikian, pengujian yang dilakukan tidak hanya memastikan kebenaran logika internal masing-masing fungsi (*unit testing*), tetapi juga konsistensi *state smart contract* dan *event on-chain* dalam satu alur proses bisnis (*integration testing*).

Pengujian yang dilakukan terhadap *smart contract* mencakup beberapa fungsi utama, yaitu:

a. submitPermohonan() untuk menguji proses pengajuan permohonan oleh warga dan pencatatan data permohonan beserta CID IPFS di blockchain.
b. verifikasiKalurahan() dan verifikasiDukcapil() untuk menguji mekanisme verifikasi permohonan berdasarkan peran aktor dan pembaruan status permohonan.
c. batalkanPermohonan() untuk menguji pembatalan permohonan oleh pemohon sebelum proses diverifikasi.
d. getPermohonanByStatus() dan getPermohonanForDukcapil() untuk menguji pengambilan data permohonan berdasarkan status atau aktor terkait.
e. *Event Log* untuk memastikan setiap perubahan status permohonan menghasilkan event yang terekam di blockchain.

Rancangan skenario pengujian terhadap fungsi-fungsi tersebut disajikan pada Tabel 3.12.

Tabel 3.12 Skenario Pengujian *Smart Contract*

No	Fokus Uji	Fungsi/Bagian	Kondisi Awal	Ekspektasi
1	Pengajuan permohonan	submitPermohonan	Warga terdaftar, data valid	ID bertambah, status=Diajukan, cidIPFS tercatat
2	*Auto-increment* ID	submitPermohonan	Dua *submit* berturut-turut	ID 0, lalu 1 (bukan lompat)
3	Verifikasi Kalurahan	verifikasiKalurahan	Status=Diajukan	Status menjadi DisetujuiKalurahan

4	Penolakan Dukcapil	verifikasiDukcapil(tolak)	Status=Disetujui Kalurahan	status=DitolakDukcapil, alasan tersimpan
5	*String helper*	getStatusPermohonan/ getJenisPermohonan	Setelah *submit*	Mengembalikan label *string* yang benar
6	Akses ilegal	verifikasiKalurahan oleh warga	Bukan peran Kalurahan	*Revert* OnlyKalurahan
7	*Idempoten* status	verifikasiKalurahan (ulang)	Sudah bukan Diajukan	*Revert* PermohonanBukanDiajukan
8	*Event* submit	submitPermohonan	—	*Emit* PermohonanDiajukan dengan argumen tepat
9	Persetujuan Dukcapil	verifikasiDukcapil(setuju)	Status=Disetujui Kalurahan	status=DisetujuiDukcapil
10	Akses Dukcapil	verifikasiDukcapil oleh warga	Bukan peran Dukcapil	Revert OnlyDukcapil
11	*Event* verifikasi	VerifikasiKalurahan/VerifikasiDukcapil	—	Event keluar dengan argumen tepat
12	Pembatalan oleh pemohon	batalkanPermohonan	Permohonan milik sendiri, belum diverifikasi	status=DibatalkanPemohon
13	Akses illegal pembatalan permohonan	batalkanPermohonan	Permohonan bukan milik sendiri	*Revert* BukanPemilikPermohonan
14	*Query* per peran	getPermohonanByKalurahanAsal/Tujuan, getPermohonanForDukcapil	Data tersedia	Mengembalikan daftar permohonan sesuai *filter*
15	Dokumen resmi	verifikasiDukcapil/unggahDokumenResmi/getDokumenResmi	Status=Disetujui Kalurahan	CID tersimpan di *mapping*, *event* DokumenResmiDiunggah
16	Cegah duplikasi dokumen	unggahDokumenResmi(ulang)	Sudah ada CID	*Revert* DokumenResmiSudahAda

17	Larangan ambil dokumen jika belum ada	getDokumenResmi	Belum diunggah	*Revert* BelumAdaDokumenResmi
18	Fitur pindah—submit	submitPermohonan(Pindah)	Asal & tujuan valid	Jenis= Pindah, masuk ke indeks wilayah & status
19	Fitur pindah—validasi input	submitPermohonan(Pindah)	Tujuan 0/*unknown*	*Revert* TujuanTidakValid/IdKalurahanTujuanTidakDikenal
20	Verifikasi Kalurahan Asal	verifikasiKalurahanAsalPindah	Status=Diajukan	Status=DisetujuiKalurahanAsal/Ditolak
21	Verifikasi Kalurahan Tujuan	verifikasiKalurahanTujuanPindah	Status=Distujui Kalurahan Asal	Status=DisetujuiKalurahanTujuan/Ditolak
22	*Gatekeeping* tujuan	verifikasiKalurahanTujuanPindah	Bukan kalurahan tujuan	*Revert* HanyaKalurahanTujuan
23	Finalisasi Dukcapil (pindah)	verifikasiDukcapil	Status=Disetujui Kalurahan Tujuan	Status=Disetujui Dukcapil
24	Registrasi warga (akses)	KontrolAkses.registerWarga	NIK belum dipakai	*Event* WargaTerdaftar, mapping sesuai.
25	Proteksi duplikasi registrasi	registerWarga	Nik/*Wallet* sudah terpakai	*Revert* NikSudahDiklaim/*Wallet*SudahDigunakan
26	*Gatekeeping* onlyWargaTerdaftar	submitPermohonan/batalkanPermohonan	Belum daftar	*Revert* OnlyWargaTerdaftar

Setiap pengujian dilakukan dengan skenario masukan dan keluaran yang telah ditentukan sebelumnya. Misalnya, pada fungsi verifikasiKalurahan(), sistem diuji untuk memverifikasi apakah permohonan dengan status Diajukan dapat berubah menjadi DisetujuiKalurahan atau DitolakKalurahan hanya jika fungsi tersebut dijalankan oleh akun yang memiliki peran Kalurahan. Jika fungsi dijalankan oleh akun lain, sistem harus menolak transaksi dan mengembalikan pesan kesalahan (*revert error*).

Pengujian terhadap *frontend* dilakukan menggunakan pendekatan *black-box* sebagai media pemicu transaksi. Setiap fitur diuji untuk mengamati perilaku *smart contract* ketika dipicu melalui antarmuka menggunakan Ethers.js. Pengujian ini mencakup proses *login* menggunakan *wallet* MetaMask, pengisian formulir pengajuan, unggah dokumen terenkripsi, pelacakan status permohonan, serta proses unduh dan dekripsi dokumen dari IPFS.

Rancangan skenario *black-box testing* terhadap antarmuka pengguna (*frontend*) disajikan pada Tabel 3.13.

Tabel 3.13 Rencana Pengujian *Black-Box*

No	Fitur	Deskripsi	Keluaran
1	Login MetaMask	Autentikasi pengguna dengan *wallet*	*Wallet* terhubung dan alamat pengguna terdeteksi
2	Pengajuan Permohonan	Warga mengisi formulir dan mengunggah dokumen	Notifikasi sukses, permohonan tercatat di blockchain, pemohon bisa melihat status permohonannya "Diajukan"
3	Verifikasi Kalurahan	Petugas Kalurahan menyetujui permohonan	Status berubah menjadi Disetujui Kalurahan
4	Verifikasi Dukcapil	Petugas Dukcapil menyetujui permohonan dan menerbitkan dokumen	Status berubah menjadi Disetujui Dukcapil dan dokumen terunggah
5	Pelacakan Status	Pemohon memeriksa status permohonan	Status tampil sesuai pembaruan terakhir di blockchain
6	Unduh Dokumen	Pemohon mengunduh dokumen hasil layanan	File hasil dekripsi tampil benar dan dapat dibuka

Seluruh hasil pengujian dicatat sebagai dasar untuk memverifikasi perilaku dan konsistensi implementasi teknologi yang digunakan, khususnya pada interaksi antara komponen *on-chain* dan *off-chain*. Pendekatan ini bertujuan untuk memastikan bahwa mekanisme eksekusi *smart contract*, penyimpanan dokumen terenkripsi di IPFS, serta proses pemanggilan melalui antarmuka aplikasi dapat berinteraksi secara benar dan terkoordinasi dalam satu alur layanan kependudukan terdesentralisasi.

Selain pengujian fungsional dan integrasi yang berfokus pada keakuratan logika program dan interaksi antar modul, dilakukan pula pengujian kinerja teknis terhadap *smart contract*

untuk menilai efisiensi eksekusi transaksi. Pengujian kinerja teknis ini bertujuan untuk mengetahui kebutuhan komputasi yang diperlukan oleh setiap operasi pada *Ethereum Virtual Machine* (EVM), khususnya terkait konsumsi gas dan waktu eksekusi fungsi-fungsi utama yang membentuk alur layanan kependudukan. Aspek ini penting mengingat implementasi *smart contract* dalam konteks pelayanan publik harus mempertimbangkan biaya operasional transaksi serta kelayakan teknis ketika diterapkan pada jaringan blockchain nyata.

Pengujian kinerja teknis dilakukan pada lingkungan pengujian lokal menggunakan Hardhat, sehingga durasi eksekusi yang dicatat mencerminkan waktu pemrosesan pada tingkat eksekusi kontrak (simulasi) dan bukan waktu konfirmasi blok pada jaringan publik. Meskipun demikian, pendekatan ini tetap memberikan gambaran representatif mengenai kompleksitas relatif setiap operasi dan estimasi kebutuhan gas yang diperlukan. Pengujian dilakukan dengan memodifikasi skrip pengujian berbasis Hardhat dan Chai.js, sehingga selain memverifikasi kebenaran fungsi, setiap transaksi yang dieksekusi juga dicatat nilai konsumsi gas dan durasi pemrosesan (dalam milidetik) berdasarkan selisih waktu sebelum dan sesudah pemanggilan fungsi.

Instrumen pengujian ini diterapkan pada berbagai operasi kunci, seperti pengajuan permohonan, verifikasi berlapis oleh Kalurahan dan Dukcapil, pembatalan permohonan, serta fungsi tambahan pada skenario permohonan pindah penduduk. Konsumsi gas dicatat dalam satuan gas unit yang menggambarkan jumlah komputasi yang diperlukan untuk menjalankan suatu operasi pada EVM dan bukan dalam satuan biaya finansial, sehingga memberikan dasar yang objektif untuk menilai efisiensi struktur data, pemanfaatan *mapping*, dan emisi *event* pada *smart contract*.

Setelah pengujian-pengujian tersebut, dilakukan validasi terhadap kesesuaian sistem dengan kebutuhan yang telah dirumuskan pada tahap analisis. Validasi ini dilakukan melalui pengamatan langsung terhadap perilaku sistem sepanjang proses pengujian untuk menilai kesesuaian hasil pengujian dengan tujuan eksplorasi teknologi yang dirumuskan dalam penelitian ini.

Validasi difokuskan pada dua aspek utama, yaitu validasi perilaku fungsional *smart contract* dan integrasi IPFS, serta validasi karakteristik non-fungsional yang melekat pada sistem terdesentralisasi, seperti keamanan, keterlacakan, dan efisiensi proses.

a. Validasi Perilaku Fungsional

Validasi perilaku fungsional dilakukan untuk memverifikasi bahwa mekanisme yang diimplementasikan pada *smart contract* dan integrasinya dengan IPFS bekerja sesuai dengan

logika dan aturan yang telah dirancang. Aspek ini dimaksudkan untuk memastikan bahwa alur proses dan kontrol yang dikodekan dalam *smart contract* dapat dieksekusi secara benar dan konsisten di blockchain. Validasi perilaku fungsional mencakup beberapa mekanisme utama, yaitu:

1. Eksekusi fungsi pengajuan permohonan oleh warga yang menghasilkan pencatatan transaksi dan penyimpanan CID dokumen ke dalam *state smart contract*.
2. Mekanisme verifikasi berlapis oleh Kalurahan dan Dukcapil melalui fungsi yang sesuai dengan hak aksesnya.
3. Penyimpanan dan pengambilan dokumen digital melalui IPFS dengan referensi CID yang tercatat di blockchain.
4. Pencatatan *event log* sebagai rekam jejak perubahan status permohonan yang dapat digunakan untuk keperluan audit dan penelusuran riwayat transaksi.
5. Konsistensi data status permohonan yang ditampilkan pada antarmuka aplikasi dengan *state* yang tersimpan di blockchain..

Setiap mekanisme tersebut divalidasi dengan membandingkan hasil eksekusi fungsi dan perubahan *state* pada blockchain terhadap skenario yang dirancang pada tahap analisis. Apabila seluruh mekanisme dapat dieksekusi sesuai dengan aturan *smart contract* tanpa menghasilkan perilaku yang menyimpang, maka implementasi dinyatakan memenuhi aspek validasi perilaku fungsional.

b. Validasi Karakteristik Non-Fungsional

Selain perilaku fungsional, dilakukan pula validasi terhadap karakteristik non-fungsional yang menjadi fokus eksplorasi dalam penelitian ini. Validasi ini bertujuan untuk menilai sejauh mana pemanfaatan blockchain, *smart contract*, dan IPFS mampu menghadirkan sifat-sifat yang diharapkan dari sistem terdesentralisasi. Validasi non-fungsional mencakup beberapa aspek berikut:

1. Keamanan Data: Memastikan bahwa seluruh dokumen yang tersimpan di IPFS telah melalui proses enkripsi AES-256-CBC dan tidak dapat diakses tanpa kunci enkripsi yang sah.
2. Keterlacakan (*Auditability*): Memastikan bahwa setiap transaksi dan perubahan status permohonan tercatat secara permanen di blockchain melalui *event log* yang tidak dapat dimodifikasi.

3. Efisiensi Proses: Menilai penyederhanaan alur administrasi melalui otomatisasi pencatatan dan verifikasi berbasis *smart contract*, dibandingkan dengan proses manual konvensional.
4. Transparansi dan Akuntabilitas: Mengamati kemampuan sistem dalam menyediakan informasi status permohonan yang bersumber langsung dari data *on-chain* tanpa bergantung pada pihak perantara..

Validasi karakteristik non-fungsional dilakukan melalui observasi langsung terhadap hasil pengujian, analisis data transaksi dan *event log* di blockchain, serta pengujian enkripsi-dekripsi dokumen IPFS. Penilaian efisiensi dilakukan secara kualitatif dengan membandingkan alur proses yang dieksplorasi dalam penelitian ini dengan alur administrasi kependudukan konvensional.

Implementasi dinyatakan memenuhi tujuan eksplorasi apabila:

a. Mekanisme *smart contract* dapat dieksekusi sesuai logika yang dirancang tanpa menghasilkan *revert error* yang tidak diharapkan.
b. Semua transaksi dan *event log* tercatat secara konsisten dan dapat ditelusuri di blockchain.
c. Dokumen terenkripsi yang disimpan di IPFS dapat didekripsi kembali secara utuh dan valid.
d. Alur layanan dapat dijalankan secara digital tanpa ketergantungan pada interaksi fisik langsung.

Dengan demikian, tahap pengujian dan validasi ini memberikan bukti empiris bahwa eksplorasi pemanfaatan blockchain, smart contract, dan IPFS mampu mendukung pengelolaan data kependudukan secara terdesentralisasi sesuai dengan tujuan penelitian. Hasil validasi ini selanjutnya menjadi dasar untuk pembahasan dan analisis pada Bab 4.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Bagian ini menyajikan hasil implementasi dari rancangan arsitektur terdesentralisasi yang telah dibahas pada Bab 3. Implementasi dilakukan sebagai bentuk realisasi teknis untuk mengevaluasi bagaimana komponen utama yaitu *smart contract* sebagai lapisan logika *on-chain*, penyimpanan dokumen terdesentralisasi menggunakan IPFS, serta antarmuka aplikasi terdesentralisasi berinteraksi dalam satu alur layanan pencatatan kependudukan.

Implementasi ini dimaksudkan sebagai purwarupa eksperimental untuk mengamati perilaku teknologi blockchain, *smart contract*, dan IPFS dalam mendukung proses administrasi kependudukan. Fokus pembahasan pada subbab ini diarahkan pada hasil implementasi tiap komponen dan implikasinya terhadap aspek transparansi, keterlacakan, serta keamanan data dalam konteks sistem terdesentralisasi.

4.1.1 Implementasi *Smart Contract*

Implementasi *smart contract* ini dilakukan menggunakan Bahasa pemrograman Solidity pada versi 0.8.28 dan dijalankan dalam lingkungan Hardhat yang menyediakan fitur kompilasi, pengujian, serta simulasi jaringan blockchain berbasis Ethereum *Virtual Machine* (EVM). Struktur *smart contract* disusun secara modular dengan memisahkan tanggung jawab logika ke dalam beberapa kontrak utama yang saling berinteraksi. Pendekatan ini memungkinkan evaluasi terhadap bagaimana pemisahan logika, kontrol akses, dan manajemen data memengaruhi keterbacaan kode, kemudahan pengujian, serta konsistensi perilaku kontrak dalam mendukung proses administrasi kependudukan berbasis blockchain.

Salah satu aspek penting yang dieksplorasi dalam implementasi *smart contract* pada penelitian ini adalah pemodelan data *on-chain* yang digunakan untuk merepresentasikan entitas dan alur proses dalam layanan pencatatan kependudukan. Model data ini menjadi dasar bagi seluruh fungsi kontrak dalam menyimpan, memvalidasi, dan melacak status setiap permohonan secara permanen di blockchain.

Salah satu tipe data penting yang digunakan adalah *enumeration* (*enum*) yang berfungsi untuk mendefinisikan daftar nilai tetap pada sistem. Misalnya, JenisPermohonan seperti pada Gambar 4.1, digunakan untuk membedakan jenis layanan kependudukan yang diajukan oleh warga, seperti Kelahiran, Kematian, Perkawinan, Perceraian, dan Pindah Domisili. Selain itu,

terdapat Status seperti yang terdapat pada Gambar 4.2 dimana terdapat beberapa Status yang menggambarkan setiap tahapan proses pelayanan pencatatan kependudukan.

```
enum JenisPermohonan {
      Kelahiran,
      Kematian,
      Kawin,
      Cerai,
      Pindah
  }
```

Gambar 4.1 Kode *Enum* Jenis Permohonan

```
  enum Status {
      Diajukan,
      DisetujuiKalurahan,
      DitolakKalurahan,
      DisetujuiDukcapil,
      DitolakDukcapil,
      DisetujuiKalurahanAsal,
      DitolakKalurahanAsal,
      DisetujuiKalurahanTujuan,
      DitolakKalurahanTujuan,
      DibatalkanPemohon,
      MenungguKonfirmasiKKTujuan,
      DikonfirmasiKKTujuan,
      DitolakKKTujuan
  }
```

Gambar 4.2 Kode *Enum* Status Permohonan

Penggunaan *enum* ini memungkinkan pengamatan terhadap bagaimana pembatasan transisi status dapat diterapkan langsung pada level logika kontrak, sehingga setiap perubahan status hanya dapat terjadi melalui jalur yang telah didefinisikan. Dengan mekanisme tersebut, alur permohonan menjadi lebih terkontrol dan konsisten, serta meminimalkan potensi kesalahan logika maupun perubahan status yang tidak sesuai dengan proses yang diharapkan.

Data utama setiap permohonan direpresentasikan melalui struktur Permohonan, berfungsi sebagai representasi formal dari setiap proses pengajuan pencatatan kependudukan yang dilakukan oleh warga. Struktur ini dirancang sedemikian rupa untuk menampung seluruh informasi penting yang berkaitan dengan permohonan, mulai dari identitas pemohon, jenis peristiwa kependudukan yang diajukan, hingga jejak verifikasi oleh berbagai pihak yang

terlibat dalam alur layanan. Desain *struct* ini menjadi inti dari sistem pencatatan *onchain* karena seluruh perubahan status, waktu, dan hasil verifikasi akan tersimpan permanen di blockchain. Potongan kode pada Gambar 4.3 menunjukkan bentuk penyimpanan struktur tersebut.

```
struct Permohonan {
        uint256 id;
        uint256 waktuPengajuan;
        address pemohon;
        JenisPermohonan jenis;
        Status status;
        uint8 idKalurahanAsal;
        uint8 idKalurahanTujuan;
        string cidIPFS;
        string alasanPenolakan;
        address verifikatorKalurahan;
        address verifikatorKalurahanTujuan;
        address verifikatorDukcapil;
        uint256 waktuVerifikasiKalurahan;
        uint256 waktuVerifikasiKalurahanTujuan;
        uint256 waktuVerifikasiDukcapil;
        address konfirmatorKKTujuan;
        uint256 waktuKonfirmasiKKTujuan;
        bool konfirmasiKKTujuan;
        JenisPindah jenisPindah;
    }
```

Gambar 4.3 Kode *Struct* Permohonan

Struktur permohonan tersebut diimplementasikan untuk mengeksplorasi bagaimana satu representasi data *on-chain* dapat mengakomodasi berbagai jenis layanan kependudukan yang memiliki perbedaan alur proses. Setiap permohonan direpresentasikan dengan atribut id sebagai pengenal unik yang memungkinkan pelacakan permohonan secara deterministik di blockchain, sementara atribut waktuPengajuan digunakan untuk merekam timestamp saat permohonan pertama kali dicatat sebagai transaksi. Atribut pemohon menyimpan alamat *wallet* warga yang mengajukan permohonan, yang dalam konteks blockchain berfungsi sebagai identitas digital yang bersifat unik dan dapat diverifikasi secara kriptografis.

Atribut jenis dan status masing-masing merepresentasikan jenis layanan kependudukan yang diajukan (misalnya kelahiran, kematian, perkawinan, perceraian, atau pindah domisili) serta status proses yang sedang berjalan dalam alur verifikasi. Implementasi kedua atribut

tersebut sebagai *enumeration* (*enum*) bertujuan untuk mengeksplorasi bagaimana pembatasan nilai pada level *smart contract* dapat membentuk alur proses yang terstruktur dan terkontrol secara onchain.

Selanjutnya, atribut idKalurahanAsal dan idKalurahanTujuan digunakan untuk merepresentasikan keterlibatan wilayah administratif dalam setiap permohonan kependudukan. Nilai ini menjadi dasar eksplorasi mekanisme pembatasan akses berbasis wilayah (*region lock*) yang diimplementasikan langsung pada level *smart contract*. Pendekatan ini penting agar kontrak dapat mengidentifikasi kalurahan yang berwenang untuk melakukan tindakan verifikasi terhadap suatu permohonan berdasarkan asal atau tujuan wilayahnya. Penerapan logika pembatasan wilayah ini memungkinkan penelitian untuk mengkaji bagaimana struktur administratif yang bersifat hierarkis dan berbasis teritorial dapat dimodelkan ke dalam sistem terdesentralisasi tanpa menghilangkan prinsip kontrol kewenangan. Dengan demikian, *smart contract* tidak hanya berfungsi sebagai pencatat transaksi, tetapi juga sebagai mekanisme teknis yang menegakkan aturan kewilayahan sesuai dengan karakteristik tata kelola kependudukan di Indonesia.

Atribut cidIPFS digunakan untuk merepresentasikan *Content Identifier* (CID) dari berkas JSON yang disimpan pada jaringan IPFS. Berkas tersebut berisi data permohonan yang telah terenkripsi di sisi klien, termasuk dokumen pendukung seperti surat pengantar, Kartu Keluarga, atau surat keterangan. Dengan demikian, blockchain hanya menyimpan *pointer* menuju data terenkripsi tanpa menyimpan isi dokumen secara langsung, guna menjaga privasi dan menghemat biaya transaksi. Pendekatan ini digunakan untuk mengeksplorasi pola pemisahan penyimpanan data dalam konteks data kependudukan yang bersifat sensitif. Dengan hanya menyimpan penunjuk data berupa *hash*, blockchain berperan sebagai lapisan pencatatan dan verifikasi integritas, sementara data aktual dikelola pada penyimpanan terdistribusi. Pola ini tidak hanya menjaga privasi data, tetapi juga memberikan gambaran mengenai efisiensi biaya transaksi dan skalabilitas yang dapat dicapai melalui kombinasi blockchain dan IPFS

Apabila suatu permohonan tidak memenuhi kriteria pada tahap verifikasi tertentu, *smart contract* merekam informasi penolakan melalui atribut alasanPenolakan. Informasi tersebut dapat dimanfaatkan oleh pemohon sebagai dasar untuk memahami konteks penolakan dan melakukan penyesuaian pada pengajuan berikutnya.

Bagian berikutnya dari struktur ini dirancang untuk mengeksplorasi mekanisme pencatatan verifikasi berjenjang dalam konteks layanan kependudukan terdesentralisasi. Atribut verifikatorKalurahan, verifikatorKalurahanTujuan, dan verifikatorDukcapil

merepresentasikan alamat *wallet* petugas yang melakukan validasi pada masing-masing tingkat, sehingga setiap tindakan verifikasi dapat dikaitkan langsung dengan identitas kriptografis yang sah. Setiap atribut verifikator dilengkapi dengan catatan waktu (waktuVerifikasiKalurahan, waktuVerifikasiKalurahanTujuan, dan waktuVerifikasiDukcapil) yang memungkinkan proses verifikasi dianalisis secara kronologis. Pembaruan status permohonan dan pencatatan *event log* pada setiap tahap verifikasi dimanfaatkan untuk menunjukkan bagaimana blockchain dapat berperan sebagai media *audit trail* terdistribusi, tanpa ketergantungan pada mekanisme pencatatan terpusat.

Untuk jenis permohonan pindah penduduk, rancangan data *onchain* ini mengeksplorasi penanganan subalur yang lebih kompleks, termasuk skenario pindah gabung keluarga. Pada skenario tersebut, digunakan atribut tambahan seperti konfirmatorKKTujuan, waktuKonfirmasiKKTujuan, dan konfirmasiKKTujuan untuk merepresentasikan persetujuan dari kepala keluarga tujuan perpindahan. Permohonan pindah domisili pada skenario ini hanya diproses lebih lanjut setelah tercatat adanya konfirmasi positif dari kepala keluarga tujuan, sehingga alur perpindahan tidak hanya bergantung pada pemohon dan instansi verifikator. Pendekatan ini dimaksudkan untuk mengkaji bagaimana mekanisme persetujuan pihak ketiga dapat direpresentasikan secara eksplisit dan dapat diaudit melalui identitas *wallet* di blockchain. Untuk mengakomodasi variasi skenario perpindahan, atribut jenisPindah digunakan sebagai penanda jenis perpindahan, seperti pindah satu keluarga, pindah sebagian anggota keluarga, atau pindah dengan bergabung ke Kartu Keluarga lain. Atribut ini memungkinkan eksplorasi perbedaan logika verifikasi dalam satu jenis layanan yang memiliki kompleksitas alur yang berbeda..

Desain struktur Permohonan ini merepresentasikan upaya untuk mengeksplorasi bagaimana mekanisme pelayanan publik di dunia nyata dapat dimodelkan ke dalam struktur data *onchain* secara eksplisit dan terverifikasi. Pencatatan setiap tindakan, peran aktor, serta waktu proses secara permanen di blockchain memungkinkan analisis terhadap tingkat integritas, akuntabilitas, dan transparansi yang dihasilkan oleh pendekatan terdesentralisasi dibandingkan dengan pola pengelolaan data konvensional. Selain itu, penggunaan atribut cidIPFS sebagai penunjuk data *off-chain* serta penerapan pembatasan akses berbasis wilayah (*region lock*) menjadi bagian dari eksplorasi efisiensi dan keamanan arsitektur *hybrid*. Pendekatan ini menunjukkan bagaimana pemisahan antara penyimpanan data sensitif dan pencatatan transaksi dapat digunakan untuk menekan beban komputasi *onchain* sekaligus

menjaga kerahasiaan data pribadi, sejalan dengan prinsip perancangan sistem terdesentralisasi yang diadopsi dalam penelitian ini.

Dalam eksplorasi sistem pencatatan kependudukan berbasis blockchain ini, pengelolaan data *onchain* dilakukan dengan memanfaatkan struktur *mapping* sebagai mekanisme utama untuk penyimpanan dan pengindeksan. Pada konteks Solidity, *mapping* berperan sebagai struktur pasangan *key–value* yang memungkinkan akses data secara langsung berdasarkan kunci tertentu, tanpa mekanisme relasional sebagaimana pada basis data konvensional. Pendekatan tersebut relevan untuk dievaluasi karena operasi iterasi terhadap kumpulan data besar pada blockchain memiliki biaya gas yang relatif tinggi, sehingga pemilihan struktur data berpengaruh langsung terhadap performa dan biaya eksekusi kontrak.

Mapping utama yang digunakan adalah permohonans, yang berfungsi sebagai penyimpanan inti seluruh data permohonan berdasarkan ID unik. Struktur ini digunakan untuk merepresentasikan setiap entitas Permohonan yang memuat informasi esensial, seperti identitas pemohon, jenis layanan kependudukan, wilayah administratif yang terlibat, status proses, serta jejak waktu dan aktor verifikator pada setiap tahapan. Penggunaan variabel jumlahPermohonan dieksplorasi sebagai mekanisme pencacahan (*counter*) yang merepresentasikan jumlah entitas permohonan yang tercatat di blockchain. Variabel ini sekaligus dimanfaatkan sebagai dasar pembentukan identitas permohonan baru secara berurutan pada saat fungsi submitPermohonan() dipanggil. Pendekatan ini memungkinkan evaluasi terhadap konsistensi identifikasi data, keterurutan pencatatan transaksi, serta implikasinya terhadap keterlacakan permohonan dalam lingkungan *smart contract* yang bersifat deterministik.

Gambar 4.4 menunjukkan potongan kode deklarasi struktur *mapping* yang digunakan dalam kontrak.

```
uint256 public jumlahPermohonan;
mapping(uint256 => PencatatanTypes.Permohonan) permohonans;
mapping(address => uint256[]) public daftarPermohonanPemohon;
mapping(uint8 => uint256[]) public daftarPermohonanKalurahanAsal;
mapping(uint8 => uint256[]) daftarPermohonanKalurahanTujuan;
mapping(PencatatanTypes.Status => uint256[])
    public daftarPermohonanPerStatus;
mapping(uint256 => string) public cidDokumenResmi;
```

Gambar 4.4 Kode Struktur *Mapping*

Selain *mapping* utama tersebut, penelitian ini juga mengeksplorasi penggunaan beberapa *mapping* sekunder sebagai mekanisme pengindeksan data berdasarkan kategori tertentu. *Mapping* sekunder ini dirancang untuk mengamati bagaimana strategi pengindeksan *onchain* dapat meningkatkan efisiensi akses data tanpa melanggar keterbatasan komputasi pada lingkungan blockchain. Struktur daftarPermohonanPemohon digunakan untuk menyimpan daftar ID permohonan berdasarkan alamat *wallet* pemohon. Melalui pendekatan ini, setiap entitas pemohon dapat menelusuri riwayat permohonan yang terkait dengan dirinya secara langsung menggunakan fungsi *getter* seperti getPermohonanIDsByPemohon(*address*), tanpa memerlukan proses iterasi terhadap seluruh data permohonan yang tersimpan di kontrak.

Sementara itu, *mapping* daftarPermohonanKalurahanAsal dan daftarPermohonanKalurahanTujuan digunakan untuk mengeksplorasi mekanisme pengelompokan data *onchain* berdasarkan wilayah administratif yang terlibat dalam suatu permohonan. Struktur ini dirancang untuk menguji penerapan prinsip *region lock* pada *smart contract*, yaitu pembatasan akses verifikasi berdasarkan identitas wilayah yang melekat pada aktor kalurahan. Melalui pendekatan ini, petugas kalurahan asal hanya dapat memverifikasi permohonan yang memiliki idKalurahanAsal sesuai dengan identitas wilayahnya, sedangkan kalurahan tujuan hanya dapat mengakses permohonan dengan idKalurahanTujuan yang cocok. Mekanisme tersebut memungkinkan evaluasi terhadap bagaimana pembatasan akses berbasis wilayah dapat diterapkan secara deterministik di level *smart contract*, sekaligus mencerminkan struktur kewenangan administratif dalam layanan kependudukan konvensional.

Selanjutnya, penelitian ini juga menerapkan *mapping* daftarPermohonanPerStatus sebagai bentuk eksplorasi pengindeksan data berdasarkan tahapan proses permohonan. Struktur ini menyimpan daftar ID permohonan sesuai status terkini, sehingga memungkinkan pengelompokan data secara dinamis berdasarkan fase verifikasi yang sedang berlangsung, seperti menunggu verifikasi kalurahan, disetujui Dukcapil, atau ditolak. Pendekatan ini digunakan untuk mengamati bagaimana pengindeksan berbasis status dapat mengurangi kebutuhan penyaringan data di sisi klien serta meminimalkan operasi komputasi berulang yang berbiaya tinggi pada lingkungan blockchain.

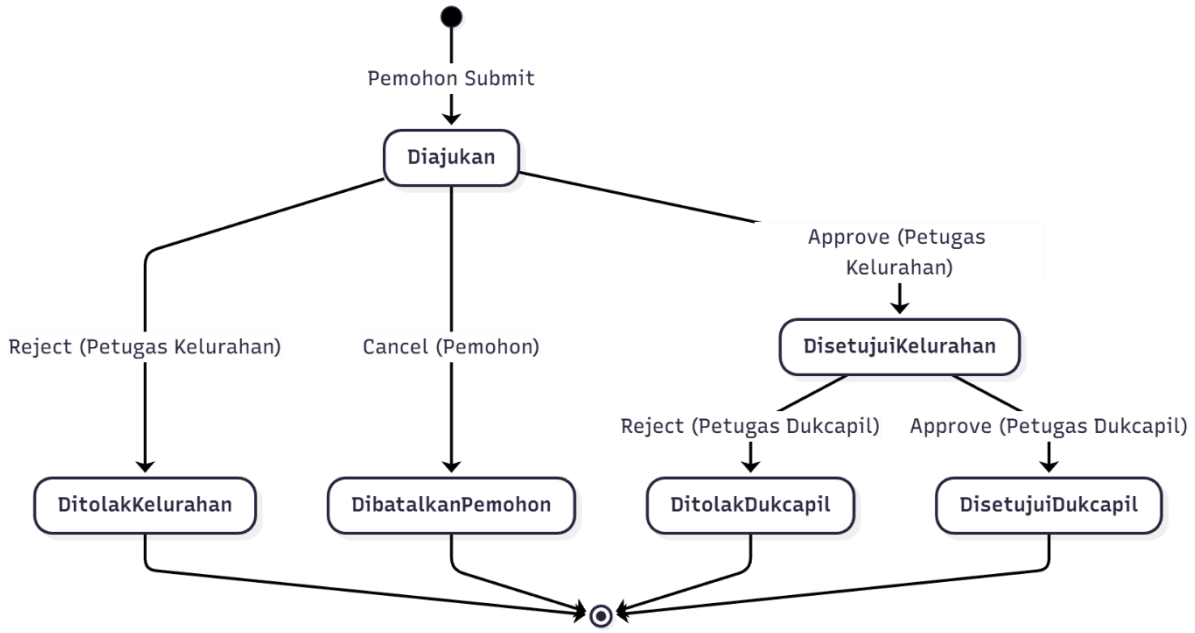
Selain pengelolaan data permohonan, kontrak juga mengeksplorasi penyimpanan *Content Identifier* (CID) dokumen resmi melalui *mapping* cidDokumenResmi. *Mapping* ini digunakan untuk merekam CID dari dokumen hasil layanan yang telah diterbitkan dan diunggah ke IPFS setelah proses verifikasi selesai. Penyimpanan CID di blockchain memungkinkan pengujian

terhadap keaslian dan integritas dokumen yang dapat diverifikasi tanpa menyimpan isi dokumen secara langsung di jaringan blockchain.

Secara keseluruhan, desain *mapping* dan strategi pengindeksan ini menunjukkan bahwa eksplorasi *smart contract* tidak hanya berfokus pada pencatatan transaksi, tetapi juga pada bagaimana struktur data *onchain* dapat dirancang untuk mendukung keterlacakan multi-aktor, efisiensi akses data, serta kebutuhan audit jangka panjang. Pendekatan ini memberikan gambaran mengenai potensi skalabilitas arsitektur data blockchain ketika diterapkan pada skenario layanan publik yang melibatkan banyak aktor dan alur proses yang berlapis.

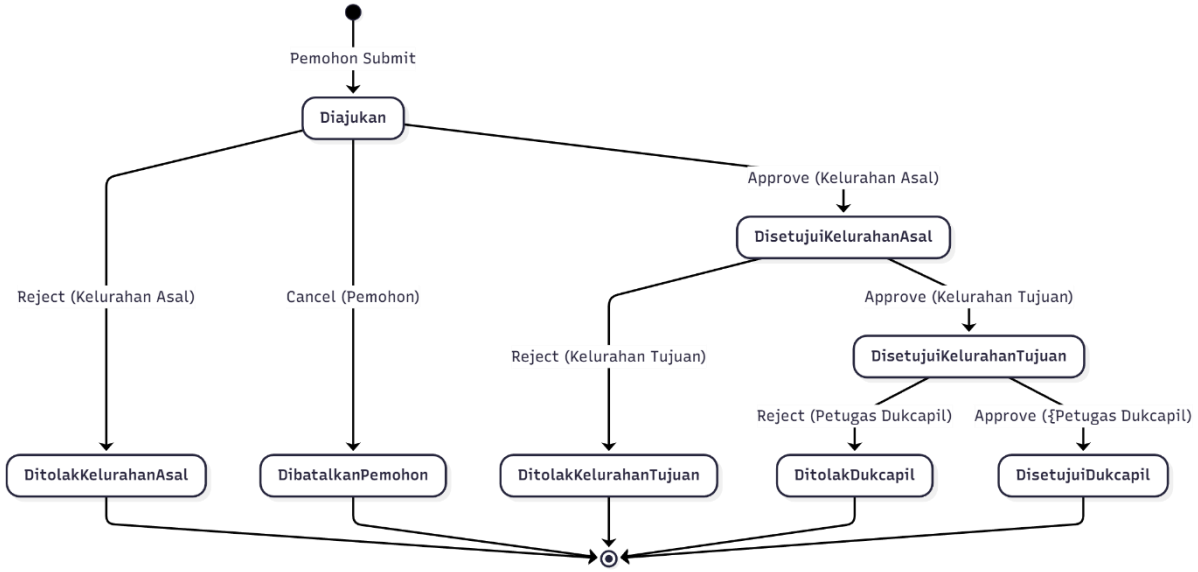
Untuk mengeksplorasi bagaimana alur layanan kependudukan dapat ditegakkan secara konsisten di tingkat *smart contract*, setiap perubahan status permohonan dirancang agar hanya dapat terjadi melalui fungsi-fungsi tertentu yang dikaitkan dengan peran aktor yang berwenang dan prasyarat yang terdefinisi dengan jelas. Setiap transisi status juga disertai dengan pencatatan *event* sebagai jejak audit *onchain*. Dengan pendekatan ini, *smart contract* tidak hanya berperan sebagai media penyimpanan data, tetapi juga sebagai mekanisme eksperimental untuk menegakkan aturan proses layanan, mulai dari pengajuan oleh warga, verifikasi berjenjang oleh kalurahan dan Dukcapil, hingga keputusan akhir berupa penerbitan atau penolakan permohonan.

Gambar 4.5 berikut menggambarkan alur perubahan status permohonan umum, mulai dari tahap pengajuan hingga keputusan akhir oleh Dukcapil.



Gambar 4.5 Alur Perubahan Status Permohonan

Alur proses dimulai ketika warga mengajukan permohonan, yang pada tahap ini direpresentasikan dengan pembentukan entitas Permohonan baru berstatus Diajukan serta pencatatan waktuPengajuan pada blockchain. Dari kondisi awal tersebut, alur selanjutnya bergantung pada jenis permohonan yang dipilih. Untuk kategori layanan umum, seperti kelahiran, kematian, perkawinan, dan perceraian, penelitian ini mengeksplorasi mekanisme verifikasi berjenjang dengan menempatkan kalurahan sebagai titik validasi awal. Melalui pemanggilan fungsi verifikasi pada *smart contract*, petugas kalurahan dapat memberikan keputusan persetujuan atau penolakan. Keputusan persetujuan memicu transisi status menuju tahap verifikasi Dukcapil, sedangkan keputusan penolakan mengakhiri alur proses dengan pencatatan alasan penolakan. Seluruh perubahan status tersebut disertai dengan penyimpanan atribut verifikator dan penanda waktu (*timestamp*), sehingga urutan proses dapat ditelusuri dan diaudit secara kronologis melalui data *onchain*.



Gambar 4.6 Alur Perubahan Status Permohonan Pindah

Pada skenario permohonan pindah penduduk sebagaimana ditunjukkan pada Gambar 4.6, penelitian ini mengeksplorasi alur verifikasi yang melibatkan lebih dari satu wilayah administratif, yaitu wilayah asal dan wilayah tujuan, serta aktor tambahan pada subalur tertentu berupa kepala keluarga tujuan (KK Tujuan). Apabila permohonan disetujui dan nilai jenisPindah mensyaratkan keterlibatan wilayah tujuan, status permohonan berpindah ke tahap verifikasi kalurahan tujuan; sementara pada kondisi tertentu, proses dapat langsung diteruskan ke tahap verifikasi Dukcapil.

Untuk subalur pindah gabung Kartu Keluarga, *smart contract* memodelkan kebutuhan persetujuan eksplisit dari kepala keluarga tujuan dengan menempatkan permohonan pada status MenungguKonfirmasiKKTujuan. Pada tahap ini, kepala keluarga tujuan memberikan keputusan persetujuan atau penolakan melalui *wallet* miliknya. Hanya apabila konfirmasi positif diberikan, permohonan dapat melanjutkan proses ke tahap verifikasi kalurahan asal, diikuti oleh verifikasi kalurahan tujuan, dan selanjutnya menuju verifikasi akhir oleh Dukcapil. Alur ini menunjukkan bagaimana *smart contract* digunakan untuk memformalkan mekanisme persetujuan multipihak dalam konteks perpindahan penduduk, sekaligus memastikan bahwa setiap tahapan keputusan dicatat secara eksplisit, berurutan, dan dapat ditelusuri sebelum dokumen kependudukan diterbitkan.

Smart contract yang diimplementasikan juga memodelkan mekanisme terminasi proses permohonan sebagai bagian dari eksplorasi pengendalian alur status. Pemohon diberikan opsi untuk membatalkan permohonan selama proses verifikasi belum dilakukan, yang

direpresentasikan dengan perubahan status menjadi DibatalkanPemohon. Di sisi lain, setiap otoritas yang berwenang pada tahap tertentu dapat memberikan keputusan penolakan dengan menyertakan alasan yang tercatat secara permanen di blockchain. Seluruh jalur keputusan, baik persetujuan, penolakan, maupun pembatalan, dirancang untuk berakhir pada status final yang tidak dapat diubah kembali. Pendekatan ini digunakan untuk mengevaluasi bagaimana *smart contract* dapat membatasi transisi status agar tidak terjadi perubahan berulang atau tidak sah setelah keputusan diambil. Dengan demikian, mekanisme ini berfungsi sebagai sarana eksplorasi penerapan pencegahan manipulasi status dalam pengelolaan proses administrasi berbasis blockchain.

Pada Gambar 4.7 berikut ditampilkan potongan kode yang menunjukkan logika pada fungsi verifikasiKalurahan, yang menjadi salah satu titik penting perubahan status dalam *smart contract*.

```
function verifikasiKalurahan(
        uint256 _id,
        bool _disetujui,
        string calldata _alasan
    ) external onlyKalurahan onlyKalurahanAsal(_id) {
        PencatatanTypes.Permohonan storage p = permohonans[_id];
        require(
            p.status == PencatatanTypes.Status.Diajukan,
            PermohonanBukanDiajukan()
        );
        _hapusByStatus(_id, PencatatanTypes.Status.Diajukan);
        if (_disetujui) {
            p.status = PencatatanTypes.Status.DisetujuiKalurahan;
            daftarPermohonanPerStatus[PencatatanTypes.Status.DisetujuiKalurahan]
                .push(_id);
        } else {
            p.status = PencatatanTypes.Status.DitolakKalurahan;
            p.alasanPenolakan = _alasan;
            daftarPermohonanPerStatus[PencatatanTypes.Status.DitolakKalurahan]
                .push(_id);
        }
        p.verifikatorKalurahan = msg.sender;
        p.waktuVerifikasiKalurahan = block.timestamp;
        emit VerifikasiKalurahan(
            _id,
            msg.sender,
            _disetujui,
```

```
            _alasan,
            block.timestamp
        );
    }
```

Gambar 4.7 Kode Fungsi VerifikasiKalurahan

Potongan kode pada Gambar 4.6 digunakan untuk mengeksplorasi bagaimana pembatasan akses dan validasi status dapat ditegakkan secara langsung melalui *smart contract*. Fungsi tersebut dirancang agar hanya dapat dipanggil oleh alamat *wallet* kalurahan yang telah terdaftar sebagai pihak berwenang (onlyKalurahan), serta mensyaratkan bahwa permohonan berada pada status awal Diajukan sebelum proses verifikasi dapat dilakukan. Apabila prasyarat terpenuhi, kontrak memperbarui atribut status dan waktuVerifikasiKalurahan sebagai bagian dari pencatatan jejak proses.. Selain pembaruan *state*, fungsi ini juga memancarkan *event* yang merepresentasikan keputusan verifikasi. *Event* tersebut dimanfaatkan oleh antarmuka dApp untuk mengamati perubahan *state* pada blockchain dan menyajikan pembaruan status permohonan secara *real-time*. Dengan pendekatan ini, potongan kode tersebut menjadi contoh bagaimana *smart contract* dapat digunakan untuk mengintegrasikan kontrol akses, transisi status, dan mekanisme notifikasi berbasis *event* dalam satu alur proses terdesentralisasi.

Selain fungsi verifikasiKalurahan() tersebut, masih ada beberapa fungsi lain yang juga bisa mengubah status, semua fungsi yang dapat mengubah status tersebut dirangkum pada Tabel 4.1.

Tabel 4.1 Ringkasan Fungsi *Smart Contract* dan Dampak Statusnya

Fungsi	**Deskripsi**	**Status**	***Event***
submitPermohonan()	Warga mengajukan permohonan baru dan mencatat waktu pengajuan serta CID dokumen.	Diajukan atau Meununggu Konfirmasi KK Tujuan (untuk permohonan pindah gabung KK)	PermohonanDiajukan
verifikasiKalurahan()	Kalurahan memverifikasi kelengkapan dan kebenaran dokumen permohonan. Jika setuju diteruskan ke Dukcapil, jika tidak ditolak.	Diajukan menjadi Disetujui Kalurahan/Ditolak Kalurahan	VerifikasiKalurahan
konfirmasiPindahGabungKK()	Kepala keluarga tujuan memberikan konfirmasi untuk permohonan pindah gabung KK.	Menunggu Konfirmasi KK Tujuan menjadi Dikonfirmasi KK Tujuan/Ditolak KK Tujuan	KonfirmasiKKTujuan

verifikasiKalurahanAsalPindah()	Kalurahan memverifikasi kelengkapan dan kebenaran dokumen permohonan untuk permohonan pindah. Jika setuju diteruskan ke Kalurahan Tujuan, jika tidak ditolak.	Diajukan atau Dikonfirmasi KK Tujuan menjadi Disetujui Kalurahan Asal/Ditolak Kalurahan Asal	VerifikasiKalurahan
verifikasiKalurahanTujuanPindah()	Dijalankan oleh Kalurahan Tujuan untuk permohonan pindah. Menentukan apakah permohonan dapat diteruskan ke Dukcapil.	DisetujuiKalurahanAsal menjadi DisetujuiKalurahanTujuan/ DitolakKalurahanTujuan	VerifikasiKalurahan
verifikasiDukcapil()	Dukcapil melakukan verifikasi akhir dan menentukan apakah dokumen diterbitkan atau ditolak.	Disetujui Kalurahan / Disetujui Kalurahan Tujuan menjadi DisetujuiDukcapil / DitolakDukcapil	VerifikasiDukcapil
batalkanPermohonan()	Pemohon membatalkan permohonan sebelum diverifikasi.	Diajukan menjadi Permohonan Dibatalkan	PermohonanDibatalkan

Fungsi-fungsi tersebut dirancang untuk mengeksplorasi penerapan mekanisme kontrol akses dan validasi transisi status secara eksplisit di tingkat *smart contract* melalui penggunaan *require statement* dan *modifier* berbasis peran. Misalnya, fungsi verifikasi kalurahan hanya dapat dipanggil oleh alamat kalurahan yang sesuai wilayah (idKalurahanAsal atau idKalurahanTujuan), fungsi verifikasi Dukcapil hanya untuk alamat dengan peran Dukcapil, dan fungsi konfirmasi KK hanya untuk alamat kepala keluarga tujuan yang tercatat. Setiap fungsi memeriksa status awal yang valid, menolak panggilan berulang, memperbarui status serta *timestamp*, dan memancarkan *event* yang spesifik. Pola ini digunakan untuk mengkaji bagaimana *smart contract* dapat menegakkan validitas transisi proses, merekam akuntabilitas aktor, dan menyediakan keterlacakan waktu secara konsisten tanpa bergantung pada mekanisme pengawasan terpusat.

Untuk mendukung mekanisme tersebut, *smart contract* dalam penelitian ini mengeksplorasi penerapan sistem kontrol akses berbasis peran (*Role-Based Access Control / RBAC*) sebagai cara untuk membatasi eksekusi fungsi berdasarkan identitas dan kewenangan aktor. Pendekatan ini digunakan untuk menguji bagaimana pembagian peran dapat ditegakkan langsung di tingkat *smart contract* tanpa bergantung pada validasi di sisi aplikasi. Selain berfungsi sebagai mekanisme keamanan, penerapan *RBAC* juga memberikan representasi

digital terhadap struktur hierarkis pelayanan kependudukan yang berlaku di dunia nyata. Tiga peran utama yang didefinisikan dalam kontrak adalah Warga Terdaftar, Kalurahan, dan Dukcapil. Masing-masing peran memiliki kewenangan spesifik yang dibatasi oleh fungsi dan wilayah tanggung jawabnya. Ringkasan hubungan antara peran dan fungsi utama ditunjukkan pada Tabel 4.2.

Tabel 4.2 Ringkasan Peran dan Fungsi

Peran	**Kewenangan Utama**	**Fungsi**	**Batasan**
Warga Terdaftar	Mengajukan dan membatalkan permohonan	submitPermohonan(), batalkanPermohonan()	Tidak dibatasi wilayah
Kalurahan	Melakukan verifikasi dokumen warga di wilayah asal atau tujuan	verifikasiKalurahan(), verifikasiKalurahanAsalPindah(), verifikasiKalurahanTujuanPindah()	Hanya untuk idKalurahanAsal atau idKalurahanTujuan
Dukcapil	Melakukan verifikasi akhir dan penerbitan dokumen resmi	verifikasiDukcapil()	Tidak dibatasi wilayah
Kepala Keluarga (khusus alur pindah gabung kk)	Memberikan konfirmasi terhadap permohonan pindah	konfirmasiKKTujuan()	Hanya untuk permohonan dengan NIK KK terkait

Sebagai contoh, akun dengan peran Kalurahan dieksplorasi kemampuannya dalam menjalankan fungsi-fungsi verifikasi permohonan yang berada dalam wilayahnya, seperti verifikasiKalurahan(), verifikasiKalurahanAsalPindah(), dan verifikasiKalurahanTujuanPindah(). Kontrak secara eksplisit menolak akses terhadap permohonan dari wilayah lain melalui mekanisme *region lock*, di mana kecocokan antara alamat *wallet* pemanggil fungsi dan idKalurahan pada data permohonan divalidasi langsung di tingkat *smart contract*. Sementara itu, peran Dukcapil hanya dapat menjalankan fungsi

verifikasiDukcapil() sebagai tahap akhir verifikasi, dan Warga Terdaftar hanya dapat menggunakan fungsi submitPermohonan() serta batalkanPermohonan().

Gambar 4.8 berikut menunjukkan potongan kode yang digunakan untuk mengeksplorasi penerapan *modifier* berbasis peran dan wilayah sebagai mekanisme otorisasi eksekusi fungsi pada *smart contract*.

```
    modifier onlyDukcapil() {
        require(msg.sender == dukcapil, OnlyDukcapil());
        _;
    }
    modifier onlyKalurahan() {
        require(kalurahan[msg.sender], OnlyKalurahan());
        _;
    }
    modifier onlyWargaTerdaftar() {
        require(
            bytes(nikByWallet[msg.sender]).length > 0,
            OnlyWargaTerdaftar()
        );
        _;
    }
```

Gambar 4.8 Kode *Modifier*

Dengan diterapkannya mekanisme kontrol akses berbasis peran dan wilayah, *smart contract* mampu membatasi setiap perubahan data dan status permohonan agar hanya dapat dipicu oleh aktor yang memiliki otoritas yang sah sesuai hierarki layanan kependudukan. Selain itu, sistem juga menerapkan prinsip *least privilege*, yaitu setiap peran hanya memiliki hak akses minimum yang diperlukan untuk menjalankan tugasnya. Hal ini tidak hanya meningkatkan keamanan, tetapi juga menjaga konsistensi proses agar seluruh tindakan dalam sistem dapat dipertanggungjawabkan dan diaudit melalui *event log* yang tercatat secara permanen di blockchain.

Selain mengatur alur status dan hak akses, *smart contract* juga dirancang untuk memiliki tingkat observabilitas yang tinggi melalui mekanisme *event logging*. Mekanisme ini berfungsi untuk merekam setiap aktivitas penting yang terjadi di dalam kontrak, seperti pengajuan, verifikasi, penolakan, hingga penerbitan dokumen resmi. Setiap kali salah satu fungsi utama dijalankan, kontrak akan memancarkan (*emit*) sebuah *event* ke jaringan blockchain, yang kemudian dapat dideteksi dan diproses oleh aplikasi *frontend* maupun pihak auditor secara

publik. Dengan demikian, penelitian ini mengeksplorasi bagaimana blockchain memungkinkan proses pencatatan dan audit dilakukan secara *real-time* melalui data *onchain*, tanpa ketergantungan pada basis data internal.

Selain mengatur alur status dan otorisasi, *smart contract* ini juga dilengkapi dengan mekanisme *event logging* sebagai sarana observabilitas sistem. Mekanisme ini berfungsi untuk merekam setiap aktivitas penting yang terjadi di dalam kontrak, seperti pengajuan permohonan, pembatalan, verifikasi oleh pejabat berwenang, konfirmasi kepala keluarga tujuan, hingga pengunggahan dokumen resmi hasil penerbitan. Melalui *event log*, penelitian ini mengeksplorasi bagaimana transaksi dan perubahan status dapat ditelusuri serta diverifikasi secara publik dan permanen, tanpa ketergantungan pada akses langsung ke data internal kontrak.

Event-event tersebut didefinisikan untuk memancarkan (*emit*) informasi penting setiap kali fungsi utama dijalankan. Setiap *event* berperan sebagai jejak audit (audit *trail*) yang *immutable* di blockchain, sehingga dapat digunakan untuk melacak siapa yang melakukan suatu tindakan, kapan tindakan tersebut dilakukan, dan apa hasilnya. Gambar 4.9 menunjukkan potongan kode deklarasi seluruh *event* yang digunakan dalam kontrak.

```
event PermohonanDiajukan(
    uint256 indexed id,
    address indexed pemohon,
    JenisPermohonan jenis,
    string cidIPFS,
    uint256 waktu
);

event PermohonanPindahDiajukan(
    uint256 indexed id,
    address indexed pemohon,
    JenisPindah jenisPindah,
    string cidIPFS,
    uint256 waktu
);

event PermohonanDibatalkan(
    uint256 indexed id,
    address indexed pemohon,
    uint256 waktu
);
```

```
event VerifikasiKalurahan(
    uint256 indexed id,
    address indexed verifikator,
    bool disetujui,
    string alasan,
    uint256 waktu
);

event VerifikasiDukcapil(
    uint256 indexed id,
    address indexed verifikator,
    bool disetujui,
    string alasan,
    uint256 waktu
);

event KonfirmasiKKTujuan(
    uint256 indexed idPermohonan,
    bytes32 indexed nikKepalaKeluargaTujuanHash,
    bool disetujui,
    uint256 waktu
);

event DokumenResmiDiunggah(
    uint256 indexed idPermohonan,
    string cidDokumen,
    uint256 waktu
);
```

Gambar 4.9 Kode Deklarasi *Event*

Event PermohonanDiajukan dan PermohonanPindahDiajukan merepresentasikan mekanisme pencatatan awal permohonan baru yang diajukan warga. Keduanya memuat informasi penting seperti jenis permohonan, CID dari berkas permohonan di IPFS, serta *timestamp* waktu pengajuan. Pemilahan antara permohonan umum dan permohonan pindah (JenisPindah) dilakukan untuk mengeksplorasi bagaimana *smart contract* digunakan untuk mengklasifikasikan pelacakan jenis layanan yang diajukan.

Event PermohonanDibatalkan merepresentasikan mekanisme pengelolaan perubahan status permohonan pada tingkat *smart contract* ketika pemohon menarik kembali permohonannya sebelum tahapan verifikasi diselesaikan. Sementara itu, *event*

VerifikasiKalurahan dirancang sebagai representasi untuk merekam seluruh hasil proses verifikasi di tingkat kalurahan, baik pada permohonan umum maupun permohonan pindah, termasuk peran kalurahan sebagai pihak asal maupun tujuan. Pendekatan penyatuan verifikasi ke dalam satu *event* yang sama mencerminkan eksplorasi desain *smart contract* dalam menjaga konsistensi struktur data *on-chain* serta mendukung keterlacakan proses lintas wilayah. *Event* ini mencatat identitas verifikator, hasil keputusan (disetujui atau ditolak), alasan keputusan, serta waktu eksekusi, sehingga memungkinkan analisis terhadap bagaimana setiap tindakan administratif dapat diaudit secara kronologis dan transparan melalui jejak transaksi blockchain.

Tahap selanjutnya adalah verifikasi akhir di tingkat Dukcapil, yang direpresentasikan dalam *event* VerifikasiDukcapil. *Event* ini memiliki struktur yang serupa dengan VerifikasiKalurahan, namun dieksplorasi lebih lanjut sebagai penanda transisi status dari fase verifikasi menuju fase penerbitan dokumen resmi. Ketika permohonan disetujui, sistem menghasilkan dokumen resmi dalam bentuk berkas terenkripsi yang disimpan secara *off-chain* pada IPFS. Integrasi antara proses *on-chain* dan penyimpanan *off-chain* tersebut direkam melalui *event* DokumenResmiDiunggah, yang menyimpan CID dokumen hasil penerbitan beserta *timestamp* waktu pengunggahan. CID yang tercatat pada *event* ini berfungsi sebagai referensi permanen yang tidak dapat dimodifikasi, sehingga memungkinkan eksplorasi mengenai bagaimana blockchain dan IPFS dapat dikombinasikan untuk menyediakan mekanisme akses dokumen digital yang terverifikasi melalui antarmuka dApp.

Khusus untuk permohonan pindah gabung KK, *smart contract* juga menyediakan *event* KonfirmasiKKTujuan yang merekam tindakan kepala keluarga tujuan ketika memberikan konfirmasi atas permohonan tersebut. *Event* ini mencatat *hash* dari NIK kepala keluarga tujuan, hasil konfirmasi (disetujui atau tidak), serta waktu konfirmasi. Dengan demikian, seluruh pihak yang terlibat termasuk keluarga tujuan memiliki jejak digital yang dapat diverifikasi secara publik.

Pencatatan *event* dalam *smart contract* tidak hanya dieksplorasi sebagai sarana pelacakan internal, tetapi juga sebagai fondasi arsitektur *event-driven* pada lapisan aplikasi. Setiap kali kontrak memancarkan *event* baru, dApp akan mendeteksinya melalui *event listener* yang terhubung dengan *node* Ethereum, sehingga perubahan status permohonan dapat direfleksikan secara otomatis pada antarmuka pengguna. Sebagai ilustrasi, ketika *event* VerifikasiKalurahan dipancarkan, dApp secara langsung memperbarui status permohonan menjadi “Disetujui Kalurahan” tanpa memerlukan interaksi ulang dari pengguna. Pendekatan ini memungkinkan

eksplorasi mengenai bagaimana mekanisme *event on-chain* dapat mendukung proses observasi secara *real-tim*e dan mengurangi ketergantungan pada pembaruan manual.

Lebih lanjut, mekanisme *event logging* ini dieksplorasi sebagai sarana observabilitas dan auditabilitas publik dalam sistem kependudukan terdesentralisasi. Setiap tindakan administratif, mulai dari pengajuan permohonan hingga penerbitan dokumen resmi, meninggalkan jejak digital yang bersifat permanen dan tidak dapat dimodifikasi pada blockchain. Dengan demikian, *smart contract* tidak hanya berperan sebagai pengatur alur proses, tetapi juga sebagai arsip transparan yang memungkinkan penelusuran dan verifikasi independen terhadap akuntabilitas layanan publik.

Selain fungsi-fungsi transaksional yang mengubah status data, *smart contract* ini juga menyediakan sejumlah fungsi baca (*view functions*) dan utilitas *query* yang digunakan untuk mengambil informasi dari blockchain tanpa memerlukan biaya gas. Fungsi-fungsi ini dirancang sebagai langkah eksploratif terhadap mekanisme penampilan data permohonan secara cepat dan efisien sesuai dengan peran pengguna yang mengaksesnya, seperti warga, kalurahan, maupun Dukcapil.

Setiap fungsi *view* pada *smart contract* diimplementasikan menggunakan kata kunci *view*, yang merepresentasikan pendekatan eksploratif dalam pemisahan antara operasi pembacaan data dan perubahan *state* pada blockchain. Fungsi-fungsi ini hanya melakukan akses terhadap data *on-chain* tanpa memicu eksekusi transaksi, sehingga pemanggilannya dapat dilakukan secara *off-chain* tanpa menimbulkan biaya gas. Mekanisme ini menjadi komponen penting untuk mengeksplorasi bagaimana antarmuka pengguna dapat menampilkan status permohonan, riwayat verifikasi, serta referensi dokumen resmi secara *real-time* tanpa membebani jaringan blockchain. Berikut adalah beberapa implementasi *view functions* utama yang digunakan di dalam sistem.

```
function getPermohonanIDsByPemohon(address _pemohon)
    external
    view
    returns (uint256[] memory)
{
    return daftarPermohonanPemohon[_pemohon];
}

function getPermohonan(uint256 _id)
    external
```

```
    view
    returns (PencatatanTypes.Permohonan memory)
{
    return permohonans[_id];
}
```

Gambar 4.10 Kode *View Functions* Pemohon

Kedua fungsi pada Gambar 4.10 dieksplorasi sebagai mekanisme penyediaan akses data *on-chain* bagi warga yang berperan sebagai pemohon. Fungsi getPermohonanIDsByPemohon() mengembalikan daftar seluruh ID permohonan yang pernah diajukan oleh suatu alamat *wallet*, sedangkan getPermohonan() digunakan untuk menampilkan detail lengkap dari satu permohonan tertentu berdasarkan ID-nya. Melalui kombinasi kedua fungsi *view* tersebut, penelitian ini mengeksplorasi bagaimana informasi status, riwayat pengajuan, dan hasil keputusan dapat diakses secara langsung oleh pemohon tanpa harus berinteraksi dengan data mentah blockchain.

Bagi petugas kalurahan, kontrak menyediakan fungsi baca yang dibatasi dengan *modifier* onlyKalurahan, yang memastikan bahwa data yang diambil hanya berasal dari wilayah administratif yang sesuai dengan alamat *wallet* pemanggil. Pendekatan ini digunakan untuk mengevaluasi penerapan *region lock* pada tingkat *smart contract*. Implementasi mekanisme tersebut dapat diamati pada fungsi getPermohonanByKalurahanAsal() dan getPermohonanByKalurahanTujuan() pada Gambar 4.11, yang masing-masing menampilkan daftar ID permohonan berdasarkan peran kalurahan sebagai wilayah asal maupun tujuan.

```
function getPermohonanByKalurahanAsal()
    external
    view
    onlyKalurahan
    returns (uint256[] memory)
{
    uint8 idKalurahan = idKalurahanByAddress[msg.sender];
    return daftarPermohonanKalurahanAsal[idKalurahan];
}

function getPermohonanByKalurahanTujuan()
    external
    view
    onlyKalurahan
    returns (uint256[] memory)
```

```
{
    uint8 idTujuan = idKalurahanByAddress[msg.sender];
    return daftarPermohonanKalurahanTujuan[idTujuan];
}
```

Gambar 4.11 Kode *View Funtions* Kalurahan

Untuk mengeksplorasi efisiensi proses verifikasi pada tingkat *smart contract*, kontrak juga menyediakan fungsi-fungsi *filtering* yang lebih spesifik. Fungsi getPermohonanBelumVerifikasiKalurahan() pada Gambar 4.12 digunakan untuk menampilkan daftar permohonan yang masih menunggu verifikasi berdasarkan status tertentu. Fungsi ini menyaring seluruh permohonan di wilayah kalurahan pemanggil dengan cara melakukan iterasi terhadap *array* daftarPermohonanKalurahanAsal dan mengembalikan hanya permohonan dengan status yang sesuai. Pendekatan ini dieksplorasi sebagai stretegi pembatasan agar petugas kalurahan hanya melihat data yang relevan dan belum diproses.

```
function getPermohonanBelumVerifikasiKalurahan(
    PencatatanTypes.Status _status
) external view onlyKalurahan returns (uint256[] memory) {
    uint8 idKalurahan = idKalurahanByAddress[msg.sender];
    uint256[] storage semua = daftarPermohonanKalurahanAsal[idKalurahan];

    uint256[] memory temp = new uint256[](semua.length);
    uint256 count = 0;

    for (uint256 i = 0; i < semua.length; i++) {
        if (permohonans[semua[i]].status == _status) {
            temp[count] = semua[i];
            count++;
        }
    }

    uint256[] memory hasil = new uint256[](count);
    for (uint256 i = 0; i < count; i++) {
        hasil[i] = temp[i];
    }

    return hasil;
}
```

Gambar 4.12 Fungsi getPermohonanBelumVerifikasiKalurahan()

Selain itu, terdapat fungsi getPermohonanSiapVerifikasiKalurahanAsal() seperti pada Gambar 4.13 yang memiliki logika penyaringan tambahan untuk menangani kasus khusus permohonan pindah gabung KK. Dalam fungsi ini, permohonan hanya akan dikembalikan apabila telah mencapai status DikonfirmasiKKTujuan (untuk kasus pindah gabung KK) atau masih dalam status Diajukan (untuk kasus lainnya). Pendekatan ini dieksplorasi sebagai upaya memindahkan kompleksitas logika alur proses ke tingkat *smart contract*, sehingga pengambilan data dapat disesuaikan secara otomatis dengan karakteristik masing-masing jenis permohonan tanpa perlu pemrosesan tambahan di sisi aplikasi.

```
function getPermohonanSiapVerifikasiKalurahanAsal()
    external
    view
    onlyKalurahan
    returns (uint256[] memory)
{
    uint8 idKalurahan = idKalurahanByAddress[msg.sender];
    uint256[] storage semua = daftarPermohonanKalurahanAsal[idKalurahan];

    uint256[] memory temp = new uint256[](semua.length);
    uint256 count = 0;
    for (uint256 i = 0; i < semua.length; i++) {
        PencatatanTypes.Permohonan storage p = permohonans[semua[i]];
        if (
            (p.jenis == PencatatanTypes.JenisPermohonan.Pindah &&
                p.jenisPindah ==
                PencatatanTypes.JenisPindah.PindahGabungKK &&
                p.status == PencatatanTypes.Status.DikonfirmasiKKTujuan) ||
            (p.status == PencatatanTypes.Status.Diajukan)
        ) {
            temp[count] = semua[i];
            count++;
        }
    }
    uint256[] memory hasil = new uint256[](count);
    for (uint256 i = 0; i < count; i++) {
        hasil[i] = temp[i];
    }
    return hasil;
}
```

Gambar 4.13 Fungsi getPermohonanSiapVerifikasiKalurahanAsal()

Sementara itu, bagi pihak Dukcapil, fungsi getPermohonanForDukcapil() seperti pada Gambar 4.14 dirancang untuk menampilkan daftar seluruh permohonan berdasarkan status tertentu, seperti permohonan yang sudah disetujui kalurahan dan siap diverifikasi akhir. Fungsi ini dibatasi oleh *modifier* onlyDukcapil, sehingga hanya dapat dipanggil oleh alamat *wallet* resmi milik petugas Dukcapil.

```
function getPermohonanForDukcapil(
    PencatatanTypes.Status _status
) external view onlyDukcapil returns (uint256[] memory) {
    return daftarPermohonanPerStatus[_status];
}
```

Gambar 4.14 Fungsi getPermohonanForDukcapil()

Rangkaian fungsi baca (*view functions*) tersebut dieksplorasi sebagai bagian dari rancangan sistem yang tidak hanya menekankan transparansi, tetapi juga efisiensi komputasional pada lingkungan blockchain. Dengan menempatkan logika penyaringan data langsung pada tingkat *smart contract* serta memastikan bahwa hasil *query* selalu selaras dengan peran dan wilayah administratif pemanggil, sistem memungkinkan pengurangan beban pemrosesan pada lapisan dApp sekaligus menjaga konsistensi data yang disajikan. Seluruh fungsi *view* ini tetap berada dalam batasan keamanan karena tidak memodifikasi *state* blockchain, sehingga memungkinkan akses informasi tanpa menimbulkan risiko terhadap integritas data maupun alur proses.

Melalui pendekatan desain tersebut, *smart contract* dieksplorasi tidak hanya sebagai pengelola transaksi dan logika proses, tetapi juga sebagai *verifiable data layer* yang menyediakan sumber data terverifikasi dan dapat diakses secara publik. Peran ini memungkinkan analisis mengenai bagaimana blockchain dapat digunakan sebagai fondasi keandalan, transparansi, dan keterlacakan dalam sistem pencatatan kependudukan terdesentralisasi.

Untuk mengeksplorasi mekanisme penjagaan integritas alur permohonan pada tingkat *smart contract*, sistem ini menerapkan validasi kondisi dan penanganan kesalahan (*error handling*) secara eksplisit. Setiap fungsi utama dilengkapi dengan pemeriksaan kondisi menggunakan perintah require() dan definisi *custom error*, yang memungkinkan kontrak mendeteksi ketidaksesuaian kondisi sebelum perubahan *state* dilakukan. Jika syarat tidak

terpenuhi, kontrak akan melakukan *revert* dan seluruh transaksi dibatalkan, sehingga tidak ada data yang tersimpan dalam keadaan tidak valid pada blockchain..

Berbeda dengan penggunaan *revert message* berbentuk *string*, kontrak ini mendefinisikan sejumlah *custom error* yang dirancang secara eksplisit untuk setiap konteks kesalahan. Pendekatan ini dievaluasi sebagai strategi untuk mengurangi biaya gas melalui representasi error yang lebih ringkas, sekaligus menyediakan struktur penanganan kesalahan yang lebih sistematis dan terstandarisasi. Gambar 4.15 menunjukkan beberapa *custom error* yang digunakan.

```
error BukanPemilikPermohonan();
error TidakDapatDibatalkan();
error PermohonanBukanDiajukan();
error BukanPermohonanPindah();
error TujuanTidakValid();
error IdKalurahanTujuanTidakDikenal();
error BelumDiverifikasiKalurahanAsal();
error HanyaKalurahanTujuan();
error PermohonanPindahBelumDisetujuiKalurahanTujuan();
error PermohonanBelumDisetujuiKalurahan();
error CidKosong();
error BelumAdaDokumenResmi();
error AksesDitolak();
error BelumDisetujuiDukcapil();
error StatusTidakValidUntukUploadDokumen();
error DokumenResmiSudahAda();
```

Gambar 4.15 Kode *Custom Error*

Setiap *error* pada Gambar 4.15 memiliki tujuan spesifik. Misalnya:

a. BukanPemilikPermohonan() dieksplorasi sebagai mekanisme untuk mencegah pihak selain pemohon asli mengubah atau membatalkan permohonan.
b. AksesDitolak() dan HanyaKalurahanTujuan() merepresentasikan penerapan pembatasan wilayah administratif (*region lock*).
c. CidKosong() dieksplorasi sebagai mekanisme untuk memastikan bahwa setiap pengajuan memiliki berkas IPFS yang valid.
d. IdKalurahanTujuanTidakDikenal() dan TujuanTidakValid() digunakan untuk memverifikasi bahwa kode wilayah tujuan telah terdaftar di dalam sistem sebelum pengajuan diproses.

e. PermohonanBelumDisetujuiKalurahan() dan BelumDisetujuiDukcapil() digunakan mencegah transisi status dilakukan sebelum tahap sebelumnya diselesaikan.
f. Sementara StatusTidakValidUntukUploadDokumen() dan DokumenResmiSudahAda() dieksplorasi sebagai mekanisme untuk memastikan bahwa dokumen resmi hanya dapat diunggah sekali, dan hanya ketika permohonan sudah diverifikasi penuh oleh Dukcapil.

Sebagai ilustrasi penerapan pada Gambar 4.16, fungsi submitPermohonan() dieksplorasi sebagai contoh bagaimana mekanisme validasi dan *error handling* diimplementasikan secara terstruktur pada tingkat *smart contract*. Fungsi ini melakukan pemeriksaan berlapis mulai dari keabsahan data yang dikirim, keberadaan wilayah asal dan tujuan, hingga pengisian data NIK kepala keluarga tujuan untuk jenis permohonan pindah gabung KK. Jika salah satu kondisi tidak terpenuhi, kontrak akan menghentikan eksekusi dan mengeluarkan *custom error* yang relevan, seperti CidKosong() ketika berkas IPFS kosong, atau IdKalurahanTujuanTidakDikenal() saat ID wilayah tidak terdaftar. Pendekatan ini memungkinkan eksplorasi mengenai bagaimana validasi berlapis dan *custom error* dapat digunakan untuk menjaga konsistensi *state* serta mencegah terbentuknya data tidak valid dalam sistem kependudukan terdesentralisasi.

```
require(bytes(_cidIPFS).length > 0, CidKosong());
require(
    addressKalurahanById[_idKalurahanAsal] != address(0),
    IdKalurahanTujuanTidakDikenal()
);

if (_jenis == PencatatanTypes.JenisPermohonan.Pindah) {
    require(_idKalurahanTujuan != 0, TujuanTidakValid());
    require(
        addressKalurahanById[_idKalurahanTujuan] != address(0),
        IdKalurahanTujuanTidakDikenal()
    );
}
```

Gambar 4.16 Contoh Penerapan *Error Handling*

Penerapan tersebut dieksplorasi sebagai upaya untuk memastikan bahwa setiap permohonan yang disimpan ke blockchain telah melewati lapisan validasi minimum yang menjamin:

a. Kelengkapan data, seluruh atribut penting (seperti CID dan ID wilayah) wajib diisi;

b. Kebenaran wilayah administrative, hanya kalurahan yang valid dan terdaftar dapat dijadikan asal atau tujuan;
c. Konsistensi alur proses, jenis permohonan harus sesuai dengan logika status awal (Diajukan atau MenungguKonfirmasiKKTujuan).

Hanya setelah seluruh prasyarat terpenuhi, barulah data permohonan disimpan secara permanen di blockchain dan *event* pengajuan dipancarkan (PermohonanDiajukan atau PermohonanPindahDiajukan). Jika salah satu prasyarat gagal, transaksi otomatis dibatalkan tanpa meninggalkan jejak perubahan pada *state* kontrak.

Dengan pendekatan ini, *smart contract* dieksplorasi sebagai mekanisme yang mampu menjaga kebenaran fungsional (*functional correctness*) sekaligus keterandalan data (*data reliability*). Setiap jalur eksekusi di dalam kontrak dilengkapi dengan *guard condition* dan *error type* yang eksplisit, sehingga tahan terhadap penyalahgunaan maupun *input* yang tidak valid. Selain itu, penggunaan *custom error* memungkinkan penyediaan umpan balik yang terstruktur dan mudah diinterpretasikan oleh lapisan aplikasi, sehingga kegagalan pada tingkat on-chain dapat diterjemahkan secara langsung menjadi notifikasi yang jelas bagi pengguna, seperti “ID Kalurahan tidak ditemukan” atau “CID dokumen tidak boleh kosong”.

Sebagai lapisan dasar dari sistem, kontrak KontrolAkses.sol dieksplorasi sebagai pusat pengaturan (*governance layer*) yang mengelola identitas, wilayah administratif, dan hak akses setiap aktor di dalam jaringan pencatatan kependudukan. Kontrak ini tidak menangani logika permohonan secara langsung, melainkan menyediakan parameter sistem, konfigurasi entitas resmi, serta validasi identitas digital yang menjadi fondasi bagi seluruh modul lain, termasuk PencatatanSipil.sol dan PermohonanManager.sol. Pendekatan ini mencerminkan eksplorasi arsitektur modular dengan pemisahan tanggung jawab pada tingkat *smart contract*.

Pada saat kontrak di-*deploy*, akun yang melakukan inisialisasi (msg.sender) otomatis ditetapkan sebagai Dukcapil, yang berperan sebagai otoritas tertinggi dalam sistem. Dukcapil memiliki kewenangan penuh untuk menambah dan menghapus akun Kalurahan yang beroperasi di wilayah administratifnya. Setiap Kalurahan disimpan dalam *mapping* yang ditunjukkan pada Gambar 4.17.

```
mapping(uint8 => address) public addressKalurahanById;
mapping(address => uint8) public idKalurahanByAddress;
mapping(address => bool) public kalurahan;
```

Gambar 4.17 *Mapping* Kalurahan

Struktur ini dieksplorasi sebagai mekanisme pemetaan identitas dua arah antara ID wilayah dan alamat *wallet* Kalurahan, sehingga *smart contract* dapat memastikan bahwa setiap permohonan hanya dapat diverifikasi oleh wilayah asal atau tujuan yang sah. Dukcapil dapat menambahkan wilayah baru melalui fungsi tambahKalurahanById() yang juga memperbarui *mapping* di IPFS dan memancarkan *event* KalurahanAdded. Sebaliknya, wilayah yang sudah tidak aktif dapat dihapus menggunakan fungsi hapusKalurahan(), yang akan menonaktifkan akun Kalurahan serta memperbarui *mapping* secara *onchain* dan *offchain* secara bersamaan. Potongan kode pada Gambar 4.18 menggambarkan sebagian dari fungsi konfigurasi tersebut.

```
function tambahKalurahanById(
    uint8 _id,
    address _akun,
    string calldata _newMappingCID
) external onlyDukcapil {
    require(_akun != address(0), AddressZero());
    require(addressKalurahanById[_id] == address(0), IdSudahDipakai());
    require(
        idKalurahanByAddress[_akun] == 0 && _id != 0,
        AddressSudahDipakai()
    );

    addressKalurahanById[_id] = _akun;
    idKalurahanByAddress[_akun] = _id;
    kalurahan[_akun] = true;
    kalurahanMappingCID = _newMappingCID;

    emit KalurahanAdded(_id, _akun, _newMappingCID);
    emit KalurahanMappingCIDUpdated(_newMappingCID);
}
```

Gambar 4.18 Kode *Fungsi* tambahKalurahanById()

Melalui struktur ini, sistem dieksplorasi untuk menjamin tidak terjadinya duplikasi baik pada ID maupun alamat *wallet* kalurahan, sekaligus memastikan bahwa setiap perubahan konfigurasi dapat ditelusuri melalui *event log* yang tercatat secara permanen pada blockchain..

Kontrak ini juga dieksplorasi sebagai registri identitas digital warga, di mana setiap pengguna yang ingin mengajukan permohonan harus terlebih dahulu melakukan registrasi NIK agar *wallet*-nya dikenali sebagai warga terverifikasi. Proses registrasi ini diimplementasikan

melalui fungsi registerWarga() atau registerWargaAndUpdateMapping(), dengan logika verifikasi ganda untuk mencegah duplikasi identitas.

```
mapping(address => string) public nikByWallet;
mapping(string => address) public walletByNik;
```

Gambar 4.19 Kode *Mapping* NIK dan *Wallet*

Kedua *mapping* pada Gambar 4.19 saling melengkapi, nikByWallet dieksplorasi agar memastikan bahwa satu alamat *wallet* hanya dapat dikaitkan dengan satu NIK, sementara walletByNik mencegah penggunaan NIK yang sama oleh lebih dari satu akun. Jika terjadi pelanggaran, kontrak akan memicu *custom error* NikSudahDiklaim() atau WalletSudahDigunakan().

Seluruh mekanisme *onchain* yang telah dijelaskan sebelumnya berfokus pada eksplorasi pencatatan metadata transaksi dan penegakan logika bisnis secara deterministik di dalam blockchain. Akan tetapi, sebagian besar data kependudukan seperti dokumen hasil pemindaian, berkas JSON permohonan, dan data keluarga tidak efisien jika disimpan langsung di blockchain. Oleh karena itu, penelitian ini melengkapi eksplorasi *smart contract* dengan lapisan *offchain* yang menangani enkripsi, penyimpanan berkas di IPFS, serta komunikasi antaraktor melalui antarmuka aplikasi terdesentralisasi. Implementasi lapisan ini dijelaskan lebih lanjut pada Subbab 4.2.2.

4.1.2 Implementasi *Offchain*

Lapisan *offchain* pada rancangan arsitektur terdesentralisasi ini berperan penting dalam mengelola data kependudukan yang tidak efisien untuk disimpan langsung di blockchain. Seperti diketahui, penyimpanan data besar dalam *smart contact* akan memerlukan biaya gas yang tinggi serta berpotensi menimbulkan masalah skalabilitas. Untuk mengekplorasi hal tersebut, rancangan arsitektur terdesentralisasi ini mengadopsi pendekatan *hybrid*, di mana blockchain berfungsi sebagai penyimpan metadata dan log transaksi yang bersifat permanen, sedangkan dokumen dan berkas kependudukan dikelola di luar rantai blok melalui sistem penyimpanan terdesentralisasi berbasis *InterPlanetary File System* (IPFS). Pendekatan ini bertujuan untuk mengeksplorasi bagaimana untuk tetap mempertahankan prinsip desentralisasi, transparansi, dan ketahanan manipulasi tanpa mengorbankan efisiensi.

Arsitektur lapisan *offchain* dirancang sebagai jembatan antara pengguna, *smart contract*, dan jaringan IPFS. Lapisan ini dibangun menggunakan Node.js dan Ethers.js untuk berinteraksi dengan *smart contract* melalui RPC endpoint, serta Axios untuk mengirim permintaan HTTP ke layanan IPFS melalui Pinata API. Komunikasi dilakukan secara terautentikasi menggunakan JSON *Web Token* (JWT) agar setiap proses unggahan dan pembaruan data dapat diverifikasi dengan aman oleh layanan penyimpanan.

Penyimpanan data permohonan warga dilakukan dengan membentuk berkas JSON yang berisi isian form (identitas, jenis permohonan, *timestamp*, dan metadata *hash* dokumen pendukung). JSON inilah yang kemudian dienkripsi di sisi klien dan diunggah ke IPFS. Berbeda dengan itu, dokumen resmi (misalnya akta/sertifikat hasil penerbitan) merupakan berkas yang dibuat terlebih dahulu oleh petugas Dukcapil. Setelah dokumen final tersedia, petugas mengunggahnya melalui antarmuka dApp, lalu berkas dienkripsi di sisi klien dan barulah diunggah ke IPFS. Hasil unggahan berupa CID dicatat *onchain*.

Sebelum diunggah ke jaringan IPFS, berkas-berkas tersebut terlebih dahulu dienkripsi di sisi klien untuk menjaga kerahasiaan isi dokumen. Enkripsi dilakukan menggunakan algoritma AES-256-CBC. Algoritma ini dipilih karena sederhana, efisien, dan banyak didukung oleh pustaka kriptografi modern, sekaligus mampu memberikan tingkat keamanan yang tinggi untuk data dalam format JSON. Proses enkripsi menghasilkan dua elemen penting, yaitu:

a. *Initialization Vector* (IV) yaitu nilai acak yang menjamin hasil enkripsi berbeda setiap kali meskipun data yang dienkripsi sama, dan
b. *Ciphertext* yaitu hasil dari proses enkripsi data asli menggunakan kunci rahasia (*secret key*).

Algoritma CBC tidak menghasilkan *authentication tag*, sehingga integritas data tidak diperiksa di tingkat enkripsi. Sebagai gantinya, rancangan arsitektur ini mengandalkan CID IPFS yang tercatat di blockchain sebagai bukti integritas, sehingga setiap kali isi data berubah, *hash* CID yang dihasilkan juga akan berbeda, memungkinkan deteksi manipulasi dengan mudah. Potongan kode pada Gambar 4.20 berikut menunjukkan implementasi dasar enkripsi yang digunakan sebelum berkas diunggah ke IPFS.

```
export async function encryptAes256CbcNodeStyle(jsonData, passphrase) {
    // 1. Generate IV
    const iv = window.crypto.getRandomValues(new Uint8Array(16));
    // 2. Derive key
    const key = await scrypt(
```

```
        new TextEncoder().encode(passphrase),
        new TextEncoder().encode('salt'),
        16384, 8, 1, 32
    );
    // 3. Import key
    const cryptoKey = await window.crypto.subtle.importKey(
        'raw',
        key,
        { name: 'AES-CBC' },
        false,
        ['encrypt']
    );
    // 4. Encrypt
    const encoder = new TextEncoder();
    const jsonString = JSON.stringify(jsonData);
    const data = encoder.encode(jsonString);
    const encryptedBuffer = await window.crypto.subtle.encrypt(
        { name: 'AES-CBC', iv },
        cryptoKey,
        data
    );
    // 5. Combine IV + ciphertext, encode to base64
    const result = new Uint8Array(iv.length + encryptedBuffer.byteLength);
    result.set(iv, 0);
    result.set(new Uint8Array(encryptedBuffer), iv.length);
    // Convert to base64
    let binary = '';
    const len = result.byteLength;
    for (let i = 0; i < len; i++) {
        binary += String.fromCharCode(result[i]);
    }
    return btoa(binary);
}
```

Gambar 4.20 Kode Enkripsi Berkas

Sebelum dienkripsi, data permohonan akan dikemas dalam bentuk JSON seperti pada Gambar 4.21.

```
{
  "metadata": {
    "jenisPermohonan": "Pindah",
    "jenisPindah": "2",
```

```
    "timestamp": "2025-07-26T05:24:13.171Z",
    "pemohon": "0x15d34aaf54267db7d7c367839aaf71a00a2c6a65",
    "version": "1.0"
  },
  "dataPindah": {
    "alamatTujuan": "Jl. Cambahan No. 27, Cambahan, Kalurahan NOGOTIRTO,
Kecamatan GAMPING, Kabupaten SLEMAN, Provinsi DAERAH ISTIMEWA YOGYAKARTA",
    "alasanPindah": "Lainnya",
    "alasanPindahLainnya": "asdasda",
    "anggotaPindah": [
      "5796941759783873"
    ],
    "nikKepalaKeluargaBaru": "",
    "nikKepalaKeluargaTujuan": "1635142482592647"
  }
}
```


Gambar 4.21 Contoh Berkas JSON Permohonan

Setelah dienkripsi maka berkas JSON Permohonan tersebut akan terlihat seperti pada Gambar 4.22.


```
BvVTIXa63ZZln1uB6c2vxOh5LnOTUMyU+O2oQnAtMviX5cQaEHxn77G44/izycTMfuj1chPqbRFO9NKOq
IdOVFLNINJHdodBWExkxmyegIkVwoezFWNpy6iMsLXqVEtXm3ZuREX7CEmERTpz0A4HZJW4MvOoQ3lsH0
PKbENcYmeFrorh0I40v04mDMJNX/tETJAIPI5QAWe5FFI3mcH2y99jrSv8V5T21kgLbnnw3e7D6rQAq7H
X3040tJlGZ71gmxJvGkkcDo5fwaV0BolZSqbYWaAaV7+B+Q6K7LkxKJ23MD/KBP0Akm53SAIeDzxhXTDK
t15LgGXqs4wABMQvNs3IJ8sD29Q+cpsD4KSitQ/sgHvEkE5ZOd5GKgM01Z6pTSNovt+Q+KdAPOQO+IMW0
NPVf+rXkCdj8wXXqJEsRw1h8QSkyf8IJMVQ37Y2ZiDnS8VzFn4tVI00uSgzr4UMdkodNwTur32cFe9Dq5
zYZoJopuHB0q5Ag5KwEQtHqS0u4xqBQnpEy3x+EQUGh5+L2jCDMB1pdQ3DM874SaQjECg+X3EQlIXpzPq
+mcqivvj52zKUud4/iu7IVtAddiZKkPLSaK1KaFNWJB+1VyslH9pKbx6B5blx+Zuec3yfqsIp7EZNLGSt
5piyJeFnzavlNuwG8uxTNslOq+4cPlU4Jvo=
```


Gambar 4.22 Berkas JSON Permohonan Setelah Enkripsi

Struktur ini mengeksplorasi agar berkas yang disimpan di jaringan publik IPFS tetap tidak dapat dibaca tanpa kunci enkripsi yang sesuai. Dengan model *client-side encryption* seperti ini, blockchain tidak pernah menerima data pribadi dalam bentuk terbuka, melainkan hanya CID yang menunjuk pada berkas terenkripsi.

```
   const generateUUID = () => {
            if (crypto.randomUUID) {
                return crypto.randomUUID();
```

```
            }
            // Fallback untuk browser lama
            return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g,
function (c) {
                const r = Math.random() * 16 | 0;
                const v = c == 'x' ? r : (r & 0x3 | 0x8);
                return v.toString(16);
            });
        };
        const filename = `${generateUUID()}.enc`;
        const cidIPFS = await uploadToPinata(encryptedData, filename);
        const uploadEndTime = Date.now();
```

Gambar 4.23 Kode Pemberian Nama Berkas

Setelah proses enkripsi selesai, berkas terenkripsi diberi nama berkas dengan *randomUUID* kemudian dikirim ke IPFS menggunakan *endpoint* milik Pinata API seperti yang terlihat pada Gambar 4.23. Pinata dieksplorasi karena menyediakan autentikasi berbasis JWT untuk keamanan tingkat aplikasi. Potongan kode pada Gambar 4.24 berikut menunjukkan mekanisme pengunggahan data ke IPFS secara terprogram.

```
export async function uploadToPinata(encryptedData, filename = 'file.enc') {
    const startTime = Date.now();

    const url = 'https://api.pinata.cloud/pinning/pinFileToIPFS';
    const formData = new FormData();
    const blob = new Blob([encryptedData], { type: 'text/plain' });
    formData.append('file', blob, filename);

    try {
        const response = await fetch(url, {
            method: 'POST',
            headers: {
                Authorization: PINATA_JWT
            },
            body: formData
        });

        const responseTime = Date.now() - startTime;

        if (!response.ok) {
            const err = await response.text();
```

```
            throw new Error('Gagal upload ke Pinata: ' + err);
        }

        const data = await response.json();
        const totalTime = Date.now() - startTime;

        return data.IpfsHash; // CID baru
    } catch (error) {
        const totalTime = Date.now() - startTime;
        throw error;
    }
}
```

Gambar 4.24 Kode untuk Unggah Berkas ke IPFS

Setelah unggahan berhasil, layanan Pinata mengembalikan CID, yaitu *hash* unik yang merepresentasikan berkas tersebut di jaringan IPFS. CID tersebut kemudian disimpan di blockchain melalui fungsi submitPermohonan() pada atribut cidIPFS dalam struktur Permohonan. Dengan mekanisme ini, blockchain hanya menyimpan *pointer* menuju lokasi data tanpa menyimpan isi berkas, sehingga menjaga privasi sekaligus efisiensi biaya transaksi. Apabila isi data diubah, CID yang dihasilkan juga berubah, menjadikan sistem ini tahan terhadap manipulasi.

Selain untuk berkas permohonan, mekanisme penyimpanan *offchain* juga dieksplorasi untuk digunakan pada tahap akhir proses verifikasi, ketika Dukcapil menerbitkan dokumen resmi hasil pelayanan (seperti akta kelahiran, surat kematian, atau surat pindah). Dokumen ini tidak disimpan dalam bentuk JSON, melainkan berupa berkas digital yang dibuat terlebih dahulu oleh petugas Dukcapil di luar sistem blockchain. Setelah dokumen siap, petugas mengunggahnya melalui antarmuka aplikasi, di mana berkas tersebut dienkripsi di sisi klien menggunakan AES-256-CBC, kemudian diunggah ke IPFS.

Hasil unggahan berupa CID dicatat ke dalam *smart contract* melalui *mapping* cidDokumenResmi, yang memetakan setiap idPermohonan ke CID dokumen resminya. Dengan demikian, blockchain tidak menyimpan dokumen secara langsung, melainkan hanya menyimpan referensi yang terverifikasi dan *immutable* terhadap berkas terenkripsi di IPFS. Setiap unggahan dokumen baru memicu *event* DokumenResmiDiunggah, yang berfungsi sebagai jejak audit digital atas publikasi dokumen oleh Dukcapil. Melalui mekanisme ini, setiap permohonan memiliki dua entitas data yang terhubung;

a. cidIPFS, yang menunjuk ke data permohonan warga dalam bentuk JSON terenkripsi, dan

b. cidDokumenResmi[id], yang menunjuk ke dokumen hasil verifikasi akhir dalam bentuk file digital terenkripsi.

Kedua data ini dapat diverifikasi publik melalui CID, namun isi berkas tetap terlindungi karena hanya dapat didekripsi oleh pihak yang memiliki kunci sah.

Untuk mengeksplorasi konsistensi antara data *onchain* dan *offchain*, setiap perubahan berkas menghasilkan CID baru. Perubahan tersebut dicatat dalam blockchain melalui *event*, sehingga sistem menerapkan konsep *immutability with versioning*, di mana seluruh versi historis dokumen tetap dapat dilacak, memungkinkan analisis mengenai bagaimana blockchain dapat digunakan tidak hanya untuk menjaga ketidakberubahan data, tetapi juga untuk mengelola perubahan data secara terverifikasi. Selain itu, data administratif seperti *mapping* NIK–*wallet* dan daftar kalurahan aktif juga dieksplorasi untuk disimpan di IPFS dan direferensikan melalui nikMappingCID dan kalurahanMappingCID pada kontrak KontrolAkses.sol. Setiap pembaruan data ini memicu *event* baru (NikMappingCIDUpdated atau KalurahanMappingCIDUpdated) sehingga semua pihak dapat menelusuri riwayat pembaruan secara transparan.

Pendekatan ini dieksplorasi sebagai kombinasi keunggulan antara blockchain dan IPFS:

a. Efisiensi penyimpanan, karena hanya *hash* yang disimpan di blockchain;
b. Keamanan dan privasi, karena berkas disimpan dalam bentuk terenkripsi;
c. Auditabilitas penuh, karena setiap versi CID dicatat secara permanen dan dapat diverifikasi publik pada blockchain; dan
d. Desentralisasi nyata, karena data tidak bergantung pada satu entitas atau arsitektur terpusat.

Dengan demikian, implementasi *offchain* ini dieksplorasi tidak sekadar sebagai solusi teknis untuk mengatasi keterbatasan blockchain, tetapi juga sebagai bagian dari desain arsitektural yang memperkuat prinsip integritas dan kedaulatan data publik. Integrasi antara blockchain dan IPFS memungkinkan analisis mengenai bagaimana sistem pencatatan sipil digital dapat dirancang agar efisien, aman, transparan, serta selaras dengan visi desentralisasi layanan publik di tingkat kalurahan.

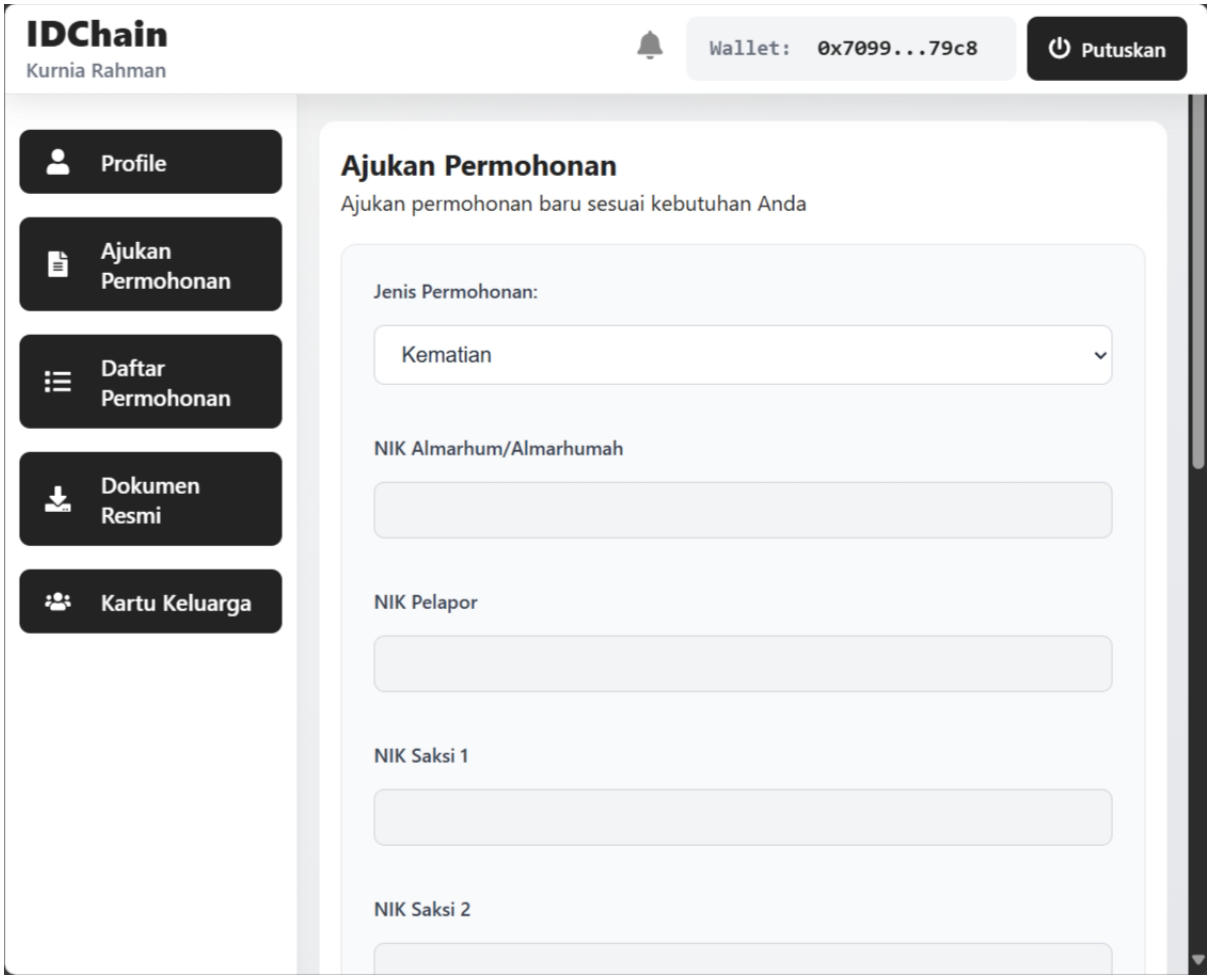
4.1.3 Implementasi *Frontend*

Antarmuka pengguna atau *frontend* dalam sistem ini dieksplorasi sebagai lapisan interaksi utama antara pengguna dengan infrastruktur terdesentralisasi yang berada di bawahnya, yaitu *smart contract* dan IPFS. Lapisan ini tidak sekadar berfungsi sebagai media

visual, tetapi sebagai komponen arsitektural yang menerjemahkan mekanisme blockchain yang kompleks seperti transaksi, validasi peran, dan pengambilan data *on-chain* menjadi alur interaksi yang intuitif bagi pengguna. *Frontend* dirancang untuk melayani tiga peran utama, yaitu warga sebagai pemohon, petugas kalurahan sebagai verifikator tingkat pertama, dan petugas Dukcapil sebagai verifikator akhir sekaligus penerbit dokumen resmi. Melalui pendekatan ini, seluruh proses pengajuan, verifikasi, dan penerbitan dokumen kependudukan dapat dijalankan tanpa menuntut pengguna berinteraksi langsung dengan detail teknis blockchain, karena kompleksitas tersebut ditangani secara otomatis oleh sistem di lapisan belakang.

Implementasi *frontend* dikembangkan menggunakan React.js sebagai *framework* utama untuk membangun komponen antarmuka yang reaktif dan modular. Interaksi dengan blockchain dieksplorasi melalui pemanfaatan Ethers.js sebagai *bridge* antara aplikasi web dan *smart contract* yang telah di-*deploy* pada jaringan Ethereum lokal menggunakan Hardhat. Selain itu, MetaMask digunakan sebagai penyedia *wallet* dan alat autentikasi pengguna, memungkinkan setiap pengguna untuk terhubung ke blockchain melalui alamat *wallet* masing-masing. Untuk komunikasi dengan lapisan penyimpanan *off-chain*, *frontend* terhubung ke IPFS melalui Pinata API, yang digunakan untuk melakukan unggah dan pengambilan berkas berdasarkan CID yang direferensikan oleh *smart contract*.

Mekanisme autentikasi dalam *frontend* dieksplorasi melalui koneksi *wallet* MetaMask yang bersifat otomatis dan terdesentralisasi. Ketika aplikasi diakses, sistem memverifikasi koneksi *wallet* dan mengambil alamat *wallet* aktif sebagai identitas digital permanen pengguna. Peran pengguna ditentukan dengan memeriksa alamat *wallet* terhadap data peran yang tersimpan di kontrak KontrolAkses.sol. Dengan begitu sistem dapat memverifikasi apakah *wallet* tersebut memiliki hak akses sebagai warga, petugas kalurahan, atau petugas Dukcapil. Dengan pendekatan ini, kontrol akses tidak lagi bergantung pada sistem autentikasi terpusat berbasis *username* dan *password*, melainkan sepenuhnya ditentukan oleh logika *on-chain* yang bersifat transparan dan dapat diverifikasi publik.



Gambar 4.25 Antarmuka Formulir Pengajuan Permohonan

Setelah proses autentikasi berbasis *wallet* berhasil, *frontend* berfungsi sebagai media eksplorasi interaksi peran pengguna dengan *smart contract* sesuai hak akses yang dimilikinya. Bagi warga sebagai pemohon, antarmuka menampilkan formulir digital yang merepresentasikan berbagai jenis peristiwa pencatatan sipil yang didukung oleh sistem, yaitu kelahiran, kematian, perkawinan, perceraian, dan pindah domisili, sebagaimana ditunjukkan pada Gambar 4.25. Pada tahap ini, frontend tidak hanya berperan sebagai alat input data, tetapi juga sebagai lapisan praproses sebelum data masuk ke ekosistem terdesentralisasi.

Data yang diisi warga akan dikonversi menjadi berkas JSON, kemudian dienkripsi di sisi klien menggunakan AES-256-CBC. Pendekatan ini mengeksplorasi penerapan *client-side encryption* untuk memastikan bahwa data sensitif tidak pernah tersimpan dalam bentuk terbuka, baik di server perantara maupun di jaringan penyimpanan terdistribusi. Berkas terenkripsi tersebut selanjutnya diunggah ke IPFS melalui Pinata API, dan sistem memperoleh CID unik sebagai representasi kriptografis dari data tersebut. CID yang dihasilkan kemudian

dikirim ke *smart contract* melalui fungsi submitPermohonan(). Proses pengiriman data ke blockchain ditangani oleh Ethers.js seperti pada Gambar 4.26. Dengan alur ini, *frontend* berperan sebagai penghubung antara data *off-chain* yang terenkripsi dan pencatatan *on-chain* yang bersifat *immutable*, sehingga memperlihatkan bagaimana lapisan aplikasi web dapat mengorkestrasi integrasi antara kriptografi, IPFS, dan *smart contract* dalam konteks sistem pencatatan sipil terdesentralisasi.

```
async submitPermohonan(jenis, cidIPFS, idKalurahanAsal, idKalurahanTujuan = 0,
jenisPindah = 0, nikKepalaKeluargaTujuan = '') {
        if (!this.contract) {
            throw new Error('Contract not initialized');
        }
        try {
            // Check if wallet is registered
            const walletAddress = await this.signer.getAddress();
            const nik = await this.contract.nikByWallet(walletAddress);
            if (!nik || nik === '') {
                throw new Error('Wallet belum terdaftar. Silakan register
terlebih dahulu.');
            }

            // Check kalurahan mapping
            const kalurahanAsalAddress = await
this.contract.addressKalurahanById(idKalurahanAsal);
                idKalurahanAsal,
                kalurahanAsalAddress
            });

            if (kalurahanAsalAddress ===
'0x0000000000000000000000000000000000000000') {
                throw new Error(`Kalurahan asal dengan ID ${idKalurahanAsal}
tidak terdaftar`);
            }

            // Check kalurahan tujuan if it's a pindah permohonan
            if (jenis === 4 && idKalurahanTujuan !== 0) { // 4 =
JenisPermohonan.Pindah
                const kalurahanTujuanAddress = await
this.contract.addressKalurahanById(idKalurahanTujuan);
                if (kalurahanTujuanAddress ===
'0x0000000000000000000000000000000000000000') {
```

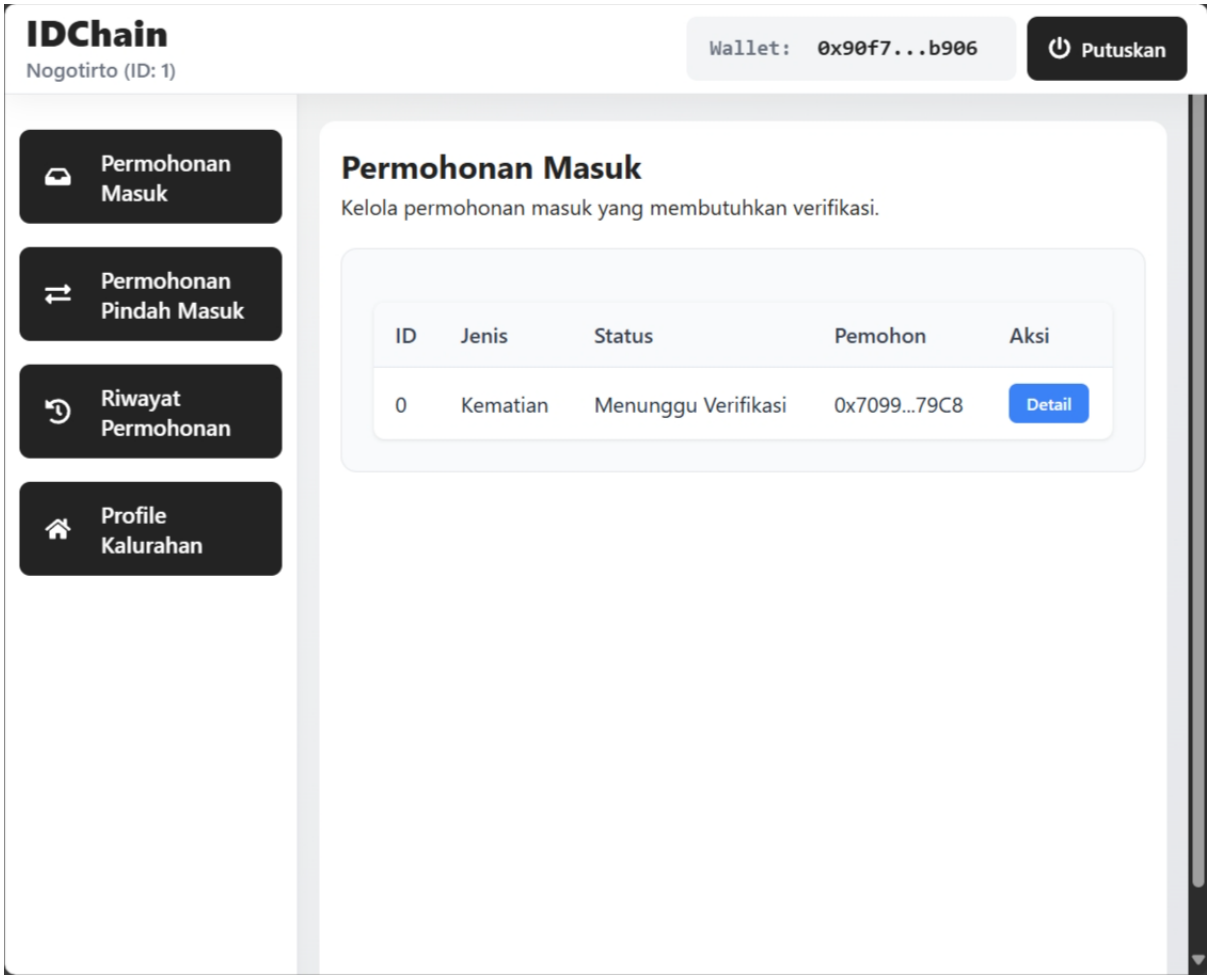
```
                    throw new Error(`Kalurahan tujuan dengan ID
${idKalurahanTujuan} tidak terdaftar`);
                }
            }

            // Hash NIK kepala keluarga tujuan jika ada
            let nikKepalaKeluargaTujuanHash =
'0x0000000000000000000000000000000000000000000000000000000000000000';
            if (nikKepalaKeluargaTujuan && nikKepalaKeluargaTujuan.trim() !== '')
{
                nikKepalaKeluargaTujuanHash = hashNIK(nikKepalaKeluargaTujuan);
            }

            const tx = await this.contract.submitPermohonan(
                jenis,
                cidIPFS,
                idKalurahanAsal,
                idKalurahanTujuan,
                jenisPindah,
                nikKepalaKeluargaTujuanHash
            );
            const receipt = await tx.wait();
            return {
                success: true,
                transactionHash: receipt.hash
            };
        } catch (error) {
            const errorMessage = handleContractError(error);
            throw new Error(errorMessage);
        }
    }
```

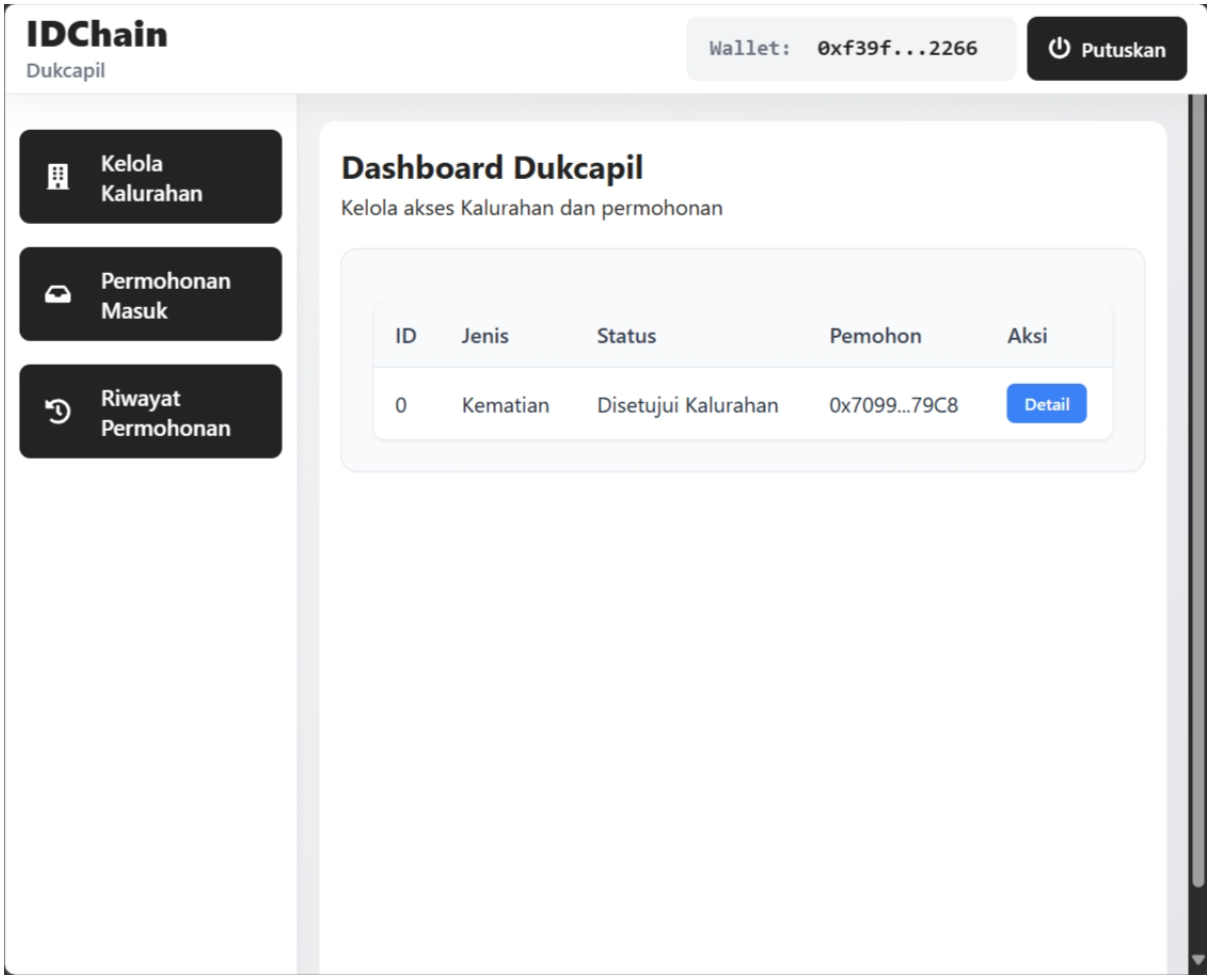
Gambar 4.26 Kode Fungsi submitPermohonan() Ethers.js

Melalui alur ini, warga tidak perlu berinteraksi langsung dengan blockchain atau memahami konsep teknis seperti *gas fee* dan *nonce*, seluruh proses disederhanakan melalui antarmuka web yang ramah pengguna. Setiap transaksi ditandatangani otomatis oleh MetaMask dan hasilnya dapat diverifikasi publik melalui *event* blockchain.



Gambar 4.27 Antarmuka Petugas Kalurahan

Petugas kalurahan mengakses antarmuka verifikasi yang menampilkan daftar permohonan yang masuk ke wilayahnya seperti pada Gambar 4.27. Data permohonan diambil langsung dari blockchain melalui fungsi getPermohonanByKalurahanAsal() atau getPermohonanBelumVerifikasiKalurahan(). Setelah membuka detail permohonan, petugas dapat meninjau dokumen yang tersimpan di IPFS, memeriksa kelengkapan data, dan memberikan keputusan untuk menyetujui atau menolak. Keputusan tersebut kemudian dikirim kembali ke blockchain menggunakan fungsi verifikasiKalurahan(). *Smart contract* secara otomatis memperbarui status permohonan dan memancarkan *event* VerifikasiKalurahan, yang kemudian ditangkap oleh *frontend* melalui mekanisme *event listener* dan ditampilkan sebagai perubahan status *real-time* pada antarmuka pengguna.



Gambar 4.28 Antarmuka Petugas Dukcapil

Bagi Dukcapil, antarmuka berfungsi untuk memeriksa hasil verifikasi kalurahan serta mengunggah dokumen resmi hasil pelayanan seperti pada Gambar 4.28. Petugas Dukcapil dapat mengunggah berkas melalui *form* khusus, di mana file tersebut dienkripsi terlebih dahulu di sisi klien menggunakan algoritma AES-256-CBC, lalu diunggah ke IPFS menggunakan Pinata API. Proses pengunggahan dokumen resmi hasil pelayanan dilakukan menggunakan fungsi yang sama pada Gambar 4.24.

Setelah CID dokumen resmi diperoleh dari IPFS, petugas Dukcapil bisa melakukan verifikasi kemudian *frontend* akan memanggil fungsi verifikasiDukcapil() pada *smart contract*. Fungsi ini nantinya akan menyimpan CID ke dalam *mapping* cidDokumenResmi, dan *event* DokumenResmiDiunggah dipancarkan sebagai bukti penerbitan dokumen digital yang sah. Mekanisme ini mengeksplorasi konsep jejak digital yang tidak dapat dimodifikasi (*immutable audit trail*), di mana keberadaan dan keabsahan dokumen tidak bergantung pada basis data terpusat, melainkan pada bukti kriptografis yang tercatat permanen di blockchain.

Selain fitur pengajuan dan verifikasi, *frontend* juga menyediakan fungsi *tracking* untuk memantau perkembangan status permohonan. Warga dapat memanggil fungsi getPermohonanIDsByPemohon() untuk menampilkan daftar permohonan miliknya, kemudian memeriksa status masing-masing melalui getPermohonan(). Setiap perubahan status, seperti dari Diajukan ke Disetujui Kalurahan atau Disetujui Dukcapil, dapat terbaca secara langsung karena *frontend* memantau *event* blockchain melalui Ethers.js.

Dari sisi desain, antarmuka *frontend* dirancang untuk mengeksplorasi bagaimana teknologi blockchain dan kriptografi dapat diadopsi oleh pengguna non-teknis tanpa menimbulkan beban kognitif. Seluruh kompleksitas teknis seperti transaksi, *gas fee*, enkripsi, dan komunikasi dengan jaringan terdesentralisasi disembunyikan di balik interaksi sederhana berupa tombol dan formulir. Pendekatan ini tidak dimaksudkan untuk mengaburkan teknologi, melainkan untuk menguji sejauh mana sistem terdesentralisasi dapat dioperasikan dalam konteks layanan publik sehari-hari tanpa mengorbankan transparansi dan keamanan data.

Secara keseluruhan, lapisan *frontend* tidak hanya berfungsi sebagai antarmuka visual, tetapi juga sebagai *orchestration layer* yang menghubungkan seluruh komponen sistem mulai dari proses enkripsi data, komunikasi dengan IPFS, hingga interaksi langsung dengan *smart contract*. Melalui implementasi ini, penelitian menempatkan arsitektur terdesentralisasi sebagai objek eksplorasi teknis untuk mengkaji bagaimana blockchain, *smart contract*, dan IPFS dapat diorkestrasi melalui lapisan *frontend* dalam skenario layanan publik, sekaligus mengevaluasi peran *frontend* sebagai penghubung utama antara kompleksitas infrastruktur terdesentralisasi dan interaksi pengguna.

4.2 Hasil Pengujian Sistem

4.2.1 Pengujian *Smart Contract*

Proses pengujian dilakukan menggunakan *framework* Hardhat dengan bantuan *library* Chai.js untuk penyusunan *assertion*. Seluruh kontrak dikompilasi pada *compiler* Solidity versi 0.8.28 dan dijalankan pada jaringan lokal Hardhat. Metode pengujian dilakukan secara *unit testing*, di mana setiap fungsi utama diuji secara terpisah, serta *integration testing*, untuk menguji interaksi antar kontrak seperti PencatatanSipil.sol dan KontrolAkses.sol.

Setiap skenario diinisialisasi lewat beforeEach, mula-mula kontrak utama di-*deploy*, kalurahan ditambahkan melalui tambahKalurahanById, dan registerWarga dipanggil untuk menyiapkan aktor warga. Pengujian mencakup *unit test* per fungsi dan *integration test* lintas

tahapan proses, termasuk jalur khusus permohonan pindah (melibatkan Kalurahan Asal, Kalurahan Tujuan, dan Dukcapil).

Seperti yang terlihat pada Gambar 4.29, bahwa seluruh 26 skenario pengujian lulus pada jaringan local Hardhat. Ringkasan kondisi dan hasil uji disajikan dalam Tabel 4.3.

```
azzra@DESKTOP-3L2R12Q:~/idchain$ npx hardhat test test/pencatatanSipilTest.js


  PencatatanSipil
    ✓ warga dapat mengajukan permohonan
    ✓ id permohonan harus bertambah satu per submit
    ✓ kalurahan dapat memverifikasi permohonan
    ✓ dukcapil dapat menolak permohonan dengan alasan
    ✓ mengembalikan status permohonan sebagai string
    ✓ mengembalikan jenis permohonan sebagai string
    ✓ gagal jika bukan kalurahan yang memverifikasi
    ✓ warga tidak boleh memverifikasi permohonan
    ✓ kalurahan tidak boleh memverifikasi jika status bukan Diajukan
    ✓ emit event saat permohonan diajukan
    ✓ dukcapil dapat menyetujui permohonan
    ✓ warga tidak bisa memverifikasi sebagai dukcapil
    ✓ emit event saat verifikasi kalurahan
    ✓ emit event saat verifikasi dukcapil
    ✓ warga dapat membatalkan permohonan yang diajukan
    ✓ tidak bisa membatalkan permohonan yang bukan milik sendiri
    ✓ kalurahan dapat mengambil daftar permohonan asalnya
    ✓ kalurahan dapat mengambil permohonan yang belum diverifikasi dengan status
tertentu
    ✓ dukcapil dapat mengambil permohonan dengan status tertentu
    ✓ dukcapil dapat mengunggah dan warga dapat mengambil dokumen resmi
    ✓ dukcapil dapat verifikasi dengan dokumen resmi dalam satu transaksi
    ✓ gagal upload dokumen resmi jika sudah ada
    ✓ tidak bisa mengambil dokumen resmi jika belum diunggah
    ✓ jumlahPermohonan bertambah setiap submit

  Fitur Permohonan Pindah
    ✓ A1. Submit permohonan pindah berhasil
    ✓ A2. Gagal: ID Kalurahan Tujuan tidak valid
```

```
    ✓ A3. Gagal: Tidak mengisi CID
    ✓ B4. Verifikasi Kalurahan Asal berhasil (disetujui)
    ✓ B5. Verifikasi Kalurahan Asal ditolak
    ✓ B6. Gagal: Verifikasi oleh kalurahan bukan asal
    ✓ B7. Gagal: Verifikasi saat status bukan Diajukan
    ✓ B8. Gagal: Verifikasi tanpa ID Kalurahan Tujuan (disetujui)
    ✓ C9. Verifikasi Kalurahan Tujuan berhasil (disetujui)
    ✓ C10. Verifikasi Kalurahan Tujuan ditolak
    ✓ C11. Gagal: Verifikasi oleh kalurahan bukan tujuan
    ✓ C12. Gagal: Verifikasi saat status bukan DisetujuiKalurahanAsal
    ✓ D13. Pemohon melihat daftar permohonan miliknya
    ✓ D14. Kalurahan Asal melihat permohonan dari wilayahnya
    ✓ D15. Kalurahan Tujuan melihat permohonan masuk
    ✓ D16. Gagal: Kalurahan melihat permohonan bukan wilayahnya
    ✓ dukcapil dapat memverifikasi permohonan pindah setelah disetujui kalurahan
tujuan

  KontrolAkses - registerWarga
    ✓ berhasil register warga baru
    ✓ gagal jika NIK sudah diklaim wallet lain
    ✓ gagal jika wallet sudah digunakan

  PencatatanSipil - onlyWargaTerdaftar
    ✓ gagal submitPermohonan jika belum registerWarga
    ✓ berhasil submitPermohonan setelah registerWarga
    ✓ gagal batalkanPermohonan jika belum registerWarga


  47 passing (2s)
```

Gambar 4.29 Hasil Pengujian *Smart Contract*

Tabel 4.3 Ringkasan Skenario Uji dan Hasil

No	Fokus Uji	Fungsi/Bagian	Kondisi Awal	Ekspektasi	Hasil
1	Pengajuan permohonan	submitPermohonan	Warga terdaftar, data valid	ID bertambah, status=Diajukan, cidIPFS tercatat	Lulus

2	*Auto-increment* ID	submitPermohonan	Dua *submit* berturut-turut	ID 0, lalu 1 (bukan lompat)	Lulus
3	Verifikasi Kalurahan	verifikasiKalurahan	Status=Diajukan	Status menjadi DisetujuiKalurahan	Lulus
4	Penolakan Dukcapil	verifikasiDukcapil(tolak)	Status=DisetujuiKalurahan	status=DitolakDukcapil, alasan tersimpan	Lulus
5	*String helper*	getStatusPermohonan/ getJenisPermohonan	Setelah *submit*	Mengembalikan label *string* yang benar	Lulus
6	Akses ilegal	verifikasiKalurahan oleh warga	Bukan peran Kalurahan	*Revert* OnlyKalurahan	Lulus
7	*Idempoten* status	verifikasiKalurahan (ulang)	Sudah bukan Diajukan	*Revert* PermohonanBukanDiajukan	Lulus
8	*Event* submit	submitPermohonan	—	*Emit* PermohonanDiajukan dengan argumen tepat	Lulus
9	Persetujuan Dukcapil	verifikasiDukcapil(setuju)	Status=DisetujuiKalurahan	status=DisetujuiDukcapil	Lulus
10	Akses Dukcapil	verifikasiDukcapil oleh warga	Bukan peran Dukcapil	Revert OnlyDukcapil	Lulus
11	*Event* verifikasi	VerifikasiKalurahan/VerifikasiDukcapil	—	Event keluar dengan argumen tepat	Lulus
12	Pembatalan oleh pemohon	batalkanPermohonan	Permohonan milik sendiri, belum diverifikasi	status=DibatalkanPemohon	Lulus
13	Akses illegal pembatalan permohonan	batalkanPermohonan	Permohonan bukan milik sendiri	*Revert* BukanPemilikPermohonan	Lulus

14	*Query* per peran	getPermohonanByKaluraha nAsal/Tujuan, getPermohonanForDukca pil	Data tersedia	Mengembalikan daftar permohonan sesuai *filter*	Lulus
15	Dokumen resmi	verifikasiDukcapil/ungga hDokumenResmi/getDok umenResmi	Status=Disetujui Kalurahan	CID tersimpan di *mapping*, *event* DokumenResmi Diunggah	Lulus
16	Cegah duplikasi dokumen	unggahDokumenResmi(ul ang)	Sudah ada CID	*Revert* DokumenResmiS udahAda	Lulus
17	Larangan ambil dokumen jika belum ada	getDokumenResmi	Belum diunggah	*Revert* BelumAdaDoku menResmi	Lulus
18	Fitur pindah—submit	submitPermohonan(Pinda h)	Asal & tujuan valid	Jenis= Pindah, masuk ke indeks wilayah & status	Lulus
19	Fitur pindah—validasi input	submitPermohonan(Pinda h)	Tujuan 0/*unknown*	*Revert* TujuanTidakVali d/IdKalurahanTu juanTidakDikena l	Lulus
20	Verifikasi Kalurahan Asal	verifikasiKalurahanAsalPi ndah	Status=Diajukan	Status=Disetujui KalurahanAsal/D itolak	Lulus
21	Verifikasi Kalurahan Tujuan	verifikasiKalurahanTujua nPindah	Status=Distujui Kalurahan Asal	Status=Disetujui KalurahanTujuan /Ditolak	Lulus
22	*Gatekeeping* tujuan	verifikasiKalurahanTujua nPindah	Bukan kalurahan tujuan	*Revert* HanyaKalurahan Tujuan	Lulus
23	Finalisasi Dukcapil (pindah)	verifikasiDukcapil	Status=Disetujui Kalurahan Tujuan	Status=Disetujui Dukcapil	Lulus

24	Registrasi warga (akses)	KontrolAkses.registerWarga	NIK belum dipakai	*Event* WargaTerdaftar, mapping sesuai.	Lulus
25	Proteksi duplikasi registrasi	registerWarga	Nik/*Wallet* sudah terpakai	*Revert* NikSudahDiklaim/*Wallet*SudahDigunakan	Lulus
26	*Gatekeeping* onlyWargaTerdaftar	submitPermohonan/batalkanPermohonan	Belum daftar	*Revert* OnlyWargaTerdaftar	Lulus

Berdasarkan hasil pengujian unit dan integrasi, seluruh fungsi pada kontrak PencatatanSipil.sol telah berjalan sesuai rancangan. Proses verifikasi bertingkat dari Kalurahan hingga Dukcapil menunjukkan bahwa setiap perubahan status permohonan hanya dapat dilakukan oleh pihak berwenang. Seluruh *event* penting berhasil dipancarkan sebagai bukti transparansi transaksi di blockchain, mendukung prinsip auditabilitas dan *non-repudiation*.

Selain itu, hasil pengujian menunjukkan bahwa mekanisme pengelolaan dokumen resmi berbasis CID IPFS bekerja dengan benar. Sistem mencegah pengunggahan ganda dan menolak akses terhadap dokumen yang belum tersedia, memastikan integritas arsip digital kependudukan.

Selain menampilkan hasil akhir seluruh skenario pengujian, beberapa cuplikan kode pengujian berikut disertakan untuk memberikan gambaran konkret mengenai cara pengujian dilakukan pada lingkungan Hardhat menggunakan *framework* Chai.js. Setiap cuplikan dirancang untuk mewakili jenis pengujian yang berbeda, mulai dari pengujian fungsi utama, pengendalian hak akses, validasi logika status, hingga pencatatan *event onchain* sebagai mekanisme audit.

Sebagai contoh, potongan kode pada Gambar 4.30 menunjukkan pengujian terhadap fungsi submitPermohonan(), yang bertugas mencatat permohonan baru dari warga dan mengatur status awal ke Diajukan.

```
    it("warga dapat mengajukan permohonan", async () => {
        const tx = await pencatatan.connect(warga).submitPermohonan(0,
"cid_json_xxx", 1, 0, 0, ethers.ZeroHash);
        await tx.wait();
```

```
        const ids = await pencatatan.getPermohonanIDsByPemohon(warga.address);
        expect(ids.length).to.equal(1);

        const data = await pencatatan.getPermohonan(ids[0]);
        expect(data.pemohon).to.equal(warga.address);
        expect(data.cidIPFS).to.equal("cid_json_xxx");
        expect(data.status).to.equal(0); // Diajukan
    });
```

Gambar 4.30 Kode Pengujian submitPermohonan()

Pengujian ini memastikan bahwa setiap warga terdaftar dapat mengajukan permohonan dengan benar, dan bahwa *smart contract* secara otomatis menambahkan entri baru dengan status awal yang sesuai. Selain itu, pengujian ini juga memverifikasi bahwa data cidIPFS tersimpan dan dapat diambil kembali oleh pemohon, menegaskan keterpaduan antara data *onchain* dan metadata *offchain*.

Contoh pada Gambar 4.31 menguji fungsi verifikasiKalurahan(), yang hanya boleh dijalankan oleh akun dengan peran kalurahan. Pengujian ini memastikan transisi status hanya terjadi jika proses dilakukan oleh pihak yang berwenang.

```
    it("kalurahan dapat memverifikasi permohonan", async () => {
        await pencatatan.connect(warga).submitPermohonan(0, "cid_json_xxx", 1, 0,
0, ethers.ZeroHash);
        const ids = await pencatatan.getPermohonanIDsByPemohon(warga.address);

        const tx = await
pencatatan.connect(kalurahan).verifikasiKalurahan(ids[0], true, "");
        await tx.wait();

        const updated = await pencatatan.getPermohonan(ids[0]);
        expect(updated.status).to.equal(1); // DisetujuiKalurahan
```

```
    });

    it("gagal jika bukan kalurahan yang memverifikasi", async () => {
        await pencatatan.connect(warga).submitPermohonan(0, "cid_xx", 1, 0, 0,
ethers.ZeroHash);
        const ids = await pencatatan.getPermohonanIDsByPemohon(warga.address);

        await expect(
            pencatatan.connect(warga).verifikasiKalurahan(ids[0], true, "")
        ).to.be.revertedWithCustomError(pencatatan, "OnlyKalurahan");
    });
```

Gambar 4.31 Kode Pengujian verifikasiKalurahan()

Melalui pengujian tersebut, terbukti bahwa fungsi verifikasiKalurahan() dapat melakukan pergantian status ke Disetujui Kalurahan dan *smart contract* berhasil menegakkan kontrol akses berbasis peran yaitu OnlyKalurahan, serta memvalidasi perubahan status sesuai alur proses yang telah dirancang.

Selanjutnya, pengujian terhadap *event* blockchain dilakukan untuk memastikan bahwa setiap aksi penting meninggalkan jejak digital (*audit trail*) melalui mekanisme *event*. Gambar 4.32 contoh pengujian terhadap *event* VerifikasiDukcapil yang dipancarkan ketika permohonan disetujui oleh Dukcapil.

```
    it("emit event saat verifikasi dukcapil", async () => {
        await pencatatan.connect(warga).submitPermohonan(2, "cid_dukcapil", 1, 0,
0, ethers.ZeroHash);
        const ids = await pencatatan.getPermohonanIDsByPemohon(warga.address);

        await pencatatan.connect(kalurahan).verifikasiKalurahan(ids[0], true,
"");

        await expect(
            pencatatan.connect(dukcapil).verifikasiDukcapil(ids[0], true, "",
"QmXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
"QmYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY")
        )
            .to.emit(pencatatan, "VerifikasiDukcapil")
            .withArgs(ids[0], dukcapil.address, true, "", anyValue);
    });
```

Gambar 4.32 Kode Pengujian *Emit Event* Verifikasi Dukcapil

Event tersebut menjadi bukti otentik transaksi di blockchain dan dapat diverifikasi secara publik, memperkuat prinsip transparansi dan *non-repudiation* dalam sistem.

Terakhir, pengujian pada Gambar 4.33 menguji mekanisme keamanan kontrak dengan mencoba memanggil fungsi verifikasi menggunakan akun yang tidak memiliki hak akses. Hasil pengujian ini menunjukkan bahwa kontrak berhasil menolak aksi tidak sah dan menampilkan *custom error* sesuai yang telah didefinisikan.

```
    it("warga tidak boleh memverifikasi permohonan", async () => {
        await pencatatan.connect(warga).submitPermohonan(0, "cid_xx", 1, 0, 0,
ethers.ZeroHash);
        const ids = await pencatatan.getPermohonanIDsByPemohon(warga.address);

        await expect(
            pencatatan.connect(warga).verifikasiKalurahan(ids[0], true, "")
        ).to.be.revertedWithCustomError(pencatatan, "OnlyKalurahan");
    });
```

Gambar 4.33 Kode Pengujian Kontrol Akses

Dengan demikian, rangkaian pengujian ini memberikan gambaran empiris mengenai bagaimana *smart contract* merespons berbagai bentuk interaksi yang melanggar batas peran, serta bagaimana mekanisme kontrol akses bekerja sebagai pengaman logika *onchain*.

Berdasarkan seluruh hasil pengujian yang telah dilakukan, *smart contract* PencatatanSipil.sol menunjukkan perilaku yang sesuai dengan rancangan fungsional yang telah dijabarkan pada Bab 3. Setiap fungsi utama, mulai dari pengajuan permohonan, verifikasi bertingkat oleh Kalurahan dan Dukcapil, hingga penerbitan dokumen resmi dieksekusi sesuai urutan yang ditentukan, dengan status dan *event* yang muncul sebagai representasi eksplisit dari perubahan *state onchain*. Hal ini memungkinkan peneliti untuk mengamati bagaimana alur administratif dapat dimodelkan sebagai *state machine* yang berjalan secara otomatis di dalam *smart contract*.

Pengujian terhadap kontrol akses berbasis peran menunjukkan bahwa *modifier* seperti onlyKalurahan, onlyDukcapil, dan onlyWargaTerdaftar secara konsisten menolak transaksi dari alamat yang tidak berwenang, yang tercermin dari pemunculan *custom error* seperti OnlyKalurahan dan OnlyDukcapil. Temuan ini menegaskan bahwa *smart contract* mampu menegakkan batas otoritas secara mandiri tanpa bergantung pada validasi eksternal, sekaligus

memperlihatkan bagaimana struktur kelembagaan administratif dapat direpresentasikan dalam bentuk aturan logika yang tidak dapat diubah secara sepihak.

Selain itu, pengujian pada mekanisme transisi status memperlihatkan bahwa perubahan *state* permohonan hanya dapat terjadi secara berurutan dan tidak dapat melompati tahap tertentu. Permohonan, misalnya, hanya dapat diverifikasi oleh Dukcapil setelah melalui persetujuan Kalurahan, dan tidak dapat mengalami perubahan lanjutan setelah mencapai status final. Pola ini menunjukkan bahwa implementasi *state machine* dalam kontrak berhasil mencegah terjadinya kondisi ambigu atau manipulatif, serta memberikan kepastian bahwa setiap perubahan *state* selalu berada dalam jalur yang telah ditentukan.

Dari sisi transparansi, setiap aksi penting dalam alur kontrak berhasil menghasilkan *event onchain* dengan parameter yang sesuai, seperti ID permohonan, identitas pelaku, dan waktu eksekusi. *Event-event* ini membentuk jejak aktivitas yang dapat diamati secara publik, sehingga pengujian tidak hanya berfokus pada hasil akhir fungsi, tetapi juga pada bagaimana setiap langkah terekam sebagai log yang dapat diaudit. Hal ini memperlihatkan peran *event* sebagai mekanisme observabilitas utama dalam sistem terdesentralisasi.

Pengujian juga memberikan wawasan mengenai integrasi antara data *onchain* dan *offchain* melalui IPFS. Kontrak hanya mengizinkan pengambilan CID dokumen resmi apabila dokumen tersebut telah diunggah secara sah oleh Dukcapil, dan menolak percobaan unggah ulang melalui error DokumenResmiSudahAda. Temuan ini menunjukkan bagaimana *smart contract* dapat digunakan untuk menjaga konsistensi referensi data *offchain*, meskipun berkas fisik tidak disimpan langsung di dalam blockchain..

Secara keseluruhan, hasil pengujian ini dimaknai sebagai eksplorasi terhadap kemampuan *smart contract* dalam menegakkan aturan, merekam proses, dan menyediakan jejak audit yang tidak dapat dimanipulasi. Pengujian ini menunjukkan bahwa blockchain dapat berfungsi sebagai lapisan eksekusi dan observasi proses administratif, di mana setiap tindakan dan keputusan menjadi bagian dari histori *onchain* yang dapat diverifikasi secara terbuka. Temuan ini memperkuat posisi penelitian sebagai kajian eksploratif terhadap penerapan teknologi blockchain dalam konteks pencatatan kependudukan, khususnya dalam memahami implikasi teknis, batasan, dan potensi desain sistem terdesentralisasi untuk layanan publik.

4.2.2 Pengujian *Black-Box*

Pengujian *black-box* pada penelitian ini dilakukan sebagai sarana eksplorasi terhadap perilaku sistem terdesentralisasi ketika *smart contract* dipicu melalui lapisan *frontend*. Fokus

pengujian tidak diarahkan pada implementasi kode internal, melainkan pada relasi antara masukan yang diberikan pengguna melalui antarmuka dan keluaran yang dihasilkan oleh sistem *onchain* dan *offchain*. Dengan pendekatan ini, pengujian *black-box* digunakan untuk mengamati bagaimana interaksi pengguna mulai dari autentikasi menggunakan *wallet* MetaMask hingga akses terhadap dokumen resmi diterjemahkan menjadi transaksi blockchain, pemanggilan fungsi *smart contract*, serta perubahan status dan *event* yang dapat diamati secara publik.

Pengujian dilakukan melalui antarmuka aplikasi web yang terhubung dengan jaringan pengujian Hardhat serta sistem penyimpanan IPFS. Setiap pengujian dilakukan menggunakan akun dengan peran yang berbeda, yaitu warga, petugas kalurahan, dan petugas Dukcapil, untuk mengeksplorasi bagaimana sistem merespons input yang sama maupun berbeda berdasarkan konteks otoritas pemanggil. Data hasil pengujian dicatat dalam Tabel 4.4 berikut yang merupakan pelaksanaan dari rancangan pengujian yang telah disusun pada Bab 3.

Tabel 4.4 Hasil Pengujian *Black-Box*

No	Fitur	Deskripsi	Keluaran	Hasil Aktual	Status
1	Login MetaMask	Autentikasi pengguna dengan *wallet*	*Wallet* terhubung dan alamat pengguna terdeteksi	Sistem berhasil menampilkan alamat *wallet* pengguna di sudut antarmuka dan status koneksi berubah menjadi "Terhubung"	Berhasil
2	Pengajuan Permohonan	Warga mengisi formulir dan mengunggah dokumen	Notifikasi sukses, permohonan tercatat di blockchain, pemohon bisa melihat status permohonannya "Diajukan"	Transaksi berhasil, *event* PermohonanDiajukan tercatat di blockchain dan status muncul sebagai "Diajukan" di *dashboard* pengguna	Berhasil
3	Verifikasi Kalurahan	Petugas Kalurahan menyetujui permohonan	Status berubah menjadi Disetujui Kalurahan	Transaksi verifikasi sukses, status dan waktu verifikasi tercatat, *event* VerifikasiKalurahan dipancarkan	Berhasil

4	Verifikasi Dukcapil	Petugas Dukcapil menyetujui permohonan dan menerbitkan dokumen	Status berubah menjadi Disetujui Dukcapil dan dokumen terunggah	Dokumen terenkripsi berhasil diunggah ke IPFS, CID tersimpan pada *mapping* cidDokumenResmi, dan *event* DokumenResmiDiunggah tercatat	Berhasil
5	Pelacakan Status	Pemohon memeriksa status permohonan	Status tampil sesuai pembaruan terakhir di blockchain	Status pada antarmuka sinkron dengan hasil getPermohonan() di *smart contract*, pembaruan terlihat secara *real-time* setelah konfirmasi transaksi	Berhasil
6	Unduh Dokumen	Pemohon mengunduh dokumen hasil layanan	File hasil dekripsi tampil benar dan dapat dibuka	Dokumen berhasil didekripsi menggunakan kunci AES-256-CBC di sisi klien, dan isi file valid dan identik dengan data hasil penerbitan Dukcapil	Berhasil

Hasil pengujian *black-box* menunjukkan bahwa interaksi pengguna melalui antarmuka *frontend* secara konsisten diterjemahkan menjadi rangkaian operasi *onchain* dan *offchain* sesuai dengan rancangan sistem. Proses autentikasi MetaMask berjalan stabil tanpa perlu pemuatan ulang halaman, dan data alamat *wallet* tersimpan otomatis untuk identifikasi pengguna. Pada tahap pengajuan permohonan, *frontend* berhasil menghasilkan berkas JSON terenkripsi, mengunggahnya ke IPFS, dan menyimpan CID ke blockchain dengan memicu pemanggilan fungsi submitPermohonan() pada *smart contract*, hal tersebut memungkinkan pengamatan langsung terhadap bagaimana input pengguna direpresentasikan sebagai transaksi blockchain.

Pengujian juga memperlihatkan bahwa mekanisme verifikasi bertingkat dapat diamati secara jelas dari sisi antarmuka. Aksi verifikasi hanya dapat dilakukan oleh akun dengan peran yang sesuai, sehingga percobaan interaksi dari peran yang tidak sah tidak menghasilkan perubahan *state*. Integrasi antara *frontend*, *smart contract*, dan IPFS berjalan sinkron. Perubahan status di blockchain segera tercermin pada tampilan pengguna tanpa keterlambatan

signifikan, dan semua *event* seperti PermohonanDiajukan, VerifikasiKalurahan, serta DokumenResmiDiunggah dapat diverifikasi melalui *log* transaksi pada konsol pengujian, memperlihatkan bagaimana *event* berfungsi sebagai penghubung antara lapisan eksekusi *onchain* dan lapisan presentasi aplikasi. Seluruh file dokumen resmi yang diunduh melalui antarmuka juga berhasil didekripsi di sisi klien menggunakan algoritma AES-256-CBC, memastikan kerahasiaan data selama proses penyimpanan dan transmisi.

Secara keseluruhan, pengujian *black-box* ini dimaknai sebagai eksplorasi terhadap bagaimana teknologi blockchain, *smart contract*, IPFS, dan *wallet* berbasis kriptografi berperilaku ketika diakses melalui antarmuka web. Hasil pengujian menunjukkan bahwa *frontend* mampu berfungsi sebagai media pemicu sekaligus observasi terhadap perilaku sistem terdesentralisasi, di mana setiap interaksi pengguna dapat ditelusuri dampaknya hingga ke level transaksi dan *event onchain*. Temuan ini memperkuat pemahaman mengenai keterkaitan antara pengalaman pengguna dan karakteristik teknis sistem terdesentralisasi dalam konteks pencatatan kependudukan.

4.2.3 Pengujian Kinerja Teknis

Secara ringkas, hasil pengukuran menunjukkan pola sebagai berikut. Proses *deployment* kontrak utama PencatatanSipil mengonsumsi gas sekitar 3.925.545 unit dengan durasi eksekusi sekitar 14–22 ms. Nilai tersebut wajar mengingat proses *deploy* mencakup inisialisasi struktur data dan penyimpanan kode kontrak ke blockchain. Operasi konfigurasi awal seperti tambahKalurahanById memerlukan sekitar 118.702 gas dengan durasi sekitar 2–4 ms, sedangkan pendaftaran warga melalui registerWarga mengonsumsi sekitar 70.000–70.300 gas dengan durasi sekitar 2–5 ms.

Untuk operasi yang langsung berkaitan dengan alur layanan, pengajuan permohonan kependudukan oleh warga melalui fungsi submitPermohonan menunjukkan kebutuhan gas sekitar 221.100–221.300 unit untuk permohonan umum, dengan durasi eksekusi sekitar 3–7 ms. Pada kasus permohonan pindah penduduk, nilai gas sedikit lebih tinggi, yaitu sekitar 248.204 gas per transaksi akibat tambahan pencatatan informasi kalurahan tujuan dan jenis pindah, namun durasi eksekusinya tetap berada pada kisaran 3–5 ms. Hal ini mengindikasikan bahwa penambahan atribut dan pencatatan *mapping* tambahan masih berada dalam batas kompleksitas yang efisien.

Tahap verifikasi juga menunjukkan pola penggunaan gas yang relatif moderat. Verifikasi kalurahan melalui fungsi verifikasiKalurahan mengonsumsi sekitar 108.346 gas dengan durasi

2–4 ms, sedangkan pada skenario permohonan pindah, verifikasi kalurahan asal (verifikasiKalurahanAsalPindah) memerlukan sekitar 113.176 gas ketika disetujui dan sekitar 133.722 gas ketika ditolak (karena adanya pencatatan alasan penolakan). Verifikasi kalurahan tujuan (verifikasiKalurahanTujuanPindah) mengonsumsi sekitar 107.880 gas untuk kasus disetujui dan sekitar 130.897 gas untuk kasus ditolak. Nilai ini menunjukkan bahwa meskipun terdapat beberapa percabangan alur dan pembaruan status, proses verifikasi di tingkat kalurahan tetap efisien dari sisi biaya gas.

Pada tahap verifikasi akhir oleh Dukcapil, fungsi verifikasiDukcapil menampilkan dua karakteristik berbeda. Untuk kasus penolakan permohonan, transaksi mengonsumsi sekitar 130.380–130.392 gas dengan durasi 3–4 ms, sedangkan untuk kasus persetujuan yang sekaligus mencatat dokumen resmi dan memicu *event* terkait, kebutuhan gas meningkat menjadi sekitar 230.269–230.457 gas dengan durasi 3–5 ms. Peningkatan ini dapat dipahami karena dalam satu transaksi kontrak perlu memperbarui status permohonan, mencatat alamat verifikator, menyimpan CID dokumen resmi ke dalam *mapping* cidDokumenResmi, serta memancarkan lebih dari satu *event*. Meskipun demikian, nilai tersebut masih berada pada rentang yang wajar untuk transaksi yang bersifat komposit.

Operasi pendukung lain seperti batalkanPermohonan juga diukur dan menunjukkan kebutuhan gas sekitar 126.394 unit dengan durasi sekitar 2–3 ms. Hal ini mengonfirmasi bahwa penambahan jalur terminasi yang aman (pembatalan oleh pemohon) tidak menimbulkan *overhead* gas yang signifikan. Seluruh durasi eksekusi untuk operasi-operasi utama berada pada kisaran 2–9 ms di lingkungan Hardhat, yang mengindikasikan bahwa kompleksitas logika kontrak tidak menyebabkan perlambatan berarti pada level eksekusi. Pada jaringan publik, waktu total yang dirasakan pengguna tentu akan dipengaruhi oleh waktu konfirmasi blok, namun kontribusi dari logika kontrak itu sendiri relatif kecil.

Untuk memperjelas hasil pengujian, Tabel 4.5 berikut menyajikan ringkasan konsumsi gas dan durasi eksekusi dari berbagai fungsi utama dalam *smart contract*. Nilai diperoleh dari hasil eksekusi fungsi secara berulang pada beberapa skenario yang berbeda, misalnya eksekusi normal (berhasil), eksekusi dengan hasil penolakan, dan eksekusi yang memicu *revert* karena pelanggaran aturan akses atau status.

Tabel 4.5 Ringkasan Hasil Pengukuran Biaya Gas dan Durasi Eksekusi *Smart Contract*

No	Operasi	Biaya Gas (unit)	Durasi Eksekusi (ms)	Keterangan
1	*Deploy* PencatatanSipil	3.925.545	14–22	Inisialisasi kontrak utama dan struktur data
2	tambahKalurahanById	118.702	2–4	Menambahkan alamat kalurahan ke daftar wilayah
3	registerWarga	70.281	2–5	Klaim identitas dengan mendaftarkan alamat *wallet* warga
4	submitPermohonan (umum)	221.183	3–7	Warga mengajukan permohonan layanan
5	submitPermohonan (pindah)	248.204	3–5	Permohonan dengan tambahan data asal–tujuan
6	verifikasiKalurahan	108.346	2–4	Persetujuan/penolakan kalurahan
7	verifikasiKalurahanAsalPindah	113.176	2–3	Verifikasi asal pada permohonan pindah
8	verifikasiKalurahanTujuanPindah	107.880	2–3	Verifikasi tujuan pada permohonan pindah
9	verifikasiDukcapil (penyetujuan dan penerbitan dokumen)	230.363	3–5	Verifikasi akhir dan unggah dokumen resmi
10	verifikasiDukcapil (ditolak)	130.386	3–4	Penolakan dengan alasan tertentu
11	batalkanPermohonan	126.394	2–3	Pembatalan oleh pemohon

Berdasarkan hasil pengukuran yang diperoleh, seluruh fungsi utama *smart contract* mulai dari pengajuan permohonan, verifikasi bertingkat, pembatalan, hingga pencatatan dokumen resmi menunjukkan konsumsi gas yang berada di bawah 250.000 unit per transaksi, di luar biaya *deploy* kontrak. Temuan ini digunakan sebagai indikator empiris untuk mengamati dampak desain struktur data, penggunaan *mapping*, serta pemanfaatan *event* terhadap biaya eksekusi *onchain*. Nilai gas yang relatif stabil ini menunjukkan bahwa penambahan mekanisme auditabilitas, kontrol akses berbasis wilayah, dan pengelolaan *state* tidak secara signifikan meningkatkan kompleksitas eksekusi pada tingkat EVM.

Selain itu, variasi durasi dan konsumsi gas antar operasi yang relatif kecil memberikan gambaran bahwa logika kontrak dieksekusi secara konsisten terlepas dari jenis interaksi yang dilakukan. Hal ini mengindikasikan bahwa pendekatan modular dan penggunaan struktur data yang eksplisit memungkinkan *smart contract* mempertahankan performa yang terprediksi, meskipun harus menangani beberapa cabang alur proses seperti permohonan umum dan permohonan pindah.

Secara keseluruhan, pengujian kinerja teknis ini memberikan wawasan mengenai karakteristik biaya dan performa dari *smart contract* yang dirancang sebagai refleksi terhadap implikasi desain teknis yang diambil. Hasil pengujian ini melengkapi pengujian *smart contract* pada subbab sebelumnya dengan menambahkan dimensi evaluasi kinerja, sehingga penelitian ini tidak hanya mengeksplorasi apa yang dapat dilakukan oleh teknologi blockchain dalam konteks pencatatan kependudukan, tetapi juga bagaimana teknologi tersebut berperilaku dari sisi efisiensi dan skalabilitas. Dengan demikian, subbab ini menegaskan posisi penelitian sebagai kajian eksploratif terhadap batasan dan potensi *smart contract* dalam mendukung proses administratif yang kompleks secara terdesentralisasi.

4.2.4 Validasi Kesesuaian Sistem

Berdasarkan pengujian unit dan integrasi menggunakan *framework* Hardhat dan Chai.js, seluruh mekanisme utama yang dikodekan dalam *smart contract* dapat dieksekusi secara konsisten di blockchain. Fungsi-fungsi seperti pengajuan permohonan, verifikasi berlapis oleh Kalurahan dan Dukcapil, unggah dokumen resmi, serta pelacakan status menunjukkan perilaku yang selaras dengan logika yang dirancang. Selain skenario normal, pengujian juga memperlihatkan bagaimana kontrak merespons interaksi yang tidak sah, seperti upaya verifikasi oleh alamat yang tidak berwenang atau pengajuan permohonan tanpa registrasi warga terlebih dahulu. Respons berupa penolakan transaksi dan *revert error* yang terkontrol memberikan indikasi bahwa mekanisme pembatasan dan validasi berjalan sebagaimana dirancang.

Dari sisi perilaku fungsional, hasil pengujian menunjukkan bahwa mekanisme utama yang menjadi fokus eksplorasi dapat diamati secara jelas melalui perubahan *state* dan *event* onchain. Beberapa temuan utama yang tervalidasi antara lain:

a. Pengajuan permohonan oleh warga melalui antarmuka dApp berhasil memicu transaksi blockchain yang mencatat data permohonan dan CID dokumen IPFS. *Event*

PermohonanDiajukan atau PermohonanPindahDiajukan berfungsi sebagai indikator bahwa data telah tersimpan secara permanen di jaringan.

b. Mekanisme verifikasi bertingkat menunjukkan bahwa *smart contract* mampu menegakkan pembagian otoritas secara deterministik. Kalurahan hanya dapat memverifikasi permohonan dari wilayah yang sesuai, sementara Dukcapil hanya dapat melakukan verifikasi akhir setelah tahap Kalurahan terpenuhi. Upaya pemanggilan fungsi oleh alamat yang tidak sah secara konsisten ditolak.
c. Penyimpanan dokumen resmi dilakukan melalui IPFS dengan referensi CID yang tercatat di blockchain. *Mapping* cidDokumenResmi berperan sebagai penghubung antara data *offchain* dan transaksi *onchain*, memperlihatkan bagaimana integrasi blockchain-IPFS dapat digunakan untuk menjaga konsistensi referensi data.
d. Setiap perubahan status permohonan menghasilkan *event* spesifik seperti VerifikasiKalurahan, VerifikasiDukcapil, dan DokumenResmiDiunggah. *Event-event* ini membentuk jejak perubahan *state* yang dapat digunakan untuk penelusuran riwayat dan audit proses secara publik.
e. Fungsi *view* seperti getPermohonan, getPermohonanIDsByPemohon, dan getPermohonanForDukcapil mampu menampilkan data status langsung dari blockchain tanpa perantara. Hal ini menunjukkan bahwa *state onchain* dapat digunakan sebagai sumber kebenaran tunggal (*single source of truth*) bagi antarmuka aplikasi.

Seluruh mekanisme tersebut menunjukkan kesesuaian antara perilaku sistem dan skenario yang dirancang pada tahap analisis, tanpa ditemukan perilaku menyimpang pada pengujian yang sah.

Selain perilaku fungsional, validasi juga dilakukan terhadap karakteristik non-fungsional yang menjadi fokus utama eksplorasi teknologi dalam penelitian ini. Observasi terhadap hasil pengujian menunjukkan beberapa temuan berikut:

a. Keamanan data tercermin dari keberhasilan proses enkripsi dan dekripsi dokumen menggunakan AES-256-CBC. Dokumen yang diunggah ke IPFS tidak dapat diakses dalam bentuk terbuka dan hanya dapat dipulihkan kembali dengan kunci yang sah, menunjukkan pemisahan yang jelas antara penyimpanan data dan kontrol akses.
b. Keterlacakan (*auditability*) dapat diamati dari pencatatan seluruh transaksi dan *event log* di blockchain. Setiap tahapan proses, termasuk identitas pelaku dan waktu eksekusi, tercatat secara permanen dan tidak dapat dimodifikasi.

c. Efisiensi proses terlihat dari penyederhanaan alur administrasi melalui otomatisasi verifikasi dan pencatatan berbasis *smart contract*. Meskipun penilaian dilakukan secara kualitatif, hasil pengujian menunjukkan bahwa alur yang dieksplorasi dapat dijalankan tanpa ketergantungan pada pertukaran dokumen fisik atau validasi manual berulang.
d. Transparansi dan akuntabilitas tercapai melalui kemampuan sistem untuk menampilkan status permohonan yang bersumber langsung dari data *onchain*. Setiap perubahan status hanya dapat terjadi melalui transaksi baru, sehingga tidak memungkinkan adanya perubahan tersembunyi.

Pengujian kinerja teknis yang telah dibahas pada subbab sebelumnya turut memperkuat validasi ini, dengan menunjukkan bahwa konsumsi gas dan waktu eksekusi berada dalam kisaran yang stabil untuk kontrak dengan kompleksitas multi-aktor. Temuan ini memberikan konteks tambahan mengenai implikasi biaya dan performa dari desain *smart contract* yang dieksplorasi.

Dengan demikian, validasi kesesuaian sistem menunjukkan bahwa seluruh mekanisme yang diuji telah memenuhi kriteria eksplorasi sebagaimana dirumuskan pada Bab 3, yaitu konsistensi eksekusi logika *smart contract*, keterlacakan transaksi, integritas data *offchain*, dan kemampuan menjalankan alur layanan secara digital. Hasil validasi ini tidak dimaknai sebagai klaim keberhasilan implementasi sistem operasional, melainkan sebagai bukti empiris bahwa teknologi blockchain, *smart contract*, dan IPFS memiliki karakteristik yang relevan untuk dimodelkan dalam konteks pencatatan kependudukan terdesentralisasi. Ringkasan keterkaitan antara tujuan penelitian dan hasil validasi disajikan pada Tabel 4.6.

Tabel 4.6 Ketercapaian Tujuan Penelitian

Tujuan Penelitian	**Hasil**	**Ketercapaian**
Merancang arsitektur sistem terdesentralisasi berbasis blockchain yang mampu mengelola proses pengajuan, verifikasi, dan penerbitan dokumen kependudukan di tingkat kalurahan secara efisien dan transparan.	Arsitektur sistem berhasil direalisasikan dalam bentuk purwarupa yang memanfaatkan mekanisme *state transition* pada *smart contract* untuk merepresentasikan alur pengajuan dan verifikasi. Hasil pengujian menunjukkan bahwa alur tersebut dapat dieksekusi secara konsisten di blockchain dan diamati melalui perubahan *state* serta *event log*.	Tercapai

Mengimplementasikan *smart contract* sebagai logika utama dalam mendukung transparansi proses, integritas data, serta pengendalian akses pada proses pengajuan dan verifikasi permohonan pencatatan sipil.	Implementasi *smart contract* berhasil menunjukkan bahwa pencatatan transaksi, kontrol akses berbasis peran dan wilayah, serta perubahan status permohonan dapat dilakukan secara deterministik dan tidak dapat dimodifikasi tanpa jejak. Pengujian membuktikan bahwa mekanisme ini dapat digunakan dalam menjaga integritas data dan transparansi proses.	Tercapai
Mengeksplorasi integrasi *InterPlanetary File System* (IPFS) sebagai mekanisme penyimpanan dokumen kependudukan secara *off-chain*, serta mengkaji karakteristik efisiensi dan ketahanan data yang dihasilkan.	Integrasi IPFS berhasil diuji dengan skema penyimpanan dokumen terenkripsi menggunakan AES-256-CBC. CID yang dihasilkan dapat direferensikan melalui blockchain dan digunakan kembali untuk proses pengambilan dokumen, menunjukkan karakteristik desentralisasi, ketahanan data, dan pemisahan beban penyimpanan dari jaringan blockchain.	Tercapai
Menguji performa penerapan teknologi blockchain dan IPFS dalam konteks sistem kependudukan yang dikembangkan	Hasil pengujian menunjukkan bahwa seluruh fungsi utama *smart contract* dapat dieksekusi dengan konsumsi gas dan waktu eksekusi yang masih berada pada batas realistis untuk konteks eksplorasi teknologi. Hasil ini memberikan gambaran mengenai kelayakan teknis, efisiensi, serta batasan yang perlu dipertimbangkan dalam pengembangan sistem terdesentralisasi untuk layanan publik.	Tercapai

Tabel 4.6 menunjukkan bahwa keempat tujuan penelitian telah tercapai dalam batasan eksplorasi yang ditetapkan. Purwarupa yang dikembangkan berhasil memperlihatkan bagaimana teknologi blockchain, *smart contract*, dan IPFS dapat digunakan untuk membangun alur pengelolaan data kependudukan yang bersifat terdesentralisasi. Hasil pengujian tidak hanya menunjukkan ketercapaian fungsi-fungsi utama, tetapi juga memberikan gambaran

empiris mengenai karakteristik teknis seperti transparansi proses, integritas data, serta efisiensi eksekusi transaksi dalam konteks layanan publik berbasis blockchain.

Namun demikian, hasil eksplorasi ini juga memiliki sejumlah keterbatasan teknis yang penting untuk dicermati. Pertama, seluruh pengujian dilakukan pada lingkungan lokal (Hardhat) sehingga belum merepresentasikan kondisi biaya transaksi dan latensi pada jaringan blockchain publik yang sesungguhnya. Kedua, purwarupa ini belum terintegrasi dengan sistem kependudukan nasional seperti SIAK, sehingga temuan penelitian masih berada pada ranah konseptual dan teknis, bukan implementasi institusional. Ketiga, aspek optimasi gas dan pengelolaan transaksi dalam skala besar belum dievaluasi secara mendalam, sehingga implikasi penerapan pada volume permohonan tinggi masih memerlukan kajian lanjutan. Keempat, penelitian ini berfokus pada eksplorasi perilaku teknis sistem dan belum mencakup pengujian non-teknis seperti studi pengguna, skalabilitas jaringan, maupun analisis beban operasional lintas instansi.

Untuk pengembangan lebih lanjut, implementasi nyata di Indonesia dapat mempertimbangkan penggunaan model blockchain *consortium*, di mana setiap instansi terkait seperti Dukcapil dan Kalurahan menjalankan *node* masing-masing. Dalam model ini, warga dapat berpartisipasi sebagai *node* sukarela, sehingga kedaulatan data pribadi tetap terjaga dan sistem tetap sejalan dengan regulasi nasional mengenai perlindungan data. Pendekatan ini sejalan dengan prinsip pada blockchain *consortium* yang menempatkan entitas terpercaya sebagai validator, memungkinkan pengelolaan *node* secara mandiri oleh instansi pemerintah, serta mendukung partisipasi sukarela dari warga untuk menjaga privasi dan transparansi sistem secara terdesentralisasi (Elisa et al., 2020).

Ke depan, penelitian ini dapat dikembangkan lebih lanjut dengan mengintegrasikan mekanisme *Decentralized Identity* (DID) dan enkripsi berbasis kunci publik untuk memperkuat otorisasi akses antar lembaga, serta penerapan *Layer-2 solutions* (seperti zkRollup atau Optimism) guna menekan biaya transaksi. Selain itu, studi pengguna dan evaluasi tingkat penerimaan masyarakat terhadap sistem terdesentralisasi di sektor pelayanan publik juga menjadi arah potensial untuk penelitian selanjutnya.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil eksplorasi, perancangan, implementasi purwarupa, serta rangkaian pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

a. Penerapan arsitektur terdesentralisasi berbasis blockchain dalam konteks pengelolaan data kependudukan di tingkat kalurahan dapat direalisasikan melalui pemisahan peran antara pencatatan transaksi *on-chain* dan penyimpanan data *off-chain*. Struktur data permohonan, mekanisme *state transition*, serta alur transaksi yang diimplementasikan dalam *smart contract* menunjukkan bahwa proses administrasi kependudukan dapat dimodelkan secara terstruktur dan konsisten di atas blockchain. Hasil eksplorasi ini memperlihatkan bahwa blockchain mampu berperan sebagai lapisan pencatat status dan proses (*process ledger*), bukan sebagai penyimpan data dokumen secara langsung.
b. Implementasi *smart contract* sebagai logika utama sistem terbukti mampu mendukung transparansi, integritas data, serta pengendalian akses dalam proses pengajuan dan verifikasi permohonan pencatatan sipil. Mekanisme kontrol akses berbasis peran yang melibatkan warga, kalurahan, dan Dukcapil dapat ditegakkan secara deterministik oleh *smart contract*, sementara setiap perubahan status permohonan tercatat melalui *event log* yang bersifat *immutable*. Temuan ini menunjukkan bahwa *smart contract* dapat digunakan untuk mengenkapsulasi aturan administratif dan otoritas kelembagaan ke dalam kode yang dapat diverifikasi secara publik.
c. Integrasi *InterPlanetary File System* (IPFS) sebagai media penyimpanan dokumen kependudukan secara terdistribusi berhasil diimplementasikan sebagai bagian dari arsitektur sistem. Proses enkripsi dokumen menggunakan algoritma AES-256-CBC sebelum unggah ke IPFS, serta pencatatan *Content Identifier* (CID) di blockchain, menunjukkan bahwa penyimpanan *off-chain* dapat dikaitkan secara aman dengan transaksi *on-chain*. Hasil pengujian memperlihatkan bahwa pendekatan ini mampu menjaga kerahasiaan, keaslian, dan ketahanan data tanpa ketergantungan pada arsitektur terpusat, sekaligus mengurangi beban penyimpanan pada blockchain.
d. Hasil uji performa teknologi blockchain dan IPFS dalam konteks purwarupa layanan administrasi kependudukan menunjukkan bahwa seluruh fungsi utama dapat dieksekusi dengan konsumsi gas dan waktu eksekusi yang masih berada dalam batas wajar untuk

skala prototipe. Pengujian mengungkap bahwa pencatatan transaksi, verifikasi bertingkat, serta integrasi penyimpanan *off-chain* dapat berjalan secara stabil dan konsisten. Temuan ini memberikan gambaran empiris mengenai karakteristik performa, efisiensi teknis, serta keterbatasan awal penerapan arsitektur terdesentralisasi pada layanan publik.

Dengan demikian, penelitian ini berhasil menjawab seluruh rumusan masalah yang diajukan serta mencapai tujuan penelitian secara menyeluruh. Hasil penelitian menunjukkan bahwa eksplorasi pemanfaatan blockchain, *smart contract*, dan IPFS mampu menghadirkan alternatif pendekatan teknis dalam pengelolaan data kependudukan, sekaligus memberikan kajian mendalam mengenai potensi, karakteristik, dan tantangan penerapan sistem terdesentralisasi.

5.2 Saran

a. Penerapan model *consortium* blockchain.

Pada skala implementasi riil, disarankan untuk menerapkan *consortium* blockchain di mana *node* dikelola oleh instansi terpercaya seperti Dukcapil, kalurahan, dan pemerintah daerah. Model ini memungkinkan distribusi kendali yang seimbang, meningkatkan privasi data warga, serta tetap sejalan dengan regulasi perlindungan data pribadi.

b. Pengembangan mekanisme identitas terdesentralisasi (*Decentralized Identity*/DID).

Integrasi konsep *Decentralized Identity* (DID) dan *verifiable credentials* dapat dieksplorasi lebih lanjut untuk memperkuat model identitas dan otorisasi antar entitas. Pendekatan ini memungkinkan pemisahan antara identitas pengguna dan alamat *wallet* semata, serta membuka peluang penerapan skema verifikasi lintas kontrak dan lintas aplikasi tanpa ketergantungan pada registri terpusat.

c. Eksplorasi optimasi *smart contract* dan manajemen gas

Penelitian lanjutan dapat difokuskan pada eksplorasi teknik optimasi *smart contract*, seperti penyederhanaan struktur data, penggunaan *storage packing*, pemanfaatan *calldata*, serta pengurangan operasi penulisan *state*. Kajian ini penting untuk memahami *trade-off* antara kompleksitas logika, keterbacaan kontrak, dan efisiensi biaya gas dalam konteks sistem dengan banyak aktor dan status transisi.

d. Evaluasi penggunaan Layer-2 dan solusi skalabilitas

Penelitian selanjutnya dapat mengeksplorasi penggunaan *Layer-2 solutions* seperti zkRollup, atau Optimistic Rollup untuk mengkaji dampaknya terhadap biaya transaksi, *throughput*, dan latensi sistem. Eksplorasi ini relevan untuk menilai kelayakan teknis

penerapan arsitektur terdesentralisasi pada skenario dengan volume permohonan yang lebih besar.

DAFTAR PUSTAKA

Addiani, F. A. F. (2023). Blockchain for the Data Storage System on Governmental Organization. *JECE - Journal of Empowerment Community and Education*, *3*(2). https://jurnalpengabdian.com/index.php/jece/article/view/796

Agus Dwiyanto. (2018). *Manajemen Pelayanan Publik:* UGM PRESS.

Agustina, N. G. V., & Hariyoko, Y. (2024). Pemanfaatan Data Kependudukan Dalam Perencanaan Kalurahan Kedung Baruk. *Policy and Maritime Review*, *3*(1), 1–12. https://doi.org/10.30649/pmr.v3i1.55

Aos, A., & Riwanti, R. (2019). PELAKSANAAN SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN (SIAK) DALAM UPAYA MENINGKATKAN KUALITAS INFORMASI PADA DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KABUPATEN CIREBON. *CENDEKIA Jaya*, *1*(1), 1–24. https://doi.org/10.47685/cendekia-jaya.v1i1.6

Argento, L., Buccafurri, F., Furfaro, A., Graziano, S., Guzzo, A., Lax, G., Pasqua, F., & Saccà, D. (2020). ID-Service: A Blockchain-Based Platform to Support Digital-Identity-Aware Service Accountability. *Applied Sciences*, *11*(1), 165. https://doi.org/10.3390/app11010165

Buterin, V. (2014). *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf

Chander, Er. N., Chaudhary, R., Arora, S., Prakash, M., Gajrani, A., & Bahl, A. (2023). Decentralized Healthcare System using Blockchain. *International Journal of Scientific Research in Engineering and Management*, *07*(11), 1–11. https://doi.org/10.55041/ijsrem27214

Damre, J., Kharche, A., Jungade, S., Sanap, V., & A. Bachwani, S. (2022). BLOCKCHAIN: TYPES AND BENEFITS. *International Journal of Current Science*, *12*(1).

Dukcapil Ende. (2024). *Sistem Informasi Administrasi Kependudukan (SIAK)*. Endekab.go.id. https://disdukcapil.endekab.go.id/dukcapil/2020-11-08-02-40-07/tentang-adminduk/317-sistem-informasi-administrasi-kependudukan-siak

Elisa, N., Yang, L., Chao, F., & Cao, Y. (2018). A framework of blockchain-based secure and privacy-preserving E-government system. *Wireless Network*, *29*. https://doi.org/10.1007/s11276-018-1883-0

Elisa, N., Yang, L., Li, H., Chao, F., & Naik, N. (2019). Consortium Blockchain for Security and Privacy-Preserving in E-government Systems. *The 19th International Conference on Electronic Business*, 99–107. https://doi.org/10.48550/arxiv.2006.14234

ethereum.org. (2025). *Technical introduction to dapps*. Ethereum.org. https://ethereum.org/developers/docs/dapps/

Faber, B., Michelet, G. C., Weidmann, N., Mukkamala, R. R., & Vatrapu, R. (2019). BPDIMS: A Blockchain-based Personal Data and Identity Management System. *52nd Hawaii International Conference on System Sciences*, 6855–6864. https://doi.org/10.24251/HICSS.2019.821

Financial Crimes Enforcement Network (FinCEN). (2019). *Application of FinCEN's Regulations to Certain Business Models Involving Convertible Virtual Currencies*. https://www.fincen.gov/system/files/2019-05/FinCEN%20Guidance%20CVC%20FINAL%20508.pdf

Jai, S. A., Setyawan, D., & Adiwidjaja, I. (2016). IMPLEMENTASI SISTEM INFORMASI ADMINISTRASI KEPENDUDUKAN. *JISIP: Jurnal Ilmu Sosial Dan Ilmu Politik*, *5*(1), 34. https://doi.org/10.33366/jisip.v5i1

Kushwaha, S. S., Joshi, S., Singh, D., Kaur, M., & Lee, H.-N. (2022). Ethereum Smart Contract Analysis Tools: A Systematic Review. *IEEE Access*, *10*, 57037–57062. https://doi.org/10.1109/access.2022.3169902

Munarja, M. T. (2014, June 18). *Pemanfaatan Data Siak Dalam Layanan Data Dan Informasi Administrasi Kependudukan Di Kabupaten Gunungkidul*. Dukcapil Gunungkidul. https://dukcapil.gunungkidulkab.go.id/2014/06/18/pemanfaatan-data-siak-dalam-layanan-data-dan-informasi-administrasi-kependudukan-di-kabupaten-gunungkidul/

Murdiani, D., & Sobirin, M. (2022). PERBANDINGAN METODOLOGI WATERFALL DAN RAD (RAPID APPLICATION DEVELOPMENT) DALAM PENGEMBANGAN SISTEM INFORMASI. *Jurnal Informatika Teknologi Dan Sains*, *4*(4), 302–306. https://doi.org/10.51401/jinteks.v4i4.2008

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press.

Nguyen, Q. K. (2016). Blockchain - A Financial Technology for Future Sustainable Development. *2016 3rd International Conference on Green Technology and Sustainable Development (GTSD)*. https://doi.org/10.1109/gtsd.2016.22

Nugroho, E. P., Afrianto, I., Piantari, E., Anisyah, A., Husaeni, A., Bisulthon, I. D., & Jundurrahmaan, I. (2024). Design Blockchain Architecture for Population Data Management to Realize a Smart City in Cimahi, West Java, Indonesia. *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, *9*(4), 1206–1222. https://doi.org/10.26555/jiteki.v9i4.27493

Nugroho, P. A., & Warsono, H. (2012). Evaluasi Penerapan Sistem Informasi Administrasi Kependudukan (SIAK) Pada Dinas Kependudukan Dan Pencatatan Sipil Kota Semarang. *Indonesian Journal of Public Policy and Management Review*, *01*(02). https://media.neliti.com/media/publications/94339-ID-evaluasi-penerapan-sistem-informasi-admi.pdf

Pilkington, M. (2015, September 18). Blockchain Technology: Principles and Applications. *Research Handbook on Digital Transformations*. https://ssrn.com/abstract=2662660

Prata, D. N., Araújo, H. X. de , & Santos, C. (2021). A Literature Review about Smart Contracts Technology. *International Journal of Advanced Engineering Research and Science*, *8*(2), 001–004. https://doi.org/10.22161/ijaers.82.1

Rahma, A. (2021, May 24). *Data Penduduk di BPJS Kesehatan Bocor, Bukti Lemahnya Perlindungan Data Pribadi*. Tempo; TEMPO.CO. https://fokus.tempo.co/read/1465176/data-penduduk-di-bpjs-kesehatan-bocor-bukti-lemahnya-perlindungan-data-pribadi

Rasheed, S., & Louca, S. (2024). Blockchain-Based Implementation of National Census as a Supplementary Instrument for Enhanced Transparency, Accountability, Privacy, and Security. *Future Internet*, *16*(1), 24. https://doi.org/10.3390/fi16010024

Rawat, R., Chougule, R., Singh, S., Dixit, S., & Kadam, A. (2019). Smart Contracts using Blockchain. *International Research Journal of Engineering and Technology*, *06*(06).

Saian, S. D. S., Sembiring, I., & Manongga, D. H. F. (2024). A Prototype of Decentralized Applications (DApps) Population Management System Based on Blockchain and Smart Contract. *JOIV International Journal on Informatics Visualization*, *8*(2), 845–845. https://doi.org/10.62527/joiv.8.2.1861

Setiawan, O. P. (2023, July 10). *Administrasi Publik di Era Disrupsi, Birokrasi Masa Depan Memanfaatkan Big Data "Disintegrasi Data Kependudukan Antara Dinas Kependudukan dan Catatan Sipil dengan BPJS Kesehatan."* https://cilacaptengah.cilacapkab.go.id/?p=26050

Siddharth. (2025, May 30). *A Comparative Study Between the Four Types of Blockchains*.

Amity University Online. https://amityonline.com/blog/types-of-blockchain-compared

Sulistyo, P. D. (2021, August 8). *Lagi-lagi, Problem Integrasi Data Kependudukan*. Kompas.id; Harian Kompas. https://www.kompas.id/baca/polhuk/2021/08/08/lagi-lagi-masalah-integrasi-data-kependudukan

Swan, M. (2015). *Blockchain: Blueprint for a new economy*. "O'Reilly Media, Inc."

Szabo, N. (1997). Formalizing and Securing Relationships on Public Networks. *First Monday*, *2*(9). https://doi.org/10.5210/fm.v2i9.548

Thesing, T., Feldmann, C., & Burchardt, M. (2021). Agile versus Waterfall Project management: Decision Model for Selecting the Appropriate Approach to a Project. *Procedia Computer Science*, *181*(1), 746–756. https://doi.org/10.1016/j.procs.2021.01.227

Trautwein, D., Raman, A., Tyson, G., Castro, I., Scott, W., Schubotz, M., Gipp, B., & Psaras, Y. (2022, August). Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web. *The ACM SIGCOMM 2022 Conference*. https://doi.org/10.1145/3544216.3544232

Wardhana, C. S. (2024). Implementasi Teknologi Blockchain Dalam Optimalisasi Keamanan Database Penduduk Di Kementerian Dalam Negeri. *Action Research Literate*, *8*(4), 642–648. https://doi.org/10.46799/arl.v8i4.305

Wijaya, S. C., Mahendra, A. A., Hamdan, T. N., Ramdan, H., & Aditya, R. (2024). Pengembangan Sistem Informasi Pelayanan Publik untuk Pemerintah Daerah. *MENTARI: Manajemen, Pendidikan Dan Teknologi Informasi*, *3*(1), 40–51. https://doi.org/10.33050/mentari.v3i1

Wu, C., Xiong, J., Xiong, H., Zhao, Y., & Yi, W. (2022). A Review on Recent Progress of Smart Contract in Blockchain. *IEEE Access*, *10*, 50839–50863. https://doi.org/10.1109/ACCESS.2022.3174052

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). Blockchain Technology Overview. *National Institute of Standards and Technology*, *1*(1). https://doi.org/10.6028/nist.ir.8202

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *2017 IEEE International Congress on Big Data (BigData Congress)*, *1*(1), 557–564. https://doi.org/10.1109/BigDataCongress.2017.85

Zieglmeier, V., Daiqui, G. L., & Pretschner, A. (2023). Decentralized Inverse Transparency with Blockchain. *Distributed Ledger Technologies: Research and Practice*, *2*(3), 1–28. https://doi.org/10.1145/3592624

LAMPIRAN